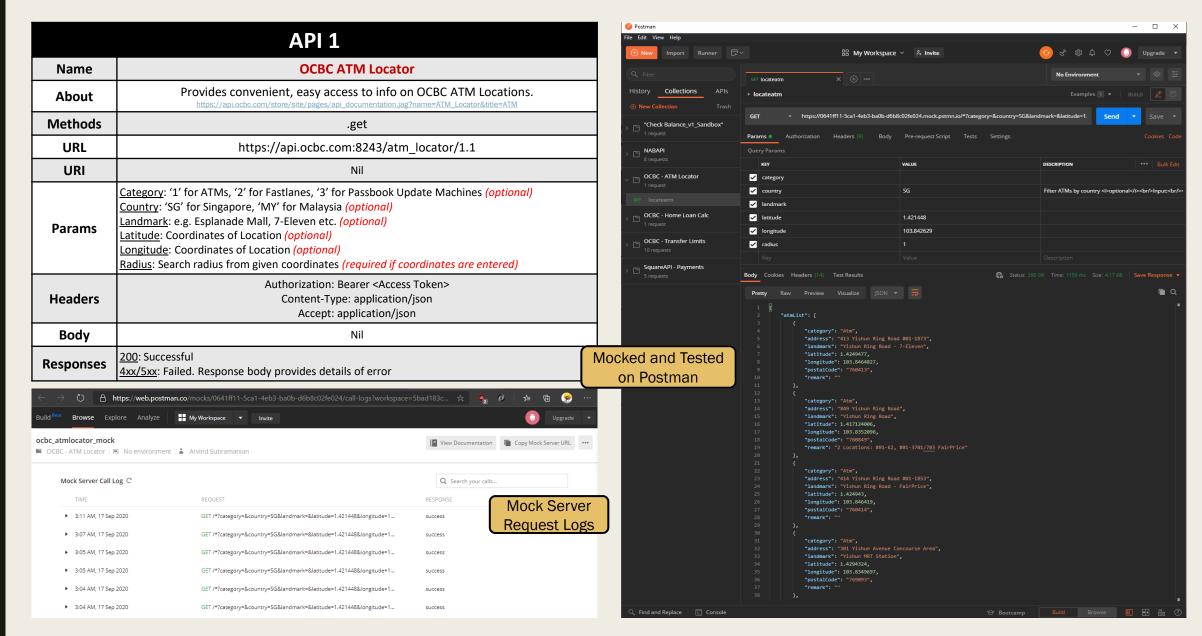
[PC-2: APIs and RPA]

