



Руководство пользователя платформы Adv.Cake

1.1 Основные понятия и принципы работы

Эта статья поможет вам, если вы только подключились к нашей системе и хотите разобраться в основных понятиях и принципах работы.

Статья состоит из следующих разделов:

1. *Что такое Adv.Cake?*;
2. *Что такое партнерская программа?*;
3. *Схема работы вебмастера*;
4. *Что такое трафик?*;
5. *Вознаграждение вебмастеру.*

1.1.1 Adv.Cake: основные определения

Adv.Cake – агентство перформанс-маркетинга. Наш основной продукт – аналитическая платформа, позволяющая полностью управлять CPA-каналом.

Вы, скорее всего, встречали понятие «CPA», однако, повторить что это такое, лишним не будет:)

CPA (от англ. Cost Per Action – «цена за действие») – модель оплаты интернет-рекламы с оплатой за действие, т.е. рекламодатель выплачивает вебмастеру вознаграждение только за совершенное действие пользователя, которого вебмастер привел на сайт рекламодателя.

Теперь разберемся с основными участниками в CPA:

1. **Рекламодатель** – организация, которая хочет разместить рекламу в интернете по дополнительному каналу (CPA);

2. **Партнерская сеть** – платформа-посредник, которая помогает рекламодателями и вебмастерам находить друг друга, а также налаживает между ними рабочий процесс;
3. **Вебмастер** – участник сети, который хочет помочь рекламодателю привести пользователей на его сайт при помощи своей площадки или путем арбитража трафика;
4. **Пользователь** – посетитель сайта вебмастера.

1.1.2 Что такое партнерская программа?

В CPA рекламодатели и вебмастеры сотрудничают в рамках партнерской программы.

Партнерская программа или оффер – коммерческое предложение рекламодателя, в котором он указывает:

1. Информацию о ставках, т.е. сколько получит вебмастер за выполненное условие;
2. Краткое описание оффера;
3. Преимущества для покупателей и вебмастеров;
4. Портрет целевой аудитории (ГЕО, Возраст, Регионы, Тип устройства);
5. Типы трафика – где можно размещать рекламу и нужно ли ее согласовывать.

Также в рамках партнерской программы рекламодатель предоставляет вебмастеру рекламные материалы (баннеры, видео), которые он может использовать в работе.

Пример оффера от «Нетологии»:

netology.ru (Активен) ID: [REDACTED]
Категория: Образование

Создать ссылку В статистику

Ставки Только активные

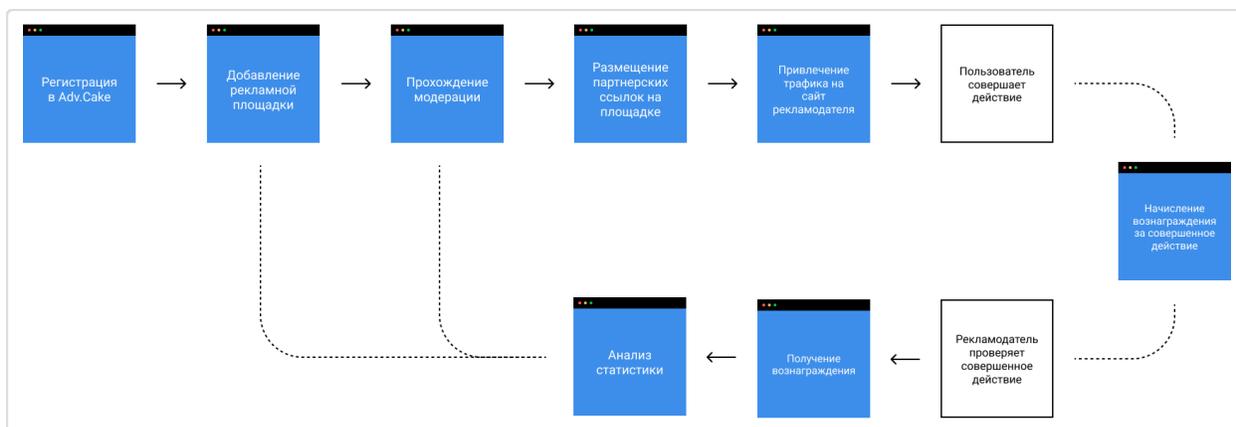
Показаны записи 1-3 из 3.

Ставка	Условия	Период действия
[REDACTED]	[REDACTED]	с 2021-01-26
[REDACTED]	[REDACTED]	с 2016-01-01
[REDACTED]	[REDACTED]	с 2016-01-01

Нетология – это образовательная платформа. Обучаем современным востребованным профессиям в области ИТ и диджитала по направлениям: Маркетинг, Бизнес и управление, Дизайн и UX, Программирование, Аналитика, Soft Skills, MBA.
Помогаем специалистам расти на всех этапах карьеры. Миссия Нетологии – помочь человеку найти свой путь роста и дать ему всё необходимое для развития в карьере: знания,

1.1.3 Схема работы вебмастера в Adv.Cake

Порядок работы вебмастера представлен на схеме.



Для начала работы в Adv.Cake, вам необходимо:

1. Зарегистрироваться в нашей системе;
2. Добавить рекламную площадку;
3. Подключиться к партнерской программе;
4. Сгенерировать партнерскую ссылку и разместить ее на вашей рекламной площадке.

Подсказка: С регистрацией и добавлением площадки вам поможет закрепленный за вами менеджер.

Дальше дело за вами: сгенерируйте партнерскую ссылку и разместите ее на вашей площадке. Также, по условиям оффера, рекламодатель может предоставить вам готовые рекламные материалы, однако, вы можете адаптировать их под разные площадки, если это, конечно, не противоречит условиям оффера.

Когда пользователи будут совершать прописанные в условиях оффера целевые действия, вы будете получать предварительно рассчитанное вознаграждение.

После этого, рекламодатель проверит, оплачен ли заказ и тогда вы сможете вывести вознаграждение на ваш счет или электронный кошелек.

Спустя некоторое время работы с программой, изучите статистику в вашем личном кабинете, чтобы понять, в каком направлении вам стоит двигаться дальше. Если работа с программой складывается успешно, но вы хотите развиваться или наоборот, не удовлетворены результатами, попробуйте поработать с другими офферами или рекламными площадками.

1.1.4 Что такое трафик и где его взять?

Итак, вебмастер регистрируется в нашей системе, подключается к офферу, создает партнерские ссылки и привлекает по ним трафик на сайт рекламодателя.

Возможно, на этом этапе, у вас возник вопрос: что такое трафик и где его взять?

Трафик – поток пользователей, перешедших по партнерской ссылке на сайт рекламодателя.

Задача вебмастера – направить по партнерской ссылке трафик разрешенного типа в различные источники и заинтересовать пользователя, чтобы он перешел на сайт рекламодателя и совершил на нем определенное действие.

Наши вебмастера работают с многими источниками трафика, чтобы узнать какие из них доступны в Adv.Cake, перейдите на страницу «Источники трафика».

1.1.5 За что рекламодатели выплачивают вознаграждение?

Рекламодатель выплачивает вознаграждение, если пользователи, перешли по партнерским ссылкам вебмастера на сайт рекламодателя и совершили там определенные целевые действия.

Целевое действие – желаемые для компании действия клиентов. Они могут быть абсолютно разные, все зависит от направления деятельности компании.

Если это интернет-магазин, то целевым действием может стать оформление заказа, для мобильных приложений – их последующая установка на устройство, а для банков – оформление кредита и т.д. Читайте подробнее о *типах целевых действий*.

1.1.5.1 Типы целевых действий

1.1.5.1.1 Оплаченный заказ

Это один из самых распространенных типов целевого действия. Все просто – вебмастер должен привести на сайт рекламодателя пользователя, который оформит и оплатит заказ или услугу.

1.1.5.1.2 Регистрация

В зависимости от партнерской программы пользователю нужно заполнить форму регистрации на сайте, в игре или мобильном приложении.

1.1.5.1.3 Оформленная заявка

В основном используется в финансовых программах: приведенный пользователь должен заполнить на сайте заявку на получение кредита, займа, кредитной карты и т.п.

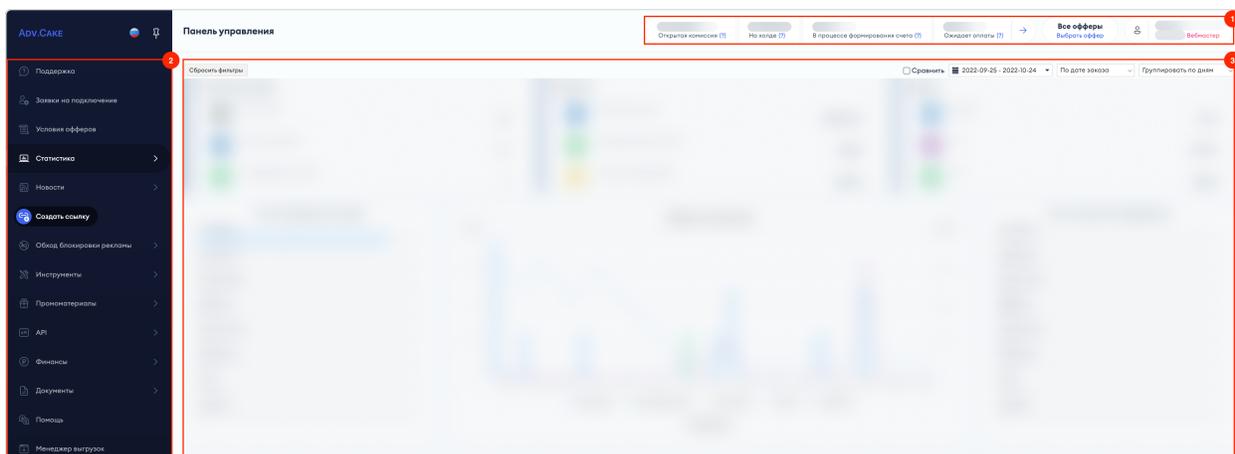
1.2 Обзор личного кабинета вебмастера

В этом разделе мы познакомим вас с интерфейсом личного кабинета, рассмотрим разделы главного меню и их основные функции.

Примечание: Разделы меню участников реферальной программы могут отличаться от представленных ниже, перейдите на страницу «Участникам реферальной программы».

1.2.1 Знакомство с интерфейсом

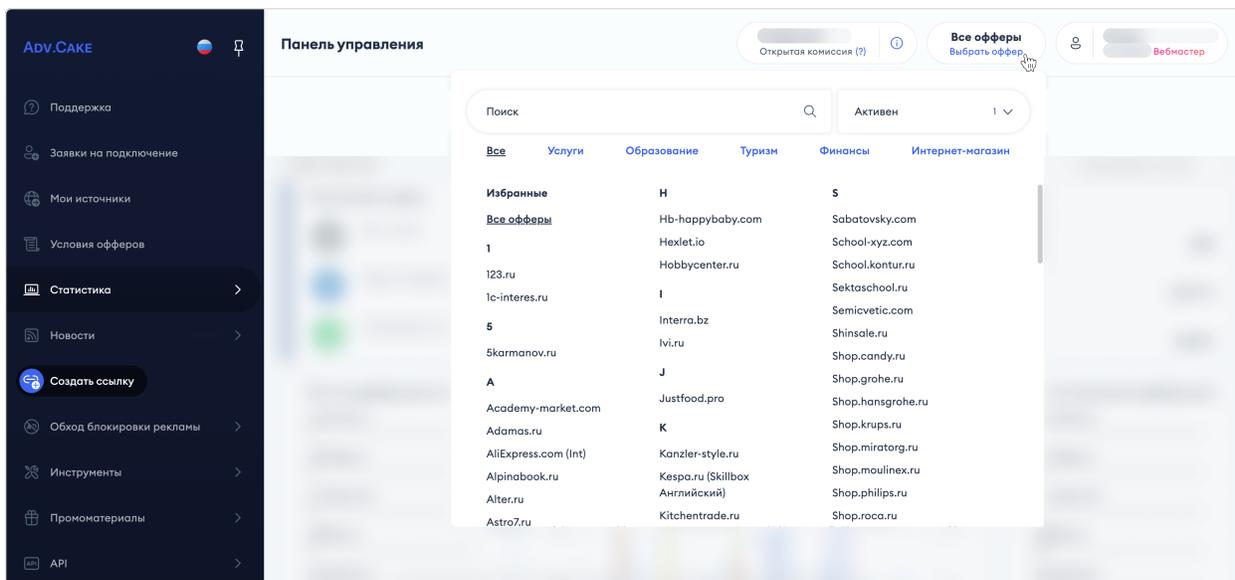
Ваш личный кабинет разделен на три части: шапка (1), главное меню (2) и рабочая область, где отображаются данные и возможные действия (3).



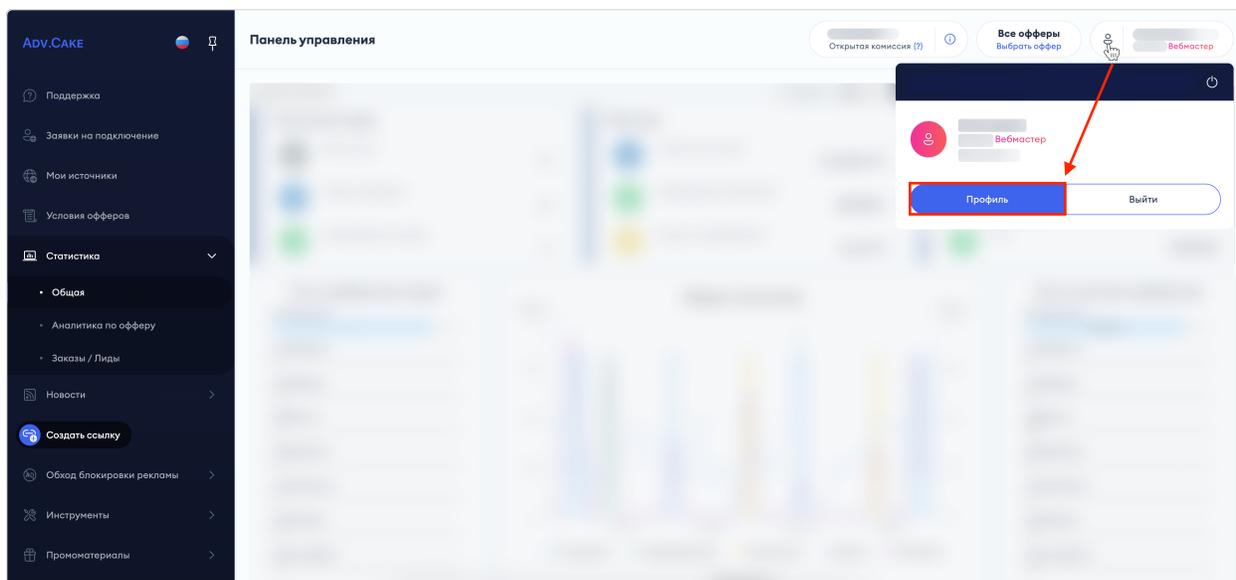
Шапка находится в верхней части личного кабинета и содержит в себе информацию о ваших *финансах*, доступных офферах и аккаунте.

Если вы работаете с несколькими офферами, нажмите на кнопку *Выбрать оффер* и введите в поле поиска название оффера, с которым вы планируете работать. Также, для удобства, все офферы разбиты по категориям и выводятся в алфавитном порядке:

Подсказка: Чтобы добавить оффер в избранное, наведите указатель на название оффера и нажмите кнопку «Звезда»



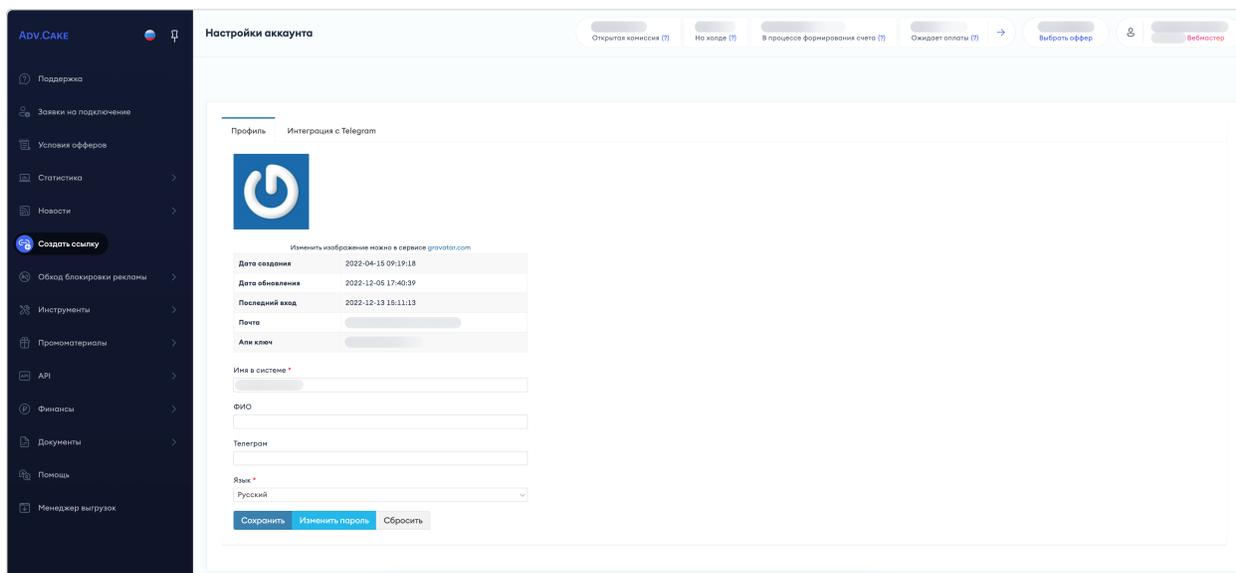
В настройки аккаунта можно попасть нажав на иконку вашего профиля и перейдя в раздел «Профиль»:



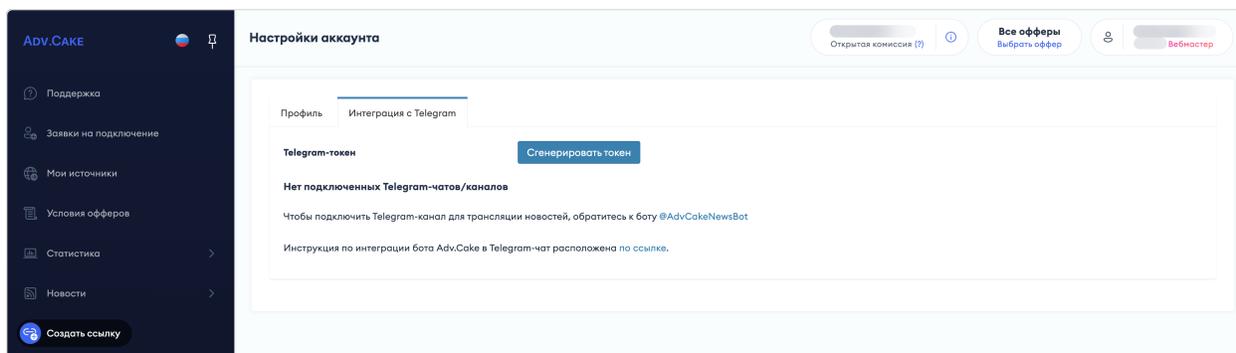
На табе **«Профиль»** представлена ваша электронная почта, дата регистрации в системе и API ключ. Здесь же, вы можете изменить ваш аватар, отображаемое имя в системе, Ф.И.О., логин в Telegram, а также язык личного кабинета.

После внесения изменений нажмите на кнопку *Сохранить*.

Чтобы изменить пароль, нажмите на кнопку *Изменить пароль* и следуйте инструкциям в появившемся окне.

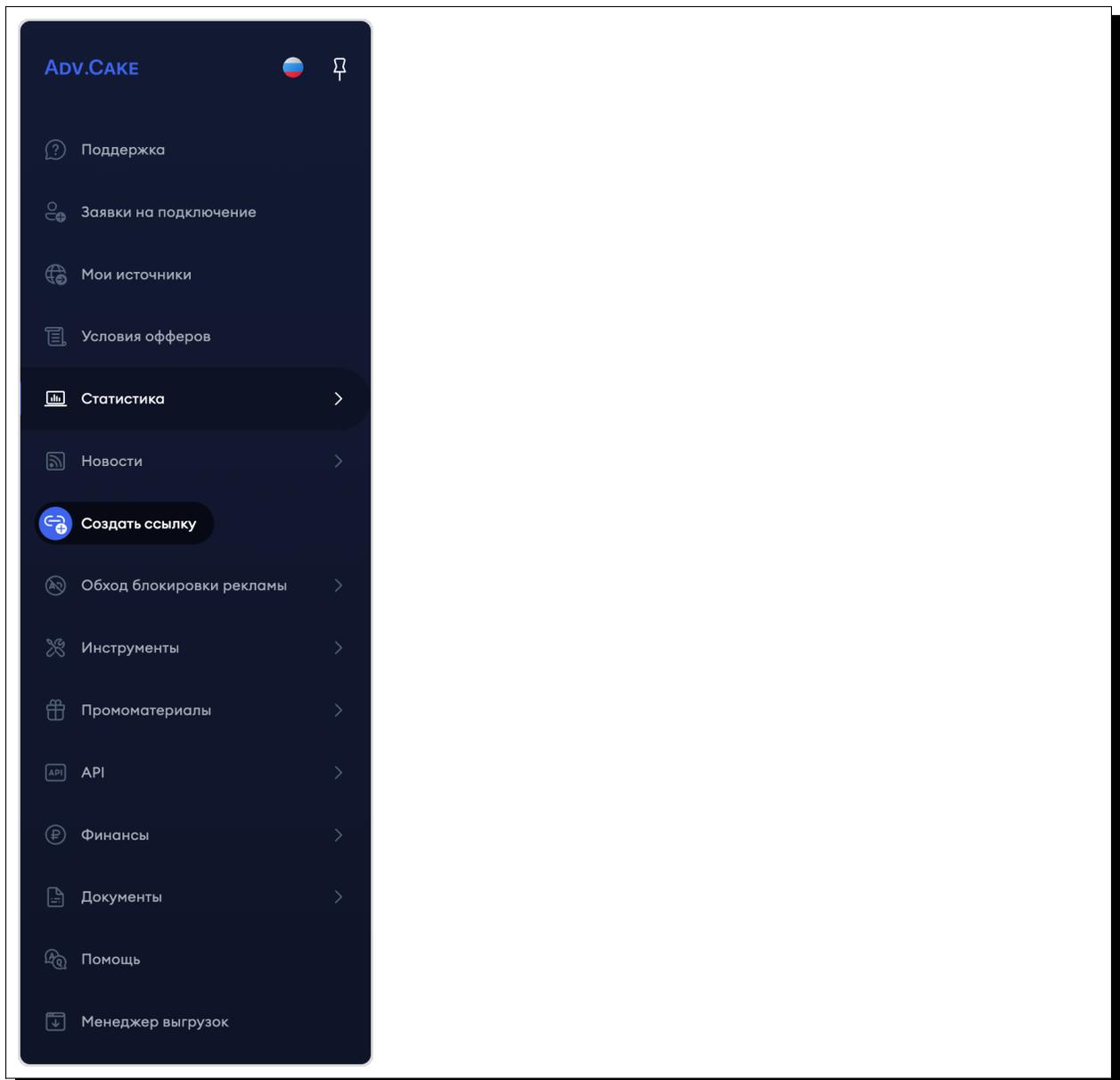


На табе **«Интеграция с Telegram»** находится инструмент по интеграции и работе с ботом Adv.Cake:



Если вы хотите автоматически получать актуальные новости об изменениях партнерской программы в ваш Telegram чат/канал, смотрите подробности в «Инструкции по интеграции бота Adv.Cake».

С помощью главного меню вы можете перейти в нужный раздел личного кабинета:



1. *Поддержка.*
2. *Заявки на подключение;*
3. *Мои источники;*
4. *Условия офферов;*
5. *Статистика;*
6. *Новости;*
7. *Создать ссылку;*
8. *Обход блокировки рекламы;*
9. *Инструменты;*
10. *Промоматериалы;*

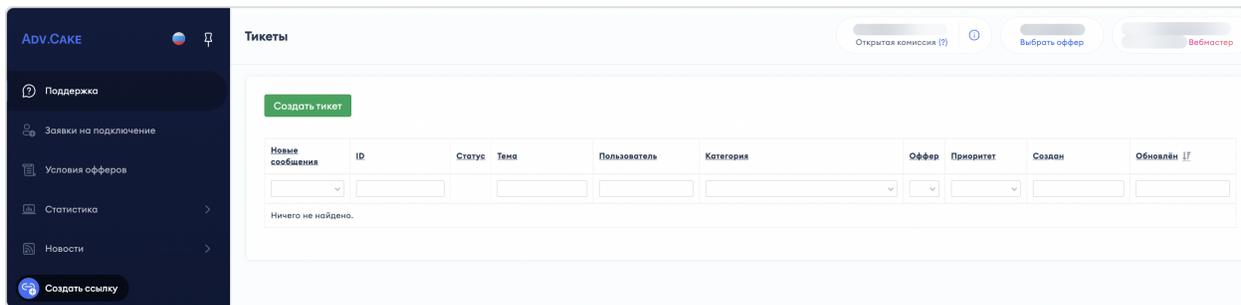
11. API;
12. Финансы;
13. Документы;
14. Помощь;
15. Менеджер выгрузок.

Подсказка: Чтобы закрепить меню в левой части экрана, нажмите на иконку с булавкой. Здесь же, можно выбрать, на каком языке будет отображаться ваш личный кабинет.

1.2.2 Поддержка

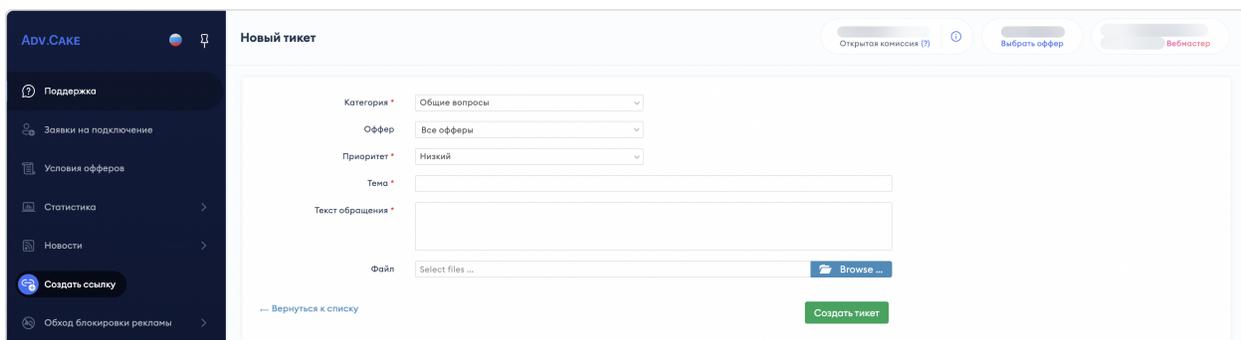
В этом разделе описывается взаимодействие вебмастера с техподдержкой Adv.Cake через создание тикетов.

Каждому тикету присваивается ID, статус, тема и категория обращения, оффер, а также приоритет рассмотрения.



Чтобы создать новый тикет, нажмите *Создать тикет* и заполните следующие поля:

1. **Категория** – общие/технические/финансовые/другие вопросы;
2. **Оффер** – оффер, в рамках которого требуется помощь;
3. **Приоритет** – низкий/нормальный/высокий;
4. **Тема** – тема вопроса, вкратце отражающая его суть;
5. **Текст обращения** – подробное описание проблемы;
6. **Файл** – скриншот, видео или иные файлы, которые иллюстрируют вашу проблему (файлы прикладывать желательно, но не обязательно).



Подсказка: Прежде чем создавать тикет, рекомендуем обратиться к нашей документации – возможно, вы найдете ответ на ваш вопрос.

После заполнения всех полей нажмите *Создать тикет*, после чего, в нижней части страницы откроется чат с менеджером поддержки Adv.Cake, где вы можете задавать дополнительные вопросы (поле «Новое сообщение») и прикреплять файлы (поле «Файл»).

Примечание: Рекомендуем своевременно отвечать на уточняющие вопросы и предоставлять дополнительную информацию, если ее запросили сотрудники поддержки – это поможет нам быстрее решить вашу проблему.

После того, как статус тикета сменится на «Закрыт», вы сможете оценить работу нашей поддержки: для этого нажмите на кнопку «Звезда» и выберите количество звезд, которое, по вашему мнению, лучше всего отражает ваш опыт общения с поддержкой.

Мы внимательно читаем каждый комментарий и всегда прислушиваемся к конструктивной критике:)

1.2.3 Заявки на подключение

Раздел содержит информацию о ваших заявках и их статусах.

Оффер	Создано	Сайт	Тип источника	Тип лица	Комментарий вебмастера	Реферер	Статус
							Одобен
							Регистрация не окончена. Добавьте источники →
							Отклонен
							На модерации

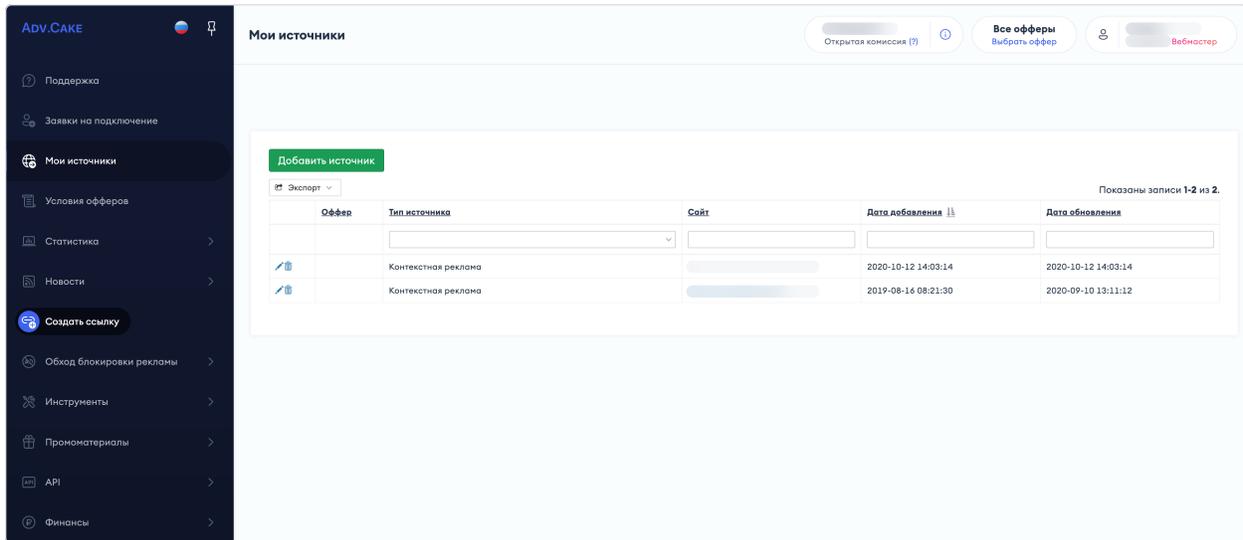
Возможные статусы заявок:

1. **Одобен** — можете приступать к работе с программой;
2. **Регистрация не окончена** — уточните дополнительные данные, нажав на кнопку «Стрелка»;
3. **Отклонен** — ознакомьтесь с правилами подачи заявок и скорректируйте вашу заявку;
4. **На модерации** — подождите, пока заявка поменяет свой статус;
5. **Заблокирован** — заявка заблокирована.

Предупреждение: Обратите внимание, если ваша заявка(-и) для работы с программой не была одобрена, вам будет доступен ограниченный список разделов личного кабинета: «Поддержка», «Заявки на подключение», «Помощь» – скорректируйте вашу заявку и дождитесь ее одобрения, после чего вы сможете приступить к полноценной работе с нашей платформой.

1.2.4 Мои источники

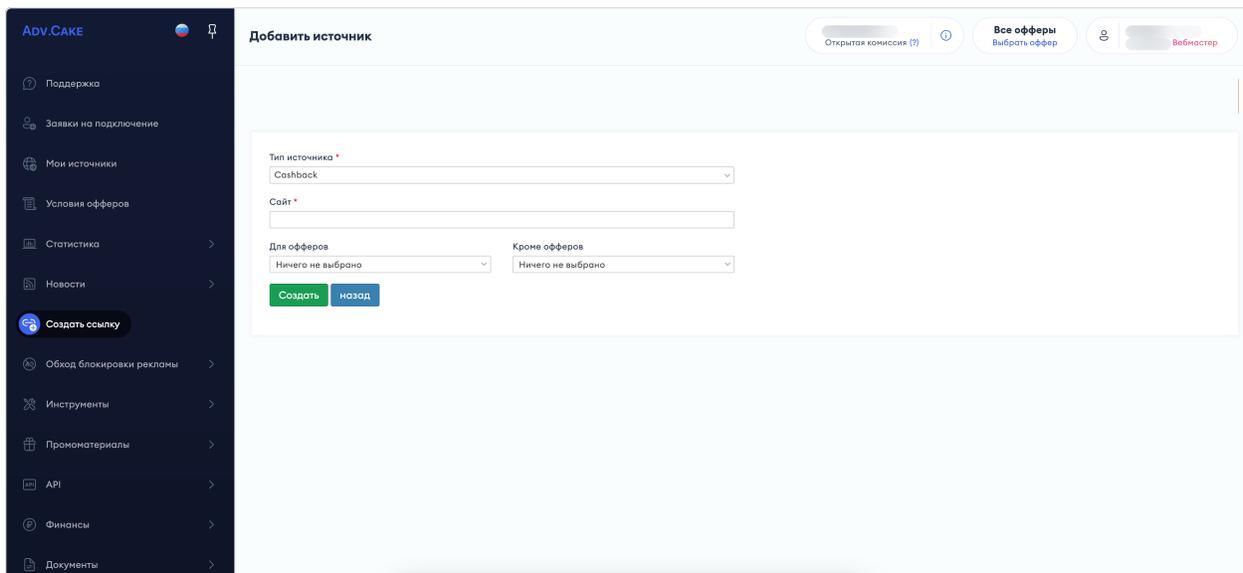
Раздел содержит информацию о ваших источниках (рекламных площадках).



Чтобы добавить источник, нажмите *Добавить источник*. В появившемся окне укажите:

1. Тип источника;
2. Адрес сайта;
3. Офферы, которым будет соответствовать или не соответствовать новый тип источника (необязательно).

После внесения необходимой информации, нажмите кнопку *Создать*:



1.2.5 Условия офферов

Раздел содержит действующие ставки, краткое описание условий партнерской программы, ее целевой аудитории, а также информацию о разрешенных и запрещенных типах трафика.

Примечание: Подробнее об источниках трафика читайте на странице «Источники трафика».

Здесь же доступны следующие показатели статистики:

1. **CR** (от англ. «Conversion Rate») – процентное отношение количества целевых действий пользователей к общему числу переходов (кликов) по партнерской ссылке;
2. **ЕСРС** (от англ. «Effective Cost Per Click») — это эффективная стоимость одного перехода (клика) в CPA-сетях (или сайтах с партнерскими программами), которая рассчитывается по следующей формуле: $\text{сумма заработка} \div \text{количество кликов}$;
3. **AR** (от англ. «Approval rate») — процент одобренных заказов от общего количества заказов;
4. **ЕРL** (от англ. «Earnings Per Lead») — величина для оценки среднего заработка с совершенного целевого действия, отношение подтвержденных финансов к количеству целевых действий.

The screenshot shows the 'Условия оффера eldorado.ru' page in the Adv.Cake interface. The page features a dark sidebar on the left with navigation options like 'Поддержка', 'Заявки на подключение', 'Условия оффера', 'Статистика', 'Новости', 'Создать ссылку', 'Обход блокировки рекламы', 'Инструменты', 'Промоматериалы', 'API', 'Финансы', 'Документы', 'Помощь', and 'Менеджер выгрузок'. The main content area has a top navigation bar with status indicators: 'Открыта комиссия', 'На кода', 'В процессе формирования счета', 'Ожидает оплаты', 'Выбрать оффер', and 'Вебмастер'. Below this, there's a header for 'Условия оффера eldorado.ru' with a 'Создать ссылку' button and a 'В статистику' button. The main content area includes a table titled 'Ставки' with columns for 'Ставка', 'Условия', and 'Период действия'. Below the table, there are sections for 'Сред жизни cookie', 'Хед для партнёра', 'Возможность атрибуции по промокодам', and 'Рассиловка категорий ставок (ссылка)'. The 'el dorado.ru' logo and name are prominently displayed at the top of the main content area.

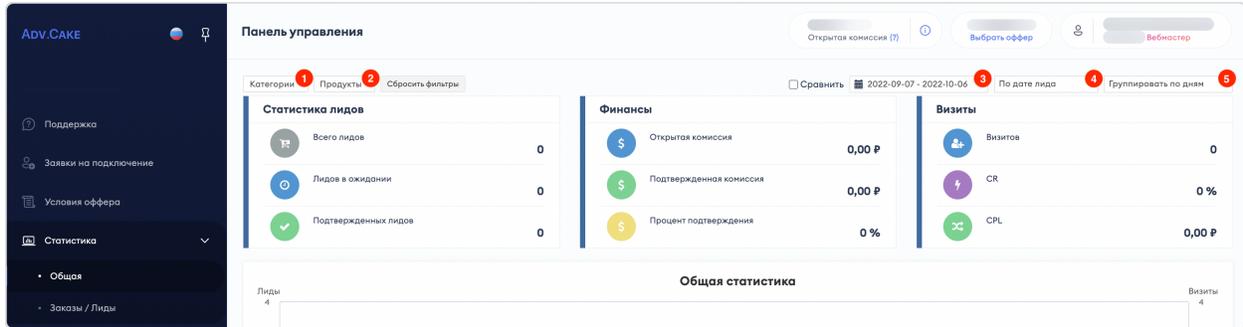
Подсказка: Ознакомиться с полным списком CPA-терминов вы также можете в нашем глоссарии.

1.2.6 Статистика

1.2.6.1 Общая

В этом разделе вы можете просматривать статистику по действиям по вашей партнерской ссылке: количество визитов, лидов, комиссию.

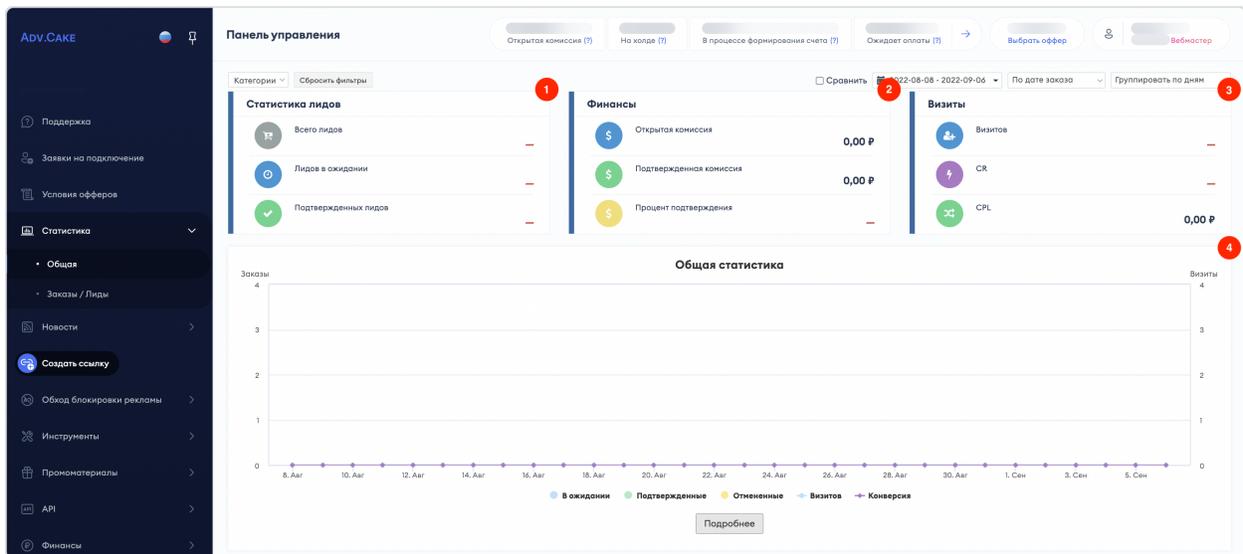
Для более точного отображения данных, воспользуйтесь фильтрами:



1. **Категории** – позволяет отображать статистику по категориям оффера;
2. **Продукты** – позволяет отображать статистику по доступным продуктам;
3. **Фильтр по дате** — позволяет отслеживать статистику по дате создания/оформления заказа;
4. **Фильтр по дате обновления** — позволяет отображать статистику по вхождению заказа по дате оформления/обновления заказа;
5. **Фильтр по группировке заказов** — позволяет группировать заказы по дням, неделям или месяцам.

Также, вы можете выбрать период, за который хотите сравнить статистику — для этого нажмите на кнопку *Сравнить* и задайте любые даты для сравнения в появившемся календаре.

Далее, рассмотрим несколько информационных блоков:



1.2.6.1.1 Статистика лидов (1)

1. **Всего лидов:** Общее количество заказов/лидов, которые привлек вебмастер за выбранный период;
2. **Лидов в ожидании:** Заказы/лиды, которые не получили конечного статуса (*Одобрено/Отменено*);
3. **Подтвержденных лидов:** Заказы/лиды со статусом *Одобрено*, за которые вебмастер получит комиссию по окончании отчетного периода.

1.2.6.1.2 Финансы (2)

1. **Открытая комиссия:** Потенциальная комиссия по всем лидам в ожидании. Другими словами, это то, сколько мог бы заработать вебмастер, если бы каждый *лид в ожидании* стал *подтвержденным лидом*;
2. **Подтвержденная комиссия:** Комиссия по подтвержденным лидам;
3. **Процент подтверждения:** То же, что и AR (Approval rate) — процент одобренных заказов от общего количества заказов.

1.2.6.1.3 Визиты (3)

1. **Визитов:** Число переходов по ссылкам вебмастера за выбранный период;
2. **CR:** Конверсия (Conversion rate) — отношение количества целевых действий к количеству визитов;
3. **CPL/EPL:** Цена(оплата) за лид/заявку (Cost per lead) — средний заработок вебмастер за один заказ.

1.2.6.1.4 Общая статистика (4)

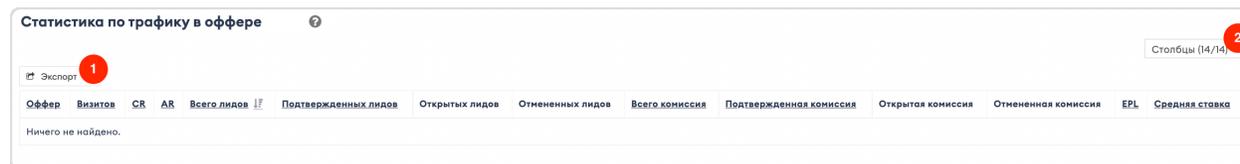
График, в котором столбцами отображены статусы заказов/лидов по дням, а линиями — количество визитов и конверсия. Для просмотра статистики по каждому дню нажмите на кнопку *Подробнее*

Также, в нижней части страницы, вы можете просматривать статистику по реферальному трафику и sub-параметрам:

1.2.6.1.5 Статистика по трафику

Для более точного отображения статистики по трафику, воспользуйтесь панелью настроек:

1. **Экспорт** — позволяет экспортировать данные в формате CSV/Excel;
2. **Столбцы** — позволяет выбрать столбцы для отображения данных.



Примечание: Статистика по офферу обновляется раз в час.

1.2.6.1.6 Статистика по сабам

Позволяет просматривать статистику в разрезе каждого sub-параметра.

Для более точного отображения статистики по sub-параметрам, воспользуйтесь панелью настроек:

1. **Группировать** — позволяет группировать статистику по sub-параметрам в произвольном порядке. Нажмите на кнопку «Перетянуть» рядом с нужным параметром и расположите их в удобном для вас порядке;
2. **Экспорт** — позволяет экспортировать данные в формате CSV/Excel;
3. **Столбцы** — позволяет выбрать столбцы для отображения данных.

Статистика по сабам

Группировать

Оффер Sub1 Sub2 Sub3 Sub4 Ключевое слово (Keyword) Sub5

Экспорт

Столбцы (12/12)

Показаны записи 1-5 из 6.

Оффер	Sub1	Sub2	Sub3	Sub4	Ключевое слово (Keyword)	Sub5	Кликов	CR	Всего лидов	IF	AR	Подтвержденных лидов	Подтвержденная комиссия	ЕPL	Всего комиссии	Открытых лидов	Открытая комиссия	Отмененных лидов	Отмененная комиссия
							-	0%	2	50%	1	0,00 P	0,00 P		0,00 P	1	0,00 P	0	0,00 P

Примечание: Подробнее об использовании sub-параметров читайте на странице .

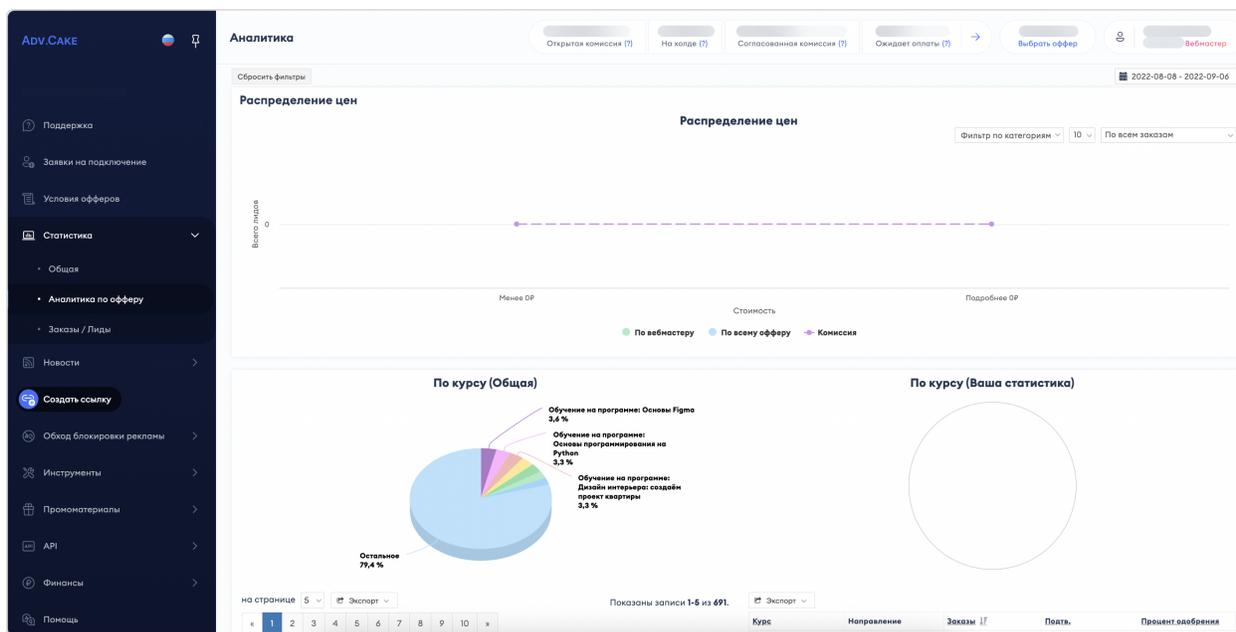
1.2.6.2 Аналитика по офферу

Раздел содержит подробную аналитику по вашим офферам.

Предупреждение: Данный раздел доступен не на всех офферах, если вы хотите видеть детальную аналитику по офферу – сообщите вашему менеджеру.

Для перехода в раздел аналитики по офферу, нажмите на кнопку *Перейти* напротив интересующего вас оффера, после чего откроется страница аналитики.

Пример аналитики образовательного оффера, в которой можно увидеть распределение цен, общую и индивидуальную аналитику вебмастера по курсу, направлению и категории:

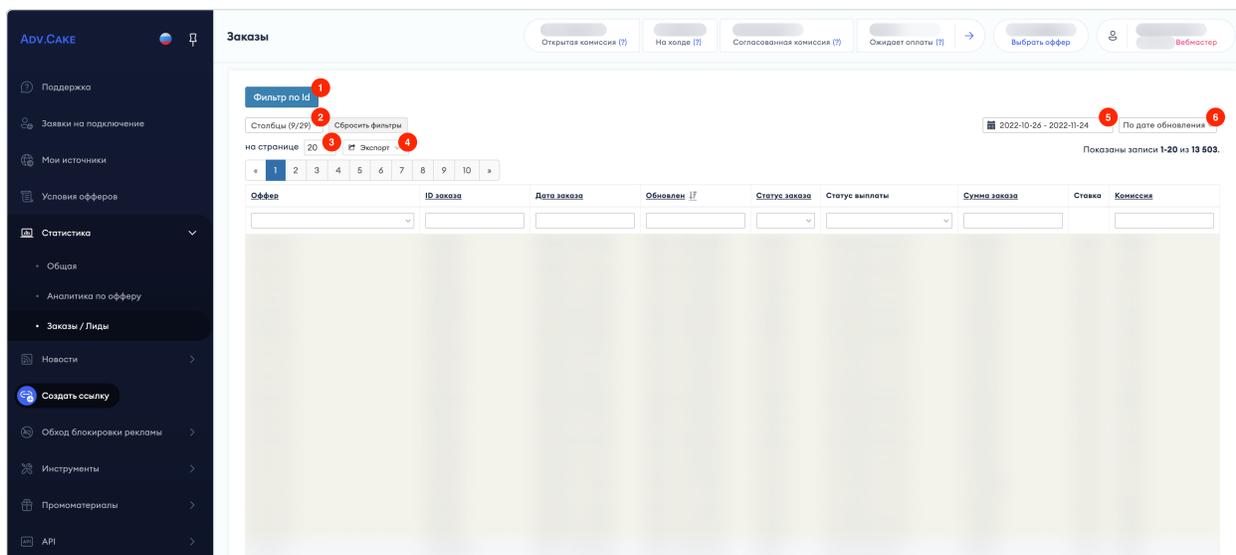


1.2.6.3 Заказы/Лиды

Раздел содержит подробную информацию о заказах/лидах, совершенных по вашей партнерской ссылке.

Для более точного отображения данных по заказам/лидам, воспользуйтесь панелью настроек:

1. **Фильтр по Id** — позволяет фильтровать заказы/лиды по Id;
2. **Столбцы** — позволяет выбрать столбцы для отображения данных;
3. **На странице** — позволяет выбрать количество отображаемых данных на странице;
4. **Экспорт** — позволяет экспортировать данные в формате CSV/Excel.
5. **Фильтр по дате** — позволяет задать временной промежуток для отображения данных;
6. **Фильтр по дате заказа/обновления** — позволяет фильтровать по дате заказа/обновления.



Примечание: В столбце «Причина» вам могут встретиться причины отмены заказа «Заказ аннулирован» и «Принадлежит другому источнику по GA».

Если заказ аннулирован – это статус из CRM клиента, т.е. заказ не был подтвержден клиентом и был аннулирован;

Если заказ принадлежит другому источнику по GA (Google Analytics) это значит, что по данным клиента, перехода по вашей партнерской ссылке не было. Мы стараемся оспаривать такие заказы, если у нас есть подтверждение того, что заказ был совершен по ссылке партнера.

1.2.6.4 Добавление потерянных заказов

Раздел содержит инструмент для добавления потерянных заказов.

Предупреждение: Обратите внимание, инструмент добавления потерянных заказов доступен не на всех офферах.

1.2.6.4.1 Одиночная загрузка

Для одиночной загрузки потерянных заказов заполните следующие обязательные поля:

1. **ID заказа** – номер заказа;
2. **Дата создания заказа** – в формате ГГГГ-ММ-ДД-ВРЕМЯ. При отсутствии времени заказа укажите 00:00:00;
3. **Оффер** – название сайта в формате site.ru;
4. **Сумма заказа**;
5. **Категория заявки** – выберите вариант: «Не отобразился в статистике»/«Не согласен со статусом»/«Не согласен с суммой заказа»/«Не согласен с комиссией»/«Другое»;
6. **Комментарий** – обязательно только для категории заявки «Другое»;

- Подтверждение** – файл или ссылка на скриншот с подтверждением. Если в выпадающем списке выбран «Файл», соседнее поле будет называться «Файл», которое предоставит возможность выбрать и загрузить графический файл. Если выбрана «Ссылка», то соседнее поле будет называться «URL», в которое можно скопировать адрес ссылки.

После заполнения всех полей нажмите *Создать заявку*.

The screenshot shows the 'Добавление потерянных заказов' (Adding lost orders) form in the Adv.Cake webmaster interface. The form is titled 'Добавление потерянных заказов' and has three tabs: 'Одиночная загрузка' (Single upload), 'Массовая загрузка' (Mass upload), and 'Заявки по потерянным заказам' (Orders by lost orders). The form contains the following fields:

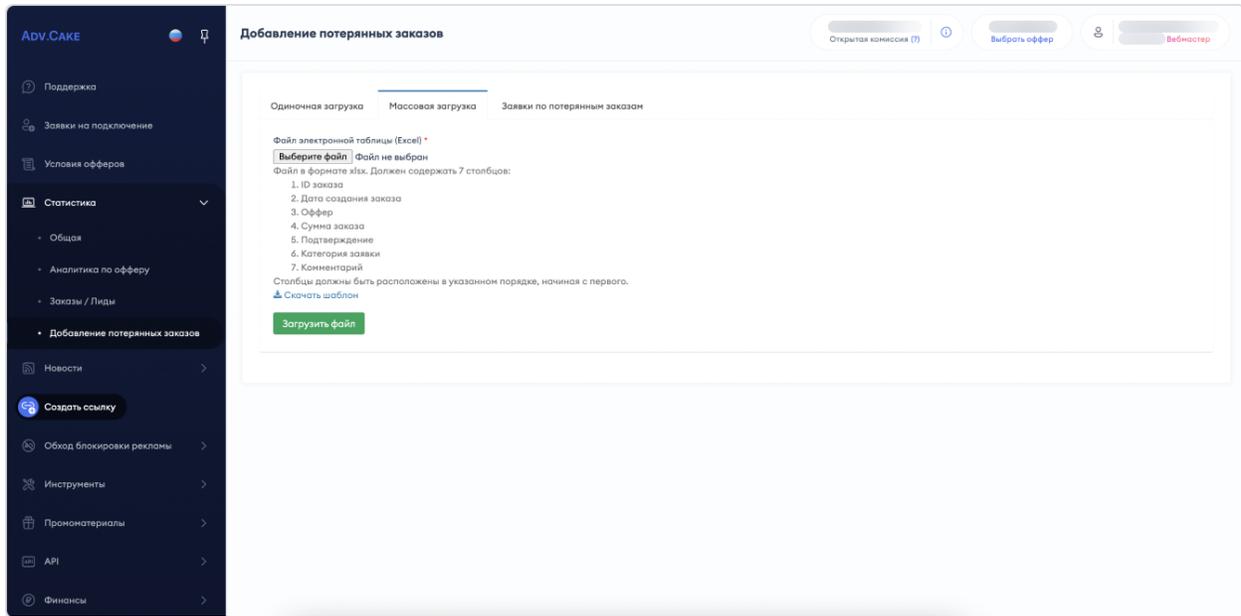
- ID заказа ***: A text input field.
- Дата создания заказа ***: A date and time picker showing '2022-08-23 17:28:00'.
- Оффер ***: A dropdown menu.
- Сумма заказа ***: A text input field with '0' entered.
- Категория заявки ***: A dropdown menu with the option 'Не отобразился в статистике' selected.
- Комментарий вебмастера**: A text area with the instruction 'Необходимо заполнить, если выбрана категория Другое'.
- Подтверждение ***: A dropdown menu with 'Ссылка' selected.
- Ссылка**: A text input field.

A green button labeled 'Создать заявку' (Create order) is located at the bottom of the form.

1.2.6.4.2 Массовая загрузка

Для массовой загрузки потерянных заказов:

- Скачайте шаблон (кнопка *Скачать шаблон*);
- Заполните обязательные поля таблицы:
 - ID заказа** – номер заказа;
 - Дата создания заказа** – в формате ГГГГ-ММ-ДД-ВРЕМЯ. При отсутствии времени заказа укажите 00:00:00;
 - Оффер** – название сайта в формате site.ru;
 - Сумма заказа**;
 - Подтверждение** – ссылка на скриншот с подтверждением;
 - Категория заявки** – выберите вариант: «Не отобразился в статистике»/«Не согласен со статусом»/«Не согласен с суммой заказа»/«Не согласен с комиссией»/«Другое»;
 - Комментарий** – обязательно только для категории заявки «Другое».
- Загрузите заполненный шаблон при помощи кнопки *Выберите файл*;
- Нажмите *Загрузить файл*.



1.2.6.4.3 Заявки по потерянным заказам

Раздел содержит все созданные вебмастером заявки на добавление потерянных заказов, а также статусы по ним («В обработке»/«Заказ добавлен»/«Заказ отменен»).

1.2.7 Новости

1.2.7.1 Все новости

Раздел содержит новости от партнерских программ об изменении условий программ, новых акциях, добавлении рекламных материалов и другую полезную информацию.

Новости выделяются тремя цветами:

1. Зеленый — новость не просмотрена;
2. Желтый — новость просмотрена;
3. Красный — важная новость.

1.2.7.2 Настройки уведомлений

Раздел содержит настройки уведомлений.

1.2.7.3 Новостной Telegram-канал

Этот раздел перенаправит вас на наш , где вы можете узнать обо всех обновлениях на платформе и поделиться своим мнением.

1.2.8 Создать ссылку

В данном разделе вы можете сгенерировать свою партнерскую ссылку на основной или дополнительные лендинги (если они доступны).

1.2.9 Обход блокировки рекламы

Раздел содержит информацию о трех эффективных решениях для обхода блокировки рекламы, а также развернутые инструкции по их установке.

1. **Ротация ссылок** — решение, доступное по умолчанию для всех вебмастеров Adv.Cake.
2. **Парковка домена** — подходит для вебмастеров, которые имеют возможность приобрести домен для оптимизации своих рекламных кампаний. Если у вас возникли вопросы по парковке доменов, попробуйте найти ответ на ваш вопрос .
3. **CakeLink API** — подходит для вебмастеров, которые могут использовать API для генерации партнёрских ссылок.

1.2.10 Инструменты

1.2.10.1 Тулбар

Браузерное расширение, позволяющее создавать партнерские ссылки и просматривать статистику без авторизации в личном кабинете.

Инструкцию по установке тулбара вы найдете в соответствующем разделе личного кабинета.

1.2.10.2 Товарные фиды

Товарный фид — это файл, который содержит данные о товарах из каталога сайта рекламодателя.

<p>Предупреждение: Товарные фиды доступны не на всех офферах — изучите условия вашего оффера, чтобы узнать возможность подключения этого инструмента.</p>
--

Подсказка: Если вам не нужно исключать какие-либо категории товаров или вы хотите получать большие товарные фиды без задержек — воспользуйтесь заранее сформированными общими фидами через API Общих фидов.

1.2.10.2.1 Как это работает?

В рамках оффера, рекламодатель предоставляет товарные фиды, которые вебмастер может использовать для создания сервисов сравнения товаров, маркетплейсов, сайтов-витрин и т.п.

Страница товарного фида содержит следующую информацию:

1. Возможные с продуктом действия (отмечены пунктиром) — «Перейти на сайт», «Создать виджет» (если доступно), «Скопировать реферальную ссылку», а также ID продукта, название, категорию, производителя, стоимость и доступность (т.е. наличие);
2. Кнопка *Получить ссылку для скачивания*, нажав которую вы получите ссылку на скачивание товарного фида в формате

Примечание: Обратите внимание, при отправлении запроса на формирование товарного фида есть две особенности:

1. Если запрошенный вами товарный фид содержит много элементов, вы увидите уведомление о том, что он еще не сгенерирован. После того, как уведомление исчезнет – товарный фид будет сгенерирован и его можно скачать по ссылке в поле «Ссылка для скачивания»;
2. Если вы не обращались к товарным фидам долгое время, то прекешированный (подготовленный заранее) товарный фид сбрасывается и создание нового может занять некоторое время.

Фид Общий товарный фид (ID 397)

Показаны записи 1-20 из 33 153.

Действия	ID продукта	Бренд	Название	Категория	Стоимость	Доступность
	10007575	DigiCare	Аккумулятор для цифрового фотоаппарата DigiCare PLN-EL14a	Аккумуляторы для фотоаппаратов и видеокамер	1 330 P 1 899 P -30 % (569 P)	Да
	10028419	Новех	Телевизор Новех NWT-24H121M5 Black	ЖК LED-телевизоры	6 999 P	Да
	4211200	Haier	Холодильник Haier CEFS35AWG	Холодильники с нижней морозильной камерой	43 999 P 60 999 P -28 % (17 000 P)	Да
	20042681	LG	Сплит-система (инвертор) LG Mega Plus P12EP1	Инверторные сплит-системы	45 999 P 60 999 P -25 % (15 000 P)	Да
	20083352	Haier	Холодильник (Side-by-Side) Haier HRF-541D7RU	Холодильники Side-by-Side и многодверные	91 999 P 129 999 P -29 % (38 000 P)	Да
	10026207	Триколор	Комплект спутникового ТВ Триколор Сибирь на LTB GS B622	Комплекты спутникового ТВ	9 899 P	Да

Выбрано товаров к экспорту: 33153

Получить ссылку для скачивания

Сохранить фид

Видите название фида? Если оставить пустым, то ссылка будет удалена через месяц с момента последнего скачивания. Если фид с такими параметрами уже есть, то он будет переименован. Названия должны быть уникальными.

Параметры ссылки

Описание фида

Формат экспорта

Поиск

Найти

Только в наличии

С уникальным url

Цена

Категории

Бренд

Если продукт из товарного фида поддерживает функцию создания товарного виджета, в строке с ним вы увидите эту иконку:

		10007575	DigiCare	Аккумулятор для цифрового фотоаппарата DigiCare PLN-EL14a	Аккумуляторы для фотоаппаратов и видеокамер	1 330 P 1 899 P -30 % (569 P)	Да
--	--	----------	----------	---	---	-------------------------------------	----

Нажав на нее у вас откроется окно создания товарного виджета, где можно:

1. Изменить цветовую схему товарного виджета;

2. Включить/отключить блок рекомендованных товаров;
3. Посмотреть ссылку на товар;
4. Задать дополнительные параметры для ссылки: поток, пять sub-параметров Sub1–Sub5, Keyword и название потока;
5. Получить код для размещения на странице вашего сайта при помощи кнопки *Сохранить и сгенерировать код*;
6. Посмотреть как будет выглядеть сгенерированный виджет на разных устройствах.

Подсказка: Некоторые товарные виджеты поддерживают редактирование стиля, размера и контента.

3. Панель настроек товарных фидов, расположенная в правой части страницы:

Сохранить фид

Введите название фида

Если оставить пустым, то ссылка будет удалена через месяц с момента последнего скачивания. Если фид с такими параметрами уже есть, то он будет переименован. Названия должны быть уникальны.

Параметры ссылок

Лендинг *

Все акции магазина –

Основной сайт [Поддерживает deeplink] –

Уценка –

Дополнительные параметры

Sub1	Sub2	Sub3
<input type="text"/>	<input type="text"/>	<input type="text"/>
Sub4	Sub5	Keyword
<input type="text"/>	<input type="text"/>	<input type="text"/>

[Описание фида](#)

[Формат экспорта](#)

Поиск

Найти

Только в наличии

С уникальным url

[Цена](#)

[Категории](#)

[Бренд](#)

1.2.10.2.2 Сохранить фид

Позволяет сохранить фид с заданными параметрами. Обратите внимание, если фид с такими параметрами уже есть, то он будет переименован.

Названия должны быть уникальными.

1.2.10.2.3 Параметры ссылок

Лендинг

Позволяет выбрать основной сайт или специальные/акционные страницы.

Дополнительные параметры

Позволяет задавать дополнительные параметры для ссылок: пять sub-параметров Sub1-Sub5 и один Keyword.

1.2.10.2.4 Описание фида

Здесь можно добавить описание фида: его название, название компании, сайт и контактный email.

1.2.10.2.5 Формат экспорта

Здесь можно выбрать формат экспорта фидов: YML, GSF (XML) или CSV, а также настроить сжатие файла в ZIP-архив перед отправкой.

1.2.10.2.6 Поиск

В этом блоке реализован поиск продуктов товарного фида по названию, цене, категории и бренду.

Также доступен поиск фидов по наличию (галочка «Только в наличии») и уникальным URL (галочка «С уникальным URL»).

1.2.10.3 Товарные виджеты

Раздел содержит инструменты для создания и управления товарными виджетами.

Обратите внимание, в этот раздел загружаются только те продукты из товарного фида, которые поддерживают функцию создания товарного виджета.

<p>Предупреждение: Функция товарных виджетов доступна не на всех офферах — изучите условия вашего оффера, чтобы узнать возможность подключения этого инструмента.</p>
--

1.2.10.4 Специальные виджеты

Раздел содержит инструменты для настройки специальные виджеты, которые предоставляет рекламодатель.

Предупреждение: Функция специальных виджетов доступна не на всех офферах — изучите условия вашего оффера, чтобы узнать возможность подключения этого инструмента.

1.2.11 Промоматериалы

1.2.11.1 Лендинги

Лендинг — веб-страница, на которую пользователь переходит по ссылке, баннеру, контекстному объявлению или другому виду рекламного объявления.

Основная задача лендинга в том, чтобы побудить пользователя совершить целевое действие — купить товар, оформить подписку, зарегистрироваться в приложении и т.д.

После того, как рекламодатель предоставит ссылки лендингов своему аккаунт-менеджеру, они добавятся в систему и вы сможете увидеть их в этом разделе.

Задача вебмастера — направлять трафик разрешенного типа на предоставленные рекламодателем ссылки, чтобы на них отображались необходимые акции.

Вы можете просматривать статистику по лендингам, которая представлена в виде таблицы (4).

Для экспорта таблицы в формате XLS нажмите *Экспорт* (2), для настройки отображения количества записей на странице нажмите кнопку (3).

Переключатель «Только активные» (1) позволяет отобразить только активные лендинги (положение «вкл») и все лендинги, в том числе неактивные (положение «выкл»).

Также, для удобства работы с лендингами, реализовано быстрое копирование ссылки лендинга по кнопке в столбце «Реферальная ссылка» (5):

Дата объявления	Оффер	Срок действия	Привязка к акции	Название	Desktop link	Сайт	CR	eCPA	Реферальная ссылка
2018-09-07 13:01:24		-	-	Основной сайт	(из url)				
2018-10-02 14:40:35									
2019-10-23 19:21:11									
2019-10-23 19:22:29									
2019-10-23 19:25:10									
2019-10-23 19:26:04									
2019-10-23 19:28:00									
2019-10-23 19:29:04									
2019-10-23 19:29:40									
2019-10-24 10:35:22									

1.2.11.2 Акции

Раздел содержит информацию об акциях.

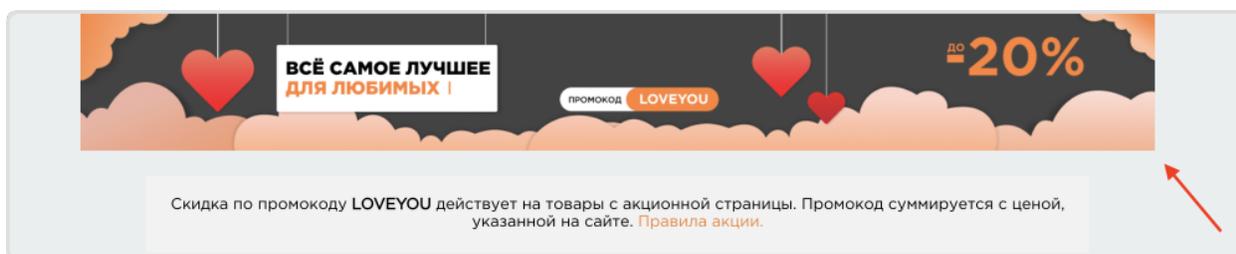
В системе Adv.Cake акции — родительский элемент, который может содержать в себе лендинги, промокоды, баннеры и видео.

У акций есть название, сроки действия, описание и материалы.

Так акция выглядит в личном кабинете:

Оффер	Обновлена	Название	Срок действия	Описание	Материалы
<input type="text"/>		<input type="text"/>		<input type="text"/>	
	2022-02-02 18:36:45	Скидки ко Дню всех влюбленных	с 2022-02-02 до 2022-02-14	Скидки на товары из списка.Скидка для каждого товара с акционной страницы рассчитывается отдельно.	Промокод Лендинг

А так на сайте:



1.2.11.3 Промокоды

Промокод — это особая комбинация букв и цифр, «тайный код» интернет-магазина, воспользовавшись которым при оформлении, пользователь может получить скидку на определённый товар или на всю сумму заказа.

У промокода есть код (например LOVE, GIFT30, ADVCAKE), срок действия, тип скидки (например бесплатная доставка, скидка на заказ, подарок к заказу), привязка к акции (как к родительскому элементу), минимальная сумма активации промокода, размер скидки, единица скидки (Проценты/Рубли), условия использования промокода и краткое описание.

1.2.11.4 Баннеры

Баннеры — изображения рекламного характера, которые содержат в себе ссылку на сайт рекламодателя.

Они предоставляются рекламодателями в рамках оффера, а в этом разделе вебмастер может скачать баннер для его дальнейшего размещения на своей рекламной площадке.

Чтобы посмотреть детали и дополнительные настройки баннера, нажмите на него в поле «Название»:

Обновлен	Оффер	Название	Категория	Лендинг	Акция	Срок действия	Тип	Размеры	Теги
	2022-10-05 22:44:57	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>		<input type="text"/>	<input type="text"/>	<input type="text"/>
		240x400	Без категории		-	-	image/jpeg	240 x 400 px	-

На вашем экране отобразится окно «Просмотр баннера», которое позволяет:

1. Посмотреть детали баннера: Id, категорию, связанный лендинг, MIME Type (тип изображения - jpeg/ png/ gif);
2. Задать адрес целевой страницы и дополнительные sub-параметры;
3. Скопировать код баннера для размещения на сайте. Чтобы скопировать баннер нажмите на кнопку *Скопировать в буфер обмена*, а чтобы он открывался в новом окне, поставьте галочку в поле «Открывать в новой вкладке».

Просмотр баннера ✕

1

Id	<input type="text"/>
Категория	-
Лендинг	<input type="text"/>
Mime Type	image/jpeg

2

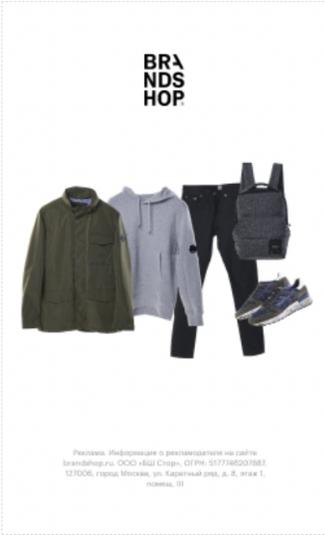
Адрес целевой страницы

Sub1	Sub2	Sub3
<input type="text"/>	<input type="text"/>	<input type="text"/>
Sub4	Sub5	Keyword
<input type="text"/>	<input type="text"/>	<input type="text"/>

Открывать в новой вкладке **3**

Результат

```
<!-- Advcake Banner -->
<div id="adb-c189d63922e639c1-72932"></div>
<script>
window.advcake_runner = function(){
window.advcake_promo.forEach(function(t){t();});
window.advcake_promo = [];
window.advcake_promo.push = function(t){t();};
```



240x400px

1.2.11.5 Видео

Здесь вы можете скачать готовые видео для работы с оффером.

У видео есть оффер, название, категория, лендинг, срок действия и тип.

1.2.12 API

Узнать общую информацию по работе с API, а также взглянуть на примеры вы можете на странице .

1.2.13 Финансы

Раздел содержит информацию о начислении вознаграждения. Информация о ваших финансах также отображается в отдельном виджете в шапке личного кабинета:

Примечание: Обратите внимание, статусы выплат для юр.лиц и физ.лиц/самозанятых различаются. Подробности ниже.

1.2.13.1 Статусы выплат для юр.лиц (без подписанных односторонних актов)

Примечание: Для получения выплат перейдите в раздел «Документы» -> «Реквизиты», заполните ваши реквизиты и скачайте комплект закрывающих документов, после чего подписанные сканы отправьте на support@advcake.com, а два подписанных экземпляра по адресу: *123001, г. Москва, ул. Большая Садовая 5, к.2, офис 2038, 4 этаж.*

1. **Открытая комиссия** — комиссия всех заказов в статусе «Новый». При получении статуса «Одобрено» переходит в раздел «На холде», при получении статуса «Отменено» — списывается. Открытая комиссия может изменяться не только за счет изменения статуса, но и за счет изменения самой комиссии. Т.е. пользователь приобрел товаров на 10 000 руб, после чего решил вернуть часть товаров — общая сумма заказа, как и общая комиссия, изменится, и если она считается в % от стоимости, тогда она тоже уменьшится;
2. **На холде** — комиссия заказов в статусе «Одобрено», будет доступна к выводу после согласования рекламодателем;
3. **Доступно к выставлению счета** — согласованная рекламодателем комиссия по всем заказам в статусе «Одобрено». Для формирования выплаты перейдите в раздел «Оплата»;
4. **Ожидает оплаты** — сумма сформированных счетов. Оплата производится в течение 15 рабочих дней после оплаты счета рекламодателем. Время оплаты счета различными рекламодателями может отличаться. Статус оплаты счета рекламодателем уточняйте в разделе «Финансы».

Примечание: У юр.лиц с подписанными односторонними актами третий пункт назван «В процессе формирования счета», а все счета хранятся в разделе «Финансы».

Подписание односторонних актов позволяет получать выплаты без лишних действий, т.е. формирование сумм происходит автоматически.

1.2.13.2 Статусы выплат для физ.лиц/самозанятых

Примечание: Для получения выплат перейдите в раздел «Финансы» → «Данные самозанятого» и внесите ваши данные.

1. **Открытая комиссия** — комиссия всех заказов в статусе «Новый». При получении статуса «Одобрено» переходит в раздел «На холде», при получении статуса «Отменено» — списывается. Открытая комиссия может изменяться не только за счет изменения статуса, но и за счет изменения самой комиссии. Т.е. пользователь приобрел товаров на 10 000 руб, после чего решил вернуть часть товаров — общая сумма заказа, как и общая комиссия, изменится, и если она считается в % от стоимости, тогда она тоже уменьшится;
2. **На холде** — комиссия заказов в статусе «Одобрено», будет доступна к выводу после согласования и оплаты рекламодателем;
3. **Согласованная комиссия** — комиссия, доступная к выводу. Выплата создается автоматически в течение 10 дней после оплаты счета рекламодателем;
4. **Ожидает оплаты** — передано на оплату в сервис выплат. Среднее время проведения выплаты — 10 дней.

1.2.13.2.1 Краткое описание статусов заказов:

1. **Новый** — заказ, который попадает в личный кабинет после того, как пользователь совершил какое-либо действие на сайте (оформил заявку, приобрел товар и т.д.);
2. **Одобен** — заказ, который рекламодатель одобрил (прошла оплата, оформлена доставка и т.д.) и готов оплатить;
3. **Отменен** — заказ, от которого пользователь отказался по разным причинам (не понравился, нашел дешевле и т.д.).

1.2.13.2.2 Способы выплат для вебмастеров

Вебмастерам доступны следующие способы выплат:

1. Для вебмастеров с гражданством РФ, оформивших самозанятость:
 - Банковская карта Visa/Mastercard/Мир (Руб.)
-

2. Для вебмастеров с гражданством других стран:
 - Банковская карта Visa/Mastercard (€)
 - Банковская карта Visa/Mastercard (\$)
 - Банковская карта Visa/Mastercard (другая валюта)
 - Банковский перевод (иностранное физ. лицо) (\$)
 - Банковский перевод (иностранное физ. лицо) (€)
 - PayPal (\$)
 - PayPal (€)
 - Криптокошелек USDT TRC-20
 - Криптокошелек USDC TRC-20
-

3. Для вебмастеров, работающих как юридическое лицо с регистрацией вне РФ и СНГ:
 - Банковский перевод (иностранное юр. лицо) (\$)
 - Банковский перевод (иностранное юр. лицо) (€)
-

4. Для вебмастеров, работающих как юридическое лицо с регистрацией РФ и СНГ:
 - Банковский перевод на расчетный счет

1.2.13.3 Финансовая статистика по офферам

В этом разделе содержится статистика по движению средства по всем офферам за все время или за выбранный период.

1.2.14 Документы

1.2.14.1 Для юр.лиц

1.2.14.1.1 Договоры

На этой вкладке содержится информация о ваших договорах.

Для начала работы с личным кабинетом вам понадобится скачать договор (кнопка *Скачать договор*) и отправить два подписанных экземпляра на указанный адрес удобным для вас способом.

Вы также можете отправить версию договора с помощью ЭДО (электронного документооборота) на указанную организацию.

Примечание: Если вы хотите изменить одно юр.лицо на другое, вам понадобится скачать новый договор и отправить два подписанных экземпляра на указанный адрес.

Обратите внимание, изменение юр.лица на физ.лицо возможно только в том случае, если у вас нет открытой комиссии.

Если вам понадобится помощь, обратитесь в службу поддержки.

Договоры

Открытая комиссия (?)

Все офферы
Выбрать оффер

Вебмастер

Скачать договор

Два подписанных экземпляра отправить удобным для вас способом по адресу: 123001, г. Москва, ул. Большая Садовая 5, к.2, офис 2038, 4 этаж., тел. +7 (495) 926-26-81. Также вы можете отправить версию договора с помощью ЭДО (электронного документооборота) на организацию ООО "Ада-Кей" ИНН 9705066208.

Экспорт

Показаны записи 1-1 из 1.

Номер	Оффер	Источник	Подписанный экземпляр	URL в ЭДО	Дата подписания	Тип Юр. лица	Форма налогообложения
					2020-02-01	ИП	УСН (без НДС)

1.2.14.1.2 Доп.соглашения

На этой вкладке содержатся дополнительные соглашения к основным договорам.

1.2.14.1.3 Реквизиты

На этой вкладке содержится информация о ваших реквизитах, которые необходимо заполнить (кнопка *Добавить*) и сохранить для начала работы с оффером.

После заполнения реквизитов, не забудьте поставить галочку в поле «Активно».

1.2.14.2 Для физ.лиц

Раздел содержит ссылку на .

Примечание: Изменение физ.лица на юр.лицо возможно только в том случае, если у вас нет открытой комиссии. Если вам понадобится помощь, обратитесь в службу поддержки.

1.2.15 Помощь

Этот раздел перенаправит вас на [на страницу поддержки](#), на которой вы сейчас находитесь.

1.2.16 Менеджер выгрузок

В этом разделе содержатся задания на выгрузку данных, которые вы запрашивали за все время работы с нашей платформой.

Например, вы заходите в раздел «Заказы/Лиды» и хотите выгрузить список всех заказов конкретного оффера. Для этого, вы нажимаете на кнопку *Экспорт*, выбираете нужный формат выгрузки и видите поп-ап сообщение о созданном задании.

