API Documentation - UPI SDK Integration

Introduction:

This document provides the necessary information and instructions for integrating the UPI (Unified Payments Interface) SDK into your Android application. The UPI SDK allows users to make payments using UPI-based payment apps. This document will guide you through the process of initiating the SDK, sending data to the SDK, receiving responses from the SDK, and handling callbacks.

Table of Contents:

1. Initiate the SDK:

Endpoint: POST https://blinkpe.co.in/api/p2p/upiSdk/initiate

Request Body:

```json
{
   "token": "xxxxxxxxxx",
   "companyCode": "BLINK123",
   "orderAmount": "1",
   "clientRefId": "BLP00167343"
}
```

- `token` (string): A token for authentication.

- `companyCode` (string): Your company code.
- `orderAmount` (string): The amount for the order.
- `clientRefId` (string): A unique client reference ID.

If the response contains the `statuscode` "TXN," proceed to initiate the SDK.

2. SDK Download Link:

Download the UPI SDK from the following link:

https://drive.google.com/drive/folders/1QczAyBp2i6-lkPojuBR_wznMagVXQ1fZ?usp=sharing

3. Manifest File Configuration:

Add the following permission in your AndroidManifest.xml file:

```xml
<uses-permission android:name="android.permission.INTERNET"/>
```

4. Gradle Build Configuration:

In the project-level build.gradle file, add the following plugin:

```groovy
id("com.google.devtools.ksp") version "1.8.10-1.0.9" apply false
```

In the app-level build.gradle file, add the following plugins:

```groovy
id 'kotlin-android'
id 'com.google.devtools.ksp'
```

5. ViewBinding Configuration:

Inside the `android` block of your app-level build.gradle file, add the following lines:

```groovy
```

```
android {
    // ...

    buildFeatures {
        viewBinding = true
    }
}
```

6. Dependencies:

Add the following dependencies to your app-level build.gradle file:

```groovy
implementation 'com.facebook.shimmer:shimmer:0.5.0'

def moshi_version = "1.13.0"
implementation "com.squareup.moshi:moshi:$moshi_version"
ksp "com.squareup.moshi:moshi-kotlin-codegen:$moshi_version"

def retrofit_version = "2.9.0"
implementation "com.github.skydoves:sandwich:1.2.7"
implementation "com.squareup.retrofit2:retrofit:$retrofit_version"
implementation "com.squareup.retrofit2:converter-moshi:$retrofit_version"
implementation "com.squareup.okhttp3:logging-interceptor:5.0.0-alpha.9"

def koin_version = "2.2.3"
implementation "io.insert-koin:koin-android:$koin_version"
implementation "io.insert-koin:koin-androidx-viewmodel:$koin

_version"
implementation "io.insert-koin:koin-androidx-fragment:$koin_version"
implementation "io.insert-koin:koin-androidx-scope:$koin_version"
```

```
ksp "io.insert-koin:koin-gradle-plugin:$koin_version"


implementation "com.jakewharton.timber:timber:5.0.1"
implementation 'com.github.bumptech.glide:glide:4.15.1'
```

7. Sending Data to the SDK:

To initiate the UPI SDK, use the following code snippet:

```kotlin
val i = Intent(this, BlinkpeActivity::class.java)
   i.putExtra(com.payments.blinkpe.utils.Constants.UPI_PAY_TOKEN_KEY ,"8n2JL1x7dRtZcP9yS6g5K4q3wX2v1B")
i.putExtra(com.payments.blinkpe.utils.Constants.UPI_PAY_COMPANY_CODE ,"SFPAY202356")
i.putExtra(com.payments.blinkpe.utils.Constants.UPI_PAY_AMOUNT ,"2")


startActivityForResult(i, com.payments.blinkpe.utils.Constants.UPI_PAY_RESULT_CODE)
```

Replace the placeholders with the appropriate values:

- `com.payments.blinkpe.utils.Constants.UPI_PAY_TOKEN_KEY`: The UPI pay token.
- `com.payments.blinkpe.utils.Constants.UPI_PAY_COMPANY_CODE`: Your company code.
- `com.payments.blinkpe.utils.Constants.UPI_PAY_AMOUNT`: The amount for the order.

8. Receiving Response from the SDK:

Override the `onActivityResult` method in your activity to receive the response from the UPI SDK:

```kotlin
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)


    try {
        if (requestCode == com.payments.blinkpe.utils.Constants.UPI_PAY_RESULT_CODE) {
            if (data != null) {
```

```
// show payment success failed ui with payment data
if(data.getParcelableExtra<com.payments.blinkpe.utils.UPIAppResponse>(com.payments.blinkpe.utils.Constants
.UPI_PAY_ON_ACTIVITY_PAYMENT_STATUS)  !=null){

                    val
paymentStatus=data.getParcelableExtra<com.payments.blinkpe.utils.UPIAppResponse>(com.payments.blinkpe.
utils.Constants.UPI_PAY_ON_ACTIVITY_PAYMENT_STATUS)

// you can create Fragment or Activity or Dialog ui

startActivity(Intent(this@MainActivity,TxnSuccessActivity::class.java)

.putExtra("payment_status_data",paymentStatus))

                 }else {

                    val statuscode =
data.getStringExtra(com.payments.blinkpe.utils.Constants.UPI_PAY_ON_ACTIVITY_STATUS_CODE)

                    val message =
data.getStringExtra(com.payments.blinkpe.utils.Constants.UPI_PAY_ON_ACTIVITY_STATUS_MESSAGE)

                    Toast.makeText(this, "==$message", Toast.LENGTH_SHORT).show()

                 }


              }       }

   } catch (e: Exception) {

       e.printStackTrace()

   }

}
```

9. Callback Information:

The following status codes can be received as a response in the `onActivityResult` method:

- `TXN`: Indicates a successful transaction.

- `TXF`: Indicates a failed transaction.

- `ERR`: Indicates an error.

Additional Information:

- `clientrefId`: The unique client reference ID passed during SDK initiation.

- `orderId`: The order ID obtained from the API panel.

- `refNo`: The UTR (Unique Transaction Reference) number.
- `orderAmount`: The amount of the transaction.

Please note that this document provides an overview of integrating the UPI SDK into your Android application. For more detailed information and code implementation, please refer to the provided code snippets and the SDK documentation.