**Story #1 : Create Wallet Account**

**Requirement**

A customer enters required personal information enough to create a wallet

**Input specification**

| **Field** | **Type** | **Length** | **Optional** | **Remark** |
| --- | --- | --- | --- | --- |
| Full Name | Text | 50 |  | English Character only + “,” + “-“ + “.” + “ “ |
| Citizen ID | Numeric | 13 |  | Must be 13 digits, validation by rule #1 |

Create a wallet account with the following parameters

| **Fields** | **Format** | **Length** | **Value** |
| --- | --- | --- | --- |
| wallet\_id | Numeric | 12 | System generated by rule #2 |
| citizen\_id | Numeric | 13 |  |
| full\_name | Text | 50 | Store in UPPERCASE |
| open\_datetime | Datetime | ISO8601 format, GMT. +7 | At least - second |
| ledger\_balance | Numeric | 18,2 | Set balance = 0.00 after opening |

*Rules*

1. Citizen ID validation
   1. Check duplicate. 1 citizen id must have only 1 wallet
2. wallet\_id generation logic
   1. 12 digit (D1D2D3…D12)
      1. Start with “1”
      2. Follow by 10 digit running number
      3. Last digit is check digit = ( ∑ from I = 2 to 11(Di \* i) ) mod 10

*Error code / desc*

* 001 : Incorrect Citizen ID
* 002 : Duplicate Citizen ID
* 003 : Incorrect Name

**API specification**

**Description Create Wallet Account**

**Method POST**

**Path v1/accounts/**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Request** | **JSON Object** | **Key** | **Type** | **Length** | **Optional** | **Description** |
| **HTTP Header** |  | x-request-id | String |  | Y | Request ID, Generated from external domain |
|  |  | x-real-ip | String |  | Y | IP of the caller |
|  |  | x-application | String |  |  | Name of the caller (domain name) |
|  |  | user-agent | String |  |  | Browser agent/mobile/etc (HTTP standard header) |
|  |  | datetime | String |  |  | Request time |
|  |  | accept-language | String |  | Y | Language of the needed response |
|  |  | accept-encoding | String |  | Y | Encoding of the message response |
|  |  | x-job-id | String |  | Y | Job unique id, Generate by API gateway |
|  |  | x-caller-service | String |  |  | Caller Service |
|  |  | x-caller-domain | String |  |  | Caller Domain |
|  |  | accept | String |  |  | Example “application/json” |
|  |  | x-device | String |  | Y | Client Device |
|  |  | x-channel | String |  | Y | Channel used to sent the request |
| Message body | rqBody | Full Name | String |  |  |  |
|  |  | Citizen ID | Numeric |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Response** | **JSON Object** | **Key** | **Type** | **Length** | **Optional** | **Description** |
| Status Code |  | 200 : Success 400 : Error exist | | | | |
| **HTTP Header** |  | x-request-id | String |  |  | Generated with RFC4122 standard by target service if blank, if not blank, use existing |
|  |  | datetime | ISO-8601 |  |  | Date and time of response |
|  |  | content-type | String |  |  | Content type |
|  |  | x-roundtrip | String |  |  | A number of milliseconds to process the request |
|  |  | x-job-id | String |  |  | Job unique id |
|  |  |  |  |  |  |  |
| **Message body** | **rsBody** | wallet\_id | Numeric |  |  |  |
|  |  | open\_datetime | Datetime |  |  |  |
|  | **error** | error | List |  | Y | List of errors |
|  |  | error/errorCode | String |  |  | Error code |
|  |  | error/errorDesc | String |  |  | Error description |
|  |  |  |  |  |  |  |

**Story #2 : Wallet Account inquiry**

**Requirement**

A customer can get the information about their wallet

Query Parameter

* wallet\_id
* citizen\_id
* full\_name (partial match, equal to “LIKE” command)

Return all wallet data

*Error code / desc*

* 004 : Data not existed (note : for query by wallet\_id and citizen\_id)

**API specification**

**Description Inquiry Wallet Account**

**Method GET**

**Path v1/accounts/{wallet\_id}**

**Path v1/accounts?citizen\_id={query value}**

**Path v1/accounts/search?full\_name ={query value}**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Request** | **JSON Object** | **Key** | **Type** | **Length** | **Optional** | **Description** |
| **HTTP Header** |  | x-request-id | String |  | Y | Request ID, Generated from external domain |
|  |  | x-real-ip | String |  | Y | IP of the caller |
|  |  | x-application | String |  |  | Name of the caller (domain name) |
|  |  | user-agent | String |  |  | Browser agent/mobile/etc (HTTP standard header) |
|  |  | datetime | String |  |  | Request time |
|  |  | accept-language | String |  | Y | Language of the needed response |
|  |  | accept-encoding | String |  | Y | Encoding of the message response |
|  |  | x-job-id | String |  | Y | Job unique id, Generate by API gateway |
|  |  | x-caller-service | String |  |  | Caller Service |
|  |  | x-caller-domain | String |  |  | Caller Domain |
|  |  | accept | String |  |  | Example “application/json” |
|  |  | x-device | String |  | Y | Client Device |
|  |  | x-channel | String |  | Y | Channel used to sent the request |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Response** |  | **Key** | **Type** | **Length** | **Optional** | **Description** |
| Status Code |  | 200 : Success 400 : Error exist | | | | |
| **HTTP Header** |  | x-request-id | String |  |  | Generated with RFC4122 standard by target service if blank, if not blank, use existing |
|  |  | datetime | ISO-8601 |  |  | Date and time of response |
|  |  | content-type | String |  |  | Content type |
|  |  | x-roundtrip | String |  |  | A number of milliseconds to process the request |
|  |  | x-job-id | String |  |  | Job unique id |
|  |  |  |  |  |  |  |
| **Message body** | **rsBody** | wallet\_id | Numeric |  |  |  |
|  |  | citizen\_id | Numeric |  |  |  |
|  |  | full\_name | Text |  |  |  |
|  |  | open\_datetime | Datetime |  |  |  |
|  |  | ledger\_balance | Numeric | 18,2 |  |  |
|  | **error** | error | List |  | Y | List of errors |
|  |  | error/errorCode | String |  |  | Error code |
|  |  | error/errorDesc | String |  |  | Error description |
|  |  |  |  |  |  |  |

**Technical Work**

* Application in GO language
  + Basic syntax
  + How to create an API server
    - Run as a HTTP server
    - JSON parsing
    - Error code
* Postman (for test API) or you can use SOAPUI
* SWC : Github
  + <https://blog.nextzy.me/มาเรียนรู้-git-แบบง่ายๆกันเถอะ-427398e62f82>
* Database : MongoBD
  + Install database
  + Load Go library to connect to MongoDB
  + Learn how to
    - Create table (collection)
    - Create and Query data