Теперь, задание появится в «Менеджере выгрузок» и отобразится в поле «Задание» и, когда оно будет готово, вы можете его скачать, нажав на соответствующую кнопку в поле «Результат»:

Создано	Задание	Статус	Время создания	Запущено	Закончено	Время выполнения	Результат
2022-02-04 13:01:15	https://my.advcake.com/stat/orders	Завершен	2022-02-04 13:01:15	2022-02-04 13:01:16	2022-02-04 13:01:22	6 сек	Скачать
2022-02-01 15:29:46	https://my.advcake.com/stat/orders	Завершен	2022-02-01 15:29:46	2022-02-01 15:29:47	2022-02-01 15:41:32	705 сек	Скачать

1.3 Как начать работать с Adv.Cake?

На этой странице мы пошагово разберем как начать работать с платформой Adv.Cake.

1.3.1 Шаг 1: Создайте партнерскую ссылку

Партнерская ссылка — ссылка с вашим уникальным идентификатором (меткой вебмастера), ведущая на сайт рекламодателя. С ее помощью можно определить, какой вебмастер привел пользователя на сайт.

Партнерская ссылка размещается на выбранной вами площадке и, если пользователь перейдет по ней на сайт рекламодателя и совершит там целевое действие, вы получите вознаграждение.

Партнерскую ссылку можно создать двумя способами: **вручную** и **динамически**.

1.3.1.1 Вручную

1. Перейдите на вкладку «Создать ссылку» в личном кабинете;
2. Выберите оффер, для которого вы хотите сгенерировать ссылку;
3. Перейдите на сайт оффера и скопируйте адрес страницы, на которую вы хотите вести свою аудиторию;
4. Вставьте адрес страницы в поле *Адрес целевой страницы* (1);
5. Если вы хотите использовать дополнительные параметры в вашей партнерской ссылке для отслеживания статистики, воспользуйтесь *sub-параметрами* (2):

Для настройки доступны шесть дополнительных параметров (один Keyword и пять sub-параметров Sub1-Sub5), которые будут добавлены к ссылке автоматически.

Предупреждение: Максимальное количество символов в полях – 255. Будьте осторожны с использованием спецсимволов, так как они могут негативно повлиять на работоспособность сайта рекламодателя.

Например, у вас есть Telegram-канал и группа ВКонтакте, откуда вы хотите вести пользователей на сайт рекламодателя – сгенерируйте вашу партнерскую ссылку с разными Keyword:

Первая ссылка с Keyword «Vk» для одного источника, а вторая ссылка с Keyword «Tg» для другого источника.

Таким образом, вы получите две разные ссылки – одну для группы ВКонтакте, а другую для Telegram-канала.

6. Нажмите на кнопку *Сгенерировать и скопировать* (3).

Способ генерации ссылки: Ручной Динамический

Адрес целевой страницы ?

1

Дополнительные параметры

Keyword	Sub1	Sub2	Sub3	Sub4	Sub5
<input type="text"/>					

2

3

Ваша партнерская ссылка будет выглядеть так: `https://go.redav.online/{[]hash}`

Подсказка: Если вы хотите вести аудиторию на главную страницу оффера, нажмите на кнопку *Сгенерировать ссылку* - ваша партнерская ссылка автоматически сгенерируется для главной страницы оффера.

1.3.1.2 Динамически

1. Перейдите на вкладку «Создать ссылку» в личном кабинете;
2. Выберите способ генерации ссылки — *Динамический*;
3. Скопируйте шаблон динамической ссылки (1) и установите необходимые параметры:

Вместо {d1} вставьте конечный адрес сайта или страницы (например `https://example.com`);

Вместо {sub1}, {sub2} и т.д., укажите нужные параметры для отслеживания (например `top`, `br`).

Способ генерации ссылки: Ручной Динамический ←

Шаблон динамической ссылки

`https://go.redav.online/99e7f53a709ab961?dl={dl}&sub1={sub1}&sub2={sub2}&sub3={sub3}&sub4={sub4}&sub5={sub5}&keyword={keyword}`

Скопировать в буфер обмена

Пример измененной ссылки: <https://go.redav.online/{hash}?dl=https://www.123.ru/#123&sub1=top>

В качестве примера, представьте, что у вас есть кешбек-сервис и вы хотите понимать какой пользователь совершил заказ. Для этого, в поле {sub1} можно каждый раз передавать необходимое значение, например id пользователя, а затем получать ответ и начислять этому пользователю кешбек.

Или же, у вас есть Excel-файл с несколькими тысячами товаров и вы хотите сформировать deeplink-ссылку на каждый товар, для этого необходимые параметры можно подставить в поле {dl} и на выходе получить готовую ссылку.

Примечание: Если вы не используете дополнительные параметры, можете их удалить.

Вы также можете сгенерировать deeplink-ссылку на любую страницу сайта, в том случае если основной *Основной* лендинг поддерживает deeplink.

Deerlink, или диплинк — разновидность партнерской ссылки, которая ведет на указанную вами страницу сайта рекламодателя — например, на страницу конкретного товара, акции или бренда. Эта функция сокращает количество промежуточных действий пользователя и помогает ему попасть на нужную страницу за минимальное количество кликов.

Например, вы пишете статью про наушники и хотите чтобы пользователь сразу перешел на конкретный товар. Соответственно, вы выбираете *Основной* сайт, который поддерживает deeplink в лендингах и вставляете страницу с товаром в поле «Адрес целевой страницы», после чего нажимаете *Сгенерировать ссылку*.

Примечание: Изучить более подробную инструкцию по генерации ссылок, а также взглянуть на примеры можно нажав на кнопку в разделе «Сгенерировать ссылку»:

1.3.2 Шаг 2: Как использовать партнерскую ссылку?

Размещайте партнерские ссылки на своих рекламных площадках.

Если по вашей ссылке будет оформлена заявка, вы увидите ее в вашем личном кабинете в разделе «Статистика» → «Общая», «Заказы/Лиды».

Если заказ будет оплачен пользователем, которого вы привели по своей партнерской ссылке, вы получите вознаграждение (например, % от суммы оплаченного курса).

Размер вознаграждения зависит от условий и акций, которые проводятся рекламодателем.

1.3.3 Шаг 3: Размещайте баннеры

1. Перейдите в раздел «Промоматериалы» → «Баннеры»;
2. Нажмите на название нужного вам баннера;
3. Введите адрес целевой страницы (1);
4. Введите sub-параметры, при необходимости (2);
5. Если вы хотите, чтобы баннер открывался в новой вкладке, поставьте галочку в поле (3);
6. Нажмите на кнопку *Скопировать в буфер обмена* (4);
7. Вставьте скопированный код на ваш сайт.

The screenshot shows a form for configuring a banner. At the top, there is a field labeled 'Адрес целевой страницы' (Target page address) with a red box and a '1' callout, containing the text 'http://site.com/path'. Below this are five sub-parameter input fields: 'Sub1', 'Sub2', 'Sub3', 'Sub4', and 'Sub5', with a red box and a '2' callout encompassing them. A checkbox labeled 'Открывать в новой вкладке' (Open in new tab) with a '3' callout is located below the sub-parameters. At the bottom, there is a 'Результат' (Result) section containing a code block with JavaScript and HTML code for the banner. A button labeled 'Скопировать в буфер обмена' (Copy to clipboard) with a '4' callout is positioned below the code block.

Подсказка: Если вам нужен баннер в формате .png, нажмите на необходимый баннер правой кнопкой мыши и выберите *Сохранить как...*

Примечание: Если вы используете баннер как картинку, не забудьте *сгенерировать партнерскую ссылку* и разместить ее так, чтобы при клике на размещенный баннер пользователь переходил по вашей партнерской ссылке.

1.3.4 Шаг 4: Отслеживайте статистику

После размещения вашей партнерской ссылки вы можете отслеживать статистику в разделе «Статистика» → «Общая».

Данные по новому заказу появятся в личном кабинете в течение часа после его совершения.

Новый заказ – это созданный заказ на сайте рекламодателя, который пока не получил конечный статус *Одобрен (Заказ оплачен)* или *Отменен (Заказ отменен по какой-либо причине)*.

Обновление статуса по заказам (Одобрен или Отклонен) происходит раз в месяц.

1.3.5 Шаг 5: Получайте вознаграждение

1.3.5.1 Для участников реферальной программы

1. Перейдите в раздел «Финансы» → «Данные физ.лица»;
2. Внесите ваши данные и нажмите кнопку *Сохранить*;
3. Перейдите на вкладку «Данные участника»;
4. В блоке «Способ оплаты» нажмите кнопку *Добавить* и добавьте способ вывода средств.

Предупреждение: Выплата вознаграждения (согласованная комиссия) производится автоматически в течение 10 рабочих дней после оплаты счета рекламодателем. Если на момент проведения выплаты нет доступного способа выплаты, она переносится еще на 10 рабочих дней.

1.3.5.2 Для физ.лиц/самозанятых

1. Перейдите в раздел «Финансы» → «Данные самозанятого»;
2. Внесите ваши данные и нажмите кнопку *Сохранить*;
3. Перейдите на вкладку «Выплаты для физ.лиц/самозанятых»;
4. В блоке «Способ оплаты» нажмите нажмите кнопку *Добавить* и добавьте способ вывода средств.

Предупреждение: Выплата вознаграждения (согласованная комиссия) производится автоматически в течение 10 рабочих дней после оплаты счета рекламодателем. Если на момент проведения выплаты нет доступного способа выплаты, она переносится еще на 10 рабочих дней.

1.3.5.3 Для юр. лиц

Если в разделе «Финансы» -> «Выплаты для юр.лиц» у вас появился счет, который вы можете выставить:

1. Перейдите в раздел «Документы» -> «Реквизиты»;
2. Нажмите кнопку *Добавить* и внесите ваши реквизиты;
3. После того как менеджер добавит в систему номер вашего договора, у вас появится возможность скачать комплект закрывающих документов из личного кабинета;
4. Скачайте комплект закрывающих документов и отправьте сканы подписанных документов на support@advcake.com, а два подписанных экземпляра по адресу: *123001, г. Москва, ул. Большая Садовая 5, к.2, офис 2038, 4 этаж.*

Примечание: Закрывающие документы за отчетный период формируются ежедневно, после окончания сверки с рекламодателем до 25 числа месяца, следующего за отчетным. Комплект документов отправляется на Email зарегистрированного аккаунта.

Подсказка: Подробнее о статусах выплат и их различиях вы можете узнать в разделе «Финансы».

1.4 ОРД

1.4.1 Информация о передаче данных в ОРД при работе с платформой Adv.Cake

В связи с вступлением в силу с 1 сентября 2022 г. изменений в ФЗ-38 «О Рекламе» и наличием с вашей площадкой договора на размещение РИМ (рекламно-информационных материалов) в сети Интернет с 1 сентября 2023г. мы, по закону, выполняем следующие обязанности:

1.4.1.1 Отчетность

1. Adv.Cake берет на себя обязанность передавать отчетность в ОРД по вашим размещениям;
2. Adv.Cake берет на себя обязанность исполнять требования разметки рекламы токенами (идентификаторами) – они вшиты во все ваши рекламные ссылки.

<p>Предупреждение: За отсутствие отчетности размер штрафа составляет до 500 000 руб.</p>

1.4.1.2 Маркировка рекламных материалов

Adv.Cake берет на себя обязанность наносить маркировку на графические изображения (баннеры), выдаваемые в вашем личном кабинете на . Подробнее о маркировке рекламных материалов читайте ниже.

1.4.2 Информация о маркировке рекламных материалов

Маркировка рекламных материалов – это нанесение информации о рекламном характере информационного блока.

Как мы упомянули ранее, команда Adv.Cake берет на себя обязанность наносить маркировку на графические изображения (баннеры), выдаваемые в вашем личном кабинете на .

Обязанность партнера, т.е. вас – промаркировать все остальные рекламные материалы, а именно: посты в группах, статьи на сторонних и собственных сайтах и т.д.

Примечание: Размещаемая вами информация, независимо от ее формата (пост, статья, рекомендация и т.п.), должна быть промаркирована фразой:

Реклама. Информация о рекламодателе на сайте site.ru

Наши рекомендации и пояснения к возможным вариантам маркировки:

1. Telegram-каналы, группы в соцсетях и т.д: **Реклама. Информация о рекламодателе по ссылкам в посте.**
2. Сайты, статьи, блоги и т.д: **Реклама. Информация о рекламодателе по ссылкам в статье.**

Важно, чтобы все страницы, на которых находятся партнерские ссылки, были промаркированы данной фразой. Это можно сделать внизу статьи, но важно, чтобы шрифт, цвет текста были читабельны.

3. Cashback-сервисы и промокодные площадки: **Реклама. Информация о рекламодателе по ссылке в карточке акции.**

Мы предлагаем делать маркировку внизу каждой карточки рекламодателя.

1.4.3 Как стать самозанятым и зачем это нужно

1.4.3.1 Зачем становиться самозанятым

В связи с ФЗ-38 «О рекламе» все ваши размещения должны маркироваться, а информация по вашим размещения должна передаваться в ОРД.

Передать данные в ОРД по физ.лицу невозможно, поэтому, при сотрудничестве с партнерскими программами, вы можете использовать юридическое лицо (ИП, ООО и т.д.) или самозанятость.

1.4.3.2 Кому подходит режим самозанятости

1. Вы ведете деятельность на территории РФ;
2. Вы зарабатываете до 2,4 миллионов рублей в год без учёта официальной зарплаты;
3. Вы гражданин РФ;
4. Вам больше 18 лет.

1.4.3.3 Как получить статус самозанятого

1. Скачайте мобильное приложение или пройдите регистрацию на сайте «Мой Налог».

Подсказка: Если у вас возникли вопросы по регистрации, посмотрите ответы на популярные вопросы.

Для получения статуса вы также можете использовать авторизованные банки.

2. После получения статуса самозанятого:

- Сообщите информацию в службу поддержки Adv.Cake;
- Привяжите карту, на которую вы планируете получать выплату;
- После проведения первой выплаты примите приглашение в личном кабинете «Моего налога» от «Киви Банка» на проведение выплат и их регистрацию.

Примечание: При сотрудничестве с Adv.Cake вам не потребуется дополнительно отчитываться и присылать чеки в ФНС – все выплаты приходят автоматически с регистрацией чеков в системе «Мой налог» банком «Киви Банк».

1.5 Вопросы и ответы

Здесь мы ответили на дополнительные вопросы, которые могли у вас возникнуть.

1.5.1 Общие вопросы

Как зарегистрироваться в Adv.Cake?

1.5.2 Финансовые вопросы

Какие способы выплат доступны для вебмастеров?

Как и когда поступит выплата за подтвержденные заказы?

Где найти историю выплат?

Как добавить платежную информацию?

1.5.3 Технические вопросы

Нужен ли новый, не привязанный к сайту домен для парковки домена?

Как настроить SSL сертификат при парковке домена?

1.5.4 Вопросы по офферам

Собираем вопросы, пишем ответы.

1.5.5 Прочие вопросы

Есть ли требования к маркировке рекламных материалов?

1.5.5.1 Как зарегистрироваться в Adv.Cake?

Зарегистрироваться в нашей системе можно двумя способами:

1. С помощью наших менеджеров;
2. Самостоятельно по партнерской ссылке.

Если с первым пунктом все понятно, то второй пункт стоит разобрать чуть подробнее.

На данный момент вебмастеры могут регистрироваться в нашей системе через партнерские ссылки. Возникает вопрос — где взять такую ссылку?

Возьмем к примеру сайт «Эльдорадо». Если вебмастер хочет с ним работать, то он может найти партнерскую ссылку непосредственно на сайте или написав в поисковом запросе «партнерская программа эльдорадо», либо каким-то другим способом.

Затем вебмастер переходит по ссылке, регистрируется по ней (если он раньше не регистрировался), либо подает заявку на работу с оффером «Эльдорадо» (если вебмастер уже зарегистрирован в нашей системе).

1.5.5.2 Какие способы выплат доступны для вебмастеров?

Вебмастерам доступны следующие способы выплат:

1. Для вебмастеров с гражданством РФ, оформивших самозанятость:

- Банковская карта Visa/Mastercard/Мир (Руб.)

2. Для вебмастеров с гражданством других стран:

- Банковская карта Visa/Mastercard (€)
- Банковская карта Visa/Mastercard (\$)
- Банковская карта Visa/Mastercard (другая валюта)
- Банковский перевод (иностранное физ. лицо) (\$)
- Банковский перевод (иностранное физ. лицо) (€)
- PayPal (\$)
- PayPal (€)

- Криптокошелек USDT TRC-20
 - Криптокошелек USDC TRC-20
-

3. Для вебмастеров, работающих как юридическое лицо с регистрацией вне РФ и СНГ:

- Банковский перевод (иностранное юр. лицо) (\$)
 - Банковский перевод (иностранное юр. лицо) (€)
-

4. Для вебмастеров, работающих как юридическое лицо с регистрацией РФ и СНГ:

- Банковский перевод на расчетный счет

1.5.5.3 Как и когда поступит выплата за подтвержденные заказы?

Выплата за подтвержденные заказы происходит после того, как рекламодатель проверит заказы и оплатит их.

Для вебмастеров **физ.лиц** выплата вознаграждения (т.е. «согласованная комиссия») производится автоматически в течение 10 дней после оплаты счета рекламодателем.

Предупреждение: Если на момент проведения выплаты нет доступного способа выплаты, она переносится еще на 10 дней.

Не забудьте добавить способ выплат!

Для вебмастеров **юр.лиц** оплата производится в течение 15 рабочих дней после получения выплаты от рекламодателя.

Примечание: Более подробную информацию по оплате смотрите в разделе «Оплата» в вашем личном кабинете.

1.5.5.4 Где найти историю выплат?

Для вебмастеров **юр.лиц**: перейдите в раздел «Оплата» → «Счета».

Для вебмастеров **физ.лиц**: перейдите в раздел «Выплаты» → «Движения по счету».

Для **участников реферальной программы**: перейдите в раздел «Данные участника» → «Движения по счету».

1.5.5.5 Как добавить платежную информацию?

Для вебмастеров **юр.лиц**: перейдите в раздел «Договоры» -> «Реквизиты».

Для вебмастеров **физ.лиц**: перейдите в раздел «Выплаты» -> «Способы оплаты» -> *Добавить*.

Для **участников реферальной программы**: перейдите в раздел «Данные участника» -> «Способы оплаты» -> *Добавить*.

1.5.5.6 Нужен ли новый, не привязанный к сайту домен для парковки домена?

Нет, подойдет любой. На домене нужно прописать запись `cname` для поддомена третьего уровня и использовать этот поддомен.

Подсказка: Чтобы узнать больше, воспользуйтесь кнопкой *Подробное описание парковки доменов* в разделе «Обход блокировки рекламы» -> «Парковка домена».

После парковки домена, обратите внимание на статус вашего домена:

1. Активен – припаркованный домен активен;
2. Ожидание активации — добавьте запись в DNS и подождите активации, обычно она выполняется в течении 15 минут;
3. Ошибка DNS — запись в DNS была удалена, добавьте запись в DNS, иначе домен будет удален через двое суток.

1.5.5.7 Как настроить SSL сертификат при парковке домена?

При парковке домена сертификаты выдаются автоматически, вебмастеру нужно только прописать запись в DNS.

1.5.5.8 Есть ли требования к маркировке рекламных материалов?

Да, есть. Требования к маркировке рекламных материалов перечислены на .

1.6 Глоссарий

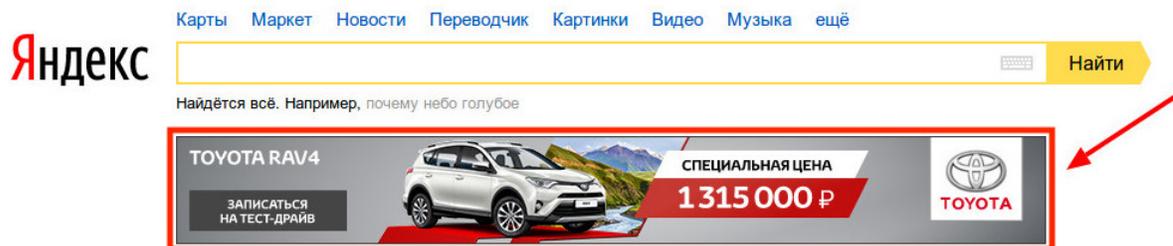
Здесь содержатся основные термины и определения, которые активно используются в практике интернет-маркетинга.

Антифрод

Одна из услуг, входящих в ведение CPA-канала, при помощи которой рекламодатели получают наши рекомендации по блокировке недобросовестных вебмастеров.

Баннер

Изображение рекламного характера, содержащее в себе ссылку на сайт рекламодателя. Является самым распространенным видом интернет-рекламы.



Баннерная реклама

Рекламные объявления, которые показываются в рекламных блоках на сайтах, сотрудничающих с рекламной сетью. В размещении баннерной рекламы могут быть задействованы любые рекламные сети, в том числе Рекламная Сеть Яндекса (РСЯ) и Контекстно-Медийная Сеть Google (КМС).

Блогеры

Владельцы аккаунтов в YouTube, Instagram*, TikTok, Дзен и т.д., которые размещают рекламные посты для своих подписчиков.

Брокерский трафик

Тип трафика, используемый преимущественно при работе с финансовыми программами. В данном случае вебмастер выступает как посредник (брокер), который собирает в офлайн- или онлайн-режиме кредитные данные пользователей, заполняет от их лица заявки на выдачу кредита, кредитной карты, займа и отправляет со своей площадки CPA-сети, где заявки проходят проверку и передаются соответствующему рекламодателю.

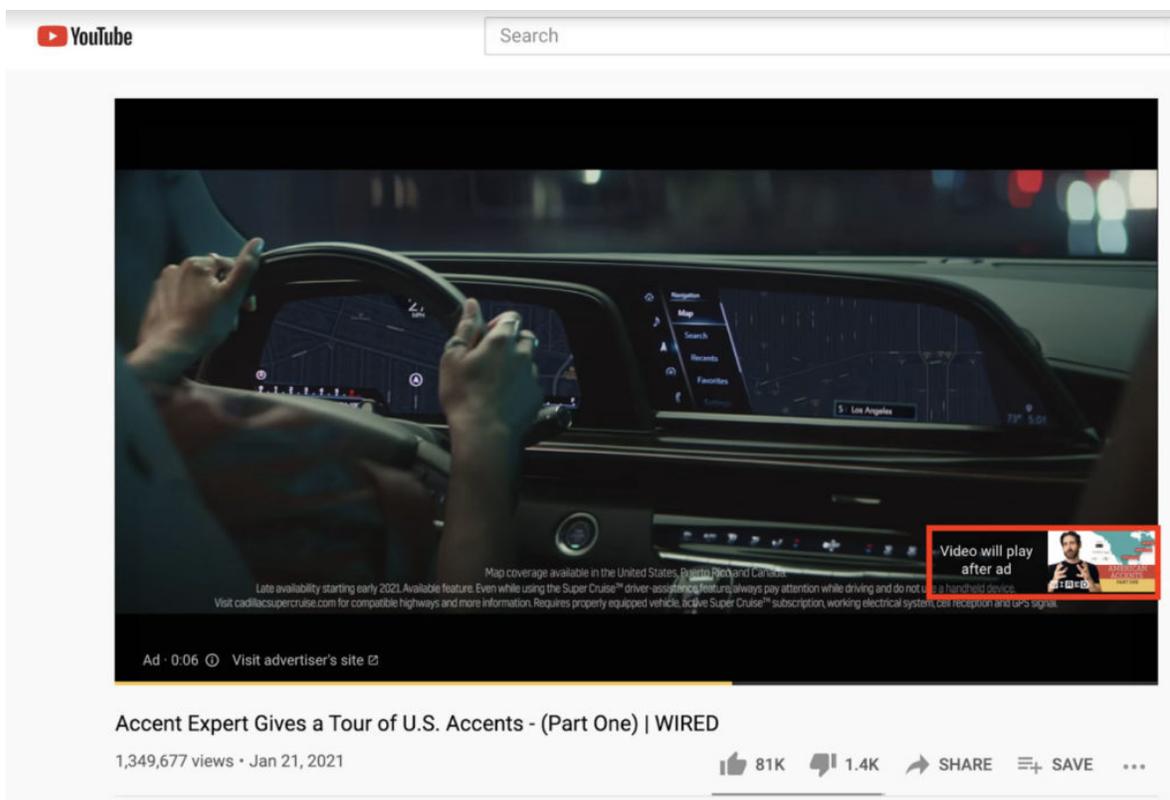
Вебмастер

Физическое или юридическое лицо, которое имеет ресурс и возможность направить пользователя на сайт потенциального рекламодателя размеченный трафик.

За каждое совершенное пользователем целевое действие вебмастер получает от рекламодателя вознаграждение.

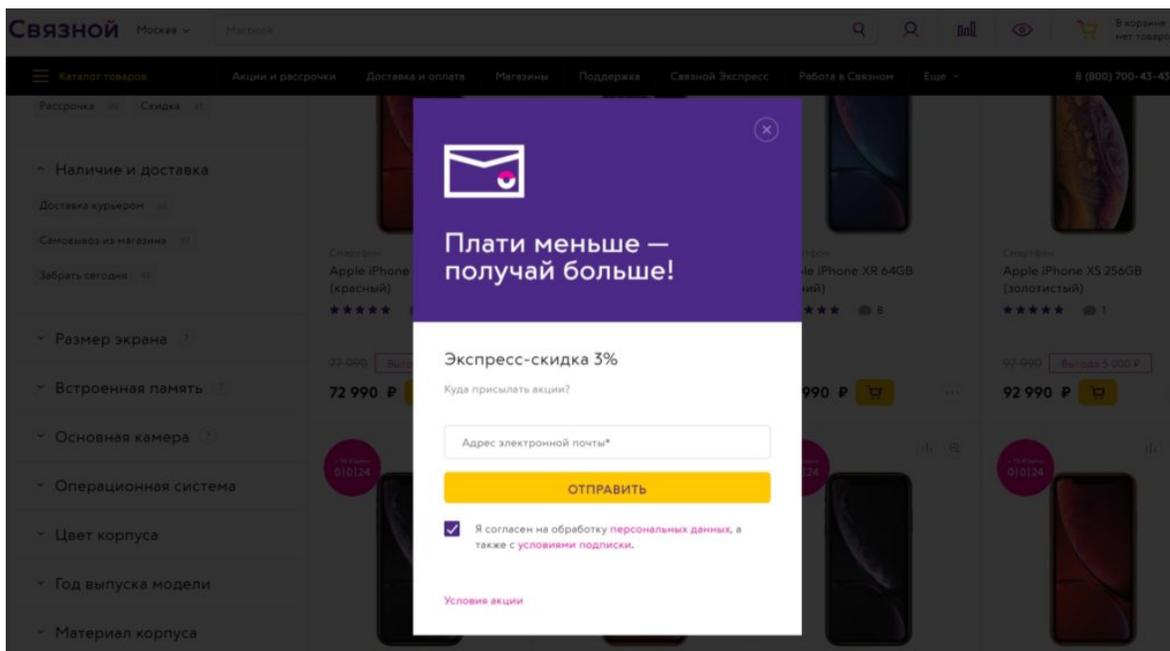
Видеореклама

Рекламные размещения в формате видеороликов в основных видеоплатформах Youtube, Яндекс Видео, MyTarget.



Виджет

Небольшое вспомогательное приложение, нацеленное на выполнение какой-либо конкретной задачи. Например, с их помощью можно быстро авторизоваться на сайте через соц.сети, вести обратную связи с клиентами, предлагать скидки и купоны, и т.д.



Витрина

Посредник между продавцом и покупателем, предоставляющий покупателю возможность вы-

брать из множества товаров с различными характеристиками необходимый ему товар, а также найти продавца с самыми выгодными условиями.

Вознаграждение

Денежное поощрение, которое рекламодатель выплачивает вебмастеру за генерацию лидов и продаж в рамках партнерской программы. Размер вознаграждения зависит от условий и акций, которые проводятся рекламодателем.

Группы в соц. сетях

Размещение рекламного объявления в группах/сообществах VK, FB*, ОК.

Действие

Действие, которое рекламодатель хочет получить от пользователя, например, оформление заказа, заполнение заявки, регистрация на сайте, покупка курса, установка приложения и т.д.

Дорвей

«Вспомогательный» сайт, задача которого — попадание на высокие места в результатах поисковой выдачи и перенаправление пользователей на продвигаемый интернет-ресурс. По умолчанию считается запрещенным типом трафика.

Интеграция

Один из этапов запуска партнерской программы в Adv.Cake: настройка трекинга действий на сайте рекламодателя и механизма отправки данных о действиях на сторону Adv.Cake. Интеграция обязательна для работы партнерской программы.

Подробнее об интеграции и ее видах можно узнать на странице .

Источник трафика

Интернет-пространство, где размещена ссылка, по которой пользователь (который и является трафиком) перешел на сайт рекламодателя. Источником трафика могут быть сайт, группа в социальной сети, YouTube-канал, email-рассылка, контекстная или таргетированная реклама.

Клик

Переход пользователя из рекламного объявления на целевую страницу.

Контекстная реклама

Тип интернет-рекламы, при котором рекламные объявления отображаются на странице поисковой выдачи в соответствии с поисковым запросом и интересом пользователя.

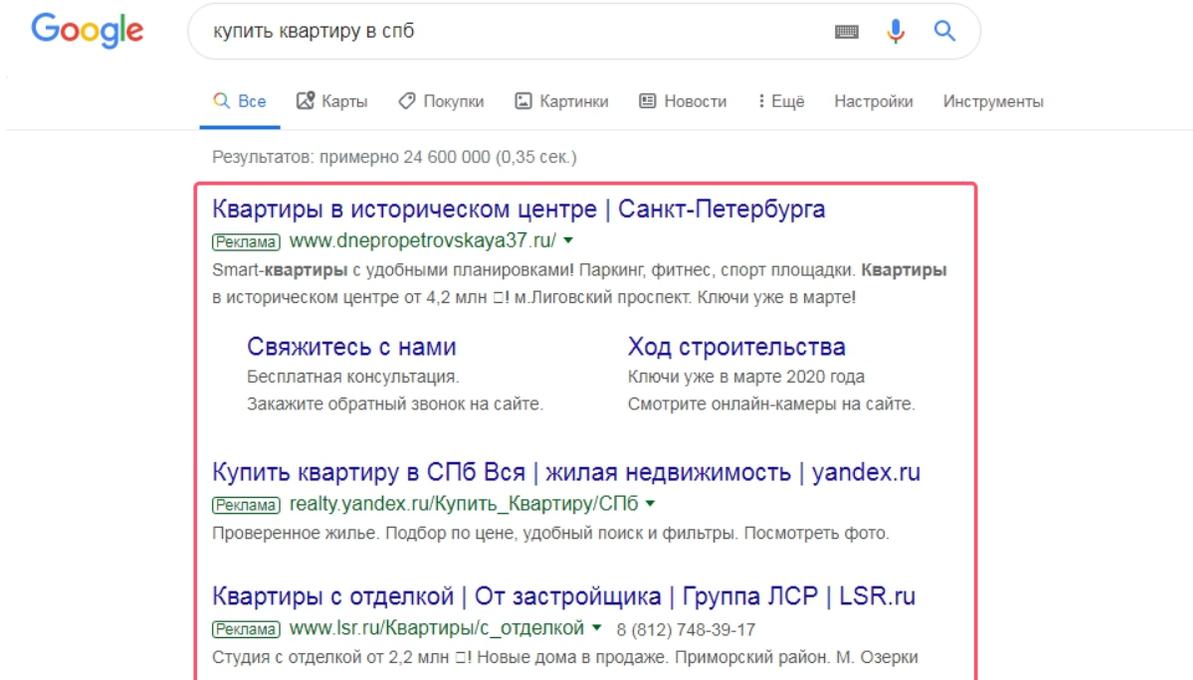
У контекстной рекламы есть ряд особенностей:

1. Рекламное объявление показывается только заинтересованным пользователям;
2. В отличие от модели CPA, где оплачивается целевое действие, в контекстной рекламе оплата за размещение объявления производится по модели CPC (оплата за клик), причем веб-мастер сам устанавливает стоимость клика;
3. Выбор объявлений для показа производится в результате своеобразного «аукциона», в ходе которого система сравнивает цену за клик всех объявлений и выбирает объявления с наибольшей ценой.

Контекстная реклама делится на *поисковую* и *тематическую*:

Поисковая реклама — демонстрируется среди результатов поиска в поисковых системах, таких как Google или Яндекс.

Рекламные блоки могут располагаться над результатами выдачи, под ними или справа от них, а формируются они следующим образом: при создании рекламного объявления вебмастер прописывает определенные ключевые слова или фразы и если они совпадают с запросом пользователя — объявление появляется в результатах выдачи.



Тематическая реклама — демонстрируется на страницах сайтов и в мобильных приложениях, входящих в рекламную сеть, например Рекламная Сеть Яндекса или Контекстно-Медийная Сеть Google.

В отличие от поисковой, тематическая реклама появляется не в выдаче, а на страницах сайтов соответствующей тематики.



Контекстная реклама на бренд

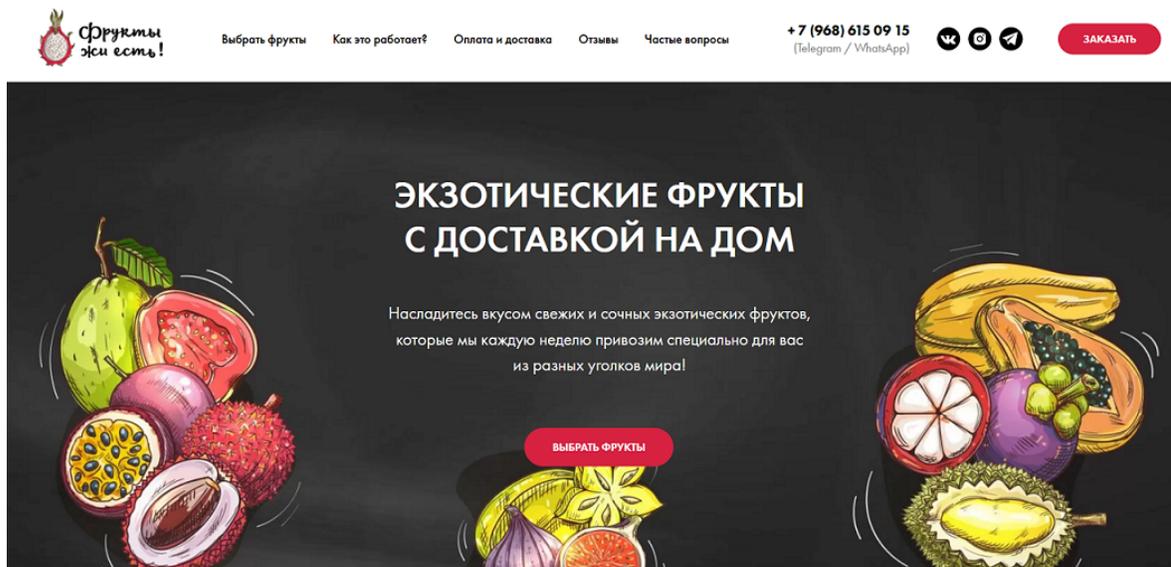
Разновидность контекстной рекламы, при которой в ключевых словах рекламного объявления содержится название бренда рекламодателя, его переводы, транслитерация и производные. По умолчанию считается запрещенным типом трафика.

Лендинг

Посадочная страница (от англ. «landing page»), на которую пользователь переходит после клика по баннеру, контекстному объявлению, ссылке в паблике в соцсетях или любому другому рекламному объявлению.

Главная задача посадочной страницы — побудить пользователя совершить целевое действие.

Пример посадочной страницы:



Лид

Данный термин (от англ. «Lead») употребляется в двух значениях: лид как заинтересовавшийся пользователь (потенциальный клиент) и как вид целевого действия.

Личный кабинет

Виртуальный личный кабинет, расположенный на серверах Adv.Cake по адресу , доступ к которому осуществляется после авторизации (ввода логина и пароля), в пределах которого стороны обмениваются технически и юридически значимой информацией.

Метка/UTM-метка

Небольшой фрагмент кода, который добавляется в адресную строку веб-сайта.

Каждый параметр в UTM-метке состоит из двух частей — название параметра и его значение:



Где:

1. *utm_source/utm_medium/utm_campaign* — названия параметров UTM;
2. *vk/cpc/campaign12* — их значения.

А значения этих меток указывают на следующее:

1. *vk* — рекламная система или источник, который привел пользователя на сайт;
2. *cpc* — модель оплаты интернет-рекламы (оплата за клик);
3. *campaign12* — название рекламной кампании.

Мотивированный трафик

Трафик, который привлекают вебмастеры, обещая пользователям какое-либо вознаграждение (баллы, бонусы, подарки) за установку приложения, регистрацию, заполнение заявки или выполнение другого действия. По умолчанию считается запрещенным типом трафика.

Партнерская программа (оффер)

Форма делового сотрудничества между рекламодателем и веб-мастером, в рамках которого веб-мастер размещает на своей площадке рекламные материалы рекламодателя и привлекает трафик на его сайт, а рекламодатель за это перечисляет ему вознаграждение.

Под рекламодателями понимаются компании-владельцы интернет-магазинов, банков, турфирм и др., рекламирующие свои товары и услуги в Интернете.

Вебмастеры — пользователи, ведущие активное общение в Интернете, занимающиеся размещением объявлений или рекламы, владельцы веб-ресурсов, сообществ в соц. сетях, онлайн-приложений или игр, — одним словом, те пользователи, которые способны направить трафик на сайт рекламодателя.

Вознаграждение может выплачиваться в виде процента от продаж либо фиксированной суммой за целевое действие, совершенное клиентом, которого привел вебмастер.

Партнерская сеть

Платформа-посредник, которая помогает рекламодателями и вебмастерам находить друг друга.

Поскольку сотрудничество с каждым вебмастером требует определенных время- и трудозатрат, партнерская сеть берет на себя основную операционную деятельность — производит интеграцию с рекламодателями, осуществляет выплаты вебмастерам, согласовывает повышения ставок и транслирует рекламные материалы. За выполнение этих функций партнерская сеть получает комиссию.

Партнерская ссылка

Ссылка, ведущая на сайт партнерской программы. В партнерской ссылке содержится уникальный для каждого вебмастера идентификатор, по которому можно определить, какой вебмастер привел пользователя на сайт.

Показ

Демонстрация рекламных объявлений на сайте вебмастера.

Пользователь

Посетитель информационных ресурсов в Интернете.

Постбек

Сервис, предоставляющий возможность получать информацию о трафике, а именно: отслеживать и анализировать информацию в трекаре о совершенных целевых действиях пользователей на странице оффера.

Приложения (игры) в соц. сетях

Рекламные объявления, которые показываются в рекламных блоках в приложениях и играх, размещенных в соц. сетях и сотрудничающих с рекламной сетью.

Программа лояльности

Программа, за участие в которой, авторизованный пользователь получает определенную выгоду, например, скидки, кэшбек, подарки или иное вознаграждение. Участие в программе для пользователя подразумевает факт покупки с выполнением определенных условий.

Промокодная площадка

Сайт, на котором размещаются промокоды различных Интернет-магазинов, позволяющие получить выгоду при покупке.

Реклама

Информация о товаре/услуге, размещаемая на страницах сайтов и содержащая в себе гипертекстовую ссылку на сайт рекламодателя.

Рекламная площадка

Интернет-пространство, в котором размещается реклама: сайт, почтовая рассылка, социальная сеть и т.д.

Например, когда владелец сайта размещает на нем баннеры или видеоблогер вставляет под видео ссылку на товар, их сайт или видеоблог становятся рекламными площадками.

Рекламодатель

Физическое или юридическое лицо, занимающееся продажей товаров или предоставлением услуг в сфере B2C и имеющее партнерскую программу, размещенную в партнерской сети или на собственном сайте.

Ретаргетинг

Механизм, позволяющий показывать рекламу пользователям, которые уже посещали сайт рекламодателя, но по каким-либо причинам не совершили покупку или иное целевое действие.

Реферал

Пользователь, зарегистрировавшийся в партнерской сети, перейдя по уникальной реферальной ссылке реферера, за что последний получает вознаграждение.

Реферальная программа

Программа, созданная для привлечения новых пользователей в партнерскую сеть.

Реферальная ссылка

Уникальная ссылка, которую получает каждый участник реферальной программы для привлечения новых пользователей.

Реферер

Пользователь, который приводит рефералов и получает за них вознаграждение; пользователь, разместивший реферальную ссылку.

Сайт

Любой контентный сайт, имеющий определенную аудиторию и размещающий рекламу в определенных рекламных блоках.

Сверка

Процесс проставления обработанным действиям финальных статусов в статистике. Сверка может быть автоматической или ручной.

Сессия

Время присутствия пользователя на сайте. Сессия начинается, когда пользователь переходит на сайт и заканчивается в момент закрытия страницы сайта или браузера.

Таргетинг

Рекламный механизм, позволяющий показывать рекламу только той части аудитории, которая удовлетворяет заданным критериям. Такая аудитория называется целевой. Критериями могут быть возраст, пол, географическое положение (геотаргетинг), увлечения и интересы (контекстная реклама) и т.д.

Используя таргетинг, рекламодатель может значительно снизить рекламные издержки, работая только с потенциально заинтересованной в его продукте аудиторией.

Таргетированная реклама

Тип рекламы в социальных сетях, которая показывается целевой аудитории с определенными параметрами: возрастом, полом, регионом проживания, интересами и т.д.

Тизерная реклама

Рекламное размещение в виде небольшой картинки с интригующим или провокационным текстом. Главная задача тизера — возбудить любопытство пользователя и «заставить» его кликнуть на картинку.

Товарный виджет

Инструмент, являющийся частью публикации, который соответствует теме и дополняет ее. Виджеты, чаще всего, используются в статьях.

Например, в статье о ТВ-приставках, блогер может рассказать какой приставкой пользуется сам и разместить соответствующий товарный виджет.

The screenshot shows a page from Yandex Market. At the top, there is a navigation bar with the Yandex logo, a search bar containing 'Искать товары', and a 'Найти' button. Below the navigation bar, there are location and category filters: 'Санкт-Петербург', 'Новый год', 'Экспресс', 'Одежда', 'Дом', 'Электроника', 'Бытовая техника', 'Детям', 'Спорт', and 'Продукты'. The main content area features an article titled 'Что умеет приставка Apple TV и зачем она нужна' with a sub-header 'ОБЗОР'. The article text discusses the author's experience with Apple TV. To the right of the text is a 'СОДЕРЖАНИЕ' (Table of Contents) section with six numbered items. At the bottom of the article, there is a product widget for the 'ТВ-приставка Apple TV 4K 64GB' with a price of 'от 17 990 ₽' and a rating of '4.7' from '79 отзывов'. A red arrow points to this widget.

Товарный фид

Файл определенного типа, который содержит данные о товарных предложениях. Вебмастера могут использовать фиды для создания динамических объявлений и смарт-баннеров.

Трафик, объем трафика

Трафик — поток пользователей, перешедших по партнерской ссылке на сайт рекламодателя.

Объем трафика — количество уникальных посетителей сайта за определенный период времени.

Трекинг

Процесс атрибуции трафика и сбора данных о нем для дальнейшей обработки, аналитики и оптимизации рекламных кампаний.

Форум

Тематический сайт для общения между пользователями интернета на одну или несколько интересующих их тем.

Фрод

Мошеннические действия для присвоения заказов, приведенных по другим каналам или другими партнерами.

К фроду можно отнести создание «копий» сайта рекламодателя, накрутку, спам, использование запрещенных источников трафика и т.д.

Целевая аудитория

Потенциальные потребители продукта или услуги, которые являются «целью» при проведении рекламных кампаний.

Целевая аудитория может быть определена следующими признаками:

1. географическими;
2. социально-демографическими: пол, возраст, национальность, семейное положение, образование, доход, место работы;
3. психологическими: интересы, взгляды, жизненная позиция.

Affiliate

Партнерская сеть, созданная компанией Adv.Cake, отличающаяся от других партнерских сетей повышенными ставками для вебмастеров за счет снижения комиссии сети.

AOP

AOP (от англ. «Average order price») — средняя выплата (вознаграждение) за одобренный заказ.

AOV

AOV (от англ. «Average order value») — средний доход с каждого одобренного заказа.

API

API (от англ. «Application Programming Interface») — интерфейс, который обеспечивает взаимодействие между приложениями. Это готовый код, предназначенный для выполнения определенной функции. При наличии API, разработчик не пишет код с нуля, а использует готовые решения, что облегчает и ускоряет работу.

AR

AR (от англ. «Approval rate») — процент одобренных заказов от общего количества заказов.

BR

BR (от англ. «Bounce rate») — процент посетителей, покинувших сайт прямо со страницы входа или просмотревших не более одной страницы сайта.

Cashback

Частный случай программы лояльности — отсроченная скидка, возвращаемая покупателю реальными деньгами по истечении периода, отведенного на обмен/возврат заказа.

Clickunder

Рекламное сообщение, открывающееся в новом окне при первом клике на любое место страницы сайта. По умолчанию считается запрещенным типом трафика.

Cookie

Небольшой фрагмент данных с информацией о пользователе, которая была собрана во время его пребывания на сайте.

Cookie передается с сервера сайта и хранится в браузере пользователя. При следующем посещении этого сайта пользователем сервер сайта запросит данные cookie у браузера, чтобы восстановить сведения о нем.

Применяется для сохранения данных на стороне пользователя, на практике обычно используется для:

1. аутентификации пользователя;
2. хранения персональных предпочтений и настроек пользователя;
3. отслеживания состояния сеанса доступа пользователя;
4. сведения статистики о пользователях.

Также, стоит упомянуть **время жизни cookie** — тот срок действия, в течение которого cookie, полученные браузером пользователя и связавшие его с конкретным вебмастером, будут храниться на у пользователя на компьютере.

Если в течение этого времени пользователь совершит целевое действие, он будет учитываться как приведённый вебмастером.

CPA

CPA (от англ. «Cost Per Action» — цена за действие) — модель оплаты интернет-рекламы, при которой оплачиваются только определенные действия пользователей на сайте рекламодателя.

CPA-модель является одним из самых экономически эффективных вариантов оплаты рекламы, поскольку рекламодатель платит не за показы или клики, эффективность которых крайне сложно измерить, а за конкретных потребителей, подтвердивших интерес к продукту целевыми действиями.

Чаще всего в качестве целевого действия в CPA используется оплаченный заказ или услуга.

CPC

CPC (от англ. «Cost Per Click» — цена за клик) — модель оплаты интернет-рекламы, при которой рекламодатель платит вебмастеру вознаграждение за клик пользователя по размещенному вебмастером рекламному баннеру или ссылке.

ECPC

ECPC (от англ. «Effective Cost Per Click») — это эффективная стоимость одного перехода (клика) в CPA-сетях (или сайтах с партнерскими программами), которая рассчитывается по следующей формуле: *сумма заработка ÷ количество кликов*.

CPI

CPI (от англ. «Cost Per Impression» — оплата за показы) — метод оплаты интернет-рекламы, при котором стоимость рекламной компании определяется числом показов рекламного объявления.

CPL

CPL (от англ. «Cost per Lead») — модель оплаты интернет-рекламы, в рамках которой рекламодатель оплачивает контактные данные потенциального клиента.

CPM

CPM (от англ. «Cost-Per-Thousand» — цена за тысячу показов) — рекламная модель, в которой цена устанавливается за тысячу показов баннера рекламодателя или другими словами за показ рекламного блока одной тысяче посетителей.

CR

CR (от англ. «Conversion Rate») — процентное отношение количества целевых действий пользователей к общему числу переходов (кликов) по партнерской ссылке. Проще говоря, это процент пользователей, которые осуществили нужное действие — зарегистрировались, подписались на рассылку, заполнили форму, подали заявку, оформили покупку и т.д.

CTR

CTR (от англ. «Click-through rate» — показатель кликабельности) — определяется как отношение числа кликов на баннер или рекламное объявление к числу показов, измеряется в процентах.

CTR вычисляется по следующей формуле: $\text{количество кликов} \div \text{количество показов} * 100$.

Например: Ваш рекламный блок был показан 100 раз и на него кликнули 5 человек. Значит по формуле $(5 \div 100 * 100)$ CTR будер равен 5%.

CPS

CPS (от англ. «Cost Per Sale») — модель оплаты интернет-рекламы, при которой рекламодатель платит за покупки, совершенные привлеченными на сайт пользователями.

CPV

CPV (от англ. «Cost Per View») — метод оплаты интернет-рекламы, при котором рекламодатель платит за просмотр каких-либо материалов: рекламных баннеров, картинок, видеороликов и т.д.

Deerlink (дипплинк)

Разновидность партнерской ссылки, которая ведет на указанную вами страницу сайта рекламодателя — например, на страницу конкретного товара, акции или бренда.

Эта функция сокращает количество промежуточных действий пользователя и помогает ему попасть на нужную страницу за минимальное количество кликов.

EPL

EPL (от англ. «Earnings Per Lead») — величина для оценки среднего заработка с совершенного целевого действия, отношение подтвержденных финансов к количеству целевых действий.

Keyword

Ключевая фраза, по которой было показано объявление.

Last Paid Click / Last Cookie Wins

Модель атрибуции, согласно которой действие засчитывается тому платному источнику, чья ссылка была последней, на которую кликнул пользователь перед совершением действия.

(*) Продукт компании Meta Platforms Inc., — организации, деятельность которой признана экстремистской и запрещена на территории РФ.

Участникам реферальной программы

2.1 Что такое реферальная программа?

Реферальная программа – это маркетинговый инструмент, который позволяет стимулировать процесс так называемого сарафанного радио. Упрощенно это выглядит так – компания поощряет своего клиента каким-нибудь подарком за то, что он посоветует другу приобрести продукт или услугу этой компании.

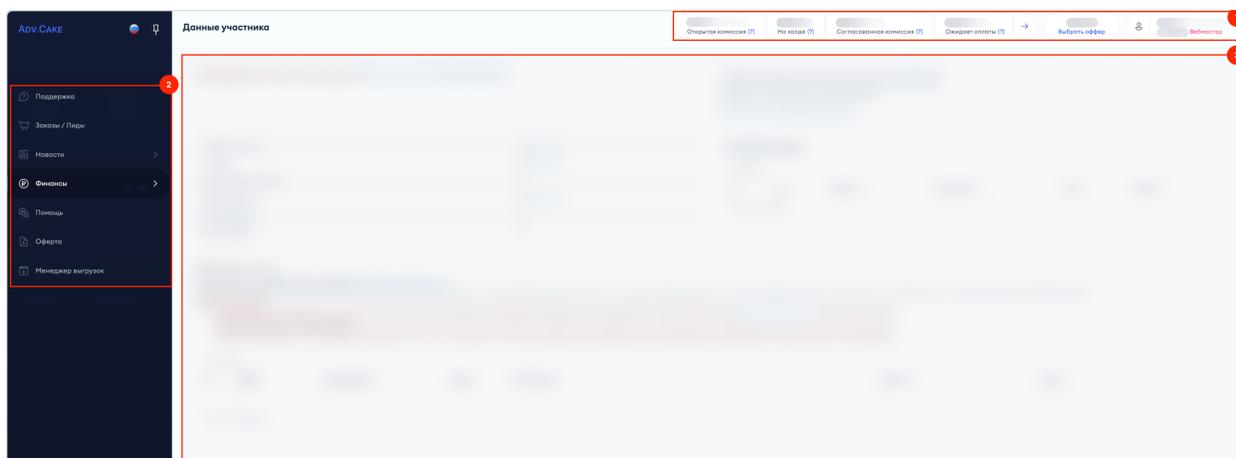
Например, это может быть приглашение преподавателя в школу учеников или рекомендация сотрудником своих знакомых для работы в компании и т.д.

2.2 Знакомство с интерфейсом личного кабинета Adv.Cake

Примечание: Перед тем, как приступить к работе с личным кабинетом, вы увидите всплывающее окно для заполнения данных физ. лица – заполните обязательные поля для продолжения работы и корректного проведения выплат.

Если у вас возникли вопросы, обращайтесь в нашу .

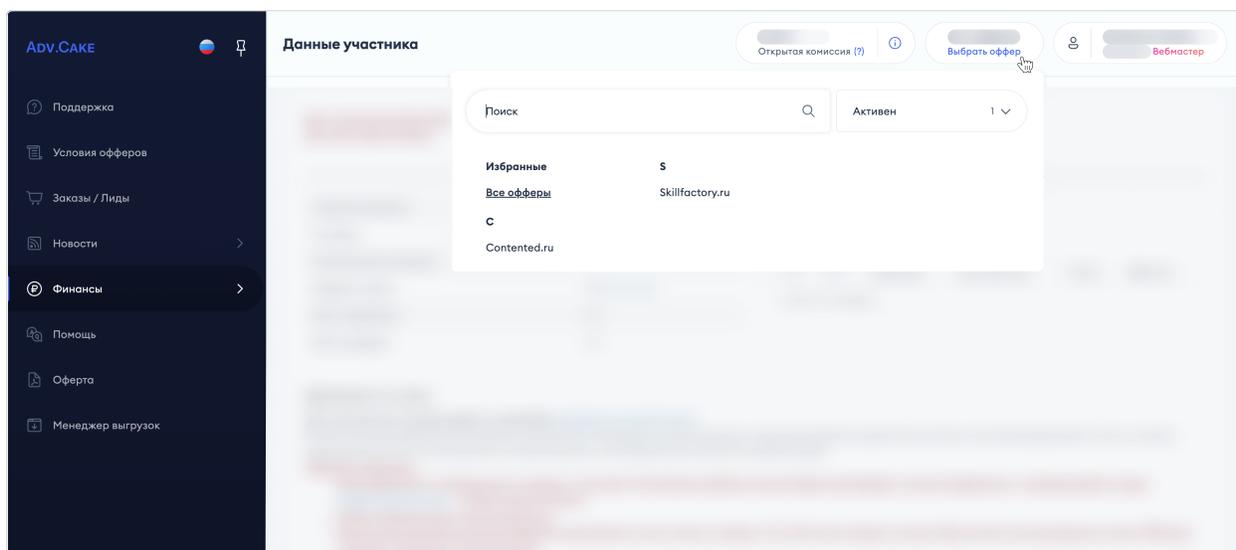
Ваш личный кабинет разделен на три части: шапка (1), главное меню (2) и рабочая область, где отображаются данные и возможные действия (3).



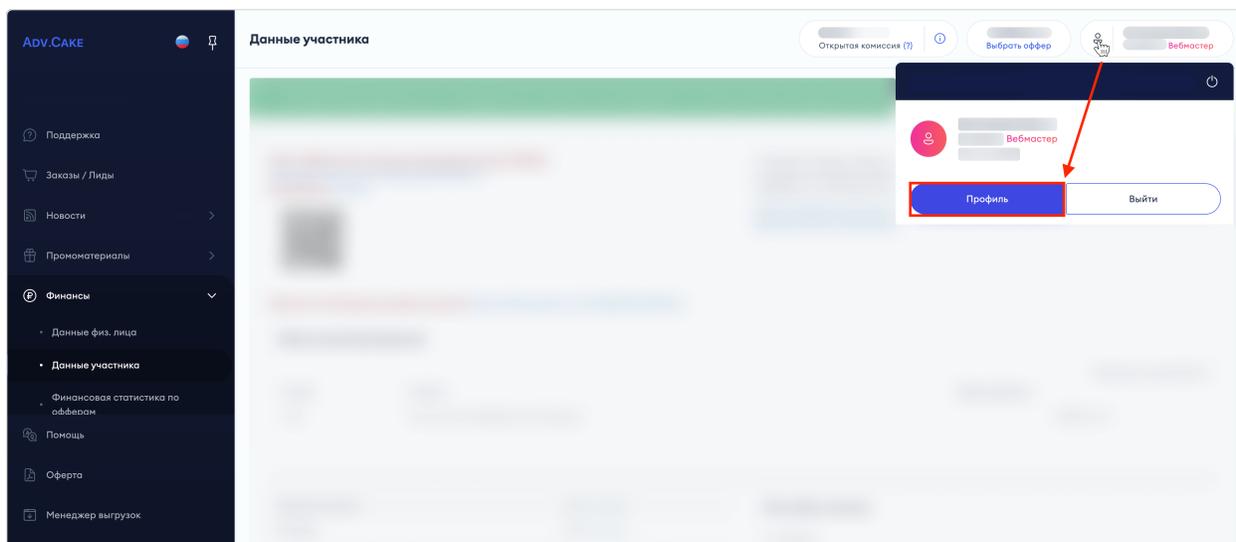
Шапка находится в верхней части личного кабинета и содержит информацию о вашем аккаунте, *финансах* и вашем текущем оффере.

Если вы работаете с несколькими офферами, нажмите на кнопку *Выбрать оффер* и введите в поле поиска название оффера, с которым вы планируете работать. Также, для удобства, все офферы разбиты по категориям и выводятся в алфавитном порядке:

Подсказка: Чтобы добавить оффер в избранное, наведите указатель на название оффера и нажмите кнопку «Звезда»



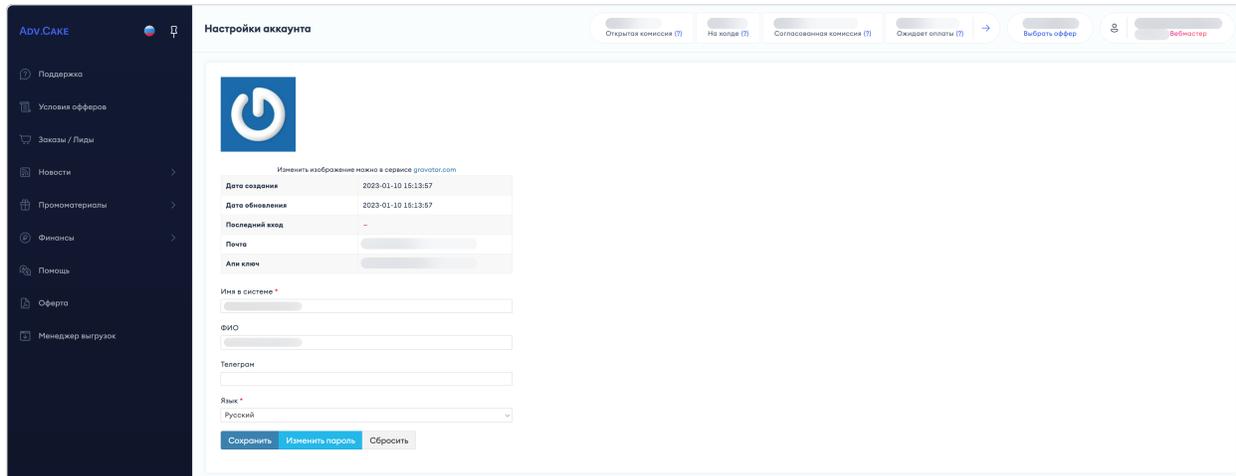
В настройки аккаунта можно попасть нажав на иконку вашего профиля и перейдя в раздел «Профиль»:



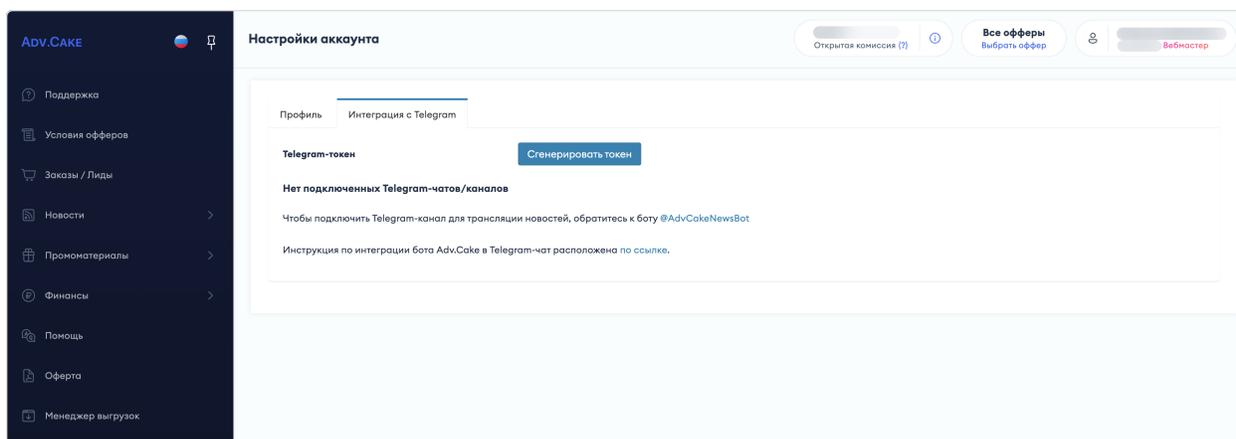
На табе «**Профиль**» представлена ваша электронная почта, дата регистрации в системе и API ключ. Здесь же, вы можете изменить ваш аватар, отображаемое имя в системе, Ф.И.О., логин в Telegram, а также язык личного кабинета.

После внесения изменений нажмите на кнопку *Сохранить*.

Чтобы изменить пароль, нажмите на кнопку *Изменить пароль* и следуйте инструкциям в появившемся окне.

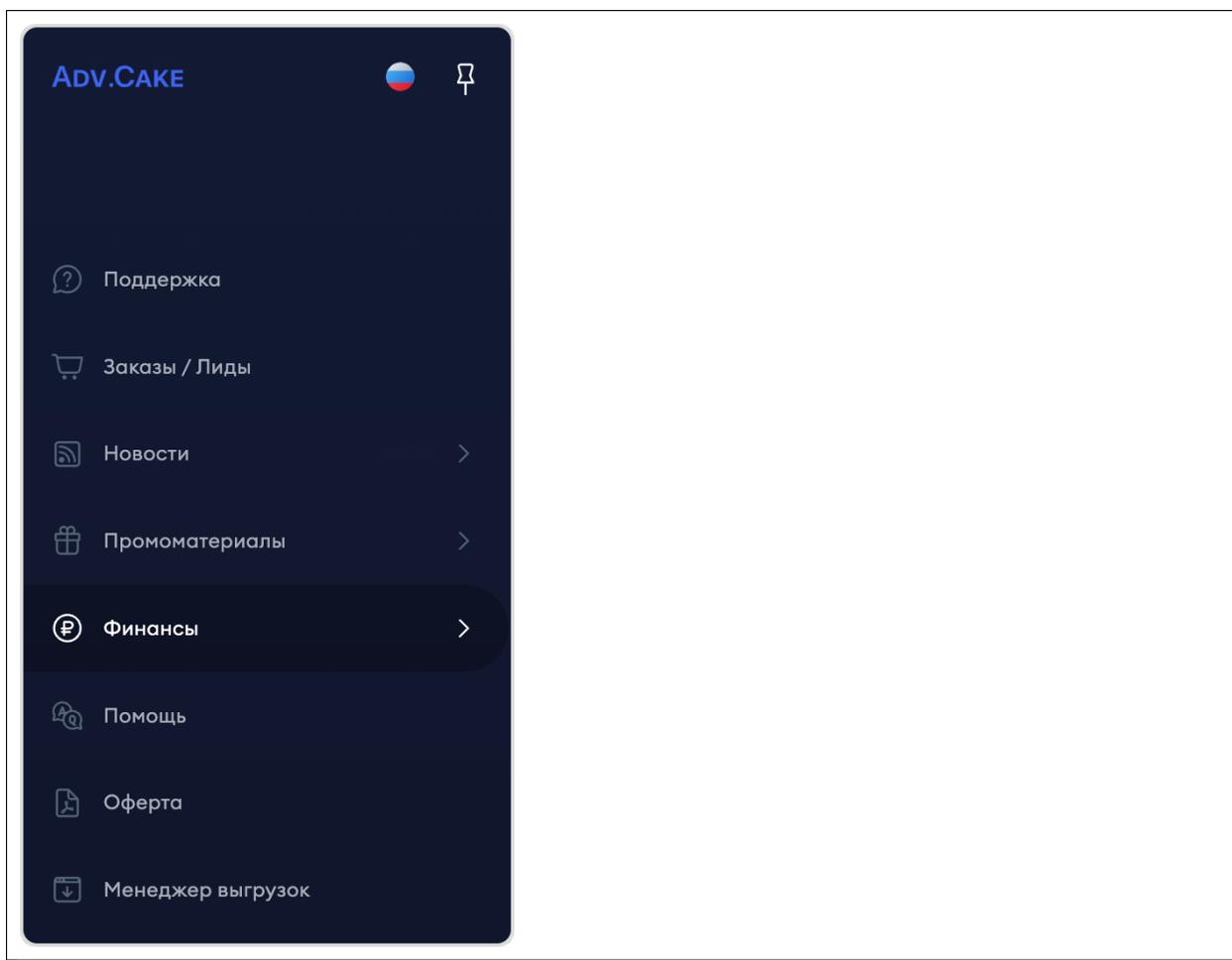


На табе «**Интеграция с Telegram**» находится инструмент по интеграции и работе с ботом Adv.Cake:



Если вы хотите автоматически получать актуальные новости об изменениях партнерской программы в ваш Telegram чат/канал, смотрите подробности в .

С помощью главного меню вы можете перейти в нужный раздел личного кабинета:



1. Поддержка;
2. Заказы/Лиды;

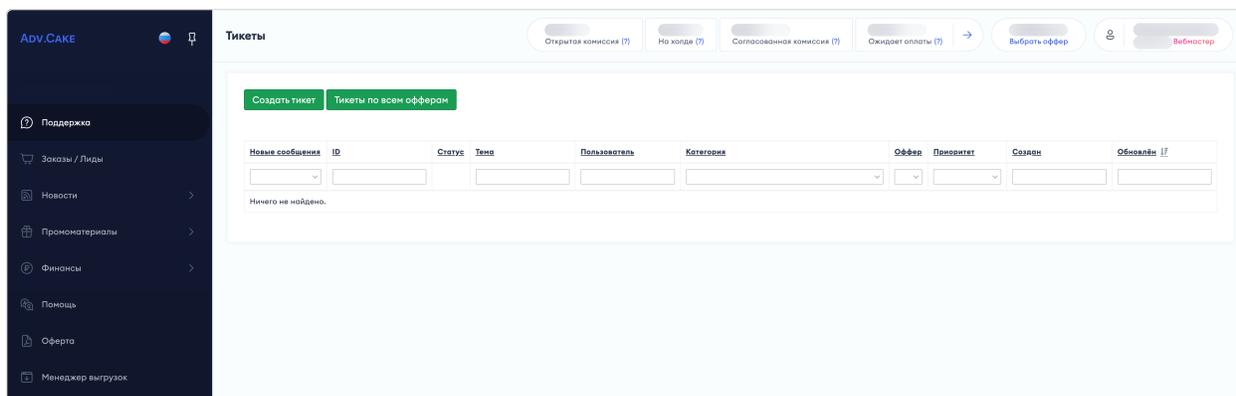
3. *Новости*;
4. *Промоматериалы*;
5. *Финансы*;
6. *Помощь*;
7. *Оферта*;
8. *Менеджер выгрузок*.

Подсказка: Чтобы закрепить меню в левой части экрана, нажмите на иконку с булавкой. Здесь же, можно выбрать, на каком языке будет отображаться ваш личный кабинет.

2.3 Поддержка

В этом разделе описывается взаимодействие участника реферальной программы с техподдержкой Adv.Cake через создание тикетов.

Каждому тикету присваивается ID, статус, тема и категория обращения, оффер, а также приоритет рассмотрения.

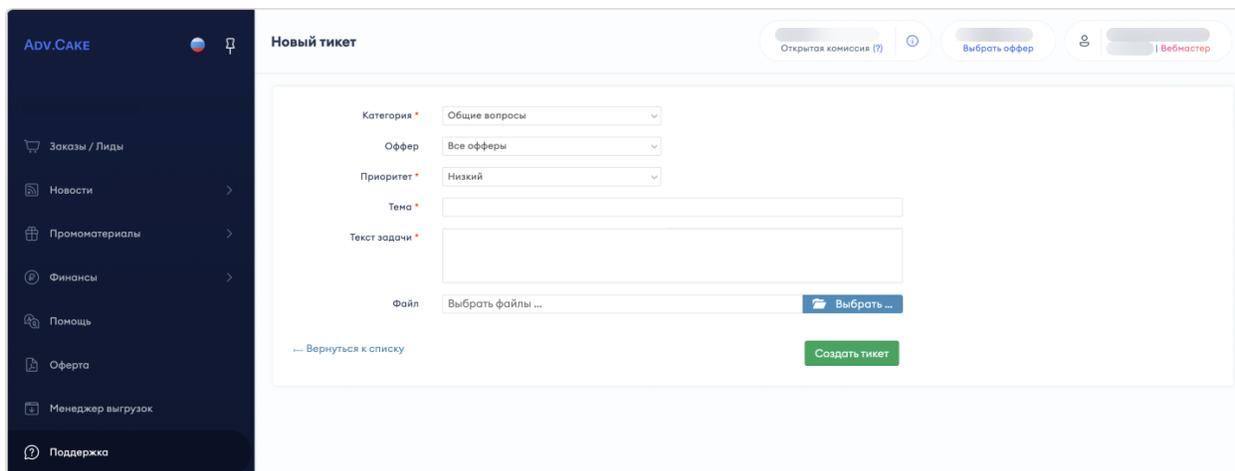


Чтобы создать новый тикет, нажмите *Создать тикет* и заполните следующие поля:

1. **Категория** – общие/технические/финансовые/другие вопросы;
2. **Оффер** – оффер, в рамках которого требуется помощь;
3. **Приоритет** – низкий/нормальный/высокий;
4. **Тема** – тема вопроса, вкратце отражающая его суть;
5. **Текст задачи** – подробное описание проблемы;
6. **Файл** – скриншот или иной файл подтверждающий проблему (желательно, но не обязательно к заполнению).

Примечание: Обратите внимание, максимальный размер прикрепляемого файла ограничен 100 МВ.

При этом формат файла может быть почти любым – csv, doc, docx, pdf, txt, xls, xlsx, bmp, ico, gif, jpg, jpeg, jfif, png, svg.

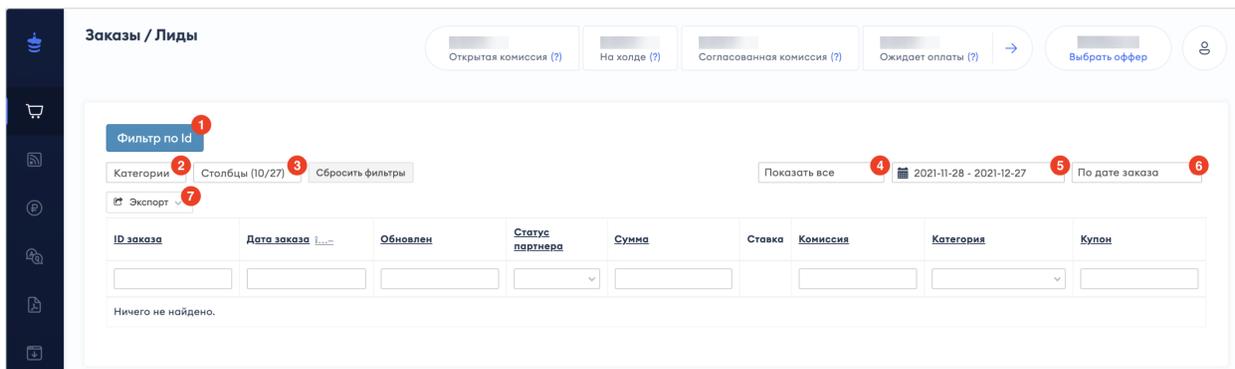


Подсказка: Прежде чем создавать тикет, рекомендуем обратиться к нашей документации – возможно, вы найдете ответ на ваш вопрос.

2.4 Заказы/Лиды

Раздел содержит подробную информацию о заказах/лидах, совершенных по вашей партнерской ссылке.

Для более точного отображения данных по заказам/лидам, воспользуйтесь панелью настроек:



1. **Фильтр по ID** — позволяет фильтровать заказы по ID;
2. **Категории** — позволяет выбрать категории для отображения данных;
3. **Столбцы** — позволяет выбрать столбцы для отображения данных;
4. **Фильтр по оплате** — позволяет фильтровать заказы по статусу оплаты;
5. **Фильтр по дате** — позволяет задать временной промежуток для отображения данных;
6. **Фильтр по дате заказа/обновления** — позволяет фильтровать по дате заказа/обновления;
7. **Экспорт** — позволяет экспортировать данные в формате CSV/Excel.

2.5 Новости

2.5.1 Все новости

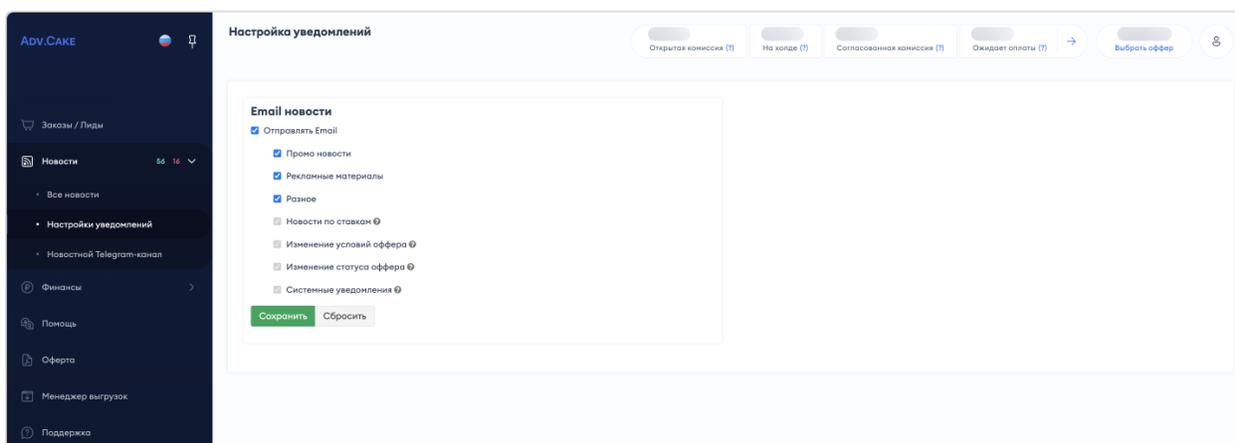
Раздел содержит новости от партнерских программ о новых акциях, изменении условий программ, добавлении рекламных материалов и др.

Новости выделяются тремя цветами:

1. Зеленый — новость не просмотрена;
2. Желтый — новость просмотрена;
3. Красный — важная новость.

2.5.2 Настройки уведомлений

Раздел содержит настройки уведомлений.



2.6 Промоматериалы

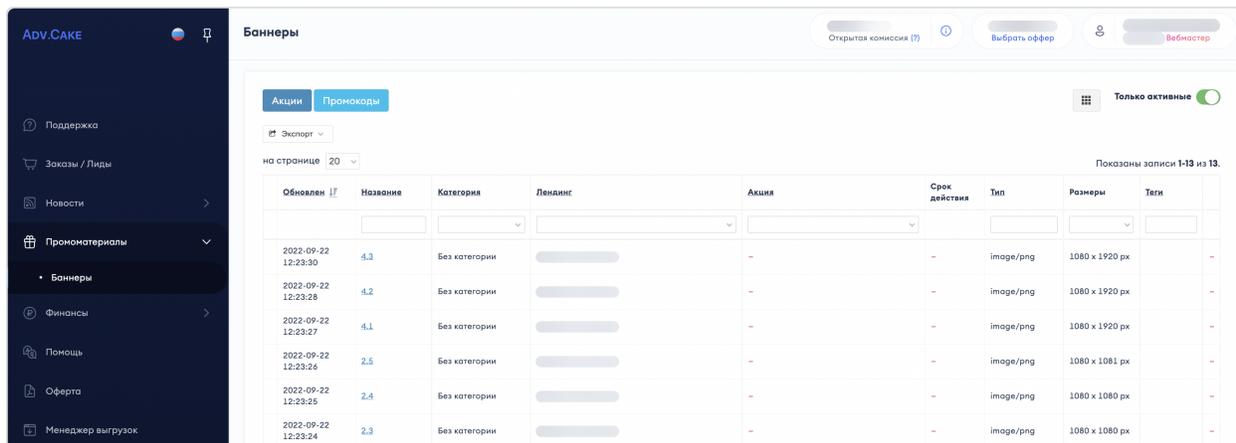
Предупреждение: Обратите внимание, данный раздел доступен не на всех офферах.

Раздел содержит доступные промоматериалы, которые рекламодатель предоставил в рамках работы с оффером.

2.6.1 Баннеры

Баннеры — изображения рекламного характера, которые содержат в себе ссылку на сайт рекламодателя.

Они предоставляются рекламодателем в рамках работы с оффером, а в этом разделе участник может скачать баннер для его дальнейшего размещения на своей рекламной площадке.



Чтобы посмотреть детали и дополнительные настройки баннера, нажмите на него в поле «Название»:

Обновлен	Название	Категория	Лендинг	Акция	Срок действия	Тип	Размеры	Теги
2022-09-22 12:23:30	4.3	Без категории		-	-	image/png	1080 x 1920 px	-

На вашем экране отобразится окно «Просмотр баннера», которое позволяет:

1. Посмотреть детали баннера: Id, категорию, связанный лендинг, MIME Type (тип изображения — jpeg/ png/ gif);
2. Задать адрес целевой страницы и дополнительные sub-параметры;
3. Скопировать код баннера для размещения на сайте. Чтобы скопировать баннер нажмите на кнопку *Скопировать в буфер обмена*, а чтобы он открывался в новом окне, поставьте галочку в поле «Открывать в новой вкладке».

Просмотр баннера ✕

Id	
Категория	—
Лендинг	
Mime Type	image/png

Адрес целевой страницы

http://site.com/path

Sub1	Sub2	Sub3
Sub4	Sub5	Keyword

Открывать в новой вкладке

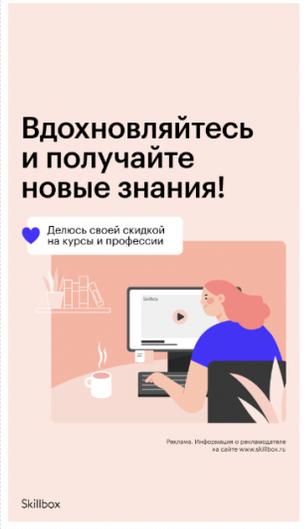
Результат

```

<!-- Advcake Banner -->
<div id="adb-addadc80569b3271-71646"></div>
<script>
window.advcake_runner = function(){
window.advcake_promo.forEach(function(t){t();});
window.advcake_promo = [];
window.advcake_promo.push = function(t){t();};

```

Скопировать в буфер обмена



1080x1920px

2.7 Финансы

2.7.1 Данные физ.лица

В этом разделе содержатся ваши персональные данные, которые нужны для проведения выплат.

2.7.2 Данные участника

Это основной раздел участника реферальной программы, который содержит материалы для работы, реферальную ссылку(-и), способы оплаты и подробную информацию о выплатах по совершенным действиям.

Ваша реферальная ссылка
Ваш QR-код: [Скачать](#)

1. Разместите вашу ссылку в соцсетях или просто отправьте другу.
2. Следите за заказом ваших друзей в разделе «Заказы/Лиды».
3. Добавьте способ выплаты вознаграждения.

