HMS-Wallet-Server Demo Development Tutorial

1. Introduction

HMS-Wallet-Server Demo is example code showing how to use the HMS-Wallet-Server interface. The HMS-Wallet-Server interface contains REST APIs for six types of passes (Loyalty Card, Offer, Gift Card, Boarding Pass, Transit Pass, and Event Ticket). You can use these REST APIs to implement operations such as adding, querying or updating passes.

Before you use this Demo, you should have a HUAWEI developer account, and have already created an app to implement the HMS-Wallet-Kit API. If you haven't, please refer to Register a HUAWEI ID and Creating an App.

Maven and Oracle Java 1.8 are required to run the Demo project.

2. Download HMS-Wallet-Server Demo Code

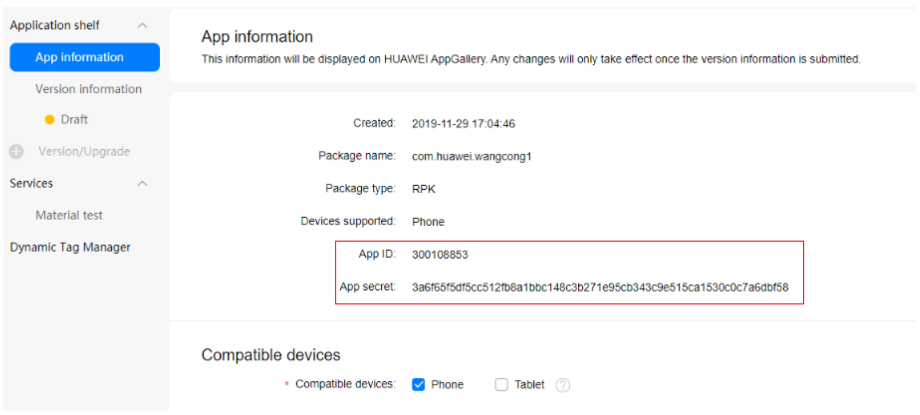
Download “HmsWalletServerDemo.zip” from HUAWEI Wallet Kit Examples and extract the zip file.

3. Set Configuration Values

Before running the Demo project， you need to set the following configuration values in the “src\test\resources\release.config.properties” file: “gw.appid”, “gw.appid.secret”, “gw.cert.name”, “gw.cert.password”, “gw.tokenUrl”, and “walletServerBaseUrl”.

3.1 Set “gw.appid” and “gw.appid.secret”

To implement the HMS Wallet Kit to an app, "gw.appid" and "gw.appid.secret" are this app's "App ID" and "App secret". Go to the HUAWEI AppGallery Connect website, login to your account, click “My apps” and then click the app you want to operate. Then you can find its App ID and App secret as shown below:



App information

This information will be displayed on HUAWEI AppGallery. Any changes will only take effect once the version information is submitted.

Created:	2019-11-29 17:04:46
Package name:	com.huawei.wangcong1
Package type:	RPK
Devices supported:	Phone
App ID:	300108853
App secret:	3a6f65f5df5cc512fb8a1bbc148c3b271e95cb343c9e515ca1530c0c7a6dbf58

Compatible devices

* Compatible devices: ✓ Phone ☐ Tablet

In this case, set gw.appid = 300108853, and gw.appid.secret = 3a6f65ffccf12fb8a1bbv148c3b271e95cb343c9e515ca1530c0c7a6dbf58.

3.2 Set “gw.tokenUrl”

Set gw.tokenUrl = https://oauth-login.cloud.huawei.com/oauth2/v3/token. This is the address to obtain a REST API authentication token.