ASSIGNMENT 1

APIs

by Arvind Subramanian

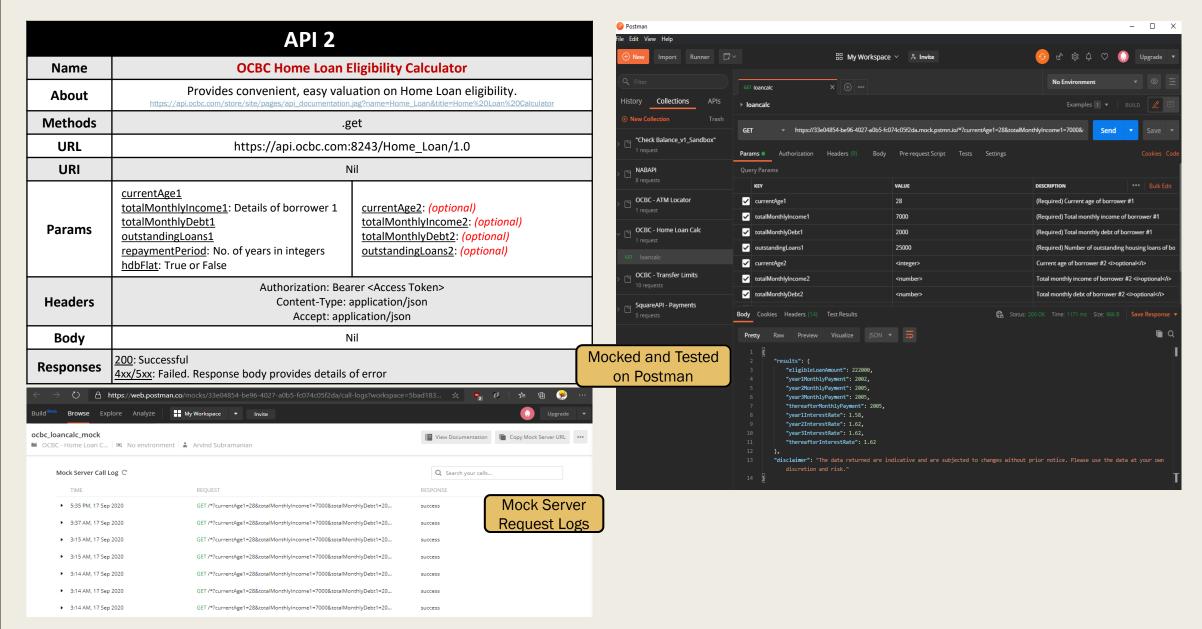


API 1					
Name	OCBC ATM Locator				
About	Provides convenient, easy access to info on OCBC ATM Locations. https://api.ocbc.com/store/site/pages/api_documentation.jag?name=ATM_Locator&title=ATM				
Methods	.get				
URL	https://api.ocbc.com:8243/atm_locator/1.1				
URI	Nil				
Params	Category: '1' for ATMs, '2' for Fastlanes, '3' for Passbook Update Machines (optional) Country: 'SG' for Singapore, 'MY' for Malaysia (optional) Landmark: e.g. Esplanade Mall, 7-Eleven etc. (optional) Latitude: Coordinates of Location (optional) Longitude: Coordinates of Location (optional) Radius: Search radius from given coordinates (required if coordinates are entered)				
Headers	Authorization: Bearer <access token=""> Content-Type: application/json Accept: application/json</access>				
Body	Nil				
Responses	200: Successful 4xx/5xx: Failed. Response body provides details of error	Calling Mock			

Code is attached in folder:

- qn1/qn1a_xx

```
① 127.0.0.1:5500/APIs-and-RPA/Assignments/Assignment%201/qn%201/qn1a_index.html
ATM Locations
"atmList": [
"category": "Atm",
"address": "413 Yishun Ring Road #01-1873",
"landmark": "Yishun Ring Road - 7-Eleven",
"latitude": 1.4249477,
"longitude": 103.8464827,
"postalCode": "760413",
"remark": ""
"category": "Atm",
"address": "849 Yishun Ring Road",
                                  qn1a_script.js - PC 2 - Visual Studio Code
Run Terminal Help
                      JS qn1a_script.js X
 APIs-and-RPA > Assignments > Assignment 1 > qn 1 > JS qn1a_script.js > ...
        //const url = "https://api.ocbc.com:8243/atm locator/1.1?country=SG&longitude=103.8
        const url = "https://0641ff11-5ca1-4eb3-ba0b-d6b8c02fe024.mock.pstmn.io/*?category=
        const options = {
            method: 'GET',
                 'Accept': 'application/json',
                 'Content-Type': 'application/json',
                 'Authorization': 'Bearer 3aab265ece717fc7f0fca79db6c5fb6a'
        fetch(url, options)
             .then(res => res.json())
             .then(function(data) {
                 document.getElementById("atms").innerHTML = JSON.stringify(data, null, '<br</pre>
```

