PayWithEaseBuzz Payment kit Integration (Cordova)

Cordova

The PaywithEaseBuzz Cordova SDK makes it quick and easy to build an excellent payment experience in your app. We provide powerful UI screens and elements that can be used out-of-the-box to collect your user's payment details. We also expose the low-level API's that power those UI's so that you can build fully custom experiences. See our Cordova Integration Guide to get started.

This SDK allows you to integrate payments via PaywithEaseBuzz into your Cordova/Ionic app(Supporting Android/iOS Platforms). It currently supports the following modes of payments:

1. Credit / Debit Cards
2. Netbanking
3. Wallets/Cash Cards
4. UPI
5. Ola-Money
6. EMI

Features

1. **Simplified Security:** With fraud detection and prevention mechanisms, we provide the most protective layer for each transaction.We are also PCI-DSS compliant.
2. **Native UI:** We provide out-of-the-box native screens and elements so that you can get started quickly without having to think about designing the right interfaces.

Requirements

The PaywithEaseBuzz Cordova SDK is supported for Android and iOS platforms.The PaywithEaseBuzz Cordova SDK is compatible with apps supporting iOS 10 and above, requires Xcode 9.2 to build from source and Android Kitkat and above.

Note: According to PCI regulations, payment processing is not allowed on TLS v1 and TLS v1.1. Hence, if the device does not have TLS v1.2, the SDK will throw an error while initiating the payment . You can learn more about TLS versions here.

SDK Configuration

The PaywithEaseBuzz Cordova SDK is supported for Android and iOS platforms.To configure PaywithEaseBuzz Cordova SDK into your Cordova application you have to follow the below steps.:

1. Execute the below commands to install the plugin into the app.

```
$ cordova platform add ios
$ cordova platform add android
$ cordova plugin add /path/to/easebuzz-cordova-kit
$ cordova clean
$ cordova build
```

2. To open the upi app, you must add the below lines outside <application> into AndroidManifest.xmlfile.

```
<queries>
   <package android:name="com.google.android.apps.nbu.paisa.user" />
   <package android:name="net.one97.paytm" />
   <package android:name="com.phonepe.app" />
   <package android:name="in.org.npci.upiapp" />
   <package android:name="in.amazon.mShop.android.shopping" />
   <package android:name="com.whatsapp" />
</queries>
```

3. To open upi psp app for ios, you must make the following changes in your iOS app's Info.plist file.

```
<key>LSApplicationQueriesSchemes</key>
<array>
        <string>tez</string>
        <string>phonepe</string>
        <string>paytm</string>
        <string>gpay</string>
</array>
```

4. Add below meta tag :

```
<meta http-equiv="Content-Security-Policy" content="default-src 'self' data: gap: https://ssl.gstatic.com 'unsafe-eval'; style-src 'self' 'unsafe-inline'; media-src *; img-src 'self' data: content:;">
```

5. Add jquery-1.11.1.min.js and import it before index.js.

Note for iOS -

- The existing plugin does not work on the simulator. The iOS framework is shipped with simulator architectures , so you have to replace simulator Easebuzz.framework (from iOS Frameworks folder) with device Easebuzz.framework.
    a. **Copy file** >> Easebuzz.framework
    b. **File path** >> "paywitheasebuzz-cordova-lib-master/iOS Frameworks/ios-simulator/Easebuzz.framework
    c. **Replace with file** >> Easebuzz.framework
    d. **Replace path** >>paywitheasebuzz-cordova-lib-master/easebuzz-cordova-kit/src/ios
- Make sure at the time of uploading the app to APPStore, use the existing framework - iOS device framework.

Note for Android -

1. Make sure your android platforms build.gradle file and project.properties files are configured properly. Eg: If your defaultBuildToolsVersion is "29.0.2", defaultTargetSdkVersion is 29 and defaultCompileSdkVersion is 29 in build.gradle file then value of target in project.properties file should be android-29.