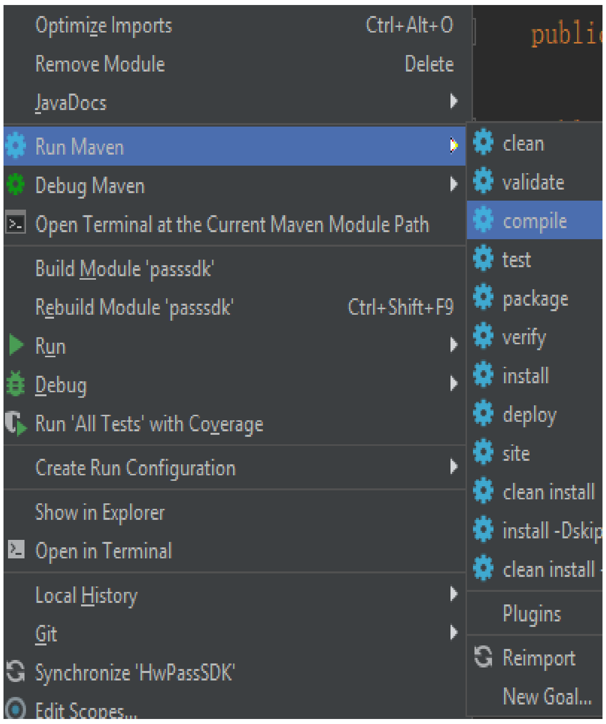
3.3 Set “walletServerBaseUrl”

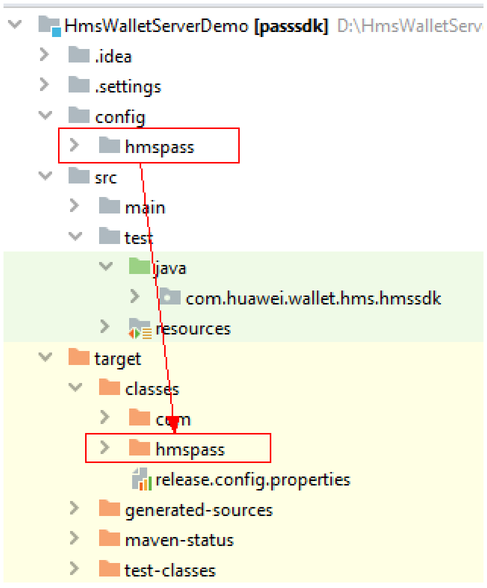
“walletServerBaseUrl” is a common section of the REST APIs’ http requests. Its format is: walletServerBaseUrl = https://${url}/hmspass/v1/. Set ${url} with the value in the following table according to your account’s location.

location	url value
China	passentrust-drcn.wallet.hicloud.com
Russia	passentrust-drru.wallet.hicloud.com
Asia, Africa, and Latin America	passentrust-dra.wallet.hicloud.com
Europe	passentrust-dre.wallet.hicloud.com

3.4 Compile the Demo as a Maven Project

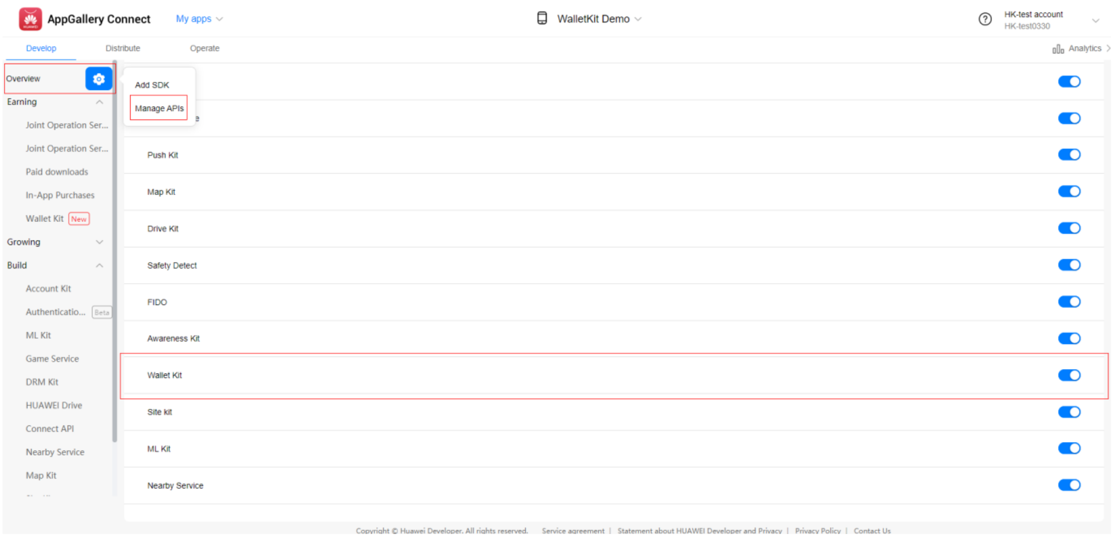
After you set all the configurations values, compile the Demo as a Maven project. Then copy the "config/hmspass" folder to the "target/classes" directory.



