JoyTel API

> R.20240222.01

Change Log

Date	Edit
2024.06.12	Added .net url for Warehouse APIs.
2024.07.18	1. Added eSIM Installation Event Notification, eSIM Profile Information Query interfaces. 2. Modified order process time from 2 minutes to 30 seconds utmost. 3. Change interface name `Transaction Status Query` into `Get Transaction Status`.
2024.07.22	Added eSIM Installation Event Notification.
2025.02.26	Removed a confusing descritipon for snCode. ~~snCode is not CID~~.
2025.04.10	Modified eSIM order callback re-try mechanism.

Introduction

Document Purpose

This document is a technical guidance for JoyTel eSIM & OTA SIM customers to interconnect with JoyTel system via API. This document illustrates the functional architecture, procedures and detailed definition of interfaces of two different JoyTel systems that collaborately support the eSIM & OTA SIM business.

Intended Audience

For developers of JoyTel's customers to integrate JoyTel APIs for eSIM and OTA SIM business.

For product managers and business managers who are in charge of business and technical integration with JoyTel.

Terms

Terms / Parameter	Description
snPin	Same with coupon. snPin is the parameter name used in Warehouse system.
Coupon	Coupon is used to get the QR code. Coupon is the parameter name used in the RSP+ system. *Same with snPin.* Coupon is also known as redemption code in older version of API docs.
snCode	Serial code of OTA SIM or eSIM, format: 898620003xxxxxxx.
CID	For OTA SIM, the CID of OTA SIM card is printed on the card board along with bar code. For eSIM, CID is the unique ID of eSIM profile.

Tech Support

Tech contact

- Group chat for technical support is available via Wechat, Skype
- Support mail: tuulis@joytelecom.com
- Tech suppport for API integration is availalbe during 9:00 - 18:00 UTC+8 work day

Demo

- Postman (offline Postman collection export).

General Architecture

Involved System

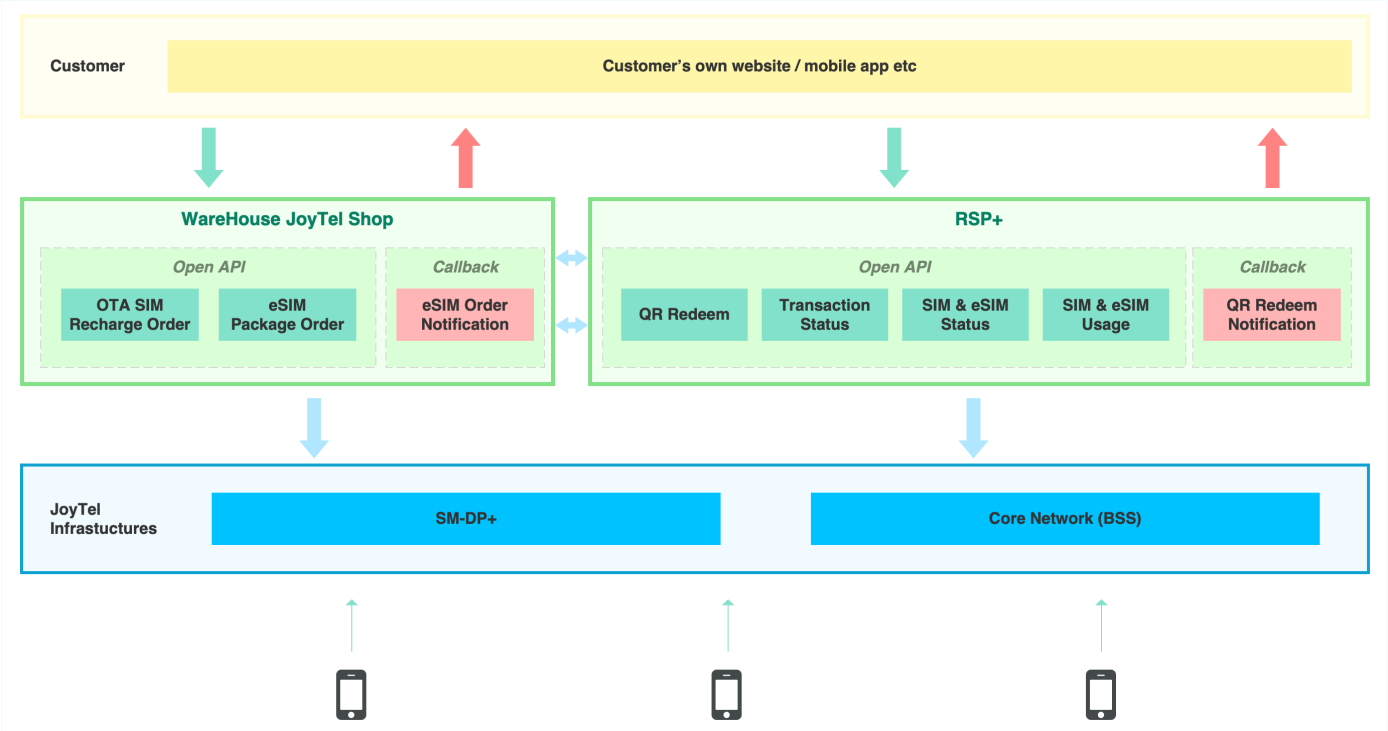
This API document involves two systems:

- **JoyTel Shop Wareshouse system**:
  - Call interfaces of this system to place order, get order response, query order.
  - This system supports both eSIM order and OTA card recharge order.
- **JoyTel RSP+ System**:
  - Call interfaces of this sytem to get eSIM QR code;
  - Query eSIM status and usage and other operation data.
  - Query OTA SIM status and usage data.
  - Query eSIM profile status.

Funcional Architecture

For eSIM business, customer needs to integrate both the WareHouse JoyTel Shop interfaces and RSP+ Interfaces.

For OTA SIM business, customer only needs to refert to Warehouse - OTA Card Recharge API.



API Authentication

To get started, please provide your **IP whitelist**, **eSIM OrderCallback(snPin)** to JoyTel.

System 1: Warehouse API:

Parameter	Provided by	Remark
customer	JoyTel	Customer name.
customerCode	JoyTel	Get from JoyTel. This is used in warehouse API authentication.
customerAuth	JoyTel	Get from JoyTel. This is used in warehouse API authentication.
eSIM Order Callback(snPin)	Customer	If you need this kind of callback,**Provide it to JoyTel.** This is for eSIM business only.

System 2: RSP+ eSIM:

Parameter	Provided by	Remark
API Config Name	JoyTel	API account name.
AppID	JoyTel	Get from JoyTel. This is used in RSP+ API authentication.
AppSecret	JoyTel	Get from JoyTel. This is used in RSP+ API authentication.
BaseURL	JoyTel	https://esim.joytelecom.com/openapi
QR Code Callback	Customer	**Provide it to JoyTel**. Must be endded with `/notify/coupon/redeem`. This is for eSIM business only, and this is used for the QR code delivery.

