

Долями SDK 2.0. Общая документация по интеграции, работе с хуками, взаимодействию с бекендом.

- [Подготовка](#)
- [Интеграция SDK](#)
- [Работа с заявкой, общий флоу](#)
- [Хуки](#)
- [Рекомендации по отображению заказа](#)
- [Подтверждение заказа](#)
- [Метод Cancel](#)
- [Метод Refund](#)
 - [При покупке использовались баллы программы лояльности](#)
 - [При покупке не использовались баллы программы лояльности](#)
- [Создание тестовой заявки \(демо-заявка\)](#)
- [Старт продаж](#)
- [Схема заявки](#)

Подготовка

1. Оставьте заявку на подключение по [ссылке](#)
2. С вами свяжется персональный менеджер и запросит информацию, необходимую для подключения к сервису Долями.
3. Получите mtls-сертификат по ссылке [OpenAPI](#) для безопасного обмена данными
4. Мы проверим предоставленную информацию и вышлем:
 - partner.id необходимый для инициализации SDK в вашем мобильном приложении.
 - Логин и пароль от демо-учетной записи. Теперь вы можете пройти демо-флоу (создать демо-заявку и использовать API для работы с демо-заявкой).
 - После прохождения всех демо-сценариев необходимо запросить логин и пароль от боевой учетной записи. Вам будет создана боевая уз. Теперь вы можете пройти флоу на проде (создать продовую заявку и использовать API для работы с продовой заявкой).

Интеграция SDK

1. Прочтите эту документацию до конца.
2. Интегрируйте SDK по соответствующей документации:

- IOS: [документация доступна по ссылке](#)
- Android: [документация доступна по ссылке](#)

Обратите внимание. При прохождении флоу в SDK, ваш бекенд не должен вызывать метод create, SDK делает это на своей стороне!

Работа с заявкой, общий флоу

- Покупатель формирует заказ и выбирает "Оплатить Долями".
- Покупатель авторизуется - вводит номер телефона и код из смс. Если покупатель авторизовался ранее, то повторно авторизовываться не надо, и этот шаг пропускается.
- SDK создает заказ в сервисе Долями. Бекенд партнера не должен вызывать метод create, SDK делает это на своей стороне

Заказ живет 24 часа с момента создания, в течение этого времени клиент может оплатить заказ Долями. После заказ отменяется по тайм-ауту.

- Покупатель указывает ФИО, дату рождения, адрес электронной почты, если это требуется.
- Сервис Долями проверяет персональные данные клиента:
 - В случае успешной проверки клиент переходит на этап оплаты. Магазин получит вебхук Approved.
 - В случае неуспешной проверки мы откажем клиенту в услуге. Магазин получит вебхук Rejected.
- Клиент переходит на экран оплаты и оплачивает первую долю от всей суммы, денежные средства холдируются, далее возможны несколько вариантов развития событий:
 - Если для партнера включено автоматическое подтверждение, "автокомит", то денежные средства списываются и партнер получает вебхук Committed. Автоматическое подтверждение означает, что партнеру не надо вызывать Commit, сервис Долями сам подтверждает заказ автоматически вместо партнера.

- Если для партнера не включено автоматическое подтверждение, партнер получает вебхук `Wait_for_commit`. С этого момента мы ждем подтверждение от партнера в течение 72 часов:
 - Если партнер не подтвердит заказ, заказ будет отменен. Партнеру будет направлен хук `Canceled`.
 - Если партнер подтвердит заказ, вызвав `Commit`, с клиента будут списаны денежные средства и партнеру будет направлен хук `Committed`.
- Далее сервис Долями рассчитывает финальное расписание платежей и после его формирования, магазин получает вебхук `Completed`.
- Сервис Долями каждые две недели списывает оставшиеся три части стоимости в соответствии с сформированным расписанием.

После подтверждения заказа, `Commit`, сервис Долями переведёт деньги на расчётный счёт магазина на следующий рабочий день. Подробные условия см. в договоре.

Внимание! Для методов `Create`, `Cancel`, `Info`, `Commit`, `Refund` необходимо использовать логин и пароль, полученные от персонального менеджера, также при вызове запросов необходимо использовать сертификат, полученный по ссылке: [OpenAPI](#)

Хуки

После создания демо-заявки, аналогично как и для продовольственной, после событий описанных ниже на ваш `notification url` будут приходить вебхуки со статусами:

- `Approved` - отправляется когда клиенту была одобрена возможность воспользоваться Долями.
- `Wait_for_commit` - отправляется после оплаты 1-го платежа (средства захолдированы, не списаны). После этого мы готовы принять от магазина метод `Commit`. Этим самым магазин подтверждает заказ. Внимание! Если у партнера включено автоматическое подтверждение (автокомит), то этот хук не отправляем, отправляем сразу следующий `Committed`.
- `Committed` - отправляем после получения `Commit` от партнера, после того как списали денежные средства клиента.
- `Completed` - отправляем после расчета финального расписания платежей.

