

# Introduction

This document provides step-by-step instructions to integrate the `pspwDmt.aar` library into an Android project.

## 1. Prerequisites

2. **Android Studio:** Ensure you have the latest version of Android Studio installed.
3. **Minimum SDK Version:** The library supports Android **API level 21 to 34**
4. **Partner Credentials:** You should have the following:
  - a. `partnerId`
  - b. `partnerApiKey`
  - c. `merchantCode`

## 2. Adding the `.aar` File to Your Project

1. **Copy the `.aar` file:** Place the `pspwDmt.aar` & `pw_ekyc.aar` file in the `libs` folder of your Android project. If the `libs` folder doesn't exist, create one in the root of your module.

2. **Update `build.gradle`:** Add the following to your module's `build.gradle` file:

```
repositories {
```

```
    flatDir {
```

```
        dirs 'libs'
```

```
    }
```

```
}
```

```
dependencies {
```

```
    implementation(name: 'pspwDmt', ext: 'aar')
```

```
    implementation(name: 'pw_ekyc', ext: 'aar')
```

```
}
```

3. **Sync the project:** Click on **Sync Now** to ensure the library is correctly added to your project.

## 3. Integration Steps

### 3.1 Initialize the Intent

Use the following code to launch the **DMT Host Activity**:

```
Intent intent = new Intent(activity,  
DmtHostActivity.class);  
intent.putExtra("partnerId", partnerId);  
intent.putExtra("partnerApiKey", partnerApiKey);  
intent.putExtra("merchantCode", merchantCode);  
  
// Launch the activity  
dmtPwLauncher.launch(intent);
```

### 3.2 Handle Activity Results

```
ActivityResultLauncher<Intent> dmtPwLauncher =  
registerForActivityResult(  
    new  
    ActivityResultContracts.StartActivityForResult(),  
    result -> {  
        if (result.getResultCode() ==  
Activity.RESULT_OK) {  
            Intent data = result.getData();  
            String response =  
data.getStringExtra("response");
```

```

        Log.d("PW_RESPONSE", ": " + response);
        // Handle success response
    } else if (result.getResultCode() ==
Activity.RESULT_CANCELED) {
        new AlertDialog.Builder(context)
            .setMessage("Transaction Aborted")
            .setPositiveButton("OK", null)
            .show();
    }
}
);

```

*Response Handling:*

- **RESULT\_OK**
  - **Key:** response
  - **Type:** String
  - **response:** JSON response or relevant data about the transaction.
- **RESULT\_CANCELED**
  - User aborted the transaction.

## 4. Permissions

Ensure your `AndroidManifest.xml` contains any required permissions. For example:

```

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

```

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

## **5. ProGuard Rules**

If your app uses ProGuard, add the following rules to avoid issues during obfuscation:

```
-keep class it.services.pspwdmt.** { *; }  
-dontwarn it.services.pspwdmt.**
```

## 6. Sample Code

Here's an example of a complete integration:

```
public class MainActivity extends AppCompatActivity {

    private ActivityResultLauncher<Intent> dmtPwLauncher;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize ActivityResultLauncher
        dmtPwLauncher = registerForActivityResult(
            new ActivityResultContracts.StartActivityForResult(),
            result -> {
                if (result.getResultCode() == Activity.RESULT_OK) {
                    Intent data = result.getData();
                    String response = data.getStringExtra("response");
                    Log.d("PW_RESPONSE", ": " + response);
                } else if (result.getResultCode() == Activity.RESULT_CANCELED) {
                    new AlertDialog.Builder(this)
                        .setMessage("Transaction Aborted")
                        .setPositiveButton("OK", null)
                        .show();
                }
            }
        );

        // Launch the intent
        findViewById(R.id.launchButton).setOnClickListener(view -> {
            Intent intent = new Intent(this, DmtHostActivity.class);
            intent.putExtra("partnerId", "your_partner_id");
            intent.putExtra("partnerApiKey", "your_api_key");
```

```
        intent.putExtra("merchantCode", "your_merchant_code");

        dmtPwLauncher.launch(intent);
    });
}
}
```