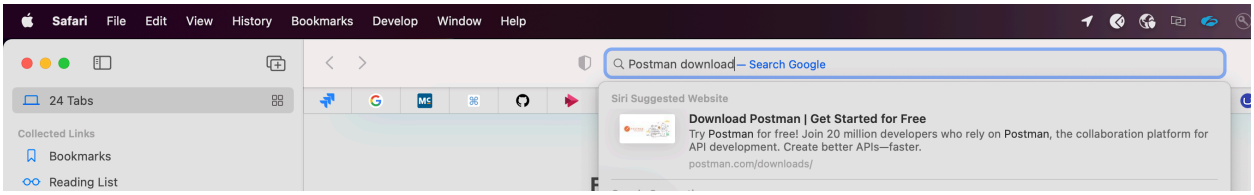
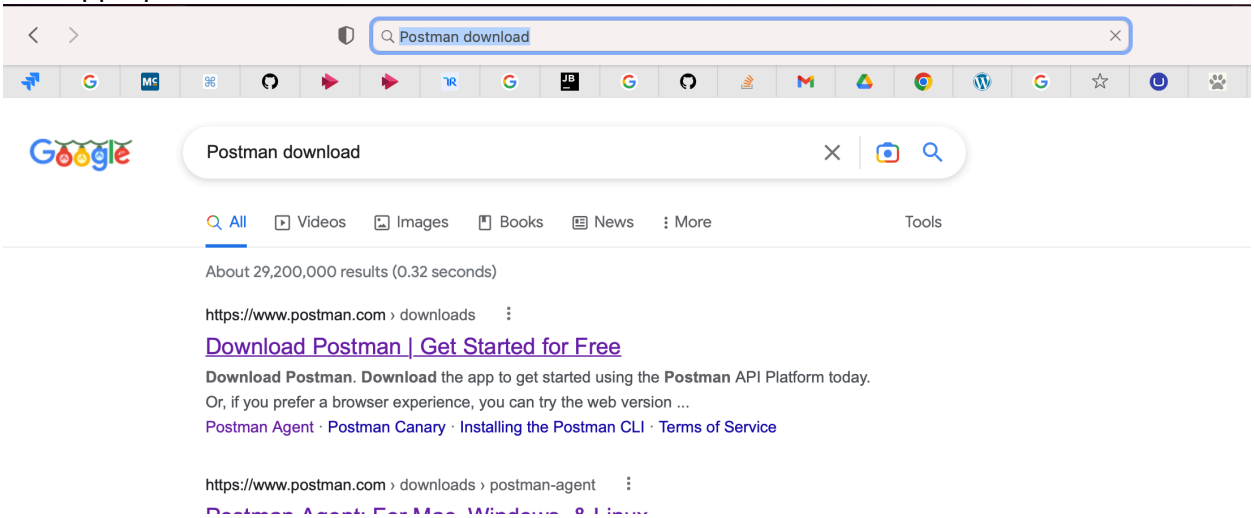
Home work for Introduction in IT

Part 1. Validate Rest API calls by using Postman tool.

