UTM UNIVERSITI TEKNOLOGI MALAYSIA SCHOOL OF COMPUTING Faculty of Engineering

SECP3843-01: Special Topic in Data Engineering

SEMESTER 2 2022/2023

Assignment 1

Bank Negara Open API

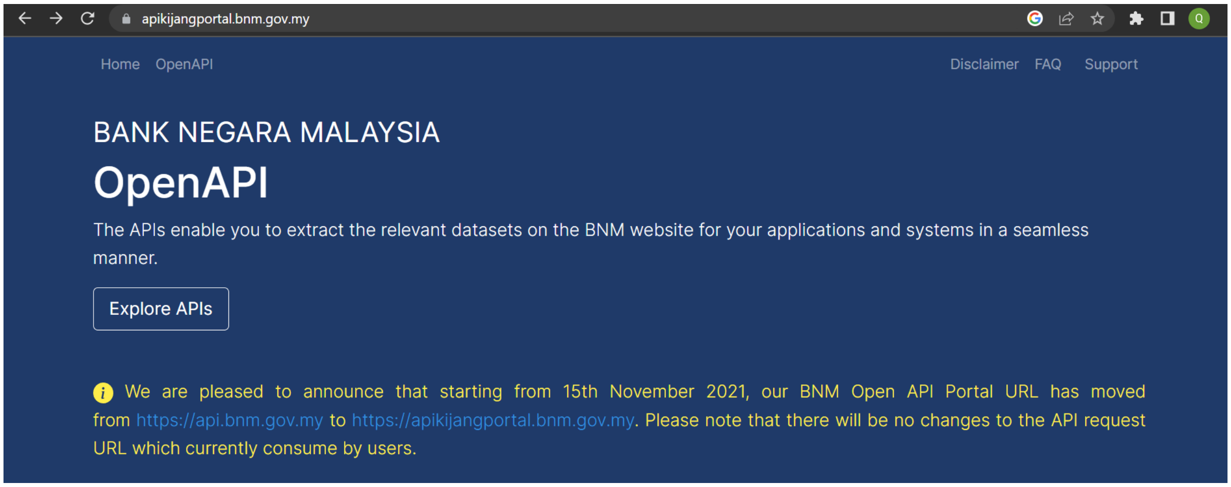
Prepared by: Group DataSphere

NO	NAME	MATRIC NO
1.	Terence A/L Loorthanathan	A20EC0165
2.	Rishma Fathima binti Basher	A20EC0137
3.	Nur Syamalia Faiqah binti Mohd Kamal	A20EC0118
4.	Nurfarrahin binti Che Alias	A20EC0121
5.	Qaisara Binti Rohzan	A20EC0133

Table of Contents

1. Get Bank Negara Open API

To retrieve the URL, simply go to https://apikijangportal.bnm.gov.my/openapi and click on **‘Explore APIs’**.



