Payeezy iOS SDK Integration Guide

April 2015

First Data.
beyond the transaction

Payeezy
DEVELOPERS

Contents

Quickly integrate your applications with Payeezy Android SDK

If you want to enable secure and convenient payments to your payment applications, this guide will get you up and running quickly. Payeezy handles all the heavy lifting of the complex tokenization and protects your customers' transactions. It is simple to create a developer test account and apply for a merchant account through our developer portal.

Payeezy℠, from First Data, is part of the Payeezy e-commerce solution which includes Clover™, Insightics℠, Perka™ and TransArmor®. The Small Business Solution Suite empowers SMBs to expand their horizons and easily grow their business online or via mobile by reaching new customers no matter where they are.

This documentation refers to the Android integration method included within the overall Payeezy eCommerce Solution. Henceforth, all references to Payeezy are in relation to Payeezy APIs.

Minimum technical requirements

- The payeezy android mobile SDK requires android OS 4.0 ice cream sandwich or higher (API level 14 or higher).
- A physical device or an emulator to use for developing and testing because Google Play services can only be installed on an emulator with an AVD that runs Google APIs platform based on Android 4.0 or higher.
- The latest version of the Android SDK, including the SDK Tools component. The SDK is available from the **Android SDK Manager**.
- Your project to compile against Android 4.0 (Ice Cream Sandwich) or higher.
- The latest version of Android Studio.
- Android SDK requires Java JRE (JDK for development) as per android guidelines.

For quick start on android sample application please refer to the link https://developers.google.com/+/quickstart/android.

Payeezy Android initialization

Build payment application to make purchases directly in your ecommerce application. This provides simple and easy In-App payment for ecommerce payment solutions.

Payeezy Prerequisites

Please go through prerequisites, to get an overview of our developer portal. Link for prerequisites https://github.com/payeezy/get_started_with_payeezy/blob/master/get_started_with_payeezy042015.pdf .
Build android app to make purchases directly in your app.

For frequently asked questions please refer to https://developer.payeezy.com/faq-page.

Steps to integrate with Payeezy SDK to make payments (Example: Credit Card Payments)

Step1: Collect credit card information

A. For sandbox region please refer to the test cards in the developer portal https://developer.payeezy.com/payeezy-api-reference/apis/credit-card-payments

Step2: Use APIkey, secret and merchant token to execute Authorize and Purchase.
Step3: Sending token to complete Authorize and Purchase.

Getting started

Using GitHub

Clone Payeezy SDK using with HTTPS or Subversion

Using clone command: git clone https://github.com/payeezy/payeezy_android.git

Or simply download zip file: https://github.com/payeezy/payeezy_android/archive/master.zip

To know more about GitHub, click http://github.com

Import the downloaded android SDK project into eclipse with android plugin.

Define merchant Token, APIkey and Secret:

Open the FirstAPIClientV2Helper.java and to update the APIkey, secret and token.
Open

```
restTemplate.setRequestFactory( new org.springframework.http.client.HttpComponentsClientHttp
this.setUrl("https://api-cert.payeezy.com/v1");
this.setAppId("y6pWAJNyJyjGv66IsVuWnklkKUPFbb0a");
this.setSecuredSecret("86fbae7030253af3cd15faef2a1f4b67353e41fb6799f576b5093ae52901e6f7");
this.setToken("fdoa-a480ce8951daa73262734cf102641994c1e55e7cdf4c02b6");

}
```

Payeezy Integration

Payeezy support integration with android, Java, php, python, ruby, nodeJS and curl

Tools & RESOURCES

Download and build

- **GitHub** – For Payeezy source code downloads – visit https://github.com/
- **Eclipse** – For client library integration and development – for more information visit https://eclipse.org/
- **IntelliJ** – For client library integration and development – for more information please visit https://www.jetbrains.com/idea/
- **JDK** 1.6 and above - For compiling java code – for more information please visit http://www.oracle.com/technetwork/java/javase/downloads/index.html
- **Text editor** - if you don't have one already, you may want one for editing properties file. A few options: Atom, Sublime Text, Textmate, Brackets.
- **Android Studio:** Android studio can be downloaded from the following location. I have all the information

about android studio. http://developer.android.com/tools/studio/index.html

- **Android SDK** – For android SDK visit http://developer.android.com/sdk/index.html . This has the android SDK tools and plugins and the android studio for development.