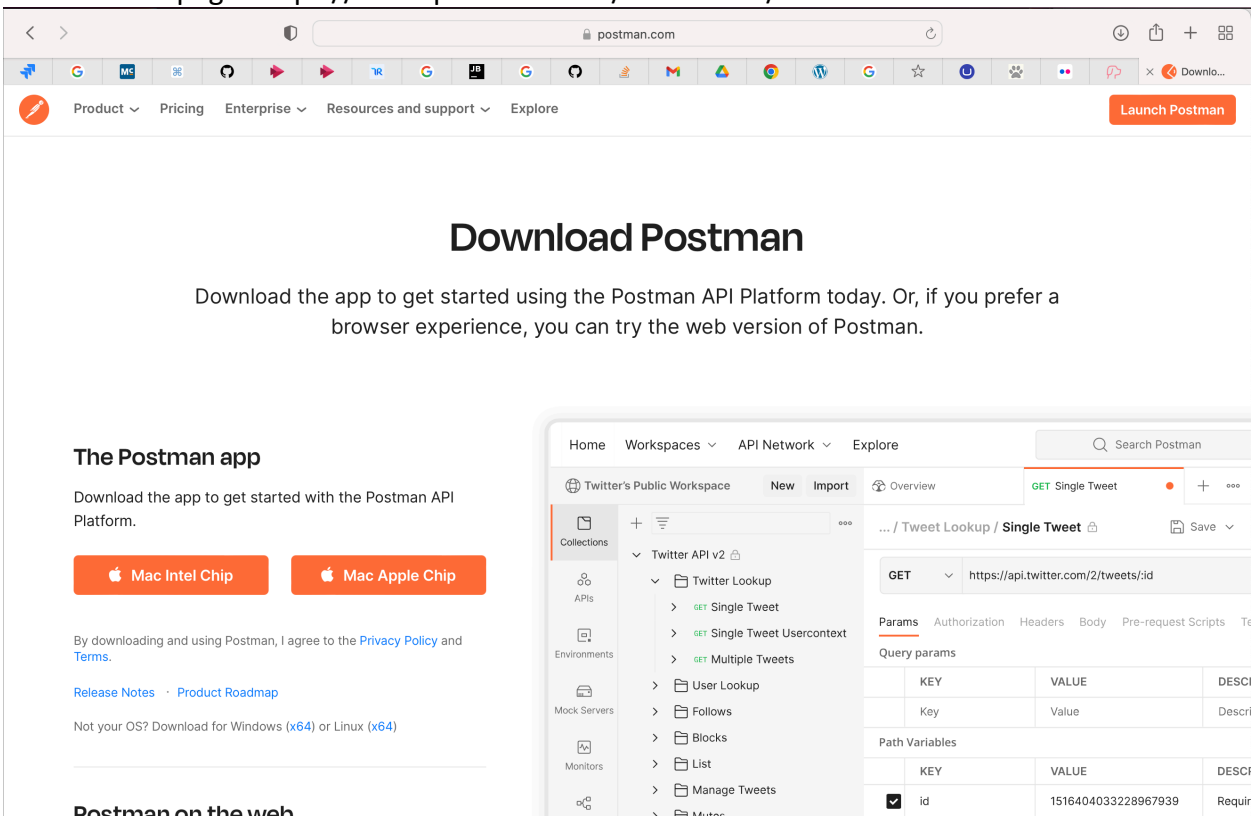
1. Download “Postman”



2. Find appropriate search result:



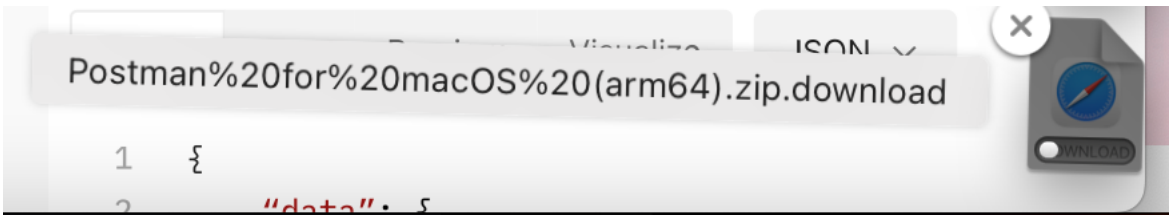
3. Go to official page: https://www.postman.com/downloads/



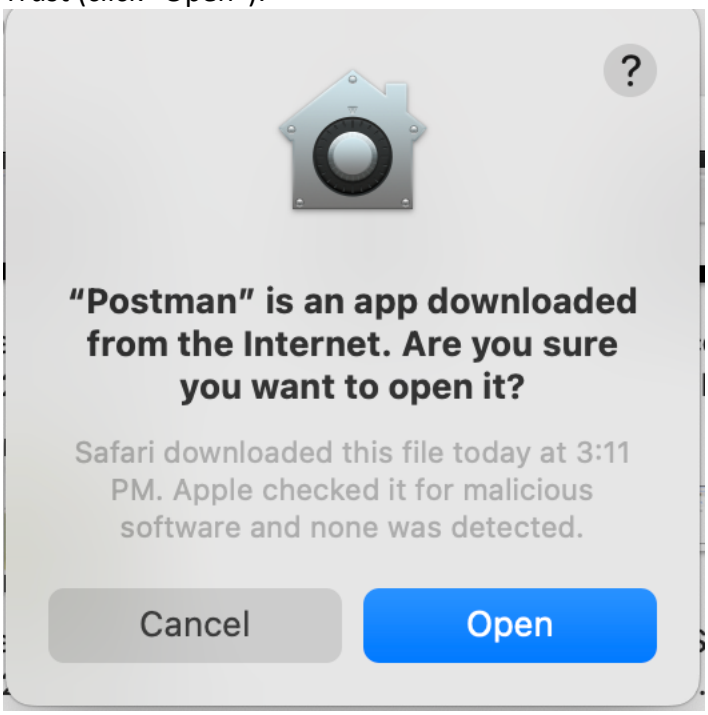
4. Check you requirements – for our example we use