IP Whitelist:

Customer's server IP must be whitelisted by JoyTel server.

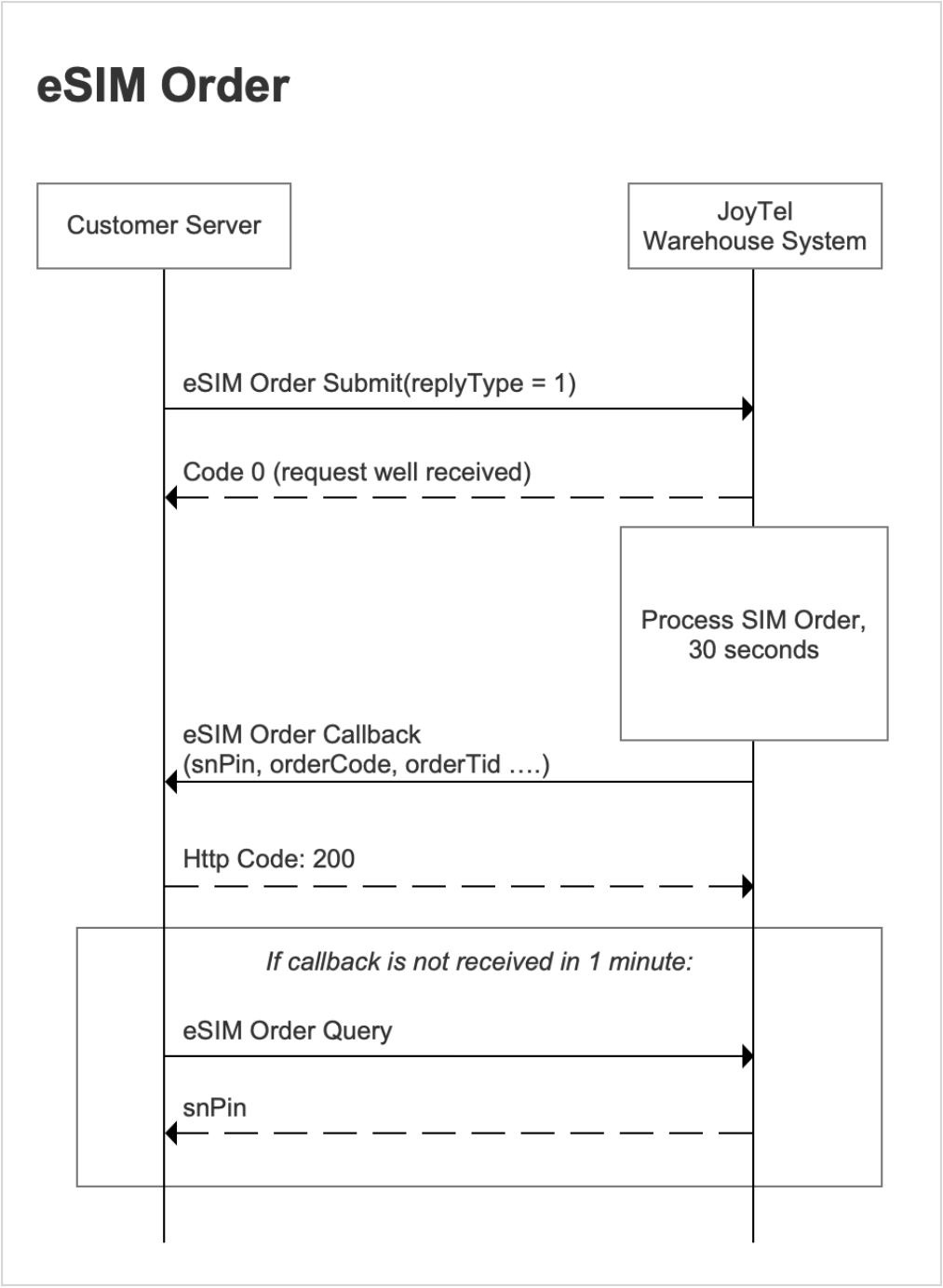
Customer can provide multiple IPs for whitelist.

Procedure

Callback mechanism is used for asynchoronous processes. Callback is also known as **webhook** and **notification**.

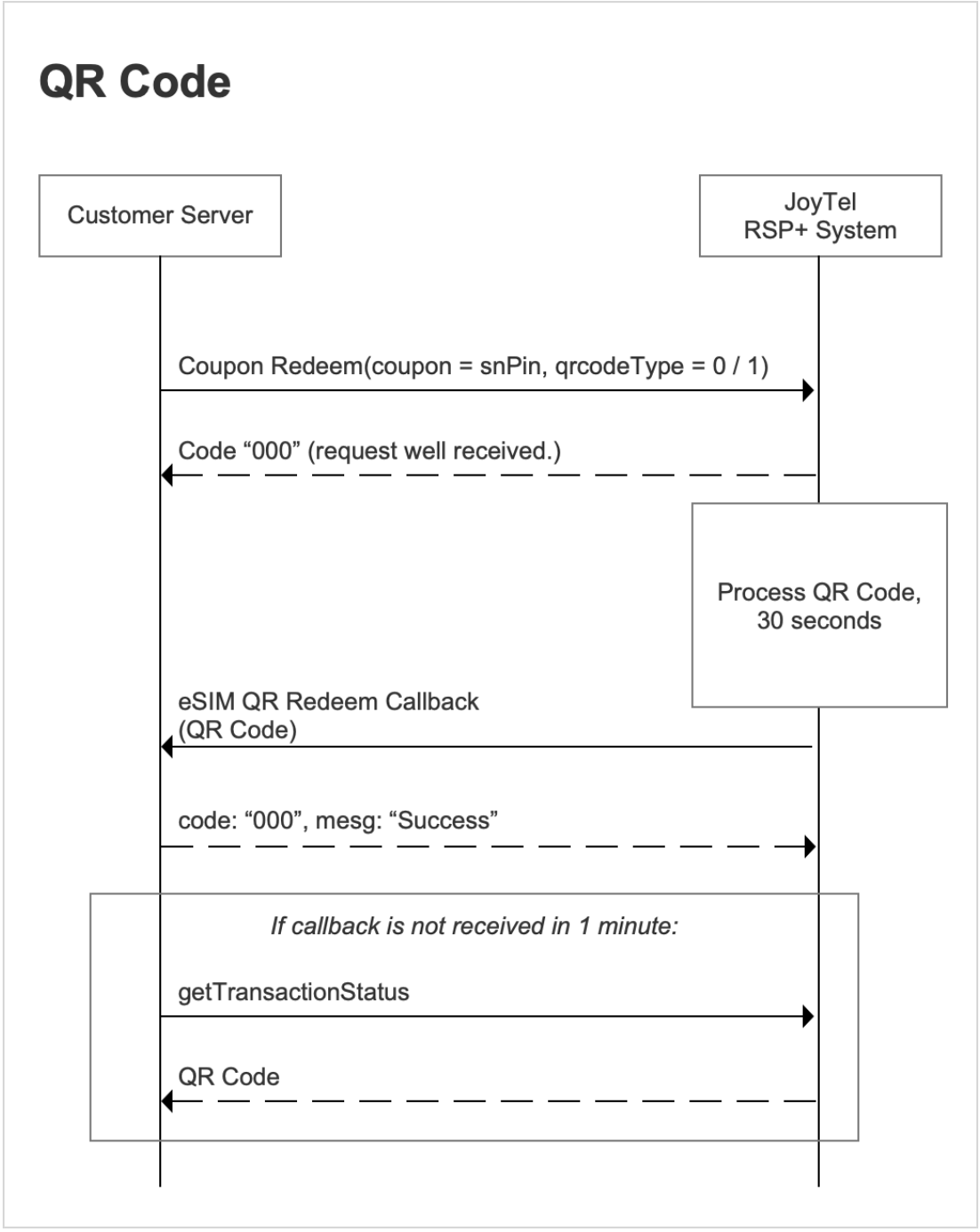
eSIM Order Submit and Callback(snPin) Procedure

