

QA Automation Code Challenge - API Test Framework Setup and Test Case Implementation

The company has developed a new API endpoint to create beneficiary when transfer moneys.

As an Automation Engineer/QA, you are required to develop an API Framework and implement test cases to cover this new endpoint.

Tasks and requirement:

- Analysis the business requirement (attached)
- Setup API Test framework using any language you are familiar with
- Write automated api tests to verify the new endpoint is working as expected
- Focus on functional testing only, both positive and negative tests
- Document any defects/issues found, and include all necessary info for Dev to reproduce and fix the issue (in Word or other type of file you think appropriate)
- Submit your project via Github or zipped file and include README file for setup and executing tests
- Tests should be able to run from command line, so it's easy for CI integration
- The tests will run in different environments, so make the endpoint url configurable, make sure include instruction in README for the usage

Business requirement for new endpoint: api/v1/beneficiaries/create

Airwallex needs an API service endpoint to create beneficiary. Here is the API reference:

https://www.airwallex.com/docs/api/#Payouts/Beneficiaries/_api_v1_beneficiaries_create/post

When Creating beneficiary, there is a field for bank detail. Please design your test framework to cover this part.

Payload fields:

- payment_method: mandatory, value should be either LOCAL or SWIFT
- bank_country_code: mandatory, can be one of US, AU, CN
- account name: mandatory, any character, length from 2 to 10
- account number: mandatory
 - for US, account number is 1-17 character long, can be any character
 - for AU, account number is 6-9 character long, can be any character
 - for CN, account number is 8-20 character long, can be any character
- swift_code:
 - mandatory when payment method is SWIFT
 - the 5th and 6th character of swift code should match the bank country code, e.g. ICBKCNBJ is a valid swift code for **CN**
 - swift code should be either 8 or 11 characters
- bsb:
 - mandatory when bank country is AU
 - 6 characters
- aba:
 - mandatory when bank country is US
 - 9 characters

If the request passes the field validation, 200 response will returned.

If the request failed the validation, we need to return 400 http response with proper error message, some sample messages as below:

- when account_number not given: "'account_number' is required"
- account_number length error: "Length of account_number should be between 7 and 11 when bank_country_code is 'US'"
- wrong swift code: "The swift code is not valid for the given bank country code: US"
- swift code length error: "Length of 'swift_code' should be either 8 or 11"

Happy path to test the endpoint:

1. First, you need to get authorization by call following request:

```
curl --location --request POST 'https://api-demo.airwallex.com/api/v1/authentication/login' \  
--header 'Content-Type: application/json' \  
--header 'x-client-id: KJbr_Xs5TLmhY03JWSi3NQ' \  
--header 'x-api-key: 2308e2dd300f6b959b7f4e0a52ba9181186fc92f075f7e64ee9fa0b6b1ada094c39c9a9f39f06693d06b17067f78d4e7' \  
--data-raw ''
```

2. Create beneficiary:

```
curl --location --request POST 'https://api-  
demo.airwallex.com/api/v1/beneficiaries/create' \  
--header 'Authorization: Bearer {{$token_from_step1}}' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "beneficiary": {  
    "additional_info": {  
      "personal_email": "john.walker@gmail.com"  
    },  
    "address": {  
      "city": "Seattle",  
      "country_code": "US",  
      "postcode": "98104",  
      "state": "Washington",  
      "street_address": "412 5th Avenue"  
    },  
    "bank_details": {  
      "account_currency": "USD",  
      "account_name": "John Walker",  
      "account_number": "50001121",  
      "account_routing_type1": "aba",  
      "bank_country_code": "US",  
      "bank_name": "JP Morgan Chase Bank",  
      "swift_code": "CHASUS33"  
    },  
    "company_name": "ABC University",  
    "entity_type": "COMPANY"  
  },  
  "nickname": "ABC University",  
  "payer_entity_type": "PERSONAL",  
  "payment_methods": [  
    "SWIFT"  
  ]  
}
```

```
"account_number": "123",  
"swift_code": "ICBCUSBJ",  
"aba": "11122233A"  
}
```

■ sample 201 response:

```
{  
  "beneficiary": {  
    "additional_info": {  
      "personal_email": "john.walker@gmail.com"  
    },  
    "address": {  
      "city": "Seattle",  
      "country_code": "US",  
      "postcode": "98104",  
      "state": "Washington",  
      "street_address": "412 5th Avenue"  
    },  
    "bank_details": {  
      "account_currency": "USD",  
      "account_name": "John Walker",  
      "account_number": "50001121",  
      "account_routing_type1": "aba",  
      "bank_country_code": "US",  
      "bank_name": "JP Morgan Chase Bank",  
      "swift_code": "CHASUS33"  
    },  
    "company_name": "ABC University",  
    "entity_type": "COMPANY"  
  },  
  "beneficiary_id": "9adec7d3-9e9e-42e1-9e10-370ea7cc83e5",  
  "nickname": "ABC University",  
  "payer_entity_type": "PERSONAL",  
  "payment_methods": [  
    "SWIFT"  
  ]  
}
```