- **Android eclipse plugin** – The android SDK eclipse can be downloaded from the site http://developer.android.com/sdk/installing/installing-adt.html .

Package Managers

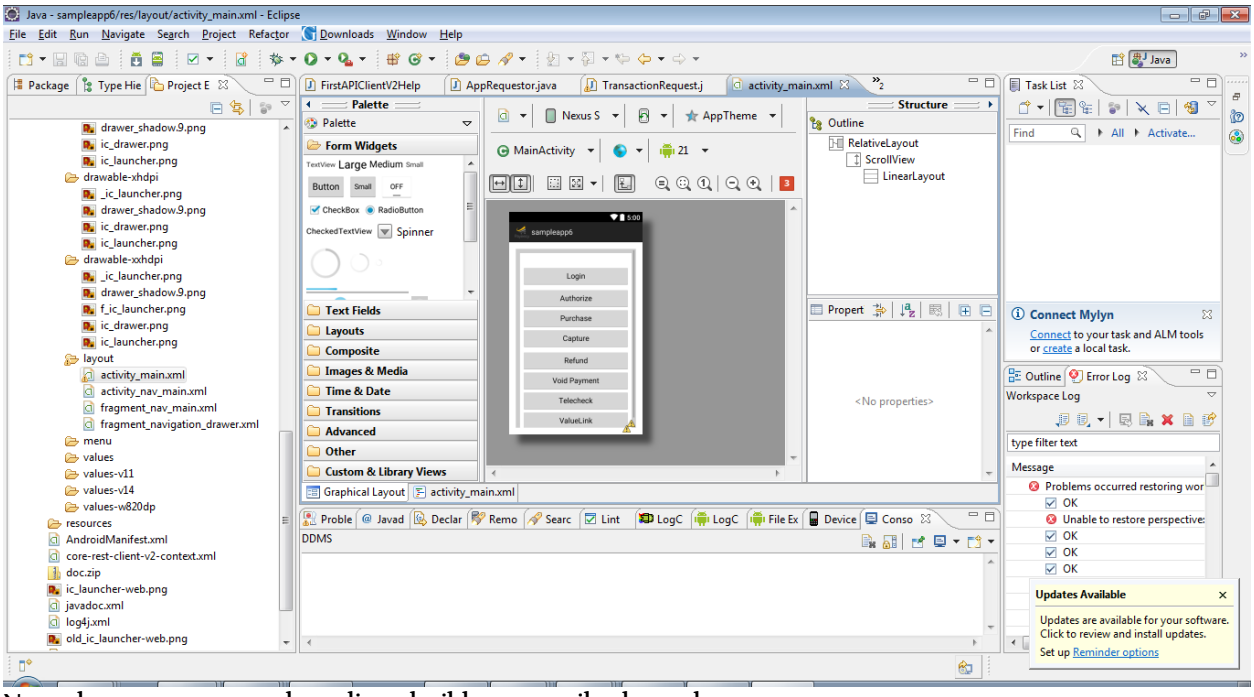
- Gradle – For build, standard gradle build can be used. You need to add SDK/platform/tools folder in the path. We are using standard eclipse build for our purpose.
- **Maven** – For build and test – for more information please visit http://maven.apache.org/