1. Submit order first and wait for JoyTel server to notify your server.
2. Make sure `replyType = 1` in your order request to get notification from JoyTel server to your server. If `replyType = 1` is sent to 0, the roder result will be returned via e-mail.
3. Order process will take utmost **30 seconds**. If your server fails to receive order result notification from JoyTel server, you can call the eSIM `Order Query interface` to check the order.
4. When your server receives order callback request from JoyTel, retreive the parameters and store them into your database if needed.
5. `snPin` is useful for your next step to get QR Code. snPin is equivalent to coupon.
6. Return `http code 200` to JoyTel server for order callback.
7. eSIM Order callback will be sent multiple times if your callback URL does not respond with `http code 200`. The repeat rule is below:
    - Callback will be re-sent, after 3 minutes from first callback;
    - Another callback will be re-sent, after 10 minutes from last callback
    - Another callback will be re-sent, after 20 minutes from last callback
    - Another callback will be re-sent, after 30 minutes from last callback
    - Callback ends if your callback url never responded.



Redeem Coupon to Get QR Code Procedure

1. When you have the snPin, it's time to get QR code by calling system 2 interface: `Coupon Redeem`
2. NOTE: parameter `coupon` is same thing with `snPin`.
3. QR Code is sent to your server with callback request.
4. When your server receives QR Code callback request, do retreive data and store them into your data base.
5. Return `code: "000, mesg: "Success"` to JoyTel RSP system for QR Code callback, otherwise your server will receive the callback repeatedly.



Product Code

Test Product Code

Test producutCodes are availalbe for integrating both eSIM API and OTA SIM API:

- **eSIM-test** for eSIM API development and testing.
- **CZ-test** for OTA card API development and testing. For CZ-test, parameter `days` can only be set to 1.

Production Product Code

Get production product code from your JoyTel account manager.

System 1: Warehouse - eSIM Order API for customers

Thisi API is for customers to place eSIM orders.

eSIM order is a assynchronous process. JoyTel server will return eSIM order result with callback mechanism, which means JoyTel server will send a request to customer's server. Customer needs to provide their server's URL to receive the callback request from JoyTel server.

Test `productCode` for eSIM order: ***eSIM-test***

1. eSIM Order Submit

1. API for eSIM Purchase Order Submit
2. JoyTel will return snPin after a successful order request. **snPin** is also known as **coupon** in RSP+ system, which is used to redeem QR code later on.
3. Refer to RSP+ `2.1.2 Coupon Redeem` API afer receiving **snPin**.
4. For server in China, this URL is recommended:

   https://api.joytel.vip/customerApi/customerOrder (also available for server outside China)

   For server outside China, this URL is recommended:

   https://api.joytelshop.com/customerApi/customerOrder

   or

   https://api.joytelshop.net/customerApi/customerOrder
5. Method： `POST`
6. Request header

Parameter	Value	M/O	Comments
Content-Type	application/json	M	

5. INPUT Param

Parameter	Type	M/O	Comments
customerCode	String	M	customer Code (Assigned by JOYTEL)
orderTid	String	M	Customer's self defined order id, Strongly suggest to have a unique ID on customer side in order to have a unique reference between customer and JOYTEL, also useful when customer want to check order status and information with JOYTEL system by using query API. Example to generate aunique ID on customer sider: (customerCode+yyyyMMddHHmmss+random six digits);this parameter needs to be unique
type	Int	M	type: 3 for eSIM order with snPin
warehouse	String	O	"上海仓库" by default. (Don't have to send this.)
receiveName	String	M	Receiver name/customer name
phone	String	M	Receiver phone number/customer phone number
timestamp	Long	M	Current order timestamp
autoGraph	String	M	Secure Parameter Check, **calculated with following *order***: customerCode+customerAuth+warehouse+type+orderTid+ receiveName+phone+timestamp+itemList（productCode+quantity）， all contents are then encrypted by SHA-1 and use the result here. If some parameter is empty then use empty string "". customerAuth is assigned with customerCode by JOYTEL, it is the unique authentication value defined to the customer, example: **LGUw5Q1J**
email	String	M	End user's e-mail.
replyType	Int	O	**replyType: 0**, order result will be returned with mail **replyType: 1**, order result with snPin will be returned to customer's server. Customer needs to redeem snPin to get QR Code. Make sure your callback/webhook settings are correct.
emailTmpl	String	O	Email contents template
remark	String	O	Order Remark
itemList	Object[]	M	Order details

itemList：(maximum 50 item per order)

Parameter	Type	M/O	Comments
productCode	String	M	Product Code/ID predefined and shared by JOYTEL
quantity	Int	M	Quantity of product ordered forthe same product code

6. Examples

```
{
    "customerCode": "test001",
    "type": 3,
    "receiveName": "test",
    "phone": "15666666666",
    "timestamp": 1667807404146,
    "autoGraph": "ae09d951095d44faabf3c91a9879afdc477dd630",
    "remark": "test",
    "itemList":
    [
        {
            "productCode": "esim615xxxx1",
            "quantity": 1
        },
```

```
        {
            "productCode": "esim615xxxx2",
            "quantity": 1
        }
    ],
    "email": "test@qq.com"
}
```