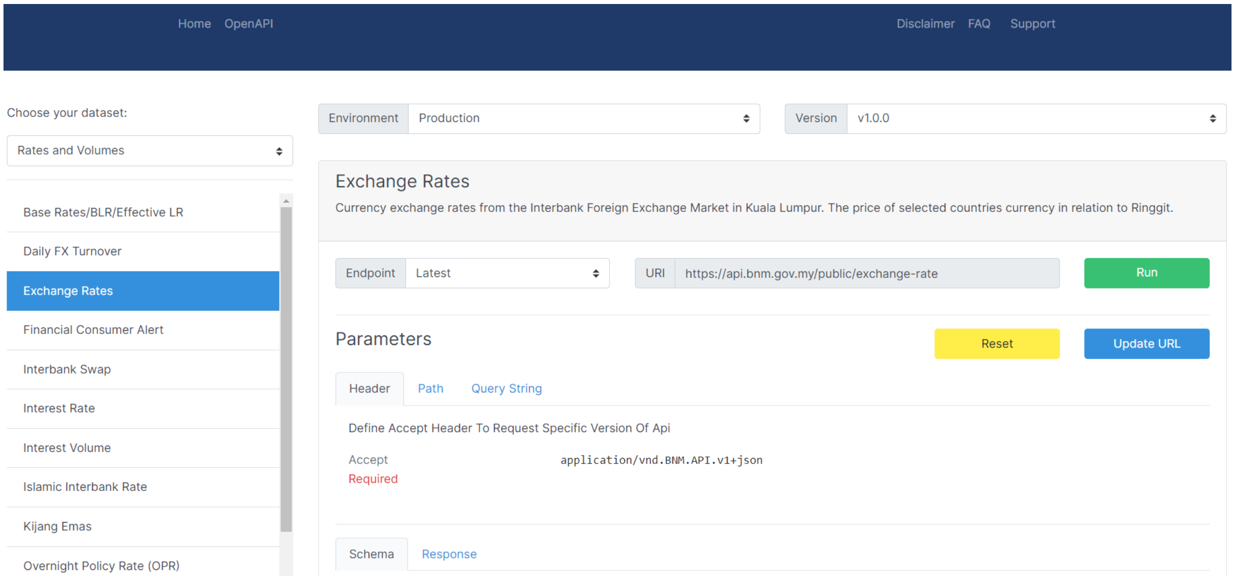
BANK NEGARA MALAYSIA

OpenAPI

The APIs enable you to extract the relevant datasets on the BNM website for your applications and systems in a seamless manner.

We are pleased to announce that starting from 15th November 2021, our BNM Open API Portal URL has moved from https://api.bnm.gov.my to https://apikijangportal.bnm.gov.my. Please note that there will be no changes to the API request URL which currently consume by users.

For this project, our group chose the ‘Exchange Rates’ option in the dataset pane. This option allows us to access the currency exchange rates from the Interbank Foreign Exchange Market in Kuala Lumpur. Copy the URL and implement it in the python script as shown below.



Choose your dataset: Rates and Volumes

Environment: Production

Version: v1.0.0

Exchange Rates

Currency exchange rates from the Interbank Foreign Exchange Market in Kuala Lumpur. The price of selected countries currency in relation to Ringgit.

Endpoint: Latest

URI: https://api.bnm.gov.my/public/exchange-rate

Parameters

Define Accept Header To Request Specific Version Of Api

Accept
Required: `application/vnd.BNM.API.v1+json`

2. Create MongoDB Database and Collection

```
# MongoDB database credentials
mongodb_uri =
'mongodb+srv://terence:qCZgfWgGHCBSqCqk@learningcluster.p8bbacm
.mongodb.net/test'
mongodb_database = 'your_mongodb_database_name'
mongodb_collection = 'your_mongodb_collection_name'
```

Explanation:

When connecting to a MongoDB instance, we need to provide the credentials using the connection string. The connection string has the following format:

Here's a breakdown of each component of the connection string:

- mongodb://: The protocol used to connect to the MongoDB instance.
- /database: The name of the database to connect to. If the database does not exist, MongoDB will create it when you first write data to it.
- /collection: The name of the collection to connect to.

3. Retrieve The Bank Negara Open API Response

```python
import requests

# Replace <YOUR_API_KEY> with your actual API key
# API endpoint URL
url = 'https://api.bnm.gov.my/public/exchange-rate'

# Parameters for API request
params = {
    'effective_date': 'latest'
}

# Set headers with your API key
headers = {
    'Accept': 'application/vnd.BNM.API.v1+json'
}

# Send API request
response = requests.get(url, params=params, headers=headers)

# Print response data
print(response.json())

# Save it to a variable
data = response.json()
```