Условия реферальной программы
[Пойдите на работу в реферальной программе](#)
[Правила участия в бонус-программе](#)

Ваше вознаграждение

Ссылка	Условия	Период действия	Показаны записи 1-4 из 4
	Тип источника: Реферальная программа		
	Для категорий: MBA, Заключение договора		< 2022-01-01
	Для категорий: MBA, Пролонгация адмиссии		< 2023-03-22
	Для категорий: MBA, Пролонгация презентации		< 2023-03-22
	Для категорий: Высшее образование, Поступление		< 2023-03-22
	Для категорий: Высшее образование, Подписание документов		< 2023-03-22

Способы оплаты

Тип	Счёт	Держатель	Срок действия	Статус	Действия
Банковская карта Visa/Mastercard/Мир (P)				✓	✕ P D

Движения по счету
Выплата вознаграждения (согласованная комиссия) производится автоматически в течение 10 рабочих дней после оплаты счета рекламодателем. Если на момент проведения выплаты отсутствует активный способ выплаты и ваш аккаунт не верифицирован в разделе «Данные физлица», она переводится еще на 10 рабочих дней.

Обратите внимание:

- При добавлении способа выплаты в валюту, отличной от Российские рубль, выплата производится в валюте USD с последующей конвертацией в валюту способа выплаты по курсу держателя. Конвертации из RUB в USD производят по курсу <https://www.ecb.eu/> + 7.15% на день выплаты.
- Может аннулироваться деп. комиссия до 30 у.д. за каждый период, в случае ее наличия у банка принимающей стороны.
- Выплата вознаграждения менее 3 000 руб. производится по вашему выбору через [заказовый](#). При выплате вознаграждения менее 3 000 руб. снимается комиссия до 100 руб.

1. **Ссылки** — в этом блоке содержатся ваша реферальная ссылка и прочая информация:

- Реферальная ссылка – ваш основной инструмент для работы. Размещайте ее на ваших рекламных площадках и следите за статистикой в разделе «Заказы/Лиды»;
- QR-код – дублирует вашу реферальную ссылку. Чтобы скачать QR-код в виде png-изображения, нажмите *Скачать*;

- Картинки и баннеры для ваших соцсетей – ссылка на облачное хранилище с медиа-файлами для размещения в ваших соцсетях (при наличии).

Примечание: Реферальная ссылка появится в вашем личном кабинете после регистрации в нашей системе.

Подсказка: На некоторых офферах есть определенные условия реферальной программы, узнать о которых, вы можете нажав на кнопку *Условия реферальной программы*.

2. **Вознаграждение** – в этом блоке перечислены ставки в фиксированном или процентом соотношении, которые применяются за определенные действия в рамках реферальной программы;
3. **Выплаты** – в этом блоке содержится информация о начислении вознаграждения. Эту же информацию вы можете видеть в шапке личного кабинета.

Подсказка: Столбец «Дата обновления» ориентирует вас по срокам выплаты – они зависят от даты добавления способа выплат.

2.7.2.1 Краткое описание статусов выплат

1. **Открытая комиссия** – комиссия всех заказов в статусе «Новый». При получении статуса «Одобен» переходит в раздел «На холде», при получении статуса «Отменен» – списывается. Открытая комиссия может изменяться не только за счет изменения статуса, но и за счет изменения самой комиссии. Т.е. пользователь приобрел товаров на 10 000 руб, после чего решил вернуть часть товаров – общая сумма заказа, как и общая комиссия, изменится, и если она считается в % от стоимости, тогда она тоже уменьшится;
2. **На холде** – комиссия заказов в статусе «Одобен», будет доступна к выводу после согласования и оплаты рекламодателем;
3. **Согласованная комиссия** – комиссия, доступная к выводу. Выплата создается автоматически в течение 10 дней после оплаты счета рекламодателем;
4. **Ожидает оплаты** – передано на оплату в сервис выплат. Среднее время проведения выплаты – 10 дней.
5. **Способы оплаты** – блок с доступными способами выплат для вывода вознаграждения. Чтобы добавить новый способ выплат нажмите кнопку *+ Добавить*.

Доступные способы вывода средств:

1. Банковская карта – для карт российских банков-эмитентов/Банковская карта Mastercard – для валютных карт;
2. Банковский перевод – иностранное физ.лицо (для счетов карт иностранных банков-эмитентов). Указывайте имя и фамилию латиницей, например Ivan Ivanov;
3. WebMoney/Payoneer – для получения выплат в долларовом эквиваленте.
5. **Движения по счету** – блок со сводной информацией по вашим финансам.

2.7.3 Финансовая статистика по офферам

В этом разделе содержится статистика по движению средств по всем офферам за все время или за выбранный период.

Финансовая статистика по офферам

Открыта комиссия (?) На заказе (?) Согласованная комиссия (?) Ожидает оплаты (?) → Выбрать оффер

По дате заказа 2016-01-01 - 2022-12-29

Оффер	Открыта комиссия	На заказе	Сейчас			Всего за выбранный период	
			Согласованная комиссия	Ожидает оплаты	Заработана	Выплачено	

2.7.4 Реестр начислений

В этом разделе содержится статистика по всем вашим начислениям.

Реестр начислений

Открыта комиссия (?) На заказе (?) Согласованная комиссия (?) Ожидает оплаты (?) → Выбрать оффер

Выплаты на сумму: [input type="text"]

ID начисления	Дата создания	Сумма	Способ вывода	Реквизиты	Офферы	Заказы	Дата выплаты

2.8 Помощь

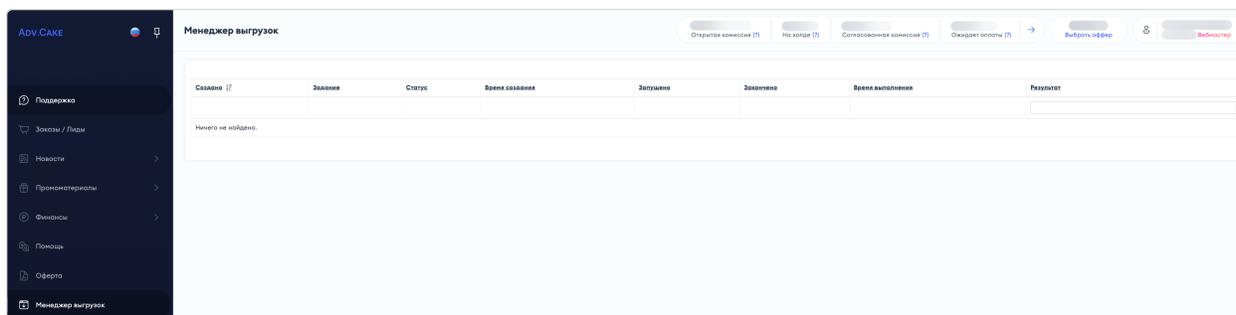
Этот раздел перенаправит вас на страницу поддержки, на которой вы сейчас находитесь.

2.9 Оферта

Раздел перенаправит вас на «Оферту о заключении договора оказания услуг».

2.10 Менеджер выгрузок

В этом разделе содержатся задания на выгрузку данных, которые вы запрашивали за все время работы с нашей платформой.



2.11 С чего начать?

2.11.1 Шаг 1: Работа с реферальными ссылками

1. Перейдите в раздел «Финансы» → «Данные участника»;
2. Скопируйте вашу реферальную ссылку (вы также можете воспользоваться QR-кодом, дублирующим вашу реферальную ссылку и готовыми креативами для соц.сетей при их наличии);
3. Разместите реферальную ссылку/QR-код/креатив на вашей рекламной площадке;
4. После того, как пользователь перейдет по вашей ссылке и совершит действие, оно загрузится в систему и отобразится в разделе «Заказы/Лиды».

2.11.2 Шаг 2: Получение вознаграждения

Вы можете отслеживать ваш баланс в шапке личного кабинета или в разделе *«Данные участника»* или в шапке личного кабинета.

Когда сумма появится в поле *«Согласованная комиссия»*, вы сможете вывести ее на свой счет или электронный кошелек.

Подсказка: Для возможности вывода средств не забудьте добавить ваши данные для проведения выплат в разделе *«Финансы»* → *«Данные физ.лица»*, а также способ выплат в разделе *«Финансы»* → *«Данные участника»*.

Примечание: Выплата вознаграждения (согласованная комиссия) производится автоматически в течение 10 рабочих дней после оплаты счета рекламодателем. Если на момент проведения выплаты нет доступного способа выплаты, она переносится еще на 10 рабочих дней.

3.1 Общая информация

Эта статья описывает основные принципы и преимущества работы Adv.Cake.

Статья состоит из следующих разделов:

1. *Зачем нужен Adv.Cake и как тут все работает?*;
2. *Отсев фрода*;
3. *Как привлечь вебмастеров?*

3.1.1 Зачем нужен Adv.Cake?

Как правило, рекламодатели сотрудничают с большим количеством вебмастеров, и работа с каждым из них отнимает довольно много времени, поэтому, чаще всего, рекламодатели пользуются услугами партнерских сетей.

Партнерская сеть — это платформа-посредник, которая помогает рекламодателями и вебмастерам находить друг друга, а также решает следующие организационные задачи:

1. Ведение статистики;
2. Техподдержка и консультирование вебмастеров и рекламодателей;
3. Осуществление выплат;
4. Трансляция рекламных материалов;
5. Решение спорных ситуаций;
6. Контроль за соблюдением правил;
7. Блокировка недобросовестных вебмастеров.

За решение этих и других задач партнерская сеть получает определенную комиссию от рекламодателей.

Adv.Cake, в свою очередь, автоматизирует работу с CPA-сетями.

Преимущества для рекламодателя:

1. Единая интеграция для всех партнеров;
2. Автоматическая сверка по статусам;
3. Точный трекинг действий;
4. Единая платформа для мониторинга статистики.

3.1.2 Схема работы рекламодателя в Adv.Cake

Порядок работы рекламодателя представлен на схеме.



3.1.3 Что такое фрод и как мы с ним боремся?

А вы знали, что рекламодателю не стоит платить за все заказы, оформленные через CPA?

CPA — модель оплаты интернет-рекламы с оплатой за действие, т.е. рекламодатель выплачивает вебмастеру вознаграждение только за совершенное действие пользователя.

Однако, даже в такой, казалось бы, беспроигрышной схеме, нашлись те, кто занимается фродом.

Фрод (от англ. fraud — «мошенничество») это когда мошенники пытаются минимальными силами создать видимость того, что их источник был последним в цепочке переходов пользователя до момента совершения заказа.

К основным методам фрода можно отнести:

1. ClickUnder на партнерский сайт с множественными куками;
2. ClickUnder на сайт рекламодателя;
3. Открытие сайтов во фрейме;
4. Тулбары;
5. Подмена кодов ретаргетинга;
6. Контекстная реклама на бренд.

Подробнее о каждом методе фрода и о том, как с ними бороться можно прочитать .

Примечание: Если у вас подключена только услуга Антифрод – разделы меню и доступные функции будут отличаться от тех, что доступны в стандартном личном кабинете рекламодателя.

Чтобы узнать больше, перейдите на страницу с описанием системы Антифрод.

3.1.4 Как привлечь вебмастеров?

У нас есть несколько советов по повышению лояльности ваших вебмастеров:

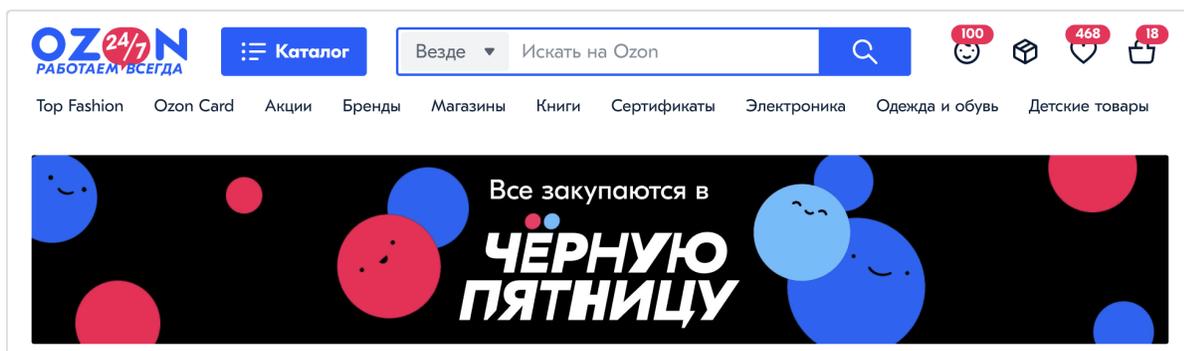
3.1.4.1 Персональные промокоды для вебмастера

Персональные промокоды позволяют вебмастеру делать более выгодные предложения для своей аудитории, тем самым повышая ее лояльность к своему сайту. Многие вебмастера любят пользоваться персональными промокодами, кроме того, это хороший повод начать с ними общение.

3.1.4.2 Работа с акциями

Когда на вашем сайте происходят акции, не забудьте рассказать об этом вебмастерам!

Например, вы участвуете в «Черной пятнице» — обновите ваши баннеры в CPA-сетях и расскажите об этом вебмастерам. Вы получите более высокие показатели, нежели вы разместитесь на официальной странице «Черной пятницы».

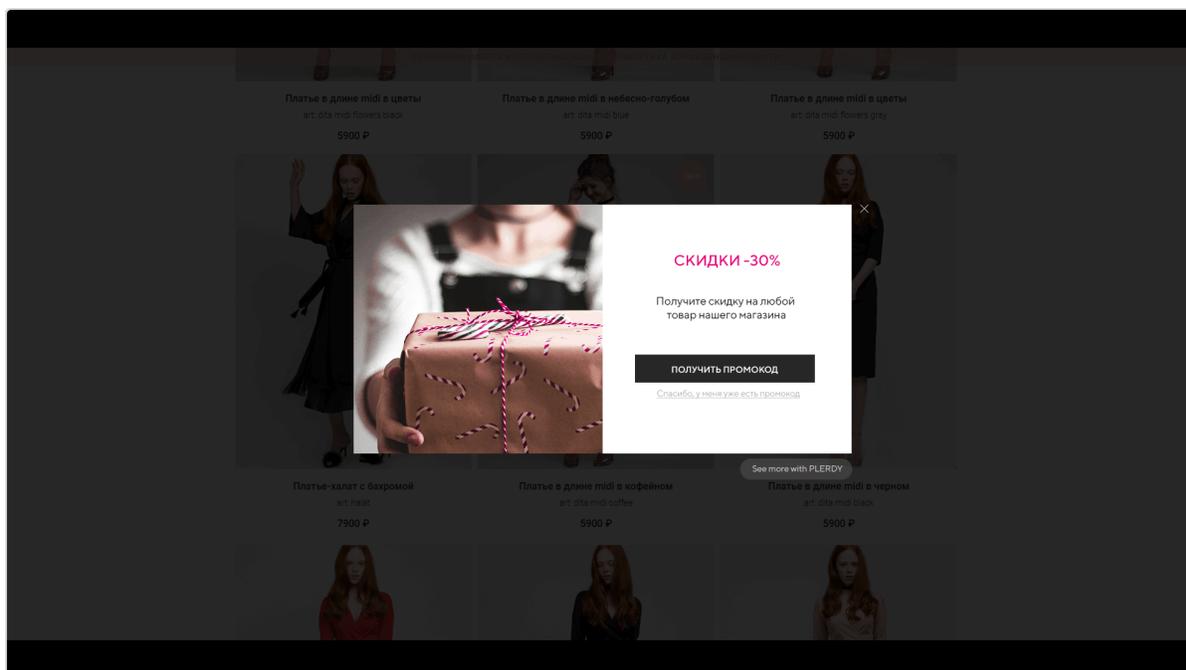


Помимо «Черной пятницы» вы можете уведомлять вебмастеров о ваших внутренних акциях — акции на выходных, сезонные акции или распродажа того, что у вас плохо продается.

О таких акциях лучше заранее сообщать вашим вебмастерам — они будут готовить для вас специальные баннеры, виджеты, статьи, что положительно скажется на их работе с вашим оффером.

3.1.4.3 Внутренние правила

Рассмотрим пример: вебмастер приводит трафик на ваш сайт, а вы, тем временем, разместили на нем Pop-Up баннер с предложением получить промокод на скидку.



Пользователь нажимает на Pop-Up, регистрируется и помечается метками, которые потом не передаются в CPA-сеть. Таким образом, вебмастер привел вам много трафика, а вы перепометили его своим источником — вебмастеру ничего не достанется.

Поэтому в CPA-сетях такие баннеры лучше не размещать, иначе вебмастеры не будут с вами работать.

3.1.4.4 Правила ретаргетинга

Если вы работаете с ретаргетингом — предоставляйте вебмастеру хороший трафик.

3.1.5 Советы по работе с вебмастерами

Главный совет - **учите вебмастеров работать с вашим оффером!**

Для примера возьмем три инструмента для взаимодействия с вебмастерами: баннер, виджет и статья.

Представим, что у нас есть сайт с travel тематикой и мы разместили на нем рекламные материалы с помощью этих инструментов.

Давайте взглянем на таблицу с показателями:

Инструмент	Кликабельность	Конверсия	Показы	Визиты	Заказы
Баннер	0,5%	0,5%	100,000	500	2-3
Виджет	5%	0,7%	100,000	5,000	35
Статья	30%	1%	100,000	30,000	300

Баннер: низкая информативность и, как следствие, плохая кликабельность и конверсия - из 100,000 показов лишь 1 из 200 пользователей кликнет на баннер, а вы получите 500 визитов и 2-3 заказа.

Товарный виджет: можно разместить на сайте вебмастера виджет, который будет вписываться в контент сайта. Разница между баннером и товарным виджетом - огромная. У виджета кликабельность и конверсия выше, чем у баннера - на 100,000 показов вы получите 5000 визитов и 35 заказов.

Пример товарного виджета:

Яндекс Маркет

Каталог

Искать товары

Найти

Санкт-Петербург

Новый год

Экспресс Одежда Дом Электроника Бытовая техника Детям Спорт Продукты

Журнал Маркета > Гаджеты

ОБЗОР

Что умеет приставка Apple TV и зачем она нужна

Последние пять лет я не смотрю телевизор. Всё, что мне интересно, нахожу в YouTube и других онлайн-сервисах по подписке прямо на ноутбуке. Конечно, экран маленький, но качество неплохое: мой старенький телевизор, кажется, обеспечивает то же Full HD, но на большей диагонали.

Скорее всего, мой телевизор продолжил бы собирать пыль, если бы на обзор ко мне не попала приставка Apple TV. Теперь я включаю его каждый день. Для тех, кто, как и я, ничего не знал о приставках и нечасто включал телевизор, расскажу, зачем вообще нужна Apple TV и как она работает.

СОДЕРЖАНИЕ

- 1 Даёт доступ к новым фильмам и сервисам
- 2 Быстро и просто подключается
- 3 Простая в управлении
- 4 Поддерживает подключение геймпадов от PS и Xbox
- 5 Чем мне понравился медиаплеер Apple TV
- 6 На что стоит обратить внимание

от 17 990 ₽

ТВ-приставка Apple TV 4K 64GB

4.7 79 отзывов Покупателям нравится звук, управление

Статья: если научить вебмастера писать статьи или составлять подборки о вашем интернет-магазине, то вы уже заранее подготовите пользователя к переходу на ваш сайт и побудите его к совершению покупки. У статьи огромная кликабельность и конверсия - на 100,000 показов вы получите 30,000 визитов и 300 заказов.

Примечание: Примеры выше применимы к одному вебмастеру и наглядно показывают разницу в выборе инструментов.

3.2 Обзор личного кабинета рекламодателя

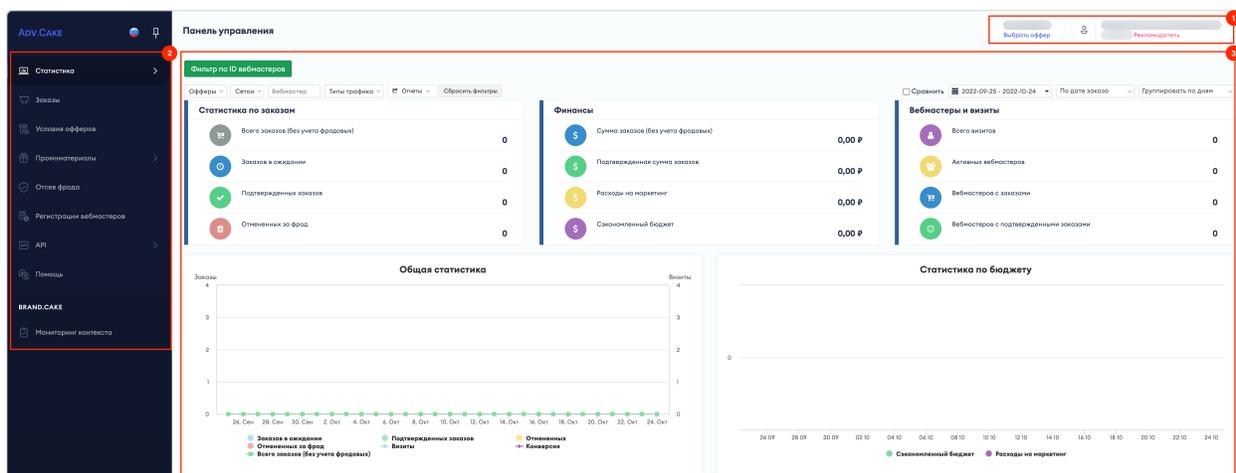
В этом разделе мы познакомим вас с интерфейсом личного кабинета, рассмотрим разделы главного меню и их основные функции.

Примечание: Если у вас подключена только услуга Антифрод – разделы меню и доступные функции будут отличаться от тех, что описаны ниже.

Чтобы узнать больше, перейдите на страницу с описанием системы Антифрод.

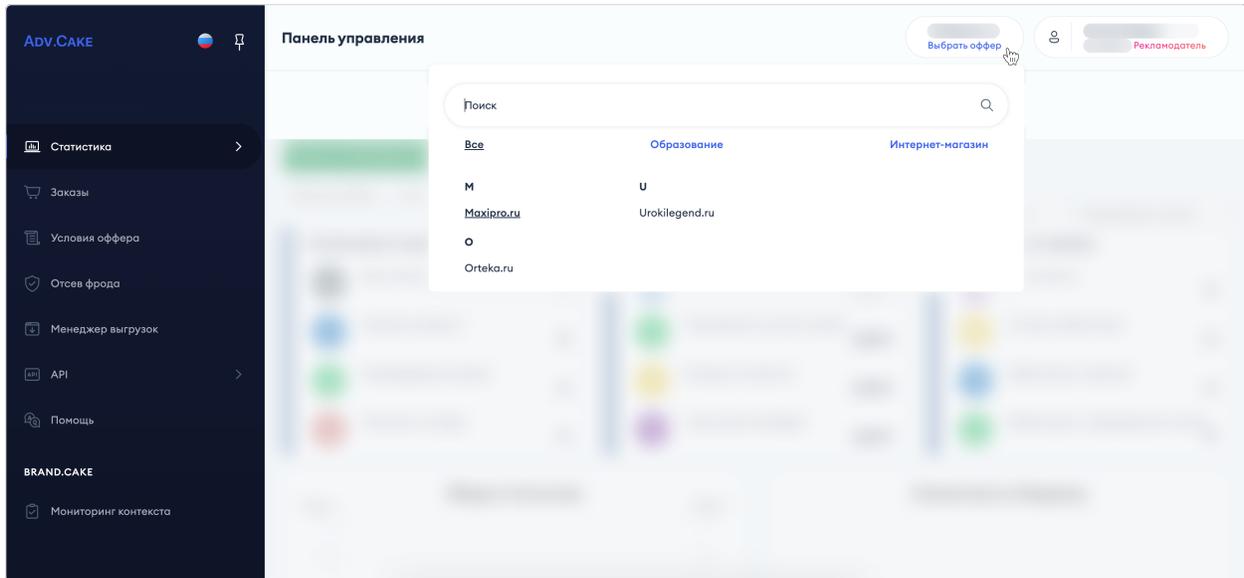
3.2.1 Знакомство с интерфейсом

Ваш личный кабинет разделен на три части: шапка (1), главное меню (2) и рабочая область, где отображаются данные и возможные действия (3).



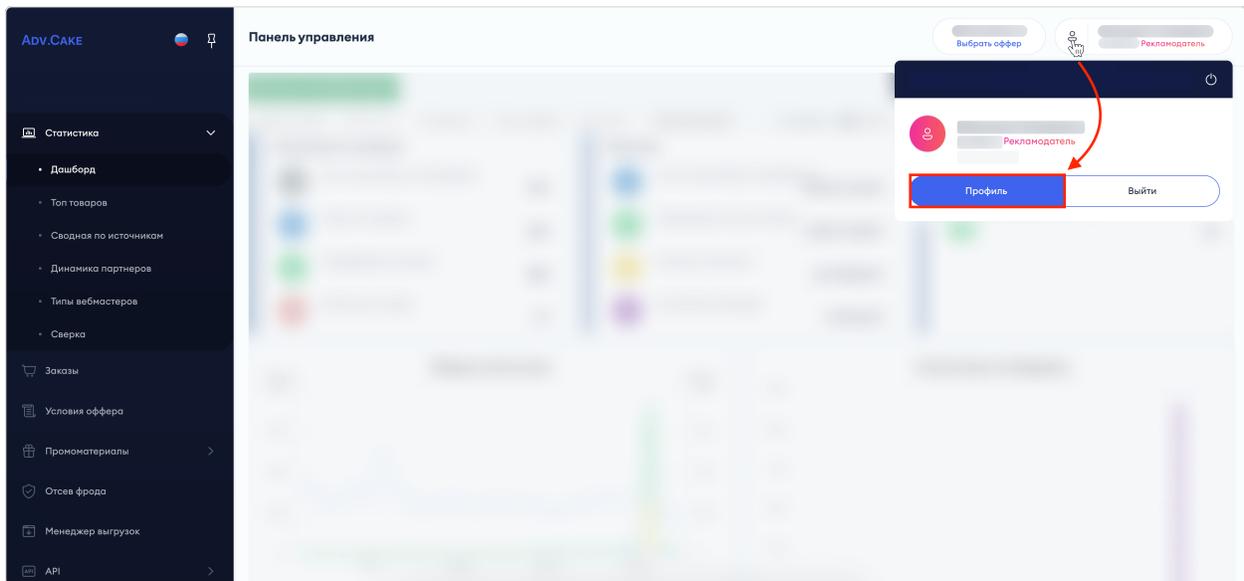
Шапка находится в верхней части личного кабинета и отображает доступные офферы и настройки профиля.

Если вы работаете с несколькими офферами, нажмите на кнопку *Выбрать оффер* и введите в поле поиска название оффера, с которым вы планируете работать. Также, для удобства, все офферы разбиты по категориям и выводятся в алфавитном порядке:



Подсказка: Чтобы добавить оффер в избранное, наведите указатель на название оффера и нажмите кнопку :guilabel:``

В настройки профиля можно попасть нажав на иконку вашего профиля и перейдя в раздел «Профиль»:

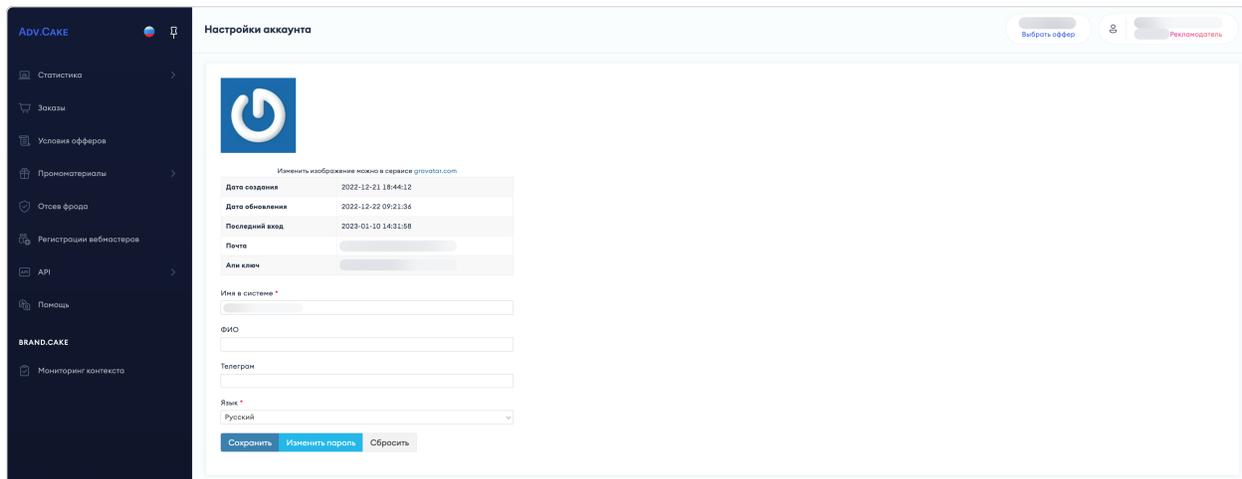


На этой странице представлена ваша электронная почта, дата регистрации в системе и API ключ.

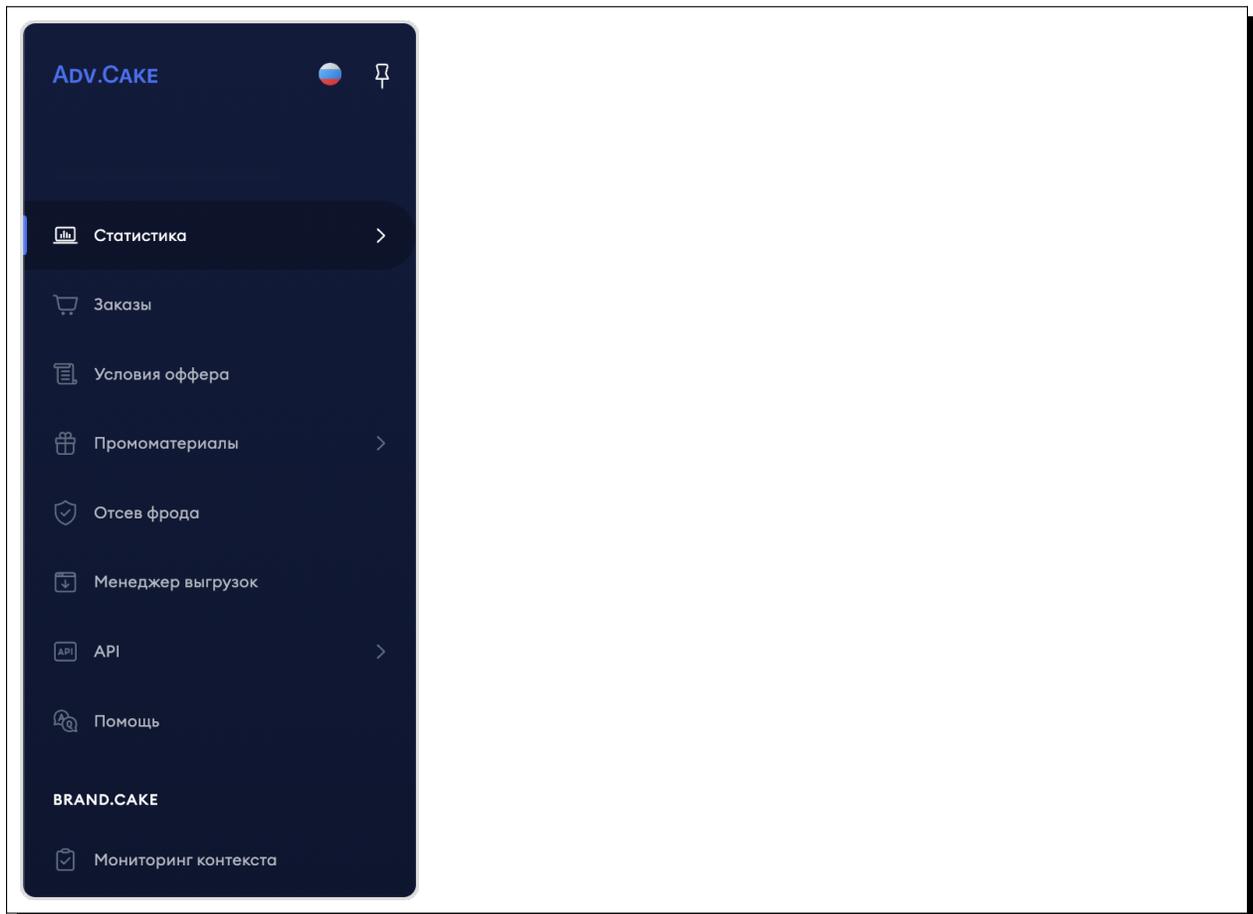
Здесь же, вы можете изменить ваш аватар, отображаемое имя в системе, вашу фамилию, имя и отчество, а также логин в Telegram.

После внесения изменений нажмите на кнопку *Сохранить*.

Чтобы изменить пароль, нажмите на кнопку *Изменить пароль* и следуйте инструкциям в появившемся окне.



С помощью главного меню вы можете перейти в нужный раздел личного кабинета:



1. *Статистика;*
2. *Заказы;*
3. *Условия оффера;*

4. Промоматериалы;
5. Отсев фрода;
6. Регистрации вебмастеров;
7. Менеджер выгрузок;
8. API;
9. Помощь;
10. Мониторинг контекста.

Примечание: Если в вашем меню не отображаются некоторые разделы — не беспокойтесь, они отличаются в зависимости от типа оффера.

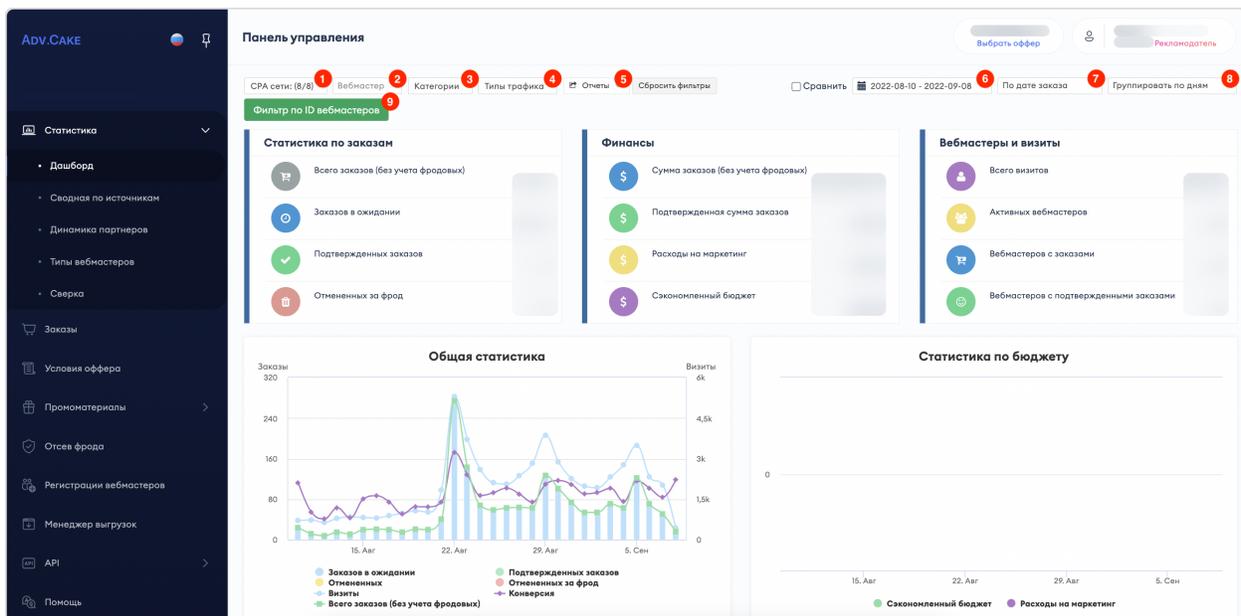
Подсказка: Чтобы закрепить меню в левой части экрана, нажмите на иконку с булавкой. Здесь же, можно выбрать, на каком языке будет отображаться ваш личный кабинет.

3.2.2 Статистика

3.2.2.1 Дашборд

Первое, что вы увидите, зайдя в личный кабинет – дашборд. Здесь представлена информация по всему CPA-каналу.

Для более точного отображения данных по каналу, воспользуйтесь панелью настроек:



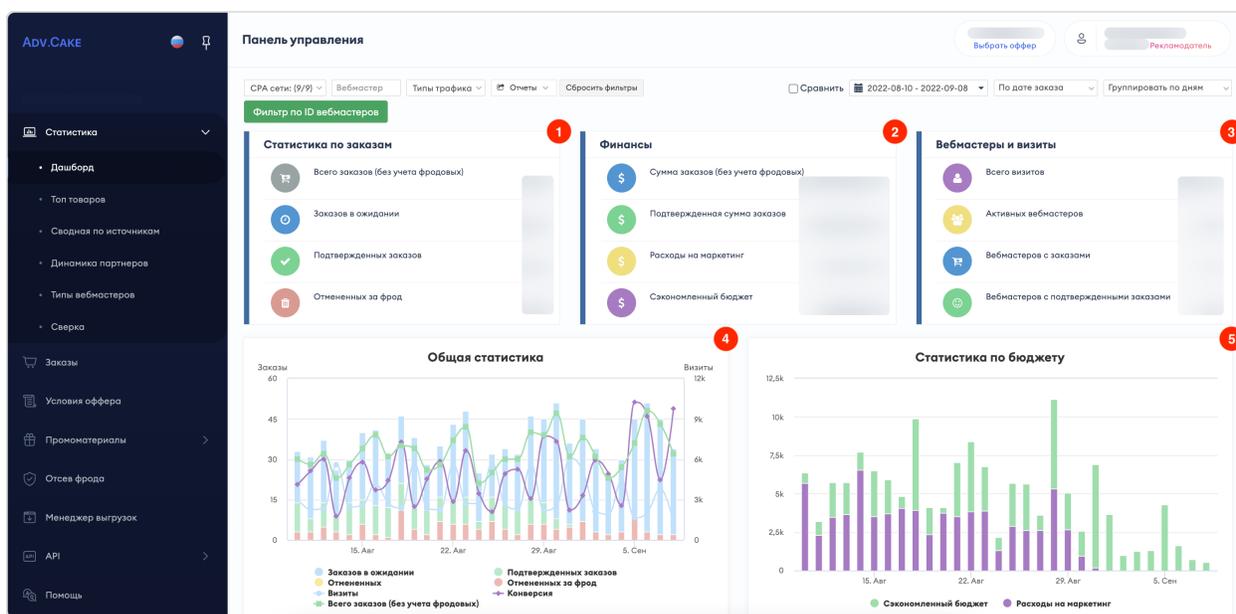
1. **CPA-сети** — позволяет фильтровать данные по CPA-сетям. В сеть *Affiliate* входят ваши прямые вебмастеры;
2. **Вебмастер** — позволяет просматривать статистику по ID вебмастера;

3. **Категории** – позволяет фильтровать данные по категориям;
4. **Типы трафика** – позволяет фильтровать данные по типу трафика вебмастеров;
5. **Отчеты** – отчет в стандартном виде. По запросу, может быть добавлена индивидуальная форма отчета;
6. **Фильтр по дате** – позволяет отслеживать статистику по дате создания/оформления заказа;
7. **Фильтр по дате обновления** – позволяет отображать статистику по вхождению заказа по дате оформления/обновления заказа;
8. **Фильтр по группировке заказов** – позволяет группировать заказы по дням, неделям или месяцам.
9. **Фильтр по ID вебмастеров** – позволяет отображать статистику по нескольким вебмастерам.

Также, вы можете выбрать период, за который хотите сравнить статистику – для этого нажмите на кнопку *Сравнить* и задайте любые даты для сравнения в появившемся календаре.

Примечание: В зависимости от типа оффера, фильтры на панели настроек могут отличаться от показанных выше.

Далее, рассмотрим несколько информационных блоков в дашборде:



3.2.2.1.1 Статистика по заказам (1)

1. **Всего заказов(без учета фродовых):** Общее количество заказов, совершенных вебмастерами за выбранный период без учета фродовых;
2. **Заказов в ожидании:** Заказы, которые за выбранный период не получили конечного статуса (*Одобрен/Отменен*);
3. **Подтвержденных заказов:** Заказы, которые за выбранный период получили статус *Одобрен*;
4. **Отмененных за фрод:** Заказы, которые за выбранный период были помечены как *фрод*.

Подсказка: Подробнее о фроде читайте в разделе

3.2.2.1.2 Финансы (2)

1. **Сумма заказов (без учета фродовых):** Общая сумма всех заказов, совершенных вебмастерами за выбранный период без учета фродовых;
2. **Подтвержденная сумма заказов:** Сумма заказов, получивших статус *Одобрен*;
3. **Расходы на маркетинг:** Комиссия, которую рекламодатель оплатит за *подтвержденные заказы* (со статусом *Одобрен*);
4. **Сэкономленный бюджет:** Комиссия, которую рекламодатель оплатит за заказы, помеченные как *фрод*.

3.2.2.1.3 Вебмастеры и визиты (3)

1. **Всего визитов:** Общее число переходов по партнерским ссылкам вебмастеров;
2. **Активных вебмастеров:** Количество вебмастеров, у которых есть хотя бы один переход по партнерским ссылкам;
3. **Вебмастеров с заказами:** Количество вебмастеров, у которых есть хотя бы один заказ, совершенный по партнерским ссылкам;
4. **Вебмастеров с подтвержденными заказами:** Количество вебмастеров, у которых есть хотя бы один *подтвержденный заказ*, совершенный по партнерским ссылкам;

3.2.2.1.4 Общая статистика (4)

График, в котором столбцами отображены дни, а линиями — статистика по заказам, финансам, вебмастерам и визитам.

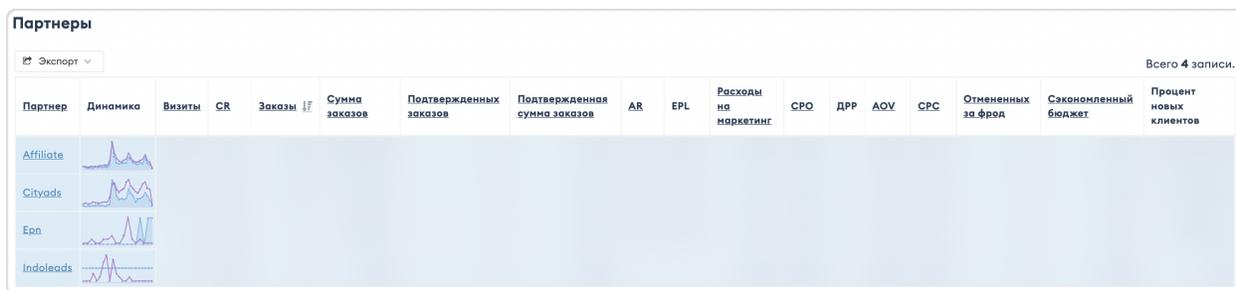
3.2.2.1.5 Статистика по бюджету (5)

График, в котором столбцами отображены дни, а линиями — сэкономленный бюджет и расходы на маркетинг.

Также, в нижней части страницы, вы можете просматривать статистику по партнерам и сводную по дате:

3.2.2.1.6 Партнеры

Позволяет просматривать статистику по каждому партнеру:



Для экспорта данных в формате CSV/Excel нажмите на кнопку *Экспорт*, а чтобы посмотреть визуализированную статистику по конкретному партнеру, нажмите на название партнера.

3.2.2.1.7 Сводная по дате

Позволяет просматривать статистику для каждого партнера по дате:

Дата	Партнер	Визиты	Заказы	Подтвержденных заказов	Отмененных	Отмененных за фрод	Сумма	Подтвержденная сумма заказов	Комиссия	Комиссия подтвержденных	AR	Сэкономленный бюджет	CPO	DRP	AOV
2022-09-08, Чт	-	500	16	0	0	0	5 596,00 P	0,00 P	1 161,26 P	0,00 P	0,0 %	0,00 P	-	-	349,75 P
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

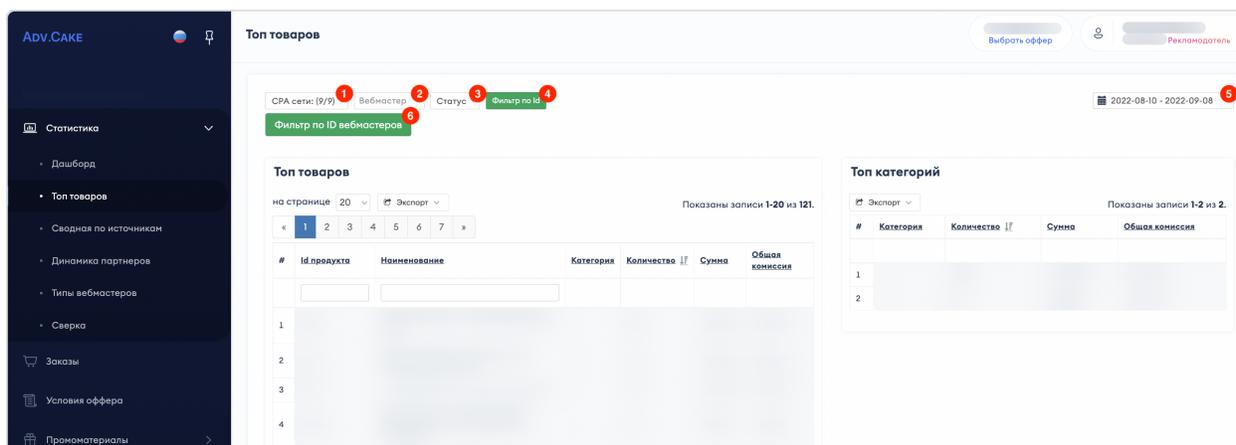
Для экспорта данных в формате CSV/Excel нажмите на кнопку *Экспорт*.

3.2.2.2 Топ товаров

Раздел содержит статистику по всем продаваемым продуктам в CPA-канале.

Предупреждение: Обратите внимание, данный раздел доступен не на всех офферах.

Для более точного отображения данных по каналу, воспользуйтесь панелью настроек:



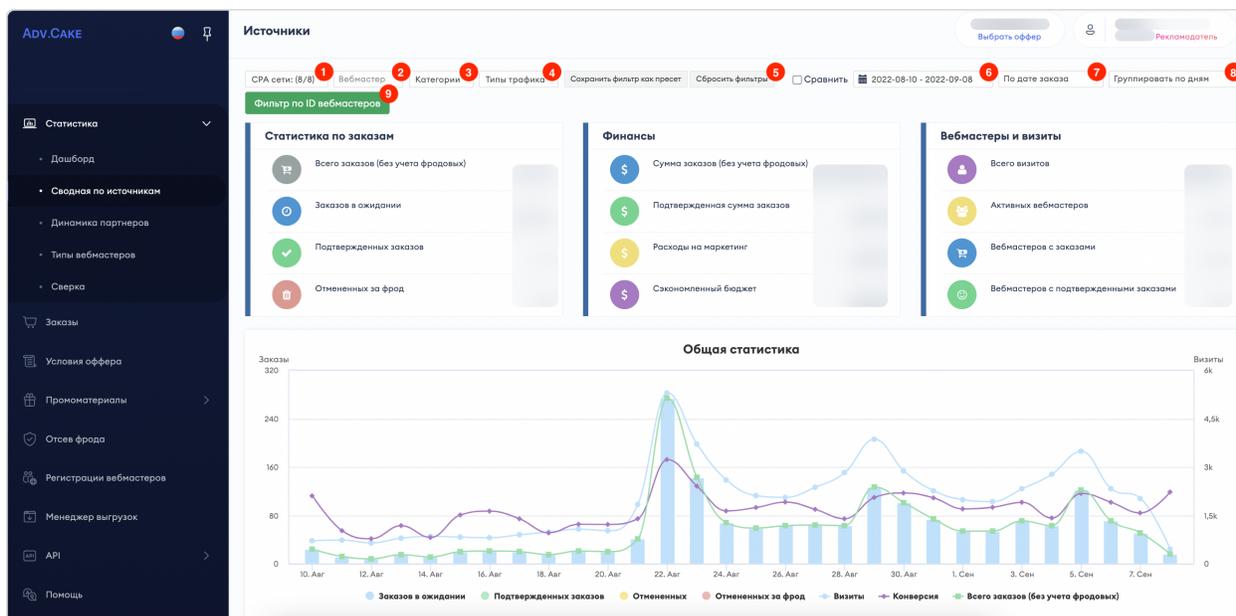
1. **СРА-сети** — позволяет фильтровать данные по СРА-сетям;
2. **Вебмастер** — позволяет просматривать статистику по ID вебмастера;
3. **Статус** — позволяет фильтровать данные по статусу;
4. **Фильтр по Id** — позволяет фильтровать данные по типу трафика вебмастеров;
5. **Фильтр по дате** — позволяет задать временной промежуток для отображения данных;
6. **Фильтр по ID вебмастеров** — позволяет отображать статистику по нескольким вебмастерам.

Примечание: Для выгрузки товаров или категорий в формате CSV/Excel нажмите на кнопку *Экспорт*.

3.2.2.3 Сводная по источникам

Раздел содержит сводную информацию по вебмастерам, работающим с партнерской программой.

Для более точного отображения данных, воспользуйтесь панелью настроек:



1. **СРА-сети** — позволяет фильтровать данные по СРА-сетям;
2. **Вебмастер** — позволяет просматривать статистику по ID вебмастера;
3. **Категории** — позволяет фильтровать данные по категориям;
4. **Типы трафика** — позволяет фильтровать данные по типу трафика вебмастеров;
5. **Сохранить фильтр как пресет/Сбросить фильтры** — позволяет сохранять/сбрасывать фильтры;
6. **Фильтр по дате** — позволяет задать временной промежуток для отображения данных;
7. **Фильтр по дате обновления** — позволяет отображать статистику по вхождению заказа по дате оформления/обновления заказа;
8. **Фильтр по группировке заказов** — позволяет группировать заказы по дням, неделям или месяцам.
9. **Фильтр по ID вебмастеров** — позволяет отображать статистику по нескольким вебмастерам.

Также, в нижней части страницы, вы можете просматривать статистику по партнерам (СРА-сетям) и вебмастерам:

3.2.2.3.1 Партнеры

Для экспорта данных в формате CSV/Excel нажмите на кнопку *Экспорт*, чтобы посмотреть визуализированную статистику по конкретному партнеру, нажмите на название партнера:

Партнеры

Экспорт

Всего 4 записи.

Партнер	Динамика	Визиты	CR	Заказы	Сумма заказов	Подтвержденных заказов	Подтвержденная сумма заказов	AR	EPL	Расходы на маркетинг	CPO	DRP	AOV	CPC	Отмененных за фрод	Сокращенный бюджет	Процент новых клиентов
Affiliate																	
Cityads																	
Epl																	
Indoleads																	

3.2.2.3.2 Вебмастеры

Для экспорта данных в формате CSV/Excel нажмите на кнопку *Экспорт*:

Вебмастеры

на странице 10 Экспорт

Показаны записи 1-10 из 36.

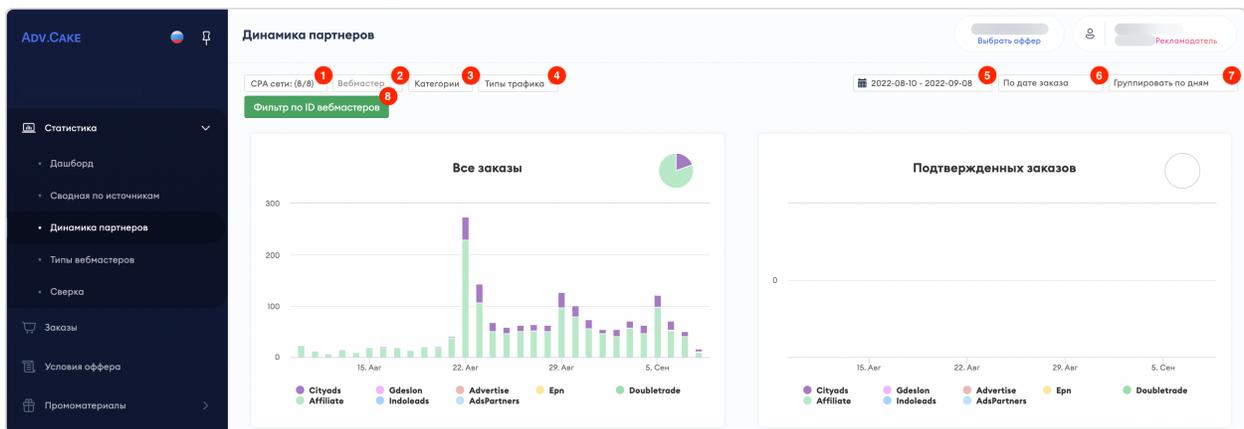
« 1 2 3 4 »

Партнер	Вебмастер	Начало сотрудничества	Начал в этом периоде	Динамика	Статус	Тип	Источник	Визиты	Заказы	Подтвержденных заказов	AR	EPL	CR	Сумма заказов	Общая комиссия

3.2.2.4 Динамика партнеров

Раздел содержит динамику работы партнеров с разбивкой на CPA-сети.

Для более точного отображения данных, воспользуйтесь панелью настроек:



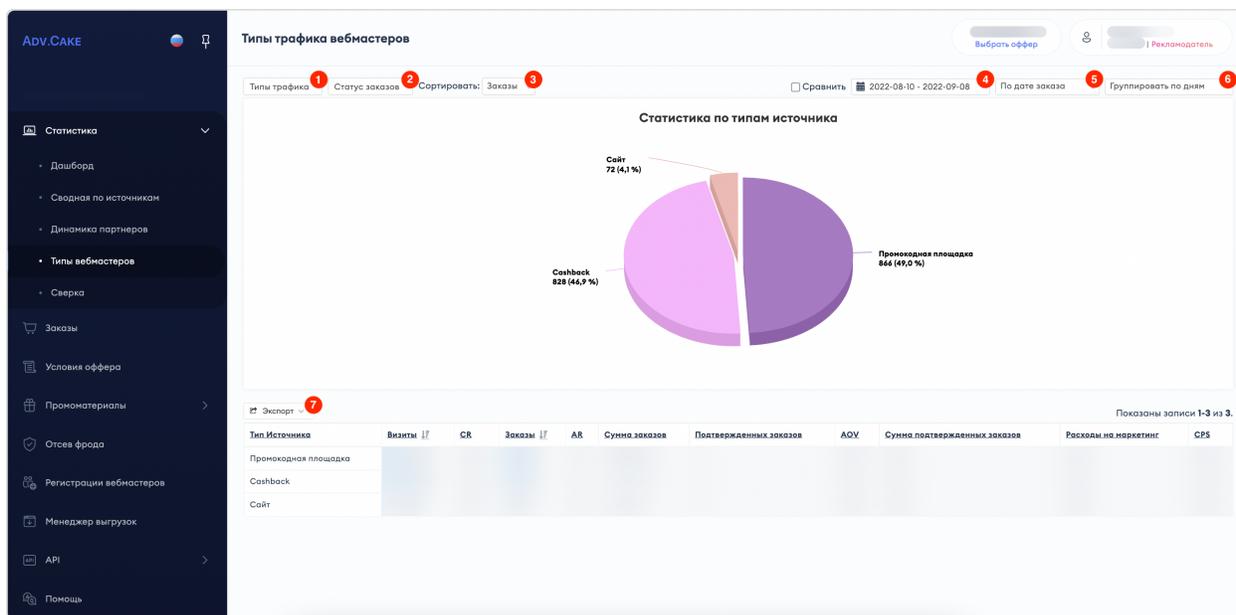
1. **CPA-сети** — позволяет фильтровать данные по CPA-сетям;
2. **Вебмастер** — позволяет просматривать статистику по ID вебмастера;
3. **Категории** – позволяет фильтровать данные по категориям;

4. **Типы трафика** — позволяет фильтровать данные по типу трафика вебмастеров;
5. **Фильтр по дате** — позволяет задать временной промежуток для отображения данных;
6. **Фильтр по дате обновления** — позволяет отображать статистику по вхождению заказа по дате оформления/обновления заказа;
7. **Фильтр по группировке заказов** — позволяет группировать заказы по дням, неделям или месяцам.
8. **Фильтр по ID вебмастеров** — позволяет отображать статистику по нескольким вебмастерам.

3.2.2.5 Типы вебмастеров

Раздел содержит визуальную статистику по типам трафика вебмастеров, а также таблицу с более детальной статистикой по типам трафика вебмастеров.

Для более точного отображения данных, воспользуйтесь панелью настроек:



1. **Тип трафика** — позволяет фильтровать данные по типам трафика вебмастеров;
2. **Статус заказов** — позволяет фильтровать данные по статусу заказов (Новый/Одобен/Отменен/Фрод);
3. **Сортировать** — позволяет сортировать данные по заказам, визитам или расходам;
4. **Фильтр по дате** — позволяет задать временной промежуток для отображения данных;
5. **Фильтр по дате обновления** — позволяет отображать статистику по вхождению заказа по дате оформления/обновления заказа;
6. **Группировать по дням** — позволяет группировать данные по дням/неделям/месяцам;
7. Кнопка *Экспорт* — позволяет выгрузить список типов трафика вебмастеров в формате CSV/Excel.

Подсказка: Если вам неизвестны определения каких-либо терминов, например «CR», наведите курсор, чтобы открыть подсказку.

Ознакомиться с полным списком CPA-терминов вы также можете в нашем глоссарии.

Примечание: Подробнее о типах трафика читайте на странице «Источники трафика».

3.2.2.6 Сверка

Раздел для получения информации по расходам.

3.2.3 Заказы

Раздел содержит полный список всех заказов CPA.

Примечание: В случае с финансовыми офферами данный раздел меню будет называться «Заказы/Лиды».

Для более точного отображения данных, воспользуйтесь панелью настроек:

1. **Фильтр по Id** — позволяет фильтровать заказы по ID;
2. **Фильтр по ID вебмастеров** — позволяет отображать статистику по нескольким вебмастерам;
3. **CPA-сети** — позволяет фильтровать данные по CPA-сетям;
4. **Вебмастер** — позволяет просматривать статистику по ID вебмастера;
5. **Категории** — позволяет фильтровать данные по категориям;
6. **Типы трафика** — позволяет фильтровать данные по типу трафика вебмастеров;
7. **Столбцы** — позволяет выбрать столбцы для отображения данных;

8. **Фильтр по дате** — позволяет задать временной промежуток для отображения данных;
9. **Фильтр по дате обновления** — позволяет отображать статистику по вхождению заказа по дате оформления/обновления заказа;
10. Кнопка *Экспорт* – позволяет выгрузить список заказов в формате CSV/Excel.

В таблице представлена детальная статистика по заказам/лидам.

Чтобы посмотреть корзину заказа нажмите на кнопку *Товар* в столбце «Корзина» (для офферов с корзиной):

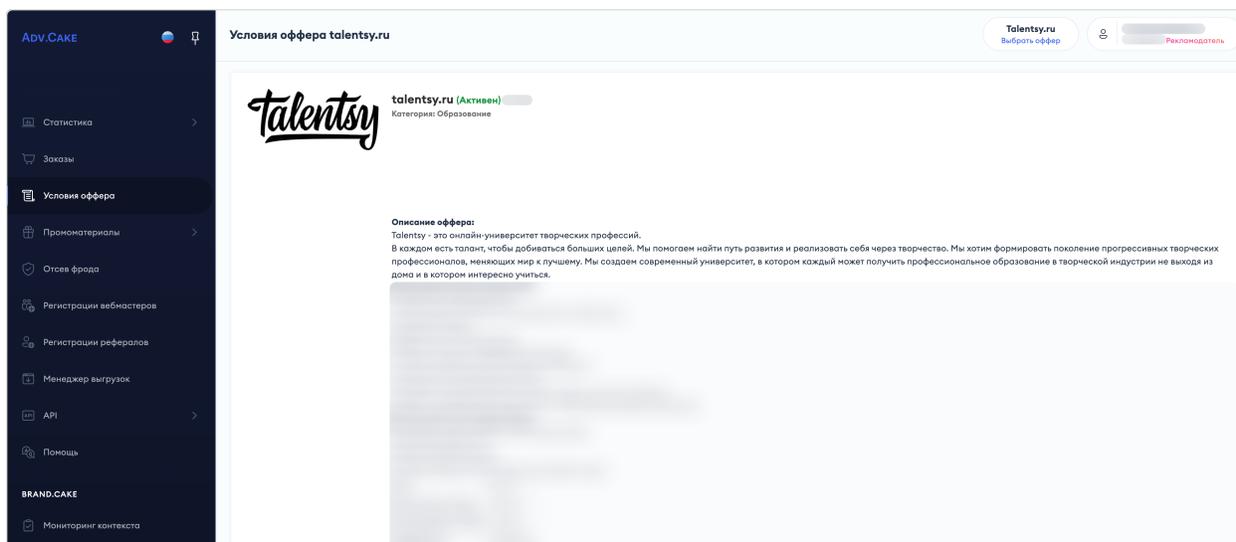
ID заказа	Дата заказа	Обновлен	Финальный статус	Партнер	Вебмастер	Сумма заказа	Общая комиссия	Общая ставка рекламодателя	Домен источника	Корзина	Тип клиента	Купон	Источник вебмастера
	2022-09-08 13:17:48	2022-09-08 13:20:49							-	1 товар	-		

Чтобы посмотреть историю переходов по заказу нажмите на кнопку  рядом с Id заказа (для лидовых офферов):

ID заказа	Дата заказа	Обновлен	Финальный статус	Партнер	Вебмастер	Сумма заказа	Общая комиссия	Общая ставка рекламодателя	Домен источника	Категория	Тип клиента	Купон	Источник вебмастера
										Без филт...			
	2022-09-08 11:18:13	2022-09-08 11:27:02						-	-	Оплаченная подписка	Новый		

3.2.4 Условия оффера

Раздел содержит краткое описание условий партнерской программы, ее преимуществ, информацию о целевой аудитории и разрешенных/запрещенных типах трафика.



The screenshot shows the 'Условия оффера talentsy.ru' page. On the left is a dark sidebar with navigation items like 'Статистика', 'Заказы', 'Условия оффера', etc. The main content area features the 'talentsy.ru' logo with a green 'Активен' status. Below the logo is the 'Описание оффера:' section, which contains text about Talentsy being an online university for creative professions. The text describes the university's mission to help creative professionals achieve their goals through education and practice.

3.2.5 Промоматериалы

Раздел содержит промоматериалы, которые рекламодатель предоставил менеджеру в рамках работы с оффером.

Предоставленные промоматериалы отображаются как у рекламодателя, так и у вебмастера, который работает с оффером рекламодателя.

3.2.5.1 Акции

В этом разделе содержатся акции вашего оффера.

Примечание: В системе Adv.Cake акции — родительский элемент, который содержит в себе промокоды, баннеры и лендинги.

Чтобы посмотреть детальную информацию по акции, а также привязку к лендингу, баннерам или промокодам, нажмите на кнопку «Просмотр» напротив акции:

Дата обновления	Имя	Название	Приватная	Срок действия	Тип трафика	Исключая типы трафика	Описание	Материалы
2022-09-06 11:10:07		Платья до 2 199	Нет	с 2022-09-06 до 2022-09-30	Реферальная программа	Да	Специальная подборка моделей	1 лендинг
2022-09-06 11:09:14		Футболки и Папы	Нет	с 2022-09-06 до 2022-09-30	Реферальная программа	Да	Мужская коллекция	1 лендинг
2022-09-06 11:08:20		Мужской трикотаж: sale	Нет	с 2022-09-06 до 2022-09-30	Реферальная программа	Да	Трикотаж со скидкой по выгодной цене	1 лендинг
2022-09-06 11:03:20		Куртки и жилеты. Ultra light	Нет	с 2022-09-06 до 2022-09-30	Реферальная программа	Да	Легкие куртки и жилеты Ultra light. Мужская и женская коллекции	1 лендинг
2022-09-06 10:58:09		Женский трикотаж: sale	Нет	с 2022-09-06 до 2022-09-30	Реферальная программа	Да	Трикотаж со скидкой по выгодной цене	1 лендинг
2022-09-06 10:54:44		Джинсы мужские до 3 999 Р	Нет	с 2022-09-06 до 2022-09-30	Реферальная программа	Да	Подборка мужских джинсов по стоимости до 3 999 рублей	1 лендинг
2022-09-06 10:54:14		Джинсы женские до 3 799 Р	Нет	с 2022-09-06 до 2022-09-30	Реферальная программа	Да	Подборка женских джинсов по стоимости до 3 799 рублей	1 лендинг
2022-09-06 10:52:34		До -80% на куртки, плащи и пуховики	Нет	с 2022-08-18 до 2022-09-30	-	Да	Скидки на верхнюю одежду Мужская и женская коллекции	1 лендинг
2022-09-06 10:51:12		Sale -80% Все товары по акции	Нет	с 2022-09-06 до 2022-09-30	Реферальная программа	Да	Sale -80% Женская и Мужская коллекции*Цены указаны с учетом скидок	1 лендинг

3.2.5.2 Промокоды

В этом разделе содержатся промокоды вашего оффера.

Чтобы посмотреть детальную информацию по промокоду, нажмите на его название в столбце «Код».

3.2.5.3 Баннеры

В этом разделе содержатся баннеры для вашего оффера.

Подсказка: Требования к формату баннеров вам предоставит ваш менеджер.

Чтобы посмотреть детали и дополнительные настройки баннера, нажмите на него в поле «Название»:

Просмотр баннера

Id	
Категория	-
Лендинг	finn-flare.ru – https://www.finn-flare.ru/ % P
Mime Type	image/jpeg



Партнер *	Affiliate	
Вебмастер *		
Адрес целевой страницы	http://site.com/path	
Sub1	Sub2	Sub3
Sub4	Sub5	Keyword
<input type="checkbox"/> Открывать в новой вкладке		

Результат

```
<!-- Advcake Banner -->
<div id="adb-fd52788fda88cb11-66387"></div>
<script>
window.advcake_runner = function(){
window.advcake_promo.forEach(function(t){t();});
window.advcake_promo = [];
window.advcake_promo.push = function(t){t();};

```

Скопировать в буфер обмена

Здесь можно:

1. Посмотреть детали баннера: его Id, категорию, связанный с ним лендинг, показатели CR/ECPC и MIME Type (тип изображения - jpeg/png/gif);
2. Задать партнера, вебмастера, адрес целевой страницы, а также дополнительные параметры (один Keyword и пять sub-параметров Sub1–Sub5) которые будут действовать для этого баннера. Если вы хотите, чтобы баннер открывался в новом окне, поставьте галочку в поле «Открывать в новой вкладке»;

3. Скопировать код баннера для размещения на сайте при помощи кнопки *Скопировать в буфер обмена*.

3.2.5.4 Лендинги

В этом разделе содержатся лендинги вашего оффера.

Чтобы посмотреть к какой акции привязан лендинг, нажмите на название акции в поле «Привязка к акции», а в поле сайт содержится ссылка на сам лендинг.

Подсказка: Информацию по промоматериалам можно выгрузить нажав на кнопку *Экспорт* в соответствующем разделе.

3.2.6 Отсев фрода

В этом разделе отображаются подозрительные и заблокированные вебмастеры. Если у вас подключена услуга антифрода, вы будете получать наши рекомендации по блокировке недобросовестных вебмастеров.

Далее, вы можете передать полученную информацию в вашу партнерскую сеть или самостоятельно принять решение о блокировке вебмастера и сообщить об этом нам.

3.2.6.1 Заблокированные

На этой вкладке отображена развернутая информация о заблокированных вебмастерах и заказах.

Переключатель (1) показывает заблокированных вебмастеров за все время, а календарь (2) позволяет задать любой диапазон дат для просмотра информации по заблокированным вебмастерам.

В окне (3) представлена краткая информация по заблокированным вебмастерам и заказам, а также сумма и комиссия по отмененным заказам.

Кнопка *Экспорт* (4) позволяет экспортировать данные в формате CSV/Excel, а поле (5) позволяет показывать вебмастеров, которые были активны/заблокированы за период.

Отсев фрода

Выбрать оффер | Рекламодатель

Заблокированные | Под подозрением | Фродовые заказы

За все время (1) | 2022-09-01 - 2022-09-08 (2)

Заблокировано вебмастеров	0 (3)
Заблокировано заказов	0
Сумма отмененных заказов	0 P
Комиссия отмененных заказов	0 P

Экспорт (4) | Показывать вебмастеров у которых: Активность за период (5)

Партнер	Вебмастер	Типы	Сайты	Причина блокировки	Доказательство	Визитов за период	Отмененных за фрод заказов	Сумма отмененных заказов	Комиссия отмененных заказов	Заблокирован

Ничего не найдено.

3.2.6.2 Под подозрением

На этой вкладке отображена информация о подозрительных вебмастерах, которые, например, могут использовать запрещенные Источники трафика.

3.2.6.3 Фродовые заказы

На этой вкладке отображена информация по фродовым заказам, с возможностью выбора периода дат для просмотра.

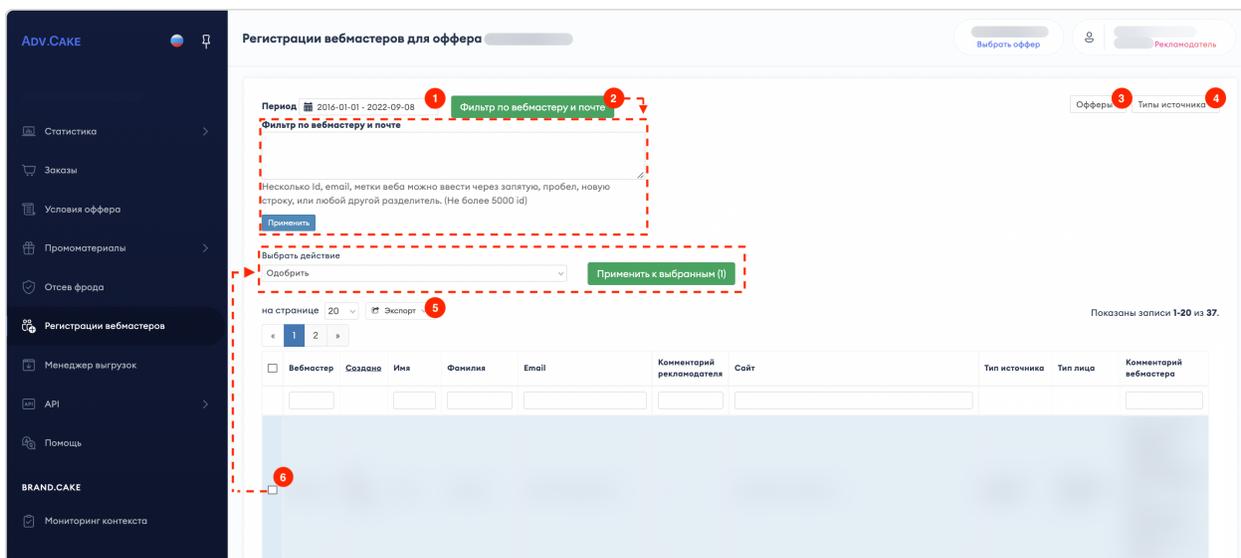
3.2.7 Регистрации вебмастеров

Примечание: Этот раздел доступен только для офферов с СПП (собственной партнерской программой).

Раздел содержит необработанные заявки регистрации вебмастеров для текущего оффера.

На странице доступны следующие настройки:

1. **Период** – позволяет задавать диапазон дат;
2. Кнопка *Фильтр по вебмастеру и почте* – открывает одноименное окно, в котором можно перечислить вебмастеров и почты, чтобы данные в таблице отфильтровались по заданным значениям;
3. **Офферы** – позволяет выбрать оффер, если их несколько;
4. **Типы источника** – позволяет отфильтровать таблицу по заданным типам источника, которые использует вебмастер;
5. Кнопка *Экспорт* – позволяет выгрузить список вебмастеров в формате CSV/Excel;
6. Кнопка `:guilabel:``` – позволяет выбрать вебмастера(-ов) и открывает окно действий, где можно одобрить, отклонить или добавить комментарий к заявке. Чтобы выбрать все заявки разом нажмите кнопку `:guilabel:``` в левом верхнем углу таблицы.



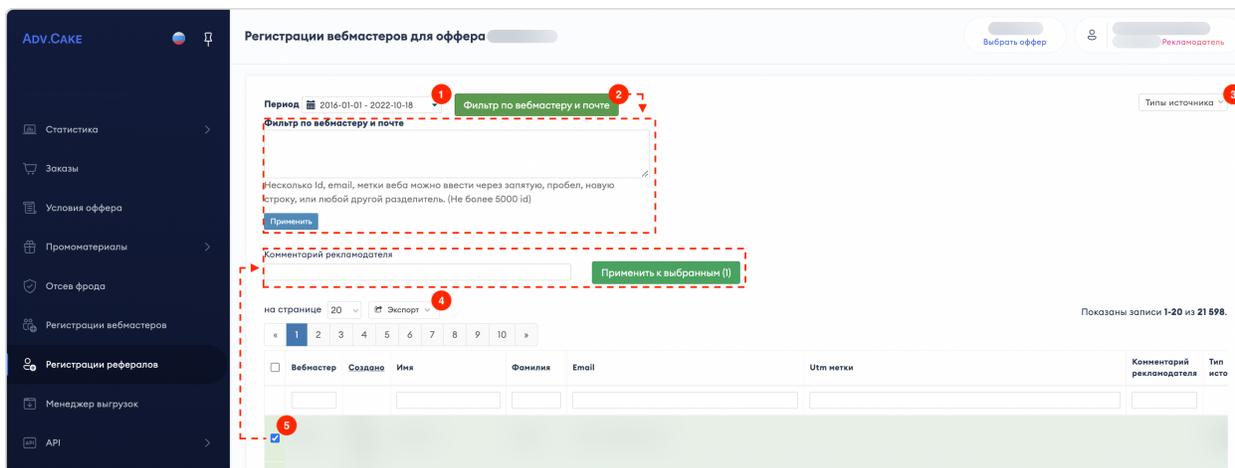
3.2.8 Регистрации рефералов

Примечание: Этот раздел доступен только для офферов с реферальной программой.

Раздел содержит необработанные заявки регистрации участников реферальной программы для текущего оффера.

На странице доступны следующие настройки:

1. **Период** – позволяет задавать диапазон дат;
2. Кнопка *Фильтр по вебмастеру и почте* – открывает одноименное окно, в котором можно перечислить вебмастеров и почты, чтобы данные в таблице отфильтровались по заданным значениям;
3. **Типы источника** – позволяет отфильтровать таблицу по заданным типам источника, которые использует участник реферальной программы;
4. Кнопка *Экспорт* – позволяет выгрузить список участников реферальной программы в формате CSV/Excel;
5. Кнопка «Отметить» – позволяет выбрать участника(-ов) реферальной программы и открывает окно действий, где можно добавить комментарий к заявке. Чтобы выбрать все заявки разом нажмите кнопку «Отметить» в левом верхнем углу таблицы.



3.2.8.1 Цвета заявок и что они означают

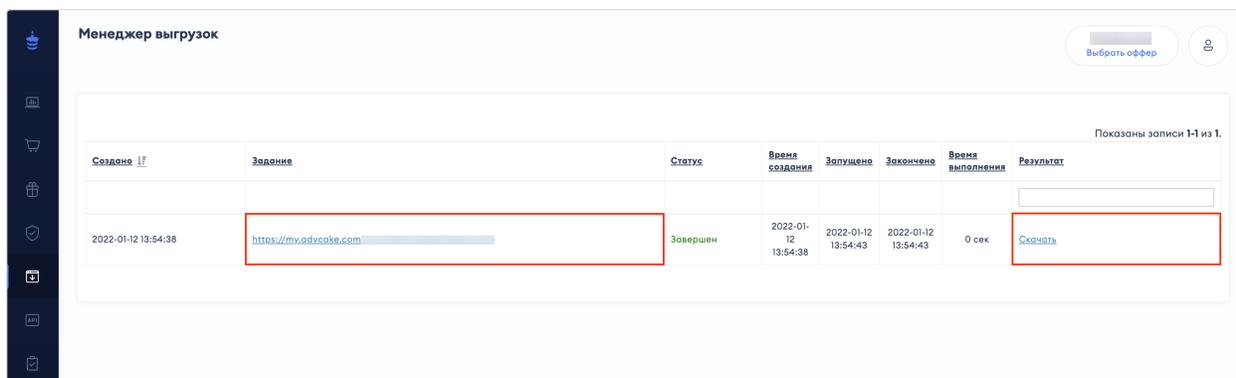
1. **Зеленый** — заявка одобрена, вебмастер может приступить к работе с программой;
2. **Желтый** — регистрация вебмастера не окончена, подождите пока вебмастер уточнит дополнительные данные;
3. **Синий** — заявка на модерации, подождите пока заявка поменяет свой статус;
4. **Красный** — заявка отклонена.

3.2.9 Менеджер выгрузок

В этом разделе содержатся задания на выгрузку данных, которые вы запрашивали за все время работы с нашей платформой.

Например, вы заходите в раздел «Заказы» и хотите выгрузить список всех заказов конкретной СРА-сети. Для этого, вы нажимаете на кнопку *Экспорт*, выбираете нужный формат выгрузки и видите поп-ап сообщение о созданном задании.

Теперь, задание появится в «Менеджере выгрузок» и отобразится в поле «Задание» и, когда оно будет готово, вы можете его скачать, нажав на соответствующую кнопку в поле «Результат»:



3.2.10 API

Рекламодателям доступно получение списков заказов и визитов в формате JSON.

Инструкции по использованию API, а также примеры запросов описаны в соответствующем разделе в вашем личном кабинете.

3.2.11 Помощь

Этот раздел перенаправит вас на страницу поддержки, на которой вы сейчас находитесь.

3.2.12 Мониторинг контекста (Brand.Cake)

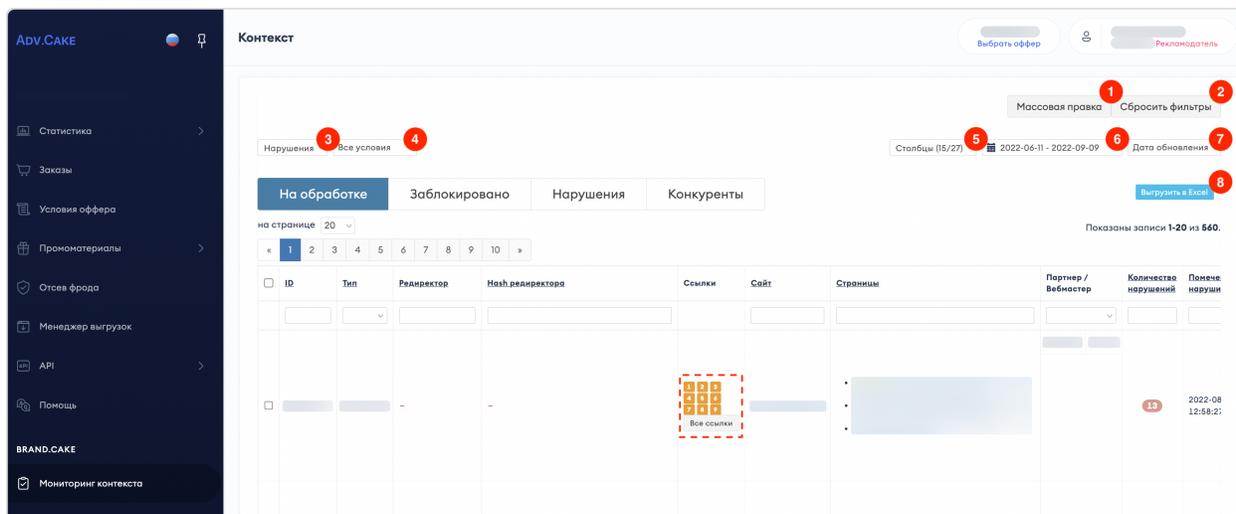
Раздел содержит систему мониторинга нарушений в контекстной рекламе. Подробнее о контекстной рекламе и ее особенностях читайте .