Encryption Example：

Original String：test001abcdefj3test1566666666616678074041 46esim615xxxx11esim615xxxx21

Encrypted String：ae09d951095d44faabf3c91a9879afdc477dd630

7. Response：

Parameter	Type	Comments
message	String	Response Description
code	Int	Response Message Code
data	Object	Business datasets

data：

Parameter	Type	Comments
orderTid	String	Defined by customer andtransferred in the request, JOYTEL API will response with the same idCould be used to queryorder information
orderCode	String	The unique order iddefined on JOYTEL side, could also be used to query order information

8. Response Example：

```
{
    "message": "Operation Success",
    "code": 0,
    "data":
    {
        "orderTid": "xxxxx",
        "orderCode": "DD-230303-xxxx"
    }
}
```

2. eSIM Order Query

1. API for eSIM Purchase Order Query

2. Reccomendation: Use this API if order callback fails. For example, if order callback is not received in 2 minutes, call eSIM Order Query API to check the eSIM order result.
3. For server in China, this URL is recommended:

   https://api.joytel.vip/customerApi/customerOrder/query (also available for server outside China)

   For server out side China, this URL is recommended:

   https://api.joytelshop.com/customerApi/customerOrder/query

   or

   https://api.joytelshop.net/customerApi/customerOrder/query
4. Method： `POST`
5. Request Header

Parameter	Value	M/O	Comments
Content-Type	application/json	M	

5. INPUT Parameter

Parameter	Type	M/O	Comments
customerCode	String	M	customer Code (Assignedby JOYTEL)
orderCode	String	M/O	The unique order iddefined on JOYTEL side, for query either this orderCode or orderTid should beused to identify a unique order
orderTid	String	M/O	customer's own definedorder id, for query either this orderTid or orderCode should be used toindentify a unique order
timestamp	Long	M	Current query Timestamp
autoGraph	String	M	Secure Parameter Check,calculated as following: customerCode+customerAuth+orderCode+orderTid+timestamp all contents are then encrypted by SHA-1 and use the result here. If some parameter is empty then use empty directly asthe string. customerAuth is assigned with customerCode by JOYTEL, it is the unique authentication value defined tothe customer, example:LGUw5Q1J

6. Examples：

```
{
    "customerCode": "test001",
    "timestamp": 1667807404146,
    "autoGraph": "cc348f99b331dc43154203ef3026c99d0a75ea54",
    "orderCode": "DD-230304-xxxxxx"
}
```

Encryption Example：

Original String：test001abcdefjDD-230304-xxxxxx1667807404146

Encrypted String：cc348f99b331dc43154203ef3026c99d0a75ea54

7. Response：

Parameter	Type	Comments
message	String	Response Description
code	Int	Response Message Code
data	Object	Business datasets

data：

Parameter	Type	Comments
orderCode	String	The unique order iddefined on JOYTEL side, could also be used to query order information
orderTid	String	Defined by customer andtransferred in the request, JOYTEL API will response with the same idCould be used to queryorder information
phone	String	Receiver phone number/customer phone number
outboundCode	String	JOYTEL warehouse deliver code, not useful for eSIM orders
receiveName	String	Receiver customer names/customer name
email	String	customer email address
status	Int	order status：1, Submitted (awaiting for validation) 2, Validated 3, Waiting for delivery 4, Delivered 0 ,Issues on Order -1,Cancelled -2,merged orders
itemList	Object[]	Order details

itemList：

Parameter	Type	Comments
productName	String	Product Name
quantity	Int	Product in quantity
snList	Object[]	sn information
productCode	String	product code

snList：

Parameter	Type	Comments
snCode	String	sn code
snPin	String	Sn pin
productExpireDate	String	Expiration date in yyyy-MM-dd

8. Response Example：


```
{
    "message": "Operation Success",
    "code": 0,
    "data":
    {
        "itemList":
        [
            {
                "productCode": "eSIM-Dxxxxx",
                "snList":
                [
                    {
                        "snCode": "898620003xxxxxxx",
                        "productExpireDate": "2023-12-30",
                        "snPin": "xxxx"
                    }
                ],
                "quantity": 1,
                "productName": "eSIM-JOY-xxxxx"
            }
        ],
        "orderCode": "DD-230311-xxxxx",
        "orderTid": "xxxx20230311120211654321",
        "phone": "18521xxxx",
        "outboundCode": "OB-230311-xxxxx",
        "receiveName": "线上销售",
        "email": "xxxxxx",
        "status": 4
    }
}
```


3. Call Back (snPin)

1. JoyTel's callback to customer's server to send snPin, and customer needs to redeem snPin to get LPA.
2. Invocation Direction: **JoyTel Server** -> **Customer Server**
3. Request URL:
    - If you want to receive this callback, please provide your server's URL to JoyTel.

    - And setup your server correctly and handle request from JoyTel.
4. After receiving this request from JoyTel, customer's server can store snPin. Then redeem snPin to get QR code. Please note that the snPin in this API cocument is equivalent to coupon in RSP+ API document.
5. Method: `POST`
6. Request Hearders

Parameter	Value	M/O	Comments
Content-Type	application/json	M	

5. INPUT Parameters：

Parameter	Type	Comments
orderCode	String	The unique order iddefined on JOYTEL side, could also be used to query order information
orderTid	String	Defined by customer and transferred in the request, JOYTEL API will response with the same idCould be used to query order information
phone	String	Receiver phone number/customer phone number
outboundCode	String	JOYTEL warehouse delivercode, not useful for eSIM orders
receiveName	String	Receiver customer names/customer name
email	String	customer email address
status	Int	order status：1, Submitted (awaiting for validation) 2, Validated 3, Waiting for delivery 4, Delivered 0, Issues on Order -1, Cancelled -2, merged orders
itemList	Object[]	Order details

6. itemList：

Parameter	Type	Comments
productName	String	Product Name
quantity	Int	Product in quantity
snList	Object[]	sn information
productCode	String	product code

7. snList：

Parameter	Type	Comments
snCode	String	sn code
snPin	String	Sn pin
productExpireDate	String	Expiration date in yyyy-MM-dd

5. Response Code

Response Code	Response Message	Response Message (EN)
0	操作成功	Operation Success.
1	请求参数鉴权失败	Request Authentication Failed.
2	公共参数必填项未按要求填写	Mandatory Parameter Missing
3	渠道商余额不足	Insufficient Balance
4	不支持该业务（具体描述请参考message）	Service not supported. (Reference the actual response message)
5	该充值单已充值	Order Already Exists(check orderCode or orderTid to avoid duplicate)
6	订单项超过了限制（具体描述请参考message）	Order Item Exceeds limit (50)
-1	请求数据异常	Request Exceptions
100000	未知的服务异常	Unknown Service Exceptions

System 1 - Warehouse - OTA Card Recharge API for customers

This part is intended for JoyTel OTA SIM card business only. There is no callback for OTA card recharge order. Customer needs to query order result after submitting a recharge order.