Download

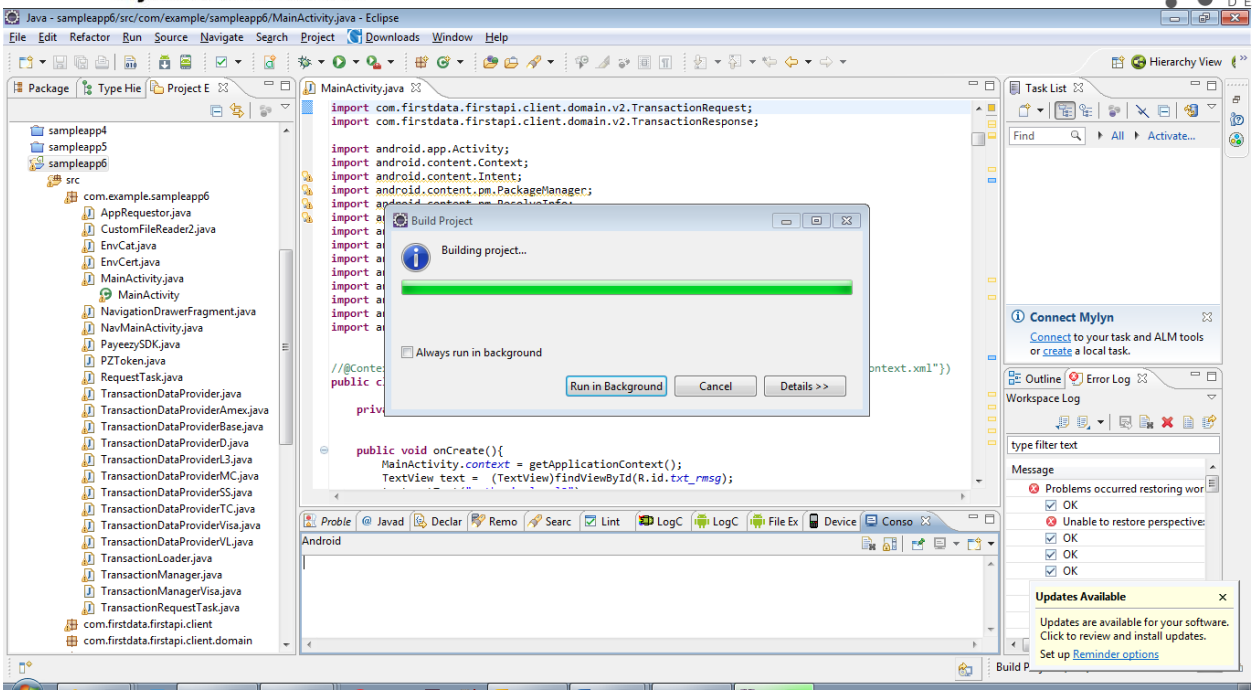
Download sample code from GitHub https://github.com/payeezy/payeezy/tree/master/java

Build/install

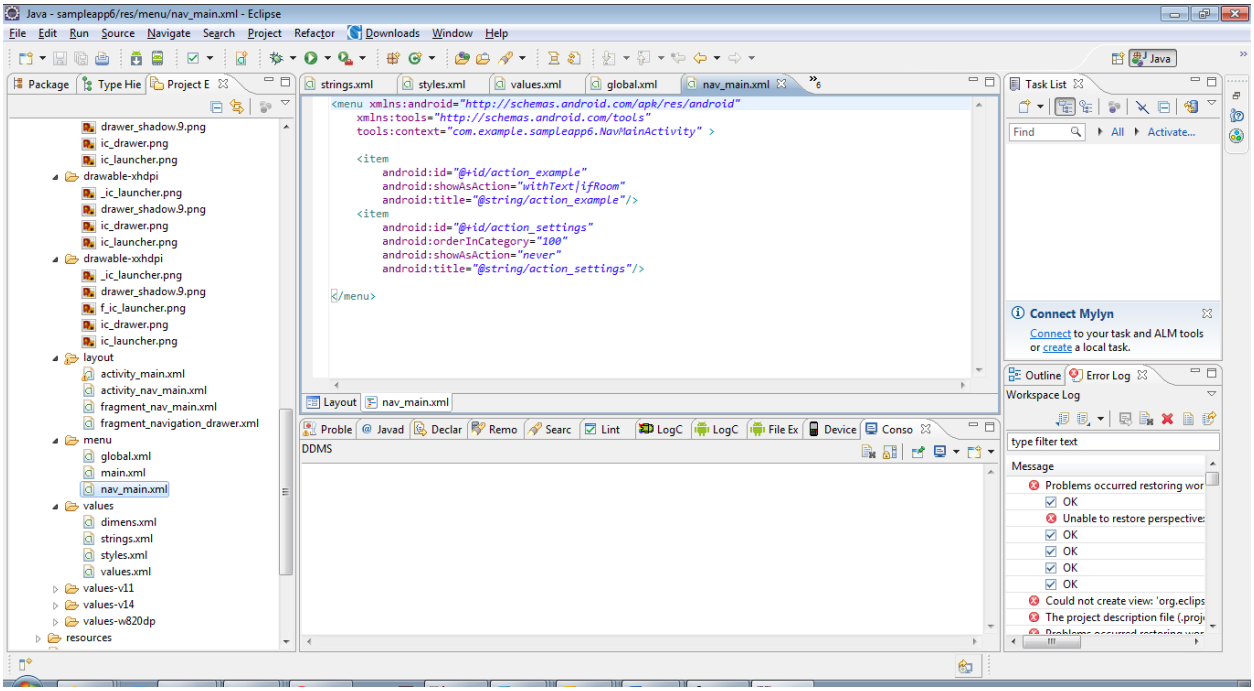
The build can be done using the eclipse build. The project code is imported into eclipse. Following are the pictures of the code in eclipse. The code can be built from eclipse with Android plugin.
You can also import code into eclipse.



