

Минимальная поддерживаемая версия Android SDK - 24.

НО, если у вас минимальная поддерживаемая версия Android SDK - 21, то вы все еще можете подключить библиотеку, но ее функционал не будет работать на версиях ниже Android SDK - 21. Вы можете воспользоваться методом `DolyameSdk.isAvailable()` для проверки доступности sdk для телефона клиента.

Подготовительные работы на бэкенде

Для работы с SDK от бэкенда требуется:

- поддержка [вебхука Долями](#)
- умение вызывать [Commit](#), [Info](#), [Cancel](#) и [Refund](#) для работы с заказом

URL, на который будет приходить вебхук, передайте вашему контакту из Долями или укажите в поле `notificationUrl` (более подробно про это поле далее в документации).

При работе с SDK Долями бэкенд НЕ должен вызывать `Create` !

Добавление зависимости

SDK Долями доступен для скачивания из MavenCentral репозитория.

Добавьте зависимость в `build.gradle` . Отслеживать актуальную версию SDK можно например на [mvnRepository](#)

```
dependencies {  
    ...  
    implementation 'ru.tinkoff.dolyame:sdk:x.x.x'  
}
```

Для подключения SNAPSHOT-версии библиотеки необходимо добавить репозиторий в `build.gradle` на уровне проекта, либо в конкретном модуле:

```
repositories {  
    maven {  
        url "https://oss.sonatype.org/content/repositories/snapshots/"  
    }  
}
```

Также в `build.gradle` модуля добавьте зависимость с суффиксом `-SNAPSHOT` :

```
dependencies {  
    ...  
    implementation 'ru.tinkoff.dolyame:sdk:x.x.x-SNAPSHOT'  
}
```

Добавление сертификатов безопасности

Так же необходимо добавить в `[network-security-config][network-security-config]` содержащий сертификаты от минцифр и доп. сертификат от тинькофф. Пример выглядит он так:

```
<?xml version="1.0" encoding="utf-8"?>  
<network-security-config xmlns:tools="http://schemas.android.com/tools">  
    <base-config>  
        <trust-anchors>  
            <!-- Trust preinstalled CAs -->  
            <certificates src="system" />  
            <!-- Additionally trust user added CAs -->  
            <certificates  
                src="user"  
                tools:ignore="AcceptsUserCertificates" />  
            <!-- related to certificates from ssl-trusted-certs library-->  
            <certificates  
                src="@raw/ministry_of_digital_development_root_cert"  
                tools:ignore="NetworkSecurityConfig" />  
            <certificates  
                src="@raw/tinkoff_root_cert_production"  
                tools:ignore="NetworkSecurityConfig" />  
        </trust-anchors>  
    </base-config>  
</network-security-config>
```

Формирование конфигурации для SDK

В SDK необходимо передать следующую конфигурацию:

DolyamePurchase

Название	Тип	Описание
partner	DolyamePurchase.Partner	Инфо о продавце
order	DolyamePurchase.Order	Инфо о заказе
clientInfo	DolyamePurchase.ClientInfo	Инфо о покупателе

<code>isDemoFlow</code>	<code>Boolean</code>	true, если нужно протестировать различные исходы на демо-заявке
<code>showPhoneNotification</code>	<code>Boolean</code>	Флаг для отображения клиенту предупреждения о передаче его номера телефона сервису Долями. Рекомендуется передавать <code>true</code> только в случае, если Google Play требует такое предупреждение, в остальных случаях можно не передавать

`DolyamePurchase.Partner`

Название	Тип	Описание
<code>id</code>	<code>String</code>	Уникальный идентификатор клиента-интегратора
<code>notificationUrl</code>	<code>String?</code>	Url для отправки вебхука о результатах оплаты. Вы можете либо задать дефолтный url на нашем бэкенде (обратитесь для этого к персональному менеджеру из Долями), либо передавать его в этот параметр
<code>pointId</code>	<code>String?</code>	Уникальный идентификатор точки. Передается в случае наличия нескольких точек у клиента-интегратора на одной учетной записи

`DolyamePurchase.Order`

Название	Тип	Описание
<code>id</code>	<code>String</code>	Идентификатор заказа в магазине партнера
<code>amount</code>	<code>Double</code>	Сумма для оплаты через Dolyame. Должно соблюдаться условие <code>amount + prepaidAmount</code>

		<code>== items.sum { it.quantity * it.price }</code>
<code>prepaidAmount</code>	<code>Double</code>	Сумма аванса, внесенного клиентом через другие способы оплаты. Должно соблюдаться условие <code>amount + prepaidAmount == items.sum { it.quantity * it.price }</code>
<code>items</code>	<code>[DolyamePurchase.OrderItem]</code>	Список с позициями в заказе
<code>mcc</code>	<code>Int</code>	МСС код, с которым нужно совершить платеж (по умолчанию используйте 5311)

`DolyamePurchase.OrderItem`

Название	Тип	Описание
<code>name</code>	<code>String</code>	Наименование товара
<code>quantity</code>	<code>Double</code>	Количество позиций данного наименования
<code>price</code>	<code>Double</code>	Цена одной позиции
<code>sku</code>	<code>String?</code>	sku товара (идентификатор уникальный для каждого товара/артикул/первичный ключ)

`DolyamePurchase.ClientInfo`

Название	Тип	Описание
<code>id</code>	<code>String</code>	Идентификатор покупателя, должен быть не пустым
<code>firstName</code>	<code>String?</code>	Имя
<code>lastName</code>	<code>String?</code>	Фамилия
<code>middleName</code>	<code>String?</code>	Отчество
<code>phone</code>	<code>String?</code>	Номер телефона в формате +7xxxxxxxxxx
<code>birthday</code>	<code>String?</code>	Дата рождения в формат dd.mm.yyyy
<code>email</code>	<code>String?</code>	E-mail

Поля ClientInfo не являются обязательными, но если данная информация о покупателе

есть, то

передавайте её в SDK для ускорения авторизации и, соответственно, покупки.