Для более точного отображения данных, воспользуйтесь панелью настроек:

1. **Массовая правка** – позволяет применить единоразовое (массовое) действие ко всем нарушениям. После выбора действия нажмите кнопку *Применить*;
2. **Сбросить фильтры** – позволяет сбросить примененные в массовых правках действия;
3. **Нарушения** — позволяет выбрать тип нарушения;
4. **Условие** — позволяет выбрать тип условия (все/любое);
5. **Столбцы** — позволяет выбрать столбцы для отображения данных;
6. **Фильтр по дате** — позволяет отслеживать статистику по дате оформления/создания заказа;

7. **Фильтр по дате обновления** — позволяет отображать статистику по вхождению заказа по дате обновления/создания заказа;
8. **Выгрузить в Excel** — позволяет выгрузить результаты мониторинга в формате Excel.

В столбце «Ссылки» (отмечено пунктиром) находится инструмент детального просмотра переходов по ссылкам.



3.3 Техническая интеграция с Adv.Cake

3.3.1 Общая информация

Для запуска программы необходимо провести интеграцию вашего сайта и базы данных с нашей системой.

Это нужно для того, чтобы мы могли получать данные о совершенных целевых действиях на вашем сайте и фиксировать их в нашей статистике.

В Adv.Cake доступны несколько способов интеграции:

1. Интеграция через Pixel Ecom, Pixel Lead, Pixel Lead без установки JS-кодов;
2. Интеграция через XML Ecom, XML Lead;
3. Интеграция при помощи Postback Ecom, Postback Lead;
4. Интеграция финансовых офферов через Pixel Lead Finance; Postback Lead Finance; XML Lead Finance;
5. Интеграция через конструктор сайтов Tilda Ecom, Tilda Lead;
6. Интеграция через конструктор сайтов Webflow Ecom, Webflow Lead;
7. Интеграция мобильных приложений через AppsFlyer (постбеки), AppsFlyer (инсталлы);
8. Интеграция мобильных приложений через AppMetrica (постбеки), AppMetrica (инсталлы);
9. Интеграция антифрода (системы мониторинга и предотвращения мошеннических действий).

Примечание: Почти все способы интеграции предназначены для разных типов офферов.

В случае с веб-версией сайта, интеграция зависит от типа оффера и делится на Ecom и Lead, а в случае с мобильными приложениями, проводится по постбекам и установкам (инсталлам).

3.3.2 Что такое Ecom и Lead ?

3.3.2.1 Ecom

На офферах этого типа обязательно присутствует корзина, т.е. на сайте есть возможность оплатить один или несколько выбранных товаров одним платежом.

Пример корзины на сайте «М.Видео»:

The screenshot displays a shopping cart interface. It features two product entries:

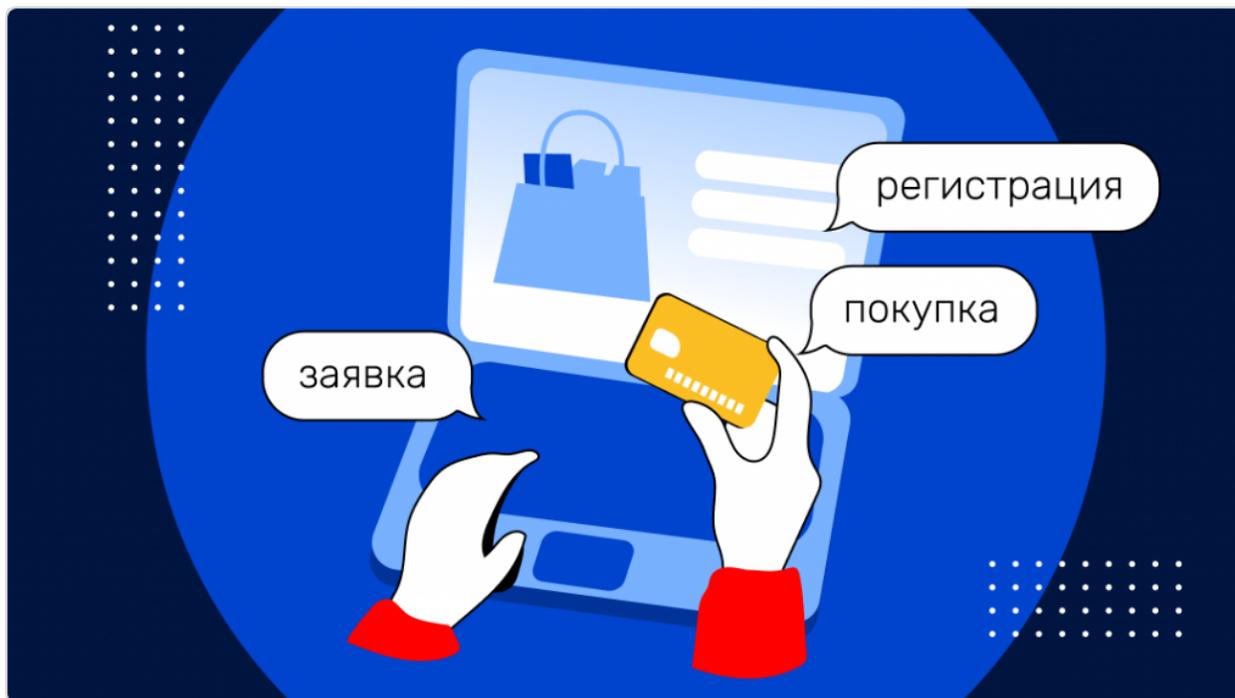
- Smartphone:** vivo Y31 4+64GB Чёрный асфальт (V2036). Price: 14,999 (crossed out), 13,999 RUB. Quantity: 1. Includes options to 'Сохранить на потом' and 'Удалить'. A checkbox for a charity service is also present.
- Laptop:** Honor MagicBook X 15 i5/8/512 Gray (BVR-WAN9). Price: 59,999 (crossed out), 54,999 RUB. Quantity: 1. Includes options to 'Сохранить на потом' and 'Удалить'.

Summary and actions:

- Total items: 2. Total price: 74,998 (crossed out), 68,998 RUB.
- Buttons: 'Перейти к оформлению' (Proceed to checkout).
- Link: 'Авторизуйтесь/зарегистрируйтесь, чтобы получить 3% от стоимости заказа Бонусными рублями'.

3.3.2.2 Lead

На офферах этого типа нет такого понятия как «корзина», вместо нее – услуга или заказ, например, оформление кредита или страховки, покупка курса и т.п.



3.3.3 Об интеграции мобильных приложений

Мы умеем интегрировать не только веб-версии сайтов, но и мобильные приложения.

Интеграция мобильных приложений в нашу систему делится на несколько типов, которые отличаются способом интеграции, настройкой офферов и вебмастерами, которые с ними работают.

С технической стороны есть два варианта провести интеграцию мобильного приложения:

1. Подключить дополнительную атрибуцию трафика в мобильном приложении к текущему офферу (подходит для поддержания спроса и работы с лояльной аудиторией);
2. Создать новый оффер, заточенный непосредственно под работу с приложением за установки (инсталлы) (подходит для работы с новой аудиторией для раскрутки непосредственно приложения).

Примечание: Также, возможен вариант одновременной интеграции по классической модели (Last Paid Click) и установкам – в таком случае, в приложении к договору, будут указаны разные ставки для разных типов интеграции.

3.3.3.1 Первый вариант интеграции

В первом варианте интеграции, мы по-прежнему работаем с основным сайтом клиента, при этом, если на сайте настроен редирект на приложение, мы не будем терять заказы – это тот же обычный оффер, только с дополнением к интеграции в виде мобильного приложения.

Соответственно, нам не важно, откуда пришел заказ – с веб/мобильного сайта или приложения, мы также смотрим каждый заказ, определяем последний клик и выплачиваем вознаграждение вебмастеру, который привел заказ.

В этом случае, мы не генерируем новых инсталлов, а работаем с лояльной к рекламодателю аудиторией и, таким образом, поддерживаем спрос на его товары или услуги.

3.3.3.2 Второй вариант интеграции

Во втором варианте интеграции — за установки (инсталлы) — мы промотируем само приложение клиента.

Здесь мы делаем упор не на поддержание спроса или лояльность аудитории, а именно на инсталл и первую покупку (это важно для рекламодателя, так как исключает оплату инсталлов, осуществленных бот-фермами).

Механизм заключается в In-app рекламе, т.е. когда пользователь видит рекламу одного приложения внутри другого.

В этом случае нам нужны абсолютно другие вебмастеры, которые ориентируются на знакомство пользователя с приложением, чтобы в дальнейшем получить вознаграждение за установку + первый заказ.

Различие между первым и вторым вариантом в том, что в первом случае каждый заказ атрибуцируется отдельно и мы смотрим на атрибуцию по Last Paid Click, как в мобильном браузере, а во втором варианте атрибуция идет только за установку и первый заказ, и все последующие заказы в этой установке атрибуцируются за приведшим вебмастером, но не оплачиваются – то есть здесь применяется атрибуция Last Paid Click по инсталлам и если инсталл привел к выкупу заказа, то такой инсталл или первая покупка будет оплачен.

3.4 Вопросы и ответы

Здесь мы ответили на дополнительные вопросы, которые могли у вас возникнуть.

3.4.1 Общие вопросы

Как зарегистрироваться в Adv.Cake?

Где посмотреть статистику по каналу?

3.4.2 Финансовые вопросы

Собираем вопросы, пишем ответы.

3.4.3 Технические вопросы

Какие у вас требования к рекламным материалам?

Что делать если я хочу сменить домен сайта?

Что делать если у меня установлен Cloudflare?

3.4.4 Прочие вопросы

Собираем вопросы, пишем ответы.

3.4.4.1 Как зарегистрироваться в Adv.Cake?

Чтобы зарегистрироваться в нашей системе, оставьте заявку на подключение, после чего с вами свяжется наш менеджер и уточнит некоторые детали.

3.4.4.2 Где посмотреть статистику по каналу?

Воспользуйтесь дашбордом! Для этого, в личном кабинете перейдите в раздел «Статистика» → «Дашборд».

На этой странице отображается сводная информация о заказах, финансах, бюджете, конверсии, веб-мастерах и визитах.

3.4.4.3 Что делать если я хочу сменить домен сайта?

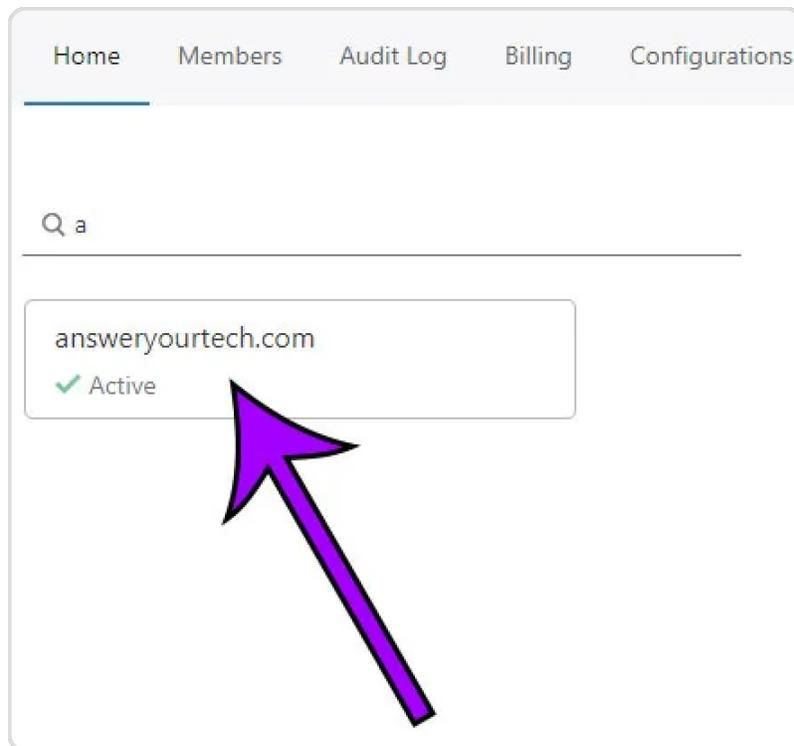
Вы можете сменить домен вашего сайта, однако, перед тем, как это сделать, просим вас сообщить ваш новый домен вашему менеджеру Adv.Cake, чтобы мы могли оперативно отреагировать на изменения и не потерять часть заказов.

3.4.4.4 Что делать если у меня установлен Cloudflare?

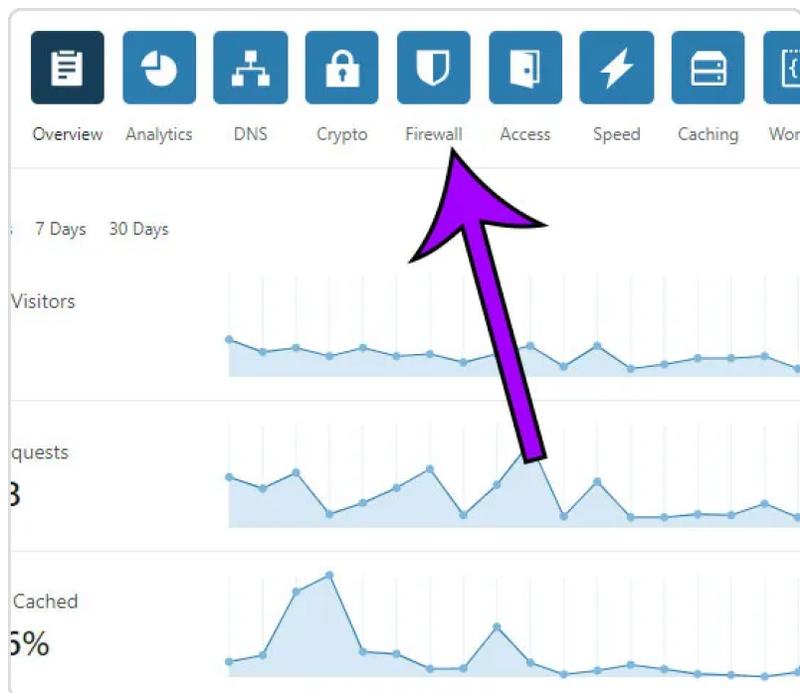
Вам необходимо добавить наш адрес в исключения, иначе заказы не будут приходить к нам в систему.

Для этого:

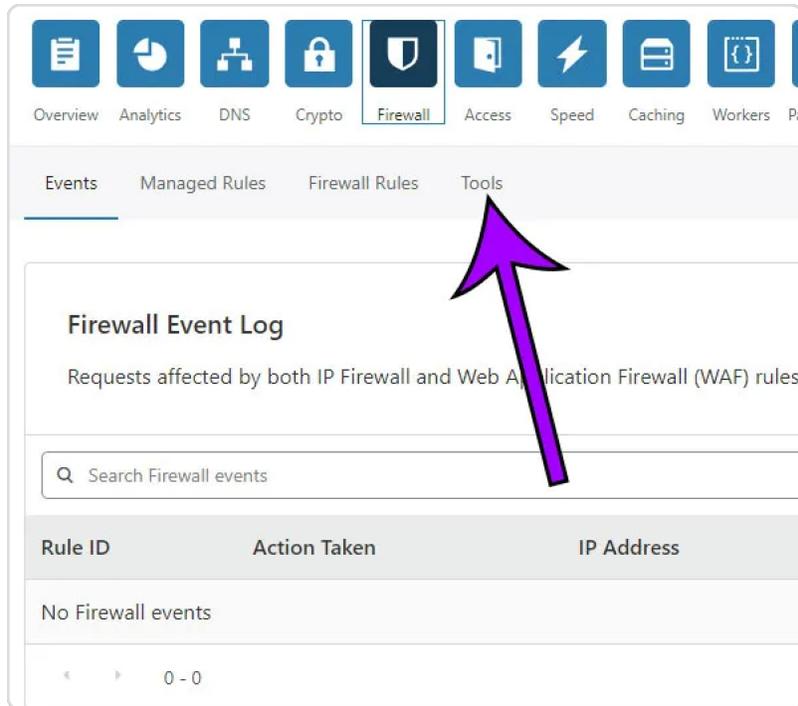
1. Войдите в ваш аккаунт Cloudflare <https://dash.cloudflare.com/login>;
2. Введите название сайта, который вы хотите добавить в исключения;



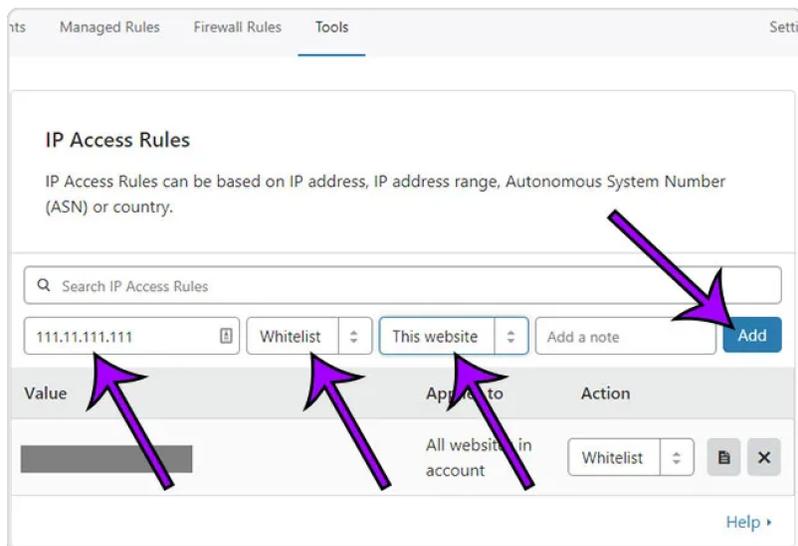
3. Нажмите кнопку *Firewall* в верхней части окна;



4. Нажмите на вкладку **Tools**;



- Введите IP-адрес (который мы вам вышлем) в поле IP-адреса, выберите опцию **Whitelist** в раскрывающемся меню и выберите, следует ли внести этот адрес в белый список для текущего сайта или для всех ваших сайтов. Затем, добавьте примечание (при желании) и нажмите кнопку *Add*:



3.4.4.5 Какие у вас требования к рекламным материалам?

Требования к рекламным материалам перечислены на .

3.5 Пример инструкции Pixel Ecom

3.5.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:
 - Cookie - обезличенный идентификатор пользователя;
 - Cookie - обезличенный идентификатор сессии;
 - Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).
2. Параметры браузера:
 - User-Agent;
 - Размер экрана;
 - Размер окна браузера;
 - Открытие страницы в iframe;
 - URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

1. *Предоставление доступа к рекламному кабинету;*
2. *Установка JS-кода на страницы сайта;*
3. *Установка кода, формирующего DATA-объект на страницы;*
4. *Настройка автоматической сверки с помощью XML-выгрузки.*

3.5.2 Шаг 1. Предоставление доступа к рекламному кабинету

Первым делом вам необходимо предоставить доступ к рекламному кабинету Google Analytics или Яндекс.Метрика:

Если вы используете Google Analytics, предоставьте доступ для аккаунтов `analytics@advcake.com` и `advcake@arcane-splicer-263114.iam.gserviceaccount.com` (аккаунт для запроса данных из Google Analytics по API);

Если вы используете Яндекс.Метрику, предоставьте доступ для аккаунта `metrika@advcake.com`.

Подсказка: Если вы не можете предоставить доступ к этим сервисам, переходите к следующему шагу.

3.5.3 Шаг 2. Установка JS-кода на страницы сайта

Установить JS-код можно двумя способами:

1. Напрямую на сайт (**рекомендуется**):

- Код из примера ниже можно поставить как в `<head>`, так и в `<body>`, однако мы рекомендуем ставить его выше, чтобы другие скрипты, в случае их поломки, не повлияли на работу нашего кода.

2. Через GTM (**не рекомендуется**, т.к. код, установленный через GTM, блокируется блокировщиками рекламы, что приводит к потере конверсий от 30% до 50%):

- Если вы все же решили устанавливать наш код через GTM, необходимо выбрать триггер активации «Page View» или «DOM Ready». Триггер «Window Load» не подходит, так как другие скрипты, которые будут загружаться у пользователя могут заблокировать загрузку нашего скрипта.

Предупреждение: Необходимо отправлять все заказы с любыми источниками.

```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///***.***.com/"; // точный домен для интеграции будет вам передан
    ↪вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

Подсказка: После запуска интеграции вы можете настроить API для автоматической сверки с помощью XML-выгрузки (подробности в шаге 4).

3.5.4 Шаг 3. Установка кода, формирующего DATA-объект на страницы

Установите предоставленные ниже коды на соответствующие страницы вашего сайта:

Примечание: Если на вашем сайте отсутствует какая-либо страница из списка ниже, переходите к настройке следующей страницы.

3.5.4.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    ↪идентификатор, если не передается, данное поле можно оставить пустым
    }
  });
</script>
```

3.5.4.2 2. Страница одного товара

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentCategory: {
      id: 441,
      name: 'Обувь'
    },
    currentProduct: {
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        price: 30000,
        quantity: 2,
        categoryId: 411,
        categoryName: 'Обувь'
    }]
  });
</script>

```

3.5.4.3 3. Страница каталога/категории/подкатегории

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ←идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentCategory: {
      id: 441,
      name: 'Обувь'
    },
    products: [{
      id: 001,
      name: 'Черная кеда',
      price: 12000
    }, {
      id: 002,
      name: 'Серый кед',
      price: 13000
    }],
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
  });
</script>

```

3.5.4.4 4. Страница корзины

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 4,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>
```

3.5.4.5 5. Страница оформления заказа (после корзины и до страницы «Спасибо за заказ»)

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 5,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        categoryName: 'Обувь'
    }]
});
</script>

```

3.5.4.6 6. Страница оформленного заказа («Спасибо за заказ»)

```

<script>
    window.advcake_data = window.advcake_data || [];
    window.advcake_data.push({
        pageType: 6,
        user: {
            email: 'md5(user@email.ru)', // email в виде хеша или уникальный
            ↪идентификатор, если не передается, данное поле можно оставить пустым
            type: 'old' // old, new - опционально, передавать пустое значение в
            ↪кавычках ('), если тип пользователя неизвестен
        },
        basketProducts: [{
            id: 3247,
            name: 'Кроссовок Левый',
            price: 30000,
            quantity: 1,
            categoryId: 411,
            categoryName: 'Обувь'
        }, {
            id: 3247,
            name: 'Кроссовок Правый',
            price: 30000,
            quantity: 2,
            categoryId: 411,
            categoryName: 'Обувь'
        }],
        orderInfo: {
            id: 'ac31004', // уникальный идентификатор заказа, используемый при сверке
            totalPrice: 34647, // сумму заказа передавать без учета доставки
            coupon: 'SKIDKA100' // опционально, передавать пустое значение в кавычках (
            ↪'), если заказ без купона или данный функционал отсутствует на сайте
        }
    });
</script>

```

3.5.5 Шаг 4. Настройка автоматической сверки с помощью XML-выгрузки

Подсказка: Этот шаг **необязателен** для старта, вы можете настроить автоматическую сверку позже.

Нажмите, чтобы развернуть пример XML-выгрузки

```
<orders>
  <order>
    <orderId>order00001</orderId>
    <orderPrice>23000</orderPrice>
    <orderStatus>3</orderStatus>
    <orderBasket>[ { "id": 1107239, "name": "Куртка детская Rip Curl Small Puffer_
↪Jacket Surf The Web", "price": 4580, "quantity": 1, "marketing_cat": "1"}, { "id":_
↪1047488, "name": "Шапка носок Armour Classic Beanie Blue", "price": 490, "quantity":1,
↪"marketing_cat": "2"} ]</orderBasket>
    <description>Не понравилось качество</description>
    <coupon>DCKAL</coupon>
    <clientType>new</clientType>
    <dateCreate>2015-10-14 12:04:00</dateCreate>
    <dateLastChange>2015-10-15 17:00:00</dateLastChange>
  </order>
  <order>
    <orderId>order00002</orderId>
    <orderPrice>51000</orderPrice>
    <orderStatus>2</orderStatus>
    <orderBasket>[{ "id": 1107239, "name": "Куртка детская Rip Curl Small Puffer_
↪Jacket Surf The Web", "price": 4580, "quantity": 1 , "marketing_cat": "2"}, { "id":_
↪1047488, "name": "Шапка носок Armour Classic Beanie Blue", "price": 490, "quantity": 1_
↪, "marketing_cat": "5"} ]</orderBasket>
    <description/>
    <coupon/>
    <clientType>old</clientType>
    <dateCreate>2015-10-13 10:36:00</dateCreate>
    <dateLastChange>2015-10-15 17:03:00</dateLastChange>
  </order>
</orders>
```

Каждый час мы будем забирать из XML-выгрузки информацию о новых заказах/заявках и обновлять статусы по старым.

При передаче GET-параметров ?from=2019-01-01&to=2019-02-01 выдаются заказы за указанный период по дате создания заказа.

3.5.5.1 Расшифровка полей выгрузки

orderId	Номер заявки/лида/заказа
orderPrice	Стоимость заявки. Если понятие стоимости заявки отсутствует, данное поле можно исключить
orderStatus	<p>Статус заявки. Данное поле можно передавать двумя способами:</p> <p>Согласно нашим внутренним идентификаторам:</p> <ul style="list-style-type: none"> – 1 – Заявка в обработке; – 2 – Заявка подтверждена; – 3 – Заявка отменена. <p>Согласно вашим внутренним идентификаторам – в таком случае нам потребуется описание каждого статуса, например:</p> <ul style="list-style-type: none"> – Принят в обработку – транслируем как статус 1; – Подтвержден – транслируем как статус 2; – Обработан – транслируем как статус 3. <p>Если ставка вебмастеру зависит от категории товара, то необходимо передавать также обновленный список товаров (<code>orderBasket</code>);</p> <p><code>orderBasket</code> – состав корзины в формате JSON, например:</p> <pre>[{ "id": 1107239, "name": "Куртка детская Rip Curl ↵ ↵Small Puffer Jacket Surf The Web", "price": 4580, "quantity": 1, "marketing_cat": "1", "category": "164", "category_name": "одежда детская" }, { "id": 1047488, "name": "Шапка носок Armour Classic ↵ ↵Beanie Blue", "price": 490, "quantity": 1, "marketing_cat": "2", "category": "164", "category_name": "одежда детская" }]</pre> <ul style="list-style-type: none"> – в поле <code>price</code> передавать стоимость за единицу товара; – в поле <code>quantity</code> передавать количество данного товара в корзине; – в поле <code>marketing_cat</code> передается маркетинговая категория товара: т.е. каждому товару на сайте присваивается номер маркетинговой категории, на которую мы будем ориентироваться при расчете комиссии (это поле необходимо, если нет разделения комиссии по категориям товаров); <p>Например:</p> <p>Одежда; <code>marketing_cat</code>: "1" – комиссия 5% от</p>

Примечание: Если у вас есть дополнительные идентификаторы по заказу, создайте дополнительные поля и сообщите нам их названия, а также параметры, которые в них будут передаваться.

3.6 Пример инструкции Pixel Lead

3.6.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:

- Cookie - обезличенный идентификатор пользователя;
- Cookie - обезличенный идентификатор сессии;
- Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).

2. Параметры браузера:

- User-Agent;
- Размер экрана;
- Размер окна браузера;
- Открытие страницы в iframe;
- URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

1. *Предоставление доступа к рекламному кабинету;*
2. *Установка JS-кода на страницы сайта;*
3. *Установка кода, формирующего DATA-объект на страницы;*
4. *Настройка автоматической сверки с помощью XML-выгрузки.*

3.6.2 Шаг 1. Предоставление доступа к рекламному кабинету

Первым делом вам необходимо предоставить доступ к рекламному кабинету Google Analytics или Яндекс.Метрика:

Если вы используете Google Analytics, предоставьте доступ для аккаунтов `analytics@advcake.com` и `advcake@arcane-splicer-263114.iam.gserviceaccount.com` (аккаунт для запроса данных из Google Analytics по API);

Если вы используете Яндекс.Метрику, предоставьте доступ для аккаунта `metrika@advcake.com`.

Подсказка: Если вы не можете предоставить доступ к этим сервисам, переходите к следующему шагу.

3.6.3 Шаг 2. Установка JS-кода на страницы сайта

Установить JS-код можно двумя способами:

1. Напрямую на сайт (**рекомендуется**):

- Код из примера ниже можно поставить как в `<head>`, так и в `<body>`, однако мы рекомендуем ставить его выше, чтобы другие скрипты, в случае их поломки, не повлияли на работу нашего кода.

2. Через GTM (**не рекомендуется**, т.к. код, установленный через GTM, блокируется блокировщиками рекламы, что приводит к потере конверсий от 30% до 50%):

- Если вы все же решили устанавливать наш код через GTM, необходимо выбрать триггер активации «Page View» или «DOM Ready». Триггер «Window Load» не подходит, так как другие скрипты, которые будут загружаться у пользователя могут заблокировать загрузку нашего скрипта.

Предупреждение: Необходимо отправлять все заказы с любыми источниками.

```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///**.*.com/"; // точный домен для интеграции будет вам передан
    ↪вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

Подсказка: После запуска интеграции вы можете настроить API для автоматической сверки с помощью XML-выгрузки (подробности в шаге 4).

3.6.4 Шаг 3. Установка кода, формирующего DATA-объект на страницы

Установите предоставленные ниже коды на соответствующие страницы вашего сайта:

Примечание: Если на вашем сайте отсутствует какая-либо страница из списка ниже, переходите к настройке следующей страницы.

3.6.4.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    }
  });
</script>
```

3.6.4.2 2. Страница просмотра лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentLead: {
      leadid: 'a12e', // статический id (курса, продукта и т.п.)
      name: "Курс «Frontend-разработчик»"
    }
  });
</script>
```

3.6.4.3 3. Страница начала создания лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentLead: {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        leadid: 'ab144', // статический id (курса, продукта и т.п.)
        name: 'Курс «Frontend-разработчик»'
    }
});
</script>

```

3.6.4.4 4. Страница созданного лида

```

<script>
    window.advcake_data = window.advcake_data || [];
    window.advcake_data.push({
        pageType: 6,
        user: {
            email: 'md5(user@email.ru)', // email в виде хеша или уникальный
            ↪идентификатор, если не передается, данное поле можно оставить пустым
            type: 'old' // old, new - опционально, передавать пустое значение в
            ↪кавычках (''), если тип пользователя неизвестен
        },
        leadInfo: {
            id: 'ac31004',
            leadid: '123', // статический id (курса, продукта и т.п.)
            name: 'Курс «Frontend-разработчик»',
            totalPrice: 4800,
            coupon: 'HELLO'
        }
    });
</script>

```

3.6.5 Шаг 4. Настройка автоматической сверки с помощью XML-выгрузки

Подсказка: Этот шаг **необязателен** для старта, вы можете настроить автоматическую сверку позже.

Нажмите, чтобы развернуть пример XML-выгрузки

```

<orders>
  <order>
    <orderId>order00001</orderId>
    <orderPrice>23000</orderPrice>
    <orderStatus>3</orderStatus>
    <description>Не понравилось качество</description>
    <coupon>DCKAL</coupon>
    <dateCreate>2015-10-14 12:04:00</dateCreate>
    <dateLastChange>2015-10-15 17:00:00</dateLastChange>
  </order>
  <order>
    <orderId>order00002</orderId>
    <orderPrice>51000</orderPrice>
    <orderStatus>2</orderStatus>

```

(continues on next page)

(продолжение с предыдущей страницы)

```
<description>отказ</description>
<coupon/>
<dateCreate>2015-10-13 10:36:00</dateCreate>
<dateLastChange>2015-10-15 17:03:00</dateLastChange>
</order>
</orders>
```

Каждый час мы будем забирать из XML-выгрузки информацию о новых заказах/заявках и обновлять статусы по старым.

При передаче GET-параметров ?from=2019-01-01&to=2019-02-01 выдаются заказы за указанный период по дате создания заказа.

3.6.5.1 Расшифровка полей выгрузки

orderId	Номер заявки/лида/заказа
orderPrice	Стоимость заявки. Если понятие стоимости заявки отсутствует, данное поле можно исключить
orderStatus	Статус заявки. Данное поле можно передавать двумя способами: Согласно нашим внутренним идентификаторам: – 1 – Заявка в обработке; – 2 – Заявка подтверждена; – 3 – Заявка отменена. Согласно вашим внутренним идентификаторам – в таком случае нам потребуется описание каждого статуса, например: – Принят в обработку – транслируем как статус 1; – Подтвержден – транслируем как статус 2; – Обработан – транслируем как статус 3.
leadName	Название лида
marketing_cat	Маркетинговая категория товара: т.е. каждому товару на сайте присваивается номер маркетинговой категории, на которую мы будем ориентироваться при расчете комиссии (это поле необязательно, если нет разделения комиссии по категориям товаров) Например: Регистрация: <code>marketing_cat: "1"</code> – комиссия 5% от стоимости товара; Оплаченный курс: <code>marketing_cat: "2"</code> – комиссия 2% от стоимости товара. Примечание: Если у вас есть различные ставки по продуктам – обратитесь к вашему менеджеру за корректной разбивкой.
orderTrackid	Значение куки <code>advcake_track_id</code> на момент совершения заказа
description	Причина отмены заявки (передавать данное поле желательно, но не обязательно)
coupon	Использованный в заказе купон/промокод, если такая опция используется. Если такой опции нет, данное поле можно исключить
clientType	Тип клиента: “новый”, “старый”. Может понадобиться для разделения ставок для новых/старых клиентов, а также для аналитики (передавать данное поле желательно, но не обязательно)
dateCreate	Дата создания заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
dateLastChange	Дата изменения статуса заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)

Примечание: Если у вас есть дополнительные идентификаторы по заказу, создайте дополнительные поля и сообщите нам их названия, а также параметры, которые в них будут передаваться.

3.7 Инструкция Pixel Lead без установки JS-кодов

3.7.1 Настройка пикселя

Пример отправки заказа/заявки:

```
https://api.ekacvda.com/postback/<siteru>?id=1411551&totalPrice=10141.14&coupon=A270&clientType=new&trackId=14511aca1b09e91f&url=https%3a%2f%2fsite.ru%2f%3futm_source%3dadvcake%26utm_campaign%3dcpa%26advcake_params%3dafacaafa15aac&email=abc@gmail.com&leadName=english_course&timeLastChanged=2021-01-01%2021%3A30%3A10&status=1
```

Подсказка: Вместо <siteru> подставьте название вашего сайта без точки: <https://api.ekacvda.com/postback/exampleru> (согласовать с нами).

3.7.1.1 Расшифровка параметров

id	Номер заявки/лида/заказа
totalPrice	Стоимость заявки. Если понятие стоимости заявки отсутствует, данное поле можно исключить
coupon	Желательно передавать при отдельной ставке для заказов с купоном
clientType	Тип клиента: “новый” (new), “старый” (old). Может понадобиться для разделения ставок для новых/старых клиентов, а также для аналитики
url	URL со значением меток последнего платного перехода
email	Почта клиента (можно передавать телефон или зашифрованный идентификатор)
leadName	Название лида
marketing_cat	Маркетинговая категория товара: т.е. каждому товару на сайте присваивается номер маркетинговой категории, на которую мы будем ориентироваться при расчете комиссии (это поле необязательно, если нет разделения комиссии по категориям товаров) Например: Регистрация: <code>marketing_cat: "1"</code> – комиссия 5% от стоимости товара; Оплаченный курс: <code>marketing_cat: "2"</code> – комиссия 2% от стоимости товара.
<hr/> <p>Примечание: Если у вас есть различные ставки по продуктам – обратитесь к вашему менеджеру за корректной разбивкой.</p> <hr/>	
dateCreate	Дата создания заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
timeLastChanged	Дата изменения статуса заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
status	Статус заявки/лида/заказа: 1 – Заявка в обработке; 2 – Заявка подтверждена; 3 – Заявка отменена.

Примечание: Если у вас есть дополнительные идентификаторы по заказу, создайте дополнительные поля и сообщите нам их названия, а также параметры, которые в них будут передаваться.

3.8 Пример инструкции XML Ecom

3.8.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:
 - Cookie - обезличенный идентификатор пользователя;
 - Cookie - обезличенный идентификатор сессии;
 - Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).
2. Параметры браузера:
 - User-Agent;
 - Размер экрана;
 - Размер окна браузера;
 - Открытие страницы в iframe;
 - URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

1. *Предоставление доступа к рекламному кабинету;*
2. *Установка JS-кода на страницы сайта;*
3. *Настройка дополнительных параметров;*
4. *Настройка автоматической сверки с помощью XML-выгрузки;*
5. *Установка кода, формирующего DATA-объект на страницы.*

3.8.2 Шаг 1. Предоставление доступа к рекламному кабинету

Первым делом вам необходимо предоставить доступ к рекламному кабинету Google Analytics или Яндекс.Метрика:

Если вы используете Google Analytics, предоставьте доступ для аккаунтов `analytics@advcake.com` и `advcake@arcane-splicer-263114.iam.gserviceaccount.com` (аккаунт для запроса данных из Google Analytics по API);

Если вы используете Яндекс.Метрику, предоставьте доступ для аккаунта `metrika@advcake.com`.

Подсказка: Если вы не можете предоставить доступ к этим сервисам, переходите к следующему шагу.

3.8.3 Шаг 2. Установка JS-кода на страницы сайта

Установить JS-код можно двумя способами:

1. Напрямую на сайт (**рекомендуется**):

- Код из примера ниже можно поставить как в `<head>`, так и в `<body>`, однако мы рекомендуем ставить его выше, чтобы другие скрипты, в случае их поломки, не повлияли на работу нашего кода.

2. Через GTM (**не рекомендуется**, т.к. код, установленный через GTM, блокируется блокировщиками рекламы, что приводит к потере конверсий от 30% до 50%):

- Если вы все же решили устанавливать наш код через GTM, необходимо выбрать триггер активации «Page View» или «DOM Ready». Триггер «Window Load» не подходит, так как другие скрипты, которые будут загружаться у пользователя могут заблокировать загрузку нашего скрипта.

```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///**/*.***/"; // точный домен для интеграции будет вам передан
    ↪ вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

3.8.4 Шаг 3. Настройка дополнительных параметров

По умолчанию трекинг-ссылка Adv.Cake выглядит так:

```
?utm_source=advcake&utm_medium=cpa&utm_campaign={partner}&utm_content={webmaster}&
↪advcake_params={click_id}
```

Предупреждение: Если вы не можете использовать такой формат ссылок, сообщите нам.

При переходе на сайт пользователю выставляются две куки кодом Adv.Cake:

1. advcake_track_id – случайная последовательность из 32-х букв и цифр.
2. advcake_track_url – полная ссылка переход на сайт с utm_source=advcake.

3.8.5 Шаг 4. Настройка автоматической сверки с помощью XML-выгрузки

Нажмите, чтобы развернуть пример XML-выгрузки

```
<orders>
  <order>
    <orderId>order00001</orderId>
    <orderPrice>23000</orderPrice>
    <orderStatus>3</orderStatus>
    <orderBasket>[ { "id": 1107239, "name": "Куртка детская Rip Curl Small Puffer_
↪Jacket Surf The Web", "price": 4580, "quantity": 1, "marketing_cat": "1"}, { "id":_
↪1047488, "name": "Шапка носок Armour Classic Beanie Blue", "price": 490, "quantity":1,
↪"marketing_cat": "2"} ]</orderBasket>
    <orderTrackid>7d12ff1fd3dd5b9a61dc1f42b927264c</orderTrackid>
    <description>Не понравилось качество</description>
    <coupon>DCKAL</coupon>
    <clientType>new</clientType>
    <dateCreate>2015-10-14 12:04:00</dateCreate>
    <dateLastChange>2015-10-15 17:00:00</dateLastChange>
    <url>http://site.ru/?utm_source=advcake&advcake=1&utm_medium=cpa&utm_
↪campaign=admitad&utm_content=263116&utm_term=2</url>
  </order>
  <order>
    <orderId>order00002</orderId>
    <orderPrice>51000</orderPrice>
    <orderStatus>2</orderStatus>
    <orderBasket>[ { "id": 1107239, "name": "Куртка детская Rip Curl Small Puffer_
↪Jacket Surf The Web", "price": 4580, "quantity": 1 , "marketing_cat": "2"}, { "id":_
↪1047488, "name": "Шапка носок Armour Classic Beanie Blue", "price": 490, "quantity": 1_
↪, "marketing_cat": "5"} ]</orderBasket>
    <orderTrackid>7d12ff1fd3dd5b9a61dc1f42b927264c</orderTrackid>
    <description/>
    <coupon/>
    <clientType>old</clientType>
    <dateCreate>2015-10-13 10:36:00</dateCreate>
    <dateLastChange>2015-10-15 17:03:00</dateLastChange>
    <url>http://site.ru/?utm_source=advcake&utm_medium=cpa&utm_campaign=admitad&utm_
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪ content=263116&advcake_params=1</url>  
  </order>  
</orders>
```

Каждый час мы будем забирать из XML-выгрузки информацию о новых заказах/заявках и обновлять статусы по старым.

При передаче GET-параметров `?from=2019-01-01&to=2019-02-01` выдаются заказы за указанный период по дате создания заказа.

3.8.5.1 Расшифровка полей выгрузки

orderId	Номер заявки/лида/заказа
orderPrice	Стоимость заявки. Если понятие стоимости заявки отсутствует, данное поле можно исключить
orderStatus	<p>Статус заявки. Данное поле можно передавать двумя способами:</p> <p>Согласно нашим внутренним идентификаторам:</p> <ul style="list-style-type: none"> – 1 – Заявка в обработке; – 2 – Заявка подтверждена; – 3 – Заявка отменена. <p>Согласно вашим внутренним идентификаторам – в таком случае нам потребуется описание каждого статуса, например:</p> <ul style="list-style-type: none"> – Принят в обработку – транслируем как статус 1; – Подтвержден – транслируем как статус 2; – Обработан – транслируем как статус 3. <p>Если ставка вебмастеру зависит от категории товара, то необходимо передавать также обновленный список товаров (<code>orderBasket</code>);</p> <p><code>orderBasket</code> – состав корзины в формате JSON, например:</p> <pre>[{ "id": 1107239, "name": "Куртка детская Rip Curl ↵ ↵Small Puffer Jacket Surf The Web", "price": 4580 , "quantity": 1 , "marketing_cat": "1" , "category": "164" , "category_name": "одежда детская" }, { "id": 1047488, "name": "Шапка носок Armour Classic ↵ ↵Beanie Blue" , "price": 490 , "quantity": 1 , "marketing_cat": "2", "category": "164" , "category_name": "одежда детская" }]</pre> <ul style="list-style-type: none"> – в поле <code>price</code> передавать стоимость за единицу товара; – в поле <code>quantity</code> передавать количество данного товара в корзине; – в поле <code>marketing_cat</code> передается маркетинговая категория товара: т.е. каждому товару на сайте присваивается номер маркетинговой категории, на которую мы будем ориентироваться при расчете комиссии. Это поле необходимо, если нет разделения комиссии по категориям товаров; <p>Например:</p> <p>Одежда; <code>marketing_cat</code>: "1" – комиссия 5% от</p>

Примечание: Если у вас есть дополнительные идентификаторы по заказу, создайте дополнительные поля и сообщите нам их названия, а также параметры, которые в них будут передаваться.

3.8.6 Шаг 5. Установка кода, формирующего DATA-объект на страницы

Установите предоставленные ниже коды на соответствующие страницы вашего сайта:

3.8.6.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    }
  });
</script>
```

←идентификатор, если не передается, данное поле можно оставить пустым

3.8.6.2 2. Страница одного товара

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    },
    currentCategory: {
      id: 441,
      name: 'Обувь'
    },
    currentProduct: {
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        name: 'Кроссовок Правый',
        price: 30000,
        quantity: 2,
        categoryId: 411,
        categoryName: 'Обувь'
    }
  ]
});
</script>
```

3.8.6.3 3. Страница каталога/категории/подкатегории

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentCategory: {
      id: 441,
      name: 'Обувь'
    },
    products: [{
      id: 001,
      name: 'Черная кеда',
      price: 12000
    }, {
      id: 002,
      name: 'Серый кед',
      price: 13000
    }
  ]],
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>
```

3.8.6.4 4. Страница корзины

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 4,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>

```

3.8.6.5 5. Страница оформления заказа (после корзины и до страницы «Спасибо за заказ»)

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 5,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
        categoryName: 'Обувь'  
    }]  
});  
</script>
```

3.8.6.6 6. Страница оформленного заказа («Спасибо за заказ»)

```
<script>  
    window.advcake_data = window.advcake_data || [];  
    window.advcake_data.push({  
        pageType: 6,  
        user: {  
            email: 'md5(user@email.ru)', // email в виде хеша или уникальный  
↳идентификатор, если не передается, данное поле можно оставить пустым  
            type: 'old' // old, new - опционально, передавать пустое значение в  
↳кавычках ('), если тип пользователя неизвестен  
        },  
        basketProducts: [{  
            id: 3247,  
            name: 'Кроссовок Левый',  
            price: 30000,  
            quantity: 1,  
            categoryId: 411,  
            categoryName: 'Обувь'  
        }, {  
            id: 3247,  
            name: 'Кроссовок Правый',  
            price: 30000,  
            quantity: 2,  
            categoryId: 411,  
            categoryName: 'Обувь'  
        }],  
        orderInfo: {  
            id: 'ac31004', // уникальный идентификатор заказа, используемый при сверке  
            totalPrice: 34647, // сумму заказа передавать без учета доставки  
            coupon: 'SKIDKA100' // опционально, передавать пустое значение в кавычках ('  
↳'), если заказ без купона или данный функционал отсутствует на сайте  
        }  
    });  
</script>
```

3.8.6.7 7. Страница списка товаров (страница поиска, страница акционных товаров и т.п.)

```

<script>
window.advcake_data = window.advcake_data || [];
window.advcake_data.push({
  pageType: 7,
  user: {
    email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    ↪идентификатор, если не передается, данное поле можно оставить пустым
  },
  products: [{
    id: 001,
    name: 'Черная кеда',
    categoryId: 411,
    categoryName: 'Обувь'
  }, {
    id: 002,
    name: 'Серый кед',
    categoryId: 411,
    categoryName: 'Обувь'
  }],
  basketProducts: [{
    id: 3247,
    name: 'Кроссовок Левый',
    price: 30000,
    quantity: 1,
    categoryId: 411,
    categoryName: 'Обувь'
  }, {
    id: 3247,
    name: 'Кроссовок Правый',
    price: 30000,
    quantity: 2,
    categoryId: 411,
    categoryName: 'Обувь'
  }
  ]
});
</script>

```

3.9 Пример инструкции XML Lead

3.9.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:

- Cookie - обезличенный идентификатор пользователя;
- Cookie - обезличенный идентификатор сессии;
- Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).

2. Параметры браузера:

- User-Agent;
- Размер экрана;
- Размер окна браузера;
- Открытие страницы в iframe;
- URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

При интеграции через XML вы создаете XML-файл определенной структуры, в который необходимо передавать данные о заказах, совершенных пользователями на вашем сайте, пришедшими по меткам CPA-канала. Каждый час к этому файлу обращается сервер Adv.Cake и выгружает из него данные по новым действиям в статистику.

Мы не собираем никакой личной информации, мы используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

1. *Предоставление доступа к рекламному кабинету;*
2. *Установка JS-кода на страницы сайта;*
3. *Настройка дополнительных параметров;*
4. *Настройка автоматической сверки с помощью XML-выгрузки;*
5. *Установка кода, формирующего DATA-объект на страницы.*

3.9.2 Шаг 1. Предоставление доступа к рекламному кабинету

Первым делом вам необходимо предоставить доступ к рекламному кабинету Google Analytics или Яндекс.Метрика:

Если вы используете Google Analytics, предоставьте доступ для аккаунтов analytics@advcake.com и advcake@arcane-splicer-263114.iam.gserviceaccount.com (аккаунт для запроса данных из Google Analytics по API);

Если вы используете Яндекс.Метрику, предоставьте доступ для аккаунта metrika@advcake.com.

Подсказка: Если вы не можете предоставить доступ к этим сервисам, переходите к следующему шагу.

3.9.3 Шаг 2. Установка JS-кода на страницы сайта

Установить JS-код можно двумя способами:

1. Напрямую на сайт (**рекомендуется**):

- Код из примера ниже можно поставить как в `<head>`, так и в `<body>`, однако мы рекомендуем ставить его выше, чтобы другие скрипты, в случае их поломки, не повлияли на работу нашего кода.

2. Через GTM (**не рекомендуется**, т.к. код, установленный через GTM, блокируется блокировщиками рекламы, что приводит к потере конверсий от 30% до 50%):

- Если вы все же решили устанавливать наш код через GTM, необходимо выбрать триггер активации «Page View» или «DOM Ready». Триггер «Window Load» не подходит, так как другие скрипты, которые будут загружаться у пользователя могут заблокировать загрузку нашего скрипта.

```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///***.***.com/"; // точный домен для интеграции будет вам передан
    ↪ вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

3.9.4 Шаг 3. Настройка дополнительных параметров

По умолчанию трекинг-ссылка Adv.Cake выглядит так:

```
?utm_source=advcake&utm_medium=cpa&utm_campaign={partner}&utm_content={webmaster}&
  ↪ advcake_params={click_id}
```

Предупреждение: Если вы не можете использовать такой формат ссылок, сообщите нам.

При переходе на сайт пользователю выставляются две куки кодом Adv.Cake:

1. `advcake_track_id` – случайная последовательность из 32-х букв и цифр.
2. `advcake_track_url` – полная ссылка переход на сайт с `utm_source=advcake`.

3.9.5 Шаг 4. Настройка автоматической сверки с помощью XML-выгрузки

Нажмите, чтобы развернуть пример XML-выгрузки

```
<orders>
  <order>
    <orderId>order00001</orderId>
    <orderPrice>23000</orderPrice>
    <orderStatus>3</orderStatus>
    <orderTrackid>7d12ff1fd3dd5b9a61dc1f42b927264c</orderTrackid>
    <description>Не понравилось качество</description>
    <coupon>DCKAL</coupon>
    <dateCreate>2015-10-14 12:04:00</dateCreate>
    <dateLastChange>2015-10-15 17:00:00</dateLastChange>
    <url>http://site.ru/?utm_source=advcake&advcake=1&utm_medium=cpa&utm_
↪campaign=admitad&utm_content=263116&utm_term=2</url>
  </order>
  <order>
    <orderId>order00002</orderId>
    <orderPrice>51000</orderPrice>
    <orderStatus>2</orderStatus>
    <orderTrackid>7d12ff1fd3dd5b9a61dc1f42b927264c</orderTrackid>
    <description>отказ</description>
    <coupon/>
    <dateCreate>2015-10-13 10:36:00</dateCreate>
    <dateLastChange>2015-10-15 17:03:00</dateLastChange>
    <url>http://site.ru/?utm_source=advcake&utm_medium=cpa&utm_campaign=admitad&utm_
↪content=263116&advcake_params=1</url>
  </order>
</orders>
```

Каждый час мы будем забирать из XML-выгрузки информацию о новых заказах/заявках и обновлять статусы по старым.

При передаче GET-параметров ?from=2019-01-01&to=2019-02-01 выдаются заказы за указанный период по дате создания заказа.

3.9.5.1 Расшифровка полей выгрузки

orde	Номер заявки/лида/заказа
orde	Стоимость заявки. Если понятие стоимости заявки отсутствует, данное поле можно исключить
orde	Статус заявки. Данное поле можно передавать двумя способами: Согласно нашим внутренним идентификаторам: – 1 – Заявка в обработке; – 2 – Заявка подтверждена; – 3 – Заявка отменена. Согласно вашим внутренним идентификаторам – в таком случае нам потребуется описание каждого статуса, например: – Принят в обработку – транслируем как статус 1; – Подтвержден – транслируем как статус 2; – Обработан – транслируем как статус 3.
desc	Причина отмены заявки (передавать данное поле желательно, но не обязательно)
coup	Использованный в заказе купон/промокод, если такая опция используется. Если такой опции нет, данное поле можно исключить
clie	Тип клиента: “новый”, “старый”. Может понадобиться для разделения ставок для новых/старых клиентов, а также для аналитики (передавать данное поле желательно, но не обязательно)
orde	Значение куки <code>advcake_track_id</code> на момент совершения заказа (см. шаг 3)
date	Дата создания заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
date	Дата изменения статуса заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
url	Значение куки <code>advcake_track_url</code> на момент совершения заказа (см. шаг 3)

Примечание: Если у вас есть дополнительные идентификаторы по заказу, создайте дополнительные поля и сообщите нам их названия, а также параметры, которые в них будут передаваться.

3.9.6 Шаг 5. Установка кода, формирующего DATA-объект на страницы

Установите предоставленные ниже коды на соответствующие страницы вашего сайта:

3.9.6.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ←идентификатор, если не передается, данное поле можно оставить пустым
    }
  });
</script>
```

3.9.6.2 2. Страница просмотра лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentLead: {
      name: "Регистрация"
    }
  });
</script>
```

3.9.6.3 3. Страница начала создания лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentLead: {
      name: 'Кредит'
    }
  });
</script>
```

3.9.6.4 4. Страница созданного лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 4,
    user: {
      email: 'md5(user@email.ru)', // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
      type: 'old' // old, new - опционально
    },
    leadInfo: {
      id: 'ac31004',
      name: 'Страховка'
    }
  });
</script>
```

3.9.6.5 5. Страница списка лидов

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 5,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    leads: [{
      name: 'Регистрация',
    }, {
      name: 'Кредит',
    }]
  });
</script>

```

3.10 Пример инструкции Postback Ecom

3.10.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:

- Cookie - обезличенный идентификатор пользователя;
- Cookie - обезличенный идентификатор сессии;
- Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).

2. Параметры браузера:

- User-Agent;
- Размер экрана;
- Размер окна браузера;
- Открытие страницы в iframe;
- URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед

совершением заказа.

Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

1. Предоставление доступа к рекламному кабинету;
2. Установка JS-кода на страницы сайта;
3. Настройка дополнительных параметров;
4. Настройка postback;
5. Установка кода, формирующего DATA-объект на страницы.

3.10.2 Шаг 1. Предоставление доступа к рекламному кабинету

Первым делом вам необходимо предоставить доступ к рекламному кабинету Google Analytics или Яндекс.Метрика:

Если вы используете Google Analytics, предоставьте доступ для аккаунтов analytics@advcake.com и advcake@arcane-splicer-263114.iam.gserviceaccount.com (аккаунт для запроса данных из Google Analytics по API);

Если вы используете Яндекс.Метрику, предоставьте доступ для аккаунта metrika@advcake.com.

Подсказка: Если вы не можете предоставить доступ к этим сервисам, переходите к следующему шагу.

3.10.3 Шаг 2. Установка JS-кода на страницы сайта

Установить JS-код можно двумя способами:

1. Напрямую на сайт (**рекомендуется**):
 - Код из примера ниже можно поставить как в `<head>`, так и в `<body>`, однако мы рекомендуем ставить его выше, чтобы другие скрипты, в случае их поломки, не повлияли на работу нашего кода.
2. Через GTM (**не рекомендуется**, т.к. код, установленный через GTM, блокируется блокировщиками рекламы, что приводит к потере конверсий от 30% до 50%):
 - Если вы все же решили устанавливать наш код через GTM, необходимо выбрать триггер активации «Page View» или «DOM Ready». Триггер «Window Load» не подходит, так как другие скрипты, которые будут загружаться у пользователя могут заблокировать загрузку нашего скрипта.

Подсказка: Вместо `siteru` подставьте название вашего сайта, например: `https://api.ekacvda.com/postback/exampleru` (без точки).

Примечание: Отправлять заказ/заявку желательно POST-запросом в формате JSON (например, с помощью XHR) т.к. длина GET-запроса ограничена 2048 символами, из-за чего часть данных может не дойти.

3.10.5.1 Расшифровка полей постбека

id	Номер заявки/лида/заказа
totalPrice	Стоимость заказа. Если понятие стоимости заказа отсутствует, данное поле можно исключить
orderStatus	<p>Статус заказа. Данное поле можно передавать двумя способами:</p> <p>Согласно нашим внутренним идентификаторам:</p> <ul style="list-style-type: none"> – 1 – Заказ в обработке; – 2 – Заказ подтвержден; – 3 – Заказ отменен. <p>Согласно вашим внутренним идентификаторам – в таком случае нам потребуется описание каждого статуса, например:</p> <ul style="list-style-type: none"> – Принят в обработку – транслируем как статус 1; – Подтвержден – транслируем как статус 2; – Обработан – транслируем как статус 3. <p>Если ставка вебмастеру зависит от категории товара, то необходимо передавать также обновленный список товаров (<code>orderBasket</code>);</p> <p><code>orderBasket</code> – состав корзины в формате JSON, например:</p> <pre>[{ "id": 1107239, "name": "Куртка детская Rip Curl ↵ ↵Small Puffer Jacket Surf The Web", "price": 4580, "quantity": 1, "marketing_cat": "1", "category": "164", "category_name": "одежда детская" }, { "id": 1047488, "name": "Шапка носок Armour Classic ↵ ↵Beanie Blue", "price": 490, "quantity": 1, "marketing_cat": "2", "category": "164", "category_name": "одежда детская" }]</pre> <ul style="list-style-type: none"> – в поле <code>price</code> передавать стоимость за единицу товара; – в поле <code>quantity</code> передавать количество данного товара в корзине; – в поле <code>marketing_cat</code> передается маркетинговая категория товара: т.е. каждому товару на сайте присваивается номер маркетинговой категории, на которую мы будем ориентироваться при расчете комиссии (это поле необходимо, если нет разделения комиссии по категориям товаров); <p>Например:</p> <p>Одежда; <code>marketing_cat</code>: "1" – комиссия 5% от</p>

Примечание: Если у вас есть дополнительные идентификаторы по заказу, создайте дополнительные поля и сообщите нам их названия, а также параметры, которые в них будут передаваться.

3.10.6 Шаг 5. Установка кода, формирующего DATA-объект на страницы

Установите предоставленные ниже коды на соответствующие страницы вашего сайта:

3.10.6.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    }
  });
</script>
```

←идентификатор, если не передается, данное поле можно оставить пустым

3.10.6.2 2. Страница одного товара

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    },
    currentCategory: {
      id: 441,
      name: 'Обувь'
    },
    currentProduct: {
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        name: 'Кроссовок Правый',
        price: 30000,
        quantity: 2,
        categoryId: 411,
        categoryName: 'Обувь'
    }
  ]
});
</script>
```

3.10.6.3 3. Страница каталога/категории/подкатегории

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ←идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentCategory: {
      id: 441,
      name: 'Обувь'
    },
    products: [{
      id: 001,
      name: 'Черная кеда',
      price: 12000
    }, {
      id: 002,
      name: 'Серый кед',
      price: 13000
    }
  ]],
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>
```

3.10.6.4 4. Страница корзины

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 4,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>

```

3.10.6.5 5. Страница оформления заказа (после корзины и до страницы «Спасибо за заказ»)

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 5,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
        categoryName: 'Обувь'  
    }]  
});  
</script>
```

3.10.6.6 6. Страница оформленного заказа («Спасибо за заказ»)

```
<script>  
    window.advcake_data = window.advcake_data || [];  
    window.advcake_data.push({  
        pageType: 6,  
        user: {  
            email: 'md5(user@email.ru)', // email в виде хеша или уникальный  
↳идентификатор, если не передается, данное поле можно оставить пустым  
            type: 'old' // old, new - опционально, передавать пустое значение в  
↳кавычках ('), если тип пользователя неизвестен  
        },  
        basketProducts: [{  
            id: 3247,  
            name: 'Кроссовок Левый',  
            price: 30000,  
            quantity: 1,  
            categoryId: 411,  
            categoryName: 'Обувь'  
        }, {  
            id: 3247,  
            name: 'Кроссовок Правый',  
            price: 30000,  
            quantity: 2,  
            categoryId: 411,  
            categoryName: 'Обувь'  
        }],  
        orderInfo: {  
            id: 'ac31004', // уникальный идентификатор заказа, используемый при сверке  
            totalPrice: 34647, // сумму заказа передавать без учета доставки  
            coupon: 'SKIDKA100' // опционально, передавать пустое значение в кавычках ('  
↳'), если заказ без купона или данный функционал отсутствует на сайте  
        }  
    });  
</script>
```

3.10.6.7 7. Страница списка товаров (страница поиска, страница акционных товаров и т.п.)

```

<script>
window.advcake_data = window.advcake_data || [];
window.advcake_data.push({
  pageType: 7,
  user: {
    email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    ↪идентификатор, если не передается, данное поле можно оставить пустым
  },
  products: [{
    id: 001,
    name: 'Черная кеда',
    categoryId: 411,
    categoryName: 'Обувь'
  }, {
    id: 002,
    name: 'Серый кед',
    categoryId: 411,
    categoryName: 'Обувь'
  }],
  basketProducts: [{
    id: 3247,
    name: 'Кроссовок Левый',
    price: 30000,
    quantity: 1,
    categoryId: 411,
    categoryName: 'Обувь'
  }, {
    id: 3247,
    name: 'Кроссовок Правый',
    price: 30000,
    quantity: 2,
    categoryId: 411,
    categoryName: 'Обувь'
  }
  ]
});
</script>

```

3.11 Пример инструкции Postback Lead

3.11.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:

- Cookie - обезличенный идентификатор пользователя;
- Cookie - обезличенный идентификатор сессии;
- Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).

2. Параметры браузера:

- User-Agent;
- Размер экрана;
- Размер окна браузера;
- Открытие страницы в iframe;
- URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

1. *Предоставление доступа к рекламному кабинету;*
2. *Установка JS-кода на страницы сайта;*
3. *Настройка дополнительных параметров;*
4. *Настройка postback;*
5. *Установка кода, формирующего DATA-объект на страницы.*

3.11.2 Шаг 1. Предоставление доступа к рекламному кабинету

Первым делом вам необходимо предоставить доступ к рекламному кабинету Google Analytics или Яндекс.Метрика:

Если вы используете Google Analytics, предоставьте доступ для аккаунтов analytics@advcake.com и advcake@arcane-splicer-263114.iam.gserviceaccount.com (аккаунт для запроса данных из Google Analytics по API);

Если вы используете Яндекс.Метрику, предоставьте доступ для аккаунта metrika@advcake.com.

Подсказка: Если вы не можете предоставить доступ к этим сервисам, переходите к следующему шагу.

3.11.3 Шаг 2. Установка JS-кода на страницы сайта

Установить JS-код можно двумя способами:

1. Напрямую на сайт (**рекомендуется**):

- Код из примера ниже можно поставить как в `<head>`, так и в `<body>`, однако мы рекомендуем ставить его выше, чтобы другие скрипты, в случае их поломки, не повлияли на работу нашего кода.

2. Через GTM (**не рекомендуется**, т.к. код, установленный через GTM, блокируется блокировщиками рекламы, что приводит к потере конверсий от 30% до 50%):

- Если вы все же решили устанавливать наш код через GTM, необходимо выбрать триггер активации «Page View» или «DOM Ready». Триггер «Window Load» не подходит, так как другие скрипты, которые будут загружаться у пользователя могут заблокировать загрузку нашего скрипта.

```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///**.*.com/"; // точный домен для интеграции будет вам передан
    ↪ вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

3.11.4 Шаг 3. Настройка дополнительных параметров

По умолчанию трекинг-ссылка Adv.Cake выглядит так:

```
?utm_source=advcake&utm_medium=cpa&utm_campaign={partner}&utm_content={webmaster}&
↪ advcake_params={click_id}
```

Предупреждение: Если вы не можете использовать такой формат ссылок, сообщите нам.

При переходе на сайт пользователю выставляются две куки кодом Adv.Cake:

1. `advcake_track_id` – случайная последовательность из 32-х букв и цифр.
2. `advcake_track_url` – полная ссылка переход на сайт с `utm_source=advcake`.

3.11.5 Шаг 4. Настройка postback

Пример отправки заказа/заявки:

Предупреждение: Необходимо отправлять все заказы с любыми источниками.

```
https://api.ekacvda.com/postback/siteru?id=1411551&totalPrice=10141.14&coupon=A270&clientType=new&trackId=14511aca1b09e91f&url=https%3a%2f%2fsite.ru%2f%3futm_source%3dadvcake%26utm_campaign%3dcpa%26advcake_params%3dafacaafa15aac&mail=abc@gmail.com&leadName=english_course&timeLastChanged=2021-01-01%2021%3A30%3A10&status=1
```

Подсказка: Вместо `siteru` подставьте название вашего сайта, например: `https://api.ekacvda.com/postback/exampleru` (без точки).

Примечание: Отправлять заказ/заявку желательно POST-запросом в формате JSON (например, с помощью XHR) т.к. длина GET-запроса ограничена 2048 символами, из-за чего часть данных может не дойти.

3.11.5.1 Расшифровка полей постбека

id	Номер заявки/лида/заказа
totalPrice	Стоимость заявки. Если понятие стоимости заявки отсутствует, данное поле можно исключить
coupon	Желательно передавать при отдельной ставке для заказов с купоном
clientType	Тип клиента: “новый” (new), “старый” (old). Может понадобиться для разделения ставок для новых/старых клиентов, а также для аналитики
trackid	Значение куки advcake_track_id на момент совершения заказа (см. шаг 3)
url	Значение куки advcake_track_url на момент совершения заказа (см. шаг 3)
email	Почта клиента (можно передавать телефон или зашифрованный идентификатор)
leadName	Название лида
marketing_cat	Маркетинговая категория товара: т.е. каждому товару на сайте присваивается номер маркетинговой категории, на которую мы будем ориентироваться при расчете комиссии (это поле необязательно, если нет разделения комиссии по категориям товаров) Например: Регистрация: marketing_cat: "1" – комиссия 5% от стоимости товара; Оплаченный курс: marketing_cat: "2" – комиссия 2% от стоимости товара.
	Примечание: Если у вас есть различные ставки по продуктам – обратитесь к вашему менеджеру за корректной разбивкой.
dateCreate	Дата создания заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
timeLastChanged	Дата изменения статуса заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
status	Статус заявки/лида/заказа: 1 – Заявка в обработке; 2 – Заявка подтверждена; 3 – Заявка отменена.

Примечание: Если у вас есть дополнительные идентификаторы по заказу, создайте дополнительные поля и сообщите нам их названия, а также параметры, которые в них будут передаваться.

3.11.6 Шаг 5. Установка кода, формирующего DATA-объект на страницы

Установите предоставленные ниже коды на соответствующие страницы вашего сайта:

3.11.6.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    }
  });
</script>
```

←идентификатор, если не передается, данное поле можно оставить пустым

3.11.6.2 2. Страница просмотра лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    },
    currentLead: {
      leadid: 'a12e', // статический id (курса, продукта и т.п.)
      name: "Курс «Frontend-разработчик»",
      price: 123 // цена - передается если есть
    }
  });
</script>
```

←идентификатор, если не передается, данное поле можно оставить пустым

3.11.6.3 3. Страница начала создания лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    },
    currentLead: {
      leadid: 'ab144', // статический id (курса, продукта и т.п.)
      name: "Курс «Frontend-разработчик»",
      price: 123 // цена - передается если есть
    }
  });
</script>
```

←идентификатор, если не передается, данное поле можно оставить пустым

(continues on next page)

(продолжение с предыдущей страницы)

```

    }
  });
</script>

```

3.11.6.4 4. Страница созданного лида

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 6,
    user: {
      email: 'md5(user@email.ru)', // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
      type: 'old' // old, new - опционально, передавать пустое значение в
      ↪кавычках (')), если тип пользователя неизвестен
    },
    leadInfo: {
      id: 'ac31004', // id заказа
      leadid: '123', // статический id (курса, продукта и т.п.)
      name: "Курс «Frontend-разработчик»",
      totalPrice: 4800,
      coupon: 'HELLO'
    }
  });
</script>

```

3.11.6.5 5. Страница списка лидов

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 5,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    leads: [{
      leadid: 'ac156', // статический id (курса, продукта и т.п.)
      name: "Курс «Frontend-разработчик»",
      totalPrice: 15800
    }, {
      leadid: 'ab144',
      name: "Курс «Frontend-разработчик»",
      totalPrice: 4800
    }
  ]
  });
</script>

```

3.12 Пример инструкции Pixel Lead для финансовых офферов

3.12.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:
 - Cookie - обезличенный идентификатор пользователя;
 - Cookie - обезличенный идентификатор сессии;
 - Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).
2. Параметры браузера:
 - User-Agent;
 - Размер экрана;
 - Размер окна браузера;
 - Открытие страницы в iframe;
 - URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

1. *Предоставление доступа к рекламному кабинету;*
2. *Установка JS-кода на страницы сайта;*
3. *Установка кода, формирующего DATA-объект на страницы;*
4. *Настройка автоматической сверки с помощью XML-выгрузки.*

3.12.2 Шаг 1. Предоставление доступа к рекламному кабинету

Первым делом вам необходимо предоставить доступ к рекламному кабинету Google Analytics или Яндекс.Метрика:

Если вы используете Google Analytics, предоставьте доступ для аккаунтов `analytics@advcake.com` и `advcake@arcane-splicer-263114.iam.gserviceaccount.com` (аккаунт для запроса данных из Google Analytics по API);

Если вы используете Яндекс.Метрику, предоставьте доступ для аккаунта `metrika@advcake.com`.

Подсказка: Если вы не можете предоставить доступ к этим сервисам, переходите к следующему шагу.

3.12.3 Шаг 2. Установка JS-кода на страницы сайта

Установить JS-код можно двумя способами:

1. Напрямую на сайт (**рекомендуется**):

- Код из примера ниже можно поставить как в `<head>`, так и в `<body>`, однако мы рекомендуем ставить его выше, чтобы другие скрипты, в случае их поломки, не повлияли на работу нашего кода.

2. Через GTM (**не рекомендуется**, т.к. код, установленный через GTM, блокируется блокировщиками рекламы, что приводит к потере конверсий от 30% до 50%):

- Если вы все же решили устанавливать наш код через GTM, необходимо выбрать триггер активации «Page View» или «DOM Ready». Триггер «Window Load» не подходит, так как другие скрипты, которые будут загружаться у пользователя могут заблокировать загрузку нашего скрипта.

Предупреждение: Необходимо отправлять все заказы с любыми источниками.

```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///**.*.com/"; // точный домен для интеграции будет вам передан
    ↪вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

Подсказка: После запуска интеграции вы можете настроить API для автоматической сверки с помощью XML-выгрузки (подробности в шаге 4).

3.12.4 Шаг 3. Установка кода, формирующего DATA-объект на страницы

Установите предоставленные ниже коды на соответствующие страницы вашего сайта:

Примечание: Если на вашем сайте отсутствует какая-либо страница из списка ниже, переходите к настройке следующей страницы.

3.12.4.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    }
  });
</script>
```

3.12.4.2 2. Страница просмотра лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentLead: {
      product: 'a12e', // статический id продукта
      name: "Кредит"
    }
  });
</script>
```

3.12.4.3 3. Страница начала создания лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentLead: {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        product: 'ab144', // статический id продукта
        name: 'Кредит'
    }
});
</script>

```

3.12.4.4 4. Страница созданного лида

```

<script>
    window.advcake_data = window.advcake_data || [];
    window.advcake_data.push({
        pageType: 6,
        user: {
            email: 'md5(user@email.ru)', // email в виде хеша или уникальный
            ↪идентификатор, если не передается, данное поле можно оставить пустым
            type: 'old' // old, new - опционально, передавать пустое значение в
            ↪кавычках (''), если тип пользователя неизвестен
        },
        leadInfo: {
            id: 'ac31004',
            product: '123', // статический id продукта
            name: 'Кредит',
            totalPrice: 4800,
            coupon: 'HELLO'
        }
    });
</script>

```

3.12.5 Шаг 4. Настройка автоматической сверки с помощью XML-выгрузки

Подсказка: Этот шаг **необязателен** для старта, вы можете настроить автоматическую сверку позже.

Нажмите, чтобы развернуть пример XML-выгрузки

```

<orders>
  <order>
    <orderId>order00001</orderId>
    <orderPrice>23000</orderPrice>
    <orderStatus>3</orderStatus>
    <description>Не понравилось качество</description>
    <coupon>DCKAL</coupon>
    <dateCreate>2015-10-14 12:04:00</dateCreate>
    <dateLastChange>2015-10-15 17:00:00</dateLastChange>
  </order>
  <order>
    <orderId>order00002</orderId>
    <orderPrice>51000</orderPrice>
    <orderStatus>2</orderStatus>

```

(continues on next page)

(продолжение с предыдущей страницы)

```

<description>отказ</description>
<coupon/>
<dateCreate>2015-10-13 10:36:00</dateCreate>
<dateLastChange>2015-10-15 17:03:00</dateLastChange>
</order>
</orders>

```

Каждый час мы будем забирать из XML-выгрузки информацию о новых заказах/заявках и обновлять статусы по старым.

При передаче GET-параметров `?from=2019-01-01&to=2019-02-01` выдаются заказы за указанный период по дате создания заказа.

3.12.5.1 Расшифровка полей выгрузки

<code>orde</code>	Номер заявки/лида/заказа
<code>orde</code>	Стоимость заявки. Если понятие стоимости заявки отсутствует, данное поле можно исключить
<code>orde</code>	Статус заявки. Данное поле можно передавать двумя способами: Согласно нашим внутренним идентификаторам: – 1 – Заявка в обработке; – 2 – Заявка подтверждена; – 3 – Заявка отменена. Согласно вашим внутренним идентификаторам – в таком случае нам потребуется описание каждого статуса, например: – Принят в обработку – транслируем как статус 1; – Подтвержден – транслируем как статус 2; – Обработан – транслируем как статус 3.
<code>orde</code>	Значение куки <code>advcake_track_id</code> на момент совершения заказа
<code>desc</code>	Причина отмены заявки (передавать данное поле желательно, но не обязательно)
<code>coup</code>	Использованный в заказе купон/промокод, если такая опция используется. Если такой опции нет, данное поле можно исключить
<code>clie</code>	Тип клиента: “новый”, “старый”. Может понадобиться для разделения ставок для новых/старых клиентов, а также для аналитики (передавать данное поле желательно, но не обязательно)
<code>date</code>	Дата создания заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
<code>date</code>	Дата изменения статуса заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
<code>date</code>	Дата изменения статуса заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
<code>prod</code>	Соответствующее значение под конкретный продукт, например: <code>credit_card</code> - Кредитная карта; <code>VKO</code> - Расчетно-кассовое обслуживание.
<code>lead</code>	При необходимости (например, наличие двух или более категорий продукта) передается категория продукта: Например, для продукта «РКО», могут быть указаны следующие категории: <code>VKO-IP</code> - Расчетно-кассовое обслуживание для ИП; <code>VKO-000</code> - Расчетно-кассовое обслуживание для ООО.

Примечание: Если у вас есть дополнительные идентификаторы по заказу, создайте дополнительные поля и сообщите нам их названия, а также параметры, которые в них будут передаваться.

3.13 Пример инструкции Postback Lead для финансовых офферов

3.13.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:
 - Cookie - обезличенный идентификатор пользователя;
 - Cookie - обезличенный идентификатор сессии;
 - Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).
2. Параметры браузера:
 - User-Agent;
 - Размер экрана;
 - Размер окна браузера;
 - Открытие страницы в iframe;
 - URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

1. *Предоставление доступа к рекламному кабинету;*
2. *Установка JS-кода на страницы сайта;*
3. *Настройка дополнительных параметров;*
4. *Настройка postback;*
5. *Установка кода, формирующего DATA-объект на страницы.*

3.13.2 Шаг 1. Предоставление доступа к рекламному кабинету

Первым делом вам необходимо предоставить доступ к рекламному кабинету Google Analytics или Яндекс.Метрика:

Если вы используете Google Analytics, предоставьте доступ для аккаунтов `analytics@advcake.com` и `advcake@arcane-splicer-263114.iam.gserviceaccount.com` (аккаунт для запроса данных из Google Analytics по API);

Если вы используете Яндекс.Метрику, предоставьте доступ для аккаунта `metrika@advcake.com`.

Подсказка: Если вы не можете предоставить доступ к этим сервисам, переходите к следующему шагу.

3.13.3 Шаг 2. Установка JS-кода на страницы сайта

Установить JS-код можно двумя способами:

1. Напрямую на сайт (**рекомендуется**):

- Код из примера ниже можно поставить как в `<head>`, так и в `<body>`, однако мы рекомендуем ставить его выше, чтобы другие скрипты, в случае их поломки, не повлияли на работу нашего кода.

2. Через GTM (**не рекомендуется**, т.к. код, установленный через GTM, блокируется блокировщиками рекламы, что приводит к потере конверсий от 30% до 50%):

- Если вы все же решили устанавливать наш код через GTM, необходимо выбрать триггер активации «Page View» или «DOM Ready». Триггер «Window Load» не подходит, так как другие скрипты, которые будут загружаться у пользователя могут заблокировать загрузку нашего скрипта.

```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///***.***.com/"; // точный домен для интеграции будет вам передан
    ↪ вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

3.13.4 Шаг 3. Настройка дополнительных параметров

По умолчанию трекинг-ссылка Adv.Cake выглядит так:

```
?utm_source=advcake&utm_medium=cpa&utm_campaign={partner}&utm_content={webmaster}&
→advcake_params={click_id}
```

Предупреждение: Если вы не можете использовать такой формат ссылок, сообщите нам.

При переходе на сайт пользователю выставляются две куки кодом Adv.Cake:

1. advcake_track_id – случайная последовательность из 32-х букв и цифр.
2. advcake_track_url – полная ссылка переход на сайт с utm_source=advcake.

3.13.5 Шаг 4. Настройка postback

Пример отправки заказа/заявки:

Предупреждение: Необходимо отправлять все заказы с любыми источниками.

```
https://api.ekacvda.com/postback/siteru?id=1411551&totalPrice=10141.14&coupo
n=A270&clientType=new&trackId=14511aca1b09e91f&url=https%3a%2f%2fsite.ru%2f%3
futm_source%3dadvcake%26utm_campaign%3dcpa%26advcake_params%3dafacaafa15aac&e
mail=abc@gmail.com&leadName=english_course&timeLastChanged=2021-01-01%2021%3A
30%3A10&status=1&product=credit_card
```

Подсказка: Вместо siteru подставьте название вашего сайта, например: <https://api.ekacvda.com/postback/exampleru> (без точки).

Примечание: Отправлять заказ/заявку желательно POST-запросом в формате JSON (например, с помощью XHR) т.к. длина GET-запроса ограничена 2048 символами, из-за чего часть данных может не дойти.

3.13.5.1 Расшифровка полей постбека

id	Номер заявки/лида/заказа
totalP	Стоимость заявки. Если понятие стоимости заявки отсутствует, данное поле можно исключить
coupon	Желательно передавать при отдельной ставке для заказов с купоном
client	Тип клиента: “новый” (new), “старый” (old). Может понадобиться для разделения ставок для новых/старых клиентов, а также для аналитики
tracki	Значение куки advcake_track_id на момент совершения заказа (см. шаг 3)
url	Значение куки advcake_track_url на момент совершения заказа (см. шаг 3)
email	Почта клиента (можно передавать телефон или зашифрованный идентификатор)
dateCr	Дата создания заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
timeLa	Дата изменения статуса заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
status	Статус заявки/лида/заказа: 1 – Заявка в обработке; 2 – Заявка подтверждена; 3 – Заявка отменена.
produc	Соответствующее значение под конкретный продукт, например: credit_card - Кредитная карта; РКО - Расчетно-кассовое обслуживание.
leadNa	При необходимости (например, наличие двух или более категорий продукта) передается категория продукта: Например, для продукта «РКО», могут быть указаны следующие категории: РКО-ИП - Расчетно-кассовое обслуживание для ИП; РКО-000 - Расчетно-кассовое обслуживание для ООО.