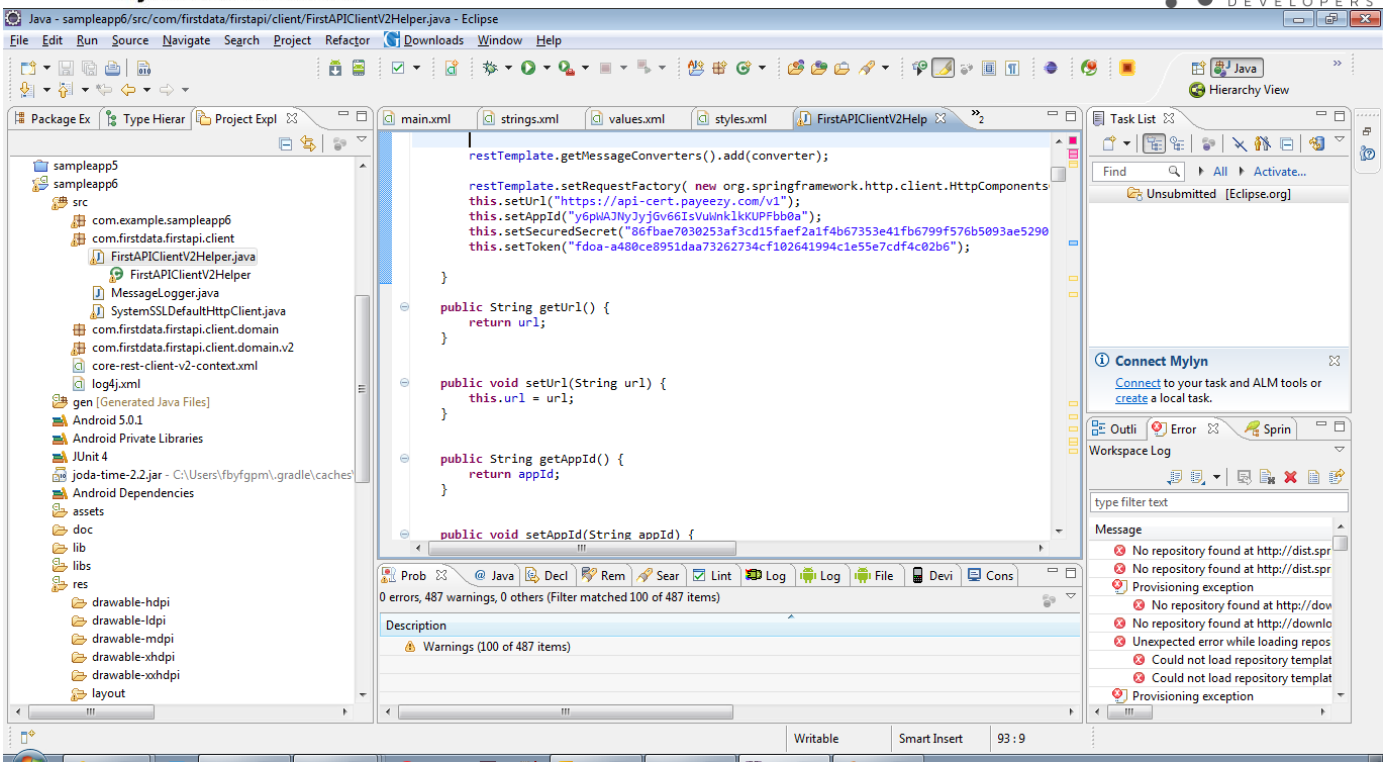
Note that you can run the eclipse build to compile the code.