Apple chip:

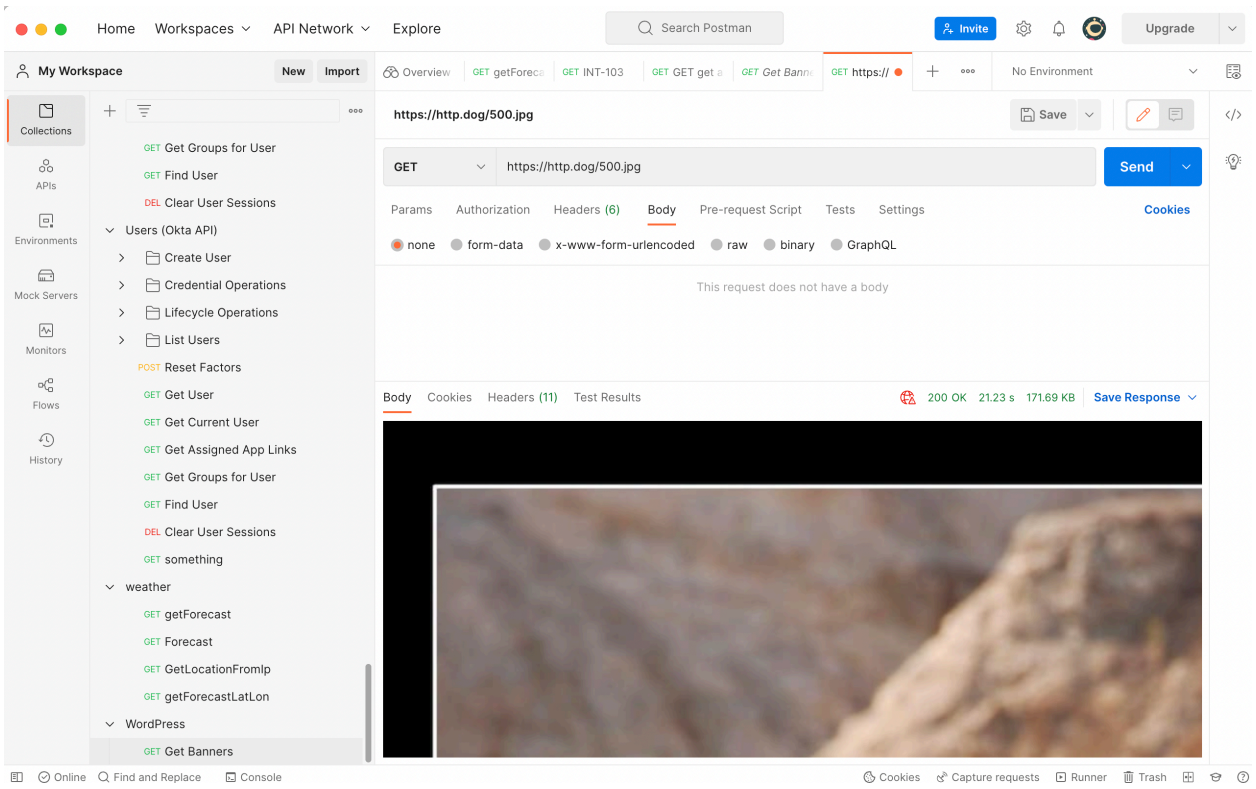
5. Dowload:



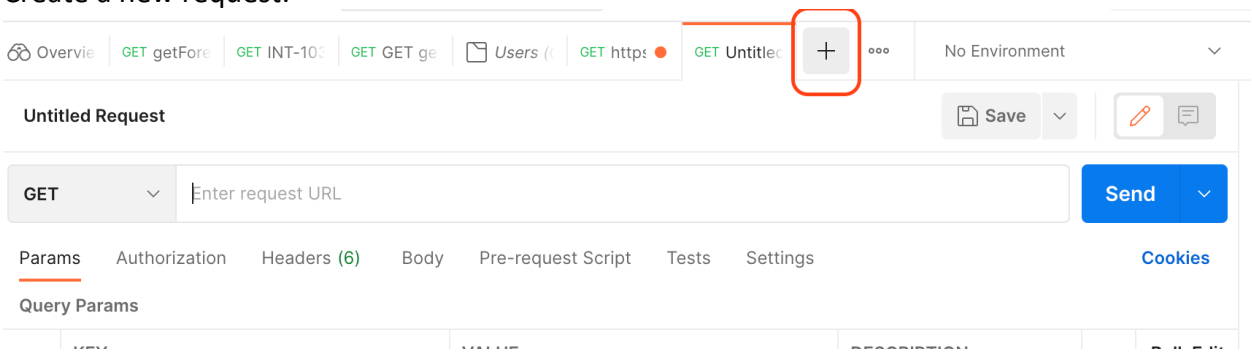
Trust (click “Open”):



Install and open:



6. Create a new request:



choose get and use
URL :
https://http.dog/[code].jpg

Instead of [code] put particular HTTP status code you want to research.

Example:
https://http.dog/200.jpg

Please analyze the result

```
Validate another API:
- https://api.coindesk.com/v1/bpi/currentprice.json
```

- https://catfact.ninja/fact
- https://www.boredapi.com/api/activity

Predict the gender of a person based on their name.

- https://api.genderize.io?name=luc

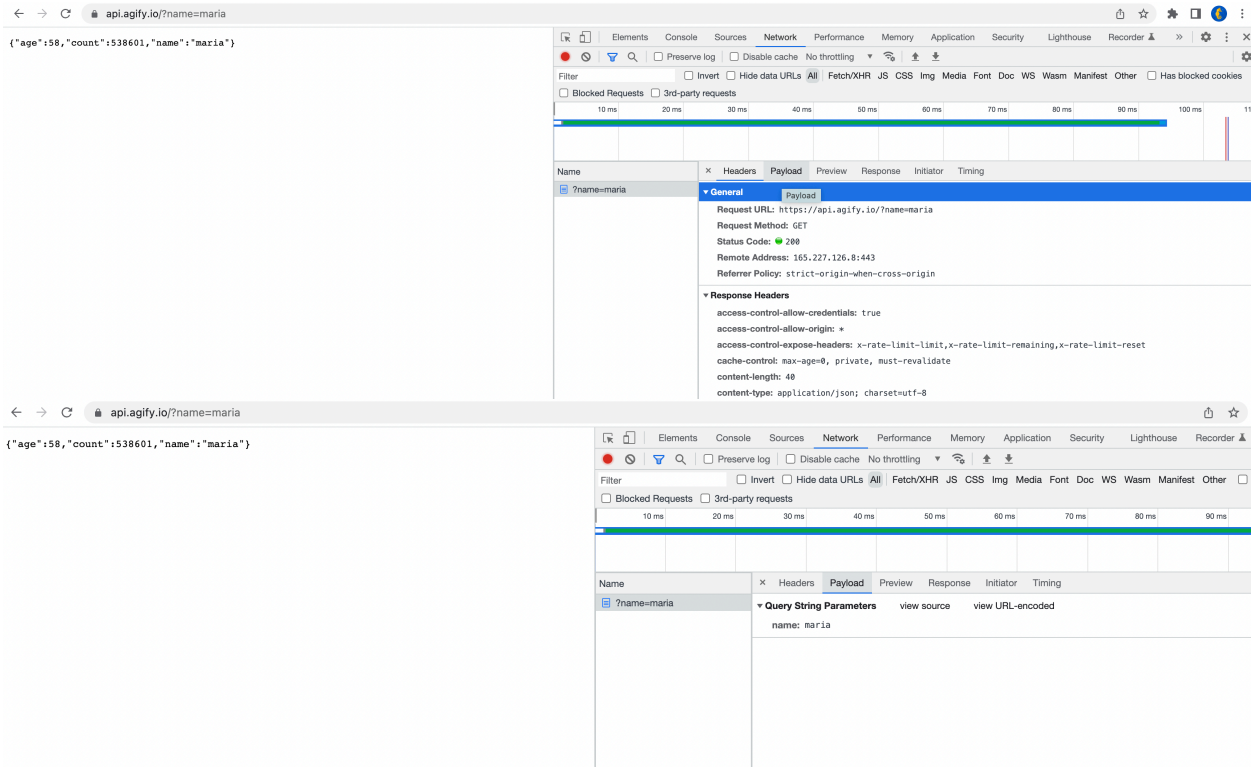
check your name.