Примечание: Если у вас есть дополнительные идентификаторы по заказу, создайте дополнительные поля и сообщите нам их названия, а также параметры, которые в них будут передаваться.

3.13.6 Шаг 5. Установка кода, формирующего DATA-объект на страницы

Установите предоставленные ниже коды на соответствующие страницы вашего сайта:

3.13.6.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    }
  });
</script>
```

3.13.6.2 2. Страница просмотра лида

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentLead: {
      product: 'a12e', // статический id (курса, продукта и т.п.)
      name: "Регистрация",
      price: 123 // цена - передается если есть
    }
  });
</script>

```

3.13.6.3 3. Страница начала создания лида

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentLead: {
      leadid: 'ab144', // статический id (курса, продукта и т.п.)
      name: "Кредит",
      price: 123 // цена - передается если есть
    }
  });
</script>

```

3.13.6.4 4. Страница созданного лида

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 6,
    user: {
      email: 'md5(user@email.ru)', // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
      type: 'old' // old, new - опционально, передавать пустое значение в
      ↪кавычках (''), если тип пользователя неизвестен
    },
    leadInfo: {
      id: 'ac31004', // id заказа

```

(continues on next page)

(продолжение с предыдущей страницы)

```
        leadid: '123', // статический id (курса, продукта и т.п.)
        name: 'Кредит',
        totalPrice: 4800,
        coupon: 'HELLO'
    }
});
</script>
```

3.13.6.5 5. Страница списка лидов

```
<script>
    window.advcake_data = window.advcake_data || [];
    window.advcake_data.push({
        pageType: 5,
        user: {
            email: 'md5(user@email.ru)' // email в виде хеша или уникальный
        ←идентификатор, если не передается, данное поле можно оставить пустым
        },
        leads: [{
            leadid: 'ac156', // статический id (курса, продукта и т.п.)
            name: 'Дебетовая карта',
            totalPrice: 15800
        }, {
            leadid: 'ab144',
            name: 'Дебетовая карта',
            totalPrice: 4800
        }
    ]
    });
</script>
```

3.14 Пример инструкции XML Lead для финансовых офферов

3.14.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:
 - Cookie - обезличенный идентификатор пользователя;
 - Cookie - обезличенный идентификатор сессии;
 - Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).
2. Параметры браузера:
 - User-Agent;

- Размер экрана;
- Размер окна браузера;
- Открытие страницы в iframe;
- URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

При интеграции через XML вы создаете XML-файл определенной структуры, в который необходимо передавать данные о заказах, совершенных пользователями на вашем сайте, пришедшими по меткам CPA-канала. Каждый час к этому файлу обращается сервер Adv.Cake и выгружает из него данные по новым действиям в статистику.

Мы не собираем никакой личной информации, мы используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

1. *Предоставление доступа к рекламному кабинету;*
2. *Установка JS-кода на страницы сайта;*
3. *Настройка дополнительных параметров;*
4. *Настройка автоматической сверки с помощью XML-выгрузки;*
5. *Установка кода, формирующего DATA-объект на страницы.*

3.14.2 Шаг 1. Предоставление доступа к рекламному кабинету

Первым делом вам необходимо предоставить доступ к рекламному кабинету Google Analytics или Яндекс.Метрика:

Если вы используете Google Analytics, предоставьте доступ для аккаунтов analytics@advcake.com и advcake@arcane-splicer-263114.iam.gserviceaccount.com (аккаунт для запроса данных из Google Analytics по API);

Если вы используете Яндекс.Метрику, предоставьте доступ для аккаунта metrika@advcake.com.

Подсказка: Если вы не можете предоставить доступ к этим сервисам, переходите к следующему шагу.

3.14.3 Шаг 2. Установка JS-кода на страницы сайта

Установить JS-код можно двумя способами:

1. Напрямую на сайт (**рекомендуется**):

- Код из примера ниже можно поставить как в <head>, так и в <body>, однако мы рекомендуем ставить его выше, чтобы другие скрипты, в случае их поломки, не повлияли на работу нашего кода.

2. Через GTM (**не рекомендуется**, т.к. код, установленный через GTM, блокируется блокировщиками рекламы, что приводит к потере конверсий от 30% до 50%):

- Если вы все же решили устанавливать наш код через GTM, необходимо выбрать триггер активации «Page View» или «DOM Ready». Триггер «Window Load» не подходит, так как другие скрипты, которые будут загружаться у пользователя могут заблокировать загрузку нашего скрипта.

```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///**.*.com/"; // точный домен для интеграции будет вам передан
    ↪ вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

3.14.4 Шаг 3. Настройка дополнительных параметров

По умолчанию трекинг-ссылка Adv.Cake выглядит так:

```
?utm_source=advcake&utm_medium=cpa&utm_campaign={partner}&utm_content={webmaster}&
↪ advcake_params={click_id}
```

Предупреждение: Если вы не можете использовать такой формат ссылок, сообщите нам.

При переходе на сайт пользователю выставляются две куки кодом Adv.Cake:

1. advcake_track_id – случайная последовательность из 32-х букв и цифр.
2. advcake_track_url – полная ссылка переход на сайт с utm_source=advcake.

3.14.5 Шаг 4. Настройка автоматической сверки с помощью XML-выгрузки

Нажмите, чтобы развернуть пример XML-выгрузки

```
<orders>
  <order>
    <orderId>order00001</orderId>
    <orderPrice>23000</orderPrice>
    <orderStatus>3</orderStatus>
    <orderTrackid>7d12ff1fd3dd5b9a61dc1f42b927264c</orderTrackid>
    <description>Не понравилось качество</description>
    <coupon>DCKAL</coupon>
    <dateCreate>2015-10-14 12:04:00</dateCreate>
    <dateLastChange>2015-10-15 17:00:00</dateLastChange>
    <url>http://site.ru/?utm_source=advcake&advcake=1&utm_medium=cpa&utm_
    ↪campaign=admitad&utm_content=263116&utm_term=2</url>
  </order>
  <order>
    <orderId>order00002</orderId>
    <orderPrice>51000</orderPrice>
    <orderStatus>2</orderStatus>
    <orderTrackid>7d12ff1fd3dd5b9a61dc1f42b927264c</orderTrackid>
    <description>отказ</description>
    <coupon/>
    <dateCreate>2015-10-13 10:36:00</dateCreate>
    <dateLastChange>2015-10-15 17:03:00</dateLastChange>
    <url>http://site.ru/?utm_source=advcake&utm_medium=cpa&utm_campaign=admitad&utm_
    ↪content=263116&advcake_params=1</url>
  </order>
</orders>
```

Каждый час мы будем забирать из XML-выгрузки информацию о новых заказах/заявках и обновлять статусы по старым.

При передаче GET-параметров ?from=2019-01-01&to=2019-02-01 выдаются заказы за указанный период по дате создания заказа.

3.14.5.1 Расшифровка полей выгрузки

orde	Номер заявки/лида/заказа
orde	Стоимость заявки. Если понятие стоимости заявки отсутствует, данное поле можно исключить
orde	Статус заявки. Данное поле можно передавать двумя способами: Согласно нашим внутренним идентификаторам: – 1 – Заявка в обработке; – 2 – Заявка подтверждена; – 3 – Заявка отменена. Согласно вашим внутренним идентификаторам – в таком случае нам потребуется описание каждого статуса, например: – Принят в обработку – транслируем как статус 1; – Подтвержден – транслируем как статус 2; – Обработан – транслируем как статус 3.
desc	Причина отмены заявки (передавать данное поле желательно, но не обязательно)
coup	Использованный в заказе купон/промокод, если такая опция используется. Если такой опции нет, данное поле можно исключить
clie	Тип клиента: “новый”, “старый”. Может понадобиться для разделения ставок для новых/старых клиентов, а также для аналитики (передавать данное поле желательно, но не обязательно)
orde	Значение куки advcake_track_id на момент совершения заказа (см. шаг 3)
date	Дата создания заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
date	Дата изменения статуса заказа в формате Y-m-d H:i:s (например, 2022-09-08 17:16:25)
url	Значение куки advcake_track_url на момент совершения заказа (см. шаг 3)
prod	Соответствующее значение под конкретный продукт, например: credit_card - Кредитная карта; PKO - Расчетно-кассовое обслуживание.
lead	При необходимости (например, наличие двух или более категорий продукта) передается категория продукта: Например, для продукта «РКО», могут быть указаны следующие категории: PKO-IP - Расчетно-кассовое обслуживание для ИП; PKO-000 - Расчетно-кассовое обслуживание для ООО.

Примечание: Если у вас есть дополнительные идентификаторы по заказу, создайте дополнительные поля и сообщите нам их названия, а также параметры, которые в них будут передаваться.

3.14.6 Шаг 5. Установка кода, формирующего DATA-объект на страницы

Установите предоставленные ниже коды на соответствующие страницы вашего сайта:

3.14.6.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    }
  });
</script>
```

←идентификатор, если не передается, данное поле можно оставить пустым

3.14.6.2 2. Страница просмотра лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    },
    currentLead: {
      name: "Регистрация"
    }
  });
</script>
```

←идентификатор, если не передается, данное поле можно оставить пустым

3.14.6.3 3. Страница начала создания лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    },
    currentLead: {
      name: 'Кредит'
    }
  });
</script>
```

←идентификатор, если не передается, данное поле можно оставить пустым

3.14.6.4 4. Страница созданного лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 4,
    user: {
      email: 'md5(user@email.ru)', // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
      type: 'old' // old, new - опционально
    },
    leadInfo: {
      id: 'ac31004',
      name: 'Страховка'
    }
  });
</script>
```

3.14.6.5 5. Страница списка лидов

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 5,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    leads: [{
      name: 'Регистрация',
    }, {
      name: 'Кредит',
    }
  ]
  });
</script>
```

3.15 Пример инструкции Tilda Ecom

3.15.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:
 - Cookie - обезличенный идентификатор пользователя;

- Cookie - обезличенный идентификатор сессии;
 - Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).
2. Параметры браузера:
- User-Agent;
 - Размер экрана;
 - Размер окна браузера;
 - Открытие страницы в iframe;
 - URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

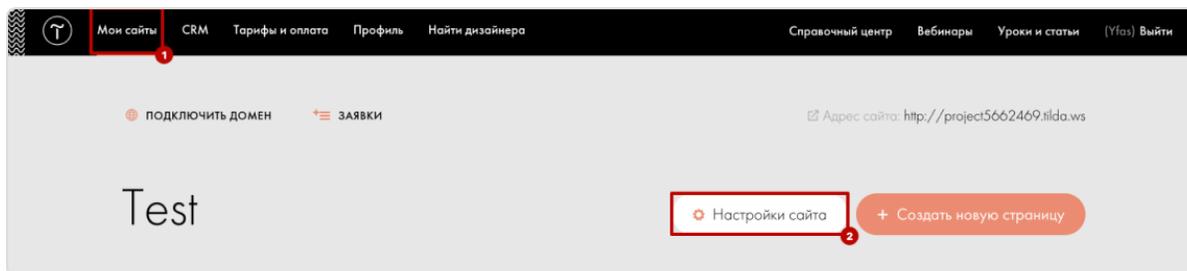
Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

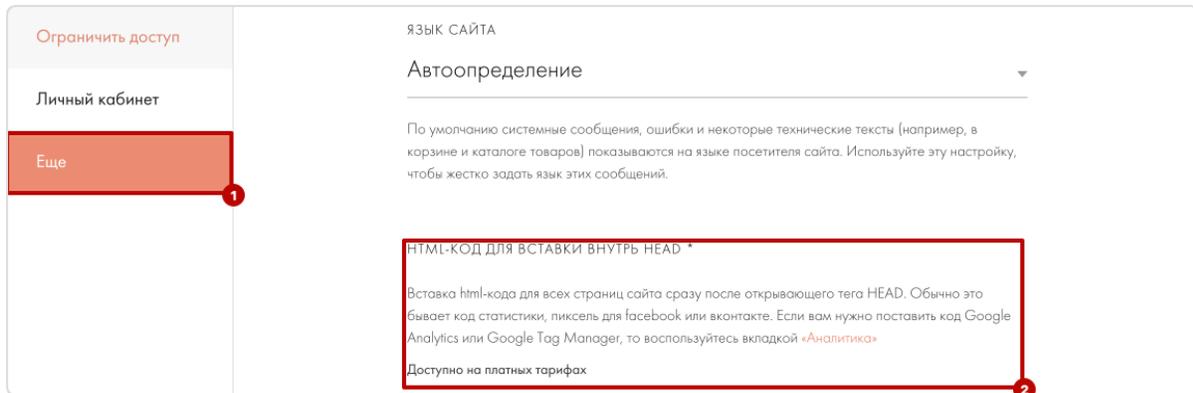
1. Установка JS-кода на страницы сайта;
2. Настройка отправки заказов;
3. Установка кода, формирующего DATA-объект на страницы.

3.15.2 Шаг 1. Установка JS-кода на страницы сайта

1. На вкладке «Мои сайты» перейдите в раздел «Настройки сайта»:



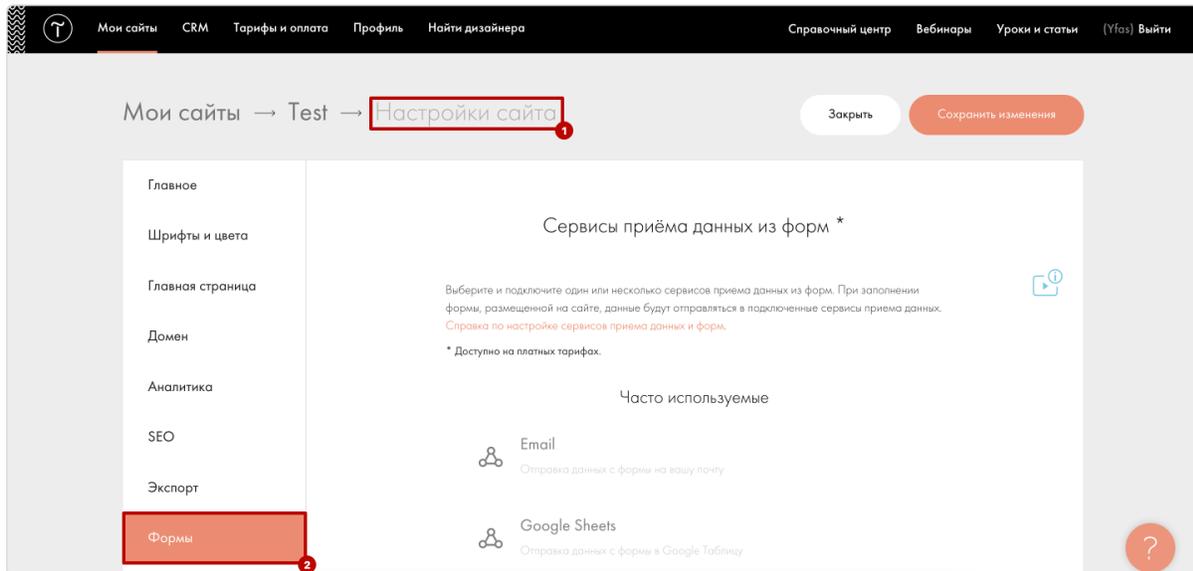
2. Перейдите в раздел «Еще» и вставьте код ниже в поле «HTML-КОД ДЛЯ ВСТАВКИ ВНУТРЬ HEAD»:



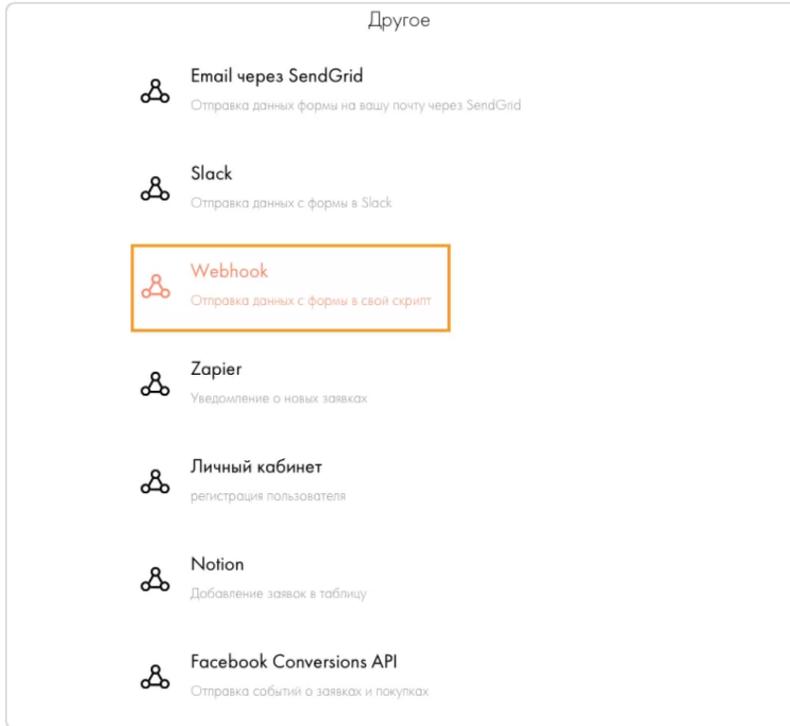
```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///***.***.com/"; // точный домен для интеграции будет вам передан,
    ↪ вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

3.15.3 Шаг 2. Настройка отправки заказов

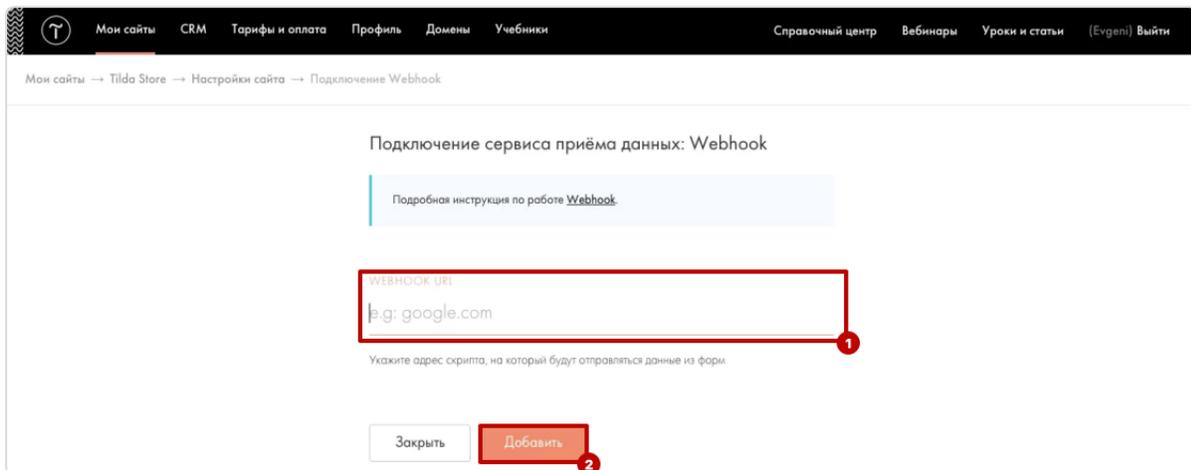
1. В разделе «Настройки сайта» перейдите на вкладку «Формы»:



2. Прокрутите страницу вниз до пункта «Webhook»:



- Вставьте ссылку для отправки постбека, которая была передана вам вместе с этой инструкцией в поле «Webhook URL» (пример ссылки: <https://api.ekacvda.com/postback/siteru>, где siteru – название вашего сайта без точки), после чего нажмите *Добавить*:



Опционально, вы можете добавить приемщик ко всем формам на сайте или добавить приемщик ко всем нужным формам позже вручную:

Подключение сервиса приёма данных: Webhook

 Сервис приема данных Webhook добавлен.

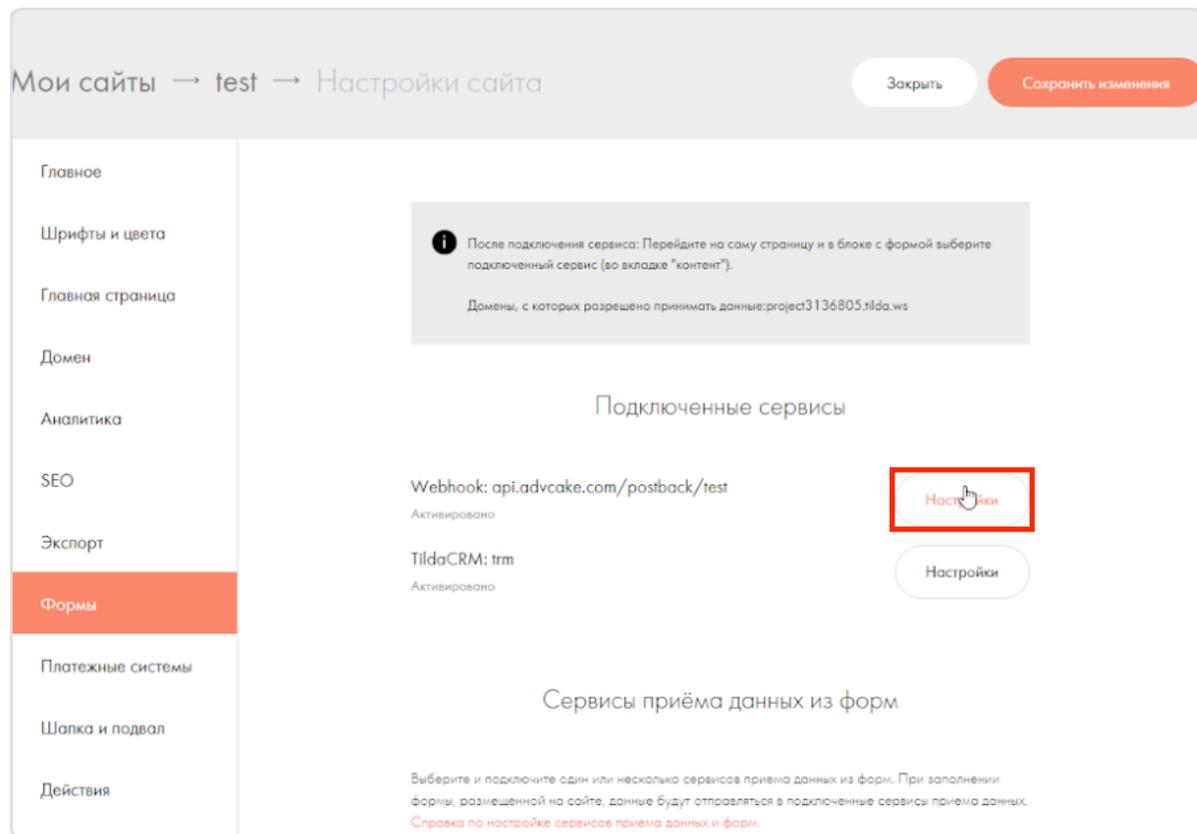
Добавить приемщик данных ко всем формам на сайте?

[Добавить](#)

Для того, чтобы заявки из формы начали отправляться в сервис приема данных, их нужно связать между собой. Для того, чтобы сделать это вручную, откройте страницу, на которой размещен блок с нужной формой. Нажмите на кнопку «Контент» этого блока, отметьте галочкой сервис приема данных и опубликуйте страницу.

[Закреть](#)

4. Перейдите в настройки Webhook («Настройки сайта» -> «Формы» -> «Подключенные сервисы»):



Мои сайты → test → Настройки сайта

Закреть Сохранить изменения

Главное
Шрифты и цвета
Главная страница
Домен
Аналитика
SEO
Экспорт
Формы
Платежные системы
Шапка и подвал
Действия

После подключения сервиса: Перейдите на саму страницу и в блоке с формой выберите подключенный сервис (во вкладке "контент").
Домены, с которых разрешено принимать данные: project3136805.tilda.ws

Подключенные сервисы

Webhook: api.advcake.com/postback/test
Активировано [Настройки](#)

TildaCRM: trm
Активировано [Настройки](#)

Сервисы приёма данных из форм

Выберите и подключите один или несколько сервисов приема данных из форм. При заполнении формы, размещенной на сайте, данные будут отправляться в подключенные сервисы приема данных.
[Справка по настройке сервисов приема данных и форм.](#)

5. Выберите все дополнительные настройки:

- Посылать Cookies;

- Передавать данные по товарам в заказе – массивом;
- Передавать `externalid` в товарах.

Примечание: Галочка для передачи `externalid` товара доступна только при выборе опции передачи данных о товаре массивом. При ее активации передается внешний код товара из каталога для связи этого товара с внешней системой (например, МойСклад или 1С).

После чего нажмите *Сохранить*.

Сервис приёма данных: Webhook

[Подробная инструкция по настройке отправки данных на Webhook](#)

WEBHOOK URL

`https://api.ekacvda.com/postback/siteru`

Укажите адрес скрипта, на который будут отправляться данные из форм

ДОПОЛНИТЕЛЬНЫЕ НАСТРОЙКИ

- Посылать Cookies
- Передавать данные по товарам в заказе - массивом
- Передавать `externalid` в товарах

НАЗВАНИЕ В СПИСКЕ

`api.ekacvda.com/postback/siteru`

Отображается в списке подключенных сервисов и в настройках формы на Тильде.

3.15.4 Шаг 3. Установка кода, формирующего DATA-объект на страницы

3.15.4.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    }
  });
</script>
```

←идентификатор, если не передается, данное поле можно оставить пустым

3.15.4.2 2. Страница одного товара

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    },
    currentCategory: {
      id: 441,
      name: 'Обувь'
    },
    currentProduct: {
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
  });
</script>
```

←идентификатор, если не передается, данное поле можно оставить пустым

3.15.4.3 3. Страница каталога/категории/подкатегории

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentCategory: {
      id: 441,
      name: 'Обувь'
    },
    products: [{
      id: 001,
      name: 'Черная кеда',
      price: 12000
    }, {
      id: 002,
      name: 'Серый кед',
      price: 13000
    }],
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>

```

3.15.4.4 4. Страница корзины

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 4,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },

```

(continues on next page)

(продолжение с предыдущей страницы)

```
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>
```

3.15.4.5 5. Страница оформления заказа (после корзины и до страницы «Спасибо за заказ»)

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 5,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ←идентификатор, если не передается, данное поле можно оставить пустым
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>
```

3.15.4.6 6. Страница оформленного заказа («Спасибо за заказ»)

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 6,
    user: {
      email: 'md5(user@email.ru)', // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
      type: 'old' // old, new - опционально, передавать пустое значение в
      ↪кавычках (''), если тип пользователя неизвестен
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]},
  orderInfo: {
    id: 'ac31004', // уникальный идентификатор заказа, используемый при сверке
    totalPrice: 34647, // сумму заказа передавать без учета доставки
    coupon: 'SKIDKA100' // опционально, передавать пустое значение в кавычках ('
    ↪'), если заказ без купона или данный функционал отсутствует на сайте
  }
});
</script>

```

3.15.4.7 7. Страница списка товаров (страница поиска, страница акционных товаров и т.п.)

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 7,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    products: [{
      id: 001,
      name: 'Черная кеда',
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]}

```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }, {
      id: 002,
      name: 'Серый кед',
      categoryId: 411,
      categoryName: 'Обувь'
    }],
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>
```

Предупреждение: ID товаров должны быть всегда одинаковыми, т.е. у каждого товара свой ID.

3.16 Пример инструкции Tilda Lead

3.16.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:
 - Cookie - обезличенный идентификатор пользователя;
 - Cookie - обезличенный идентификатор сессии;
 - Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).
2. Параметры браузера:
 - User-Agent;
 - Размер экрана;
 - Размер окна браузера;

- Открытие страницы в iframe;
- URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

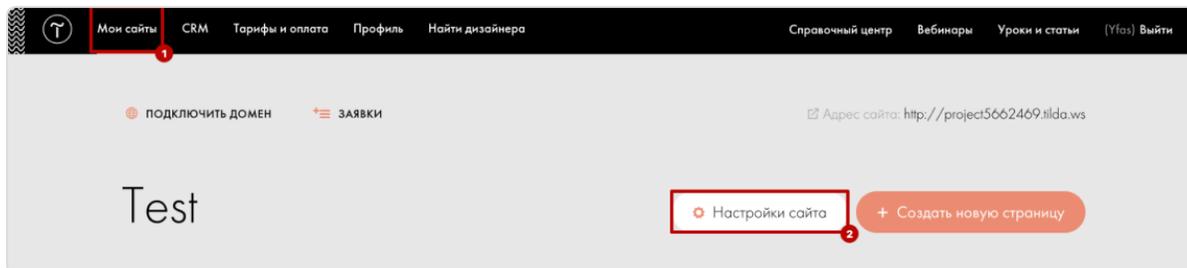
Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

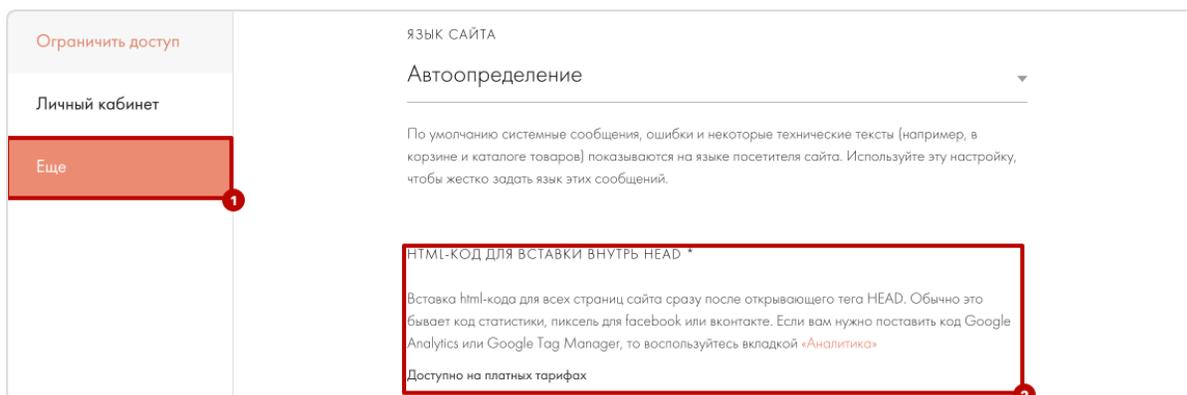
1. Установка JS-кода на страницы сайта;
2. Настройка отправки заказов;
3. Установка кода, формирующего DATA-объект на страницы.

3.16.2 Шаг 1. Установка JS-кода на страницы сайта

1. На вкладке «Мои сайты» перейдите в раздел «Настройки сайта»:



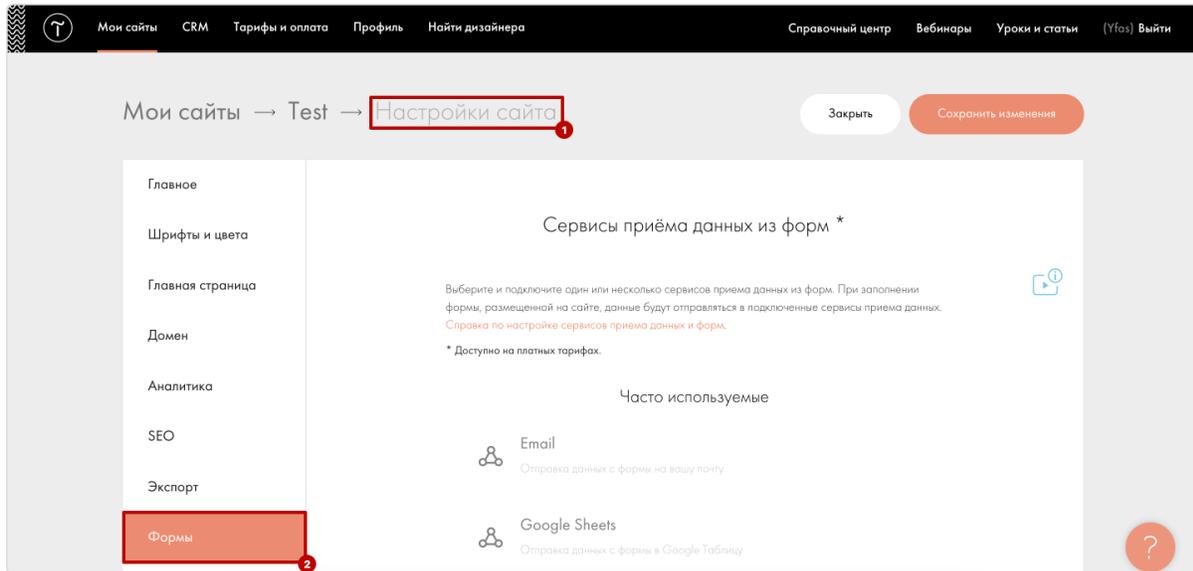
2. Перейдите в раздел «Еще» и вставьте код ниже в поле «HTML-КОД ДЛЯ ВСТАВКИ ВНУТРЬ HEAD»:



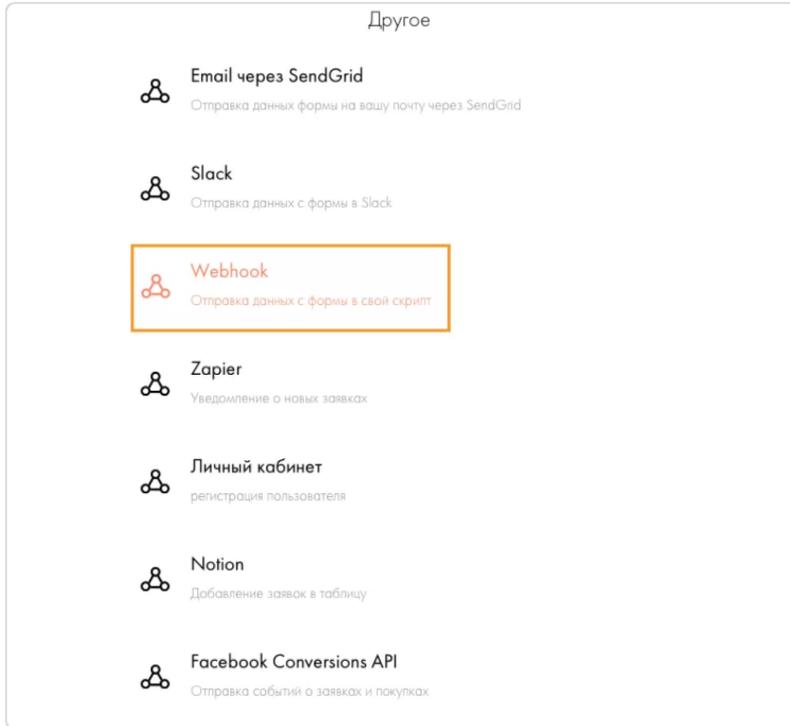
```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///**.*.com/"; // точный домен для интеграции будет вам передан,
    ↪ вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

3.16.3 Шаг 2. Настройка отправки заказов

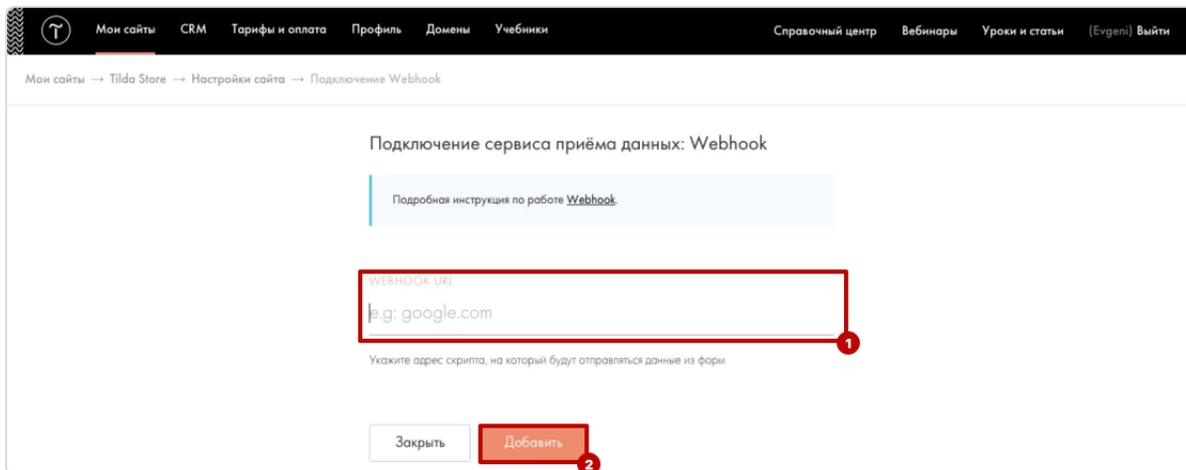
1. В разделе «Настройки сайта» перейдите на вкладку «Формы»:



2. Прокрутите страницу вниз до пункта «Webhook»:



- Вставьте ссылку для отправки постбека, которая была передана вам вместе с этой инструкцией в поле «Webhook URL» (пример ссылки: <https://api.ekacvda.com/postback/siteru>, где siteru – название вашего сайта без точки), после чего нажмите *Добавить*:



Опционально, вы можете добавить приемщик ко всем формам на сайте или добавить приемщик ко всем нужным формам позже вручную:

Подключение сервиса приёма данных: Webhook

 Сервис приема данных Webhook добавлен.

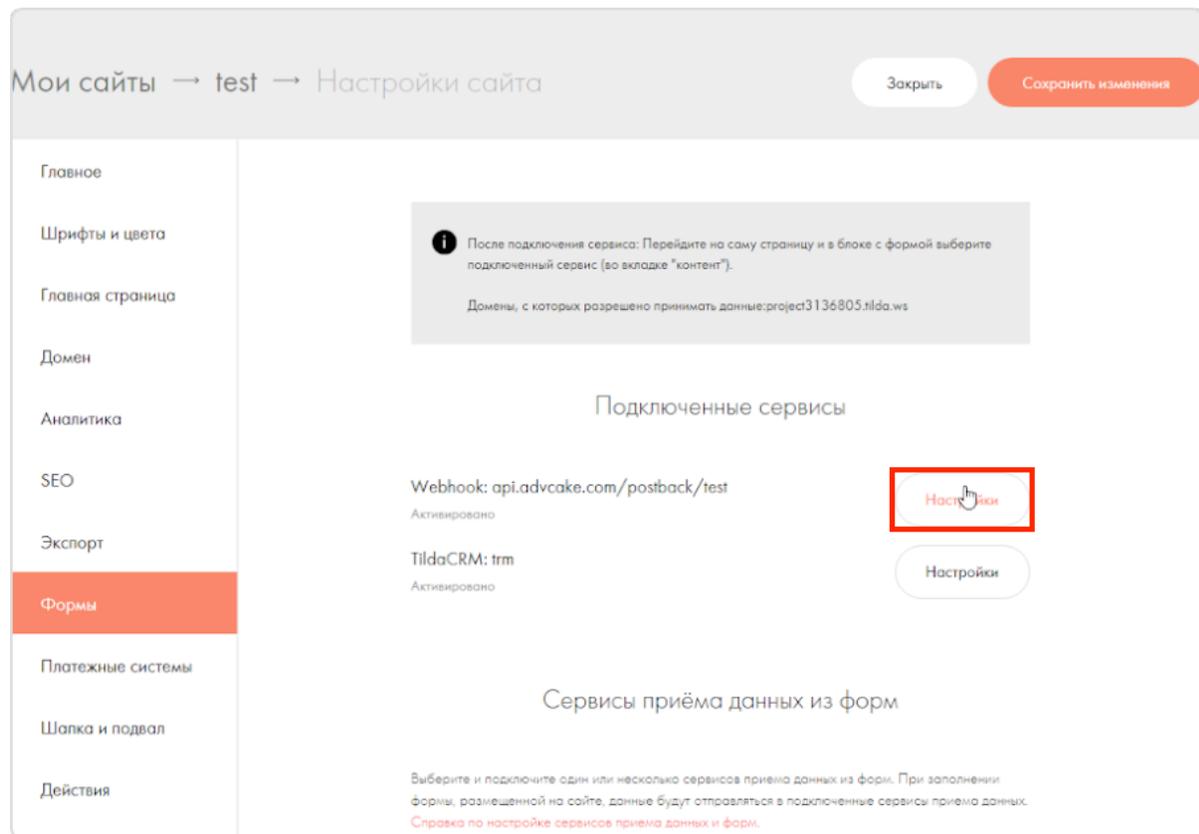
Добавить приемщик данных ко всем формам на сайте?

[Добавить](#)

Для того, чтобы заявки из формы начали отправляться в сервис приема данных, их нужно связать между собой. Для того, чтобы сделать это вручную, откройте страницу, на которой размещен блок с нужной формой. Нажмите на кнопку «Контент» этого блока, отметьте галочкой сервис приема данных и опубликуйте страницу.

[Закреть](#)

4. Перейдите в настройки Webhook («Настройки сайта» -> «Формы» -> «Подключенные сервисы»):



Мои сайты → test → Настройки сайта

Закреть Сохранить изменения

Главное

Шрифты и цвета

Главная страница

Домен

Аналитика

SEO

Экспорт

Формы

Платежные системы

Шапка и подвал

Действия

Подключенные сервисы

Webhook: api.advcake.com/postback/test
Активировано

TildaCRM: trm
Активировано

Настройки

Настройки

Сервисы приёма данных из форм

Выберите и подключите один или несколько сервисов приема данных из форм. При заполнении формы, размещенной на сайте, данные будут отправляться в подключенные сервисы приема данных.
[Справка по настройке сервисов приема данных и форм.](#)

5. Выберите все дополнительные настройки:

- Посылать Cookies;

- Передавать данные по товарам в заказе – массивом;
- Передавать `externalid` в товарах.

Примечание: Галочка для передачи `externalid` товара доступна только при выборе опции передачи данных о товаре массивом. При ее активации передается внешний код товара из каталога для связи этого товара с внешней системой (например, МойСклад или 1С).

После чего нажмите *Сохранить*.

Сервис приёма данных: Webhook

[Подробная инструкция по настройке отправки данных на Webhook](#)

WEBHOOK URL

Укажите адрес скрипта, на который будут отправляться данные из форм

ДОПОЛНИТЕЛЬНЫЕ НАСТРОЙКИ

- Посылать Cookies
- Передавать данные по товарам в заказе - массивом
- Передавать `externalid` в товарах

НАЗВАНИЕ В СПИСКЕ

Отображается в списке подключенных сервисов и в настройках формы на Тильде.

3.16.4 Шаг 3. Установка кода, формирующего DATA-объект на страницы

3.16.4.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    ↪идентификатор, если не передается, данное поле можно оставить пустым
    }
  });
</script>
```

3.16.4.2 2. Страница просмотра лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentLead: {
      name: "Регистрация"
    }
  });
</script>
```

3.16.4.3 3. Страница начала создания лида

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentLead: {
      name: 'Кредит'
    }
  });
</script>
```

3.16.4.4 4. Страница созданного лида

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 6,
    user: {
      email: 'md5(user@email.ru)', // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
      type: 'old' // old, new - опционально, передавать пустое значение в
      ↪кавычках (''), если тип пользователя неизвестен
    },
    leadInfo: {
      id: 'ac31004',
      name: 'Кредит',
      totalPrice: 4800
    }
  });
</script>

```

3.16.4.5 5. Страница списка лидов

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 5,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    leads: [{
      id: 'ac31004',
      name: 'Кредит',
      totalPrice: 4800
    }, {
      id: 'ab31005',
      name: 'Дебетовая карта',
      totalPrice: 5000
    }
  ]
  });
</script>

```

3.17 Пример инструкции Webflow Ecom

3.17.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:
 - Cookie - обезличенный идентификатор пользователя;
 - Cookie - обезличенный идентификатор сессии;
 - Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).
2. Параметры браузера:
 - User-Agent;
 - Размер экрана;
 - Размер окна браузера;
 - Открытие страницы в iframe;
 - URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

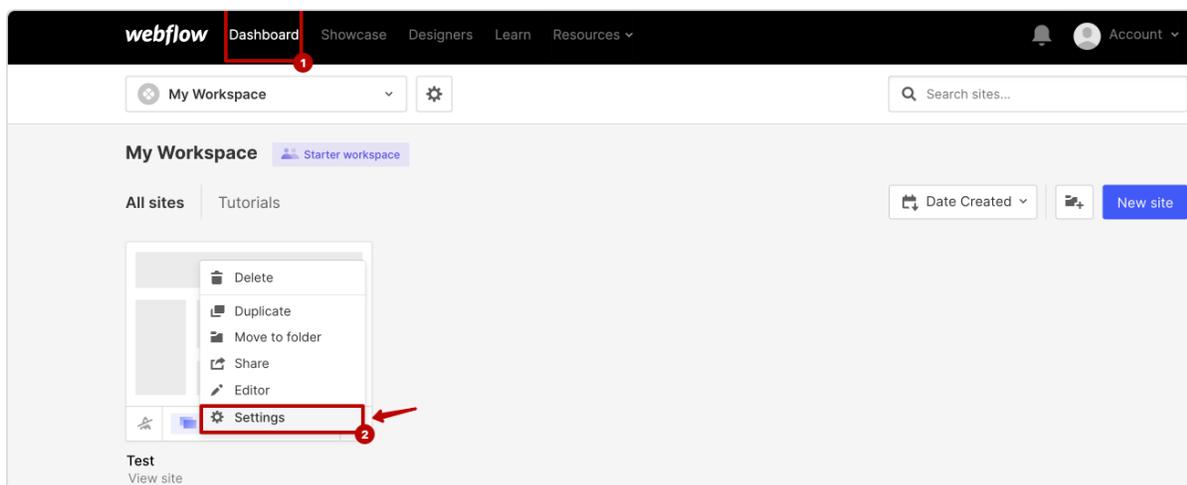
Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

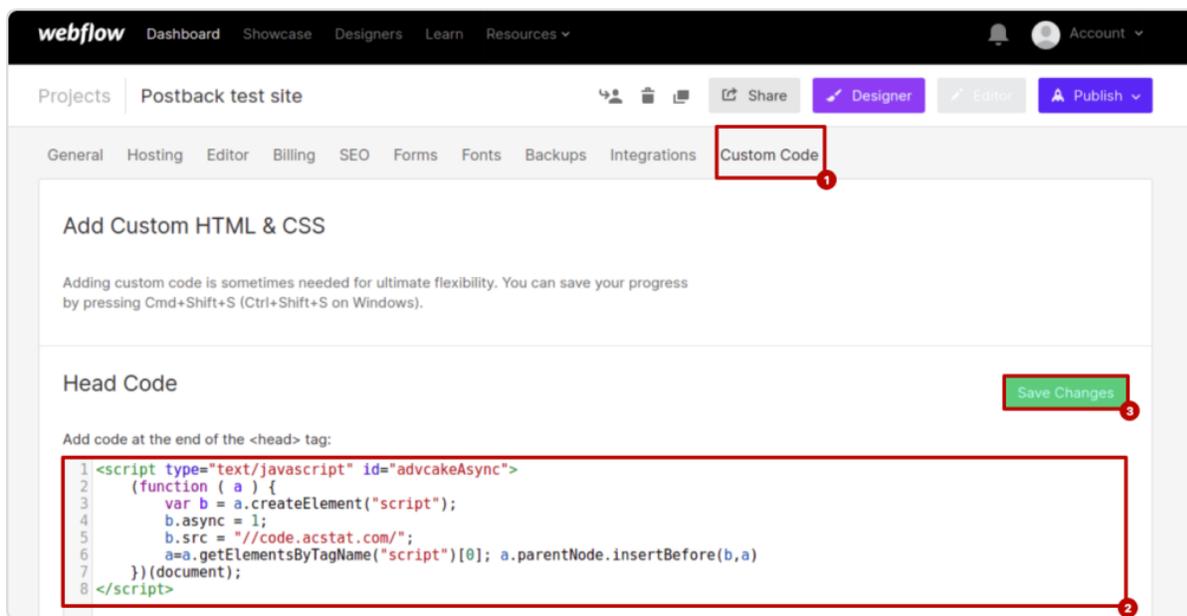
1. *Установка JS-кода на страницы сайта;*
2. *Настройка отправки заказов;*
3. *Установка кода, формирующего DATA-объект на страницы.*

3.17.2 Шаг 1. Установка JS-кода на страницы сайта

1. В панели администратора Webflow на вкладке «Dashboard» (1), выберите ваш сайт, нажмите на ... и перейдите в настройки сайта (2):



2. В настройках сайта перейдите на вкладку «Custom Code» (1) и вставьте код ниже в поле «Head Code» (2), после чего нажмите *Save Changes* (3):



```
<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///**.*.com/"; // точный домен для интеграции будет вам передан
    a = a.getElementsByTagName("script")[0]; a.parentNode.insertBefore(b,a)
  })(document);
</script>
```

↪ вместе с кастомной инструкцией

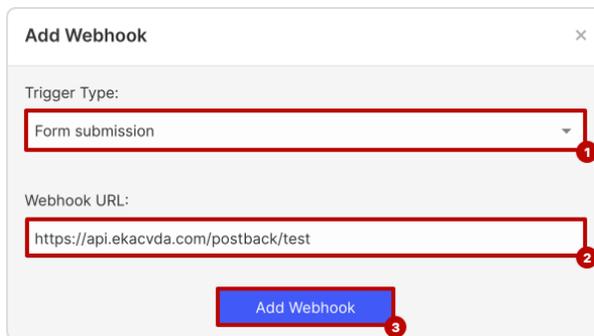
(continues on next page)

(продолжение с предыдущей страницы)

```
a.parentNode.insertBefore(b, a)
})(document);
</script>
```

3.17.3 Шаг 2. Настройка отправки заказов

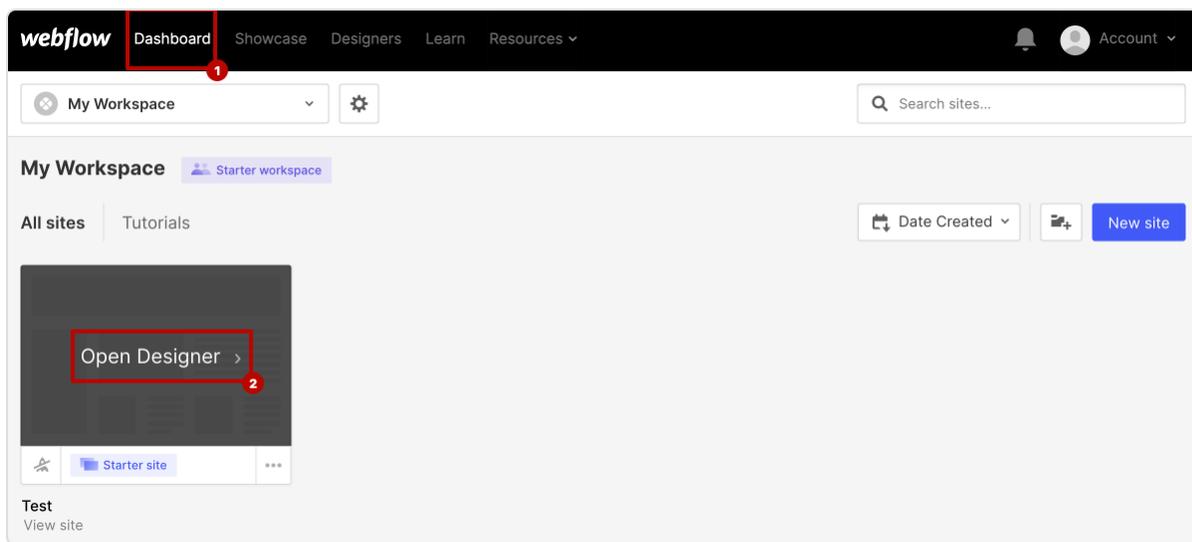
1. В настройках сайта перейдите на вкладку «Integrations», прокрутите страницу вниз до пункта «Webhooks» и нажмите *Add Webhook*:
2. Установите «Trigger Type» = «Form submission» (1);
3. В поле «Webhook URL» укажите ссылку для отправки постбека, которая была передана вам вместе с этой инструкцией (пример ссылки – <https://api.ekacvda.com/postback/test>) (2), после чего нажмите *Add Webhook* (3):



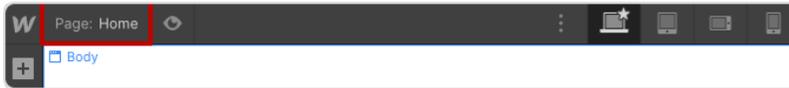
The screenshot shows a modal window titled "Add Webhook". It has a close button (X) in the top right corner. Below the title, there are three main sections: "Trigger Type:" with a dropdown menu showing "Form submission" (marked with a red box and number 1), "Webhook URL:" with a text input field containing "https://api.ekacvda.com/postback/test" (marked with a red box and number 2), and a blue "Add Webhook" button at the bottom (marked with a red box and number 3).

3.17.4 Шаг 3. Установка кода, формирующего DATA-объект на страницы

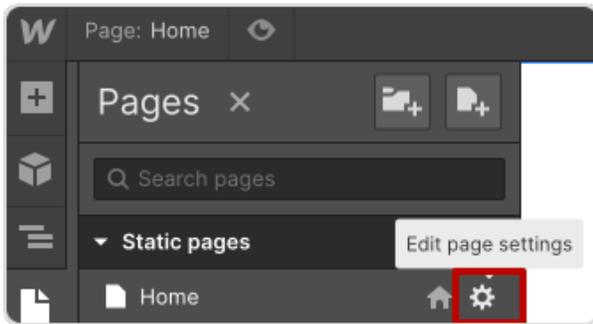
1. В панели администратора Webflow на вкладке «Dashboard» (1) наведите курсор на ваш сайт и откройте редактор сайта (2):



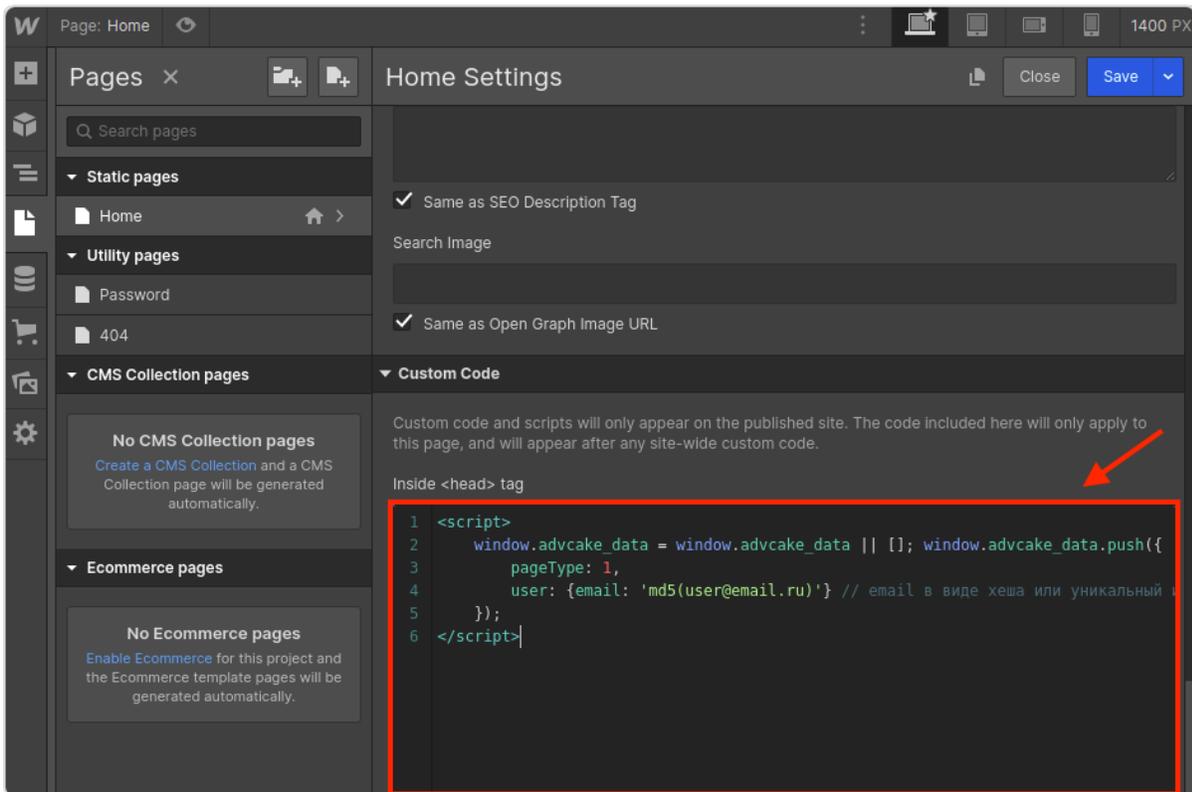
2. В левом верхнем углу нажмите на название страницы:



3. В списке страниц наведите курсор на строку с названием страницы, после чего, нажмите на иконку настроек:



4. В появившемся окне прокрутите страницу вниз до пункта «Custom Code» и вставьте коды ниже в поле «Inside <head> tag»:



3.17.4.1 1. Главная страница

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    }
  });
</script>
```

3.17.4.2 2. Страница одного товара

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentCategory: {
      id: 441,
      name: 'Обувь'
    },
    currentProduct: {
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
  });
</script>
```

3.17.4.3 3. Страница каталога/категории/подкатегории

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 3,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    currentCategory: {
      id: 441,
      name: 'Обувь'
    },
    products: [{
      id: 001,
      name: 'Черная кеда',
      price: 12000
    }, {
      id: 002,
      name: 'Серый кед',
      price: 13000
    }],
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>

```

3.17.4.4 4. Страница корзины

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 4,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },

```

(continues on next page)

(продолжение с предыдущей страницы)

```
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>
```

3.17.4.5 5. Страница оформления заказа (после корзины и до страницы «Спасибо за заказ»)

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 5,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ←идентификатор, если не передается, данное поле можно оставить пустым
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>
```

3.17.4.6 6. Страница оформленного заказа («Спасибо за заказ»)

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 6,
    user: {
      email: 'md5(user@email.ru)', // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
      type: 'old' // old, new - опционально, передавать пустое значение в
      ↪кавычках (''), если тип пользователя неизвестен
    },
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]},
  orderInfo: {
    id: 'ac31004', // уникальный идентификатор заказа, используемый при сверке
    totalPrice: 34647, // сумму заказа передавать без учета доставки
    coupon: 'SKIDKA100' // опционально, передавать пустое значение в кавычках ('
    ↪'), если заказ без купона или данный функционал отсутствует на сайте
  }
});
</script>

```

3.17.4.7 7. Страница списка товаров (страница поиска, страница акционных товаров и т.п.)

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 7,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
      ↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    products: [{
      id: 001,
      name: 'Черная кеда',
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});

```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }, {
      id: 002,
      name: 'Серый кед',
      categoryId: 411,
      categoryName: 'Обувь'
    }],
    basketProducts: [{
      id: 3247,
      name: 'Кроссовок Левый',
      price: 30000,
      quantity: 1,
      categoryId: 411,
      categoryName: 'Обувь'
    }, {
      id: 3247,
      name: 'Кроссовок Правый',
      price: 30000,
      quantity: 2,
      categoryId: 411,
      categoryName: 'Обувь'
    }
  ]
});
</script>
```

Предупреждение: ID товаров должны быть всегда одинаковыми, т.е. у каждого товара свой ID.

3.18 Пример инструкции Webflow Lead

3.18.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:
 - Cookie - обезличенный идентификатор пользователя;
 - Cookie - обезличенный идентификатор сессии;
 - Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).
2. Параметры браузера:
 - User-Agent;
 - Размер экрана;
 - Размер окна браузера;

- Открытие страницы в iframe;
- URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

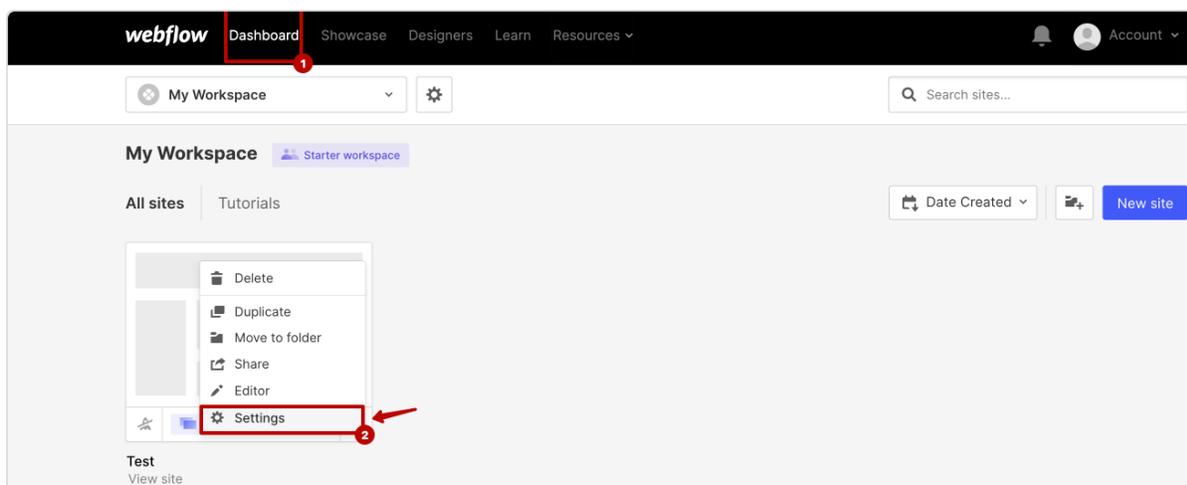
Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

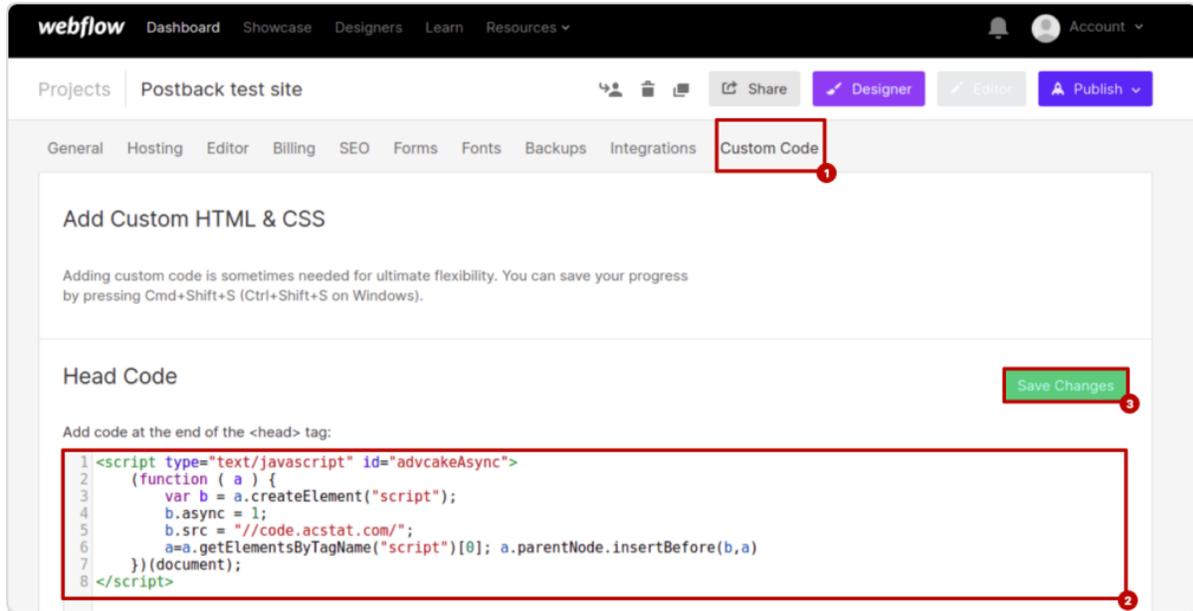
1. Установка JS-кода на страницы сайта;
2. Настройка отправки заказов;
3. Установка кода, формирующего DATA-объект на страницы.

3.18.2 Шаг 1. Установка JS-кода на страницы сайта

1. В панели администратора Webflow на вкладке «Dashboard» (1), выберите ваш сайт, нажмите кнопку «Три точки» и перейдите в настройки сайта (2):



2. В настройках сайта перейдите на вкладку «Custom Code» (1) и вставьте код ниже в поле «Head Code» (2), после чего нажмите *Save Changes* (3):

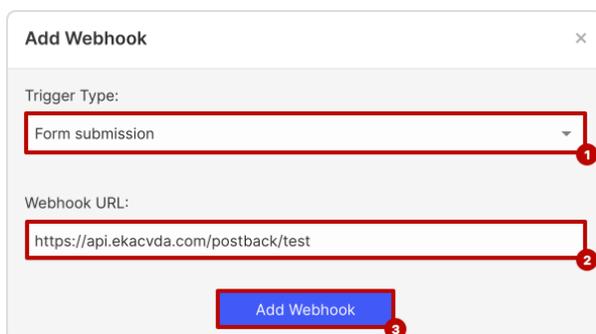


```

<script type="text/javascript" id="advcakeAsync">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///***.***.com/"; // точный домен для интеграции будет вам передан
    ↪ вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
    
```

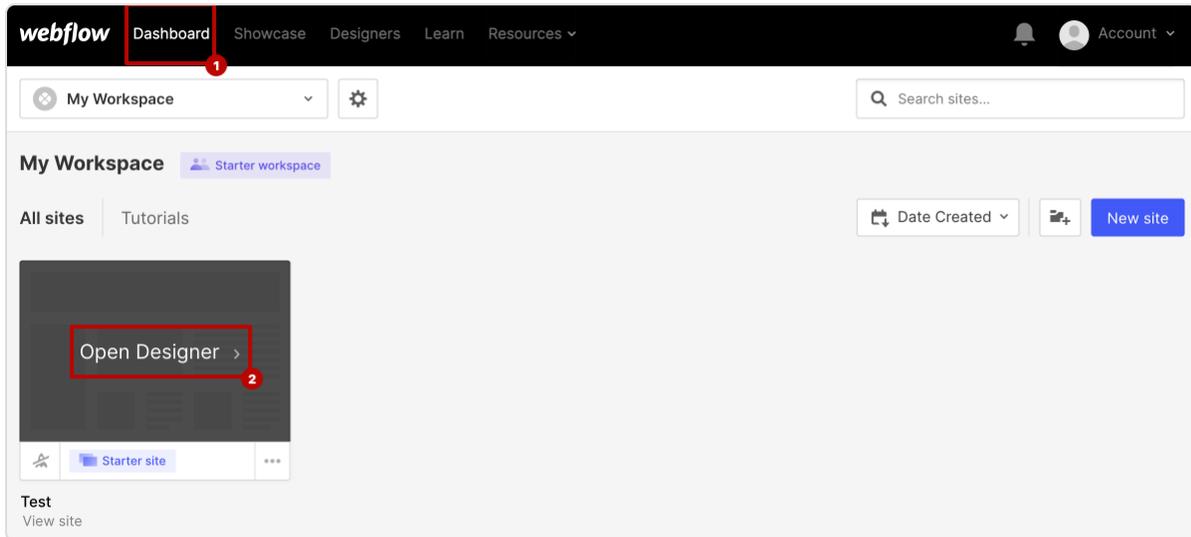
3.18.3 Шаг 2. Настройка отправки заказов

1. В настройках сайта перейдите на вкладку «Integrations», прокрутите страницу вниз до пункта «Webhooks» и нажмите *Add Webhook*:
2. Установите «Trigger Type» = «Form submission» (1);
3. В поле «Webhook URL» укажите ссылку для отправки постбека, которая была передана вам вместе с этой инструкцией (пример ссылки – <https://api.ekacvda.com/postback/test>) (2), после чего нажмите *Add Webhook* (3):



3.18.4 Шаг 3. Установка кода, формирующего DATA-объект на страницы

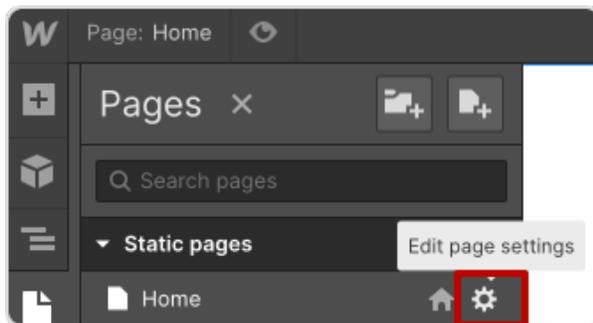
1. В панели администратора Webflow на вкладке «Dashboard» (1) наведите курсор на ваш сайт и откройте редактор сайта (2):



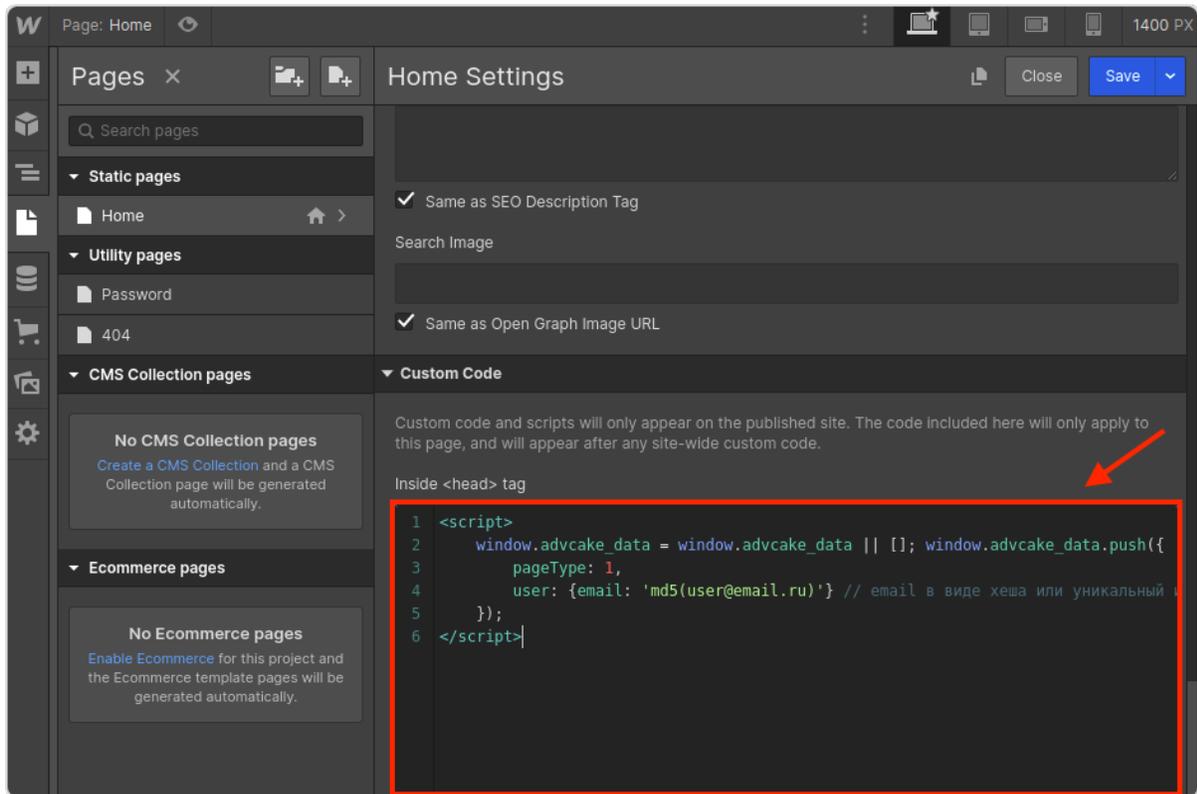
2. В левом верхнем углу нажмите на название страницы:



3. В списке страниц наведите курсор на строку с названием страницы, после чего, нажмите на иконку настроек:



4. В появившемся окне прокрутите страницу вниз до пункта «Custom Code» и вставьте коды ниже в поле «Inside <head> tag»:



3.18.4.1 1. Главная страница

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 1,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    }
  });
</script>

```

←идентификатор, если не передается, данное поле можно оставить пустым

3.18.4.2 2. Страница просмотра лида

```

<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 2,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный
    },
  },

```

←идентификатор, если не передается, данное поле можно оставить пустым

(continues on next page)

(продолжение с предыдущей страницы)

```

        currentLead: {
            name: "Регистрация"
        }
    });
</script>

```

3.18.4.3 3. Страница начала создания лида

```

<script>
    window.advcake_data = window.advcake_data || [];
    window.advcake_data.push({
        pageType: 3,
        user: {
            email: 'md5(user@email.ru)' // email в виде хеша или уникальный
        ↪идентификатор, если не передается, данное поле можно оставить пустым
        },
        currentLead: {
            name: 'Кредит'
        }
    });
</script>

```

3.18.4.4 4. Страница созданного лида

```

<script>
    window.advcake_data = window.advcake_data || [];
    window.advcake_data.push({
        pageType: 6,
        user: {
            email: 'md5(user@email.ru)', // email в виде хеша или уникальный
        ↪идентификатор, если не передается, данное поле можно оставить пустым
            type: 'old' // old, new - опционально, передавать пустое значение в
        ↪кавычках (''), если тип пользователя неизвестен
        },
        leadInfo: {
            id: 'ac31004',
            name: 'Кредит',
            totalPrice: 4800
        }
    });
</script>

```

3.18.4.5 5. Страница списка лидов

```
<script>
  window.advcake_data = window.advcake_data || [];
  window.advcake_data.push({
    pageType: 5,
    user: {
      email: 'md5(user@email.ru)' // email в виде хеша или уникальный_
↪идентификатор, если не передается, данное поле можно оставить пустым
    },
    leads: [{
      id: 'ac31004',
      name: 'Кредит',
      totalPrice: 4800
    }, {
      id: 'ab31005',
      name: 'Дебетовая карта',
      totalPrice: 5000
    }
  ]
});
</script>
```

Предупреждение: ID товаров должны быть всегда одинаковыми, т.е. у каждого товара свой ID.

3.19 Инструкция AppsFlyer (постбеки)

Эта инструкция поможет вам настроить интеграцию мобильных приложений через AppsFlyer — платформы для мобильной атрибуции и маркетинговой аналитики.

Данный тип интеграции настраивается для того, чтобы заказы, созданные через мобильное приложение, корректно учитывались в канале CPA:

1. Если у пользователя установлено мобильное приложение — мы будем отправлять его в приложение, после чего наша система будет атрибутировать все его покупки.
2. Если у пользователя не установлено мобильное приложение — он будет как прежде совершать заказы на сайте по CPA-модели.

Важно понимать, что рекламодатель платит не за установки, а за конкретные покупки.

Мы приводим и атрибутируем каждую покупку за нужным партнером и вебмастером.

Содержание:

1. *Добавление приложения;*
2. *Установка SDK AppsFlyer;*
3. *Интеграция SDK AppsFlyer;*
4. *Настройка отправляемых из приложения событий;*
5. *Настройка панели администратора AppsFlyer;*
6. *Настройка брендированных доменов;*

7. Проведение тестов.

3.19.1 Шаг 1. Добавление приложения (общее)

Примечание: Если у вас еще нет аккаунта AppsFlyer, зарегистрируйте его .

После регистрации аккаунта необходимо добавить ваше приложение в AppsFlyer:

1. Перейдите на ;
2. Нажмите кнопку *+ Add app* и введите данные приложения;
3. В пункте *Select the app type* выберите тип приложения (в нашем случае – мобильное) (1);
4. В пункте *Enter app details* выберите платформу (Android/iOS) и статус приложения (опубликовано / в разработке) (2)
 - В поле «App Store URL» необходимо прикрепить ссылку из маркетплейса на приложение, если оно уже опубликовано. Если нет – необходимо ввести «Apple App ID» для iOS или «Android package name» для Android;
5. Выберите часовой пояс и валюту (3).

Add an app ⓘ
Add an app to get your data flowing

1 Select the app type (Mobile)

2 Enter app details

Select platform

Android iOS Windows

Select App Store status

In store Pending approval/not published

App Store URL (required) ⓘ
https://apps.apple.com/ru/app/macfamilytree-10/id...

Next

3 Set time zone and currency

3.19.2 Шаг 2. Установка SDK AppsFlyer

Чтобы работать через AppsFlyer необходимо интегрировать SDK AppsFlyer в приложение.

3.19.2.1 Для iOS

Установить SDK AppsFlyer для iOS можно несколькими способами:

3.19.2.1.1 Через Cocoapods

1. Установите ;
2. Добавьте зависимости – необходимо добавить последнюю версию AppsFlyerFramework в Podfile проекта с помощью команды:

```
pod 'AppsFlyerFramework'
```

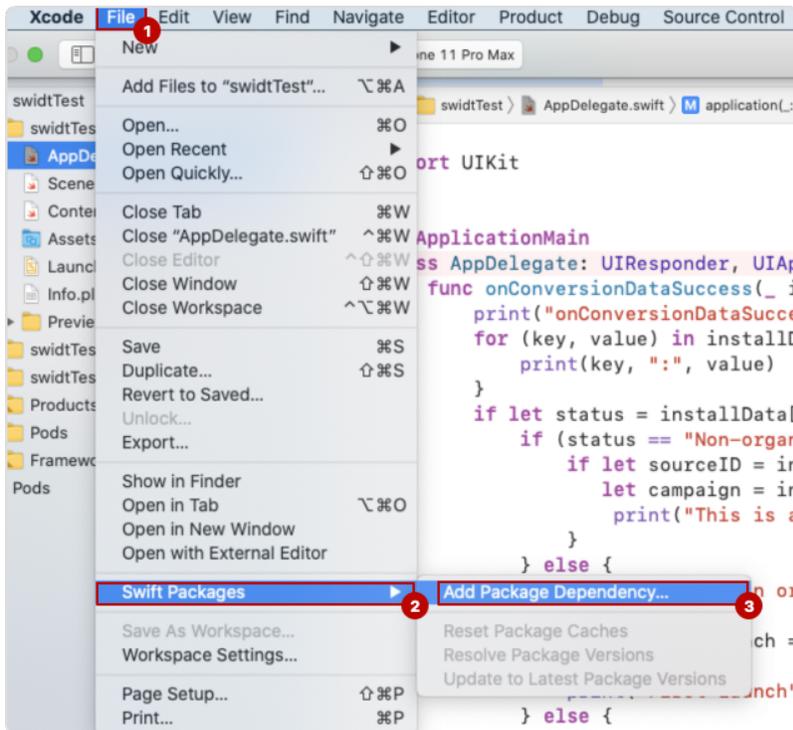
3. Установите зависимости с помощью команды:

```
pod 'install'
```

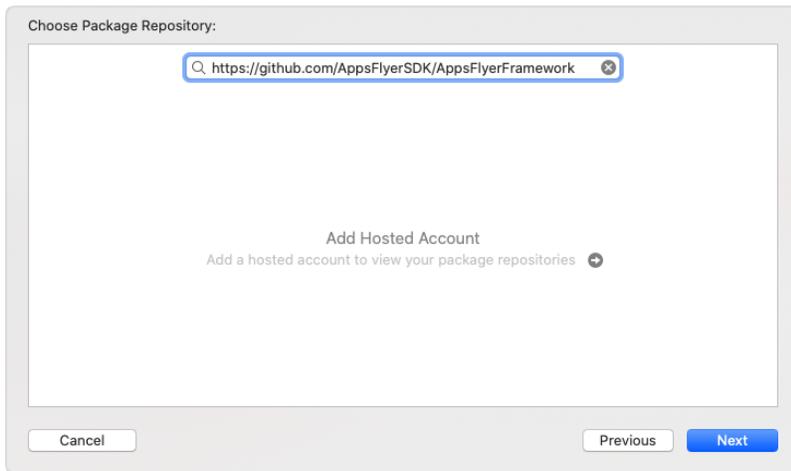
4. После установки необходимо использовать .xcworkspace файл для открытия проекта, вместо .xcodeproj.