Following screen shows the menu options



Following screen shows specifying the APIkey, secret and token specification.



Customize

Update FirstAPIClientV2Helper.java file.

Please go to $PAYEEZY /sampleapp6/src/com/firstdata/firstapi/client/FirstAPIClientV2Helper.java and in the constructor update with your APIkey, Secret and token. Further these values can be modified in the individual calls to the FirstAPIClientV2Helper class.

Where

For example: $PAYEEZY = path to SDK folder

Note that

securedSecret = API Secret

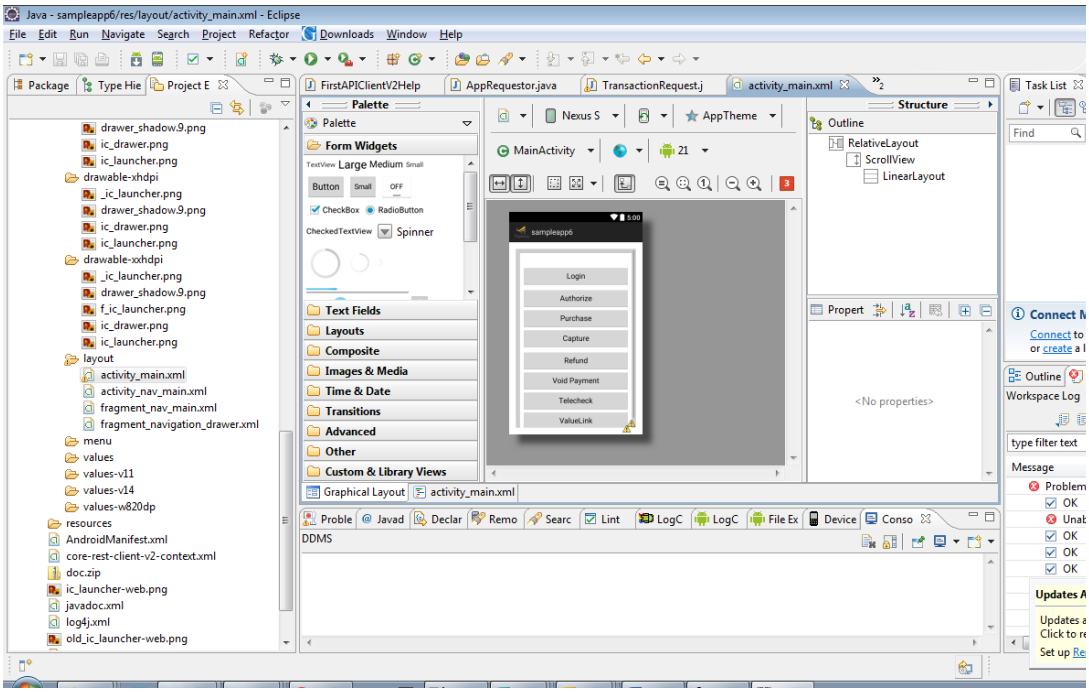
appid = API Key

token= Merchant token

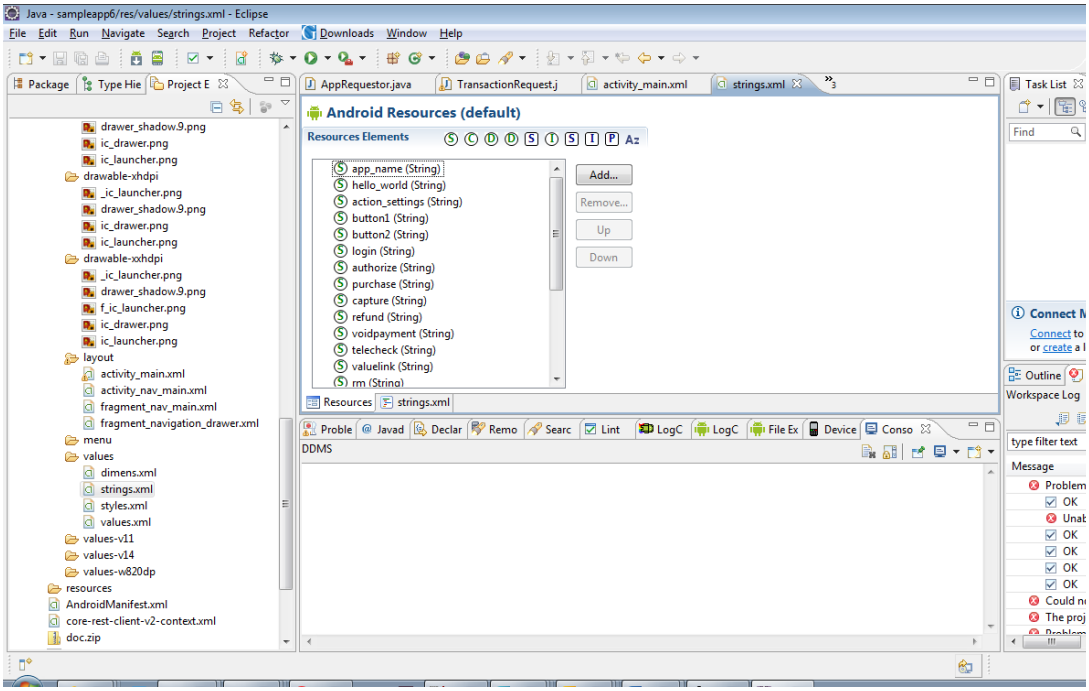
