Mobile SDK - Test Documentation

Table of Contents

Limitations

Android

Regarding the Javascript/CSS capabilities of WebViews on Android, please have a look at the following Websites:

- CanIUse.com
- MobileHtml5.com
- Html5test.com

Errors

Error Response odes

Link: https://docs.axepta.bnpparibas/display/DOCBNP/Response+codes

Format of Error codes

Error code example: 2 105 0014 = [severity] [category] [details]

1. The first value defines the severity of the action.
   All values higher than 0 represent alerts or errors.
   Note: an error code can also occur if a transaction did not succeed
   (e.g. bank refused a payment)

2. The 2nd up top the 4th value of the action code indicate the error category.
   The category include encryption errors (001) and formatting errors (010) as well as payment methods such as credit card payments (100), electronic direct debits (110) and Cash&Go (140).

3. The 5th until the 8th value of the action code indicates the details of the action. The details may contain indications of configuration problems such as missing TerminalIDs (0047), downtimes of the data processing centre of the issuing bank (121). However, they also include regular declining of credit card payments due to expired cards (110) or declined-messages (0100).

Naming Conventions

1. All namings should match Axepta’s namespace.
2. User 3 letter prefix.
3. The names of the params in CMPPaymentData should be equal to the actually query param names that are sent to retrieve payment forms.

Implementation

Merchant parameter validation

A Merchant should insert values to a set of mandatory parameters before he requests any payment forms or interact with Axepta API in general.
From the side of the mobile SDK, in order to provide immediate response to the Merchant if the value of these mandatory parameters is not as expected, it is realizing internal validation based on Axepta‘s documentation.

Payment request parameters

Find a complete list of payment request parameters here:
https://docs.axepta.bnpparibas/display/DOCBNP/Card+payments+Integration+guide

RegEx patterns

alphanumericEx: "^[a-zA-Z0-9\x20_'.,@:?!()$#/\\+-]{%lu,%lu}$"
numericEx: "^(0|[1-9][0-9]{%lu,%lu})$"
textEx: "^[a-zA-Z\x20_'.,@:?!()$#/\\+-]{%lu,%lu}$"

Where ‘%lu’ is an 64-bit unsigned integer defining the char range of the string

Selection of Payment request parameters

Parameter	RegEx pattern	Acceptable char range	Error-code (internal)	Localization
merchantID	alphanumericEx	{1 - 30}	10001	“Please specify a MerchantId”
transID	alphanumericEx	{1 - 64}	10003	"Please specify a Payment-TransID parameter"
Amount	numericEx	{1 - 12}	10004	"Please specify a Payment-Amount parameter"
Currency	textEx	{1 - 3}	10005	"Please specify a Payment-currency parameter"
URLSuccess	alphanumericEx	{1 - 256}	10006	"Please specify anPayment-URLSuccess parameter"
URLFailure	alphanumericEx	{1 - 256}	10007	"Please specify anPayment-URLFailure parameter"
URLNotify	alphanumericEx	{1 - 256}	10008	"Please specify a Payment-URLNotify parameter"

Axepta payment - form URL construction

All the payment form URLs should have the following mandatory parameters:

1. **“MerchantID”**
2. **“Template”**
3. **“Data”**
    a. **PayID**
    b. **TransID**
    c. **Amount**
    d. **Currency**
4. **“Len”**

MerchantID

Name: “MerchantID”
Value: *“User’s MerchantID provided by Axepta”*

Template

Name: “Template”
Value: “ct_responsive”

Data

Name: “Data”
Value: The value consists of an encrypted aggregation of url query parameters.
These parameter are:

All parameters from CMPPaymentData in the form of

Name: “*CMPPaymentData.propertyName*”
Value: “*CMPPaymentData.propertyValue*”

PLUS

1. **Name**: “MerchantID”
   Value: *“User’s MerchantID provided by Axepta”*

2. **Name**: “MAC”
   Value: HMAC using *“payID*transID*merchantID*Amount*Currency”* string as input

For example, an unencryted "Data" parameter's value could be:
MerchantID=Exozet_test&MAC=2c3d3f05ab94a5a2fac34afa004cd09bf192181c7fffcaaf88d3b95d7e6edb6c&Amount=100&Currency=EUR&transID=c28e4434-20db-4518-9674-727c4c09f561&RefNr=c28e4434-20db-4518&OrderDesc=Test:0203&AddrCity=Berlin&FirstName=Lorem&LastName=Ipsum&AddrZip=12049&AddrStreet=Berlin&AddrCountryCode=DEU&URLSuccess=https://preview55.test.com/axepta/index.php&URLFailure=https://preview55.exozet.com/axepta/index.php&URLNotify=https://preview55.test.com/axepta/index.php&Phone=01775289124&LandingPage=Login&BuyerEMail=developer-buyer@test.com&ShopID=1&Subject=developer@test.com

Len

Name: "Len"
Value: "The length of the unencrypted 'Data' parameter's value"

Encryption

Blowfish encryption

HMAC

https://en.wikipedia.org/wiki/HMAC
In cryptography, an **HMAC** (sometimes expanded as either **keyed-hash message authentication code** or **hash-based message authentication code**) is a specific type of message authentication code (MAC) involving a cryptographic hash function and a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the data integrity and the authenticity of a message.