2. To enable android.useAndroidX=true and android.enableJetifier=true in your app you have to add the below line under <platform name="android"> section of the config.xml file and build your application.
   <preference name="AndroidXEnabled" value="true" />

Initiate Payment

This is a mandatory and important step for initiating the payment. To generate an accesskey you have to integrate the Initiate Payment API at your backend. After you successfully call the Initiate Payment API then you will get an access key which is the value of the data key of the response. You can check the Initiate Payment API Doc Click here.

Note -

- It is mandatory to integrate the **Initiate Payment API** at your Backend only ,also mandatory to integrate the **Transaction API**.
- Also, it is critical to configure **webhooks** as this will allow the Easebuzz system to 'push' data to your system whenever any event occurs, ensuring that both systems are always in sync.

Integration code

After having successfully set up the sdk configuration and Initiate Payment API at your backend, write the below code to start payment on the click of Pay button from your app.

1. To start payment you have to fetch the access key into your app which will get in response from the Initiate Payment api and pass that access key to the object as below code.

JavaScript (For Cordova) -

```
function proceedToPayment() {
 params = {
        "access_key": "Access key generated by the Initiate Payment API",
        "pay_mode": “This will either be “test” or “production”"
        } // Please do not change the key names in the parameters.

window.plugins.PayWithEasebuzz.EasebuzzPay(params, function(payment_response) {
        var result = payment_response.result
        var detail_response = payment_response.response
        // Do your own code to handle the payment response based on result and detail_response.
        },
        function(err) {
        // Do your own code to handle errors.
        });
        }
```

TypeScript (For Ionic) -

Add the below line above @component section

```
declare let window: any;

public proceedToPayment() {
    let params = {
        "access_key": "Access key generated by the Initiate Payment API",
        "pay_mode": “This will either be “test” or “production""
        }
        window['plugins'].PayWithEasebuzz.EasebuzzPay(params, function(payment_response) {
            var result = payment_response.result
            var detail_response = payment_response.response
            // Do your own code to handle the payment response based on result
            // and detail_response.
        }, function(err) {
            // Do your own code to handle errors.
        });
    }
```

In the above code you have to pass the access_key and pay_mode.
The access key will be like -
“555a2b009214573bd833feca997244f1721ac69d7f2b09685911bc943dcf5201” and you will be get it from the initiate payment API. The **pay_mode** parameter will either be "test" or "production". Your key and salt will depend on the pay_mode which you pass.

Handle Payment Response

The payment response will be received in the callback of the PayWithEasebuzz.EasebuzzPay method which is used to start the payment.

In the app you can get the detailed result by:

String result = data.result;

The following are the values of the result you can get

"payment_successfull"
"payment_failed"
"txn_session_timeout"
"back_pressed"
"user_cancelled"
"error_server_error"
"error_noretry"
"invalid_input_data"
"retry_fail_error"
"trxn_not_allowed"
"bank_back_pressed"

In the app you can get the detailed response by:

String detailed_response = data.payment_response;

This detailed_response can be parsed to get the details about the ongoing transaction.

Note: Description of the result values and detailed response are given at the end of the document.

Response Description

1. Payment result values description and equivalent constants

Response	Value
"payment_successfull"	It is a string constant and its value is “payment_successfull.” The result contains this value, if the payment transaction is completed successfully.
“txn_session_timeout”	It is a string constant and its value is “txn_session_timeout”. The result contains this value, if the payment transaction failed because of the transaction time out.
"back_pressed"	It is a string constant and its value is “back_pressed”. The result contains this value, if the user pressed the back button on coupons Activity
“user_cancelled”	It is a string constant and its value is “user_cancelled”. The result contains this value, if the user pressed the cancel button during the payment process.
“error_server_error”	It is a string constant and its value is “error_server_error”. The result contains this value, if the server side error occurred during the payment process.
“trxn_not_allowed”	It is a string constant and its value is “trxn_not_allowed”
“bank_back_pressed”	It is a string constant and its value is “bank_back_pressed”. The result contains this value if the user presses the back button on the bank page.
“invalid_input_data"	It is a string constant and its value is “invalid_input_data". The result contains this value if payment request input parameters are not valid.
“payment_failed”	It is a string constant and its value is “payment_failed” . result contains this value if payment fails from the bank side.
“error_noretry”	It is a string constant and its value is “error_noretry”. This result can be considered as a failed payment.
“retry_fail_error”	It is a string constant and its value is “retry_fail_error”. This result can be considered as a failed payment.