3.19.2.1.2 Через SwiftPackageManager (рекомендуется)

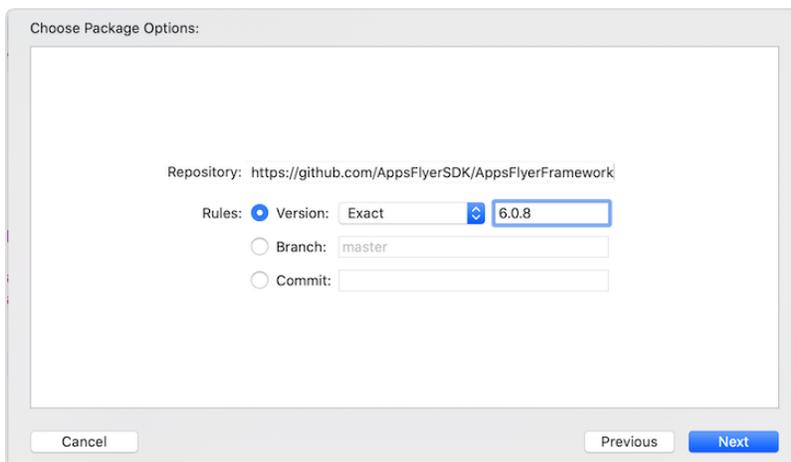
1. В Xcode перейдите по пути: **File** -> **Swift Packages** -> **Add Package Dependency**;



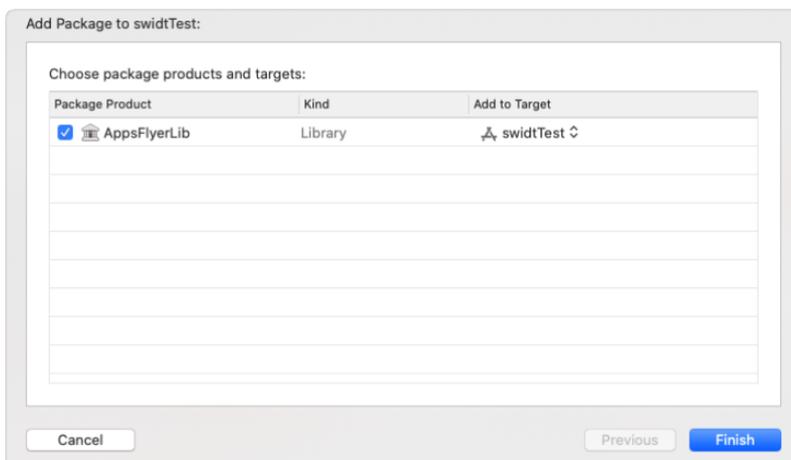
2. Добавьте IOS SDK репозиторий AppsFlyer (<https://github.com/AppsFlyerSDK/AppsFlyerFramework>);



3. Выберите версию SDK – Ехаст (текущая);



4. Добавьте AppsFlyerLib к нужному таргету;



3.19.2.2 Для Android

Установить SDK AppsFlyer для Android можно также несколькими способами:

3.19.2.2.1 Установка с помощью Gradle

1. Добавьте репозиторий `mavenCentral()` в файл `build.gradle`:

```
37 repositories {
38     mavenCentral()
39 }
```

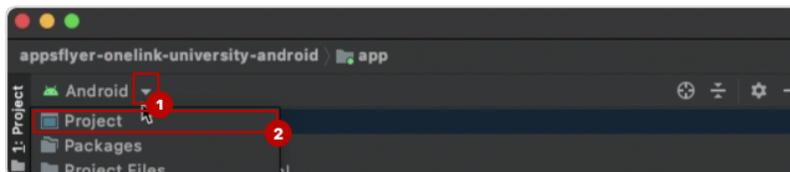
2. Добавьте зависимости в файл `build.gradle` – здесь можно сразу добавить зависимость `installreferrer`, поскольку она понадобится нам в дальнейшем:

```
implementation 'com.android.installreferrer:installreferrer:2.2'
implementation 'com.appsflyer:af-android-sdk:6.3.2'
```

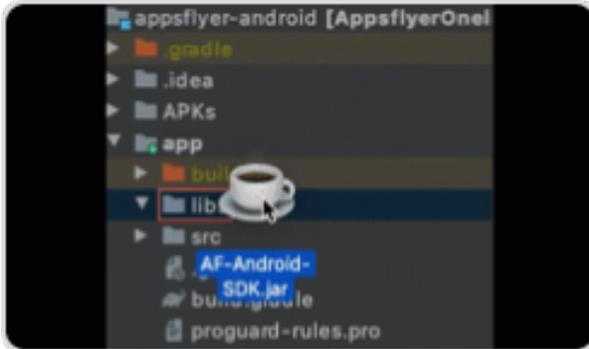
```
41 dependencies {
42     implementation 'com.android.installreferrer:installreferrer:2.2'
43     implementation 'com.appsflyer:af-android-sdk:6.3.2'
44     implementation 'androidx.core:core-ktx:1.7.0'
45     implementation 'androidx.appcompat:appcompat:1.3.0'
46     implementation 'com.google.android.material:material:1.4.0'
47     implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
48     implementation 'androidx.navigation:navigation-fragment-ktx:2.3.5'
49     implementation 'androidx.navigation:navigation-ui-ktx:2.3.5'
50     testImplementation 'junit:junit:4.13.2'
51     androidTestImplementation 'androidx.test.ext:junit:1.1.3'
52     androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
53 }
```

3.19.2.2.2 Установка без Gradle

1. Переведите структуру проекта из **Android в Project**;



2. Скачайте ;
3. Переместите скачанный SDK в app/libs;



3. Правой кнопкой мыши кликните по jar-файлу и нажмите *+ Add as Library*, после чего кликните повторно и нажмите *Refactor*;
4. После установки необходимо добавить разрешения в файл `AndroidManifest.xml`:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="com.google.android.gms.permission.AD_ID" />
```

Предупреждение: Если установка была выполнена без Gradle, необходимо добавить зависимость ниже (для установки с помощью Gradle мы добавили ее ранее) в `build.gradle`:

```
implementation 'com.android.installreferrer:installreferrer:2.2'
```

Далее перейдем к интеграции SDK в приложение.

3.19.3 Шаг 3. Интеграция SDK AppsFlyer

Примечание: Перед началом интеграции SDK AppsFlyer необходимо узнать ключ разработчика (в панели AppsFlyer перейдите в раздел **Configuration** → **App Settings** → поле «Dev key») и Apple App ID (в панели перейдите в раздел **Мои приложения** → выберите ваше приложение → **Информация о приложении** → поле «Apple ID»).

3.19.3.1 Swift

1. Импортируйте AppsFlyerLib:

```
import AppsFlyerLib
```

2. Инициализируйте SDK AppsFlyer:

```
AppsFlyerLib.shared().appsFlyerDevKey="your-key"
AppsFlyerLib.shared().appleAppID="your-id"
```

3. Вызовите `AppsFlyerLib.shared().start()`;

Пример:

```
10 import AppsFlyerLib
11
12
13 @main
14 class AppDelegate: UIResponder, UIApplicationDelegate, AppsFlyerLibDelegate {
15     func onConversionDataSuccess(_ conversionInfo: [AnyHashable : Any]) {
16         print("success data conversion")
17     }
18
19     func onConversionDataFail(_ error: Error) {
20         print("fail data conversion")
21     }
22
23     var window: UIWindow?
24
25     func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
26     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
27         AppsFlyerLib.shared().appsFlyerDevKey = "your-key"
28         AppsFlyerLib.shared().appleAppID = "your-id"
29         AppsFlyerLib.shared().delegate = self
30
31         window = UIWindow(frame: UIScreen.main.bounds)
32         window?.rootViewController = ViewController()
33         window?.makeKeyAndVisible()
34
35         return true
36     }
37
38     @objc func sendLaunch() {
39         AppsFlyerLib.shared().start()
40     }
41 }
```

После этого, добавляем поддержку `AppTrackingTransparency`.

4. Перед вызовом `start()`, добавьте `waitForATTUserAuthorization(timeoutInterval: 60)` в `didFinishLaunchingWithOptions`;

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    AppsFlyerLib.shared().waitForATTUserAuthorization(timeoutInterval: 60)
    AppsFlyerLib.shared().appsFlyerDevKey = "your-key"
    AppsFlyerLib.shared().appleAppID = "your-id"
    AppsFlyerLib.shared().delegate = self

    window = UIWindow(frame: UIScreen.main.bounds)
    window?.rootViewController = ViewController()
    window?.makeKeyAndVisible()

    return true
}
```

5. Запросите одобрение пользователя в `applicationDidBecomeActive()`;

```
func applicationDidBecomeActive(_ application: UIApplication) {
    if #available(iOS 14, *) {
        ATTrackingManager.requestTrackingAuthorization { status in
            switch status {
            case .authorized:
                // Tracking authorization dialog was shown
                // and we are authorized
                print("Authorized")
            case .denied:
                // Tracking authorization dialog was
                // shown and permission is denied
                print("Denied")
            case .notDetermined:
                // Tracking authorization dialog has not been shown
                print("Not Determined")
            case .restricted:
                print("Restricted")
            @unknown default:
                print("Unknown")
            }
        }
    }
}
```

3.19.3.2 Kotlin

1. Импортируйте AppsFlyerLib:

```
import com.appsflyer.AppsFlyerLib
```

2. Инициализируйте SDK AppsFlyer:

```
AppsFlyerLib.getInstance().init("your-key", null, this)
```

3. Запустите Android SDK:

```
AppsFlyerLib.getInstance().start(this)
```

Полный пример:

```
import com.appsflyer.AppsFlyerLib

class MainActivity : AppCompatActivity() {

    private lateinit var appBarConfiguration: AppBarConfiguration
    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        AppsFlyerLib.getInstance().init("your-key", null, this)
        AppsFlyerLib.getInstance().start(this)
    }
}
```

3.19.3.3 React Native

1. Импортируйте appsFlyer из react-native-appsflyer:

Список 1: TypeScript

```
import appsFlyer from 'react-native-appsflyer';
```

Список 2: JavaScript

```
import appsFlyer from 'react-native-appsflyer';
```

2. Инициализируйте SDK AppsFlyer:

Список 3: TypeScript

```
appsFlyer.initSdk({
  devKey: 'K2*****99',
  isDebug: false,
  appId: '41*****44',
  onInstallConversionDataListener: true, //Optional
  onDeepLinkListener: true, //Optional
  timeToWaitForATTUserAuthorization: 10 //for iOS 14.5
});
```

Список 4: JavaScript

```
appsFlyer.initSdk({
  devKey: 'K2*****99',
  isDebug: false,
  appId: '41*****44',
  onInstallConversionDataListener: true, //Optional
  onDeepLinkListener: true, //Optional
```

(continues on next page)

(продолжение с предыдущей страницы)

```
timeToWaitForATTUserAuthorization: 10 //for iOS 14.5
});
```

После интеграции SDK в приложение перейдем к настройке событий.

3.19.4 Шаг 4. Настройка отправляемых из приложения событий

Для начала необходимо определить имя события для отправки. В нашем примере событием является тап по кнопке «Старт», поэтому мы назвали его «start_button_tap». В реальном приложении события могут называться по-разному, например: «purchase», «level_up», «bet» и т.д.

После того, как вы определитесь с именем события для отправки, необходимо подготовить массив (*map/dictionary*) `withValues`, который будет содержать следующие данные о событии:

1. Сумма заказа/доход. В качестве ключа используется константа `AFInAppEventParameterName.REVENUE` (Kotlin) или `AFFEventParamRevenue` (Swift), либо строка `af_revenue` (React Native). Опционально для Android.
2. ID заказа или события в системе рекламодателя. Ключ «order_id» или любой другой (по согласованию). **Обязательно для передачи!**
3. Другие данные, которые вы готовы нам предоставить для атрибутирования заказов, например, категория заказа («category»), тип клиента («clientType») или промокод («coupon»). Опционально.

Далее, необходимо вызвать функцию `logEvent()` из `AppsFlyerLib`, передав ей имя события и массив с данными.

Предупреждение: Реализация интеграции для iOS на этом не заканчивается – смотрите раздел *Реализация на iOS (Swift, React Native)*.

3.19.4.1 Реализация на Kotlin

Пример реализации:

```
val eventData = HashMap<String, Any>()
eventData.put("orderId", orderId)
eventData.put(AFInAppEventParameterName.REVENUE, revenue)
eventData.put("category", category)
eventData.put("clientType", clientType)
eventData.put("coupon", coupon)

AppsFlyerLib.getInstance().logEvent(context, "start_button_tap", eventData)
```

3.19.4.2 Реализация на React Native

Пример реализации:

Список 5: TypeScript

```
appsFlyer.logEvent("purchase", {
  orderId: orderId,
  af_revenue: revenue,
  category: category,
  clientType: clientType,
  coupon: coupon,
});
```

Список 6: JavaScript

```
appsFlyer.logEvent("purchase", {
  orderId: orderId,
  af_revenue: revenue,
  category: category,
  clientType: clientType,
  coupon: coupon,
});
```

3.19.4.3 Реализация на iOS (Swift, React Native)

Реализовать интеграцию AppsFlyer на платформе iOS значительно сложнее, поскольку есть две проблемы:

1. Постбеки событий, отправленных с iOS (независимо от использованной реализации AppsFlyer SDK), приходят с крайне ограниченным количеством данных, а именно: из нетрекинговых данных (данные, которые отправляются через `logEvent()` мы получаем только «af_revenue». Остальные нестандартные поля нам недоступны (в том числе «orderId»)).
2. Постбек события установки приложения (первого запуска приложения) не содержит AppsFlyerID – случайный ID, генерируемый SDK при первом запуске приложения. Из-за этого мы не сможем связывать покупки (или другие действия) с событием установки (нам важны трекинговые метки события установки).

Однако эти препятствия можно обойти.

3.19.4.3.1 Отправка события покупки

Как мы уже писали, нам доступно исключительно поле «af_revenue» – строковое поле, в котором AppsFlyer ожидает числовое значение.

Поле передаётся в постбеке без изменений, а при обработке числа на своей стороне AppsFlyer обрезает его до 5 знаков после запятой. Благодаря этому мы можем использовать это поле для хранения данных, если предварительно закодировать их в числовую последовательность.

Для этих целей мы разработали алгоритм, реализацию которого вы можете увидеть в примерах ниже.

Список 7: Swift

```

class NumSeqEncoder {
    /// Encodes byte array into one large numeric string.
    ///
    /// - Parameter bytes: Byte array with binary data.
    /// - Returns: String with a long number.
    static func encode(_ bytes: [UInt8]) -> String {
        let size = bytes.count

        var string: String = ""
        string.reserveCapacity(Int(ceil(Double(size) * 8 / 3)))

        var buffer: UInt32 = 0
        var bufferSize = 0
        var pos = 0

        while pos < size {
            let pickBytes = min(size - pos, bufferSize != 0 ? 3 : 4)

            buffer = buffer << (pickBytes * 8)
            for y in 0..

```

(continues on next page)

(продолжение с предыдущей страницы)

```

/// Encodes order data into a string as a long float number.
///
/// - Parameters:
///   - data: Data to encode. Allowed up to 1441 bytes (not characters!).
/// - Returns: String with a long float number; up to 3851 characters/bytes.
static func encodeToRevenue(_ data: String) -> String {
    let dataBytes = Array(data.utf8)
    // 3852 is max known length for Appsflyer's revenue field (it's probably greater
    →though).
    if Int(ceil(Double(dataBytes.count) * 8 / 3)) > 3843 {
        return "0.000000"
    }
    return "0.000000" + encode(dataBytes)
}

/// Decodes binary data from a string with a long number.
///
/// - Parameter string: String with a long number.
/// - Returns: Byte array with binary data.
static func decode(_ string: String) -> [UInt8] {
    let bytes: [UInt8] = Array(string.utf8)

    var data = [UInt8](repeating: 0, count: bytes.count * 4 / 8)
    var bitIndex = 0

    for i in 0..

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    }
    let dataEncoded = String(string[string.index(range!.lowerBound, offsetBy: 7)...])
    let dataDecoded = decode(dataEncoded)
    return String(bytes: dataDecoded, encoding: .utf8)
  }
}

```

Список 8: TypeScript

```

class NumSeqEncoder {
  /**
   * Encodes binary string into one large numeric string.
   * @param {string} data String with binary data.
   * @returns {string} String with a long number.
   */
  static encode(data: string): string {
    const bytes = (new TextEncoder().encode(data));
    const size = bytes.length;

    let string = "";

    let buffer = 0;
    let bufferSize = 0;
    let pos = 0;

    while (pos < size) {
      let pickBytes = Math.min(size - pos, bufferSize != 0 ? 3 : 4);

      buffer = buffer << (pickBytes * 8);
      for (let i = 0; i < pickBytes; ++i) {
        buffer |= (bytes[pos + i] & ((1 << 8) - 1)) << ((pickBytes - i - 1) * 8);
      }
      bufferSize += pickBytes * 8;
      pos += pickBytes;

      while (bufferSize >= 3) {
        if (bufferSize >= 4) {
          const quad = buffer >> (bufferSize - 4) & ((1 << 4) - 1);
          if (quad === 0b1000 || quad === 0b1001) {
            string = string.concat(quad.toString());
            bufferSize -= 4;
            buffer &= (1 << bufferSize) - 1;
            continue;
          }
        }
        const triple = buffer >> (bufferSize - 3) & ((1 << 3) - 1);
        string = string.concat(triple.toString());
        bufferSize -= 3;
        buffer &= (1 << bufferSize) - 1;
      }
    }
  }
}

```

(continues on next page)

```
    if (bufferSize > 0) {
        string = string.concat((buffer << (3 - bufferSize)).toString());
    }

    return string;
}

/**
 * Encodes order data into a string as a long float number.
 * @param {string} data String to encode. Allowed up to 1441 bytes (not characters!).
 * @returns {string} String with a long float number; up to 3851 characters/bytes.
 */
static encodeToRevenue(data: string): string {
    const size = new TextEncoder().encode(data).length;
    // 3852 is max known length for Appsflyer's revenue field (it's probably greater
    →though).
    if (Math.ceil(size * 8 / 3) > 3843) {
        return "0.000000";
    }
    return "0.000000" + NumSeqEncoder.encode(data);
}

/**
 * Decodes binary data from a string with a long number.
 * @param {string} string String with a long number.
 * @returns {string} Decoded binary string.
 */
static decode(string: string): string {
    const bytes = (new TextEncoder().encode(string));
    const size = bytes.length;

    let data = new Uint8Array(size * 4 / 8);
    let bitIndex = 0;

    for (let i = 0; i < size; ++i) {
        if (bytes[i] < 48 || bytes[i] > 57) {
            break;
        }
        const number = bytes[i] - 48;
        let len = number >= 8 ? 4 : 3;
        for (let y = len - 1; y >= 0; --y) {
            const byteIndex = ~~(bitIndex / 8);
            const byteBitIndex = bitIndex % 8;
            data[byteIndex] = (data[byteIndex] || 0) | ((number >> y & 1) << (7 -
            →byteBitIndex));
            ++bitIndex;
        }
    }

    return new TextDecoder().decode(data.slice(0, Math.floor(bitIndex / 8)));
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

/**
 * Parses a string of data encoded as a long float number.
 * @param {string} string String with a long float number.
 * @returns {string|null} Decoded binary string. Null if string is invalid.
 */
static decodeFromRevenue(string: string): string | null {
    const pos = string.indexOf('.');
    if (pos === -1) {
        return null;
    }
    return NumSeqEncoder.decode(string.substring(pos + 7));
}
}

```

Список 9: JavaScript

```

"use strict";

class NumSeqEncoder {
    /**
     * Encodes binary string into one large numeric string.
     * @param {string} data String with binary data.
     * @returns {string} String with a long number.
     */
    static encode(data) {
        const bytes = (new TextEncoder()).encode(data);
        const size = bytes.length;

        let string = "";

        let buffer = 0;
        let bufferSize = 0;
        let pos = 0;

        while (pos < size) {
            let pickBytes = Math.min(size - pos, bufferSize !== 0 ? 3 : 4);

            buffer = buffer << (pickBytes * 8);
            for (let i = 0; i < pickBytes; ++i) {
                buffer |= (bytes[pos + i] & ((1 << 8) - 1)) << ((pickBytes - i - 1) * 8);
            }
            bufferSize += pickBytes * 8;
            pos += pickBytes;

            while (bufferSize >= 3) {
                if (bufferSize >= 4) {
                    const quad = buffer >> (bufferSize - 4) & ((1 << 4) - 1);
                    if (quad === 0b1000 || quad === 0b1001) {
                        string = string.concat(quad.toString());
                        bufferSize -= 4;
                        buffer &= (1 << bufferSize) - 1;
                        continue;
                    }
                }
            }
        }
    }
}

```

(continues on next page)

```

    }
    }
    const triple = buffer >> (bufferSize - 3) & ((1 << 3) - 1);
    string = string.concat(triple.toString());
    bufferSize -= 3;
    buffer &= (1 << bufferSize) - 1;
  }
}

if (bufferSize > 0) {
  string = string.concat((buffer << (3 - bufferSize)).toString());
}

return string;
}

/**
 * Encodes order data into a string as a long float number.
 * @param {string} data String to encode. Allowed up to 1441 bytes (not characters!).
 * @returns {string} String with a long float number; up to 3851 characters/bytes.
 */
static encodeToRevenue(data) {
  const size = new TextEncoder().encode(data).length;
  // 3852 is max known length for Appsflyer's revenue field (it's probably greater
↳though).
  if (Math.ceil(size * 8 / 3) > 3843) {
    return "0.000000";
  }
  return "0.000000" + NumSeqEncoder.encode(data);
}

/**
 * Decodes binary data from a string with a long number.
 * @param {string} string String with a long number.
 * @returns {string} Decoded binary string.
 */
static decode(string) {
  const bytes = (new TextEncoder().encode(string));
  const size = bytes.length;

  let data = new Uint8Array(size * 4 / 8);
  let bitIndex = 0;

  for (let i = 0; i < size; ++i) {
    if (bytes[i] < 48 || bytes[i] > 57) {
      break;
    }
    const number = bytes[i] - 48;
    let len = number >= 8 ? 4 : 3;
    for (let y = len - 1; y >= 0; --y) {
      const byteIndex = ~(bitIndex / 8);
      const byteBitIndex = bitIndex % 8;

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        data[byteIndex] = (data[byteIndex] | 0) | ((number >> y & 1) << (7 -
↪byteBitIndex));
        ++bitIndex;
    }
}

return new TextDecoder().decode(data.slice(0, Math.floor(bitIndex / 8)));
}

/**
 * Parses a string of data encoded as a long float number.
 * @param {string} string String with a long float number.
 * @returns {string|null} Decoded binary string. Null if string is invalid.
 */
static decodeFromRevenue(string) {
    const pos = string.indexOf('.');
    if (pos === -1) {
        return null;
    }
    return NumSeqEncoder.decode(string.substring(pos + 7));
}
}

```

Необходимо отправлять два события: «purchase» и «advcake_purchase». Оба должны содержать данные о заказе, но второй должен содержать закодированное «revenue» вместо настоящей суммы дохода/цены.

Включите в своё приложение класс NumSeqEncoder. При создании пользователем заказа сформируйте массив с данными о заказе и дополнительно добавьте в него поле «appsflyerId» со значением функции AppsFlyerLib.shared().getAppsFlyerUID() (appsFlyer.getAppsFlyerUID()). Затем сформируйте такой же массив (либо переиспользуйте старый) и замените в нём поле AFEventParamRevenue (af_revenue) на результат функции NumSeqEncoder.encodeToRevenue(json) (строка), передав ему в качестве аргумента этот же массив с данными о заказе, закодированный в JSON-строку.

Смотрите пример реализации ниже.

Примечание: Обратите внимание, что размер JSON-строки с данными не должен превышать 1441 байт (не символов), что равняется <=3851 байт в закодированном виде. Это ограничение AppsFlyer, выявленное путём проведения крайне ограниченного числа тестов. Вполне вероятно, что допустимы строки большего размера. В ближайшее время информация будет уточнена, а инструкция и код обновлены.

Если переданная в функцию encodeToRevenue() строка будет превышать описанное ограничение, данные закодированы не будут (функция вернёт «0.000000»).

Поэтому перед добавлением в JSON очередного поля убедитесь, что суммарная максимальная длина значений всех полей, а также длина ключей и токенов не превышает описанного ограничения.

Пример реализации:

Список 10: Swift

```
struct OrderInfo: Codable {
    let appsflyerId: String
    let orderId: String
    let revenue: Double
    let category: String
    let clientType: String
    let coupon: String?

    var dictionary: [String: Any?] {
        return [
            "appsflyerId": appsflyerId,
            "orderId": orderId,
            "revenue": revenue,
            "category": category,
            "clientType": clientType,
            "coupon": coupon,
        ]
    }
}

//...

let orderInfo = OrderInfo(
    appsflyerId: AppsFlyerLib.shared().getAppsFlyerUID(),
    orderId: orderId,
    revenue: revenue,
    category: category,
    clientType: clientType,
    coupon: coupon
)

var afEventValues = orderInfo.dictionary
afEventValues[AfEventParamRevenue] = afEventValues["revenue"]
AppsFlyerLib.shared().logEvent("purchase", withValues: afEventValues)

let orderInfoJsonData = try JSONEncoder().encode(orderInfo)
let orderInfoJsonStr = String(data: orderInfoJsonData, encoding: .utf8) ?? ""
afEventValues[AfEventParamRevenue] = NumSeqEncoder.encodeToRevenue(orderInfoJsonStr)
AppsFlyerLib.shared().logEvent("advcake_purchase", withValues: afEventValues)
```

Список 11: TypeScript

```
const eventValues = {
    appsflyerId: appsFlyer.getAppsFlyerUID(),
    orderId: orderId,
    af_revenue: revenue,
    category: category,
    clientType: clientType,
    coupon: coupon,
};
```

(continues on next page)

(продолжение с предыдущей страницы)

```

const eventValuesAdvcake = {...eventValues};
eventValuesAdvcake.af_revenue = NumSeqEncoder.encodeToRevenue(JSON.
↳stringify(eventValues));

appsFlyer.logEvent("purchase", eventValues);
appsFlyer.logEvent("advcake_purchase", eventValuesAdvcake);

```

Список 12: JavaScript

```

const eventValues = {
  appsflyerId: appsFlyer.getAppsFlyerUID(),
  orderId: orderId,
  af_revenue: revenue,
  category: category,
  clientType: clientType,
  coupon: coupon,
};

const eventValuesAdvcake = {...eventValues};
eventValuesAdvcake.af_revenue = NumSeqEncoder.encodeToRevenue(JSON.
↳stringify(eventValues));

appsFlyer.logEvent("purchase", eventValues);
appsFlyer.logEvent("advcake_purchase", eventValuesAdvcake);

```

3.19.4.3.2 Отправка дополнительного события установки

Дополнительное событие установки необходимо отправлять при первом запуске приложения, а также если пользователь попал в приложение через Universal Links.

Это событие никак не повлияет на статистику в AppsFlyer.

3.19.4.3.3 Swift

1. Включите в своё приложение класс `NumSeqEncoder`, как это описано в *предыдущем разделе*;
2. Реализуйте в своём приложении выполнение следующего кода при каждом запуске приложения на устройстве пользователя:
 - Имплементируйте метод `UIApplicationDelegate: application:continue:restorationHandler:`. Этот метод будет вызываться каждый раз при открытии приложения через deeplink. Проверьте, что у вас имеется параметр `pid = advcake_int` и хост совпадает с хостом вашей ссылки:

Список 13: Swift

```

func application(
  _ application: UIApplication,
  continue userActivity: NSUserActivity,

```

(continues on next page)

(продолжение с предыдущей страницы)

```

restorationHandler: @escaping ([UIUserActivityRestoring]?) -> Void
) -> Bool {

/// Если приложение открыто через диплинк и содержит параметр
↳pid=advcake_int

    let hosts = [
        "yourapp.onelink.me"
    ]
    if userActivity.activityType == NSUserActivityTypeBrowsingWeb,
        let incomingURL = userActivity.webpageURL,
        let components = NSURLComponents(url: incomingURL,
↳resolvingAgainstBaseURL: true),
        let params = components.queryItems,
        let host = components.host,
        hosts.contains(host),
        params.first(where: { $0.name == "pid" })?.value == "advcake_int"
        {
            UserDefaults.standard.set(true, forKey: "advcakeDeepLink")
        }

    return true
}

```

Примечание: Вместо «yourapp» в yourapp.onelink.com укажите поддомен вашего OneLink-шаблона. Если у вас его ещё нет, мы создадим его в 5 шаге – *Настройка панели администратора AppsFlyer*.

- Добавьте следующий код в функцию applicationDidBecomeActive():

Список 14: Swift

```

/// Если приложение не было установлено ранее, либо произошел переход
↳через Universal Links - отправляем событие advcake_install

if !UserDefaults.standard.bool(forKey: "launchedBefore")
    || UserDefaults.standard.bool(forKey: "advcakeDeepLink")
{
    let values = [
        AFEventParamRevenue: NumSeqEncoder().enrichRevenue(0,
↳AppsFlyerLib.shared().getAppsFlyerUID())
    ]
    AppsFlyerLib.shared().logEvent("advcake_install", withValues: values,
↳as [AnyHashable: Any])
    UserDefaults.standard.set(false, forKey: "advcakeDeepLink")
}

/// После необходимых действий, устанавливаем ключ true для
↳launchedBefore

UserDefaults.standard.set(true, forKey: "launchedBefore")

```

3.19.4.3.4 React Native

1. Включите в своё приложение класс `NumSeqEncoder`, как это описано в предыдущем разделе;
2. Реализуйте в своём приложении выполнение следующего кода при первом запуске на устройстве пользователя или открытии через deeplink:

Список 15: JavaScript

```
appsFlyer.initSdk({
  devKey: 'K2*****99',
  isDebug: false,
  appId: '41*****44',
  onInstallConversionDataListener: true, //Optional
  onDeepLinkListener: true, //Optional
  timeToWaitForATTUserAuthorization: 10 //for iOS 14.5
});

appsFlyer.logEvent("advcake_install", {
  af_revenue: NumSeqEncoder.encodeToRevenue(appsFlyer.getAppFlyerUID())
});
```

Список 16: TypeScript

```
appsFlyer.initSdk({
  devKey: 'K2*****99',
  isDebug: false,
  appId: '41*****44',
  onInstallConversionDataListener: true, //Optional
  onDeepLinkListener: true, //Optional
  timeToWaitForATTUserAuthorization: 10 //for iOS 14.5
});

appsFlyer.logEvent("advcake_install", {
  af_revenue: NumSeqEncoder.encodeToRevenue(appsFlyer.getAppFlyerUID())
});
```

3.19.5 Шаг 5. Настройка панели администратора AppsFlyer

3.19.5.1 Включение ретаргетинга в настройках приложения

1. Выберите ваше приложение и перейдите в раздел **Configuration** → **App Settings**;
2. Выставьте окно реатрибуции (рекомендуемое значение – 3 месяца);
3. Установите галочку на «Re-engagement attribution»;
4. Выставьте время между разными привязками (рекомендуемое значение – none).

Attribution

Re-attribution window ⓘ
Set a timer in months, starting with the first install, during which reinstalls aren't attributed as new installs.

90 days (3 months) ▾

Enable view-through attribution via probabilistic modeling ⓘ

Re-engagement attribution ⓘ

Minimum time between re-engagement conversions ⓘ
Select the time frame from the occurrence of a re-engagement conversion, during which other consequent re-engagement events will be ignored.

None Hours Days

Ретаргетинг включен! Перейдем к настройке OneLink-шаблона.

3.19.5.2 Настройка OneLink-шаблона

Настройка OneLink-шаблона нужна для того, чтобы пользователь при переходе по ссылке попадал сразу в приложение. При этом, будет работать ретаргетинг.

1. В панели администратора AppsFlyer перейдите в раздел **Experiences & Deep Linking** → вкладка **OneLink Custom Links**;
2. Нажмите кнопку «Три точки» и выберите пункт *New OneLink template* (или *Edit OneLink template* если шаблон уже имеется):

OneLink Management ⓘ

OneLink template: test ⓘ ⚙️ ⭐️

Template ID: dA2P | Modification date: Sep 07, 2023 | ⓘ ⋮

⌵ Edit OneLink template
➕ New OneLink template

Link name	Type	Media source	Short URL	Last modified ↓
 Create your first link using this template <small>Links created with OneLink give a specific experience to your users, and redirect, deep link, and attribute them.</small> <input type="button" value="+ New link"/>				

3. Задайте произвольный поддомен:

* Subdomain

Subdomain cannot be changed once there are custom links based on this template.

testtest1234 .onelink.me

Примечание: Позже этот домен будет использоваться для создания самих ссылок.

4. Выберите поведение ссылки:

В разделе «When app isn't installed» (если приложение **не установлено**) в соответствующих полях укажите:

- **iOS:** Redirect users to your domain: <https://your-domain.ru/>;
- **Android:** Redirect users to your domain: <https://your-domain.ru/>;

Если у вас нет веб-версии, а только приложение, то вместо перенаправления на веб, используйте перенаправление в магазин:

- **iOS:** Redirect users to AppStore;
- **Android:** Redirect users to Google Play.

В разделе «When app is installed» (если приложение **установлено**) в соответствующих полях укажите:

- **iOS:** Launching the app using Universal links;
- **Android:** Launching the app using Android App links;
- **Android and iOS fallback:** No fallback is set (стоит по умолчанию).

А также «Apple Team ID» (для iOS) и SHA256 (для Android) по кнопке «Карандаш» (о том, что это такое и где найти, смотрите в подсказке ниже).

В разделе «When link is clicked on desktop» (если на ссылку кликнули на десктопе) укажите:

- **Desktop:** Redirect users to your domain: <https://your-domain.ru/>.

Примечание: Вместо <https://your-domain.ru/> укажите ссылку на свой основной лендинг.

OneLink custom links > OneLink template

Update OneLink template

When app isn't installed

iOS	Redirecting users to: https://your-domain.ru
	<input type="checkbox"/> Display App Clip for users with Private Relay. Learn more <p>Note: Only turn this on if you have an App Clip configured.</p>
Android	Redirecting users to: https://your-domain.ru

Redirection when the app isn't installed
Send iOS or Android users to the app's page on the App Store or Google Play, or to another web page.

When app is installed

Configuring the following setup should be done in collaboration between the marketer and developer

iOS	Launching the app using Universal links Add test app
Android	Launching the app using Android App Links (1 SHA256 fingerprint) Code snippet for AndroidManifest.xml
Android and iOS fallback	No fallback is set Add URI scheme to launch the app

When link is clicked on desktop

Desktop	Redirecting users to: https://your-domain.ru
---------	---

Подсказка: Где найти «Apple Team ID» для iOS и SHA256 для Android?

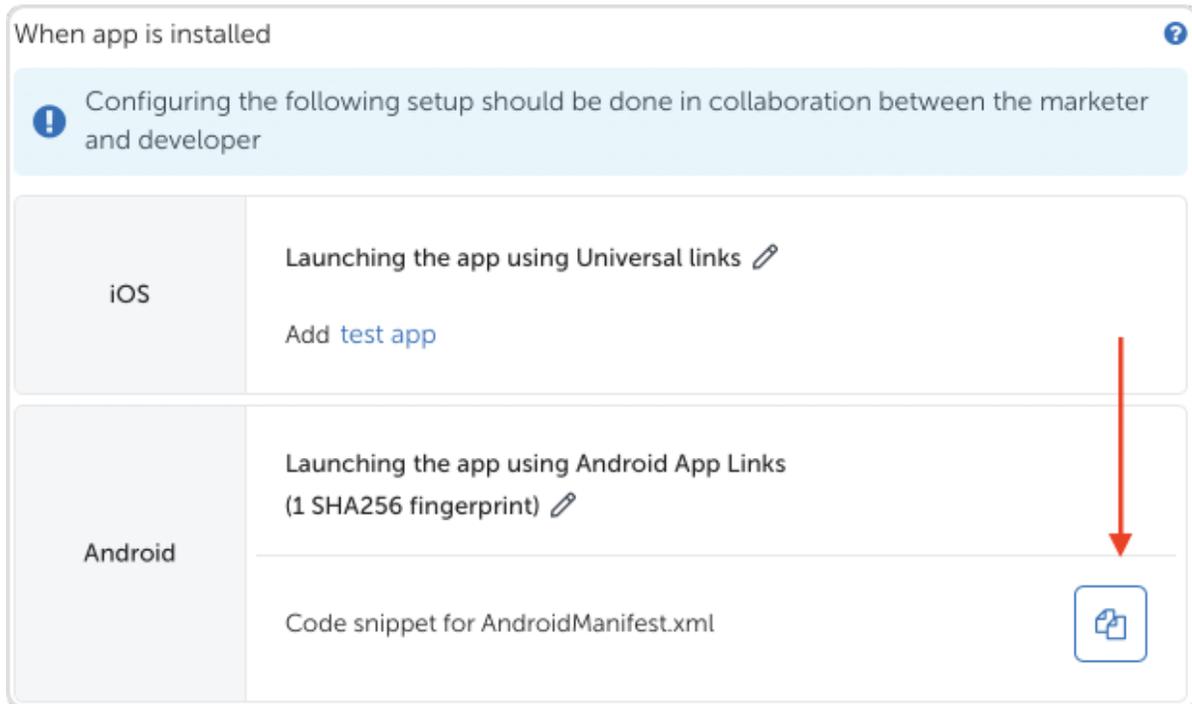
Чтобы найти «Apple Team ID», перейдите на и авторизируйтесь под учетной записью разработчика – в верхней части страницы (там, где ваши Ф.И.) «Имя компании» и будет означать «Apple Team ID».

Чтобы найти SHA256, перейдите на , затем, в меню слева, выберите «Настройка» -> «Целостность приложения» -> раздел «Сертификат ключа для подписи приложения». SHA256 будет указан в поле «Цифровой отпечаток сертификата SHA-256».

5. Внесите правки в приложение:

Для Android

После заполнения SHA256 для Android вы получите возможность скопировать код, который необходимо добавить в `AndroidManifest.xml`;



Скопируйте код нажатием на кнопку и добавьте его в файл `AndroidManifest.xml` в блок, описывающий `main activity`.

Скопированный код можно расположить перед закрывающим тегом `</activity>`.

Ваш `AndroidManifest.xml` будет выглядеть так:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.advcake.getluckyapp">
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="com.google.android.gms.permission.AD_ID" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.GetLuckyApp">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.GetLuckyApp.NoActionBar">
```

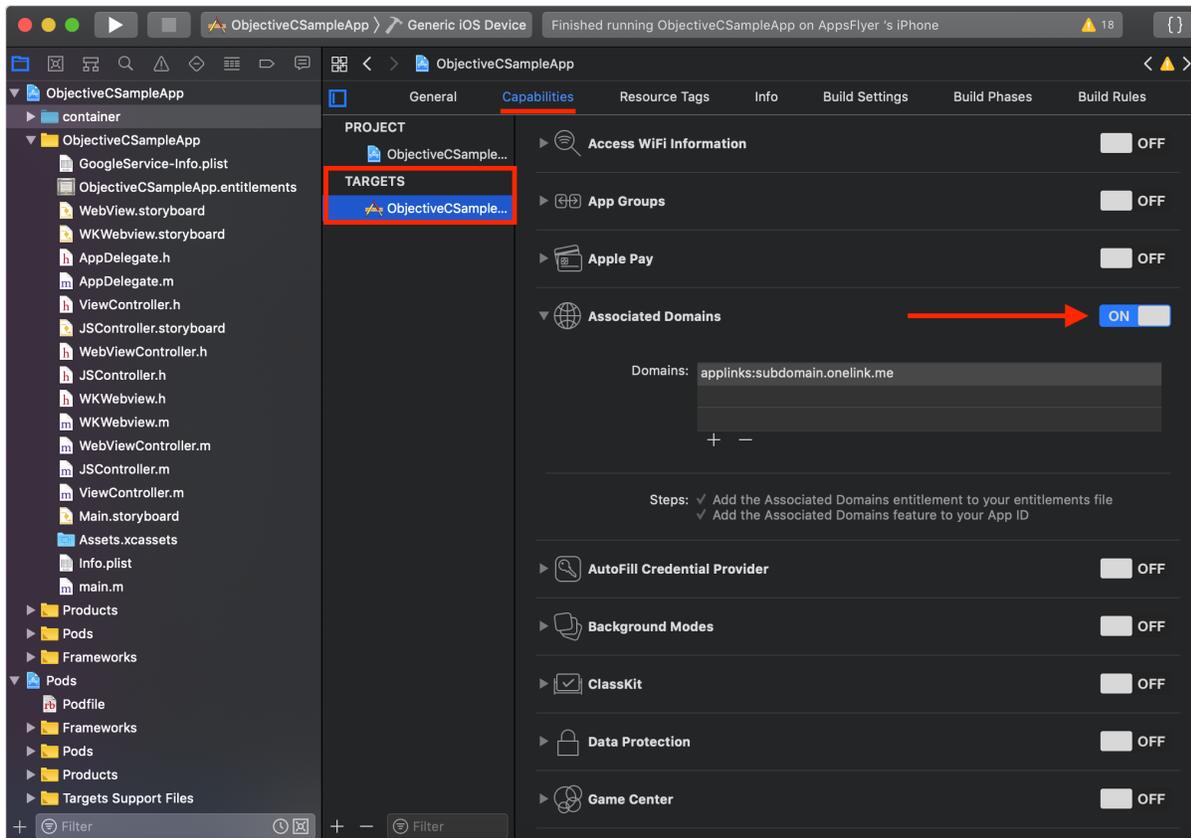
(continues on next page)

(продолжение с предыдущей страницы)

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
<!-- Начало скопированного кода -->
<intent-filter android:autoVerify="true">
  <action android:name="android.intent.action.VIEW" />
  <category android:name="android.intent.category.DEFAULT" />
  <category android:name="android.intent.category.BROWSABLE" />
  <data android:scheme="https"
        android:host="subdomain.onelink.me"
        android:pathPrefix="/ZBZ8" />
</intent-filter>
<!-- Конец скопированного кода -->
</activity>
</application>
</manifest>
```

Для iOS

1. Перейдите в Xcode;
2. Нажмите на свой проект;
3. Нажмите на таргет проекта;
4. Перейдите на вкладку **Capabilities**;
5. Активируйте переключатель *Associated domains*;
6. Добавьте в список доменов ту ссылку, которую вы получили после добавления поддомена. К ней нужно добавить префикс `applinks`, после чего ссылка будет выглядеть так – `applinks:subdomain.onelink.me`:

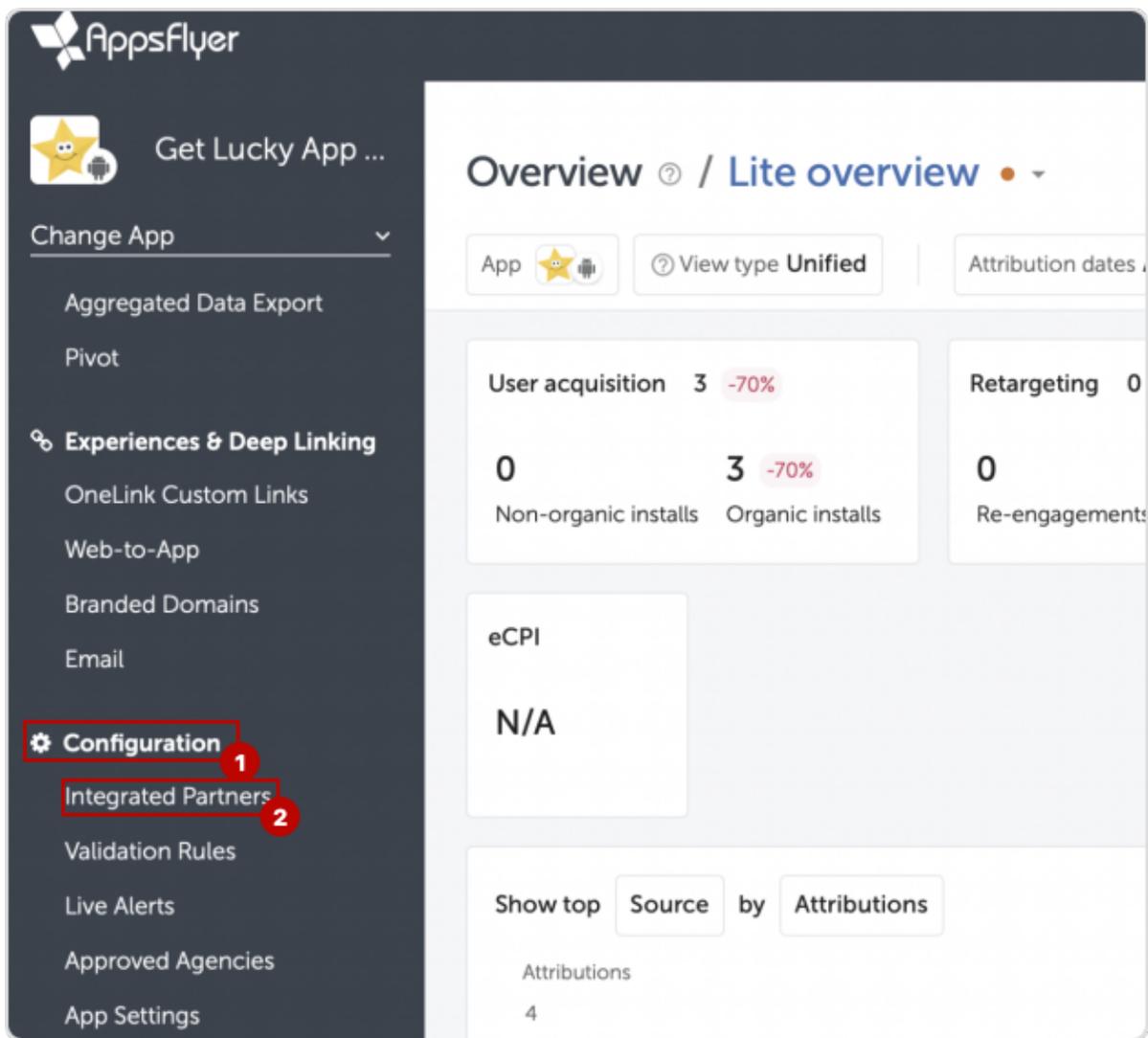


7. Вернитесь в файл `AppDelegate.swift` к методу `UIApplicationDelegate:application:continue:restorationHandler` и замените поддомен `yourapp` в `<yourapp.onelink.me>` на ваш новый поддомен.

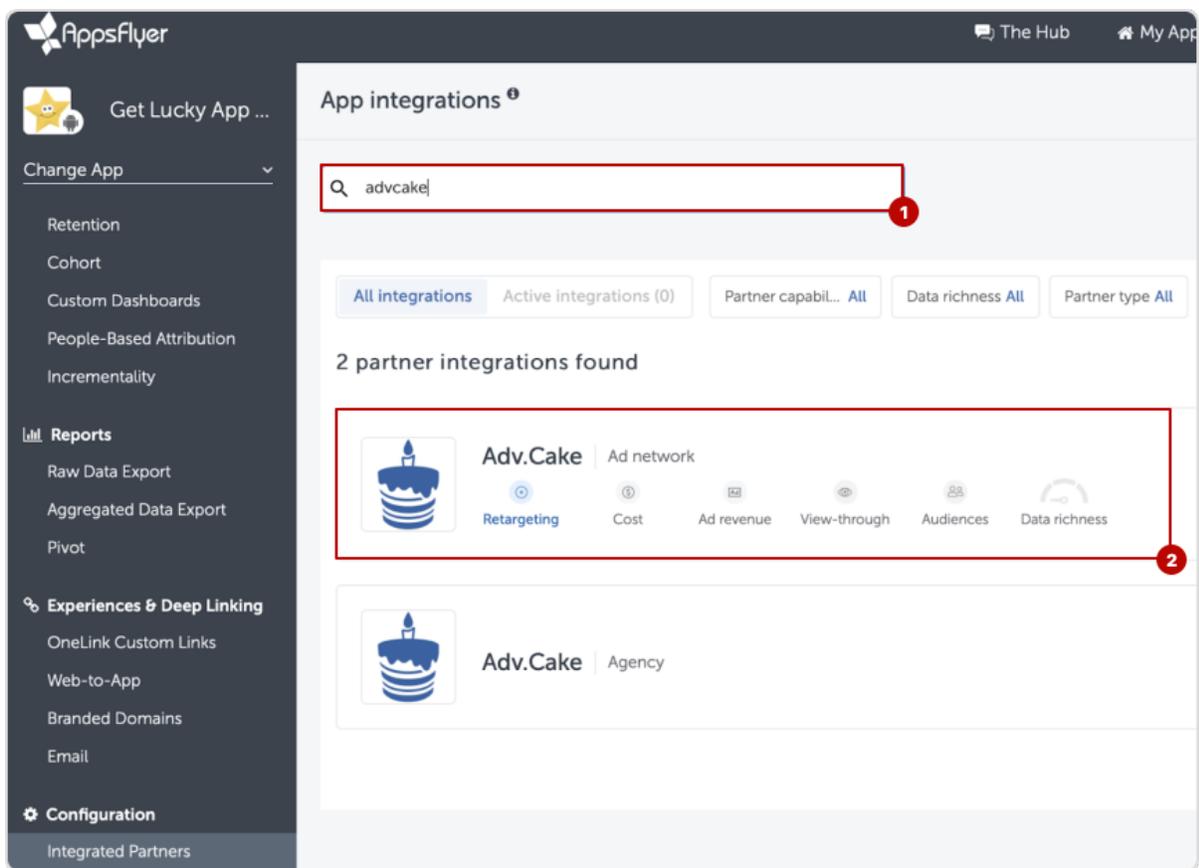
Поведение ссылок настроено, теперь добавим Adv.Cake как рекламную сеть:

3.19.5.3 Добавление Adv.Cake

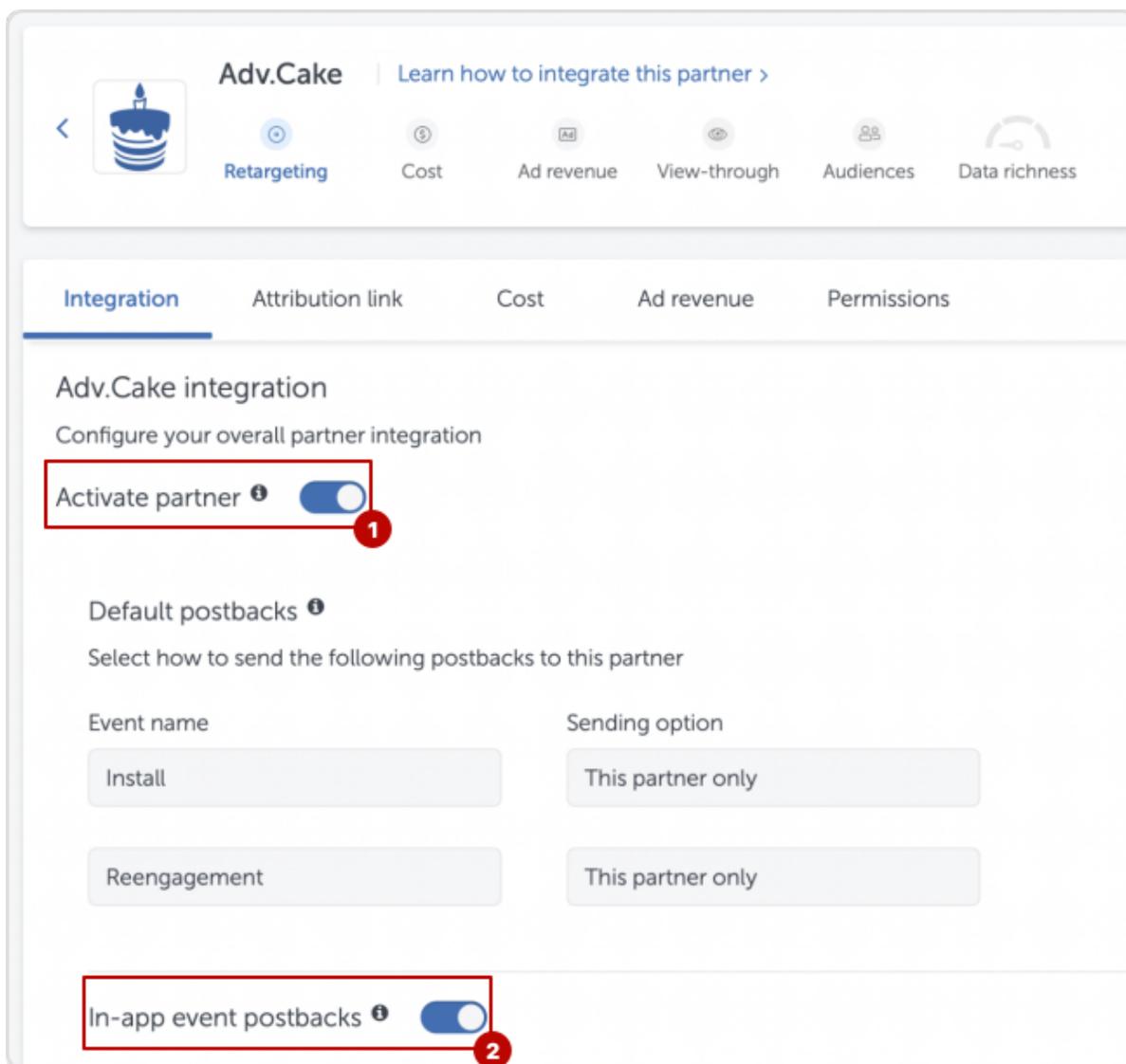
1. В панели администратора AppsFlyer перейдите в раздел **Configuration** → **Integrated Partners**;



-
2. Вбейте в поиск Adv.Cake, после чего появятся два партнера – выберите с пометкой *Ad Network* (рекламная сеть):



3. На вкладке **Integration** активируйте галочки «Activate partner» и «In-app event postbacks»;



4. Затем, добавьте события, которые отправляются из нашей сети (настраивали *ранее*). Для этого, нажмите *Add event* и добавьте нужное событие:

- В «Partner event identifier» можно указать название события (например, `start_button_tapped`);
- В «Send revenue» – **values & revenue**.

Предупреждение: Обратите внимание, что в постбеках для iOS не будут приходить «values». Для этого, мы настраивали «revenue».

In-app event postbacks

In-app events postback window [?]

Hours Days Months Lifetime

Bulk actions ▾

SDK event name [?]

start_button_tapped ▾ >

Partner event identifier [?]

start_button_tapped

Sending option [?]

This partner only ▾

Send revenue [?]

Values & revenue ▾

Add event ¹

< Back to list

⚠ You have unsaved changes [Discard changes](#) [Save Integration](#)

5. Перейдите на вкладку **Attribution link**, выберите шаблон в поле «Select OneLink template» (настраивали *ранее*) и активируйте галочку «Retargeting Settings» в нижней части страницы;

AppsFlyer

PublicTimer (id16081...)

Integration **Attribution link** Cost Ad revenue

Choose your attribution link type:

Use a single-platform link

Use **OneLink**

OneLink redirects users to the right app store, web URL, or into the app, and helps you boost conversion rates by building personalized, secure user journeys for your campaigns. [Learn more](#)

Select OneLink template [?]

test ▾

Deep linking

Attribution link parameters

Select parameters to add to this attribution link

6. Перейдите на вкладку **Permissions** и установите галочки на указанных пунктах – это позволит самостоятельно настраивать ссылки. После чего, в нижней части страницы, нажмите *Save Permissions*.

Интеграция

Ссылка атрибуции

Затраты

Доход от рекламы

Права доступа

Доступ Adv.Cake

Выберите права для предоставления данному партнеру [Узнать больше >](#)

Доступ рекламной сети

Разрешите рекламной сети просматривать свой трафик по вашему приложению на их дэшборде.

Откройте доступ членам команды Adv.Cake

aa@advcake.com (admin)



[+ Добавить члена команды](#)

Предоставьте следующие права доступа членам команды Adv.Cake:

Разрешить конфигурировать интеграцию



Конфигурировать постбэки внутренних событий приложения



Открыть доступ к отчету об удержании



Открыть доступ к агрегированным данным лояльных пользователей



Access aggregate conversions data ⁱ



Открыть доступ к агрегированным данным внутренних событий приложения ⁱ



View validation rules



Доступ к агрегированным данным о доходе



Импорт данных о расходах ⁱ

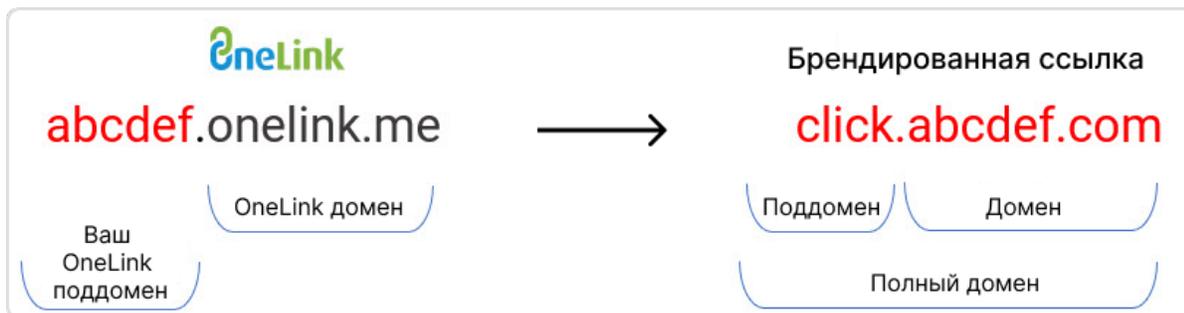


Открыть доступ к дэшборду Protect360 и к сырым данным ⁱ



3.19.6 Шаг 6. Настройка брендированных доменов

Брендированные домены позволяют указывать в ссылках атрибуции название бренда и домен.



Многие блокировщики рекламы и VPN-сервисы блокируют доступ к некоторым доменам, среди которых – onelink.me.

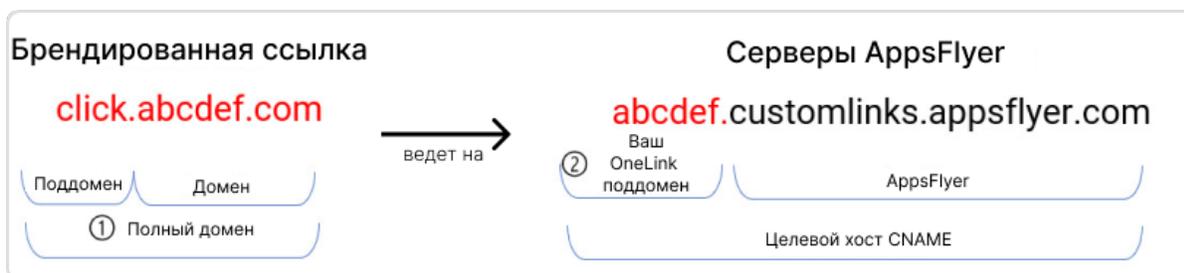
Пользователь не может перейти по ссылке и попросту теряется, однако эту проблему можно решить с помощью брендированных доменов.

Брендированный домен связывается с доменом AppsFlyer с помощью стандартных параметров DNS.

Вам также потребуется внести небольшие правки в приложение и привязать домен в панели AppsFlyer.

Чтобы подключить брендированный домен, необходимо выполнить следующие шаги:

3.19.6.1 Настройка DNS



1. Выберите полное доменное имя, например, `click.abcdef.com`, где `abcdef.com` — название вашего бренда. Поддомен (например, `click`) можно также настроить;
2. Попросите администратора DNS создать полный домен (хост);
3. Попросите администратора DNS настроить запись CNAME, чтобы полный домен (домен бренда) указывал на заданный URL-адрес (хост AppsFlyer), как это показано на схеме выше – брендированная ссылка указывает на серверы AppsFlyer.

3.19.6.2 Привязка домена к AppsFlyer

1. В панели администратора AppsFlyer перейдите в раздел **Engagement & Deep Linking** -> **Branded Domains**;
2. Нажмите *+ Add branded domain*;
3. В поле «Brand Domain» введите полный домен, как в записи DNS (например, `click.abcdef.com`);
4. Выберите поддомен OneLink из открывшегося списка;

The screenshot shows the 'Add Branded Link' form. The 'Brand domain' field contains 'click.example.com'. The 'OneLink subdomain' section has 'abcdef.onelink.me' selected. The 'DNS settings status' section shows 'Domain exists' and 'AppsFlyer CNAME found' with a 'Learn more >' link. The CNAME record field contains 'abcdef.customlinks.appsflyer.com'. The 'Add' button is highlighted in blue.

5. Нажмите *Verify*.

Предупреждение: Если диплинкинг выполняется с использованием брендированных доменов, SDK AppsFlyer не может получить данные о конверсиях по установкам и диплинкингу.

3.19.6.3 Настройка мобильного приложения

Необходимо использовать методы `setOneLinkCustomDomain` и `oneLinkCustomDomains` для Android и iOS, соответственно.

Так производится отправка запроса по брендированной ссылке, возвращается ссылка OneLink, которой она сопоставлена, после чего направляется запрос OneLink для получения данных о конверсиях.

3.19.6.3.1 Для Android

```
public class AFApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();

        AppsFlyerConversionListener conversionListener = new AppsFlyerConversionListener() {
        }
        AppsFlyerLib.getInstance().setOneLinkCustomDomain("promotion.greatapp.com");
        AppsFlyerLib.getInstance().init(AF_DEV_KEY, conversionListener, this);
        AppsFlyerLib.getInstance().start(this, AF_DEV_KEY);

    }
}
```

Если у вас несколько брендированных доменов, передайте их все в API, чтобы всегда получать данные о конверсиях и иметь возможность использовать диплинки:

```
AppsFlyerLib.getInstance().setOneLinkCustomDomain(
    "promotion.greatapp.com",
    "click.greatapp.com",
    "deals.greatapp.com");
```

3.19.6.3.2 Для iOS

Настройте брендированную ссылку, чтобы она получала данные о конверсии, используя приведенный ниже метод:

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    AppsFlyerLib.shared().appsFlyerDevKey = "6CQi4Be6Zs9oNLsCusPbUL"
    AppsFlyerLib.shared().appleAppID = "340954504"
    AppsFlyerLib.shared().oneLinkCustomDomains = ["promotion.greatapp.com"]
    //...
    //...
}
```

Подсказка: Если у вас несколько брендированных доменов, их можно передать в API как массив строк. Так, независимо от того, какая именно брендированная ссылка используется, вы всегда будете получать данные о конверсиях и сможете использовать диплинки:

```
AppsFlyerLib.shared().oneLinkCustomDomains = ["promotion.greatapp.com", "click.greatapp.
→com"]
```

Теперь, ссылку с брендированным доменом необходимо «защитить» в приложение.

3.19.6.3.3 Для Android

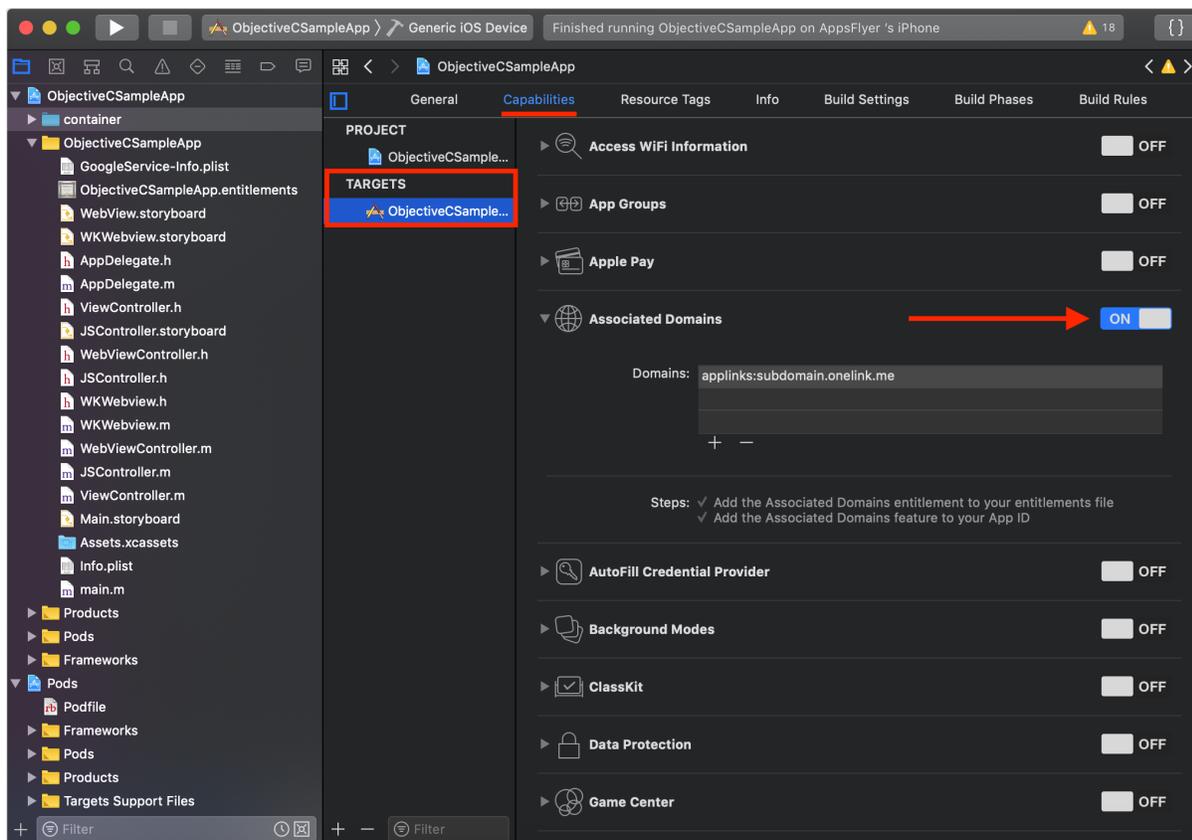
1. Откройте файл `AndroidManifest.xml`;
2. Найдите блок `intent-filter`, содержащий в себе `<action android:name=android.intent.action.VIEW />` (если такого нет, вы можете скопировать его из примера ниже и отредактировать под свои ссылки);
3. Добавьте в него объект `<data />`, описывающий ваш брендированный домен. Например, он может быть таким: `<data android:scheme=https android:host=click.yourdomain.me />`;

Пример файла `AndroidManifest.xml` после правок:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.advcake.getluckyapp">
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="com.google.android.gms.permission.AD_ID" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.GetLuckyApp">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.GetLuckyApp.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter android:autoVerify="true">
                <action android:name="android.intent.action.VIEW" />
                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.BROWSABLE" />
                <data android:scheme="https"
                    android:host="subdomain.onelink.me"
                    android:pathPrefix="/ZBZ8" />
                <!-- Начало добавленного кода -->
                <data android:scheme="https"
                    android:host="click.your-domain.me" />
                <!-- Конец добавленного кода -->
            </intent-filter>
        </activity>
    </application>
</manifest>
```

3.19.6.3.4 Для iOS

1. Перейдите в Xcode;
2. Нажмите на свой проект;
3. Нажмите на таргет проекта (показано на скриншоте снизу);
4. Перейдите на вкладку **Capabilities**;
5. Активируйте переключатель *Associated domains*;
6. Добавьте в список доменов новую ссылку с брендированным доменом. К ней нужно добавить префикс `applinks`, после чего ссылка будет выглядеть так – `applinks:click.your-domain.me`:



7. Вернитесь в файл `AppDelegate.swift` к методу `UIApplicationDelegate:application:continue:restorationHandler` и добавьте ваш брендированный домен в массив `hosts`:

Список 17: Swift

```

/// Если приложение открыто через диплинк и содержит параметр pid=advcake_int
let hosts = [
    "yourapp.onelink.me",
    "app.yourbrand.com", // Ваш брендированный домен
]
if userActivity.activityType == NSUserActivityTypeBrowsingWeb,

```

(continues on next page)

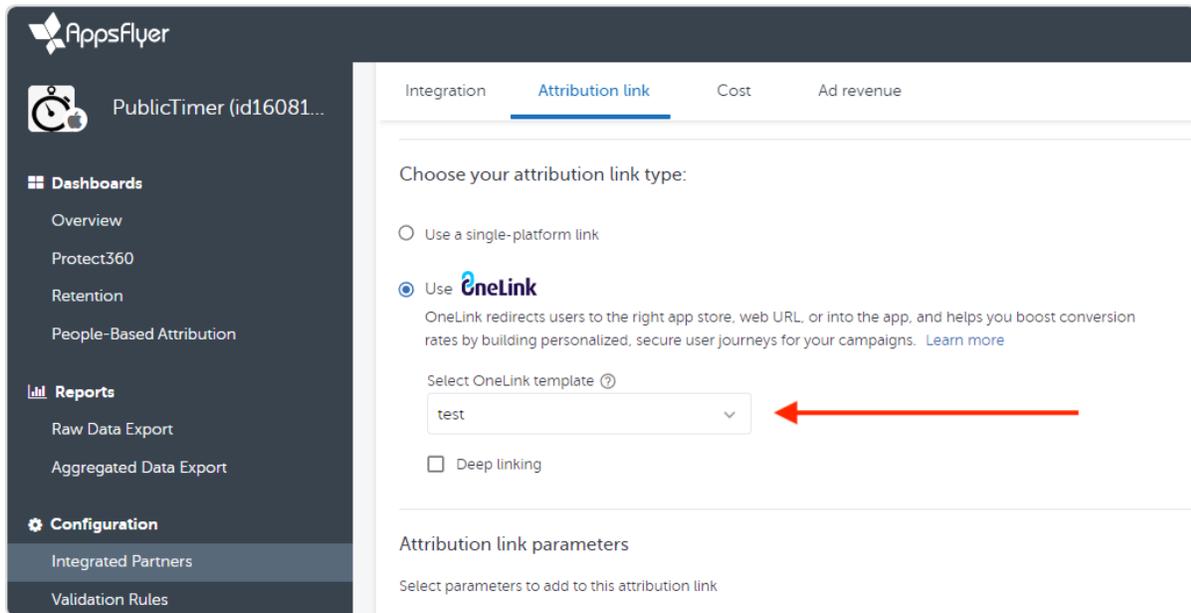
(продолжение с предыдущей страницы)

```
let incomingURL = userActivity.webpageURL,
let components = NSURLComponents(url: incomingURL, resolvingAgainstBaseURL: true),
let params = components.queryItems,
let host = components.host,
hosts.contains(host),
params.first(where: { $0.name == "pid" })?.value == "advcake_int"
{
UserDefaults.standard.set(true, forKey: "advcakeDeeplink")
}
```

3.19.7 Шаг 7. Проведение тестов

После настройки поведения ссылок, добавления Adv.Cake как рекламной сети и настройки брендированных доменов необходимо создать тестовую ссылку в панели администратора AppsFlyer.

1. В панели администратора перейдите в раздел **Configuration** → **Integrated Partners** → **Adv.Cake** → **Attribution link** → **Use OneLink**;



2. Выберите созданный ранее шаблон и внизу страницы скопируйте ссылку из поля «Click attribution link».

Ссылка будет выглядеть так:

```
https://your-domain.onelink.me/BQt3?pid=advcake_int&c=affiliate&af_siteid=testweb&is_
↔retargeting=true&af_reengagement_window=30d&af_click_lookback=7d
```

Предупреждение: На данном этапе необходимо подменить домен на брендированный! Таким образом, ссылка, из примера выше, изменится на:

```
https://click.your-domain.me/BQt3?pid=advcake_int&c=affiliate&af_siteid=testweb&is_
↔retargeting=true&af_reengagement_window=30d&af_click_lookback=7d
```

- Лишние GET-параметры можно удалить, так как для теста они не понадобятся.

Подсказка: Подробнее о настройке брендированных доменов читайте в *этом разделе*.

3. Перейдите по ссылке с ПК – должна открыться веб-версия вашего приложения. Если этого не произошло – проверьте настройки OneLink-шаблона.
4. Отправьте ссылку на мобильные устройства для тестирования;
5. Удалите приложения с мобильных устройств и перейдите по ссылке. Если после перехода открылась веб-версия вашего приложения – это значит, что всё настроено правильно. В противном

случае, перепроверьте настройки OneLink-шаблона;

6. Установите приложения на мобильные устройства и перейдите по ссылке. Ваши приложения должны перехватывать ссылки и открывать их изнутри. Если этого не происходит – проверьте, все ли правки из предыдущего раздела вы внесли в приложение;
7. Перейдите по новой ссылке и совершите тестовую покупку в приложении. Должен выполняться код, который вы настраивали *ранее*.

Чтобы посмотреть события – перейдите в раздел **Dashboards** → **Events** (обычно события подгружаются в течение часа):

Если события не появились – возможно, вы неправильно настроили отправку событий из приложения. Обратите внимание на источник, за которым закрепилось событие.

Если событие пришло как **organic** это значит, что вы не настроили ретаргетинг.

Если источник – **advcake_int** это значит, что ретаргетинг настроен правильно.

The screenshot shows the AppsFlyer interface for the 'PublicTimer (id...)' app. The left sidebar contains a navigation menu with categories: Dashboards, Reports, and Experiences & Deep Linking. The 'Events' option is highlighted in the sidebar. The main content area displays the 'Events Drill Down' section, which includes a table of 'All Media Sources'. The table has columns for 'Media Source', 'In-apps events', and 'U'. The '222' value in the 'U' column is highlighted, and a red arrow points from it to the 'Events' menu item in the sidebar.

Media Source	In-apps events	U
organic	start_button_tapped	
	af_proc_random	
advcake_int	start_button_tapped	
	af_purchase	
	af_proc_random	
222		
	start_button_tapped	

3.20 Инструкция AppsFlyer (инсталлы)

Эта инструкция поможет вам настроить интеграцию мобильных приложений через AppsFlyer — платформы для мобильной атрибуции и маркетинговой аналитики.

Данный тип интеграции настраивается для того, чтобы установки и соответствующие им заказы, созданные через мобильное приложение, корректно учитывались в канале CPI.

Важно понимать, что при интеграции через инсталлы, мы отслеживаем только покупки, совершенные после установки приложения и рекламодатель оплачивает только установку, подтвержденную заказом.

Содержание:

1. *Добавление приложения;*
2. *Установка SDK AppsFlyer;*
3. *Интеграция SDK AppsFlyer;*
4. *Настройка отправляемых из приложения событий;*
5. *Настройки панели администратора AppsFlyer;*
6. *Настройка брендированных доменов.*

3.20.1 Шаг 1. Добавление приложения (общее)

Примечание: Если у вас еще нет аккаунта AppsFlyer, зарегистрируйте его .

После регистрации аккаунта необходимо добавить ваше приложение в AppsFlyer:

1. Перейдите на ;
2. Нажмите кнопку *+ Add app* и введите данные приложения;
3. В пункте *Select the app type* выберите тип приложения (в нашем случае – мобильное) (1);
4. В пункте *Enter app details* выберите платформу (Android/iOS) и статус приложения (опубликовано / в разработке) (2)
 - В поле «App Store URL» необходимо прикрепить ссылку из маркетплейса на приложение, если оно уже опубликовано. Если нет – необходимо ввести «Apple App ID» для iOS или «Android package name» для Android;
5. Выберите часовой пояс и валюту (3).

Add an app ⓘ

Add an app to get your data flowing

1 Select the app type (Mobile)

2 Enter app details

Select platform

Android iOS Windows

Select App Store status

In store Pending approval/not published

App Store URL (required) ⓘ

https://apps.apple.com/ru/app/macfamilytree-10/id...

Next

3 Set time zone and currency

3.20.2 Шаг 2. Установка SDK AppsFlyer

Чтобы работать через AppsFlyer необходимо интегрировать SDK AppsFlyer в приложение.

3.20.2.1 Для iOS

Установить SDK AppsFlyer для iOS можно несколькими способами:

3.20.2.1.1 Через Cocoapods

1. Установите ;
2. Добавьте зависимости – необходимо добавить последнюю версию AppsFlyerFramework в Podfile проекта с помощью команды:

```
pod 'AppsFlyerFramework'
```

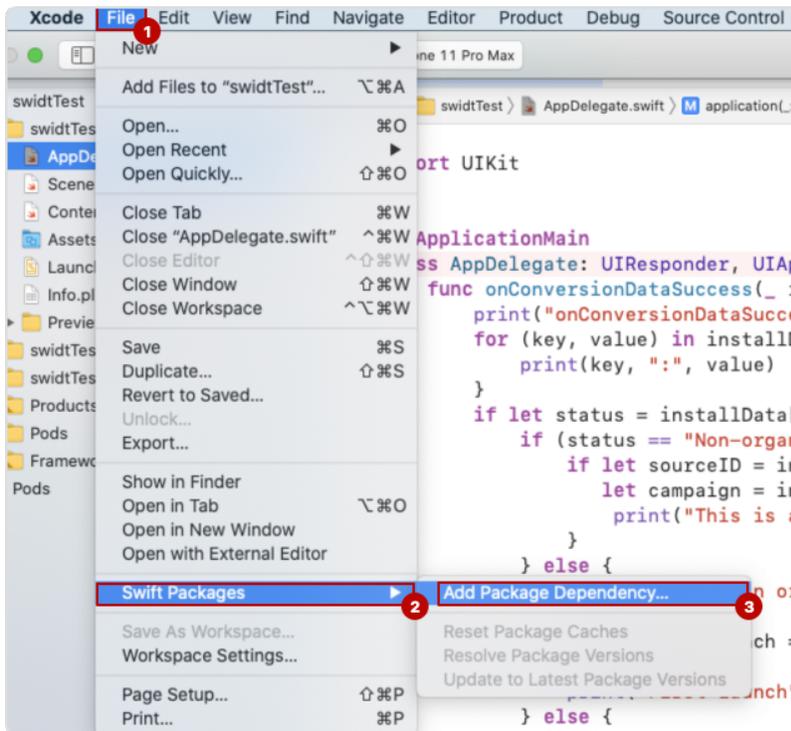
3. Установите зависимости с помощью команды:

```
pod 'install'
```

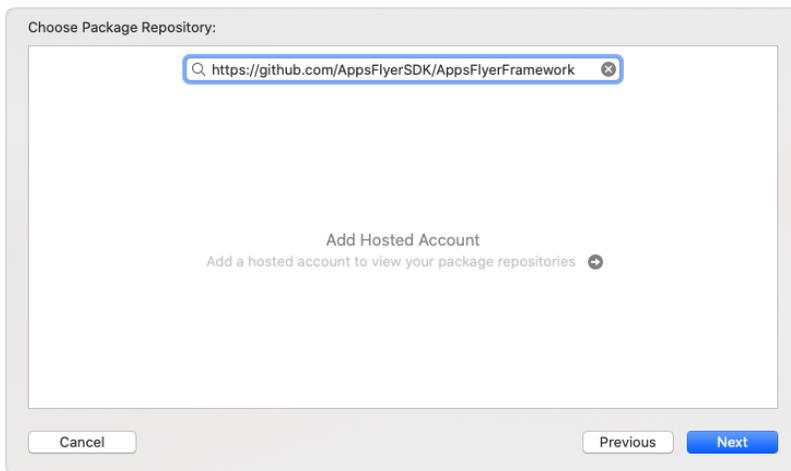
4. После установки необходимо использовать .xcworkspace файл для открытия проекта, вместо .xcodeproj.