Predict the age of a person based on their name.

https://api.agify.io?name=meelad

Could you validate it in Browser as well? See difference/

Validate with using Development tool in the browser (Chrome) header and payload:



As well as in Postman.

Part 2. Swagger – tool for documentation and testing APIs.

Will work with Swagger:
https://petstore3.swagger.io
https://petstore3.swagger.io/#/user/loginUser
user : test
password abs123

Follow our video please :
- Create the ser,
- Get information about this user,

- Login as a user,
- Update information using PUT method,
- Check what kind of information about your user you have,
- Delete the user

Try to think about cases you need to test functionality for User Creation, User Update and User deletion.

Ticket was:
Create functionality for "Operations about user" information about user, update the information, Delete the information about the user and get information about user.

Need to use next fields:

Field	Defenition	Type	Example
id	Unique identifier	Integer	id
username	Unique identifier	String	ItExpertSchool
firstName	First name of the user	String	Alex
lastName	Last name of the user	String	Simpson
email	email	String	asimpson@egmail.com
password	Password, will allow to log into the system	String	abc123
phone	Phone number	Digits only 11 max, "-" are allowed	222-333-4567
userStatus	User Status	Digits 0..9	1

Response should be in XML format,

Request body – Json format.

How will you test it?
User provided you the link to Swagger and asked to test it.

You need to think about test case scenarios and test cases for these scenarios.

Example:

- Check USER creation Functionality
- Check Login Functionality

Please write test cases like this: for each scenarios you created:

- Test Case 1: Check results on entering valid User Id & Password

  Input values:
  Output values:

- Test Case 2: Check results on entering Invalid User ID & Password
- Test Case 3: Check response when a User ID is Empty & Login Button is pressed, and many more

Appendix.

Introduction:

What is API?

API stands for **Application Programming Interface.** It defines how two pieces of software talk to each other. There are several types of APIs, but the swagger specifically deals with the Web API.

How do Web APIs work?

Let's understand the working the Web API through an example. Suppose we opened the **Facebook**on our phone and made a request to the **Facebook** server. The request sent to the **Facebook** server is known as an API request and the **Facebook** server will send the

response known as API response. The server will only send the data, not the whole web page. It is the responsibility of the app to display the web page.

Here, API definition works:

- What requests are available
- What the response looks like for each request.

Swagger and Open API specification are mainly designed for the Rest API, where Rest is a type of web API. In Rest word, R stands for Representational, S stands for State, and T stands for Transfer.

What is API Definition?

The API Definition is a file that describes all the things that we can do with an API. It contains all the requests that we can make to an API. It also describes what request to make and how would response look like for each request.

Why create an API definition?

There are several advantages of writing an API definition:

- It allows you to design the API before implementing it. The developers can review the API before writing the code for the API.
- It also helps in automated testing.
- It can automatically create a code in several languages.
- It can also be used to generate the documentation automatically.

API Definition File

API Definition File is a file that contains all the things that you can do with a file. This file contains the following things:

- Server location
- How security is handled, i.e., authorization.
- All the available resources in that API.
- All the different data that you can send in a request.
- What data is returned

- What HTTP status codes can be returned

Anatomy of a Request

There are five different parts to be found in the Http request:

1. Method: The method describes the action to be performed. The methods could be POST, PUT, DELETE, GET.
2. URL: It specifies the name on which the action is to be performed.
3. Query parameters
4. Headers: Headers are used to store the information about the request.
5. Body: Body contains the additional data.

URL is broken down into several pieces:

For example: the request URL is: https://api.example.com/v2/user

- Scheme: https
- Host: api.example.com
- Base path: /v2
- Path: user

Request Body

We mainly specify the request body in JSON format for some methods such as PUT, POST, etc. The body is treated as parameters like path in url. Unlike these parameters, we create the schema for the request body that specifies how the JSON body would look like.

