



**INTEGRATION GUIDE  
FOR  
MOBILE APPLICATION DEEPLINK SETUP  
iOS & ANDROID**

(Version 1.2)

Last updated on 07th October 2024

RAZER MERCHANT SERVICES		
  	<p>J-39-1, Block J, Persiaran Multimedia, i-City, 40000 Shah Alam, Selangor, Malaysia.</p> <p>+(603) - 5521 8438</p> <p>support@Fiuu.com</p> <p>Fiuu.com</p>	
<b>Social Networks</b> <a href="https://twitter.com/FiuuPayment">https://twitter.com/FiuuPayment</a> <a href="https://facebook.com/FiuuPayment">https://facebook.com/FiuuPayment</a> <a href="https://youtube.com/@FiuuPayment">https://youtube.com/@FiuuPayment</a> <a href="https://instagram.com/FiuuPayment">https://instagram.com/FiuuPayment</a> <a href="https://linkedin.com/company/FiuuPayment">https://linkedin.com/company/FiuuPayment</a>		
<b>Developer Platforms</b> <a href="https://t.me/FiuuDeveloperForum">https://t.me/FiuuDeveloperForum</a> (for tech support) <a href="https://github.com/FiuuPayment">https://github.com/FiuuPayment</a> Mobile XDK, seamless and in-page checkout, and many shopping carts payment plugin/module/addon/extension are available		
		

# ChangeLogs

Date	Version	Author	Description
2024/08/15	v1.0	Ashraf Amin	Initial document
2024/09/11	v1.1	Nik Faris	Update Deeplink Handling
2024/10/07	v1.2	Nik Faris	Update sample code

# Table of Contents

## [INTRODUCTION](#)

### [1 - Add deep link scheme](#)

#### [1.1 - ANDROID](#)

#### [1.2 - Deeplink Handling](#)

#### [1.3 - iOS](#)

### [2 - Register app scheme](#)

### [3 - Run and test](#)

# INTRODUCTION

This document showing step by step to enable Deep Link Redirection for Fiuu Mobile XDK

## 1 - Add deep link scheme

For this guide, application scheme is “**yourappscheme**”

### 1.1 - ANDROID

1- Create new activity (DeeplinkActivity)

Add intent-filter as below in **AndroidManifest.xml**

\* scheme must be **letter only & all lowercase** + unique identifier that represent your app

```
<activity
  android:name=".DeeplinkActivity">
  ...

  <intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    <data android:scheme="yourappscheme" />
  </intent-filter>
  ...
</activity>
```

## 1.2 - Deeplink Handling

Handle the scheme in activity using intent

Kotlin

```
class DeeplinkActivity : ReactActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val intent = intent
        if (intent != null) {
            Log.d("TAG", "Intent received: ${intent.data}")
            if (isTaskRoot) {
                // Handle root activity logic here
                intent.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT or
Intent.FLAG_ACTIVITY_CLEAR_TASK)
                intent.setClass(this, MainActivity::class.java)
                startActivity(intent)
                finish()
            } else {
                finish()
            }
        } else {
            Log.d("TAG", "Intent is null")
        }
    }
}
```

**\*\* this example for ReactNative but the same idea for any other hybrid platform. Need to handle the intent returned**