Test `productCode` for Recharge: ***CZ-test***

MOC:

Mandatory, Optional, Conditional

1 Joy Card Recharge

This interface is used to submit top-up/recharge order.

· **Request URL:**

https://api.joytel.vip/joyRechargeApi/rechargeOrder

For server outside China, use the proxy URL:

https://api.joytelshop.com/joyRechargeApi/rechargeOrder

or

https://api.joytelshop.net/joyRechargeApi/rechargeOrder

· **Request Method:**

POST

· **Request Header:**

Parameter Name	Value	MOC	Description
Content-Type	application/json	M	

· **Body**:

Parameter Name	Parameter Type	MOC	Description
customerCode	String	M	Channel code, provided by JoyTel
orderTid	String	M	Channel platform order id: A unique id for recharge order, with a format *customerCode+xxx.* The *xxx* part can be customized following your own rule within a length limit of 32bit string. Make sure it is unique and cannot be repeated.
timestamp	Long	M	Timestamp of current request
autoGraph	String	M	Parameter security verification, with the calculation method: customerCode+customerAuth+timestamp+itemList + orderTid itemList is the accumulated calculation of *productCode+snCode+days* customerAuth is provided by JoyTel. It is a unique id for each channel, for example coded like this: LGUw5Q1J. Calculate SHA-1 with the result of the above calculation.
itemList	Object [ ]	M	List of recharge plans. Maximum length of itemList is 50.

· ***itemList* object**

Parameter Name	Parameter Type	MOC	Description
productCode	String	M	Recharge Prodcut Code.
snCode	String	M	Serial number. **Get a snCode from JoyTel for test purpose, if you don't have one.**
days	Int	M	days counts. For Daily bundle, pass days >= 1. For Total bundle, pass days = 1. For CZ-test, parameter `days` can only be set to 1.

- **Response**:

Parameter Name	Parameter Type	Description
message	String	Response description
code	Int	Response code
data	Object	Response data object list below

- ***data* object**

Parameter Name	Parameter Type	Description
rechargeCode	String	Recharge order code, generated by JoyTel.
orderTid	String	Channel platform recharge order id which was previously send in request body.

- **Response Example:**

```
{
   "message": "操作成功",
   "code": 0,
   "data": {
     "rechargeCode": "CZ-230203-xxxx",
     "orderTid": "customerCode-xxx"
   }
 }
```

2 Recharge Order Status Inquiry

This interface is used to inquire the recharge order status, after submitting a top-up/recharge order.

· **Request URL:**
https://api.joytel.vip/joyRechargeApi/rechargeOrder/query
For server outside China, use proxy URL:
https://api.joytelshop.com/joyRechargeApi/rechargeOrder/query

or

https://api.joytelshop.net/joyRechargeApi/rechargeOrder/query

· **Request Method:**
POST
· **Request Header:**

Parameter Name	Value	MOC	Description
Content-Type	application/json	M	

· **Body:**

Parameter Name	Parameter Type	MOC	Description
customerCode	String	M	Channel code, provided by JoyTel
orderTid	String	C	Channel platform order id which was sent in the JoyTel Card Recharge request. If orderTid is passed in this request, rechargeCode can be null, vice versa.
rechargeCode	String	C	If orderTid is not null in this request, rechargeCode must be included, vice versa.
timestamp	Long	M	Timestamp of current request.
autoGraph	String	M	Parameter security verification, with the calculation method: *customerCode+customerAuth+timestamp +rechargeCode+orderTid* customerAuth is provided by JoyTel. It is a unique id for each channel, for example coded like this: LGUw5Q1J. Calculate SHA-1 with the result of the above calculation.

· **Response:**

Parameter Name	Parameter Type	Description
message	String	Response description
code	Int	Response code
data	Object	Response data object list below

· ***data* object**

Parameter Name	Parameter Type	Description
orderTid	String	Channel platform recharge order id which was previously send in request body.
rechargeCode	String	Recharge order code,
itemList	Object [ ]	JoyTel Card Recharge List

- ***itemList***

Parameter Name	Parameter Type	Description
status	Int	0: Recharging 1: Success 2: Fail 3: Submitted Successfully 4. Need to Recharge
productCode	String	Recharge product list.
productExpireData	String	Product last valid date (YYYY-MM-DD)
snCode	String	Platform sn code.
rspOrderId	String	RSP order id. Use this ID to query OTA SIM usage.
rspTid	String	RSP Tid.
statusDesc	String	Recharge status description.

- **Response Example:**


```
{
   "message": "操作成功",
   "code": 0,
   "data": {
     "orderTid": "customerCode-xxx",
     "itemList": [
       {
         "snCode": "8985xxxx",
         "productCode": "joy-Dxxxxx",
         "productExpireDate": "2023-12-30",
         "status": 1,
         "statusDesc": "成功"
       }
     ]
   }
 }
```

3 Response Code

Response Code	Response Message	Response Message (EN)
0	操作成功	Operation Success.
1	请求参数鉴权失败	Request Authentication Failed.
2	公共参数必填项未按要求填写	Mandatory Parameter Missing
3	渠道商余额不足	Insufficient Balance
4	不支持该业务（具体描述请参考message）	Service not supported. (Reference the actual response message)
5	该充值单已充值	Order Already Exists(check orderCode or orderTid to avoid duplicate)
6	订单项超过了限制（具体描述请参考message）	Order Item Exceeds limit (50)
-1	请求数据异常	Request Exceptions
100000	未知的服务异常	Unknown Service Exceptions

System 2: RSP Business Integration Interface Specification v1.2.2

1. Document Overview

1.1 Basic Information

This document mainly introduces the relevant content for integrating with RSP, including:

- Integration methods
- Security authentication
- Interaction process
- Interface list

1.2 Public Information

Information Name	Provider	Description
RSP Interface Base Path (BaseUrl)	RSP	Base path for RSP interfaces. Used for client invocation of RSP interfaces.
Server IP	Client	Used for RSP server whitelist verification.
Callback Notification Base Path (NotifyBaseUrl)	Client	Used for RSP server to send notifications to the client.
AppId	RSP	Client ID. Used as a request header parameter for the interface.
AppSecret	RSP	Encryption key. Used to encrypt request header parameters. Refer to the 'Security Authentication' section for specific encryption methods.