3.20.2.1.2 Через SwiftPackageManager (рекомендуется)

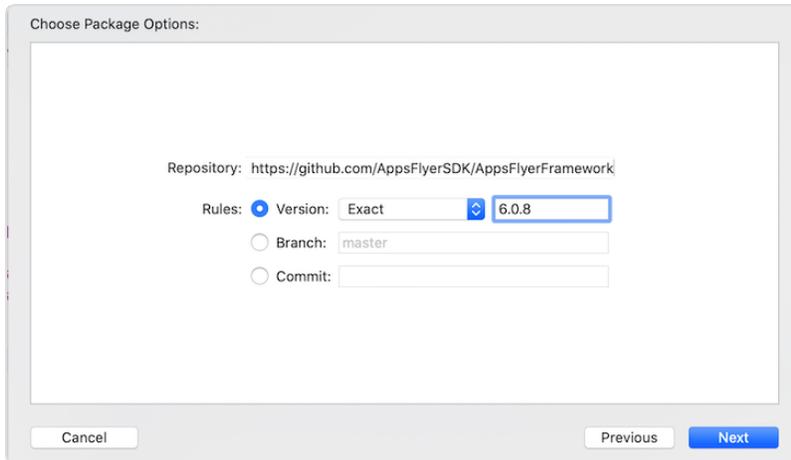
1. В Xcode перейдите по пути: **File** -> **Swift Packages** -> **Add Package Dependency**;



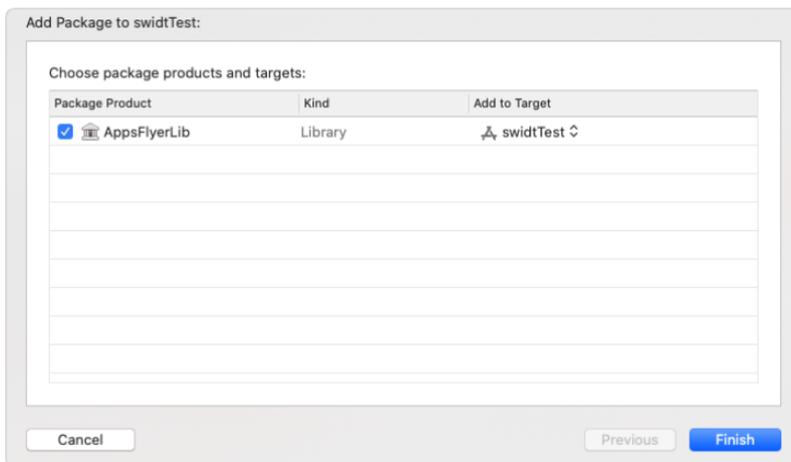
2. Добавьте IOS SDK репозиторий AppsFlyer (<https://github.com/AppsFlyerSDK/AppsFlyerFramework>);



3. Выберите версию SDK – Ехаст (текущая);



4. Добавьте AppsFlyerLib к нужному таргету;



3.20.2.2 Для Android

Установить SDK AppsFlyer для Android можно также несколькими способами:

3.20.2.2.1 Установка с помощью Gradle

1. Добавьте репозиторий `mavenCentral()` в файл `build.gradle`:

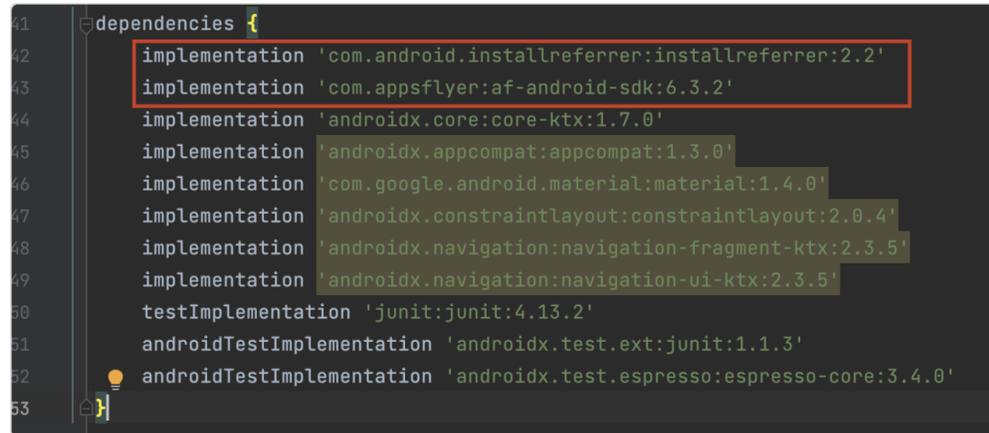
```

37     repositories {
38         mavenCentral()
39     }

```

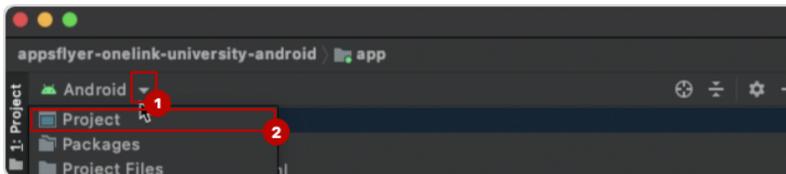
2. Добавьте зависимости в файл `build.gradle` – здесь можно сразу добавить зависимость `installreferrer`, поскольку она понадобится нам в дальнейшем:

```
implementation 'com.android.installreferrer:installreferrer:2.2'  
implementation 'com.appsflyer:af-android-sdk:6.3.2'
```

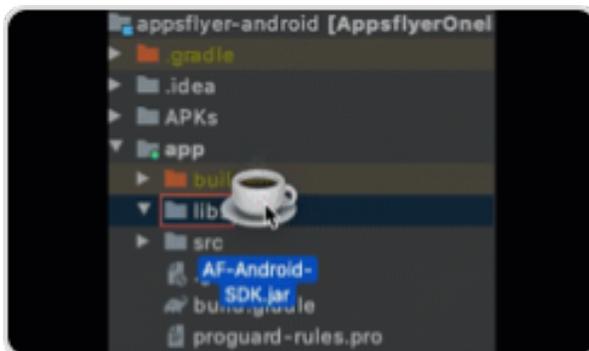


3.20.2.2.2 Установка без Gradle

1. Переведите структуру проекта из **Android** в **Project**;



2. Скачайте ;
3. Переместите скачанный SDK в `app/libs`;



3. Правой кнопкой мыши кликните по `jar`-файлу и нажмите *+ Add as Library*, после чего кликните повторно и нажмите *Refactor*;
4. После установки необходимо добавить разрешения в файл `AndroidManifest.xml`:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="com.google.android.gms.permission.AD_ID" />
```

Предупреждение: Если установка была выполнена без Gradle, необходимо добавить зависимость ниже (для установки с помощью Gradle мы добавили ее ранее) в `build.gradle`:

```
implementation 'com.android.installreferrer:installreferrer:2.2'
```

Далее перейдем к интеграции SDK в приложение.

3.20.3 Шаг 3. Интеграция SDK AppsFlyer

Примечание: Перед началом интеграции SDK AppsFlyer необходимо узнать ключ разработчика (в панели AppsFlyer перейдите в раздел **Configuration** → **App Settings** → поле «Dev key») и Apple App ID (в панели перейдите в раздел **Мои приложения** → выберите ваше приложение → **Информация о приложении** → поле «Apple ID»).

3.20.3.1 Swift

1. Импортируйте AppsFlyerLib:

```
import AppsFlyerLib
```

2. Инициализируйте SDK AppsFlyer:

```
AppsFlyerLib.shared().appsFlyerDevKey="your-key"
AppsFlyerLib.shared().appleAppID="your-id"
```

3. Вызовите `AppsFlyerLib.shared().start;`

Пример:

```
10 import AppsflyerLib
11
12
13 @main
14 class AppDelegate: UIResponder, UIApplicationDelegate, AppsflyerLibDelegate {
15     func onConversionDataSuccess(_ conversionInfo: [AnyHashable : Any]) {
16         print("success data conversion")
17     }
18
19     func onConversionDataFail(_ error: Error) {
20         print("fail data conversion")
21     }
22
23     var window: UIWindow?
24
25     func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
26     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
27         AppsflyerLib.shared().appsFlyerDevKey = "your-key"
28         AppsflyerLib.shared().appleAppID = "your-id"
29         AppsflyerLib.shared().delegate = self
30
31         window = UIWindow(frame: UIScreen.main.bounds)
32         window?.rootViewController = ViewController()
33         window?.makeKeyAndVisible()
34
35         return true
36     }
37
38     @objc func sendLaunch() {
39         AppsflyerLib.shared().start()
40     }
41 }
```

После этого, добавляем поддержку AppTrackingTransparency.

4. Перед вызовом `start()`, добавьте `waitForATTUserAuthorization(timeoutInterval: 60)` в `didFinishLaunchingWithOptions`;

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    AppsflyerLib.shared().waitForATTUserAuthorization(timeoutInterval: 60)
    AppsflyerLib.shared().appsFlyerDevKey = "your-key"
    AppsflyerLib.shared().appleAppID = "your-id"
    AppsflyerLib.shared().delegate = self

    window = UIWindow(frame: UIScreen.main.bounds)
    window?.rootViewController = ViewController()
    window?.makeKeyAndVisible()

    return true
}
```

5. Запросите одобрение пользователя в `applicationDidBecomeActive()`;

```
func applicationDidBecomeActive(_ application: UIApplication) {
    if #available(iOS 14, *) {
        ATTrackingManager.requestTrackingAuthorization { status in
            switch status {
            case .authorized:
                // Tracking authorization dialog was shown
                // and we are authorized
                print("Authorized")
            case .denied:
                // Tracking authorization dialog was
                // shown and permission is denied
                print("Denied")
            case .notDetermined:
                // Tracking authorization dialog has not been shown
                print("Not Determined")
            case .restricted:
                print("Restricted")
            @unknown default:
                print("Unknown")
            }
        }
    }
}
```

3.20.3.2 Kotlin

1. Импортируйте AppsFlyerLib:

```
import com.appsflyer.AppsFlyerLib
```

2. Инициализируйте SDK AppsFlyer:

```
AppsFlyerLib.getInstance().init("your-key", null, this)
```

3. Запустите Android SDK:

```
AppsFlyerLib.getInstance().start(this)
```

Полный пример:

```
import com.appsflyer.AppsFlyerLib

class MainActivity : AppCompatActivity() {

    private lateinit var appBarConfiguration: AppBarConfiguration
    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        AppsFlyerLib.getInstance().init("your-key", null, this)
        AppsFlyerLib.getInstance().start(this)
    }
}
```

3.20.3.3 React Native

1. Импортируйте appsFlyer из react-native-appsflyer:

Список 18: TypeScript

```
import appsFlyer from 'react-native-appsflyer';
```

Список 19: JavaScript

```
import appsFlyer from 'react-native-appsflyer';
```

2. Инициализируйте SDK AppsFlyer:

Список 20: TypeScript

```
appsFlyer.initSdk({
  devKey: 'K2*****99',
  isDebug: false,
  appId: '41*****44',
  onInstallConversionDataListener: true, //Optional
  onDeepLinkListener: true, //Optional
  timeToWaitForATTUserAuthorization: 10 //for iOS 14.5
});
```

Список 21: JavaScript

```
appsFlyer.initSdk({
  devKey: 'K2*****99',
  isDebug: false,
  appId: '41*****44',
  onInstallConversionDataListener: true, //Optional
  onDeepLinkListener: true, //Optional
```

(continues on next page)

(продолжение с предыдущей страницы)

```
timeToWaitForATTUserAuthorization: 10 //for iOS 14.5
});
```

После интеграции SDK в приложение перейдем к настройке событий.

3.20.4 Шаг 4. Настройка отправляемых из приложения событий

Для начала необходимо определить имя события для отправки. В нашем примере событием является тап по кнопке «Старт», поэтому мы назвали его «start_button_tap». В реальном приложении события могут называться по-разному, например: «purchase», «level_up», «bet» и т.д.

После того, как вы определитесь с именем события для отправки, необходимо подготовить массив (*map/dictionary*) `withValues`, который будет содержать следующие данные о событии:

1. Сумма заказа/доход. В качестве ключа используется константа `AFInAppEventParameterName.REVENUE` (Kotlin) или `AFFEventParamRevenue` (Swift), либо строка `af_revenue` (React Native). Опционально для Android.
2. ID заказа или события в системе рекламодателя. Ключ «order_id» или любой другой (по согласованию). **Обязательно для передачи!**
3. Другие данные, которые вы готовы нам предоставить для атрибутирования заказов, например, категория заказа («category»), тип клиента («clientType») или промокод («coupon»). Опционально.

Далее, необходимо вызвать функцию `logEvent()` из `AppsFlyerLib`, передав ей имя события и массив с данными.

Предупреждение: Реализация интеграции для iOS на этом не заканчивается – смотрите раздел *Реализация на iOS (Swift, React Native)*.

3.20.4.1 Реализация на Kotlin

Пример реализации:

```
val eventData = HashMap<String, Any>()
eventData.put("orderId", orderId)
eventData.put(AFInAppEventParameterName.REVENUE, revenue)
eventData.put("category", category)
eventData.put("clientType", clientType)
eventData.put("coupon", coupon)

AppsFlyerLib.getInstance().logEvent(context, "start_button_tap", eventData)
```

3.20.4.2 Реализация на React Native

Пример реализации:

Список 22: TypeScript

```
appsFlyer.logEvent("purchase", {
  orderId: orderId,
  af_revenue: revenue,
  category: category,
  clientType: clientType,
  coupon: coupon,
});
```

Список 23: JavaScript

```
appsFlyer.logEvent("purchase", {
  orderId: orderId,
  af_revenue: revenue,
  category: category,
  clientType: clientType,
  coupon: coupon,
});
```

3.20.4.3 Реализация на iOS (Swift, React Native)

Реализовать интеграцию AppsFlyer на платформе iOS значительно сложнее, поскольку есть две проблемы:

1. Постбеки событий, отправленных с iOS (независимо от использованной реализации AppsFlyer SDK), приходят с крайне ограниченным количеством данных, а именно: из нетрекинговых данных (данные, которые отправляются через `logEvent()` мы получаем только «af_revenue». Остальные нестандартные поля нам недоступны (в том числе «orderId»)).
2. Постбек события установки приложения (первого запуска приложения) не содержит AppsFlyerID – случайный ID, генерируемый SDK при первом запуске приложения. Из-за этого мы не сможем связывать покупки (или другие действия) с событием установки (нам важны трекинговые метки события установки).

Однако эти препятствия можно обойти.

3.20.4.3.1 Отправка события покупки

Как мы уже писали, нам доступно исключительно поле «af_revenue» – строковое поле, в котором AppsFlyer ожидает числовое значение.

Поле передаётся в постбеке без изменений, а при обработке числа на своей стороне AppsFlyer обрезает его до 5 знаков после запятой. Благодаря этому мы можем использовать это поле для хранения данных, если предварительно закодировать их в числовую последовательность.

Для этих целей мы разработали алгоритм, реализацию которого вы можете увидеть в примерах ниже.

Список 24: Swift

```

class NumSeqEncoder {
    /// Encodes byte array into one large numeric string.
    ///
    /// - Parameter bytes: Byte array with binary data.
    /// - Returns: String with a long number.
    static func encode(_ bytes: [UInt8]) -> String {
        let size = bytes.count

        var string: String = ""
        string.reserveCapacity(Int(ceil(Double(size) * 8 / 3)))

        var buffer: UInt32 = 0
        var bufferSize = 0
        var pos = 0

        while pos < size {
            let pickBytes = min(size - pos, bufferSize != 0 ? 3 : 4)

            buffer = buffer << (pickBytes * 8)
            for y in 0..

```

(continues on next page)

(продолжение с предыдущей страницы)

```

/// Encodes order data into a string as a long float number.
///
/// - Parameters:
///   - data: Data to encode. Allowed up to 1441 bytes (not characters!).
/// - Returns: String with a long float number; up to 3851 characters/bytes.
static func encodeToRevenue(_ data: String) -> String {
    let dataBytes = Array(data.utf8)
    // 3852 is max known length for Appsflyer's revenue field (it's probably greater_
↳though).
    if Int(ceil(Double(dataBytes.count) * 8 / 3)) > 3843 {
        return "0.000000"
    }
    return "0.000000" + encode(dataBytes)
}

/// Decodes binary data from a string with a long number.
///
/// - Parameter string: String with a long number.
/// - Returns: Byte array with binary data.
static func decode(_ string: String) -> [UInt8] {
    let bytes: [UInt8] = Array(string.utf8)

    var data = [UInt8](repeating: 0, count: bytes.count * 4 / 8)
    var bitIndex = 0

    for i in 0..

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    }
    let dataEncoded = String(string[string.index(range!.lowerBound, offsetBy: 7)...])
    let dataDecoded = decode(dataEncoded)
    return String(bytes: dataDecoded, encoding: .utf8)
  }
}

```

Список 25: TypeScript

```

class NumSeqEncoder {
  /**
   * Encodes binary string into one large numeric string.
   * @param {string} data String with binary data.
   * @returns {string} String with a long number.
   */
  static encode(data: string): string {
    const bytes = (new TextEncoder().encode(data));
    const size = bytes.length;

    let string = "";

    let buffer = 0;
    let bufferSize = 0;
    let pos = 0;

    while (pos < size) {
      let pickBytes = Math.min(size - pos, bufferSize != 0 ? 3 : 4);

      buffer = buffer << (pickBytes * 8);
      for (let i = 0; i < pickBytes; ++i) {
        buffer |= (bytes[pos + i] & ((1 << 8) - 1)) << ((pickBytes - i - 1) * 8);
      }
      bufferSize += pickBytes * 8;
      pos += pickBytes;

      while (bufferSize >= 3) {
        if (bufferSize >= 4) {
          const quad = buffer >> (bufferSize - 4) & ((1 << 4) - 1);
          if (quad === 0b1000 || quad === 0b1001) {
            string = string.concat(quad.toString());
            bufferSize -= 4;
            buffer &= (1 << bufferSize) - 1;
            continue;
          }
        }
        const triple = buffer >> (bufferSize - 3) & ((1 << 3) - 1);
        string = string.concat(triple.toString());
        bufferSize -= 3;
        buffer &= (1 << bufferSize) - 1;
      }
    }
  }
}

```

(continues on next page)

```

    if (bufferSize > 0) {
        string = string.concat((buffer << (3 - bufferSize)).toString());
    }

    return string;
}

/**
 * Encodes order data into a string as a long float number.
 * @param {string} data String to encode. Allowed up to 1441 bytes (not characters!).
 * @returns {string} String with a long float number; up to 3851 characters/bytes.
 */
static encodeToRevenue(data: string): string {
    const size = new TextEncoder().encode(data).length;
    // 3852 is max known length for Appsflyer's revenue field (it's probably greater
    →though).
    if (Math.ceil(size * 8 / 3) > 3843) {
        return "0.000000";
    }
    return "0.000000" + NumSeqEncoder.encode(data);
}

/**
 * Decodes binary data from a string with a long number.
 * @param {string} string String with a long number.
 * @returns {string} Decoded binary string.
 */
static decode(string: string): string {
    const bytes = (new TextEncoder().encode(string));
    const size = bytes.length;

    let data = new Uint8Array(size * 4 / 8);
    let bitIndex = 0;

    for (let i = 0; i < size; ++i) {
        if (bytes[i] < 48 || bytes[i] > 57) {
            break;
        }
        const number = bytes[i] - 48;
        let len = number >= 8 ? 4 : 3;
        for (let y = len - 1; y >= 0; --y) {
            const byteIndex = ~~(bitIndex / 8);
            const byteBitIndex = bitIndex % 8;
            data[byteIndex] = (data[byteIndex] || 0) | ((number >> y & 1) << (7 -
            →byteBitIndex));
            ++bitIndex;
        }
    }

    return new TextDecoder().decode(data.slice(0, Math.floor(bitIndex / 8)));
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

/**
 * Parses a string of data encoded as a long float number.
 * @param {string} string String with a long float number.
 * @returns {string|null} Decoded binary string. Null if string is invalid.
 */
static decodeFromRevenue(string: string): string | null {
    const pos = string.indexOf('.');
    if (pos === -1) {
        return null;
    }
    return NumSeqEncoder.decode(string.substring(pos + 7));
}
}

```

Список 26: JavaScript

```

"use strict";

class NumSeqEncoder {
    /**
     * Encodes binary string into one large numeric string.
     * @param {string} data String with binary data.
     * @returns {string} String with a long number.
     */
    static encode(data) {
        const bytes = (new TextEncoder()).encode(data);
        const size = bytes.length;

        let string = "";

        let buffer = 0;
        let bufferSize = 0;
        let pos = 0;

        while (pos < size) {
            let pickBytes = Math.min(size - pos, bufferSize !== 0 ? 3 : 4);

            buffer = buffer << (pickBytes * 8);
            for (let i = 0; i < pickBytes; ++i) {
                buffer |= (bytes[pos + i] & ((1 << 8) - 1)) << ((pickBytes - i - 1) * 8);
            }
            bufferSize += pickBytes * 8;
            pos += pickBytes;

            while (bufferSize >= 3) {
                if (bufferSize >= 4) {
                    const quad = buffer >> (bufferSize - 4) & ((1 << 4) - 1);
                    if (quad === 0b1000 || quad === 0b1001) {
                        string = string.concat(quad.toString());
                        bufferSize -= 4;
                        buffer &= (1 << bufferSize) - 1;
                        continue;
                    }
                }
            }
        }
    }
}

```

(continues on next page)

```

        }
    }
    const triple = buffer >> (bufferSize - 3) & ((1 << 3) - 1);
    string = string.concat(triple.toString());
    bufferSize -= 3;
    buffer &= (1 << bufferSize) - 1;
}
}

if (bufferSize > 0) {
    string = string.concat((buffer << (3 - bufferSize)).toString());
}

return string;
}

/**
 * Encodes order data into a string as a long float number.
 * @param {string} data String to encode. Allowed up to 1441 bytes (not characters!).
 * @returns {string} String with a long float number; up to 3851 characters/bytes.
 */
static encodeToRevenue(data) {
    const size = new TextEncoder().encode(data).length;
    // 3852 is max known length for Appsflyer's revenue field (it's probably greater
    →though).
    if (Math.ceil(size * 8 / 3) > 3843) {
        return "0.000000";
    }
    return "0.000000" + NumSeqEncoder.encode(data);
}

/**
 * Decodes binary data from a string with a long number.
 * @param {string} string String with a long number.
 * @returns {string} Decoded binary string.
 */
static decode(string) {
    const bytes = (new TextEncoder().encode(string));
    const size = bytes.length;

    let data = new Uint8Array(size * 4 / 8);
    let bitIndex = 0;

    for (let i = 0; i < size; ++i) {
        if (bytes[i] < 48 || bytes[i] > 57) {
            break;
        }
    }
    const number = bytes[i] - 48;
    let len = number >= 8 ? 4 : 3;
    for (let y = len - 1; y >= 0; --y) {
        const byteIndex = ~(bitIndex / 8);
        const byteBitIndex = bitIndex % 8;

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        data[byteIndex] = (data[byteIndex] || 0) | ((number >> y & 1) << (7 -
↪byteBitIndex));
        ++bitIndex;
    }
}

return new TextDecoder().decode(data.slice(0, Math.floor(bitIndex / 8)));
}

/**
 * Parses a string of data encoded as a long float number.
 * @param {string} string String with a long float number.
 * @returns {string|null} Decoded binary string. Null if string is invalid.
 */
static decodeFromRevenue(string) {
    const pos = string.indexOf('.');
    if (pos === -1) {
        return null;
    }
    return NumSeqEncoder.decode(string.substring(pos + 7));
}
}

```

Необходимо отправлять два события: «purchase» и «advcake_purchase». Оба должны содержать данные о заказе, но второй должен содержать закодированное «revenue» вместо настоящей суммы дохода/цены.

Включите в своё приложение класс NumSeqEncoder. При создании пользователем заказа сформируйте массив с данными о заказе и дополнительно добавьте в него поле «appsflyerId» со значением функции AppsFlyerLib.shared().getAppsFlyerUID() (appsFlyer.getAppsFlyerUID()). Затем сформируйте такой же массив (либо переиспользуйте старый) и замените в нём поле AFEventParamRevenue (af_revenue) на результат функции NumSeqEncoder.encodeToRevenue(json) (строка), передав ему в качестве аргумента этот же массив с данными о заказе, закодированный в JSON-строку.

Смотрите пример реализации ниже.

Примечание: Обратите внимание, что размер JSON-строки с данными не должен превышать 1441 байт (не символов), что равняется <=3851 байт в закодированном виде. Это ограничение AppsFlyer, выявленное путём проведения крайне ограниченного числа тестов. Вполне вероятно, что допустимы строки большего размера. В ближайшее время информация будет уточнена, а инструкция и код обновлены.

Если переданная в функцию encodeToRevenue() строка будет превышать описанное ограничение, данные закодированы не будут (функция вернёт «0.000000»).

Поэтому перед добавлением в JSON очередного поля убедитесь, что суммарная максимальная длина значений всех полей, а также длина ключей и токенов не превышает описанного ограничения.

Пример реализации:

Список 27: Swift

```

struct OrderInfo: Codable {
    let appsflyerId: String
    let orderId: String
    let revenue: Double
    let category: String
    let clientType: String
    let coupon: String?

    var dictionary: [String: Any?] {
        return [
            "appsflyerId": appsflyerId,
            "orderId": orderId,
            "revenue": revenue,
            "category": category,
            "clientType": clientType,
            "coupon": coupon,
        ]
    }
}

//...

let orderInfo = OrderInfo(
    appsflyerId: AppsFlyerLib.shared().getAppsFlyerUID(),
    orderId: orderId,
    revenue: revenue,
    category: category,
    clientType: clientType,
    coupon: coupon
)

var afEventValues = orderInfo.dictionary
afEventValues[AfEventParamRevenue] = afEventValues["revenue"]
AppsFlyerLib.shared().logEvent("purchase", withValues: afEventValues)

let orderInfoJsonData = try JSONEncoder().encode(orderInfo)
let orderInfoJsonStr = String(data: orderInfoJsonData, encoding: .utf8) ?? ""
afEventValues[AfEventParamRevenue] = NumSeqEncoder.encodeToRevenue(orderInfoJsonStr)
AppsFlyerLib.shared().logEvent("advcake_purchase", withValues: afEventValues)

```

Список 28: TypeScript

```

const eventValues = {
    appsflyerId: appsFlyer.getAppsFlyerUID(),
    orderId: orderId,
    af_revenue: revenue,
    category: category,
    clientType: clientType,
    coupon: coupon,
};

```

(continues on next page)

(продолжение с предыдущей страницы)

```

const eventValuesAdvcake = {...eventValues};
eventValuesAdvcake.af_revenue = NumSeqEncoder.encodeToRevenue(JSON.
↳stringify(eventValues));

appsFlyer.logEvent("purchase", eventValues);
appsFlyer.logEvent("advcake_purchase", eventValuesAdvcake);

```

Список 29: JavaScript

```

const eventValues = {
  appsflyerId: appsFlyer.getAppsFlyerUID(),
  orderId: orderId,
  af_revenue: revenue,
  category: category,
  clientType: clientType,
  coupon: coupon,
};

const eventValuesAdvcake = {...eventValues};
eventValuesAdvcake.af_revenue = NumSeqEncoder.encodeToRevenue(JSON.
↳stringify(eventValues));

appsFlyer.logEvent("purchase", eventValues);
appsFlyer.logEvent("advcake_purchase", eventValuesAdvcake);

```

3.20.4.3.2 Отправка дополнительного события установки

Дополнительное событие установки необходимо отправлять при каждом запуске приложения. Таким образом, событие можно отправлять сразу после запуска AppsFlyer SDK.

Это событие никак не повлияет на статистику в AppsFlyer.

1. Включите в своё приложение класс `NumSeqEncoder`, как это описано в *предыдущем разделе*;
2. Реализуйте в своём приложении выполнение следующего кода при каждом запуске приложения на устройстве пользователя:

Список 30: Swift

```

AppsFlyerLib.shared().start()

let encodedRevenue = NumSeqEncoder.encodeToRevenue(AppsFlyerLib.shared().
↳getAppsFlyerUID())
let afEventValues = [
  AFEventParamRevenue: encodedRevenue,
]
AppsFlyerLib.shared().logEvent("advcake_install", withValues: afEventValues)

```

Список 31: JavaScript

```
appsFlyer.initSdk({
  devKey: 'K2*****99',
  isDebug: false,
  appId: '41****44',
  onInstallConversionDataListener: true, //Optional
  onDeepLinkListener: true, //Optional
  timeToWaitForATTUserAuthorization: 10 //for iOS 14.5
});

appsFlyer.logEvent("advcake_install", {
  af_revenue: NumSeqEncoder.encodeToRevenue(appsFlyer.getAppsFlyerUID())
});
```

Список 32: TypeScript

```
appsFlyer.initSdk({
  devKey: 'K2*****99',
  isDebug: false,
  appId: '41****44',
  onInstallConversionDataListener: true, //Optional
  onDeepLinkListener: true, //Optional
  timeToWaitForATTUserAuthorization: 10 //for iOS 14.5
});

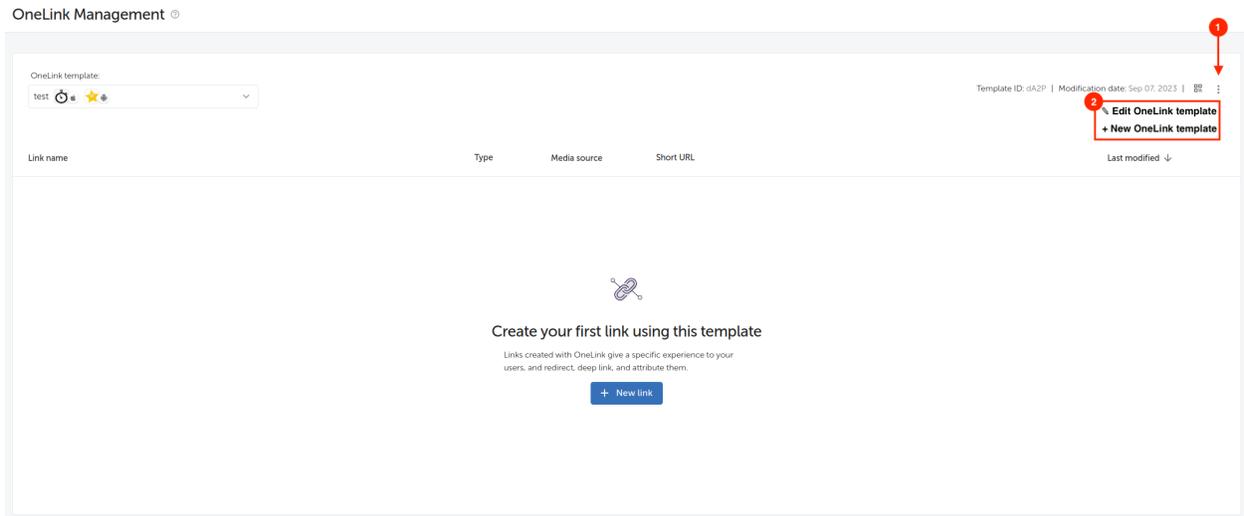
appsFlyer.logEvent("advcake_install", {
  af_revenue: NumSeqEncoder.encodeToRevenue(appsFlyer.getAppsFlyerUID())
});
```

3.20.5 Шаг 5. Настройки панели администратора AppsFlyer

3.20.5.1 Настройка OneLink-шаблона

Настройка OneLink-шаблона нужна для того, чтобы пользователь при переходе по ссылке попадал в маркетплейс и мог установить приложение.

1. В панели администратора AppsFlyer перейдите в раздел **Experiences & Deep Linking** → вкладка **OneLink Custom Links**;
2. Нажмите кнопку «Три точки» и выберите пункт *New OneLink template* (или *Edit OneLink template* если шаблон уже имеется):



3. Задайте произвольный поддомен:

* Subdomain

Subdomain cannot be changed once there are custom links based on this template.

testtest1234 .onelink.me

Примечание: Позже этот домен будет использоваться для создания самих ссылок.

4. Выберите поведение ссылки:

В разделе «When app isn't installed» (если приложение **не установлено**) в соответствующих полях укажите:

- **iOS:** Redirect users to AppStore;
- **Android:** Redirect users to Google Play;

Раздел «When app is installed» (если приложение **установлено**) заполните произвольно.

OneLink Management > Update OneLink template

Update OneLink template

When app isn't installed

iOS	<p>Redirect users to App Store</p> <p>Change to a different URL</p> <p><input type="checkbox"/> Display App Clip for users with Private Relay. Learn more</p> <p>Note: Only turn this on if you have an App Clip configured.</p>
Android	<p>Redirect users to Google Play</p> <p>Redirect to third-party Android app stores ▼</p> <p>Or</p> <p>Change to a different URL</p>

Redirection when the app isn't installed

Send iOS or Android users to the app's page on the App Store or Google Play, or to another web page.

When app is installed

Configuring the following setup should be done in collaboration between the marketer and developer

iOS	<p>Launching the app using Universal links ↗</p> <p>Add test app</p>
Android	<p>Launching the app using Android App Links (1 SHA256 fingerprint) ↗</p> <p>Code snippet for AndroidManifest.xml ↗</p>
Android and iOS fallback	<p>Launching the app using URI scheme:</p> <p>tickertimer:// ↗</p>

Подсказка: Где найти «Apple Team ID» для iOS и SHA256 для Android?

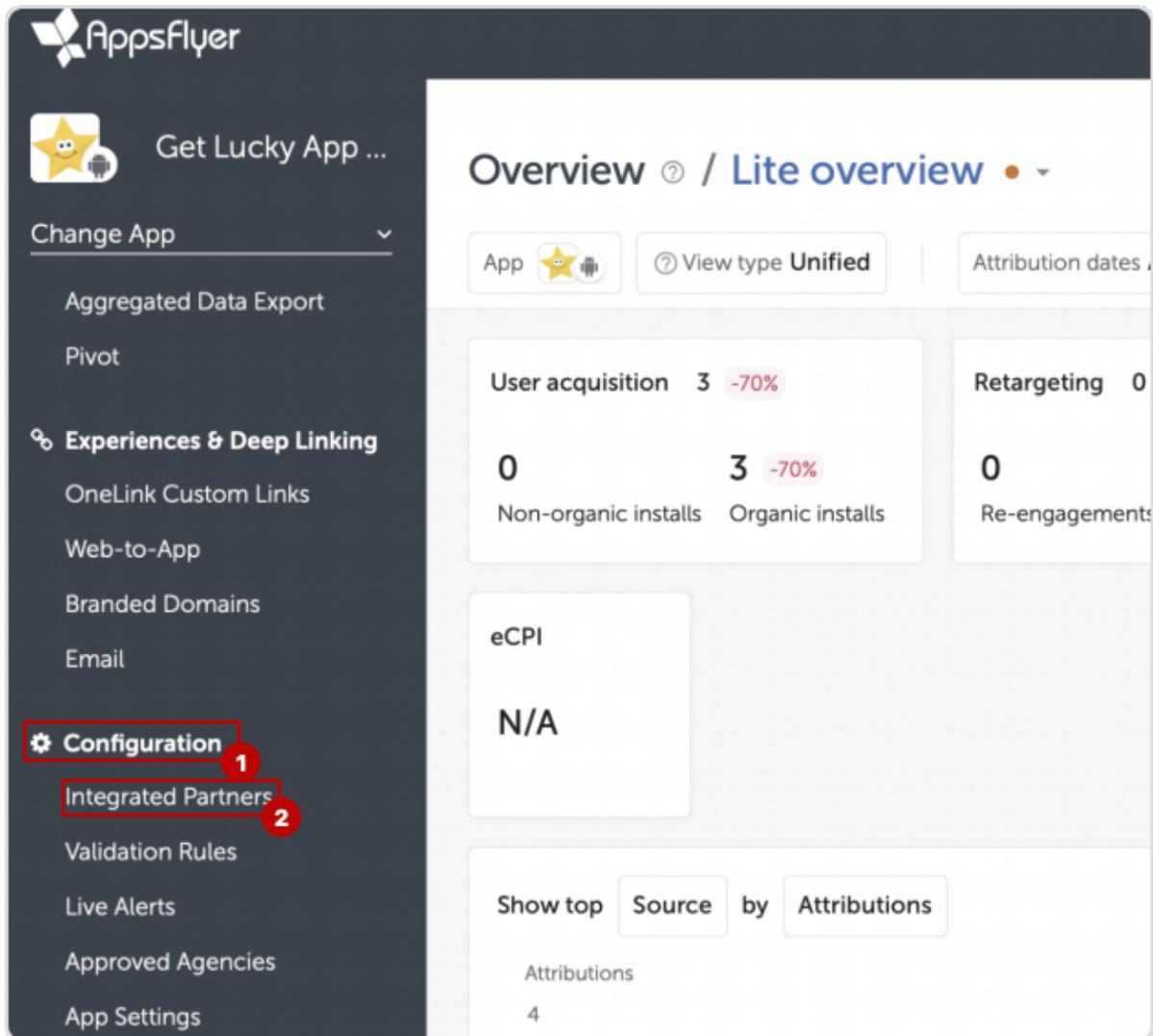
Чтобы найти «Apple Team ID», перейдите на и авторизируйтесь под учетной записью разработчика – в верхней части страницы (там, где ваши Ф.И.) «Имя компании» и будет означать «Apple Team ID».

Чтобы найти SHA256, перейдите на , затем, в меню слева, выберите «Настройка» -> «Целостность приложения» -> раздел «Сертификат ключа для подписи приложения». SHA256 будет указан в поле «Цифровой отпечаток сертификата SHA-256» .

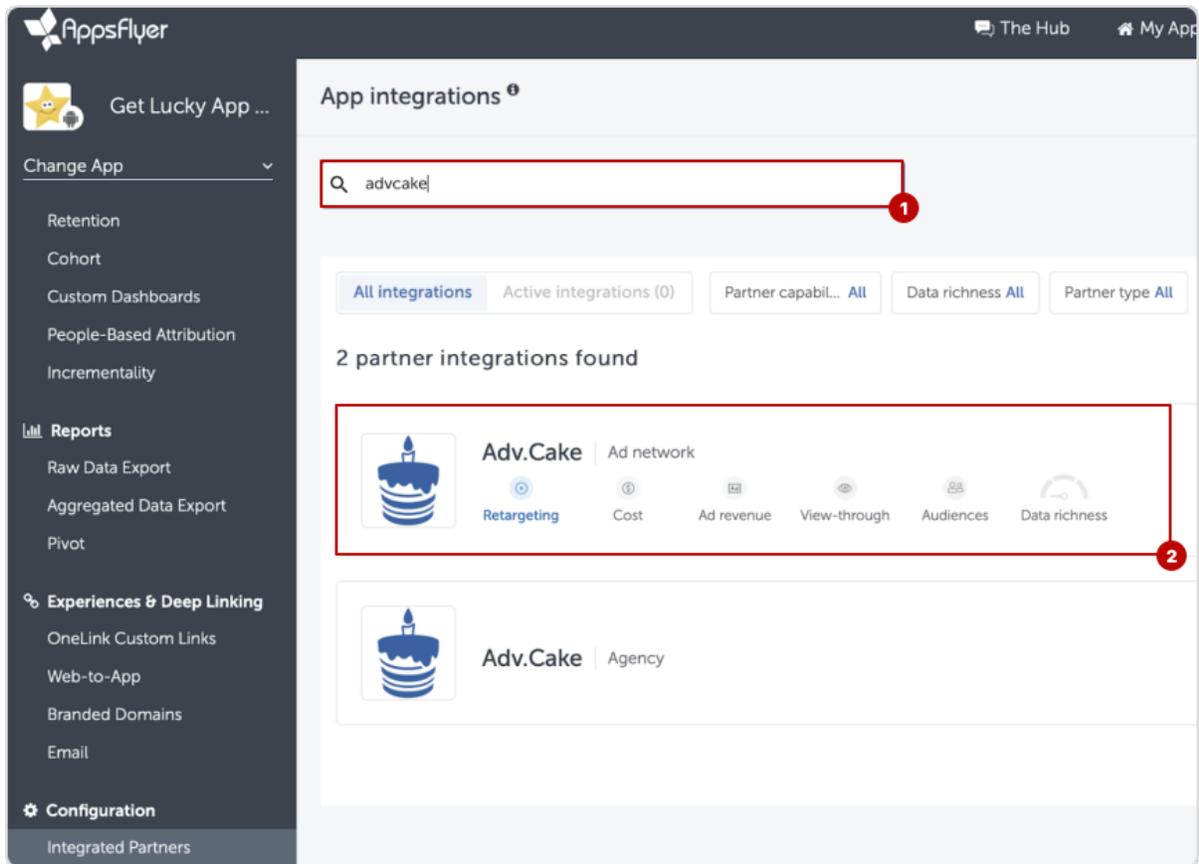
Поведение ссылок настроено, теперь добавим Adv.Cake как рекламную сеть:

3.20.5.2 Добавление Adv.Cake

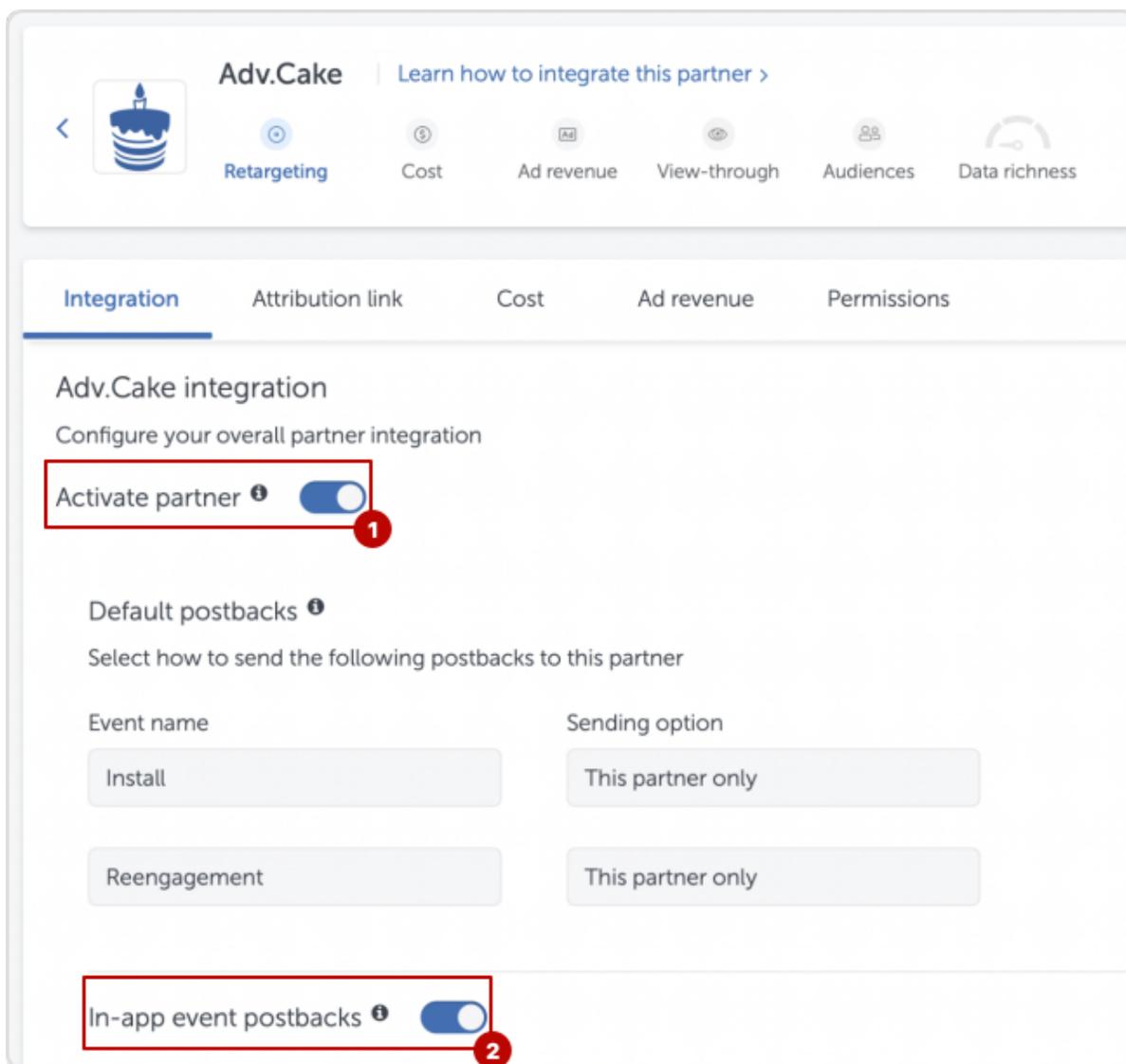
1. В панели администратора AppsFlyer перейдите в раздел **Configuration** -> **Integrated Partners**;



2. Вбейте в поиск Adv.Cake, после чего появятся два партнера – выберите с пометкой *Ad Network* (рекламная сеть):

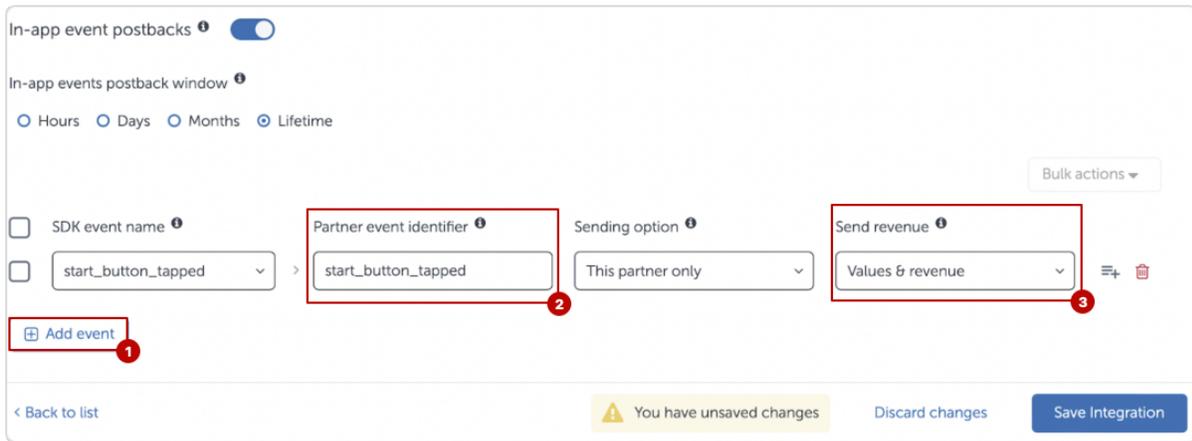


3. На вкладке **Integration** активируйте галочки «Activate partner» и «In-app event postbacks»;

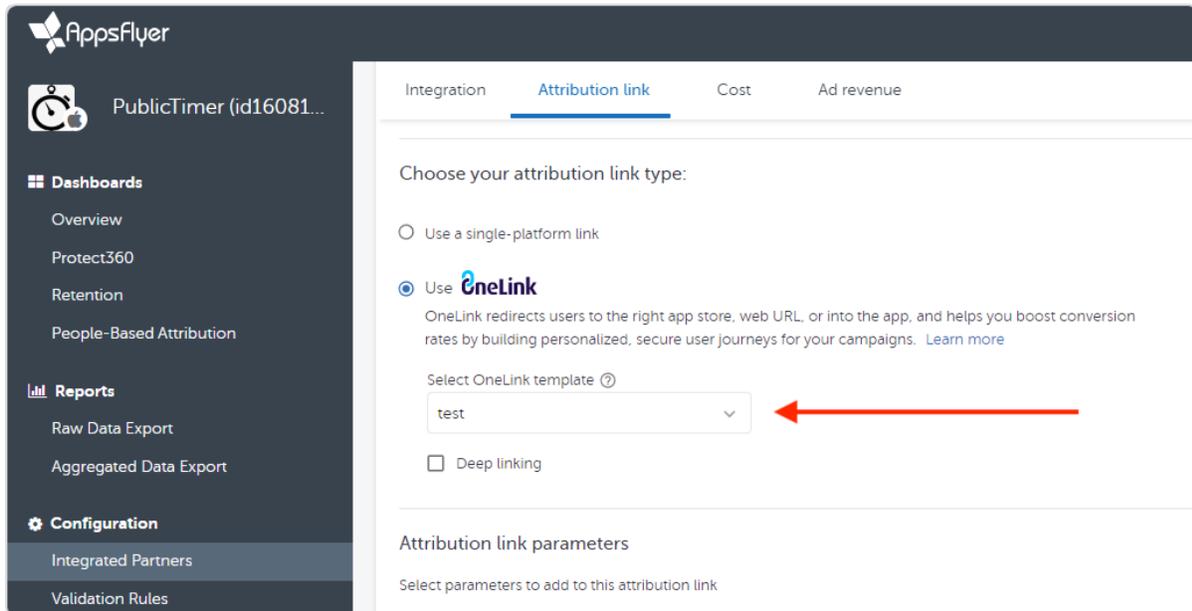


4. Затем, добавьте события, которые отправляются из нашей сети (настраивали *ранее*). Для этого, нажмите *Add event* и добавьте нужное событие:
 - 4.1. В «Partner event identifier» можно указать название события (например, «start_button_tapped»);
 - 4.2. В «Send revenue» – values & revenue.

Предупреждение: Обратите внимание, что в постбеках для iOS не будут приходить «values». Для этого, мы настраивали «revenue».



5. Перейдите на вкладку **Attribution link**, выберите шаблон в поле «Select OneLink template» (настраивали ранее) и активируйте галочку «Retargeting Settings» в нижней части страницы;



6. Перейдите на вкладку **Permissions** и установите галочки на указанных пунктах – это позволит самостоятельно настраивать ссылки. После чего, в нижней части страницы, нажмите *Save Permissions*.

[Интеграция](#)[Ссылка атрибуции](#)[Затраты](#)[Доход от рекламы](#)[Права доступа](#)

Доступ Adv.Cake

Выберите права для предоставления данному партнеру [Узнать больше >](#)

Доступ рекламной сети

Разрешите рекламной сети просматривать свой трафик по вашему приложению на их дэшборде.

Откройте доступ членам команды Adv.Cake

aa@advcake.com (admin)



[+ Добавить члена команды](#)

Предоставьте следующие права доступа членам команды Adv.Cake:

Разрешить конфигурировать интеграцию



Конфигурировать постбэки внутренних событий приложения



Открыть доступ к отчету об удержании



Открыть доступ к агрегированным данным лояльных пользователей



Access aggregate conversions data ⁱ



Открыть доступ к агрегированным данным внутренних событий приложения ⁱ



View validation rules



Доступ к агрегированным данным о доходе



Импорт данных о расходах ⁱ

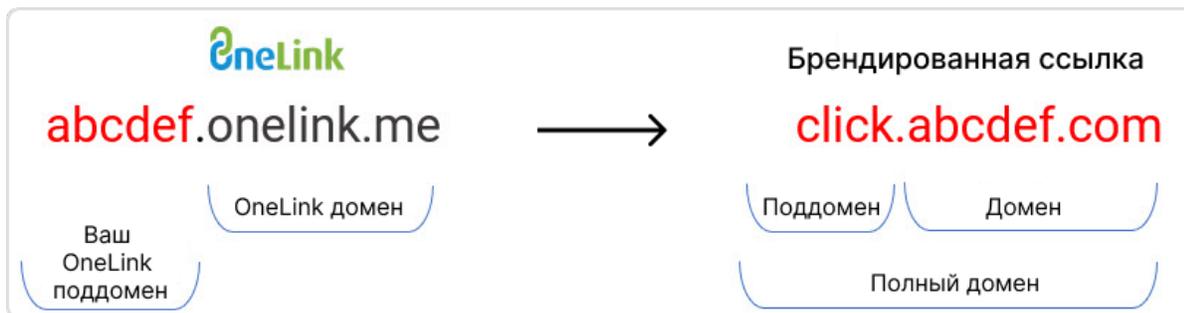


Открыть доступ к дэшборду Protect360 и к сырым данным ⁱ



3.20.6 Шаг 6. Настройка брендированных доменов

Брендированные домены позволяют указывать в ссылках атрибуции название бренда и домен.



Многие блокировщики рекламы и VPN-сервисы блокируют доступ к некоторым доменам, среди которых – onelink.me.

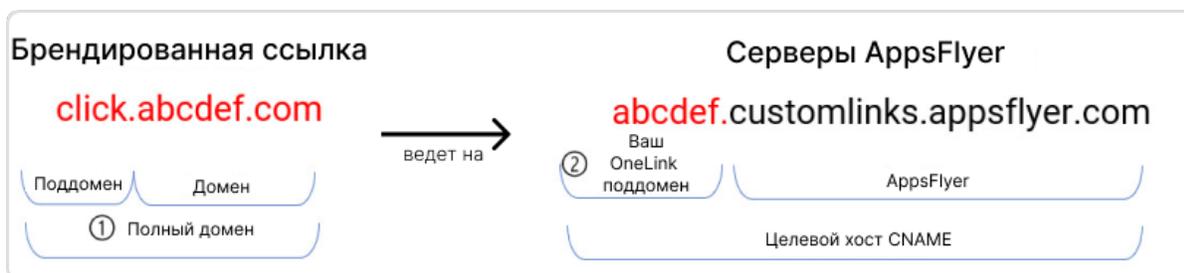
Пользователь не может перейти по ссылке и попросту теряется, однако эту проблему можно решить с помощью брендированных доменов.

Брендированный домен связывается с доменом AppsFlyer с помощью стандартных параметров DNS.

Вам также потребуется внести небольшие правки в приложение и привязать домен в панели AppsFlyer.

Чтобы подключить брендированный домен, необходимо выполнить следующие шаги:

3.20.6.1 Настройка DNS



1. Выберите полное доменное имя, например, `click.abcdef.com`, где `abcdef.com` — название вашего бренда. Поддомен (например, `click`) можно также настроить;
2. Попросите администратора DNS создать полный домен (хост);
3. Попросите администратора DNS настроить запись CNAME, чтобы полный домен (домен бренда) указывал на заданный URL-адрес (хост AppsFlyer), как это показано на схеме выше – брендированная ссылка указывает на серверы AppsFlyer.

3.20.6.2 Привязка домена к AppsFlyer

1. В панели администратора AppsFlyer перейдите в раздел **Engagement & Deep Linking** -> **Branded Domains**;
2. Нажмите *+ Add branded domain*;
3. В поле «Brand Domain» введите полный домен, как в записи DNS (например, `click.abcdef.com`);
4. Выберите поддомен OneLink из открывшегося списка;

5. Нажмите *Verify*.

Предупреждение: Если диплинкинг выполняется с использованием брендированных доменов, SDK AppsFlyer не может получить данные о конверсиях по установкам и диплинкингу.

3.20.6.3 Настройка мобильного приложения

Необходимо использовать методы `setOneLinkCustomDomain` и `oneLinkCustomDomains` для Android и iOS, соответственно.

Так производится отправка запроса по брендированной ссылке, возвращается ссылка OneLink, которой она сопоставлена, после чего направляется запрос OneLink для получения данных о конверсиях.

3.20.6.3.1 Для Android

```
public class AFApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();

        AppsFlyerConversionListener conversionListener = new AppsFlyerConversionListener() {
        }
        AppsFlyerLib.getInstance().setOneLinkCustomDomain("promotion.greatapp.com");
        AppsFlyerLib.getInstance().init(AF_DEV_KEY, conversionListener, this);
        AppsFlyerLib.getInstance().start(this, AF_DEV_KEY);

    }
}
```

Если у вас несколько брендированных доменов, передайте их все в API, чтобы всегда получать данные о конверсиях и иметь возможность использовать диплинки:

```
AppsFlyerLib.getInstance().setOneLinkCustomDomain(
    "promotion.greatapp.com",
    "click.greatapp.com",
    "deals.greatapp.com");
```

3.20.6.3.2 Для iOS

Настройте брендированную ссылку, чтобы она получала данные о конверсии, используя приведенный ниже метод:

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    AppsFlyerLib.shared().appsFlyerDevKey = "6CQi4Be6Zs9oNLsCusPbUL"
    AppsFlyerLib.shared().appleAppID = "340954504"
    AppsFlyerLib.shared().oneLinkCustomDomains = ["promotion.greatapp.com"]
    //...
    //...
}
```

Подсказка: Если у вас несколько брендированных доменов, их можно передать в API как массив строк. Так, независимо от того, какая именно брендированная ссылка используется, вы всегда будете получать данные о конверсиях и сможете использовать диплинки:

```
AppsFlyerLib.shared().oneLinkCustomDomains = ["promotion.greatapp.com", "click.greatapp.
→com"]
```

Теперь, ссылку с брендированным доменом необходимо «защитить» в приложение.

3.20.6.3.3 Для Android

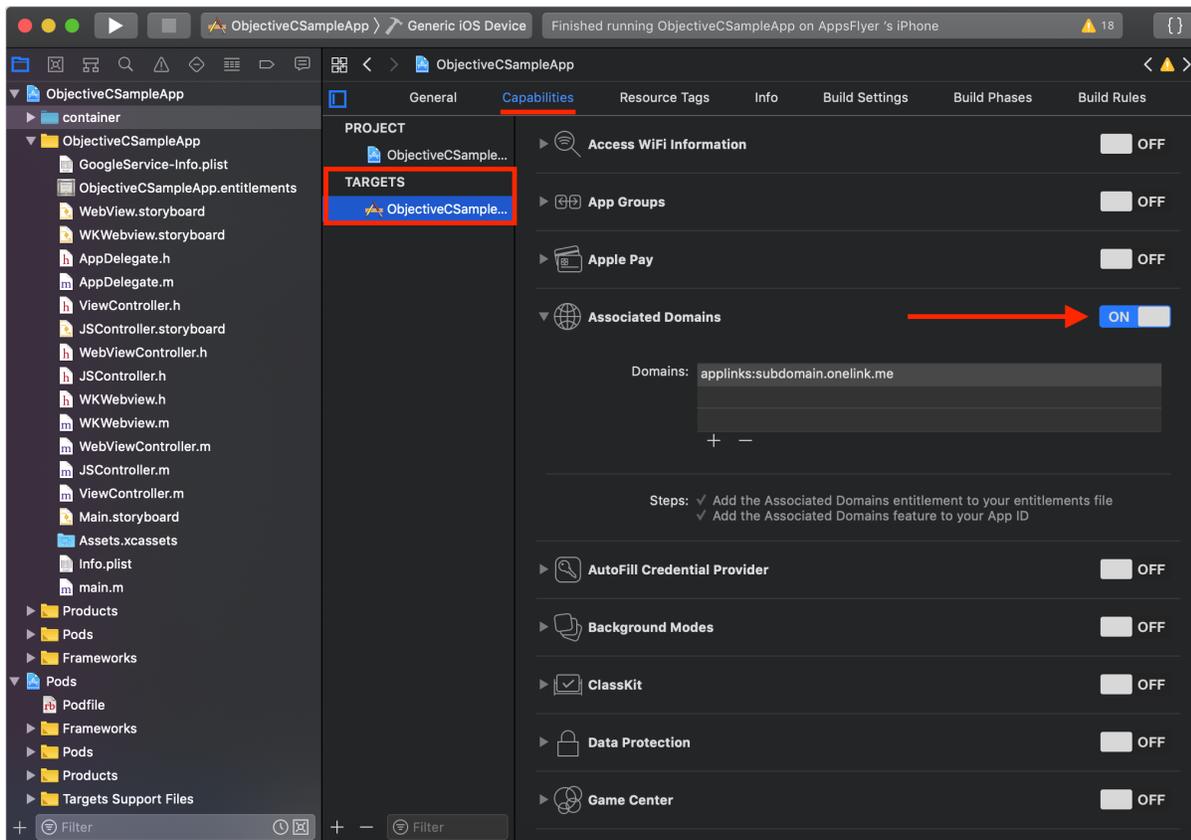
1. Откройте файл `AndroidManifest.xml`;
2. Найдите блок `intent-filter`, содержащий в себе `<action android:name=android.intent.action.VIEW />` (если такого нет, вы можете скопировать его из примера ниже и отредактировать под свои ссылки);
3. Добавьте в него объект `<data />`, описывающий ваш брендированный домен. Например, он может быть таким: `<data android:scheme=https android:host=click.yourdomain.me />`;

Пример файла `AndroidManifest.xml` после правок:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.advcake.getluckyapp">
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="com.google.android.gms.permission.AD_ID" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.GetLuckyApp">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.GetLuckyApp.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter android:autoVerify="true">
                <action android:name="android.intent.action.VIEW" />
                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.BROWSABLE" />
                <data android:scheme="https"
                    android:host="subdomain.onelink.me"
                    android:pathPrefix="/ZBZ8" />
                <!-- Начало добавленного кода -->
                <data android:scheme="https"
                    android:host="click.your-domain.me" />
                <!-- Конец добавленного кода -->
            </intent-filter>
        </activity>
    </application>
</manifest>
```

3.20.6.3.4 Для iOS

1. Перейдите в Xcode;
2. Нажмите на свой проект;
3. Нажмите на таргет проекта (показано на скриншоте снизу);
4. Перейдите на вкладку **Capabilities**;
5. Активируйте переключатель *Associated domains*;
6. Добавьте в список доменов новую ссылку с брендированным доменом. К ней нужно добавить префикс `applinks`, после чего ссылка будет выглядеть так – `applinks:click.your-domain.me`;



3.21 Инструкция AppMetrica (постбеки)

Эта инструкция поможет вам настроить интеграцию мобильных приложений через AppMetrica — платформы для аналитики и маркетинга приложений.

Данный тип интеграции настраивается для того, чтобы заказы, созданные через мобильное приложение, корректно учитывались в канале CPA:

Если у пользователя установлено мобильное приложение — мы будем отправлять его в приложение, после чего наша система будет атрибутировать все его покупки.

Если у пользователя не установлено мобильное приложение — он будет как прежде совершать заказы на сайте по CPA-модели.

Важно понимать, что рекламодатель платит не за установки, а за конкретные покупки. Мы приводим и атрибуцируем каждую покупку за нужным партнером и вебмастером.

Примечание: AppMetrica также позволяет отслеживать установки не из магазина приложений – т.е. напрямую с сайта рекламодателя по клику на поп-ап сообщение «Скачать приложение/В приложении удобнее/Перейти в приложение» и т.д.

В таком случае это будет единственным вариантом привлечения пользователя в приложение, однако оплачиваться в канале CPA будут только совершенные покупки, а не установки приложения.

Содержание:

1. *Добавление приложения;*
2. *Установка и интеграция SDK AppMetrica;*
3. *Настройка DeepLink;*
4. *Настройка отправляемых из приложения событий;*
5. *Настройки панели AppMetrica.*
6. *Проведение тестов.*

3.21.1 Шаг 1. Добавление приложения

1. Перейдите на ;
2. Укажите следующие данные: название, категория, тип приложения (приложение/игра), ссылка на приложение в маркетплейсе и часовой пояс;
3. Нажмите *Добавить приложение*

The screenshot shows a three-step wizard for adding an application. Step 1, 'Добавьте приложение', is active. It contains the following fields and options:

- Название приложения:** A text input field with the placeholder text 'Например, Едадил'.
- Тип приложения:** Two radio buttons: 'Приложение' (selected) and 'У меня игра!'.
- Категория приложения:** A dropdown menu currently showing 'Не выбрано'.
- Ссылка на приложение:** A dropdown menu with a minus sign, followed by a text input field containing 'https://apps.apple.com/ru/app/id123456789', and a '+ добавить ещё' button.
- Buttons:** A yellow 'Продолжить' button is located at the bottom of the form.

3.21.2 Шаг 2. Установка и интеграция SDK AppMetrica

Чтобы работать через AppMetrica необходимо интегрировать SDK AppMetrica в приложение.

3.21.2.1 Для Android

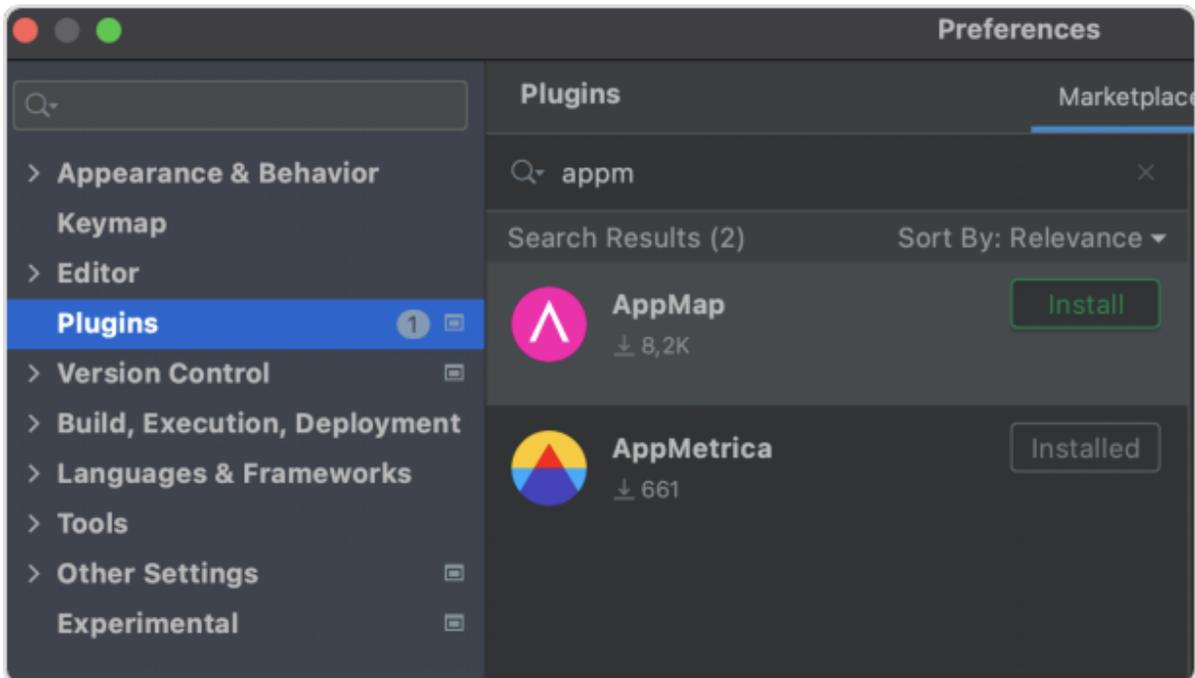
Установить SDK AppMetrica для Android можно двумя способами:

3.21.2.1.1 Через плагин (простой способ)

При установке через плагин, SDK AppMetrica установится и инициализируется сама, после этого нужно будет только настроить отправляемые из приложения события.

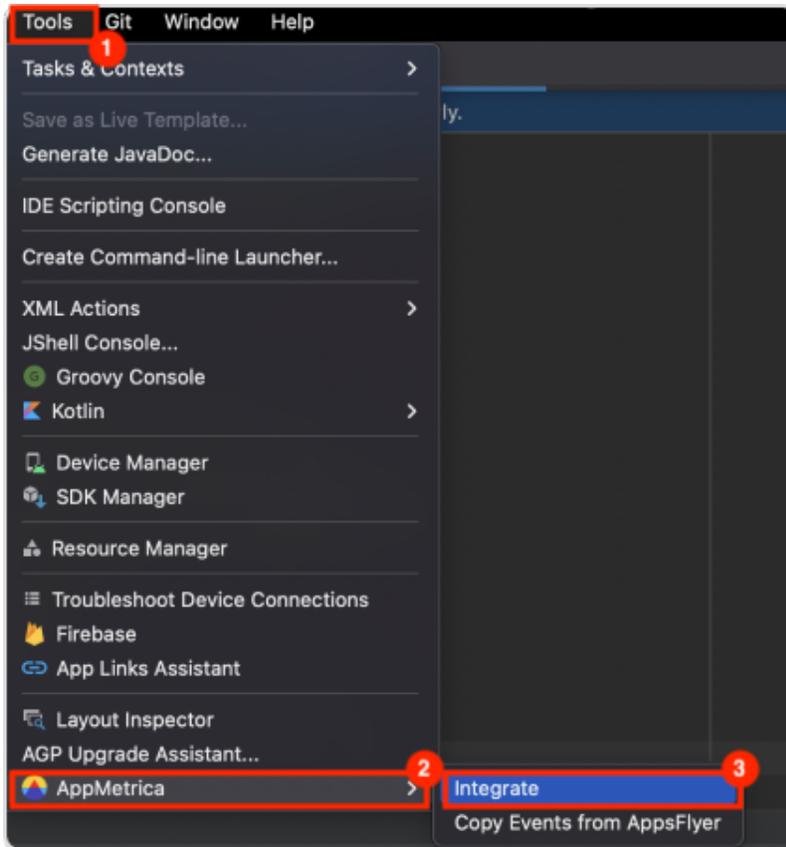
Чтобы добавить плагин AppMetrica:

1. Откройте менеджер плагинов в Android Studio («Preferences» → «Plugins»);
2. Найдите плагин AppMetrica, для этого в строке поиска введите «AppMetrica»;
3. Нажмите *Install*



Плагин добавлен, теперь перейдем к установке SDK:

1. В главном меню выберите «Tools» → «AppMetrica» → «Integrate»;



2. Выберите приложение, в которое хотите добавить SDK:

- Скопируйте API-key из веб-интерфейса AppMetrics и вставьте в приложение (API-key можно найти здесь –);
- Выберите свое приложение из списка, после чего перейдите в настройки (ключ будет записан как API-key (для использования в SDK));
- Вставьте API-key в окно, открытое плагином.

3. Укажите настройки:

- **First activation as update**

Если в вашем приложении уже есть пользователи, тогда AppMetrics по умолчанию засчитает их за новых пользователей. Чтобы не учитывать текущую аудиторию новой, необходимо включить флаг `firstActivationAsUpdate` и указать условия, по которым пользователь будет учитываться как старый.

- **Location tracking**

С этой настройкой AppMetrics будет автоматически определять местоположение устройства и передавать эту информацию в отчеты и Logs API.

- **Logs**

Включает логирование работы библиотеки.

- **Statistic sending**

Включает/отключает отправку статистики на сервер AppMetrica. Например, если для отправки статистических данных требуется согласие пользователя.

Для последующего включения отправки статистики используйте метод `YandexMetrica.setStatisticsSending(Context context, boolean enabled)`.

2. Нажмите *ОК*, после чего библиотека будет добавлена и инициализирована.

3.21.2.1.2 Установка без плагина

Если вы используете Gradle, необходимо добавить следующую зависимость в `build.gradle`:

```
dependencies {  
    implementation 'com.yandex.android:mobmetricalib:5.0.0'  
}
```

Если вы не используете Gradle, необходимо скачать и добавить библиотеки ниже самостоятельно:

- 1.
- 2.

3.21.2.2 Инициализация библиотеки

Необходимо добавить следующий код в класс, производный от базового `Application`:

```
YandexMetricaConfig config = YandexMetricaConfig.newConfigBuilder(API_key).build();  
YandexMetrica.activate(getApplicationContext(), config);  
YandexMetrica.enableActivityAutoTracking(this);
```

В `API_key` необходимо вставить ключ, который мы ранее нашли в настройках приложения в веб-интерфейсе AppMetrica.

Пример:

```
class MyApplication : Application() {  
    override fun onCreate() {  
        super.onCreate()  
        activateAppMetrica()  
    }  
  
    private fun activateAppMetrica() {  
        val appMetricaConfig: YandexMetricaConfig =  
            YandexMetricaConfig.newConfigBuilder("your-api-key")  
                .handleFirstActivationAsUpdate(isFirstActivationAsUpdate())  
                .withLocationTracking(true)  
                .withStatisticsSending(true)  
                .build()  
        YandexMetrica.activate(applicationContext, appMetricaConfig)  
    }  
}
```

3.21.2.3 Для iOS

Установить SDK Appsflyer для iOS можно тремя способами:

3.21.2.3.1 Через CocoaPods

1. Установите ;
2. Подключите статическую или динамическую версию фреймворка. В зависимости от выбора, нужно добавить следующую строку в Podfile проекта:

- **Статическая**

```
pod 'YandexMobileMetrica', '4.2.0'
```

- **Динамическая**

```
pod 'YandexMobileMetrica/Dynamic', '4.2.0'
```

3.21.2.3.2 Через Carthage

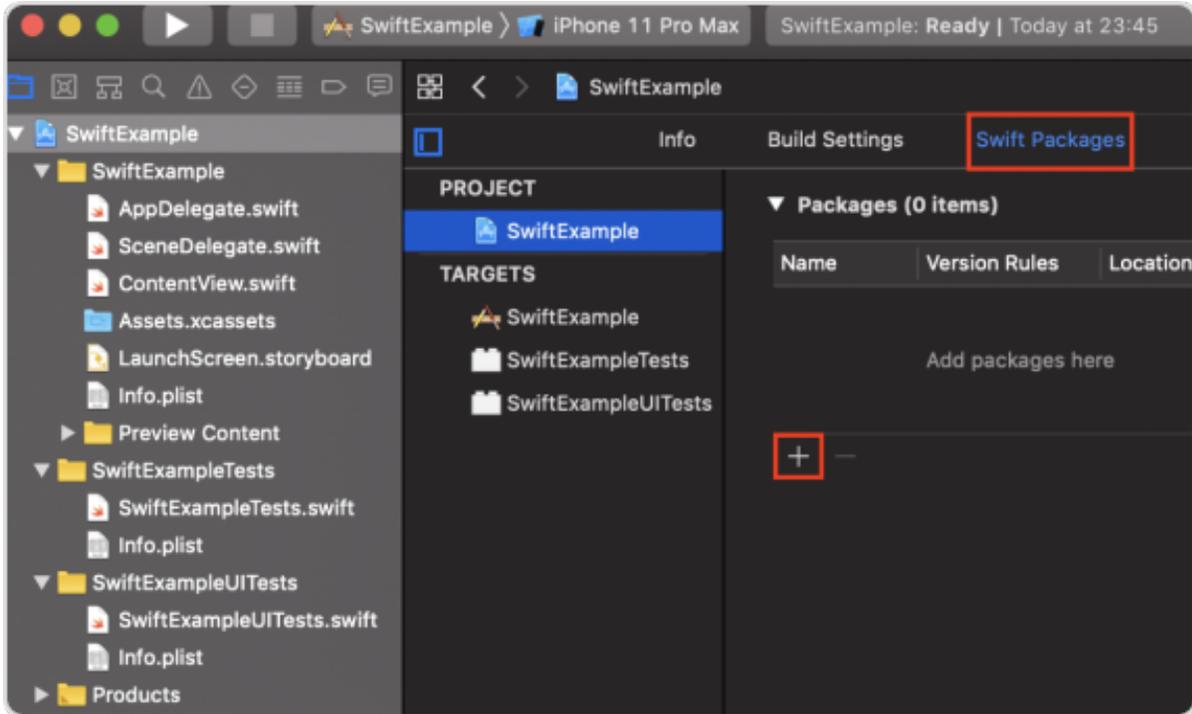
1. Добавьте зависимость в Cartfile проекта:

```
binary "https://raw.githubusercontent.com/yandexmobile/metrica-sdk-ios/master/  
↳YandexMobileMetrica.json" ~> 4.2.0
```

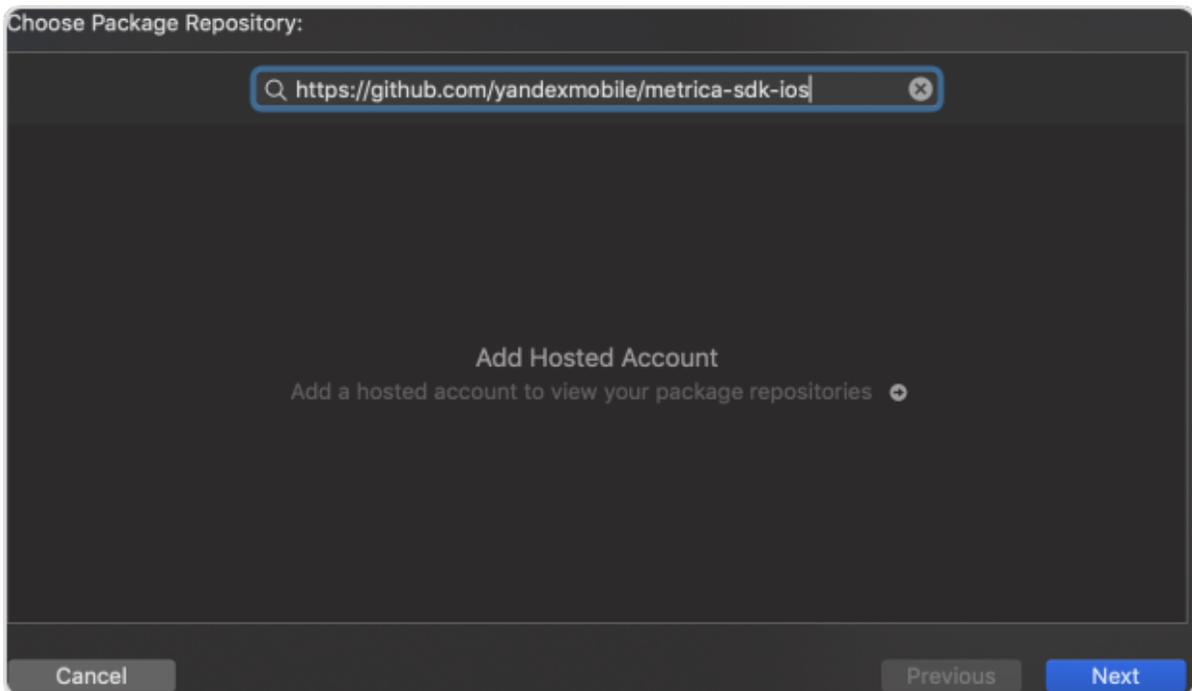
3.21.2.3.3 Через SwiftPackageManager

Чтобы подключить библиотеку через SwiftPackageManager, выполните следующие действия:

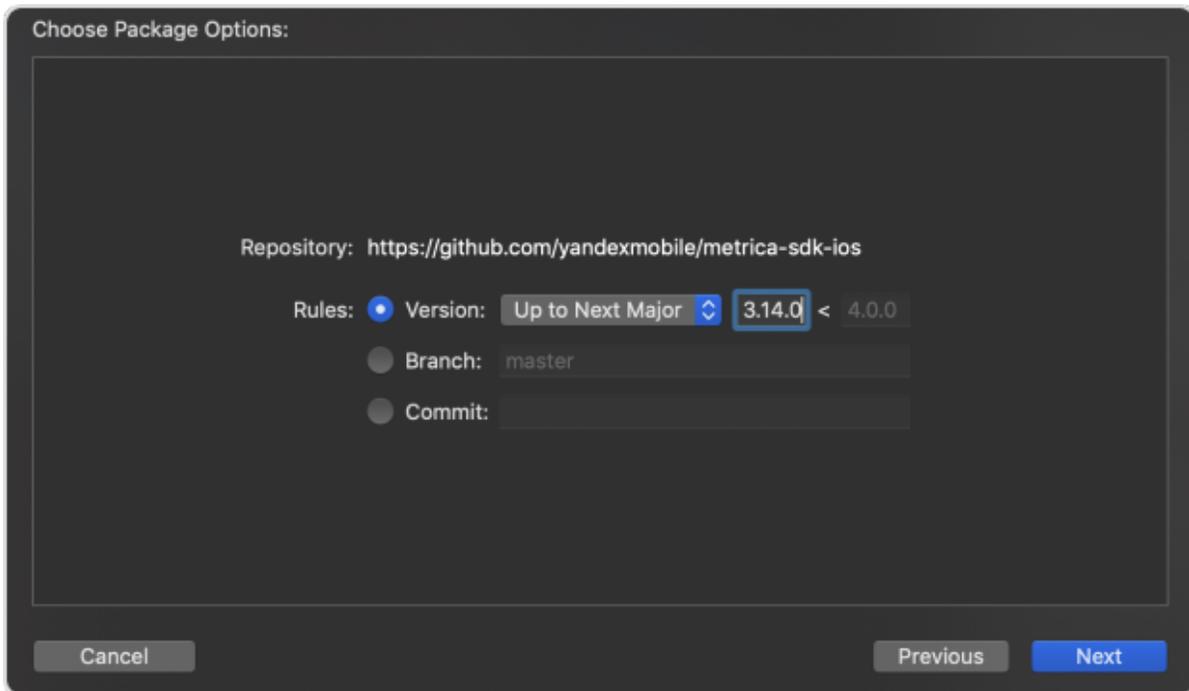
1. В Xcode, в папке вашего проекта, выберите вкладку «Swift Packages» и нажмите +;



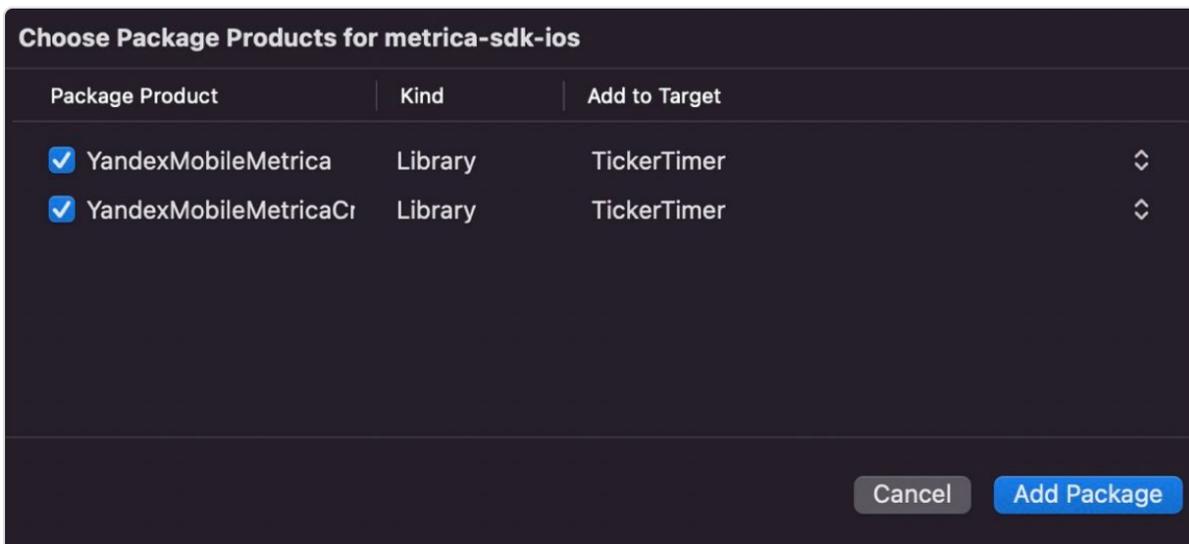
2. Укажите URL репозитория (<https://github.com/yandexmobile/metrica-sdk-ios>), в котором находится Swift-пакет:



3. Настройте правило для выбора версии пакета:



4. Выберите необходимые библиотеки и нажмите *Add Package*:



Если вы не используете указанные системы управления зависимостями ↓

Для подключения библиотеки выполните следующее:

1. Загрузите библиотеку AppMetrica: <https://storage.mds.yandex.net/get-appmetrica-mobile-sdk/50347/YandexMobileMetrica-4.2.0-ios-e34502a2-1cdd-4226-b575-86d35844b33c.zip>;

2. Добавьте `YandexMobileMetrics.framework` в проект;
3. Для подключения обработки крашей добавьте `YandexMobileMetricsCrashes.framework` (опционально);
4. Добавьте следующие зависимости: «SystemConfiguration», «UIKit», «Foundation», «CoreTelephony», «CoreLocation», «CoreGraphics», «AdSupport», «z», «sqlite3», «Security», «c++», «WebKit», «SafariServices» (с настройкой **Optional**);
5. Добавьте `-ObjC` в `Other Linker Flags`.