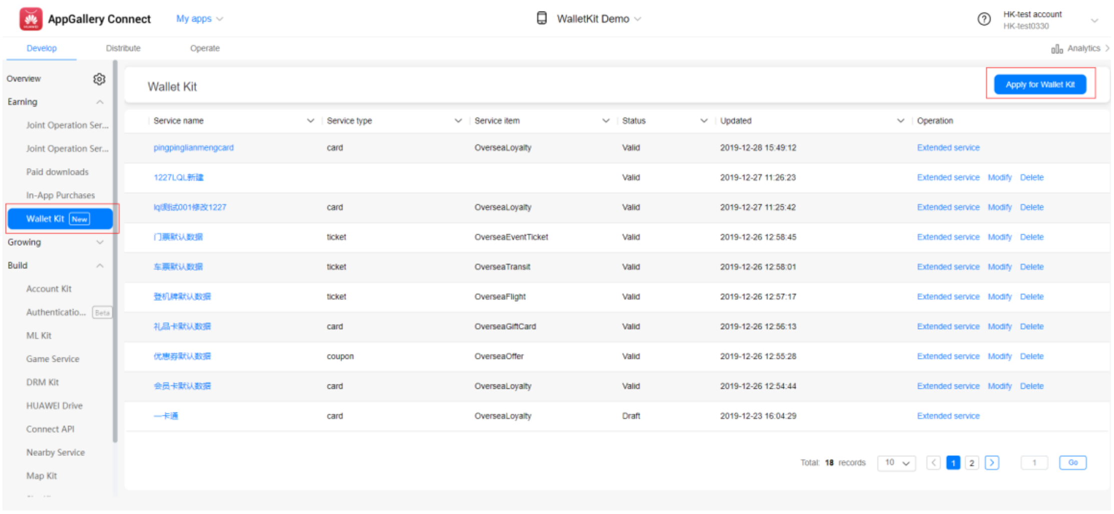


4 Apply for Wallet Services

Make sure Wallet Kit API is enabled. Go to “Develop”->“Manage APIs” on the AGC website to check it.



Then apply for a service for one of the six passes. Please refer to Preparations.