2. The below is detail response of payment

2.1 Success response json.


```
{
        "name_on_card": "Rizwan",
        "bank_ref_num": "214057035798",
        "udf3": "",
        "hash":
"5cd269f97088bf88ee9aef20864f6c86b738b4450851a7e518b21fc4bb2253ee153e6bd29ddc06a
adbc6a846d08b1452c488aed20f31c05ff6442894b994366a",
        "firstname": "Sagar",
        "net_amount_debit": "2.0",
        "payment_source": "Easebuzz",
        "surl": "http://localhost/paywitheasebuzz-php-lib-master/response.php",
        "error_Message": "Successful Transaction",
        "issuing_bank": "NA",
        "cardCategory": "NA",
        "phone": "8805596828",
        "easepayid": "E220520TZ7VGC2",
        "cardnum": "541919XXXXXX1220",
        "key": "D1K8SLB2XW",
        "udf8": "",
        "unmappedstatus": "NA",
        "PG_TYPE": "NA",
        "addedon": "2022-05-20 07:15:07",
        "cash_back_percentage": "50.0",
        "status": "success",
        "card_type": "Debit Card",
        "merchant_logo": "NA",
        "udf6": "",
        "udf10": "",
        "upi_va": "NA",
        "txnid": "EBZTXN0000357",
        "productinfo": "Test",
        "bank_name": "NA",
        "furl": "http://localhost/paywitheasebuzz-php-lib-master/response.php",
        "udf1": "",
        "amount": "2.0",
        "udf2": "",
        "udf5": "",
        "mode": "DC",
        "udf7": "",
        "error": "Successful Transaction",
        "udf9": "",
        "bankcode": "NA",
        "deduction_percentage": "0.0",
        "email": "sagar.sawant@easebuzz.in",
        "udf4": ""
}
```

2.2 Failure response json.


```
{
        "name_on_card": "Rizwan",
        "bank_ref_num": "",
        "udf3": "",
        "hash":
"bbf14e1a746abc6fba3267020d9d743c7952ad6c0927bacedac20b16eda0971ff734f98b15641805
5668a262f87b9e32b07476bca379992446359fd5bf1ce9d3",
        "firstname": "Sagar",
        "net_amount_debit": "20.0",
        "payment_source": "Easebuzz",
        "surl": "http://localhost/paywitheasebuzz-php-lib-master/response.php",
        "error_Message": "!ERROR!-GV00004-PARes status not sucessful.",
        "issuing_bank": "NA",
        "cardCategory": "NA",
        "phone": "8805596828",
        "easepayid": "E220520SIWNB81",
        "cardnum": "541919XXXXXX1220",
        "key": "D1K8SLB2XW",
        "udf8": "",
        "unmappedstatus": "NA",
        "PG_TYPE": "NA",
        "addedon": "2022-05-20 06:35:40",
        "cash_back_percentage": "50.0",
        "status": "failure",
        "card_type": "Debit Card",
        "merchant_logo": "NA",
        "udf6": "",
        "udf10": "",
        "upi_va": "NA",
        "txnid": "EBZTXN0000351",
        "productinfo": "Test",
        "bank_name": "NA",
        "furl": "http://localhost/paywitheasebuzz-php-lib-master/response.php",
        "udf1": "",
        "amount": "20.0",
        "udf2": "",
        "udf5": "",
        "mode": "DC",
        "udf7": "",
        "error": "!ERROR!-GV00004-PARes status not sucessful.",
        "udf9": "",
        "bankcode": "NA",
        "deduction_percentage": "0.0",
        "email": "sagar.sawant@easebuzz.in",
        "udf4": ""
 }
```