Below is the data output retrieved:

```
{'data': [{'currency_code': 'CHF', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 4.8667, 'selling_rate': 4.8717, 'middle_rate': 4.8692}},
{'currency_code': 'CAD', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 3.2566, 'selling_rate': 3.26, 'middle_rate': 3.2583}},
{'currency_code': 'BND', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 3.307, 'selling_rate': 3.3113, 'middle_rate': 3.3092}},
{'currency_code': 'AUD', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 2.937, 'selling_rate': 2.9423, 'middle_rate': 2.9397}},
{'currency_code': 'AED', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 119.7866, 'selling_rate': 119.9281, 'middle_rate': 119.8573}},
{'currency_code': 'CNY', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 0.64, 'selling_rate': 0.6406, 'middle_rate': 0.6403}},
{'currency_code': 'NPR', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 3.3583, 'selling_rate': 3.3618, 'middle_rate': 3.36}},
{'currency_code': 'SDR', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': None, 'selling_rate': None, 'middle_rate': 5.9361}},
{'currency_code': 'VND', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 0.0188, 'selling_rate': 0.0188, 'middle_rate': 0.0188}},
{'currency_code': 'USD', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 4.4, 'selling_rate': 4.404, 'middle_rate': 4.402}},
{'currency_code': 'TWD', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 14.4494, 'selling_rate': 14.4697, 'middle_rate': 14.4596}},
{'currency_code': 'THB', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 12.8938, 'selling_rate': 12.9093, 'middle_rate': 12.9015}},
{'currency_code': 'SGD', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 3.307, 'selling_rate': 3.3113, 'middle_rate': 3.3092}},
{'currency_code': 'SAR', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 117.2708, 'selling_rate': 117.393, 'middle_rate': 117.3319}},
{'currency_code': 'PKR', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 1.5575, 'selling_rate': 1.5813, 'middle_rate': 1.5694}},
{'currency_code': 'PHP', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 8.0734, 'selling_rate': 8.0837, 'middle_rate': 8.0785}},
{'currency_code': 'NZD', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 2.7553, 'selling_rate': 2.7582, 'middle_rate': 2.7568}},
{'currency_code': 'MMK', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 0.2101, 'selling_rate': 0.2103, 'middle_rate': 0.2102}},
{'currency_code': 'KRW', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 0.3339, 'selling_rate': 0.3346, 'middle_rate': 0.3343}},
{'currency_code': 'KHR', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 0.1075, 'selling_rate': 0.1088, 'middle_rate': 0.1082}},
{'currency_code': 'JPY', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 3.3376, 'selling_rate': 3.3409, 'middle_rate': 3.3393}},
{'currency_code': 'INR', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 5.3752, 'selling_rate': 5.3837, 'middle_rate': 5.3795}},
{'currency_code': 'IDR', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 0.0295, 'selling_rate': 0.0295, 'middle_rate': 0.0295}},
{'currency_code': 'HKD', 'unit': 100, 'rate': {'date': '2023-04-07',
'buying_rate': 56.0502, 'selling_rate': 56.1033, 'middle_rate': 56.0768}},
{'currency_code': 'GBP', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 5.4727, 'selling_rate': 5.4781, 'middle_rate': 5.4754}},
{'currency_code': 'EUR', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 4.8022, 'selling_rate': 4.8078, 'middle_rate': 4.805}},
{'currency_code': 'EGP', 'unit': 1, 'rate': {'date': '2023-04-07',
'buying_rate': 0.1422, 'selling_rate': 0.1428, 'middle_rate': 0.1425}}],
'meta': {'quote': 'rm', 'session': '1700', 'last_updated': '2023-04-07
23:01:55', 'total_result': 27}}
```

Explanation:

To retrieve the data from the Bank Negara Open API. The connection string has the following format. Here's a step by step for data retrieving using the Bank Negara Open API:

A. Create an API endpoint URL

```
url = 'https://api.bnm.gov.my/public/exchange-rate'
```

URL for the API: `Api.bnm.gov.my`

Chosen Endpoint: `Exchange-rate`

B. Choose the parameter for the API request

```
params = {
    'effective_date': 'latest'}
```

Here we chose the latest effective date.