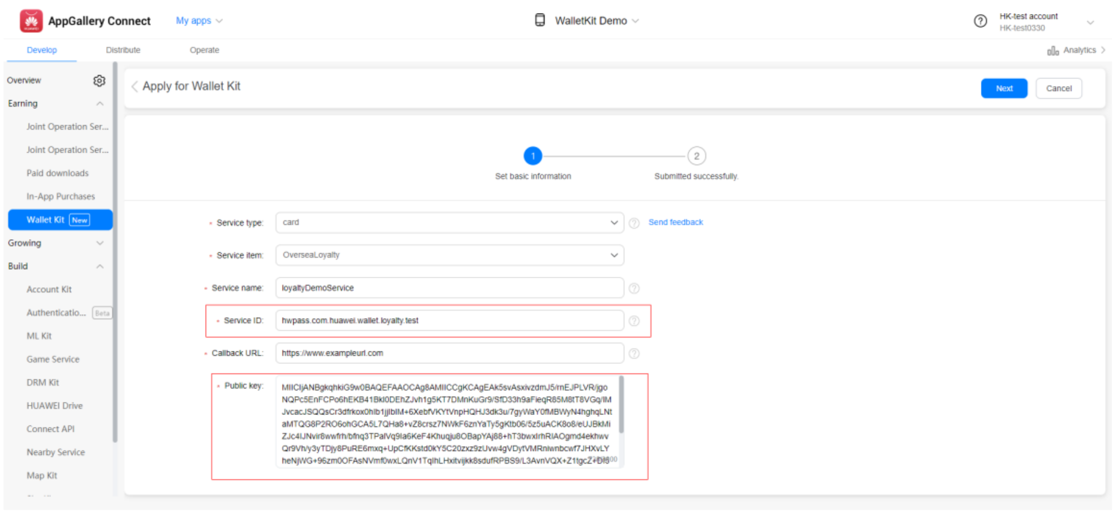


Wallet Kit

Service name	Service type	Service item	Status	Updated	Operation
pingpinglianmengcard	card	OverseaLoyalty	Valid	2019-12-28 15:49:12	Extended service
1227LQL新建			Valid	2019-12-27 11:26:23	Extended service Modify Delete
lq测试001修改1227	card	OverseaLoyalty	Valid	2019-12-27 11:25:42	Extended service Modify Delete
门票默认数据	ticket	OverseaEventTicket	Valid	2019-12-26 12:58:45	Extended service Modify Delete
车票默认数据	ticket	OverseaTransit	Valid	2019-12-26 12:58:01	Extended service Modify Delete
登机牌默认数据	ticket	OverseaFlight	Valid	2019-12-26 12:57:17	Extended service Modify Delete
礼品卡默认数据	card	OverseaGiftCard	Valid	2019-12-26 12:56:13	Extended service Modify Delete
优惠券默认数据	coupon	OverseaOffer	Valid	2019-12-26 12:55:28	Extended service Modify Delete
会员卡默认数据	card	OverseaLoyalty	Valid	2019-12-26 12:54:44	Extended service Modify Delete
一卡通	card	OverseaLoyalty	Draft	2019-12-23 16:04:29	Extended service