1.3 Security Authentication

To ensure the security of information transmission during interface calls, the following 5-layer security mechanisms are adopted:

1. RSP server interfaces use `HTTPS` secure connections.
2. Clients provide server IPs for RSP server whitelist verification.
3. Ensure the security of requests through MD5:
    - The requesting party provides 4 parameters in the request header: `AppId` / `TransId` / `Timestamp` / `Ciphertext`.
    - `Ciphertext` is the encrypted string, calculated as: `Ciphertext = MD5(AppId + TransId + Timestamp + AppSecret)`.
    - The responding party, upon receiving the request, extracts the 4 parameters from the request header, performs the same MD5 encryption, and confirms the request by checking whether the results match `Ciphertext`.
4. The request header parameter `TransId` (Transaction ID) must ensure uniqueness for each request, otherwise, the request fails.
5. The request header parameter `Timestamp` (Timestamp) must be within a 10-minute interval around the current time, otherwise, the request fails.

1.4 Interface Standards

- Request Protocol

  `HTTP`
- Request Methods

  `GET` / `POST`

- Data Transmission Format

  `application/json`

- Basic Request Header Parameters

Parameter Name	Mandatory	Parameter Type	Maximum Length	Description
AppId	Yes	String	32	Client ID. Provided uniformly by **RSP**.
TransId	Yes	String	50	Transaction ID. Ensure uniqueness for each request.
Timestamp	Yes	Long	20	Request timestamp. Example: `1653546537101`. Note: The request time must be within a 10-minute interval around the current time.
Ciphertext	Yes	String	32	Encryption string (case-insensitive). Formed by concatenating `AppId + TransId + Timestamp + AppSecret` and then performing `MD5` encryption.

- Basic Request Parameters

  None

- Basic Response Parameters

Parameter Name	Mandatory	Parameter Type	Maximum Length	Description
code	Yes	String	10	Result code. `000` represents success; refer to '3. Error Code Definition' for other codes.
mesg	No	String	500	Result message.
data	No	Object	-	Business data object. The object type varies according to the specific interface definition.

2. Interface List

2.1 eSIM Interface

2.1.1 Coupon Information Query

- Interface Description
  - Customers input coupon codes to query coupon information, with a maximum of 20 coupon codes per batch, separated by commas in English symbols.
  - RSP synchronously returns the result.
- URL

  `[BaseUrl]/openapi/coupon/query`

- Invocation Direction

  Customer -> **RSP**

- Request Parameters

Parameter	Required	Type	Max Length	Description
coupons	Yes	String	800	Coupon codes, separated by English commas.

- Response Parameters

  `data` is defined as follows:

Parameter	Required	Type	Max Length	Description
coupon	Yes	String	32	Coupon code.
couponStatus	Yes	Integer	2	Coupon status: 0 (Not activated); 1 (Unused); 2 (Used); 3 (Occupied); 4 (Invalidated); 5 (Expired); 6 (Invalid).

- Request Example

```
{
  "coupons": "j6TR54H1,j6TR54H2"
}
```

- Response Example

  Successful Request:

```
{
  "code": "000",
  "mesg": "success",
  "data": [{
    "coupon": "j6TR54H1",
    "couponStatus": 0
  },{
    "coupon": "j6TR54H2",
    "couponStatus": 1
  }]
}
```

  Request Error:

```
{
  "code": "999",
  "mesg": "System Error"
}
```

2.1.2 Coupon Redeem

- Interface Description
  - Customers input the coupon to initiate a redemption request.
  - RSP immediately responds upon receiving the request. Note: at this point, only the acknowledgment of receiving the request is synchronized, and the redemption result is not synchronized.
  - RSP asynchronously initiates the redemption process.

    - The redemption result will be notified to the customer through the interface `Coupon Redemption Result Notification`.
- URL

  `[BaseUrl]/openapi/coupon/redeem`
- Invocation Direction

  Customer -> **RSP**
- Request Parameters

Parameter	Required	Type	Max Length	Description
coupon	Yes	String	32	Coupon code.
qrcodeType	No	Integer	2	Returned QR code type. Default is `0`. `0` - QR code image URL, `1` - QR code content text. Note: After successfully redeeming, the interface `Coupon Redemption Result Notification` will return the QR code based on this field.

- Response Parameters

  `data` is not defined.
- Request Example

```
{
  "coupon": "j6TR54H1",
  "qrcodeType": 1
}
```

- Response Example

  Successful Request:

```
{
  "code": "000",
  "mesg": "success"
}
```

  Request Error:

```
{
  "code": "999",
  "mesg": "System Error"
}
```

2.1.3 Coupon Redeem Result Notification

- Interface Description
    - This interface is provided by the customer.

- After the coupon redemption process is completed, the URL of the QR code corresponding to the coupon will be sent to the customer through this interface.

- URL

  `[NotifyBaseUrl]/notify/coupon/redeem`

- Invocation Direction

  RSP -> **Customer**

- Request Parameters

Parameter	Required	Type	Max Length	Description
transId	Yes	String	50	`transId` passed in the redemption initiation interface.
resultCode	Yes	String	10	Result code. `000` represents success; refer to section 3 for other error code definitions.
resultMesg	No	String	500	Result message. If redemption fails, the failure reason is provided here.
finishTime	No	String	20	Completion timestamp, for example: `1653546537101`.
data	No	Object	-	Business data object `data`.

`data` is defined as follows:

Parameter	Required	Type	Max Length	Description
coupon	Yes	String	32	Coupon code.
qrcodeType	No	Integer	2	QR code type. `0` - QR code image URL, `1` - QR code content.
qrcode	No	String	500	QR code. The content type is determined by `qrcodeType`.
cid	No	String	20	CID, The Unique identification of eSIM
salePlanName	No	String	256	Sales product name.
salePlanDays	No	Integer	4	Sales product days.
pin1	No	String	4	PIN1
pin2	No	String	4	PIN2
puk1	No	String	8	PUK1
puk2	No	String	8	PUK2

- Response Parameters

  `data` is not defined.

