# Easebuzz React Native Integration

Download React Native SDK from github using the link below, extract it and put it in the relevant directory. https://github.com/easebuzz/paywitheasebuzz-react-native-lib

## Installing SDK:

1. Run the below commands to install sdk into your project.

   ```
   1.1 npm install $(npm pack <Path of React Native SDK>/react-native-easebuzz-kit | tail -1)
       Example : npm install $(npm pack HomeDirectory/SDKfolder/react-native-easebuzz-kit | tail -1)
   ```

   *For React Native 0.59 and lower*

   ```
   1.2. react-native link react-native-easebuzz-kit
   ```

## Android Setup:

1. Copy **peb-lib.aar** file into android/app/libs/ folder of your react native application(If libs folder is not there, Please create it manually).

2. **Build.gradle(app) modifications.**

   Add this line to build.gradle (app)

   ```
   defaultConfig {
               multiDexEnabled true
         }
   ```

   Add the following lines to packagingOptions,

   ```
       exclude 'META-INF/DEPENDENCIES'
       exclude 'META-INF/NOTICE'
       exclude 'META-INF/LICENSE'
        exclude 'META-INF/LICENSE.txt'
       exclude 'META-INF/NOTICE.txt'
   ```

   Add the following line to dexOptions.

   ```
       javaMaxHeapSize "4g"
   ```

   Add repositories section as follows.

   ```
   repositories {
       flatDir {
           dirs 'libs'
           }
       }
   ```

Add the following dependencies

```
implementation (name: 'peb-lib', ext: 'aar')
implementation 'com.android.support:appcompat-v7:28.0.0'
implementation 'com.android.support:design:28.0.0'
implementation 'com.android.support:multidex:1.0.1'
implementation 'com.squareup.okhttp:okhttp:2.4.0'
implementation 'com.squareup.okhttp:okhttp-urlconnection:2.2.0'
implementation 'com.squareup.retrofit2:retrofit:2.3.0'
implementation 'com.squareup.retrofit2:converter-gson:2.3.0'
```

**3. Add Below code in your launcher activity of app**

```
import android.content.Intent;
      @Override
      public void onNewIntent(Intent intent) {
         this.setIntent(intent);
      }
```

**4. Manifest Change.**

Change <Application> tag of androidManifest.xml file as below in your projects android folder.

```
android:allowBackup="true"
```

Your Android setup is done.

## iOS Setup:

1. Copy easebuzz.xcframework of your application in embedded binaries.
2. Press + and add framework using ‘Add other’ button.
3. Browse framework: file from your folder and select ‘copy items if needed’.
4. Set Always embed & sign from frameworks and Libraries  and swift standard libraries to YES from project build settings

5. ***For React Native 0.60+***

*1. npm install $(npm pack <Path of React Native SDK>/react-native-easebuzz-kit | tail -1)*
*Example : npm install $(npm pack HomeDirectory/SDKfolder/react-native-easebuzz-kit | tail -1)*
*(ignore if already installed)*

*2. cd ios && open podfile # Change the platform from iOS 9.0 to 10.0*
*pod install && cd .. # CocoaPods on iOS needs this extra step*

***For React Native 0.59 and lower***

*1. npm install $(npm pack <Path of React Native SDK>/react-native-easebuzz-kitt | tail -1)*
*Example : npm install $(npm pack HomeDirectory/SDKfolder/react-native-easebuzz-kit | tail -1)*
*(ignore if already installed)*
*2. Link the SDK with React Native Project using Xcode.*
*react-native link react-native-easebuzz-kit*

6. Remove unused architectures, refer below link.
https://docs.easebuzz.in/mobile-integration-ios/remove-architectures

7. The existing plugin does not work on simulator. The iOS framework is shipped with simulator architectures ,so you have to replace simulator Easebuzz.framework (from iOS-Frameworks/ios-simulator folder) with device Easebuzz.framework(from iOS-Frameworks/ios-device folder). Reinstall the plugin after changing the framework. Please find the below folder path for replacing the framework in to plugin folder =>
**Replace path: react-native-easebuzz-kit -> ios -> Easebuzz.framework**

Note: If you changed the simulator framework with existing,make sure at the time of uploading the app to APPStore, use the existing framework - iOS-Frameworks/ios-device/Easebuzz.Framework.

Your iOS setup is done.

**Initiate Payment:** Generate access key using the Initiate Payment API at your backend.

**(It is mandatory to integrate the Initiate Payment API at Backend only)**

Initiate Payment API Doc : https://docs.easebuzz.in/api/initiate-payment

## React Native Integration Code (Java Script):

1.1 Write below JavaScript Code to Start Payment using Easebuzz Payment Gateway

1.2 Import following components

```
    import {Platform, Button, NativeModules,NativeEventEmitter} from 'react-native';
    import EasebuzzCheckout from 'react-native-easebuzz-kit';
```

1.3  Call Payment Method

Call EasebuzzCheckout.open method with the payment request parameters as options. This method returns a JS Promise where **then** part corresponds to a successful payment response or failure response and the catch part corresponds to any sdk failure i.e event failed etc.

```
 const callPaymentGateway  = () => {
var options = {
   access_key: "Access key generated by the Initiate Payment API",
   pay_mode: "This can be “test” or “production"
 }

  EasebuzzCheckout.open(options).then((data) => {
             //handle the payment success & failed response here
               console.log("Payment Response:")
                console.log(data);
          }).catch((error) => {
                  //handle sdk failure issue here
                   console.log("SDK Error:")
                   console.log(error);
           });
         }
```

Note:

1. *For More description of request parameter  and optional parameter refer below link.* https://docs.easebuzz.in/api/initiate-payment
2. *For More description of request and Response refer below link. https://docs.easebuzz.in/mobile-integration-react-native/handle-response*