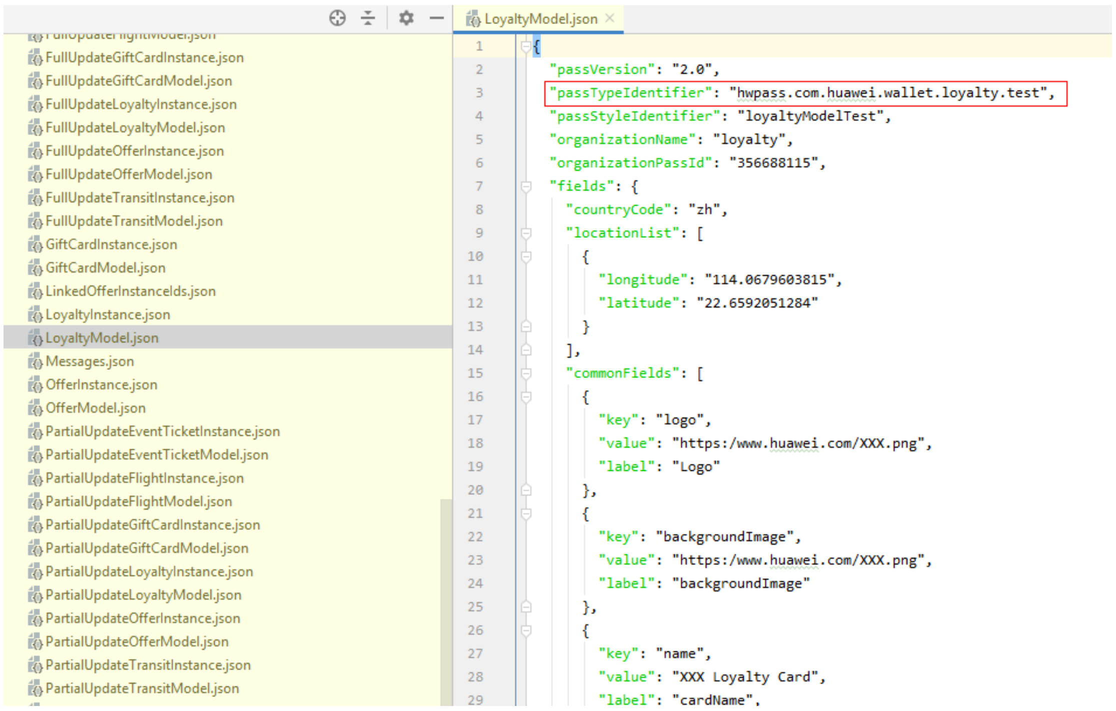
Total: **18** records 10 < 1 2 > 1 Go

Please notice that you will set a “Service ID” while applying for a wallet service. The “Service ID” here is the “passTypeIdentifier” you will use in the Demo code (in JSON input files). Make sure you use the correct “passTypeIdentifier” in JSON input files before calling example methods.



Apply for Wallet Kit

Service type:	card
Service item:	OverseaLoyalty
Service name:	loyaltyDemoService
Service ID:	hwpass.com.huawei.wallet.loyalty.test
Callback URL:	https://www.exampleurl.com
Public key:	MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAk5svAsxivzdmJ5/rnEJPLVR/jgo NQPc5EnFCPo6hEKB41BkI0DEhZJvh1g5KT7DMnKuGr9/SfD33h9aFieqR85M8tT8VGq/IM JvcacJSQQsCr3dfrkox0hIb1jjIbIM+6XebfVKYtVnpHQHJ3dk3u/7gyWaY0fMBWyN4hghqLNt aMTQG8P2RO6ohGCA5L7QHa8+vZ8crsz7NWkF6znYaTy5gKtb06/5z5uACK8o8/eUJBkMi ZJc4IJNvir8wwfrh/bfnq3TPalVq9Ia6KeF4Khuqju8OBapYAj88+hT3bwxlrhRIAOgmd4ekhwv Qr9Vh/y3yTDjy8PuRE6mxq+UpCfKKstd0kY5C20zxz9zUvw4gVDytVMRniwnbcwf7JHXvLY heNjWG+96zm0OFAsNVmf0wxLQnV1TqlhLHxitvijkk8sdufRPBS9/L3AvnVQX+Z1tgcZ+Dl6



FullUpdateGiftCardInstance.json
FullUpdateGiftCardModel.json
FullUpdateLoyaltyInstance.json
FullUpdateLoyaltyModel.json
FullUpdateOfferInstance.json
FullUpdateOfferModel.json
FullUpdateTransitInstance.json
FullUpdateTransitModel.json
GiftCardInstance.json
GiftCardModel.json
LinkedOfferInstanceIds.json
LoyaltyInstance.json
LoyaltyModel.json
Messages.json
OfferInstance.json
OfferModel.json
PartialUpdateEventTicketInstance.json
PartialUpdateEventTicketModel.json
PartialUpdateFlightInstance.json
PartialUpdateFlightModel.json
PartialUpdateGiftCardInstance.json
PartialUpdateGiftCardModel.json
PartialUpdateLoyaltyInstance.json
PartialUpdateLoyaltyModel.json
PartialUpdateOfferInstance.json
PartialUpdateOfferModel.json
PartialUpdateTransitInstance.json
PartialUpdateTransitModel.json

```json
{
  "passVersion": "2.0",
  "passTypeIdentifier": "hwpass.com.huawei.wallet.loyalty.test",
  "passStyleIdentifier": "loyaltyModelTest",
  "organizationName": "loyalty",
  "organizationPassId": "356688115",
  "fields": {
    "countryCode": "zh",
    "locationList": [
      {
        "longitude": "114.0679603815",
        "latitude": "22.6592051284"
      }
    ],
    "commonFields": [
      {
        "key": "logo",
        "value": "https:/www.huawei.com/XXX.png",
        "label": "Logo"
      },
      {
        "key": "backgroundImage",
        "value": "https:/www.huawei.com/XXX.png",
        "label": "backgroundImage"
      },
      {
        "key": "name",
        "value": "XXX Loyalty Card",
        "label": "cardName",
```

