

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

8. Add the pass to Google Wallet

Now that you have verified that the Google Wallet API is available and have created a signed JWT, you can prompt the user to add the pass to their wallet. In this step, you will add a listener to the **Add to Google Wallet** button that uses the Google Wallet API to save the pass in the user's wallet.

1. Open the `app/src/main/CheckoutActivity.kt` file
2. Replace the value of `token` with the JWT you created previously

```
// TODO: Save the JWT from the backend "response"
private val token = "TOKEN"
```

3. Create a class property to store the request code

```
// TODO: Add a request code for the save operation
private val addToGoogleWalletRequestCode = 1000
```

4. Set a listener for the **Add to Google Wallet** button

```
// TODO: Set an on-click listener on the "Add to Google Wallet" button
addToGoogleWalletButton = layout.addToGoogleWalletButton.

addToGoogleWalletButton.setOnClickListener {
    walletClient.savePassesJwt(token, this, addToGoogleWalletRequestCode)
}
```

Zurücksetzen

When the **Add to Google Wallet** button is clicked, the `walletClient.savePassesJwt` method is called. This

method prompts the user to add the new pass object to their Google Wallet.

Weiter