Java

```
public class DeepLinkActivity extends ReactActivity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

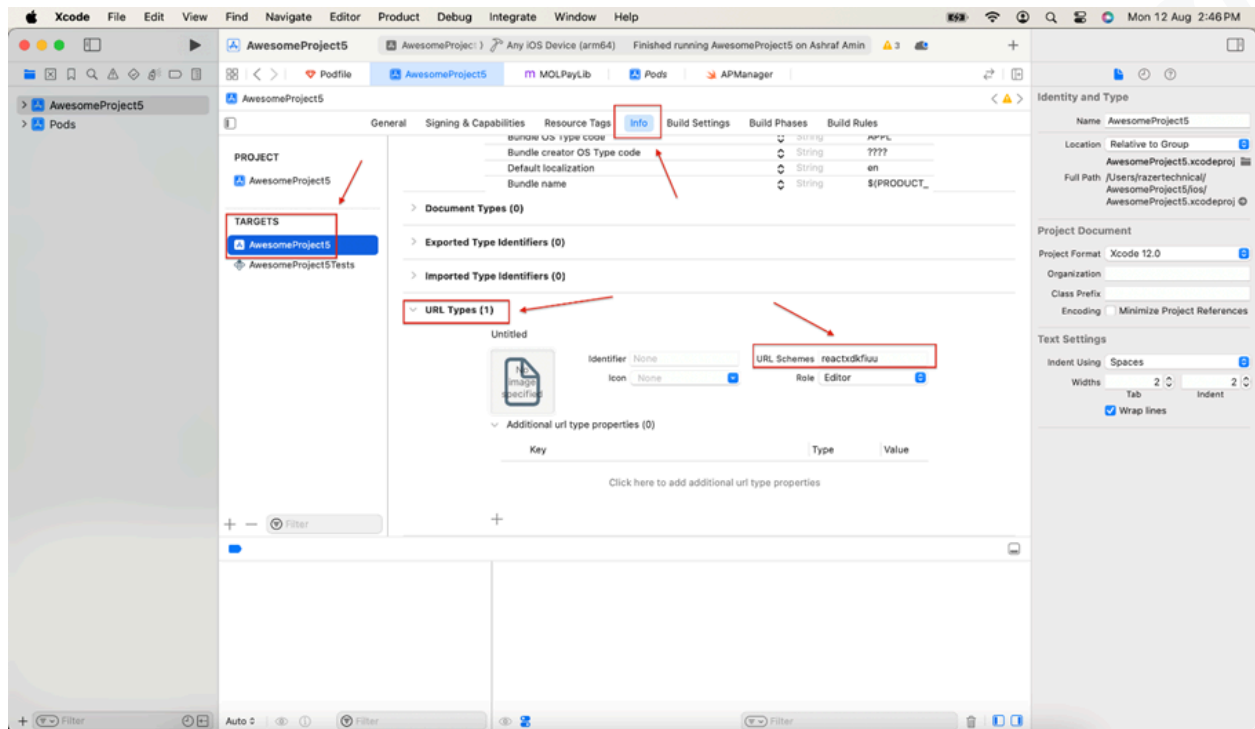
        Intent intent = getIntent();
        if (intent != null) {
            if (isTaskRoot()) {
                intent.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT |
                Intent.FLAG_ACTIVITY_CLEAR_TASK);
                intent.setClass(this, MainActivity.class);
                startActivity(intent);
                finish();
            } else {
                finish();
            }
        } else {
            Log.d("TAG", "intent is null");
        }
    }
}
```

***\*\* this example for ReactNative but the same idea for any other hybrid platform. Need to handle the intent returned***

# 1.3 - iOS

Add Xcode **URL Schemes** in TARGETS > Info > URL Types

\* URL Schemes must be **letter only & all lowercase** + unique identifier that represent your app