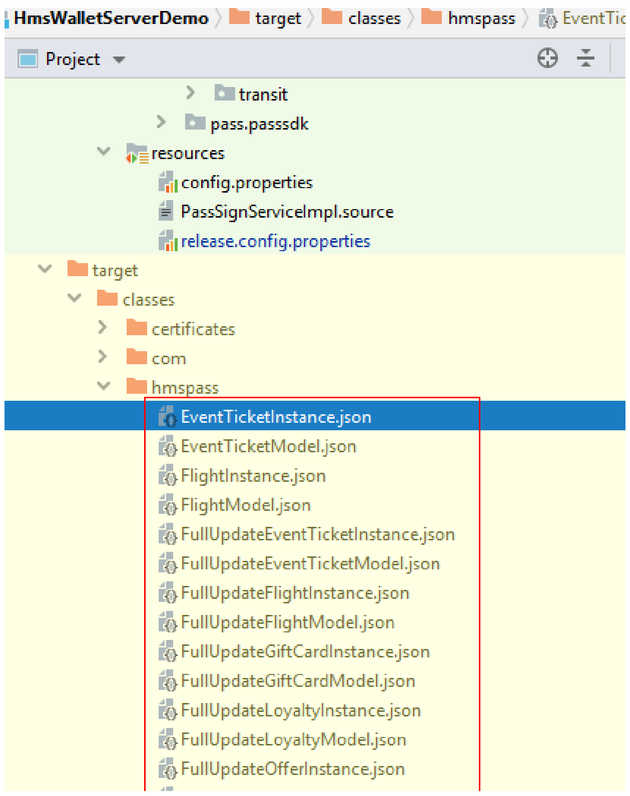
After you finished applying for a service, you can begin to test the corresponding pass. Apply for other services if you want to test other passes.

5 Wallet Models and Wallet Instances

A wallet model is a style of wallet instances. Instances belonging to the same model share some common parameters, which are stored in the model. For example, a boarding pass model contains information about departure time and arrival time, while a boarding pass instance contains a passenger's name, his seat, his boarding sequence, etc. Each wallet instance belongs to a specific wallet model. Hence, you should first create a model before creating instances and performing other operations.

All wallet models and instances have the same data format, which is HwWalletObject. Refer to HwWalletObject Definition for more details.

Input parameters of models and instances are passed by JSON files in the Demo project. You can generate your own data by modifying these JSON files.



```
HmsWalletServerDemo 〉 target 〉 classes 〉 hmspass 〉 EventTic
Project
        > transit
      > pass.passsdk
    ∨ resources
        config.properties
        PassSignServiceImpl.source
        release.config.properties
  ∨ target
    ∨ classes
      > certificates
      > com
      ∨ hmspass
          EventTicketInstance.json
          EventTicketModel.json
          FlightInstance.json
          FlightModel.json
          FullUpdateEventTicketInstance.json
          FullUpdateEventTicketModel.json
          FullUpdateFlightInstance.json
          FullUpdateFlightModel.json
          FullUpdateGiftCardInstance.json
          FullUpdateGiftCardModel.json
          FullUpdateLoyaltyInstance.json
          FullUpdateLoyaltyModel.json
          FullUpdateOfferInstance.json
```

6. Example Methods for Wallet Models

6.1 Create a Wallet Model

The HMS wallet server provides REST APIs to create wallet models. You can add a loyalty model to the server's database by calling the "createLoyaltyModel" method, and create other types of models likewise. You should create a wallet model first before calling any other methods.