3.21.2.4 Инициализация SDK

1. Добавьте импорт:

```
import YandexMobileMetrics
```

2. Инициализируйте библиотеку в методе `application(_:didFinishLaunchingWithOptions:)` вашего `UIApplicationDelegate`:

```
let configuration = YMMYandexMetricsConfiguration.init(apiKey: "API_key")
YMMYandexMetrics.activate(with: configuration!)
```

Подсказка: Вместо `API_key` необходимо вставить ваш ключ из веб-интерфейса AppMetrica.

Пример:

```
import YandexMobileMetrics

@main
class AppDelegate: UIResponder, UIApplicationDelegate {
    var window: UIWindow?

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
        [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

        let configuration = YMMYandexMetricsConfiguration.init(apiKey: "api-key")
        YMMYandexMetrics.activate(with: configuration!)
        window = UIWindow(frame: UIScreen.main.bounds)
        window?.rootViewController = ViewController()
        window?.makeKeyAndVisible()

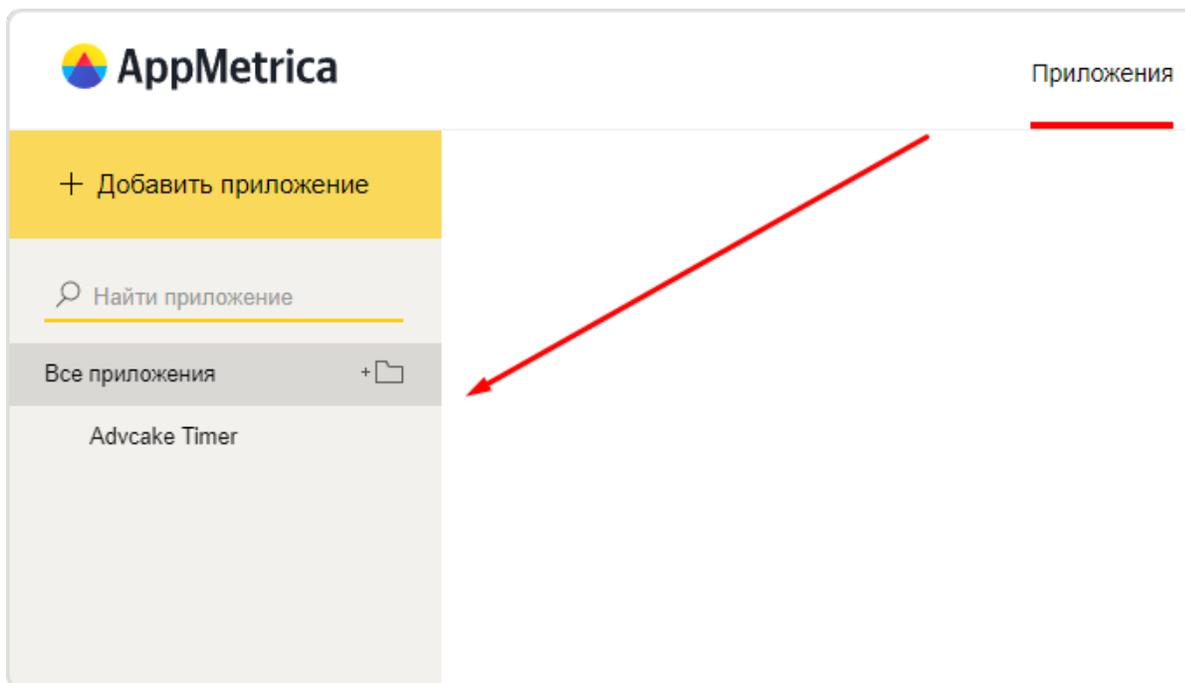
        return true
    }
}
```

3.21.3 Шаг 3. Настройка DeepLink

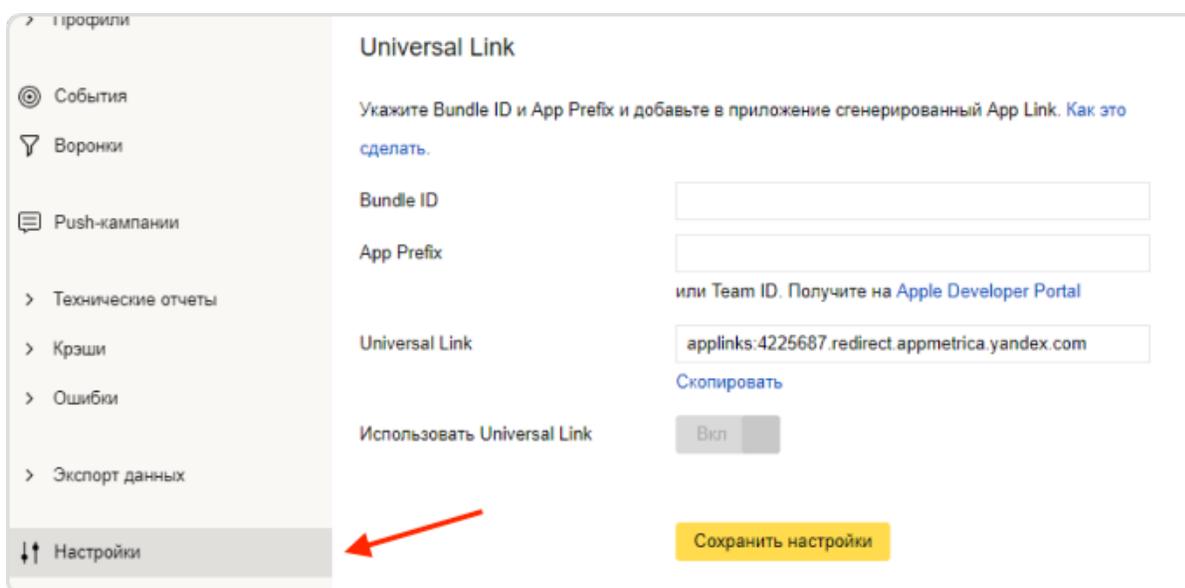
3.21.3.1 Для iOS

Необходимо включить «Universal Link» в AppMetrica, для этого:

1. В панели AppMetrica перейдите в раздел «Приложения» и выберите ваше приложение;



2. На странице выбранного приложения перейдите в раздел «Настройки»;



3. На вкладке «Основное» спуститесь вниз страницы до блока «Universal Link» и введите «Bundle ID» и «App Prefix» приложения в соответствующие поля;
4. Установите переключатель «Использовать Universal Link» в положение «Вкл» и нажмите *Сохранить настройки*, после чего, в поле «Universal Link» появится ссылка вида `applinks:<app_id>.redirect.appmetrica.yandex.com`, где `<app_id>` — идентификатор вашего приложения в AppMetrica (ID приложения).

Universal Link

Укажите Bundle ID и App Prefix и добавьте в приложение сгенерированный App Link. [Как это сделать.](#)

Bundle ID

App Prefix

или Team ID. Получите на [Apple Developer Portal](#)

Universal Link

[Скопировать](#)

Использовать Universal Link

Подсказка: Как получить Bundle ID:

- Значение Bundle ID можно найти в , в разделе «Organization Profile» → «Account Summary», а также в Xcode, в разделе «Target» → «Gen».

Как получить App Prefix:

- Значение App Prefix можно найти в . В большинстве случаев App Prefix . Этот идентификатор доступен также в консоли разработчика Apple (в разделе «Member Center» нажмите на свое имя в правом верхнем углу окна и выберите «View Account» → «Developer Account Summary»).

Для корректной работы «Universal Link» необходимо добавить в приложение следующие функции:

```
func application(_ application: UIApplication, handleOpenURL url: URL) -> Bool {
    return YMMYandexMetrica.handleOpen(url)
}

func application(_ application: UIApplication, openURL url: URL, sourceApplication: String?, annotation: AnyObject) -> Bool {
    return YMMYandexMetrica.handleOpen(url)
}
```

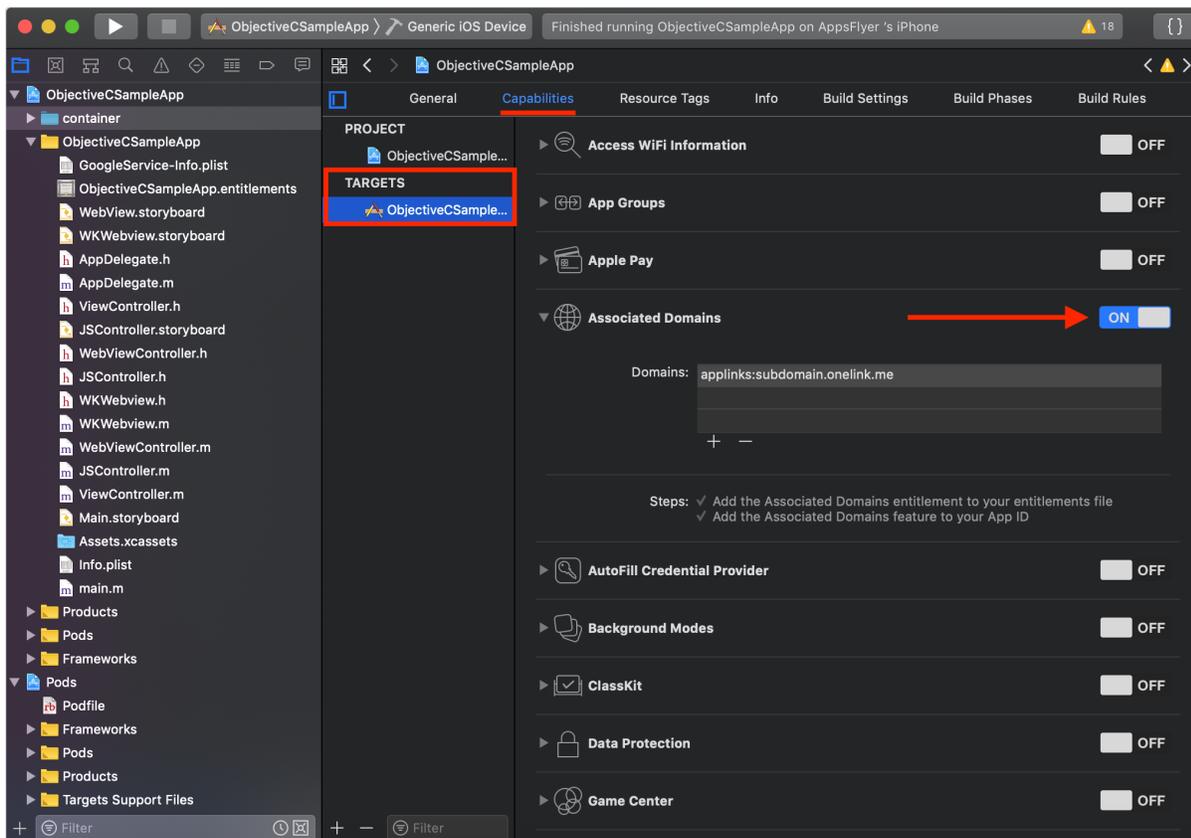
(continues on next page)

(продолжение с предыдущей страницы)

```
// Delegate for tracking Universal links.
func application(_ application: UIApplication, continueUserActivity userActivity:
↳ NSUserActivity, restorationHandler: ([AnyObject]?) -> Void) -> Bool {
    if userActivity.activityType == NSUserActivityTypeBrowsingWeb {
        if let url = userActivity.webpageURL {
            YMMYandexMetrica.handleOpen(url)
        }
    }
    return true
}
```

Далее необходимо добавить поддержку открытия ссылок через ваше приложение:

1. Нажмите на свой проект;
2. Нажмите на таргет проекта (показано на скриншоте снизу);
3. Перейдите на вкладку **Capabilities**;
4. Активируйте переключатель *Associated domains*;
5. Добавьте в список доменов новую ссылку с брендированным доменом. К ней нужно добавить префикс `applinks`, после чего ссылка будет выглядеть так: `applinks:1234567.redirect.appmetrica.yandex.com`, где 1234567 – ID вашего приложения в «AppMetrica»;



3.21.3.2 Для Android

Чтобы отслеживать открытия приложения с помощью DeepLink, нужно модифицировать активити, которое связано с DeepLink.

```
public class DeeplinkActivity extends Activity {

    @Override
    protected void onCreate(@Nullable final Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (savedInstanceState == null) {
            YandexMetrica.reportAppOpen(this);
        }
    }
    @Override
    protected void onNewIntent(final Intent intent) {
        super.onNewIntent(intent);
        YandexMetrica.reportAppOpen(intent);
    }
}
```

Теперь необходимо добавить поддержку открытия ссылок через ваше приложение. В разделе для iOS мы настроили Universal Link и получили ссылку для нашего приложения. Необходимо добавить такую же ссылку в файл AndroidManifest.xml в следующем виде:

```
<data android:scheme="https"
      android:host="1234567.redirect.appmetrica.yandex.com"
      android:pathPrefix="/"
/>
```

Вместо 1234567 у вас будет ID вашего приложения в «AppMetrica». Например, ваш файл AndroidManifest.xml может выглядеть следующим образом:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.advcake.getluckyapp">
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
  <uses-permission android:name="com.google.android.gms.permission.AD_ID" />
  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.GetLuckyApp">
    <activity
      android:name=".MainActivity"
      android:exported="true"
      android:label="@string/app_name"
      android:theme="@style/Theme.GetLuckyApp.NoActionBar">
      <intent-filter>
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter android:autoVerify="false">
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <!-- Начало добавленного кода -->
        <data android:scheme="https"
            android:host="1234567.redirect.appmetrica.yandex.com"
            android:pathPrefix="/"
        />
        <!-- Конец добавленного кода -->
    </intent-filter>
</activity>
</application>
</manifest>

```

3.21.4 Шаг 4. Настройка отправляемых из приложения событий

При интеграции через AppMetrica, мы получаем информацию о заказе через значение отправленного события. Обязательные для нас параметры – «orderId» (ID заказа) и «price» (цена). Также, можно добавлять другие параметры, например, город или категорию.

Существует два типа подходящих событий – «purchase» (рекомендуется для покупок) и «event» (подходит, если вы отправляете постбеки не по покупкам или в других случаях).

3.21.4.1 Отправка на Android

Для начала вам необходимо создать объект Revenue с помощью метода `newBuilderWithMicros()`.

Первый аргумент метода – цена, умноженная в 1000000 раз, второй аргумент – объект Currency, который можно получить с помощью метода `getInstance()`.

События типа «purchase» отправляются с помощью `reportRevenue()`.

```

Revenue revenue = Revenue.newBuilderWithMicros(99000000, Currency.getInstance("RUB"))
    .withProductID("com.yandex.service.299")
    .withQuantity(2)
    .withPayload("{\"orderId\":\"Identifier\"}")
    .build();
YandexMetrica.getReporter(getApplicationContext(), "Testing API key")
    .reportRevenue(revenue);

```

При отправке событий типа «purchase», необходимо передать «orderId» в `withPayload`.

Любые другие события отправляются с помощью метода `YandexMetrica.reportEvent(eventName, eventParameters)`, где «eventName» – имя события, а «eventParameters» – параметры события. Параметры могут быть JSON-строкой или объектом `map`.

```

Map<String, Object> eventParameters = new HashMap<String, Object>();
eventParameters.put("orderId", "123123123");

```

(continues on next page)

(продолжение с предыдущей страницы)

```
eventParameters.put("price", 480);  
  
YandexMetrica.reportEvent("customEventName", eventParameters);
```

3.21.4.2 Отправка на iOS

Для начала необходимо получить `revenueInfo` с помощью класса `YMMMutableRevenueInfo`. Объект этого класса должен быть отправлен с помощью метода `reportRevenue()`:

```
let price = NSDecimalNumber(string: "2100.5")  
let revenueInfo = YMMMutableRevenueInfo.init(priceDecimal: price, currency: "BYN")  
  
revenueInfo.productID = "TV soundbar"  
revenueInfo.quantity = 2  
revenueInfo.payload = ["OrderID": "Identifier"]  
  
let reporter = YMMYandexMetrica.reporterForApiKey("API_key")  
reporter.reportRevenue(revenueInfo, onFailure: { (error) in  
    print("REPORT ERROR: \(error.localizedDescription)")  
})
```

Здесь необходимо указать «OrderID» в `payload`.

Остальные события отправляются с помощью метода `YMMYandexMetrica.reportEvent(eventName, parameters, onFailure)`, где «eventName» – имя события, «params» – параметры, а «onFailure» – блок, выполняемый при возникновении ошибки.

Параметры должны быть парами ключ-значение:

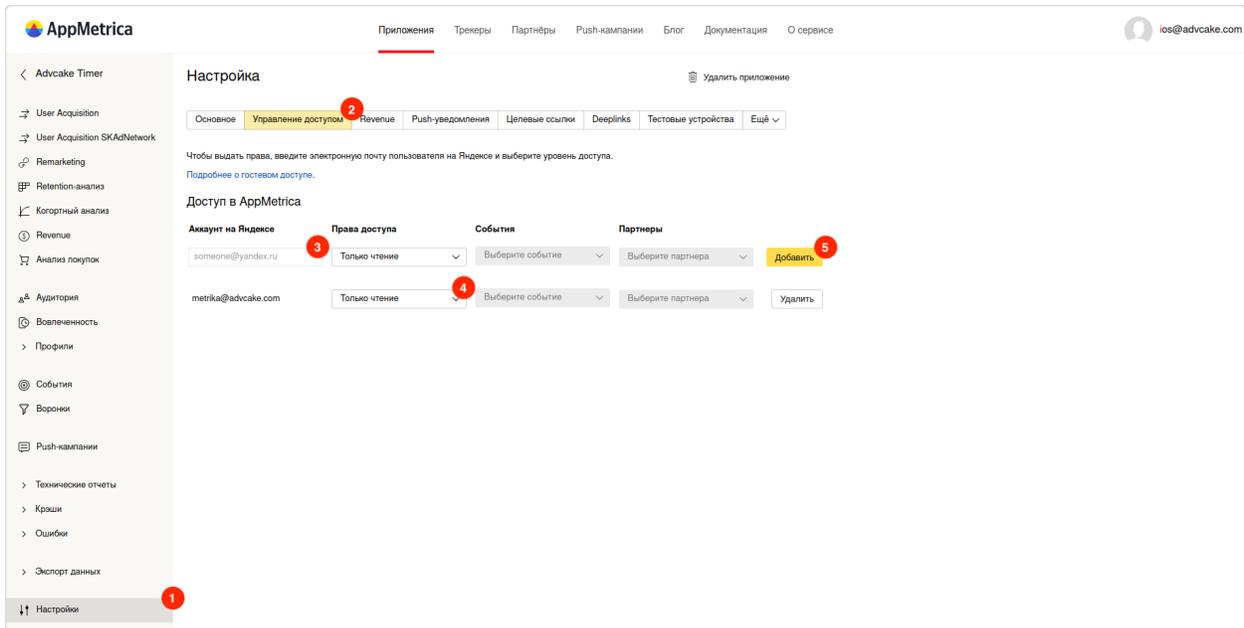
```
let params : [AnyHashable : Any] = ["orderId": "123123123", "price": 480]  
YMMYandexMetrica.reportEvent("purchase", parameters: params, onFailure: { (error)  
in  
    print("DID FAIL REPORT EVENT: %@", message)  
    print("REPORT ERROR: %@", error?.localizedDescription)  
})
```

3.21.5 Шаг 5. Настройки панели AppMetrica

Первым делом необходимо предоставить нам доступ на чтение (), а также создать трекер.

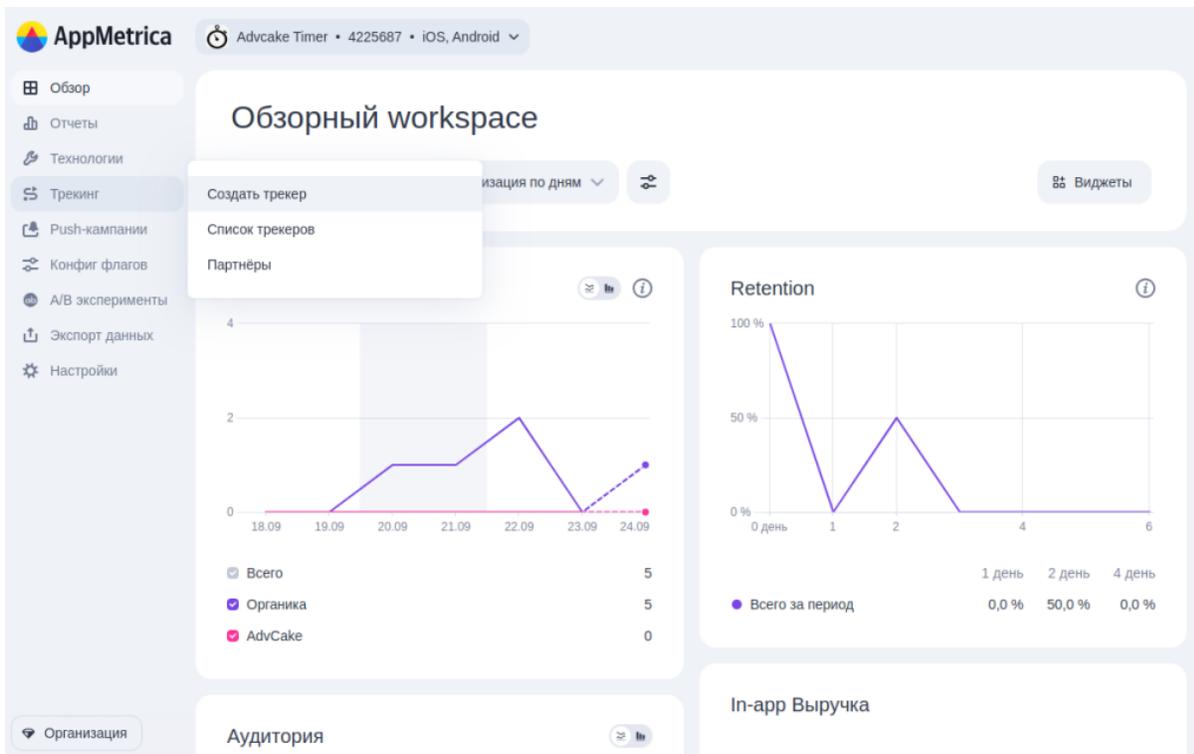
3.21.5.1 Предоставление доступа на чтение

1. Перейдите в раздел «Настройки» (1) -> вкладка «Управление доступом» (2);
2. Введите почту нашего пользователя – `metrika@advcake.com` в поле «Аккаунт на Яндексе» (3);
3. В поле «Права доступа» выберите «Только чтение» (4);
4. Нажмите *Добавить* (5).



3.21.5.2 Создание трекера

1. Перейдите на и нажмите *Создать трекер*;



3.21.5.2.1 Блок «Описание кампании»

1. Установите галочку на «Это ремаркетинговый трекер» и «Использовать SmartLink»;
2. Укажите название в формате “Appname_Advcake_Remarketing”, например “Mvideo_Advcake_Remarketing”;
3. Выберите нужное приложение из списка;
4. В поле партнер укажите «AdvCake».

Новый трекер

Описание кампании

Это ремаркетинговый трекер Вкл

Использовать SmartLink Вкл

SmartLink позволяет использовать единую трекинг-ссылку для нескольких платформ. Не используйте эту опцию, если нужно сгенерировать диплинк

Название

Приложение

Партнёр + Новый партнёр

3.21.5.2.2 Блок «SmartLink»

В этом блоке настраиваются ссылки.

1. Целевая ссылка должна вести на веб-версию вашего приложения. Если у вас нет такой целевой ссылки, то её необходимо создать;
2. Обязательно добавьте Deeplink – он будет выглядеть как URI-схема вашего приложения.

SmartLink

SmartLink позволяет использовать единую трекинг-ссылку для нескольких платформ. [Подробнее](#) ✕

Магазин приложений	<div style="display: flex; gap: 5px;"> <div style="background-color: #ffeb3b; padding: 2px 10px; border: 1px solid #ccc;">App Store (iOS) ✕</div> <div style="padding: 2px 10px; border: 1px solid #ccc;">Google Play ✕</div> <div style="padding: 2px 10px; border: 1px solid #ccc;">+ Добавить</div> </div>	
Целевая ссылка (i)	<div style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between;"> App Store ▼ </div>	+ Новая ссылка
Deerlink (i) опционально	<div style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between;"> Не выбрано ▼ </div>	+ Новая ссылка
Post-click лэндинг (i) опционально	URL	

3.21.5.2.3 Блок «Настройки атрибуции»

Установите галочку «Реатрибуция»:

Настройка атрибуции (i) ^

Типы матчинга (i)

Fingerprint (i)	<div style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between;"> 24 часа ▼ </div>	
Device ID / Google Referrer (i)	<div style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between;"> 10 дней ▼ </div>	
Реатрибуция (i)	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px; margin-right: 10px;"> <input checked="" type="checkbox"/> Вкл </div> ← </div>	

Для проверки атрибуции **добавьте тестовые устройства**. Для устройств из этого списка атрибуция будет выполняться всегда (независимо от разрешения реатрибуции).

3.21.5.2.4 Блок «Настройки постбеков»

Мы подготовили шаблоны для reengagement, установки, события и покупки:

1. Reengagement – Reeng advcake;
2. Установка – Install advcake;
3. Событие – Event advcake;
4. Покупка – Purchase advcake.

Нажмите на кнопку *+ Добавить постбек* (1), после чего выберите тип постбека (2), а затем получателя «AdvCake» (3).

Необходимо настроить установку, Reengagement и покупку. Если вы настраивали отправку событий не через покупку, а через кастомное событие – нужно выбрать событие.

После того, как вы выберете тип события и получателя, у вас появится список шаблонов – выберите соответствующие шаблоны для каждого типа постбеков (наименования указаны сверху).

Например, для типа постбека «Событие» (на скриншоте ниже) – нужно нажать на Event Advcake (4). Тогда в поле «Postback URL» автоматически подставится наш шаблон для кастомных событий.

Подсказка: Если вы хотите передавать дополнительные метки – их можно вставить, дополнив шаблон.

Предупреждение: Уберите галочки с «Отправлять постбек только для первого целевого события» и «Отправлять постбеки install-партнеру», если они установлены.

Получатель постбека: AdvCake

Событие: Reengagement | Установка | **Событие** | Ecommerce | Покупка

Postback URL: `https://api.advcake.com/postback/appmetrica?event=event`
Шаблон постбека: Event advcake | Ecommerce advcake | Purchase advcake

Отправлять постбек только для первого целевого события

Отправлять постбек по действиям всех пользователей

Метод запроса: POST

Тело запроса: `{ "keywords": "{keywords}", "partner": "{afpub_id}", "webmaster": "{site_id}", "click_id": "{click_id}", "conversion_event_name": "{conversion_event_name}" }`

Re-engagement окно: Выключено

[Удалить](#)

[+ Добавить постбек](#)

После настройки постбеков, можно сохранять трекер. Теперь он готов к работе.

3.21.6 Шаг 6. Проведение тестов

После внесения всех правок в приложение и создания трекера можно приступить к тестированию интеграции.

Это поможет вам выявить и исправить допущенные ошибки.

Предварительно необходимо подготовить мобильные устройства для тестирования. Для этого, перейдите в раздел «Настройки» -> «Управление доступом» и добавьте тестовые устройства.

1. Прокрутите страницу редактирования трекера вниз и скопируйте ссылку «Tracking URL for apps»;

Tracking URL for apps

```
https://1234567.redirect.appmetrica.yandex/?
appmetrica_tracking_id=188572452941&referrer=retribution%3D1&ios_ifa_md5=
{ios_ifa_md5}&c=(campaign)&click_timestamp=(click_timestamp)&google_aid_sha1=
{google_aid_sha1}&ios_ifa_sha1={ios_ifa_sha1}&afpub_id={afpub_id}&ios_ifa=
{ios_ifa}&google_aid_md5={google_aid_md5}&creative_id={creative_id}&device_ip=
{device_ip}&google_aid={google_aid}&click_id={click_id}&device_ua=
{device_ua}&site_id={site_id}
```

Скопировать
Отправить

2. Удалите из ссылки все параметры, кроме `appmetrica_tracking_id` и `referrer`;
3. Добавьте в ссылку параметры `afpub_id=affiliate`, `site_id=test` и `click_id` (случайное число);

Пример измененной ссылки: https://1234567.redirect.appmetrica.yandex.com/?appmetrica_tracking_id=2415974239893120&referrer=retribution%3D1&afpub_id=affiliate&site_id=test&click_id=00001

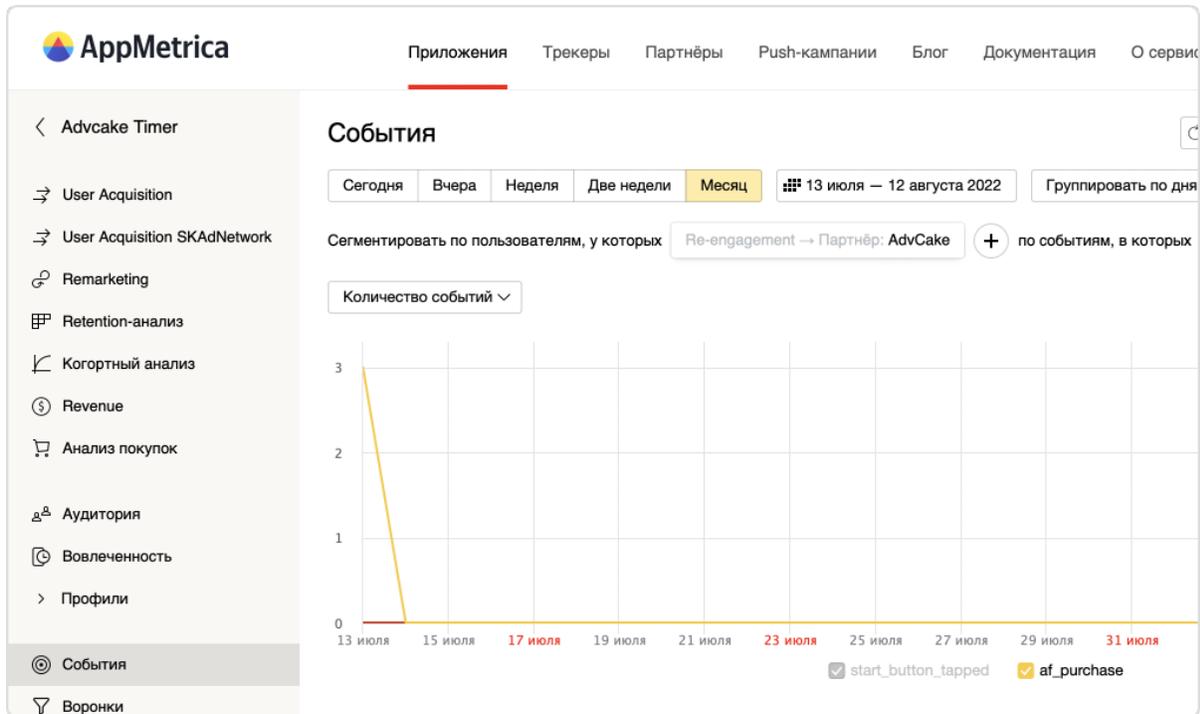
Подсказка: Параметр `click_id` должен быть уникальным значением. Это значит, что после каждого перехода по ссылке нужно заменять `click_id` на другое значение.

4. Перейдите по ссылке с ПК, после чего у вас должна открыться веб-версия вашего приложения. Если этого не произошло – проверьте настройки трекера;
5. Отправьте ссылку на мобильные устройства для тестирования. Не забудьте изменить `click_id` в ваших ссылках;
6. Удалите приложения с мобильных устройств и перейдите по ссылке. Если после перехода открылась веб-версия вашего приложения – это значит, что всё настроено правильно. В противном случае, перепроверьте настройки трекера;
7. Установите приложения на мобильные устройства и перейдите по ссылке. Ваши приложения должны перехватывать ссылки и открывать их изнутри. Если этого не происходит – проверьте, все ли правки вы внесли в приложение (`AndroidManifest.xml` / `Associated domains`);
8. Перейдите по новой ссылке и совершите тестовую покупку в приложении. Должен выполняться код, который вы настраивали ранее.

Чтобы посмотреть события перейдите в раздел «События» (обычно подгружаются в течение часа) и выберите фильтр *Reengagement* -> *партнер: AdvCake*.

Если события отображаются в общем фильтре, но не отображаются в фильтре с *reengagement* и нашим партнером – проверьте, внедрили ли вы код, позволяющий отслеживать открытия с помощью *Deerlink*.

Если события не отображаются вообще – проверьте, добавили ли вы код, который отправляет события.



3.22 Инструкция AppMetrica (инсталлы)

Эта инструкция поможет вам настроить интеграцию мобильных приложений через AppMetrica — платформы для аналитики и маркетинга приложений.

Данный тип интеграции настраивается для того, чтобы установки и соответствующие им заказы, созданные через мобильное приложение, корректно учитывались в канале CPI.

Важно понимать, что при интеграции через инсталлы, мы отслеживаем только покупки, совершенные после установки приложения и рекламодатель оплачивает только установку, подтвержденную заказом.

Содержание:

1. *Добавление приложения;*
2. *Установка и интеграция SDK AppMetrica;*
3. *Настройка отправляемых из приложения событий;*
4. *Настройки панели AppMetrica.*

3.22.1 Шаг 1. Добавление приложения

1. Перейдите на ;
2. Укажите следующие данные: название, категория, тип приложения (приложение/игра), ссылка на приложение в маркетплейсе и часовой пояс;
3. Нажмите *Добавить приложение*

1 Добавьте приложение > **2** Настройте детали > **3** Интегрируйте SDK

Название приложения

Категория приложения

Ссылка на приложение

3.22.2 Шаг 2. Установка и интеграция SDK AppMetrica

Чтобы работать через AppMetrica необходимо интегрировать SDK AppMetrica в приложение.

3.22.2.1 Для Android

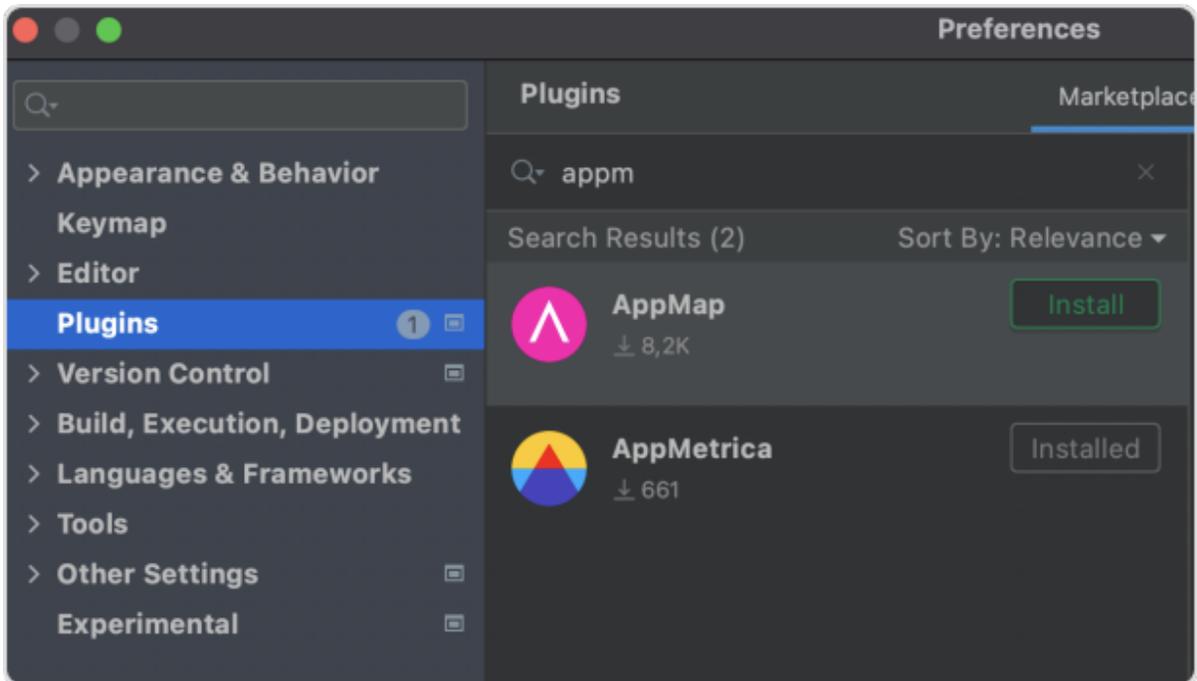
Установить SDK AppsFlyer для Android можно двумя способами:

3.22.2.1.1 Через плагин (простой способ)

При установке через плагин, SDK AppMetrica установится и инициализируется сама, после этого нужно будет только настроить отправляемые из приложения события.

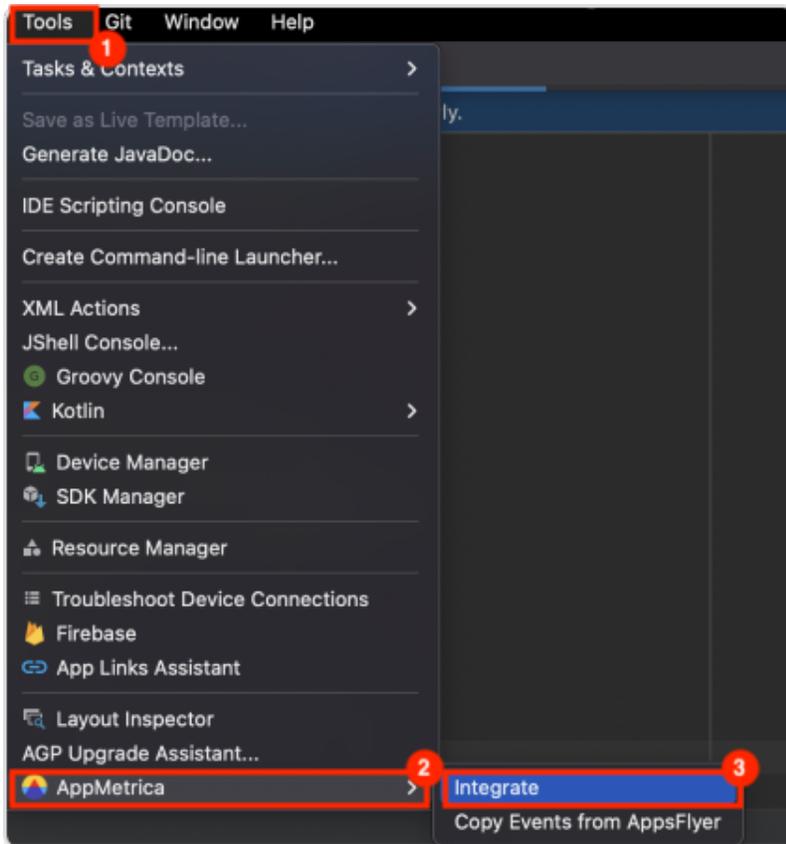
Чтобы добавить плагин AppMetrica:

1. Откройте менеджер плагинов в Android Studio («Preferences» → «Plugins»);
2. Найдите плагин AppMetrica, для этого в строке поиска введите «AppMetrica»;
3. Нажмите *Install*



Плагин добавлен, теперь перейдем к установке SDK:

1. В главном меню выберите «Tools» → «AppMetrica» → «Integrate»;



2. Выберите приложение, в которое хотите добавить SDK:

- Скопируйте API-key из веб-интерфейса AppMetrica и вставьте в приложение (API-key можно найти здесь –);
- Выберите свое приложение из списка, после чего перейдите в настройки (ключ будет записан как API-key (для использования в SDK));
- Вставьте API-key в окно, открытое плагином.

3. Укажите настройки:

- **First activation as update**

Если в вашем приложении уже есть пользователи, тогда AppMetrica по умолчанию засчитает их за новых пользователей. Чтобы не учитывать текущую аудиторию новой, необходимо включить флаг `firstActivationAsUpdate` и указать условия, по которым пользователь будет учитываться как старый.

- **Location tracking**

С этой настройкой AppMetrica будет автоматически определять местоположение устройства и передавать эту информацию в отчеты и Logs API.

- **Logs**

Включает логирование работы библиотеки.

- **Statistic sending**

Включает/отключает отправку статистики на сервер AppMetrica. Например, если для отправки статистических данных требуется согласие пользователя.

Для последующего включения отправки статистики используйте метод `YandexMetrica.setStatisticsSending(Context context, boolean enabled)`.

2. Нажмите *OK*, после чего библиотека будет добавлена и инициализирована.

3.22.2.1.2 Установка без плагина

Если вы используете Gradle, необходимо добавить следующую зависимость в `build.gradle`:

```
dependencies {  
    implementation 'com.yandex.android:mobmetricalib:5.0.0'  
}
```

Если вы не используете Gradle, необходимо скачать и добавить библиотеки ниже самостоятельно:

- 1.
- 2.

3.22.2.2 Инициализация библиотеки

Необходимо добавить следующий код в класс, производный от базового `Application`:

```
YandexMetricaConfig config = YandexMetricaConfig.newConfigBuilder(API_key).build();
YandexMetrica.activate(getApplicationContext(), config);
YandexMetrica.enableActivityAutoTracking(this);
```

В `API_key` необходимо вставить ключ, который мы ранее нашли в настройках приложения в веб-интерфейсе AppMetrica.

Пример:

```
class MyApplication : Application() {
    override fun onCreate() {
        super.onCreate()
        activateAppMetrica()
    }

    private fun activateAppMetrica() {
        val appMetricaConfig: YandexMetricaConfig =
            YandexMetricaConfig.newConfigBuilder("your-api-key")
                .handleFirstActivationAsUpdate(isFirstActivationAsUpdate())
                .withLocationTracking(true)
                .withStatisticsSending(true)
                .build()
        YandexMetrica.activate(applicationContext, appMetricaConfig)
    }
}
```

3.22.2.3 Для iOS

Установить SDK AppsFlyer для iOS можно тремя способами:

3.22.2.3.1 Через Cocoapods

1. Установите ;
2. Подключите статическую или динамическую версию фреймворка. В зависимости от выбора, нужно добавить следующую строку в Podfile проекта:

- Статическая

```
pod 'YandexMobileMetrica', '4.2.0'
```

- Динамическая

```
pod 'YandexMobileMetrica/Dynamic', '4.2.0'
```

3.22.2.3.2 Через Carthage

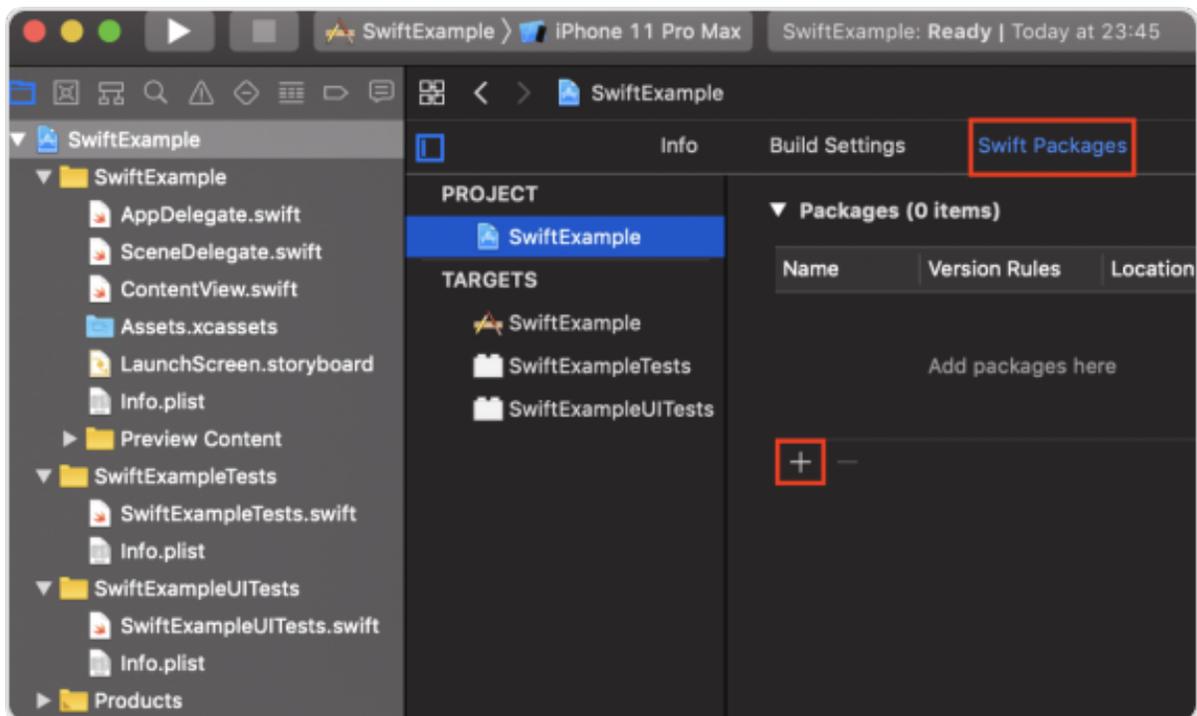
1. Добавьте зависимость в Cartfile проекта:

```
binary "https://raw.githubusercontent.com/yandexmobile/metrica-sdk-ios/master/  
↳YandexMobileMetrica.json" ~> 4.2.0
```

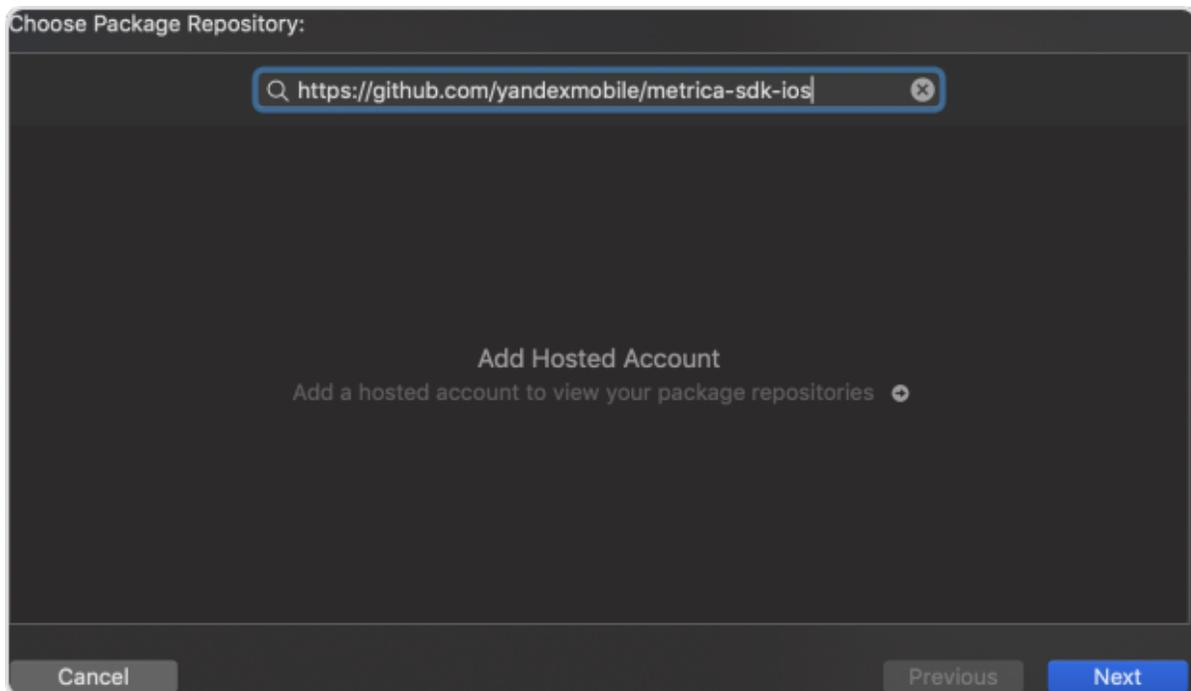
3.22.2.3.3 Через SwiftPackageManager

Чтобы подключить библиотеку через SwiftPackageManager, выполните следующие действия:

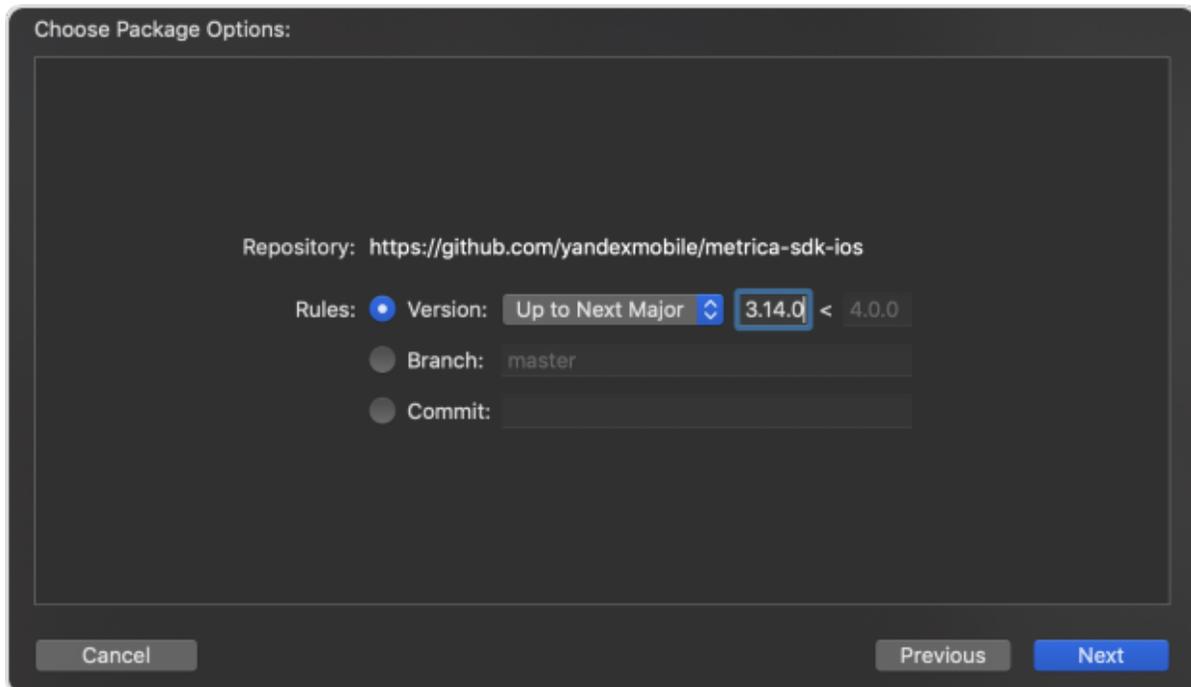
1. В Xcode, в папке вашего проекта, выберите вкладку «Swift Packages» и нажмите +;



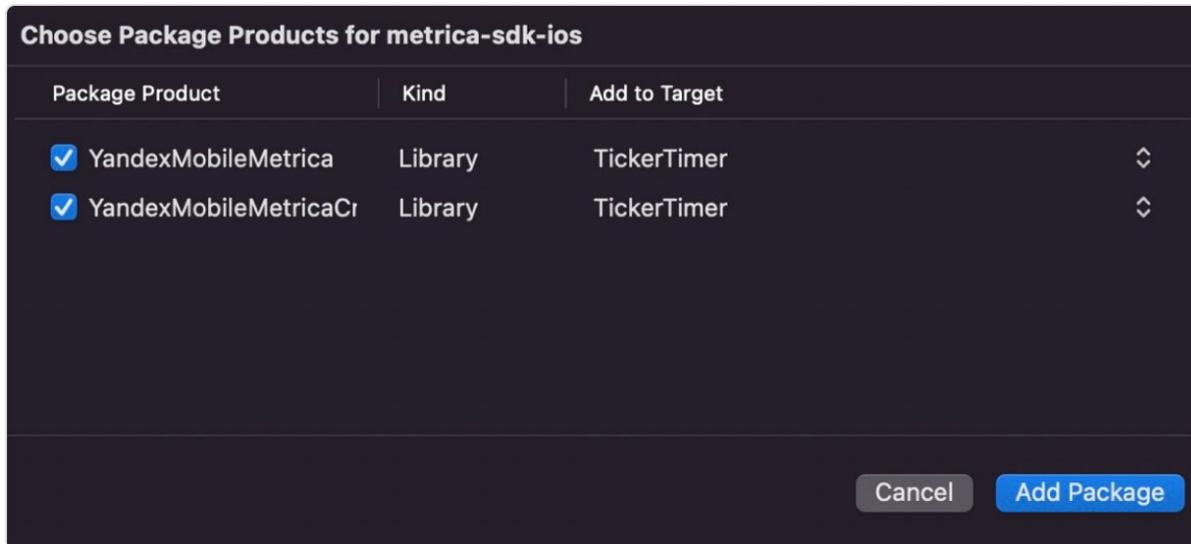
2. Укажите URL репозитория (), в котором находится Swift-пакет:



3. Настройте правило для выбора версии пакета:



4. Выберите необходимые библиотеки и нажмите *Add Package*:



Без использования систем управления зависимостями ↓

Для подключения библиотеки выполните следующее:

1. Загрузите библиотеку AppMetrica: <https://storage.mds.yandex.net/get-appmetrica-mobile-sdk/50347/YandexMobileMetrica-4.2.0-ios-e34502a2-1cdd-4226-b575-86d35844b33c.zip>;
2. Добавьте `YandexMobileMetrica.framework` в проект;
3. Для подключения обработки крашей добавьте `YandexMobileMetricaCrashes.framework` (опционально);
4. Добавьте следующие зависимости: «SystemConfiguration», «UIKit», «Foundation», «CoreTelephony», «CoreLocation», «CoreGraphics», «AdSupport», «z», «sqlite3», «Security», «c++», «WebKit», «SafariServices» (с настройкой **Optional**);
5. Добавьте `-ObjC` в `Other Linker Flags`.

3.22.2.4 Инициализация SDK

1. Добавьте импорт:

```
import YandexMobileMetrica
```

2. Инициализируйте библиотеку в методе `application(_:didFinishLaunchingWithOptions:)` вашего `UIApplicationDelegate`:

```
let configuration = YMMYandexMetricaConfiguration.init(apiKey: "API_key")
YMMYandexMetrica.activate(with: configuration!)
```

Подсказка: Вместо `API_key` необходимо вставить ваш ключ из веб-интерфейса AppMetrica.

Пример:

```
import YandexMobileMetrica

@main
class AppDelegate: UIResponder, UIApplicationDelegate {
    var window: UIWindow?

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
        [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

        let configuration = YMMYandexMetricaConfiguration.init(apiKey: "api-key")
        YMMYandexMetrica.activate(with: configuration!)
        window = UIWindow(frame: UIScreen.main.bounds)
        window?.rootViewController = ViewController()
        window?.makeKeyAndVisible()

        return true
    }
}
```

3.22.3 Шаг 3. Настройка отправляемых из приложения событий

При интеграции через AppMetrica, мы получаем информацию о заказе через значение отправленного события. Обязательные для нас параметры – «orderId» (ID заказа) и «price» (цена). Также, можно добавлять другие параметры, например, город или категорию.

Существует два типа подходящих событий – «purchase» (рекомендуется для покупок) и «event» (подходит, если вы отправляете постбеки не по покупкам или в других случаях).

3.22.3.1 Отправка на Android

Для начала вам необходимо создать объект Revenue с помощью метода `newBuilderWithMicros()`.

Первый аргумент метода – цена, умноженная в 1000000 раз, второй аргумент – объект Currency, который можно получить с помощью метода `getInstance()`.

События типа «purchase» отправляются с помощью `reportRevenue()`.

```
Revenue revenue = Revenue.newBuilderWithMicros(99000000, Currency.getInstance("RUB"))
    .withProductID("com.yandex.service.299")
    .withQuantity(2)
    .withPayload("{\"orderId\":\"Identifier\"}")
    .build();
YandexMetrica.getReporter(getApplicationContext(), "Testing API key").
    reportRevenue(revenue);
```

При отправке событий типа «purchase», необходимо передать «orderId» в `withPayload`.

Любые другие события отправляются с помощью метода `YandexMetrica.reportEvent(eventName, eventParameters)`, где «eventName» – имя события, а «eventParameters» – параметры события. Параметры могут быть JSON-строкой или объектом `map`.

```
Map<String, Object> eventParameters = new HashMap<String, Object>();
eventParameters.put("orderId", "123123123");
eventParameters.put("price", 480);
```

(continues on next page)

(продолжение с предыдущей страницы)

```
YandexMetrica.reportEvent("customEventName", eventParameters);
```

3.22.3.2 Отправка на iOS

Для начала необходимо получить `revenueInfo` с помощью класса `YMMMutableRevenueInfo`. Объект этого класса должен быть отправлен с помощью метода `reportRevenue()`:

```
let price = NSDecimalNumber(string: "2100.5")
let revenueInfo = YMMMutableRevenueInfo.init(priceDecimal: price, currency: "BYN")

revenueInfo.productID = "TV soundbar"
revenueInfo.quantity = 2
revenueInfo.payload = ["OrderID": "Identifier"]

let reporter = YMMYandexMetrica.reporterForApiKey("API_key")
reporter.reportRevenue(revenueInfo, onFailure: { (error) in
    print("REPORT ERROR: \(error.localizedDescription)")
})
```

Здесь необходимо указать «OrderID» в `payload`.

Остальные события отправляются с помощью метода `YMMYandexMetrica.reportEvent(eventName, parameters, onFailure)`, где «eventName» – имя события, «params» – параметры, а «onFailure» – блок, выполняемый при возникновении ошибки.

Параметры должны быть парами ключ-значение:

```
let params : [AnyHashable : Any] = ["orderId": "123123123", "price": 480]
YMMYandexMetrica.reportEvent("purchase", parameters: params, onFailure: { (error)
in
    print("DID FAIL REPORT EVENT: %@", message)
    print("REPORT ERROR: %@", error?.localizedDescription)
})
```

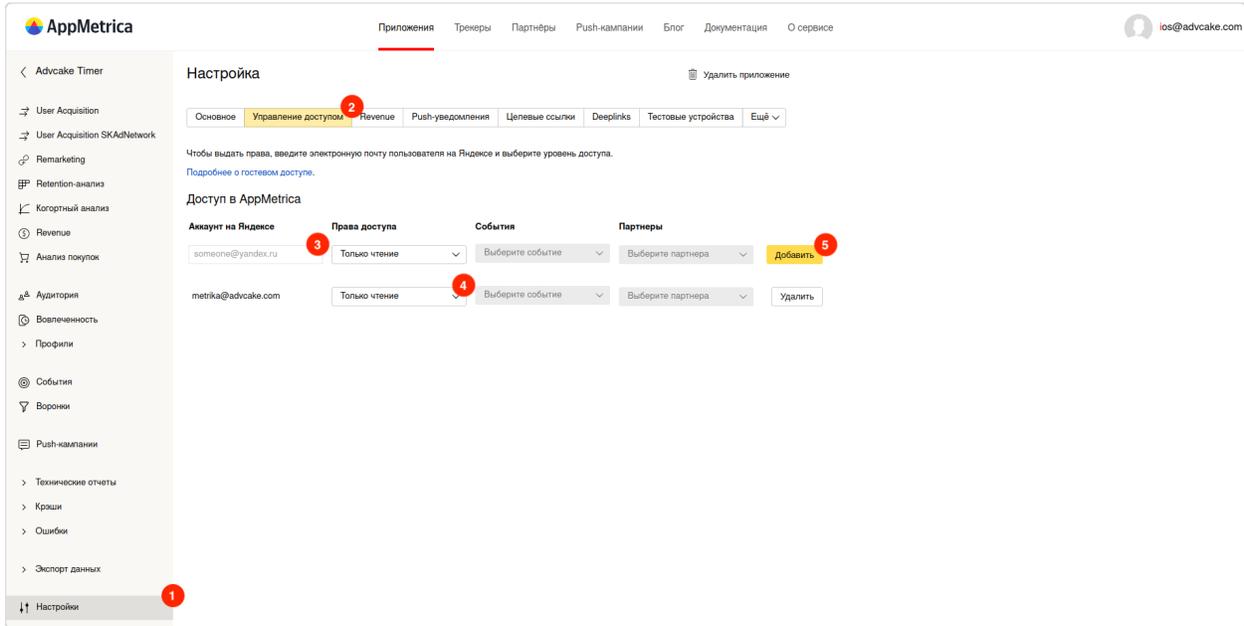
3.22.4 Шаг 4. Настройки панели AppMetrica

Первым делом необходимо предоставить нам доступ на чтение (), а также создать трекер.

Подсказка: Если вы хотите, что бы мы сами создали трекер – предоставьте нам доступ на чтение/редактирование.

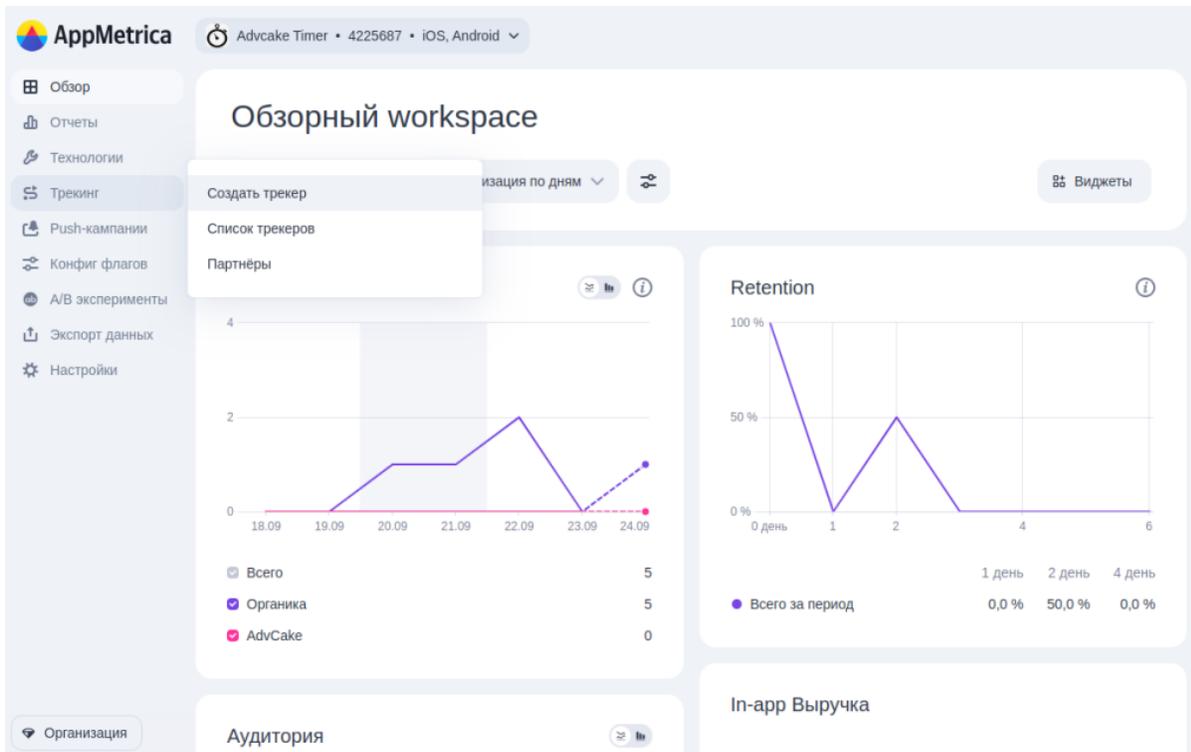
3.22.4.1 Предоставление доступа на чтение

1. Перейдите в раздел «Настройки» (1) -> вкладка «Управление доступом» (2);
2. Введите почту нашего пользователя – *metrika@advcake.com* в поле «Аккаунт на Яндексе» (3);
3. В поле «Права доступа» выберите «Только чтение» (4);
4. Нажмите *Добавить* (5).



3.22.4.2 Создание трекера

1. Перейдите на и нажмите *Создать трекер*;



3.22.4.2.1 Блок «Описание кампании»

1. Оставьте галочку «Это ремаркетинговый трекер» выключенной;
2. Укажите название в формате “Appname_Advcake_Install”, например “Mvideo_Advcake_Install”;
3. Выберите нужное приложение из списка;
4. В поле партнер укажите «AdvCake».

Новый трекер

Описание кампании

Это ремаркетинговый трекер Откл

Название

Приложение

Партнёр [+ Новый партнёр](#)

3.22.4.2.2 Блок «SmartLink»

Добавьте нужные платформы и укажите для них ссылки на магазин приложений:

SmartLink

SmartLink позволяет использовать единую трекинг-ссылку для нескольких платформ. [Подробнее](#) ✕

Магазин приложений App Store (iOS) ✕ Google Play ✕ + Добавить

Целевая ссылка ⓘ ▼ [+ Новая ссылка](#)

DeepLink ⓘ опционально ▼ [+ Новая ссылка](#)

Post-click лэндинг ⓘ опционально

3.22.4.2.3 Блок «Настройки атрибуции»

Данный блок настроен по умолчанию. Сверьте ваши настройки с настройками на скриншоте:

Настройка атрибуции ⓘ

Типы матчинга ⓘ

Fingerprint ⓘ ▼

Device ID / Google Referrer ⓘ ▼

Реатрибуция ⓘ Откл

 Для проверки атрибуции [добавьте тестовые устройства](#). Для устройств из этого списка атрибуция будет выполняться всегда (независимо от разрешения реатрибуции).

3.22.4.2.4 Блок «Настройки постбеков»

Мы подготовили шаблоны для reengagement, установки, события и покупки:

1. Reengagement – Reeng advcake;
2. Установка – Install advcake;
3. Событие – Event advcake;
4. Покупка – Purchase advcake.

Нажмите на кнопку *+ Добавить постбек* (1), после чего выберите тип постбека (2), а затем получателя «AdvCake» (3).

Необходимо настроить установку и покупку. Если вы настраивали отправление событий не через покупку, а через кастомное событие – нужно выбрать событие.

После того, как вы выберете тип события и получателя, у вас появится список шаблонов – выберите соответствующие шаблоны для каждого типа постбеков (наименования указаны сверху).

Например, для типа постбека «Событие» (на скриншоте ниже) – нужно нажать на Event Advcake (4). Тогда в поле «Postback URL» автоматически подставится наш шаблон для кастомных событий.

Подсказка: Если вы хотите передавать дополнительные метки – их можно вставить, дополнив шаблон.

Предупреждение: Уберите галочки с «Отправлять постбек только для первого целевого события» и «Отправлять постбеки install-партнеру», если они установлены.

Получатель постбека: AdvCake

Событие: Reengagement, Установка, **Событие**, Ecommerce, Покупка

Postback URL: `https://api.advcake.com/postback/appmetrica?event=event`
 Шаблон постбека: Event, Ecommerce advcake, Purchase advcake
 Event advcake

Метод запроса: POST

Тело запроса: `{ "keywords": "{keywords}", "partner": "{afpub_id}", "webmaster": "{site_id}", "click_id": "{click_id}", "conversion_event_name": "{conversion_event_name}" }`

Re-engagement окно: Выключено

Удалить

+ Добавить постбек

После настройки постбеков, можно сохранять трекер. Теперь он готов к работе.

3.23 Пример инструкции по интеграции антифрода

3.23.1 Что происходит во время интеграции?

Наш код позволяет отслеживать историю действий пользователя на сайте, в том числе для выявления нарушений правил оффера вебмастерами (например, cookie-stuffing).

Во время процесса интеграции код устанавливает обезличенные файлы cookie, которые позволяют отслеживать путь пользователя, а также изменения в источнике трафика.

Мы собираем и используем следующие данные:

1. Параметры URL перехода:
 - Cookie - обезличенный идентификатор пользователя;
 - Cookie - обезличенный идентификатор сессии;
 - Cookies - параметров последнего перехода (полный URL и UTM-метки отдельно).
2. Параметры браузера:
 - User-Agent;
 - Размер экрана;
 - Размер окна браузера;

- Открытие страницы в iframe;
- URL текущей страницы и ее реферер;

Нам также необходимо сформировать DATA-объект, содержащий информацию о покупке.

Мы не собираем личную информацию, а используем только те файлы cookie, которые настраиваем сами.

Подсказка: В CPA-канале используется модель атрибуции «Last Paid Click» – согласно этой модели, заказ присваивается платному источнику, который был последним в истории переходов перед совершением заказа.

Примечание: Это пример инструкции по интеграции Adv.Cake. Инструкцию с актуальными данными для интеграции именно вашего оффера вам передаст ваш менеджер.

Содержание:

1. *Отправка брифа;*
2. *Установка JS-кода на страницы сайта;*
3. *Вызов кода и передача номера заказа при срабатывании события создания заказа.*

3.23.2 Шаг 1. Отправка брифа

1. Заполните и отправьте бриф менеджеру Adv.Cake;
2. Получите кастомную инструкцию для интеграции с вашим персональным доменом JS-кода.

3.23.3 Шаг 2. Установка JS-кода на страницы сайта

Установить JS-код можно двумя способами:

1. Напрямую на сайт (**рекомендуется**):
 - Код из примера ниже можно поставить как в <head>, так и в <body>, однако мы рекомендуем ставить его выше, чтобы другие скрипты, в случае их поломки, не повлияли на работу нашего кода.
2. Через GTM (**не рекомендуется**, т.к. код, установленный через GTM, блокируется блокировщиками рекламы, что приводит к потере конверсий от 30% до 50%):
 - Если вы все же решили устанавливать наш код через GTM, необходимо выбрать триггер активации «Page View» или «DOM Ready». Триггер «Window Load» не подходит, так как другие скрипты, которые будут загружаться у пользователя могут заблокировать загрузку нашего скрипта.

```
<script type="text/javascript">
  (function(a) {
    var b = a.createElement("script");
    b.async = 1;
    b.src = "///**.*.com/"; // точный домен для интеграции будет вам передан
```

(continues on next page)

(продолжение с предыдущей страницы)

```
← вместе с кастомной инструкцией
    a = a.getElementsByTagName("script")[0];
    a.parentNode.insertBefore(b, a)
  })(document);
</script>
```

3.23.4 Шаг 3. Вызов кода и передача номера заказа при срабатывании события создания заказа (всплытие рор-уп окна через аjax или переход на страницу «Спасибо за заказ»)

```
<script type="text/javascript">
  (function() {
    window.advcake_order = window.advcake_order || function(ord_id) {
      window.advcake_order_id = ord_id;
    };
    window.advcake_order("123376"); // 123376 - order_id
  })();
</script>
```

API Adv.Cake – это интерфейс, который позволяет интегрировать возможности Adv.Cake в ваш личный проект.

На этой странице представлена основная информация и примеры по работе с API.

4.1 Описание

REST API Adv.Cake работает по протоколу HTTPS и представляет собой набор методов, с помощью которых совершаются запросы и возвращаются ответы для каждой операции.

Все ответы приходят в виде JSON или XML-структур.

4.2 Основной URL

Все ссылки на запросы к API включают обязательный основной URL: <https://api.advcake.com/> или <https://cakelink.ru/>.

4.3 Авторизация

Каждый запрос содержит персональный API-ключ, с помощью которого Вы можете выполнять запросы к Adv.Cake API.

Подсказка: При переходе по ссылке из личного кабинета все необходимые данные подставляются автоматически.

4.4 Возможности

4.4.1 API вебмастеров

API	Описание
Пост-бек (Webhook)	Получите информацию о заказах, товарах, а также изменении статуса партнерской программы.
XML с заказами	Получите XML-список заказов.
Условия офферов	Получите список офферов в формате JSON или XML.
Лендинги	Получите список лендингов в формате JSON или XML.
Акции	Получите список акций в формате JSON или XML.
Промокоды	Получите список купонов в формате JSON или XML.
Баннеры	Получите список баннеров в формате JSON или XML.
Категории	Получите список категорий в формате JSON или XML.
Типы трафика	Получите список типов трафика в формате JSON или XML.
Фиды	Получите список товарных фидов в формате JSON или XML.
	Подсказка: Товарные фиды, сгенерированные и сохраненные вами фиды по определенному набору условий.
Общие фиды	Получите список общих товарных фидов в форматах XML, GSF и CSV.
	Подсказка: Общие товарные фиды – это товарные фиды рекламодателей с уже сформированными вашими кликовыми ссылками.
Статистика использования купонов	Получите статистику использования купонов в формате JSON или XML.
Cakelink	Перенаправьте пользователей на сайт партнёрской программы напрямую, без использования редиректора.
Бестселлеры	Получите список бестселлеров в формате JSON или XML.

4.4.2 API рекламодателей

API	Описание
Статистика заказов	Получите список заказов в формате JSON.
Статистика визитов	Получите список визитов в формате JSON.

4.5 Обновления

4.5.1 11 августа 2022

API	Что изменилось
API со статистикой визитов (для рекламодателей)	Добавили API для получения списка визитов в формате JSON.

4.5.2 16 февраля 2022

API	Что изменилось
API для заказов (для вебмастеров)	Обновили описание. Параметр <code>days=1</code> применяется по умолчанию; Параметр <code>basket=1</code> выводит все заказы, в том числе с корзиной.

API	Что изменилось
Постбек вебмастеров (Webhook)	Добавили поле <i>Статус выплаты вознаграждения</i> в разделе Получать информацию о заказах . В данное поле передаются статусы выплаты вознаграждения, актуальные для вашей системы. Чтобы задать пользовательские статусы, перейдите в раздел <i>Статусы выплаты вознаграждения</i> (перед кнопкой Сохранить) и заполните поля своими значениями.

4.5.3 11 февраля 2022

API	Что изменилось
API со статистикой заказов (для рекламодателей) API для заказов (для вебмастеров)	Добавили GET-параметры <code>update_from</code> и <code>update_to</code> . Позволяют отобразить заказы, обновленные в промежутке указанных дат (включительно) в формате <code>YYYY-MM-DD</code> .

4.5.4 9 февраля 2022

API	Что изменилось
Постбек (Webhook) (для вебмастеров)	Добавили поле <i>Дата заказа</i> . Позволяет передавать вебмастеру дату заказа в постбеке.

4.5.5 2 февраля 2022

API	Что изменилось
API для заказов (для вебмастеров)	Добавили примеры получения списка заказов по выбранному офферу/ID оффера.

4.5.6 1 февраля 2022

API	Что изменилось
API для заказов (для вебмастеров)	Добавили GET-параметр <code>reward_state</code> . Позволяет отобразить заказы с указанным статусом выплаты вознаграждения. Содержит поля: <i>not_payable</i> - не подлежит выплате - заказ находится в статусе Новый или Отменён; <i>awaiting_adv_payment</i> - ожидает оплаты рекламодателем - заказ одобрен, но счёт рекламодателя не оплачен; <i>ready_for_payment</i> - готово к выплате - заказ одобрен, счет рекламодателя оплачен, но заказ не привязан к оплаченному счету или выплате вебмастера; <i>paid</i> - выплачено - заказ одобрен, счет рекламодателя оплачен, вознаграждение выплачено вебмастеру.

4.5.7 11 сентября 2021

API	Что изменилось
API лендингов; API акций; API баннеров; API купонов (для вебмастеров)	Добавили поле <code>short_name</code> , в которое выводятся данные из поля <i>Краткое название</i> при создании промокодов.