```
restTemplate.setRequestFactory( new org.springframework.http.client.HttpComponentsClientHttp
this.setUrl("https://api-cert.payeezy.com/v1");
this.setAppId("y6pWAJNyJyjGv66IsVuWnklkKUPFbb0a");
this.setSecuredSecret("86fbae7030253af3cd15faef2a1f4b67353e41fb6799f576b5093ae52901e6f7");
this.setToken("fdoa-a480ce8951daa73262734cf102641994c1e55e7cdf4c02b6");

}
```

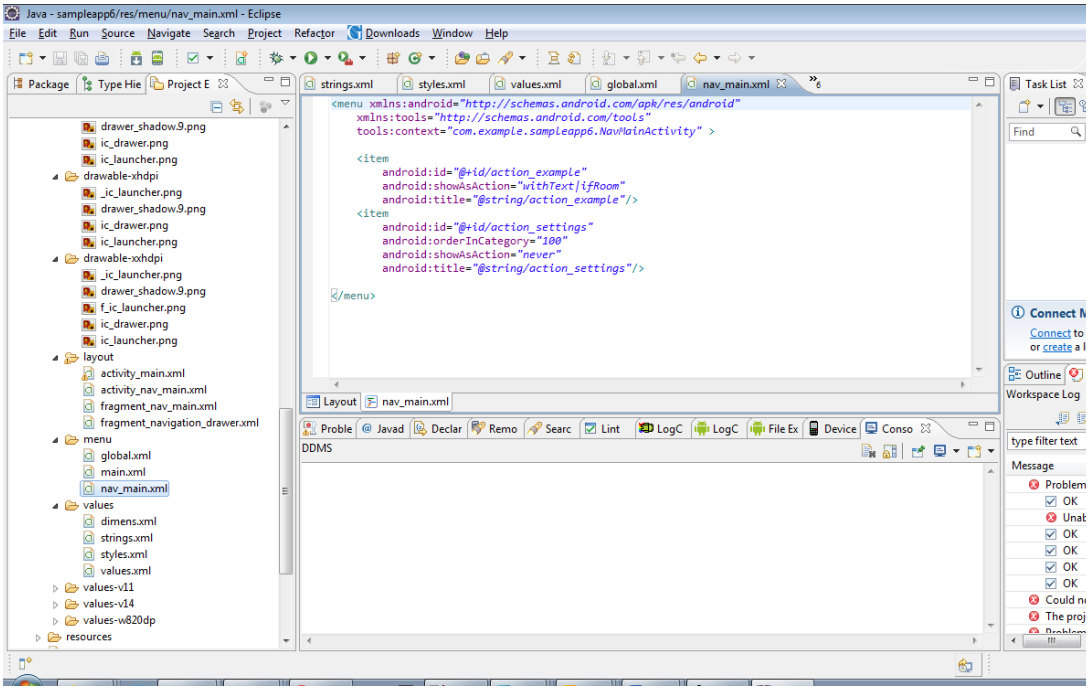
Download the latest android SDK from the GitHub https://www.github.com/payeezy/payeezy_android/payeezy_android . Open/Import the source code with eclipse. Following picture shows how the source code looks like.