- Rejected - отправляется когда клиенту отказано в возможности воспользоваться Долями.
- Canceled - отправляем когда произошла отмена заявки (партнером, по таймауту и др. причины).

Подробнее про вебхуки тут <https://dolyame.ru/develop/help/webhooks/>

Рекомендации по отображению заказа

При завершении работы SDK вы получите статус:

- Success - оплата успешно прошла. Означает, что заказ оплачен и подтвержден бэкендом Долями
- Pending - оплата в процессе, пользователь вышел из SDK до окончания оплаты. Означает, что клиент закрыл SDK в момент проведения оплаты, либо от бэкенда Долями не было получено подтверждение после успешно проведенной оплаты
- Failure - заказ не был оформлен/оплачен из-за ошибки. Например клиент получил отказ в предоставлении услуги Долями, не хватило денег на карте и др. ошибки.
- Dismiss - заказ не был оформлен/оплачен т.к. клиент вышел из SDK не пройдя флоу до конца, нажав кнопку "закрыть".

Вне зависимости от того какой результат получили от SDK или если никакой результат не получили, партнеру рекомендуется заложить следующую логику:

- Пока от бекенда Долями не пришел хук со статусом Wait_for_commit или Committed или Completed, то считать/отображать заказ неоплаченным.
- Рекомендуется отображать заказ в личном кабинете (у партнера) оплаченным в следующих случаях:
 - Пришел хук со статусом Wait_for_commit - рекомендуется отображать заказ оплаченным, если есть возможность можно детализировать, что первый платеж захолдирован, ожидается подтверждение заказа от партнера, после чего произойдет списание захолдированных средств за первый платеж.
 - Пришел хук со статусом Committed - необходимо отображать заказ оплаченным, партнер подтвердил заказ и денежные средства были списаны с клиента.

- Пришел хук со статусом Completed - необходимо отображать заказ оплаченным, сформировано финальное расписание графика платежей по заказу.
- Можно за актуальным статусом заказа также сходить бек ту бек (метод info).
 - Если статус заказа WaitForCommit - аналогично, как при получении хука со статусом Wait_for_commit, см. выше.
 - Если статус заказа Committed - считаем что заказ оплачен клиентом, как при получении хука со статусом Committed /Completed.
 - Если статус заказа Completed - значит заказ полностью оплачен клиентом (все 4е платежа) или по заказу был полный возврат. Рекомендуется отображать такой заказ как завершенный.

Статус хука может не соответствовать статусу заказа, это разные сущности. Подробнее ниже:

Hook.Status	Когда хук отправляется	Order.Status (статус заказа для партнера, который вернет метод info)	Описание статуса заказа
-	-	New	Заказ создан и не проходил скоринг. Заказ автоматически будет отменен через 24 часа с момента создания.
Approved	Отправляем когда клиенту была одобрена возможность воспользоваться Долями.	Approved	Заказ одобрен клиенту. Заказ автоматически будет отменен через 24 часа с момента создания

Wait_for_commit	Отправляем после оплаты 1-го платежа (средства захолдированы, не списаны). После этого мы готовы принять от магазина метод Commit. Этим самым магазин подтвердит заказ. Внимание! Если у партнера включено автоматическое подтверждение (автокомит), то этот хук не отправляем, отправляем сразу следующий Comitted.	WaitForCommit	Заказ оплачен клиентом, средства захолдированы. Ждем Commit от партнера. Если магазин не вызовет Commit заказ будет отменен через 72 часа после перехода в этот статус.
Committed	Отправляем после списания денежных средств с клиента.	Committed	Заказ подтвержден партнером, либо автоматически подтвержден, если для партнера включено автоматическое подтверждение. Средства списаны с клиента.
Completed	Отправляем после формирования финального расписания графика платежей.		
Rejected	Отправляем когда клиенту отказано в возможности воспользоваться Долями.	Rejected	Заказ не одобрен клиенту
Canceled	Отправляем когда произошла отмена заявки (партнером, по таймауту и др. причины)	Canceled	Заказ отменен

-	-	Completed	Заказ полностью оплачен или по заказу был полный возврат
---	---	-----------	--

Подробнее про методы бек ту бек см. тут: <https://dolyame.ru/develop/help/api/>

Необходимо проверять IP-адрес с которого пришел вебхук. Уведомления банка приходят с маски сети 91.194.226.0/23. IP адрес первого хоста - 91.194.226.1 и IP адрес последнего хоста – 91.194.227.254 , хостов в сети 510

Подтверждение заказа

После того как вы получили хук `Wait_for_commit` или если при проверке статуса заказа вам пришел статус `WaitForCommit`, вызовите метод `Commit` с теми же параметрами заказа, которые отправили в SDK.