In REST, the response body could be anything, but mainly the response body is written in JSONformat. The response body is included in the response object. The response body has a schema to represent the structured data. We can also have a separate response object for each HTTP status code returned.

Security

Here, Security means authentication and authorization. Authentication means to validate the user through their username and password. The authorization means allowing the user to access the data.

The security can be set in the following ways:

- **None:** Here, **None** means that no security is set to access the API.
- **Basic Auth:** It means that the username and password are set for each request.
- **API Key:** The key is set to access the API.
- **OATH:** It is an authorization scheme.

Documentation

The OAS file or API file contains the human-readable description of elements that generates the documentation automatically. In other words, we can say that a description section is added for the API, for each operation which is a combination of path and method, for each parameter, and for each response element.

Structured Data Formats

The Open API Specification uses the structured data format for its API definition files. We can use one of the two structured formats, either YAML or JSON.

YAML

YAML stands for **Ain't Markup Language.** It is not a Markup language like **HTML.** It is used for the data, not for the content. YAML uses minimum characters as compared to JSON and XML. It is mainly used for the configuration

files rather than the files which are passed over the web like JSON.

Key/value pairs

The data in YAML is represented in the form of key/value pairs. Key/value pairs are indicated by a colon followed by a space.

For Example:

1. Date: 2021-07-08
2. First name: John

In the above example, Date and First Name are the keys, and 2021-07-08 and John are the values.

Levels

Levels are indicated by white space indenting, but we cannot use tab indent. This is the biggest difference between the YAML and the other structured formats. XML uses tags that add one level, and inside the tag, there are other tags that add another level; so, this increases the number of characters. In JSON, opening and closing brackets indicate one level that occupies many characters. In YAML, the only indentation is used, which occupies fewer characters.

XML:

```
<name>
    <firstname> John </firstname>
    <lastname> Malik </lastname>
</name>
```

JSON:

```
name: {
  "firstname": "John"
  "lastname": "Malik"
}
```

YAML

```
name:
  firstname: John
  lastname: Malik
```

Types

The types in YAML are determined from the context.

For example:

```
part_no: A4786
description: Photoresistor
price: 1.47
quantity: 4
```

In the above scenario, **part_no** will be treated as a string, **description** will also be treated as a string, **price** will be treated as a floating type, and **quantity** will be treated as an integer.

List

- List in YAML is similar to the JSON. We need to use a dash to indicate a list item.
- We do not need to declare the list.

```
cart:
  -part_no: A4786
   Description: Photoresistor
   Price: 1.47
   Quantity: 4
-part_no: B3443
   Description: LED
   color: blue
   price: 0.29
   quantity: 12
```

As we can observe in the above example, that cart is the name of the list, and there are two list items in the cart. Both the list items are represented by the dash. The first list item contains 4 key-value pairs, whereas the second list item contains 5 key-value pairs.

Multi-line Strings

As we know that strings do not contain quotation marks so we need special characters for multiline strings. The following are the characters used for the multi-line strings:

- |: It preserves the lines and spaces.
- >: It means fold lines.

```
S: |
  YAML
    and JSON.
```

In the above example, we have used '|' character so its output would be same as it is written above.

Output

```
 YAML
and JSON
```

If we use > character instead of '|' character:

```
S: >
 YAML
    and JSON.
```

Output

YAML and JSON

History of Swagger

- Historically, Swagger was a specification for how to create an API definition file.
- When the new version was released, i.e., Swagger 2.0, specification became the Open API Specification (OAS).
- Now, swagger is no longer a specification but it is a collection of tools that use the Open API specification (OAS).
- Many people refer OAS as Swagger but technically it is not.

Open API Initiative

- The Open API initiative is an organization created by consortium of industry experts.
- It is focused on creating, evolving, and promoting a vendor neutral API description format.
- It is in charge of Open API Specification but not in charge of any tools that use it.

Before understanding what is swagger, we will first understand what is Open API specification?

Adding a Request

Let's define requests for getting a photo albums. The following is the information that will be included in the request:

- URL endpoint
- HTTP Method
- Path parameters

- Query parameters

Let's understand the query parameters through an example.

Get *https://api.javatpoint.com/photo/album?start=2021-05-01&end=2021-05-31*

API Definition file