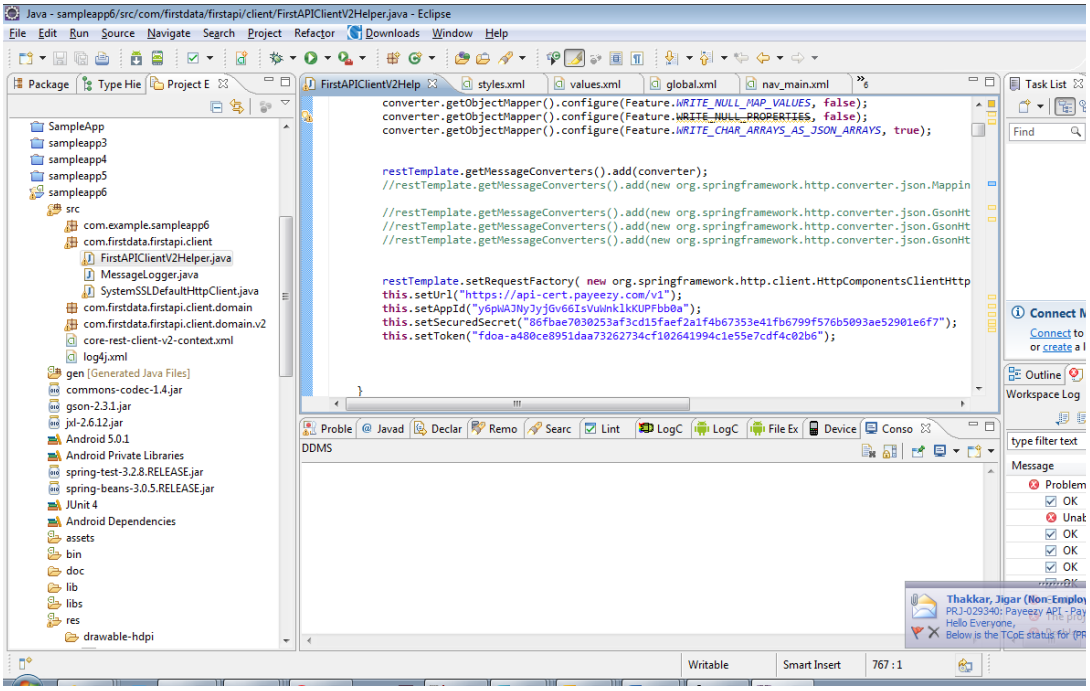
The next screen shows the various string value mappings used to show the screen text.



Following screen shows the menu options



Following screen shows specifying the APIkey, secret and token specification.