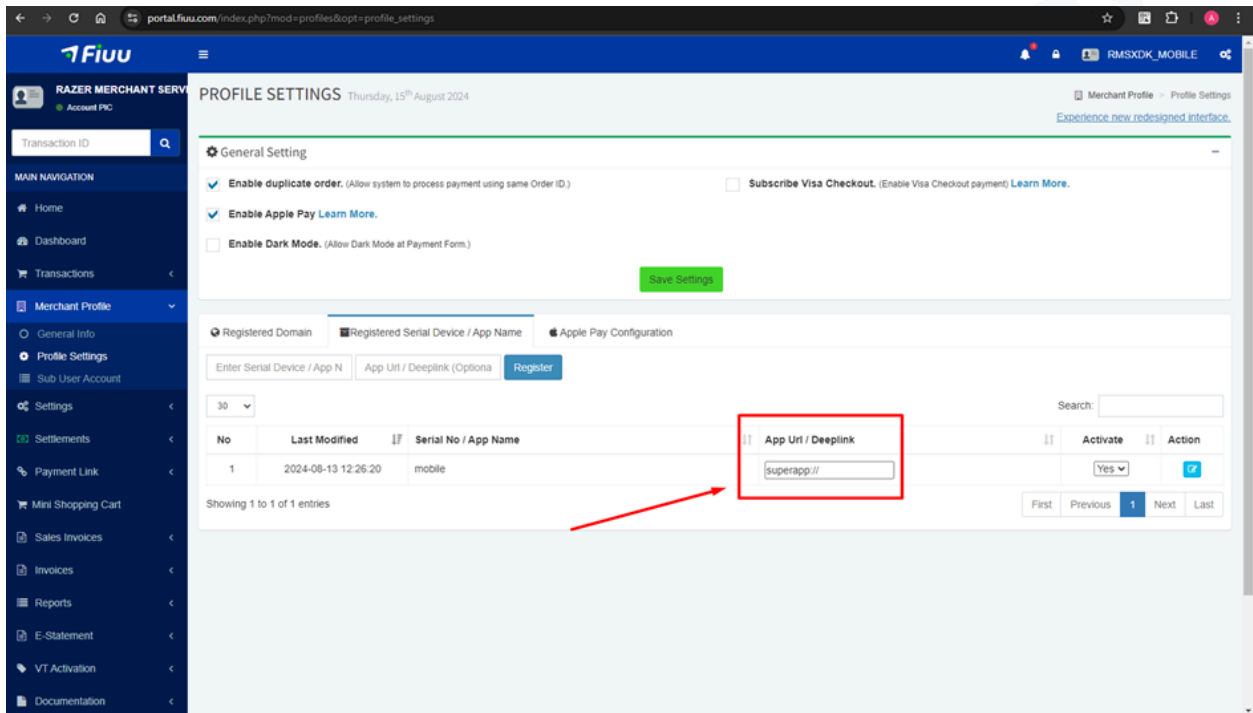


## 2 - Register app scheme

1) Login to merchant portal

([https://portal.fiuu.com/index.php?mod=profiles&opt=profile\\_settings](https://portal.fiuu.com/index.php?mod=profiles&opt=profile_settings)) & go to Merchant Profile > Profile Settings > Registered a Serial Device / App Name.

2) Add your scheme ended with "://" e.g. "**yourappscheme://**" in App Url / Deeplink. If your app set host include it at the end e.g. "**yourappscheme://yourapphost**". Serial No / App Name is your mp\_app\_name parameter value.



The screenshot shows the 'PROFILE SETTINGS' page for a merchant. The 'Registered Serial Device / App Name' tab is active. Below the tabs, there is a table with one entry. The 'App Url / Deeplink' column for this entry contains the text 'superapp://', which is highlighted by a red box. A red arrow points to this box. The table has columns for 'No', 'Last Modified', 'Serial No / App Name', 'App Url / Deeplink', 'Activate', and 'Action'. The 'Activate' column has a dropdown menu set to 'Yes'. The 'Action' column has a blue icon. The table is titled 'Showing 1 to 1 of 1 entries'.

No	Last Modified	Serial No / App Name	App Url / Deeplink	Activate	Action
1	2024-08-13 12:26:20	mobile	superapp://	Yes	

## 3 - Run and test

1) **Clean & rebuild** your app.

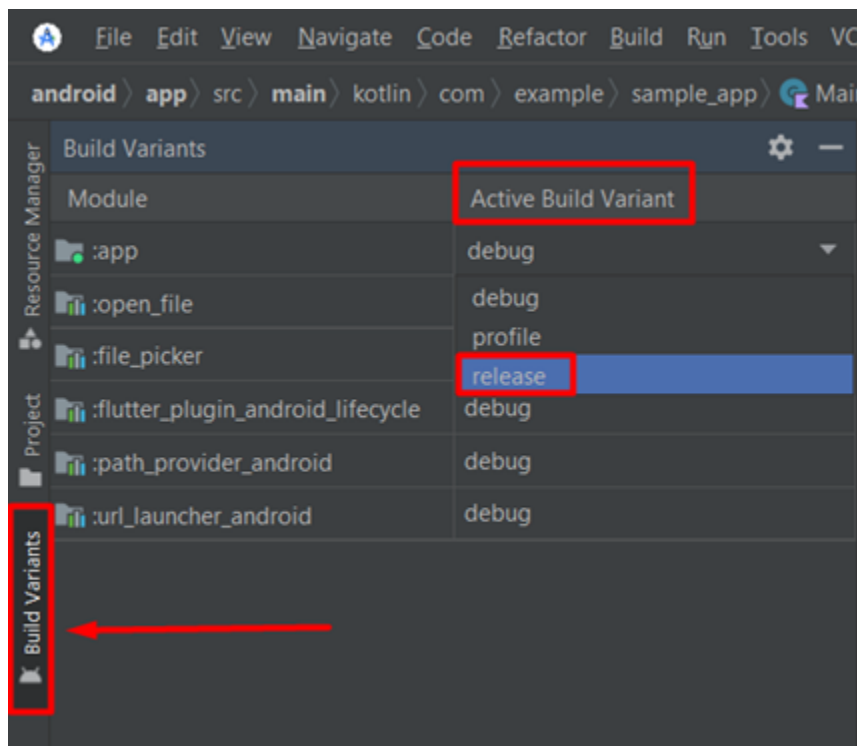
2) **Run your app using release build** version once you have registered your App Url.