- Request Example

  Successful Redemption (QR code type is `0`):

```
```

```
  {
    "transId": "2022031020001252",
    "resultCode": "000",
    "resultMesg": "success",
    "finishTime": "1653546537101",
    "data": {
      "coupon": "j6TR54H1",
      "pin1": "8568",
      "pin2": "",
      "puk1": "49065521",
      "puk2": "",
      "qrcodeType": 0,
      "qrcode": "https://xxx.xxx/qrcode.jpg",
      "salePlanName": "xxx",
      "salePlanDays": 90
    }
  }
```


Successful Redemption (QR code type is `1`):


```
{
    "transId": "2022031020001253",
    "resultCode": "000",
    "resultMesg": "success",
    "finishTime": "1653546537101",
    "data": {
      "coupon": "j6TR54H1",
      "pin1": "8568",
      "pin2": "",
      "puk1": "49065521",
      "puk2": "",
      "qrcodeType": 1,
      "qrcode": "LPA:1$rsp.demo.com$0913F6176020B7C603E3R42B61P686D3",
      "salePlanName": "xxx",
      "salePlanDays": 90
    }
  }
```


Redemption Failure:


```
{
  "transId": "2022031020001254",
  "resultCode": "600",
  "resultMesg": "业务处理失败",
  "finishTime": "1653546537101"
}
```


- Response Example

  Successful Request:

```
{
  "code": "000",
  "mesg": "success"
}
```

Request Error:

```
{
  "code": "999",
  "mesg": "System Error"
}
```

2.1.4 Get Transaction Status

- **Interface Description:**
  - Customers input the transaction ID information they want to query, initiating a transaction status query request. Only supports querying transaction status within the last `3 months`.
  - RSP synchronously returns the result.
- **URL:**

```
[BaseUrl]/openapi/getTransactionStatus
```

- **Invocation Direction:**
  Customer -> **RSP**
- **Request Parameters:**

Parameter	Required	Type	Max Length	Description
qTransId	Yes	String	50	Transaction ID to be queried.

- **Response Parameters:**

Parameter	Required	Type	Max Length	Description
qTransId	Yes	String	50	Transaction ID to be queried.
resultCode	Yes	String	10	Result code. `000` represents success; refer to section 3 for other error code definitions.
resultMesg	No	String	500	Result message. If the transaction fails, the failure reason is provided here.
createTime	No	String	20	Creation timestamp, for example: `1653546536000`.
finishTime	No	String	20	Completion timestamp, for example: `1653546537101`.
data	No	Object	-	Business data object `data`, currently only supports coupon redemption transactions.

For coupon redemption, `data` is defined as follows:

Parameter	Required	Type	Max Length	Description
coupon	Yes	String	32	Coupon code.
qrcodeType	No	Integer	2	QR code type. `0` - QR code image URL, `1` - QR code content.
qrcode	No	String	500	QR code. The content type is determined by `qrcodeType`.
cid	No	String	20	CID, The Unique identification of eSIM
salePlanName	No	String	256	Sales product name.
salePlanDays	No	Integer	4	Sales product days.
pin1	No	String	4	PIN1
pin2	No	String	4	PIN2
puk1	No	String	8	PUK1
puk2	No	String	8	PUK2

- **Request Example:**

```
{
  "qTransId":"2022031020001252"
}
```

- **Response Example:**
  Successful Request:

```
{
  "qTransId": "2022031020001252",
  "resultCode": "000",
  "resultMesg": "success",
  "finishTime": "1653546537101",
  "data": {
    "coupon": "j6TR54H1",
    "qrcodeType": 0,
    "qrcode": "https://xxx.xxx/qrcode.jpg",
    "salePlanName": "xxx",
    "salePlanDays": 90
  }
}
```

  Request Error:

```
{
  "qTransId": "2022031020001252",
  "resultCode": "600",
  "resultMesg": "业务处理失败",
  "finishTime": "1653546537101"
}
```

2.1.5 eSIM Usage Query

- **Interface Description:**
  - Customers input eSIM coupon to query eSIM usage information.
  - RSP synchronously returns the result.
- **URL:**

```
[BaseUrl]/openapi/esim/usage/query
```

- **Invocation Direction:**
  Customer -> **RSP**
- **Request Parameters:**

Parameter	Required	Type	Max Length	Description
coupon	Yes	String	32	Coupon

- **Response Parameters:**
  `data` is defined as follows:

Parameter	Required	Type	Max Length	Description
dataUsageList	Yes	Object	-	List of usage data.
usageDate	Yes	String	8	Usage date, format: "yyyymmdd", e.g., 20230526.
mcc	Yes	String	3	MCC code for usage region, e.g., 454.
usage	Yes	String	15	Data usage amount, unit: byte.

- **Request Example:**

```
{
  "coupon": "j6TR54H1"
}
```

- **Response Example:**
  Successful Request:

```
{
  "code": "000",
  "mesg": "success",
  "data": {
```

```
    "dataUsageList": [{
      "usageDate": "20230526",
      "mcc": "454",
      "usage": "8102492"
    },{
      "usageDate": "20230525",
      "mcc": "454",
      "usage": "3145728"
    }]
  }
}
```

Request Error:

```
{
  "code": "999",
  "mesg": "System Error"
}
```

2.1.6 eSIM Status Query

- **Interface Description:**
  - Customers input eSIM coupon to query eSIM status.
  - RSP synchronously returns the result.
- **URL:**

```
[BaseUrl]/openapi/esim/status/query
```

- **Invocation Direction:**
  Customer -> **RSP**
- **Request Parameters:**

Parameter	Required	Type	Max Length	Description
coupon	Yes	String	32	Coupon

- **Response Parameters:**
  `data` is defined as follows:

Parameter	Required	Type	Max Length	Description
status	Yes	String	1	eSIM status: 0(Unknown); 1(Activated); 2(Expired).
statusTime	Yes	String	20	eSIM status change timestamp, e.g., `1653546537101`.

- **Request Example:**

```
{
  "coupon": "j6TR54H1"
}
```

- **Response Example:**
  Successful Request:

```
{
  "code": "000",
  "mesg": "success",
  "data": {
    "status": "1",
    "statusTime": "1653546537101"
  }
}
```

  Request Error:

```
{
  "code": "999",
  "mesg": "System Error"
}
```

2.1.7 eSIM Installation Event Notification

- Interface Description
  - Interface is provided by Merchant
  - eSIM installation process event notification
- URL

  `[NotifyBaseUrl]/notify/esim/esim-progress`
- Call Direction

  RSP -> **Merchant Server**
- Request Parameter

Parameter Name	Mandatory	Parameter Type	Max Length	Description
transId	Yes	String	50	Request `transId`
resultCode	Yes	String	10	Result code. `000`-Success, others for Failure. Refer to `3. Error Code Definition`
resultMesg	No	String	500	Result Message
finishTime	No	String	20	Finish timestamp, Example: `1653546537101`
data	No	Object	-	Business data object `data`

`data` define:

Parameter Name	Mandatory	Parameter Type	Max Length	Description
cid	Yes	String	20	The Unique identification of eSIM
eid	No	String	40	Device ID
profileType	Yes	String	128	The profile type name to which the profile belongs
timestamp	Yes	String	20	The timestamp of the request execution, for example: `1653546537101`
notificationPointId	Yes	Integer	4	Notification type: 1(ELIGIBILITY_AND_RETRY_LIMIT_CHECK); 2(CONFIRMATION_FAILURE); 3(BPP_DOWNLOAD); 4(BPP_INSTALL_NOTIFICATION); 5(DELETE_NOTIFICATION); 6(ENABLE_NOTIFICATION); 7(DISABLE_NOTIFICATION); 101(EID_BLOCKED); 102(TAC_BLOCKED);
notificationPointStatus	Yes	Object	-	Execution Status
resultData	Yes	String	1024	Final result of profile installation returned by Euicc

