

[1 Introduction](#)[2 Setup](#)[3 Create a Generic pass class](#)[4 Open the project in Android Studio](#)[5 Create the Add to Google Wallet button](#)[6 Check if the Google Wallet API is available](#)[7 Create a Generic pass object](#)[8 Add the pass to Google Wallet](#)[9 Handle the savePassesJwt result](#)[10 Congratulations](#)

6. Check if the Google Wallet API is available

If a user opens your app on a device that does not support the Google Wallet API, it may create a negative experience when they try to add the pass. If the user's device does not support the Google Wallet API, hiding the **Add to Google Wallet** button avoids potential confusion. There are various reasons why the API may be unavailable, such as the Android or Google Play Services versions being out of date or Google Wallet being unavailable in the user's country.

In this step, you will add logic to your app that checks if the Google Wallet API is available on the device. If so, the button will be rendered in the activity. Otherwise, the button will be hidden.

Consider placing domain-specific business logic appropriately if you are using other design patterns. For example, if you are using the Model-View-ViewModel (MVVM) pattern, place UI-related business logic in your Activity or Fragment (e.g.: UI elements and activity results), and operational logic in your ViewModel (e.g.: client instantiation and network call triggers).

1. Open the `CheckoutActivity.kt` file in `app/src/main/java/com/google/android/gms/samples/wallet/activity/`
2. Create a class property for the `PayClient` instance

```
// TODO: Create a client to interact with the Google Wallet API
private lateinit var walletClient: PayClient
```

3. Instantiate the `PayClient` property in the `onCreate` method

[Zurücksetzen](#) : Instantiate the client

```
walletClient = Pay.getClient(this)
```

[Weiter](#)