Внимание!

- Метод `Commit` должен быть вызван в течение 72 часов от момента вызова метода `Create`, в противном случае заказ будет отменен.
- Состав заказа может полностью отличаться, в т.ч. наименования товаров и `sku` могут быть абсолютно другими.
- Сумма подтвержденного заказа не может превышать изначальную сумму заказа, указанную при создании заказа (переданную в SDK). Сумма заказа в `Commit` может быть либо равной, либо отличаться в меньшую сторону от суммы переданной в SDK.

Если в `Commit` будет меньшая сумма, мы пересчитаем платежи и вернем клиенту разницу между суммой переданной в SDK и суммой в `Commit`. То есть мы спишем только часть изначально захолдированной суммы за первый платеж, остальное вернется клиенту. График будет пересчитан с учётом новой суммы. В ответ партнеру будут направлены вебхуки `Committed` и `Completed`.

Метод `Cancel`

Чтобы отменить заказ, партнер может вызвать метод `Cancel`. Отменить заказ можно только в статусах `New/Approved /WaitForCommit`. В ответ получите вебхук `Canceled`.

Метод Refund

Сделать полный или частичный возврат заказа можно в статусе Committed. В случае успеха, в ответ вы получите код "200".

Внимание! Просьба передавать в sdk, в order.items, и в refund, в returned_items, не только наименования товаров (name), но и sku. В таком случае при возврате товар будет идентифицироваться по sku и мы исключим проблемы которые могут возникнуть при идентификации по наименованию из-за некорректного наименования товара (пробелы, пентаграммы, символы и т.д.).

Для того, чтобы оформить возврат по заказу необходимо вызвать метод [Refund](#).

При покупке использовались баллы программы лояльности

Клиент покупает 2 футболки на сайте магазина, каждая из них стоит 1100 рублей. Общая сумма покупки равна 2200 рублей. У клиента есть 200 баллов, один балл равен одному рублю. 200 рублей клиент оплачивает баллами, оставшуюся часть (2000) - с помощью сервиса Долями.

SDK вызывает метод [Create](#) со следующими параметрами:

- order.prepaid_amount = 200
- item quantity = 2
- item price = 1100
- order.amount+order.prepaid_amount = i.quantity*i.price 2000+200=1100*2

Далее, если у партнера не настроено автоматическое подтверждение, то после получения хука Wait_for_commit, магазин вызывает метод [Commit](#) и передает такие же значения в поля amount, prepaid.amount, items, что и в SDK.

После подтверждения заказа клиент решает вернуть одну из футболок, тогда магазин вызывает метод Refund и передает следующие значения:

- refunded.prepaid_amount = 100
- returned.item quantity = 1
- returned.item price = 1100
- amount+refunded.prepaid_amount = returned.item.quantity*returned.item.price 1000+100 = 1*1100

Если возврат прошел успешно в ответе получите код 200.

При покупке не использовались баллы программы лояльности

Клиент покупает 2 футболки на сайте магазина, каждая из них стоит 1100 рублей. Общая сумма покупки равна 2200 рублей. У клиента нет баллов лояльности, клиент оплачивает заказ (2200 рублей) с помощью сервиса Долями. SDK вызывает метод [Create](#) со следующими параметрами:

- `order.prepaid_amount = 0`
- `item quantity = 2`
- `item price = 1100`
- `order.amount+order.prepaid_amount = i.quantity*i.price2200+0 = 2*1100`

Далее, если у партнера не настроено автоматическое подтверждение, то после получения хука `Wait_for_commit`, магазин вызывает метод `Commit` и передает такие же значения в поля `amount`, `prepaid.amount`, `items`, что и в SDK. После подтверждения заказа клиент решает вернуть одну из футболок, тогда магазин вызывает метод `Refund` и передает следующие значения:

- `refunded.prepaid_amount = 0`
- `returned.item quantity = 1`
- `returned.item price = 1100`
- `amount + refunded.prepaid_amount = returned.item.quantity * returned.item.price1100+0=1 * 1100`

Если возврат прошел успешно в ответе получите код 200.

Создание тестовой заявки (демо-заявка)

Создание тестовой заявки описано в руководстве по интеграции SDK IOS и Android по ссылкам выше. Пройдите по всем вариантам.

Обращаем внимание что для работы с демо-заявками для использования методов бек ту бек необходимо использовать демо-учетную запись. Для продовых заявок вам будет выдана другая продовая учетная запись.

Старт продаж

- После тестирования и настройки sdk, методов и вебхуков можно перейти к созданию реальных заявок.
- Для создания реальных заявок определите параметр `isDemoFlow/DemoFlow:false`, подробнее в документации SDK.
- Расскажите своим клиентам о новом способе оплаты частями, используя [маркетинговые материалы](#).

Схема заявки