C. Set your headers with the API key

```
headers = { 'Accept': 'application/vnd.BNM.API.v1+json'}
```

D. Send API request to the website

```
response = requests.get(url, params=params, headers=headers)
```

E. Print the output of the data

```
print(response.json())
```

F. Save it in a variable

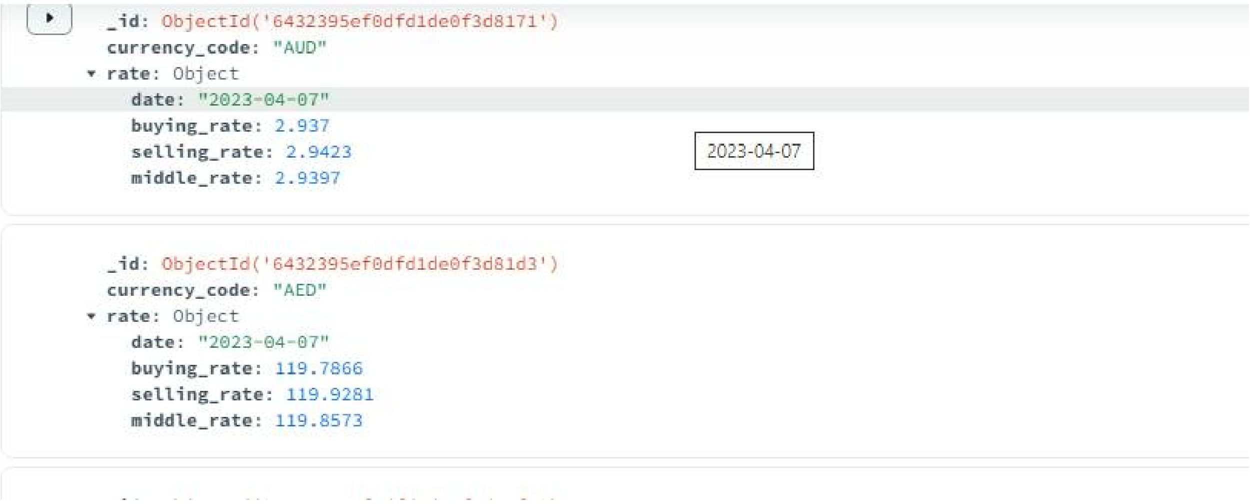
```
data = response.json()
```

Save the output in a variable called ‘data’

4. Connect To The Collection and Save In MongoDB

The code creates a MongoClient instance by passing in the connection string for the MongoDB server. The connection string includes the username and password for authentication, as well as the host and port of the MongoDB server. Then, the code gets a reference to the "db01" database within the MongoDB server using the client instance. After that, the code gets a reference to a collection within the "db01" database, which is named "Assg1". Finally, the code loops through the items in the "data" field and updates the MongoDB collection with the currency rate data. For each item in the "data" field, the code constructs a query to find a document with a matching "currency_code" field, and constructs an update operation to set the "rate" field to the value in the item. The update_one() method is used to execute the update operation on the collection, with upsert=True to create a new document if a matching document is not found.

5. Data in MongoDB



```
_id: ObjectId('6432395ef0dfd1de0f3d8171')
currency_code: "AUD"
rate: Object
  date: "2023-04-07"
  buying_rate: 2.937
  selling_rate: 2.9423
  middle_rate: 2.9397

_id: ObjectId('6432395ef0dfd1de0f3d81d3')
currency_code: "AED"
rate: Object
  date: "2023-04-07"
  buying_rate: 119.7866
  selling_rate: 119.9281
  middle_rate: 119.8573
```

Explanation:

The sample of output data in MongoDB

6. Import The Dataset To CSV File

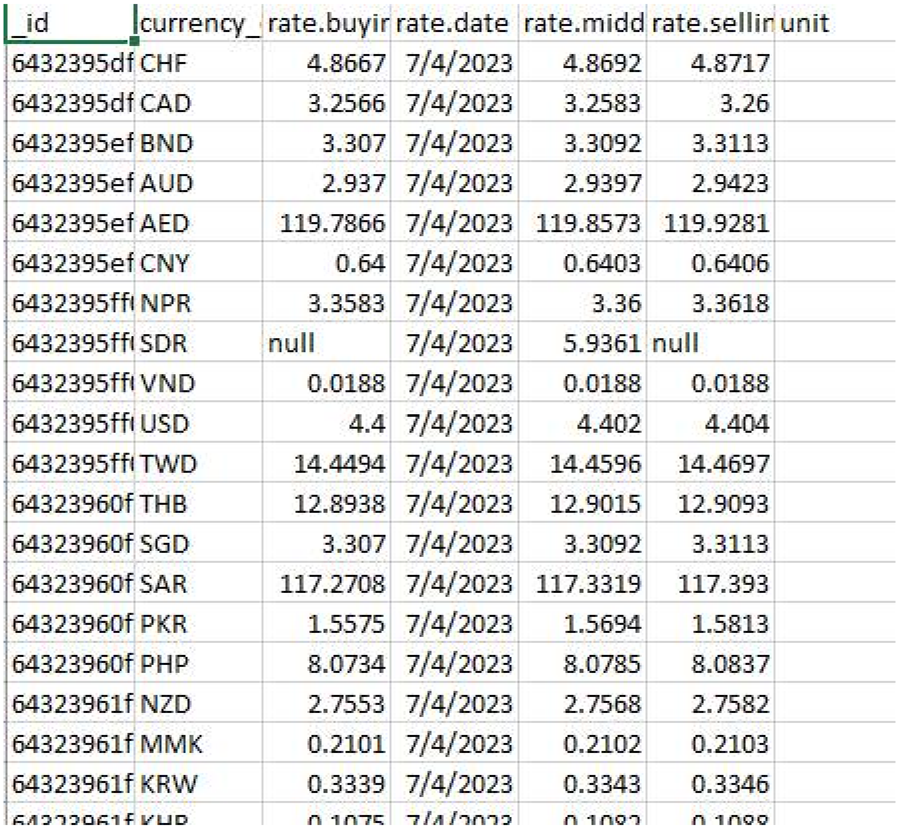
```
import csv

with open('MYRExchangeRate.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(data.keys())
    writer.writerow(data.values())
```

Explanation:

1. Import the csv library
2. The open() function is used to create a file object that can be used to read or write a file. In this project, the file is opened for writing and given the name 'MYRExchangeRate.csv'. The newline='' parameter is included to ensure that no extra blank lines are added to the CSV file.
3. Then, the 'writer’ object is used to write to the CSV file.
4. Then, 'writer.writerow(data.keys())’ This line writes the keys of the data dictionary object as the header row in the CSV file.
5. Finally, the values of the data dictionary writes objects as the data row in the CSV file.

7. CSV Output



_id	currency_	rate.buyir	rate.date	rate.midd	rate.sellin	unit
6432395df	CHF	4.8667	7/4/2023	4.8692	4.8717	
6432395df	CAD	3.2566	7/4/2023	3.2583	3.26	
6432395ef	BND	3.307	7/4/2023	3.3092	3.3113	
6432395ef	AUD	2.937	7/4/2023	2.9397	2.9423	
6432395ef	AED	119.7866	7/4/2023	119.8573	119.9281	
6432395ef	CNY	0.64	7/4/2023	0.6403	0.6406	
6432395ff	NPR	3.3583	7/4/2023	3.36	3.3618	
6432395ff	SDR	null	7/4/2023	5.9361	null	
6432395ff	VND	0.0188	7/4/2023	0.0188	0.0188	
6432395ff	USD	4.4	7/4/2023	4.402	4.404	
6432395ff	TWD	14.4494	7/4/2023	14.4596	14.4697	
64323960f	THB	12.8938	7/4/2023	12.9015	12.9093	
64323960f	SGD	3.307	7/4/2023	3.3092	3.3113	
64323960f	SAR	117.2708	7/4/2023	117.3319	117.393	
64323960f	PKR	1.5575	7/4/2023	1.5694	1.5813	
64323960f	PHP	8.0734	7/4/2023	8.0785	8.0837	
64323961f	NZD	2.7553	7/4/2023	2.7568	2.7582	
64323961f	MMK	0.2101	7/4/2023	0.2102	0.2103	
64323961f	KRW	0.3339	7/4/2023	0.3343	0.3346	
64323961f	KHR	0.1075	7/4/2023	0.1082	0.1088	

Explanation:

The sample of output data in CSV