Following is the sample code to specify the APIkey, secret, token values. This code needs to be modified in the constructor of the FirstAPIClientV2Helper.

```
this.setUrl("https://api-cert.payeezy.com/v1");
        this.setAppId("y6pWAJNyJyjGv66IsVuWnklkKUPFbb0a");
```

```
this.setSecuredSecret("86fbae7030253af3cd15faef2a1f4b67353e41fb6799f576b5093ae52901e6f7");
this.setToken("fdoa-a480ce8951daa73262734cf102641994c1e55e7cdf4c02b6");
```

Following is the sample code to make payment. The APIkey, secret, token values can be modified in each calls.

```
public String ProcessPayment(String appId, String secretId, String token, String url, TransactionRequest request) {
        String result = "";
        try
        {
                FirstAPIClientV2Helper clientHelper = new  FirstAPIClientV2Helper();
                clientHelper.setAppId(appId);
                clientHelper.setSecuredSecret(secretId);
                clientHelper.setToken(token);
                clientHelper.setUrl(url);

                TransactionRequest request=new TransactionRequest();

                request.setAmount("1100");
                request.setCurrency("USD");
                request.setPaymentMethod("credit_card");

                Card card=new Card();
                card.setCvv("123");
                card.setExpiryDt("1220");
                card.setName("Test data ");
                card.setType("visa");
                card.setNumber("4012000033330026");

                request.setCard(card);
                Address address=new Address();
                request.setBilling(address);
                address.setState("NY");
                address.setAddressLine1("sss");
                address.setZip("11747");
                address.setCountry("US");

                TransactionResponse response = clientHelper.purchaseTransaction(request);
                result = ((UserTransactionResponse)(response)).getResponseString();
        }
        catch(Exception e)
        {
                System.out.println(e.getMessage());
        }
        return result;
}
```

Payeezy Android SDK Installation

To install the android application package (apk) file directly on the android device, download the android application package (apk) file on the android device from the network via GitHub http://www.github.com or email. Touch it and it will ask for install. Select Install. It should get installed on your android device.
Click on various buttons to perform desired functions.

Submitting/Generating transaction (method of payments) with example

Payeezy supports the following method of payments

- Credit Card Payments
- PayPal Transactions
- Gift Card (via ValueLink) Transactions
- eCheck (via TeleCheck) Transactions

For API processing details, click here https://developer-qa.payeezy.com/integration

Additional capabilities API (if applicable)

- Partner Reporting API - Use our powerful query engine to retrieve payment records. Supports complex filtering, sorting, pagination and more. This is exclusively for Third Party Partners and applicable for a **live** environment only.

For Reporting API processing details, click here https://developer-qa.payeezy.com/payeeyz_ref_docs/apis/get/transactions-0

Security related (HMAC, Token generation) with example

GENERATE HMAC

Construct the data param by appending the parameters below in the same order as shown. a. apikey - API key of the developer. b. nonce - secure random number. c. timestamp - epoch timestamp in milliseconds. d. token - Merchant Token. e. payload - Actual body content passed as post request. ***Compute HMAC SHA256 hash on the above data param using the key below*** f. apiSecret - Consumer Secret token for the given api key ***Calculate the base64 of the hash which would be our required Authorization header value.***

```
public String getMacValue(Map<String,String> data) throws Exception{
      Mac mac=Mac.getInstance("HmacSHA256");
      String apiSecret= data.get(APISECRET);
      MessageLogger.logMessage(String.format("API_SECRET:{}",apiSecret));
      SecretKeySpec secret_key = new SecretKeySpec(apiSecret.getBytes(), "HmacSHA256");
      mac.init(secret_key);
      StringBuilder buff=new StringBuilder();
      buff.append(data.get(APIKEY))
      .append(data.get(NONCE))
      .append(data.get(TIMESTAMP));
      if(data.get(TOKEN)!=null)
         buff.append(data.get(TOKEN));
      if(data.get(PAYLOAD)!=null)
         buff.append(data.get(PAYLOAD));
      String bufferData = buff.toString();
      MessageLogger.logMessage(String.format(bufferData));
      byte[] macHash=mac.doFinal(bufferData.getBytes("UTF-8"));
      MessageLogger.logMessage(Integer.toString(macHash.length));
      MessageLogger.logMessage(String.format("MacHAsh:{}",Arrays.toString( macHash)));

      String authorizeString=android.util.Base64.encodeToString(toHex(macHash), android.util.Base64.NO_WRAP);

      MessageLogger.logMessage(String.format("Authorize: {}",authorizeString));
      return authorizeString;
}
```

Reference docs for Payeezy Android SDK

Reference docs (https://github.com/payeezy/payeezy_android/tree/master/javadocs/index.html)