6.2 Query a Wallet Model

The HMS wallet server provides REST APIs to query a wallet model by its model ID. You can query a loyalty model by calling the "getLoyaltyModel" method, and query other types of models likewise. You can only query or update models and instances created by your own app.

6.3 Query a List of Wallet Models

If your app created multiple models of a pass type (e.g. a gold loyalty model and a diamond loyalty model), you can use these APIs to query a list of these models. You can query a list of loyalty models by calling the "getLoyaltyModelList" method, and query other types of models likewise. You can set the number of loyalty models you want to get, or simply get all loyalty models created by your app.

6.4 Overwrite a Wallet Model

The HMS wallet server provides REST APIs to overwrite a wallet model by its model ID. You can overwrite a loyalty model by calling the “fullUpdateLoyaltyModel” method, and overwrite other types of models likewise.

6.5 Update a Wallet Model

The HMS wallet server provides REST APIs to update a wallet model by its model ID. You can overwrite a loyalty model by calling the “partialUpdateLoyaltyModel” method, and update other types of models likewise.

6.6 Add Messages to a Wallet Model

“messageList” is one of the attributes in a wallet model, which is a list of messages. You can add messages to a loyalty model by calling the “addMessageToLoyaltyModel” method, and add to other types of models likewise. The “messageList” in a wallet model has at most 10 messages. You cannot add more than 10 messages at a time. If the list’s size is already 10 and you keep adding messages, the oldest messages will be removed.

7. Example Methods for Wallet Instances

7.1 Create a Wallet Instance

The HMS wallet server provides REST APIs to create wallet instances. You can add a loyalty instance to the server’s database by calling the “createLoyaltyInstance” method, and create other types of instances likewise. You should create a wallet model first before creating instances belonging to it.

7.2 Query a Wallet Instance

The HMS wallet server provides REST APIs to query a wallet instance by its instance ID. You can query a loyalty instance by calling the “getLoyaltyInstance” method, and query other types of instances likewise.

7.3 Query a List of Wallet Instances

You can query a list of loyalty instances belonging to a specific loyalty model by calling the “getLoyaltyInstanceList” method, and query other types of instances likewise. You can set the number of loyalty instances you want to get, or simply get all loyalty instances belonging to that model.

7.4 Overwrite a Wallet Instance

The HMS wallet server provides REST APIs to overwrite a wallet instance by its instance ID. You can overwrite a loyalty instance by calling the “fullUpdateLoyaltyInstance” method, and overwrite other types of instances likewise.

7.5 Update a Wallet Instance

The HMS wallet server provides REST APIs to update a wallet instance by its instance ID.

You can overwrite a loyalty instance by calling the “partialUpdateLoyaltyInstance” method, and update other types of instances likewise.

7.6 Add Messages to a Wallet Instance

“messageList” can also be an attribute of a wallet instance. You can add messages to a loyalty instance by calling the “addMessageToLoyaltyInstance” method, and add to other types of instances likewise. The “messageList” in a wallet instance has at most 10 messages. You cannot add more than 10 messages at a time. If the list’s size is already10 and you keep adding messages, the oldest messages will be removed.

7.7 Link/unlink Offer Instances to/from a Loyalty Instance

This API is only provided for loyalty instances. You can link/unlink offer instances to/from a loyalty instance by calling the “updateLinkedOffersToLoyaltyInstance” method. You should make sure the offers you want to add already exist before you use this API. Otherwise, the client cannot show an offer that is not in the server’s database. These offer instances can belong to other apps or other developers.