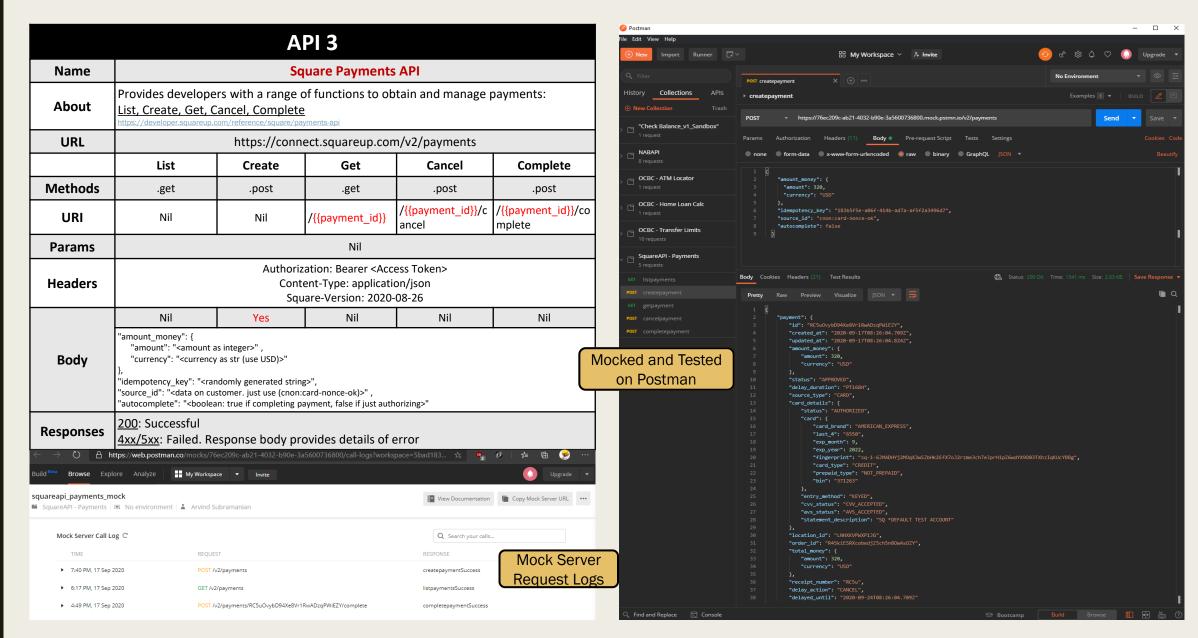


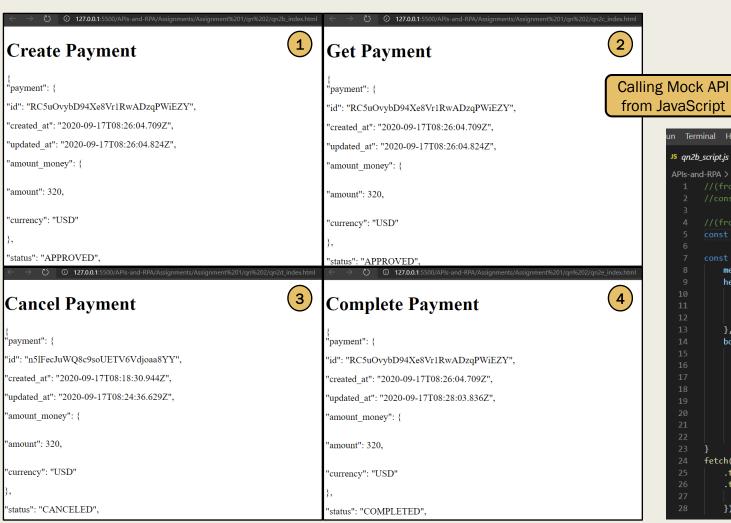
	API 2		
Name	OCBC Home Loan Eligibility Calculator		
About	Provides convenient, easy valuation on Home Loan eligibility. https://api.ocbc.com/store/site/pages/api_documentation.jag?name=Home_Loan&title=Home%20Loan%20Calculator		
Methods	.get		
URL	https://api.ocbc.com:8243/Home_Loan/1.0		
URI	Nil		
Params	currentAge1 totalMonthlyIncome1: Details of borrower 1 totalMonthlyDebt1 outstandingLoans1 repaymentPeriod: No. of years in integers hdbFlat: True or False	currentAge2: (optional) totalMonthlyIncome2: (o totalMonthlyDebt2: (optioutstandingLoans2: (optioutstandingLoans2)	onal)
Headers	Authorization: Bearer <access token=""> Content-Type: application/json Accept: application/json</access>		
Body	Nil		
Responses			Calling Mock A from JavaScrip

Code is attached in folder:

-qn1/qn1b_xx

```
127.0.0.1:5500/APIs-and-RPA/Assignments/Assignment%201/qn%201/qn1b_index.html
Loan Eligibility
"results": {
"eligibleLoanAmount": 222000,
"year1 Monthly Payment": 2002,
"year2MonthlyPayment": 2005,
"year3MonthlyPayment": 2005,
"thereafterMonthlyPayment": 2005,
"year1InterestRate": 1.58,
"year2InterestRate": 1.62,
"year3InterestRate": 1.62,
                                     qn1b_script.js - PC 2 - Visual Studio Code
                        JS qn1b_script.js X
   APIs-and-RPA > Assignments > Assignment 1 > qn 1 > J5 qn1b_script.js > ...
          const url = "https://33e04854-be96-4027-a0b5-fc074c05f2da.mock.pstmn.io/*?currentAg
          const options = {
              method: 'GET',
              headers: {
                   'Content-Type': 'application/json',
                   'Authorization': 'Bearer 3aab265ece717fc7f0fca79db6c5fb6a'
          fetch(url, options)
               .then(res => res.json())
               .then(function(data) {
                   document.getElementById("loan").innerHTML = JSON.stringify(data, null, '<br/>br
```

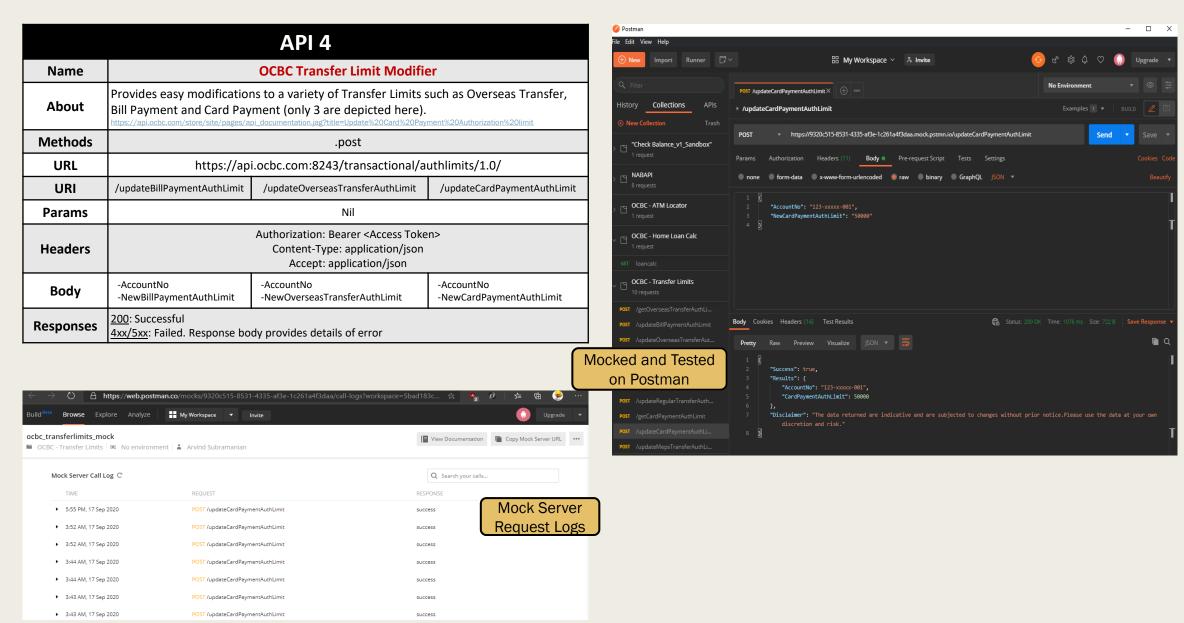


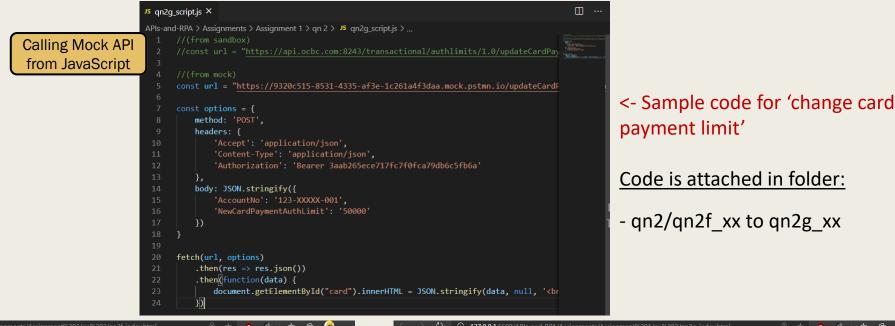


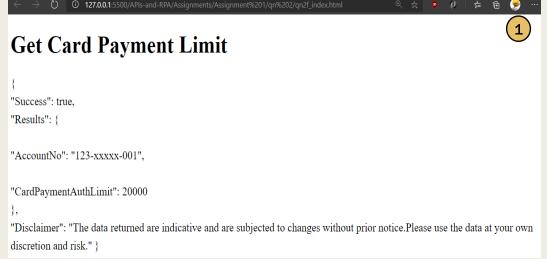
Code is attached in folder
- qn2/qn2a xx to qn2e xx

