Partner API Documentation

Section: Deposit Inquiry

Version: 1.0

CONFIDENTIAL DOCUMENT

Version Control

Revision	Issue Date	Description	Review
1.0	27 Feb 2022	Internal release	

Table of Contents

1. Introduction

Partner API endpoints are built around REST web service technology over HTTP. Our Partner API request and response payload are using JSON format as its standard communication format.

2. Security

Upon initial registration, every partner will be provided two sets of keys: API Key (AK) and Private Key (PK) to be used for authentication and decryption method.

API Key will be transacted on every request, while Private Key must be kept secret on partner side for decryption purposes.

2.1. Encrypted Content

Some of the content in response payload might be encrypted from our server, and partner must perform decryption in order to obtain the real content. All parts that require decryption (if any) will be mentioned clearly on each chapter.

2.1.1. Decryption Method

AES (Advanced Encryption Standard) algorithm is used to decrypt the encrypted content. To be exact, we are using **AES-256** with **ECB** block cipher mode and **PKCS#5** padding. AES-256 requires key size of 256 bits, so a key with length of 32 characters must be generated.

Steps to generate key:

1. Generate SHA-512 hash from Private Key (PK). This will generate hash with size of 64 bytes.
2. Take 16 bytes out of 64 bytes generated on Step 1 by copying byte 17 to 32.
3. Convert 16 bytes from [Step 2] to its hex representation in string, which will result in 32 characters. (All lowercase and no spaces)
4. Key is ready to use.

After the 32-characters key is generated, it can be used to perform AES-256 decryption on the encrypted content (Base64) to obtain the real content.

Example of Decryption

API Key	`358012bfe06a4145bc839e738107c76c`
Private Key	`db6996f16ffc45eaa14eac0834df4aa6`
Encrypted Content (Base64)	`P9qM+gAafnKdXqioeHokterKqpSJKZJng+mXRpYaoGc=`

Step 1, 2, 3: Generate SHA-512 from Private Key, take byte 17-32 from result of SHA-512, and then convert to hex string representation with all lowercase and no spaces. The key will be:

```
837de83c222255018642a2e4d9127556
```

Perform AES-256 decryption with above key on the encrypted content. Using ECB mode and PKCS#5 padding. The sample result should be:

```
This is a sample plain content.
```

Please note. Please use above sample to test the correctness of your algorithm.

3. Token Request

This endpoint is used to obtain the token for using Partner API endpoints.

This token will be valid for the next 24 hours. However, this token will be invalidated if there is a new token request with the same API Key. There is only a maximum of 1 (one) active token at any given time for each partner.

Method	GET
Path	`/api/external/v1/inquiry/token`
Request Body	-
Response Body	application/json

URL Parameters

Paramater	Data type	Mandatory	Description
apikey	String	Yes	Partner API Key

Sample URL

URL	Description
`/api/external/v1/inquiry/token?apikey=358012bfe06a4145bc839e738107c76c`	Default

Request Header

Header Name	Data type	Mandatory	Description
Accept	String	Yes	application/json

Request Body

N/A

Response Body

Field Name	Data type	Mandatory	Description
responseCode	String	Yes	Response code
responseMessage	String	Yes	Response message
responseTime	String	No	yyyy-MM-dd HH:mm:ss
externalId	String	No	External ID from Partner API system for reference purpose
token	String	No	Encrypted token. To obtain the real token, perform decryption explained in Chapter 2.1
expiredTime	String	No	yyyy-MM-dd HH:mm:ss Token expiry time

Sample Response


```
{
  "responseCode": "1000",
  "responseMessage": "Success",
  "responseTime": "2021-11-22 23:18:00",
  "externalId": "CPM1634885045463",
  "token": "t5GsJCUgAHzvBySLXfxxsRyyq0BrBVozd+VKdDXtYlFodIdi360wCuMHASeXKTcP",
  "expiredTime": "2021-11-23 23:17:58"
}
```


Decrypt Token

Using above sample response, the content of field "token" is encrypted. And hence, decryption must be performed to obtain the real token (refer to Chapter 2.1).

In this case, if we decrypt it using the key on Chapter 2.1.1, we will obtain the real token: **abcdefghijklmnopqrstuvwxyz123456**.

Please use this sample to ensure the decryption algorithm is correct.

4. Get Deposit Details

This endpoint is used to get deposit details of a specific machine within a specified time period.

Method	POST
Path	`/api/external/v1/inquiry/deposit/details`
Request Body	application/json
Response Body	application/json

URL Parameters

Paramater	Data type	Mandatory	Description
apikey	String	Yes	Partner API Key

Sample URL

URL	Description
`/api/external/v1/inquiry/deposit/details?apikey=358012bfe06a4145bc839e738107c76c`	Default

Request Header

Header Name	Data type	Mandatory	Description
Content-Type	String	Yes	application/json
Accept	String	Yes	application/json

Request Body

Field Name	Data type	Mandatory	Description
token	String	Yes	Active token, obtained from Token Request endpoint.
machineSn	String	No	Depos machine serial number. Either this or machineCode must be defined.

machineCode	String	No	Machine code. For some partners, this can be filled with outlet code. Either this or machineSn must be defined.
rangeStart	String	Yes	yyyy-MM-dd Start of day-end date range.
rangeEnd	String	Yes	yyyy-MM-dd End of day-end date range
refId	String	Yes	Partner reference ID. It is recommended to be unique per request for easier tracing in case of problem. Partner API will only store it as reference and will not perform any validation.

Response Body

Field Name	Data type	Description
responseCode	String	Response code
responseMessage	String	Response message
responseTime	String	yyyy-MM-dd HH:mm:ss
externalId	String	External ID from Partner API system for reference purpose
deposits	Array	List of deposit transactions
id	Long	Transaction ID
machineSn	String	Depos machine serial number (optional)
machineCode	String	Machine code (optional)
depositCode	String	Deposit code, this is printed on receipt
depositTime	String	yyyy-MM-dd HH:mm:ss Deposit time
totalValue	Long	Total deposit value

username	String	User login name
info	String	Deposit information
denoms	Array	List of denomination breakdown
denom	String	Denomination name
qty	Integer	Quantity

Sample Request

```
{
  "token": "abcdefghijklmnopqrstuvwxyz123456",
  "machineCode": "77-000-69",
  "rangeStart": "2021-11-25",
  "rangeEnd": "2021-11-25",
  "refId": "REF3a2s1d5099"
}
```

Sample Response

```
{
  "responseCode": "1000",
  "responseMessage": "Success",
  "responseTime": "2021-11-26 23:38:59",
  "externalId": "CPM1634885066720",
  "deposits": [
    {
      "id": 78667,
      "machineSn": "Kun520A20025",
      "machineCode": "77-000-69",
      "depositCode": "Kun520A20025211125D001",
      "depositTime": "2021-11-25 09:36:26",
      "totalValue": 88500000,
      "username": "timburton",
      "info": "",
      "denoms": [
        {
          "denom": "100000",
          "qty": 237
        },
        {
          "denom": "50000",
          "qty": 212
        },
        ...TRUNCATED...
      ]
    },
    ...TRUNCATED...
  ]
}
```

Appendix A: Response Code Table

Code	Message
1000	Success
9000	General Error
9101	Invalid API Key
9102	Invalid request body
9103	Invalid token