CI

Bitrise

Axepta portal

https://backoffice.axepta.bnpparibas

Testing payments

If an UNKNOWN_URL_SCHEME error occurs, please ensure to have used correct Success- & Failure und NoticeURLS for the according payment method.

Required

- testMerchantID
- blowfishEncryptionKey
- hmacKey

Credit card:

Until you have completed the programming, your payment platform account is in test mode: card payments are authorized but there is no cashflow because the payment platform has not instigated a capture.

Notice: Please use only small amounts between 0.11 and 2 euros in test mode because the card authorizations are authentic even in the test mode and reduce the limit of your card. If you use large amounts and reach the card limit, your card will no longer work temporarily.

In the case of successful payments, the payment platform returns the value zero in the Code parameter. If a payment fails, the Code parameter is greater than zero, for which there may be many reasons: an incorrect expiry date, an exceeded card limit or even a blocked card are just a few examples. You can find a full list of error codes in an Excel file (Error codes list).

Simulation Mode

To simulate an error, transmit the keyword Test in the **OrderDesc** parameter followed by the four-digit detailed error code, for example "Test:0110" to simulate an expired card. The payment platform then returns the four-digit detailed error code with the respective response-parameters.

Parameter	Value
OrderDesc	Test:0000

Example:
OrderDesc=Test:0000

Value „0000” simulates a successful transaction.
Value “0305” simulates a failed transaction.
Look up the Axepta error table to simulate other cases.
(https://docs.axepta.bnpparibas/display/DOCBNP/Response+codes)
Always use the 4 last digits of the error code.

Test case with timeout

A card payment is normally completed within one to two seconds. In a few cases however, payments may be terminated due to long processing times in the banking network. The payment platform terminates card payments after 90 seconds. If you prefer shorter timeouts our Support can configure it.

Important
Since no transaction data is transferred to the payment provider, in simulation mode no payment-method-related parameters will be returned (e.g. PayPal BillingAgreementID)

Configuration for PayPal

Required

- PayPal business account (and setup)
  Guide: https://docs.axepta.bnpparibas/display/DOCBNP/PayPal+direct+integration

To process PayPal transactions via the Payment platform as merchant you need to have a PayPal business account. You also need to enable the PayPal interface for use via Payment platform and to make the necessary settings (set parameters) for receiving payments.

The SDK needs information on PayPal-Merchant e-mail address. Note, that the PayPal response contains the delivery address for the order.

PayPal normal
PayPal Express Checkout Shortcut

- Connects customers from the shopping cart directly to PayPal, where they can then select their delivery and invoicing address.

The Payment platform also supports the separation of authorisation and capture for PayPal Express Checkout. You can set parameters via the Payment platform to control whether the amount on the customer's PayPal account should be blocked or captured immediately.

*****************Capture on Demand*****************

Capture Parameter
Auto or **Manual**: determines whether the requested amount is settled immediately or at a later stage.

Conditional Parameter TXType
Obligatory for Capture=Manual: Transaction type with possible values **Order** or **Auth** and **BAID** (BillingAgreementID). If txType Order is used no multiple transactions can be made.
Auth:

1.) capture=manual + txtype=order

The transaction is registered but not yet authorized (no PayPal credit check). The PayPal account can be closed after this action. As soon as a capture is received by Axepta, Axepta executes authorization and transaction.
"Order" is usually valid for 20 days.

2.) capture=manual + txtype=auth
Authorization for the transaction is carried out immediately.
The amount should be settled within 3 days.

Parameter: Capture

Typ:	alphanumeric
Länge:	6

Parameter: TxType

Typ:	alphanumeric
Länge:	5

********************Sandbox********************
To use the PayPal Sandbox, a Sandbox account is needed. After setting up your Sandbox account, inform Axepta of the according e-mail-address. After Axepta added the e-mail to your account, according tests can be carried out.

Information on PayPal Sandbox setup: https://developer.paypal.com/docs/business/test-and-go-live/sandbox/

Activation of / API Usage of PayPal Sandbox:

1.) only use the FireFox Browser
2.) open a Browser TAB
    a. open PayPal Sandbox and log in (https://developer.paypal.com-)
3.) Log in separately to the Sandbox PayPal configuration page http://www.sandbox.paypal.com
    log in with your Sandbox Business e-mail-address and according password
4.) The rest of the setup is analogue to setting up your LIVE account (except for Point 5)
5.) Grant API access for test_paypal_api_api1.axepta.com and check the checkmark for 3rd party providers

You can test
- PayPal Express Checkout Shortcut
- refunding
- authorization
- getting information on a transaction
- search transactions by different criteria (articles…)