Открытие экрана SDK

Основной экран SDK является отдельной Activity. Точкой входа является класс `DolyameSdk`, для показа

экрана Долями необходимо вызвать один из его методов:

- При использовании Activity и Activity Result APIs из AndroidX. `DolyameObserver` обязан быть создан строго до `Activity#onStart`, параметр `observerKey` должен быть уникальным для каждой `Activity` или `Fragment` в случае запуска SDK из разных `Activity / Fragment` - по этому ключу будет возвращаться результат оплаты из SDK в `DolyameObserver`

```
class BasketActivity : AppCompatActivity() {
    private val observer = DolyameObserver(
        registry = activityResultRegistry,
        observerKey = OBSERVER_KEY
    ) { result: DolyameResult ->
        ...
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        lifecycle.addObserver(observer)
        ...
        binding.openDolyame.setOnClickListener {
            val purchase = DolyamePurchase(...)
            DolyameSdk.openDolyame(
                observer = observer,
                purchase = purchase,
            )
        }
    }

    private companion object {
        const val OBSERVER_KEY = "BasketActivityKey"
    }
}
```

- При использовании Fragment и Activity Result APIs из AndroidX

```
class BasketFragment : Fragment() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val observer =
```

```

        DolyameObserver(
            registry = requireActivity().activityResultRegistry,
            observerKey = OBSERVER_KEY
        ) { result: DolyameResult ->
            ...
        }
        lifecycle.addObserver(observer)
        ...
        binding.openDolyame.setOnClickListener {
            val purchase = DolyamePurchase(...)
            DolyameSdk.openDolyame(
                observer = observer,
                purchase = purchase,
            )
        }
    }

    private companion object {
        const val OBSERVER_KEY = "BasketFragmentKey"
    }
}

```

- С использованием Activity и onActivityResult. Необходимо переопределить onActivityResult и добавить обработку результата вызова Долями

```

class BasketActivity : AppCompatActivity() {
    val DOLYAME_REQUEST_CODE = 1234
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        ...
        binding.openDolyame.setOnClickListener {
            val purchase = DolyamePurchase(...)
            DolyameSdk.openDolyame(
                activity = this,
                purchase = purchase,
                requestCode = DOLYAME_REQUEST_CODE,
            )
        }
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        if (requestCode == DOLYAME_REQUEST_CODE) {
            val result = DolyameSdk.parse(resultCode, data)
            ...
        }
    }
}

```

- Аналогично есть метод для вызова с использованием Fragment. Во фрагменте необходимо переопределить onActivityResult и добавить обработку результата вызова Долями, аналогично коду в разделе для Activity

```
// in BasketFragment.kt
DolyameSdk.openDolyame(
    fragment = this,
    purchase = purchase,
    requestCode = DOLYAME_REQUEST_CODE,
)
```

Обработка результата работы SDK

Sdk возвращает результат в виде объекта класса `DolyameResult`. Возможны 4 состояния:

- Success - оплата успешно прошла. Означает, что заказ оплачен и подтвержден бэкендом Долями
- Pending - неиспользуемый параметр, будет удален в следующих версиях SDK. Возврат результата Pending заменен на возврат Failure из SDK
- Failure - заказ не был оформлен по некой причине. Содержит поле `exception: Throwable?`, в котором может лежать поясняющая ошибка
- Dismiss - пользователь вышел из Sdk до начала оплаты, без получения ошибок

Авторизация в SDK

При авторизации в SDK сохраняется полученный `ClientInfo.id`. При последующих запусках SDK происходит проверка `ClientInfo.Id`:

- если был получен идентичный id, то авторизация в SDK сохраняется
- если был получен id, отличный от сохранённого при предыдущей авторизации, то пользователь будет разлогинен и должен будет заново пройти процесс авторизации

Google Pay

Если у вас не подключен Google Pay:

- Необходимо создать в Google Business Console новый merchant id
- Для этого merchant указать тип интеграции tokenization method: gateway
- Интегрировать SDK Долями
- Провести тестовый платеж
- Отправить запрос в Google для prod-доступа: <https://developers.google.com/pay/api/android/guides/test-and-deploy/request-prod-access>
- После получения доступа, опубликовать приложение в Google Play

Тестирование

Для тестирования различных исходов заявки необходимо передать в `DolyamePurchase` параметр `isDemoFlow: true`. Будет создана демо-заявка и реальная оплата заказа производиться не будет. Все ответы от SDK и уведомления от бэкенда Долями будут отправлены аналогично прод-версии. Для выбора исхода доступно 3 сценария:

- отказ скоринга
- успешная оплата 1го платежа
- неуспешная оплата 1го платежа

Для полноценного тестирования, при открытии SDK обязательно передавайте параметр `partner.notificationUrl` - в данном случае на этот адрес будут отправлены уведомления об оплате от Долями для демо-заявки. Иначе при тестировании уведомления никуда отправлены не будут (даже на ваш дефолтный notificationUrl на нашем бэкенде).