`notificationPointStatus` define:

Parameter Name	Mandatory	Parameter Type	Max Length	Description
status	Yes	String	128	"Executed-Success"(Execution succeeded) "Failed"(Execution failed) "Executed-Withwarning"(Execution completed (warning status)) "Expired"(Expired)
statusCodeData	Yes	Object	-	SGP standard definition format exists when request execution fails

`statusCodeData` define:

Parameter Name	Mandatory	Parameter Type	Max Length	Description
subjectCode	No	String	100	Object number
reasonCode	No	String	100	Reason number
message	No	String	1024	Detailed error information
subjectIdentifier	No	String	100	Object identification

- Response Parameter

  `data` undefined

- Request Example

```
{
  "transId": "2022031020001252",
  "resultCode": "000",
  "resultMesg": "success",
  "finishTime": "1653546537101",
  "data": {
```

```
      "cid": "89852000010000000001",
      "eid": "xxx",
      "profileType": "xxx",
      "timestamp": "2022-02-22T18:21:47Z",
      "notificationPointId": 1,
      "resultData": "xxx",
      "notificationPointStatus": {
        "status": "Executed-Success",
        "statusCodeData": {
          "subjectCode": "xxx",
          "reasonCode": "xxx",
          "message": "xxx",
          "subjectIdentifier": "xxx"
        }
      }
    }
  }
```


- Response Example

  Request Correct:


```
{
  "code": "000",
  "mesg": "success"
}
```


  Request Error:


```
{
  "code": "999",
  "mesg": "System Error"
}
```


2.1.8 eSIM Profile Information Query

- Interface Description
  - Merchant input eSIM Profile CID to query eSIM Profile information
  - RSP synchronize return result
- URL

  `[BaseUrl]/openapi/esim/profile/query`
- Call Direction

  Merchant Server -> **RSP**
- Request Parameter

Parameter Name	Mandatory	Parameter Type	Max Length	Description
cid	Yes	String	20	eSIM Profile CID whitch is the Unique identification of eSIM

- Response Parameter

  `data` define：

Parameter Name	Mandatory	Parameter Type	Max Length	Description
state	Yes	String	32	The current status of the profile： AVAILABLE; ALLOCATED; LINKED; CONFIRMED; RELEASED; DOWNLOADED; INSTALLED; ERROR; ENABLED; DISABLED; DELETED; UNAVAILABLE;
profileType	No	String	64	The Profile Type to which the Profile belongs
eid	No	String	128	If the profile has been bound to EID and the order has not been deleted, then return

- Request Example

```
{
  "cid": "89852000010000000000"
}
```

- Response Example

  Request Correct:

```
{
  "code": "000",
  "mesg": "success",
  "data": {
    "state": "AVAILABLE",
    "profileType": "Profile_Type_Common",
    "eid": "89000012345678900000000123456789"
  }
}
```

  Request Error:

```
{
  "code": "999",
  "mesg": "System Error"
}
```

2.2 OTA Card Interface

2.2.1 OTA Card Status Query

- **Interface Description:**
  - Customers input the card CID to query the card status.
  - RSP synchronously returns the result.
- **URL:**

```
[BaseUrl]/openapi/sim/status/query
```

- **Invocation Direction:**
  Customer -> **RSP**
- **Request Parameters:**

Parameter	Required	Type	Max Length	Description
cid	Yes	String	20	CID. The code printed on the OTA SIM card board.

- **Response Parameters:**
  `data` is defined as follows:

Parameter	Required	Type	Max Length	Description
simStatus	Yes	String	6	Card status: 00(Inventory); 10(Not activated); 11(Activated); 12(Recharge period); 13(Expired); 98(Invalidated).
simStatusTime	Yes	String	20	Timestamp of card status change, for example: `1653546537101`.

- **Request Example:**

```
{
  "cid": "89851100000000000001"
}
```

- **Response Example:**
  Successful Request:

```
{
  "code": "000",
  "mesg": "success",
  "data": {
    "simStatus": "10",
    "simStatusTime": "1653546537101"
  }
}
```

Request Error:

```
{
  "code": "999",
  "mesg": "System Error"
}
```

2.2.2 OTA Card Usage Query

- **Interface Description:**
  - Customers input the card CID and service order ID to query card usage information.
  - RSP synchronously returns the result.
- **URL:**

```
[BaseUrl]/openapi/sim/usage/query
```

- **Invocation Direction:**
  Customer -> **RSP**
- **Request Parameters:**

Parameter	Required	Type	Max Length	Description
cid	Yes	String	20	CID. The code printed on the OTA SIM card board.
orderId	Yes	String	32	The **rspOrderId** returned in the `Recharge Order Status Inquiry` interface.
imsi	No	String	15	Resource IMSI.

- **Response Parameters:**
  `data` is defined as follows:

Parameter	Required	Type	Max Length	Description
effTime	Yes	String	20	Effective timestamp, for example: `1685003072000`.
expTime	Yes	String	20	Expiry timestamp, for example: `1685116799000`.
totalUsage	Yes	String	15	Total data usage, unit: byte.
dataUsageList	Yes	Object	-	List of usage data.

`dataUsageList` is defined as follows:

Parameter	Required	Type	Max Length	Description
usageDate	Yes	String	8	Usage date, format: "yyyymmdd", e.g., 20230526.
mcc	Yes	String	3	MCC code for usage region, e.g., 454.
usage	Yes	String	15	Data usage amount, unit: byte.

- **Request Example:**

```
{
  "cid": "89851100000000000001",
  "orderId": "20220522600000001",
  "imsi": "454001234567891"
}
```

- **Response Example:**
  Successful Request:

```
{
  "code": "000",
  "mesg": "success",
  "data": {
    "effTime": "1685003072000",
    "expTime": "1685116799000",
    "totalUsage": "11248220",
    "dataUsageList": [{
      "usageDate": "20230526",
      "mcc": "454",
      "usage": "8102492"
    },{
      "usageDate": "20230525",
      "mcc": "454",
      "usage": "3145728"
    }]
  }
}
```

  Request Error:

```
{
  "code": "999",
  "mesg": "System Error"
}
```

---

3. Error Code Definition

3.1 Response Code Definition

Error Code	Description
000	Success
300	Business processing in progress
401	Invalid customer ID
403	Encryption message verification failed
406	No interface permissions
407	IP not in whitelist
500	Parameter exception
501	Duplicate transaction ID
502	Invalid request time
600	Business processing failed
601	Card does not exist
602	Coupon does not exist
603	Insufficient resource inventory
999	System exception