Sample code for 'create payment'

```
un Terminal Help
                                  qn2b_script.js - PC 2 - Visual Studio Code
 5 qn2b_script.js ×
 APIs-and-RPA > Assignments > Assignment 1 > gn 2 > Js gn2b_script.js > [@] url
       //(from mock)
        const url = "https://76ec209c-ab21-4032-b90e-3a5600736800.mock.pstmn.io/v2/payments
        const options = {
            method: 'POST',
                 'Square-Version': '2020-08-26',
                'Content-Type': 'application/json',
                'Authorization': 'EAAAEDT0g5R8jH5fHpiNr318F2hHj nsyj2Iw9fbZlTeEsJS1Cv6qpSw7
            body: JSON.stringify({
                "amount money": {
                     "amount": 320,
                    "currency": "USD"
                  "idempotency_key": "c7fecba2-9185-4900-b3c4-12d36b83919d",
                  "source id": "cnon:card-nonce-ok",
                  "autocomplete": "False"
        fetch(url, options)
            .then(res => res.json())
            .then(function(data) {
                document.getElementById("crtpmt").innerHTML = JSON.stringify(data, null, '<</pre>
```







```
Change Card Payment Limit

{
"Success": true,
"Results": {

"AccountNo": "123-xxxxx-001",

"CardPaymentAuthLimit": 50000
},

"Disclaimer": "The data returned are indicative and are subjected to changes without prior notice. Please use the data at your own discretion and risk." }
```

Qn 3: Designing an API

API IDEA					
SMRT Train Location and Crowd Tracker					
About	Provide Accurate and Real-Time Data on: - Location of each train in each MRT line - Crowd concentration in each train and in each compartment - Crowd concentration in every station and passenger walkway	1	g time and overall commute duration edness and enhanced comfort level		
Current Situation	Lack of awareness on the arrival times of trains until at or very near the MRT stations Estimates displayed by SMRT may be inaccurate Lack of awareness on the crowdedness of trains and platforms				
Possible Consumers	Map Service Providers (Google, Apple etc.) Taxi Companies (Grab, Comfort etc.)				
Value-Add to Consumer	Map Service Providers Increased Usage by MRT Commuters - Provide more accurate train commute timings on their own services - Maps generally more utilized by drivers - Providing quality-of-life upgrade encourages day-to-day usage Effective Promotion of Other Services - Better awareness of MRT service conditions allows more targeted advertising - e.g. '6-mins till the next train, why not grab a quick curry-puff at the OCK near you?'	Improve Assessment of Passenger Demand - Monitor MRT service conditions alongside weather, traffic etc. - Adjust taxi pricing in accordance with real-time 'attractiveness' of MRT commute rgeted advertising			
End-Users of Client App	Everyday MRT Commuters				
Value-Add to End-Users	Enhanced Quality of Life in MRT Commute - Lower travel durations - Know where to stand/wait to enjoy the most comfortable ride - Flexibility to squeeze in other personal activities during/before commute (e.g. grab a quick toilet break)		Better Overall Commuting Experience - Know for sure whether to walk to the MRT or just call a Grab instead - Improved accessibility to taxi services when needed		
Value-Add to Company	Greater ridership Improved crowd distribution on trains and platforms (no need for station marshals) Automated data collection on train run-times, mileage and passenger load - Improved maintenance and train scheduling Accelerated response to train delays/faults Revenue generation from API usage →		Pricing Model Pay-per-Use or Subscription Based No direct transactions provided to consumers to charge commission off Focus is on enhancing business efficiency and improving commuter quality of life		
Transmitted Data			[Request] .get TrainCrowdedness train for a given station [Response] crowd density across the length of a train for a given train		

END