3.1) For **Android** can run via Terminal command:

e.g. **React Native** : `npx react-native run-android release`

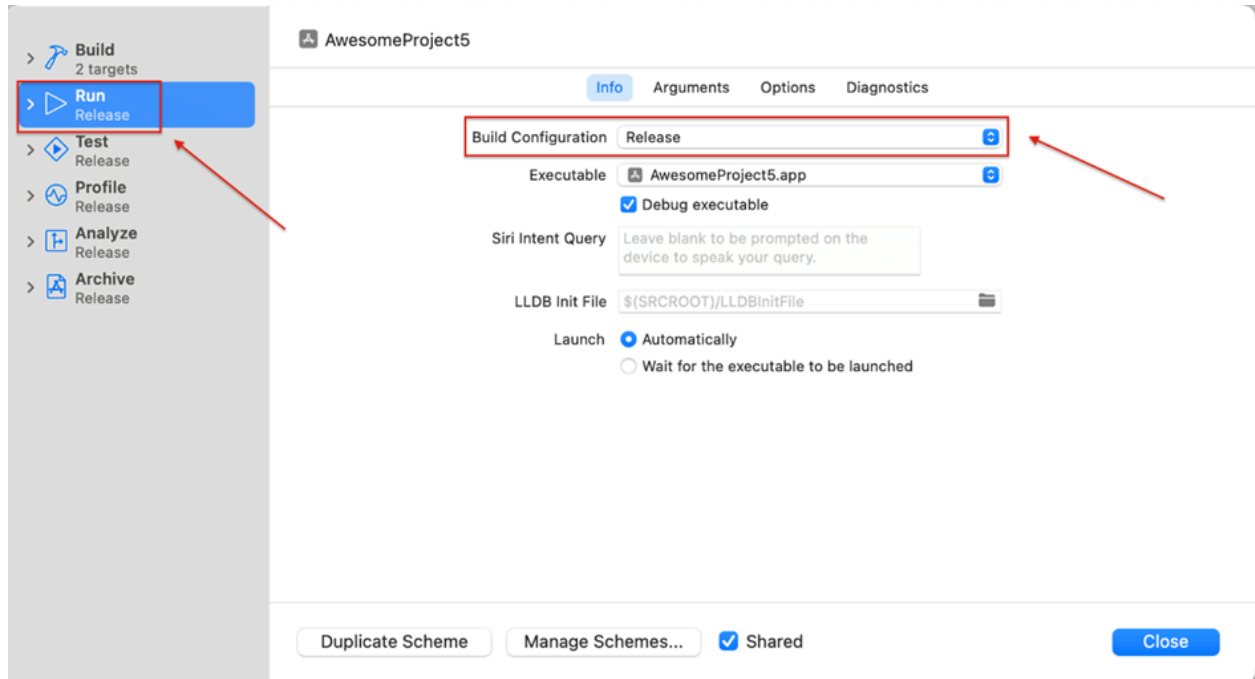
e.g. **Flutter** : `flutter run --release`

Or change **Active Build Variant to release** before run



Or installing release build **APK**

3.2) For **iOS** can set Run – **Build Configuration as Release** in Product > Scheme > Edit Scheme



4) **You're done.** Test your app.