**GIT – version control**

Open source common repository for the distributed version control system

Installation notes – download from the official git, install in program files, use git from windows command prompt, use min TTY setting,

Once the git is ready, then we need git hub

x- writes codes and create test cases - EST

y - IST

GIT HUB repository can be used to posts the code, we need to know the git commands for the communication

The git merges the codes together, for example, there is something the developer updated in the code, and after that he pushed it to the git hub, then the code will be merged with previous version and provide the tester with the updated version.

Sometimes the git gets confused but 90 % of times it will merge in a right way. Git can give the conflict

**Creating git config and repositories**

We need to create an empty repository; you can use the **public or private** as per the requirement

There are some tools, which can help use the git:

* Tortoise git
* Eclipse – push pull buttons
* IntelliJ

But is better to the core git commands from the command prompt to have the basic information

<https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html>

The above link contains the basic git commands that can be used in a live project

1. Tell git who you are, etc.

**Create a folder in the users> Amrit> Gitstuff** – in this add a code which can be anything

Then in the command line interface, we will change the directory to the git stuff folder that we created and added the code into.

Then we will run the **git Init** command **to initialise the git in the folder** and this command adds a git file to this folder. (Sometimes the file is hidden file)

We have to commit the code. Git hub will access the code that is committed.

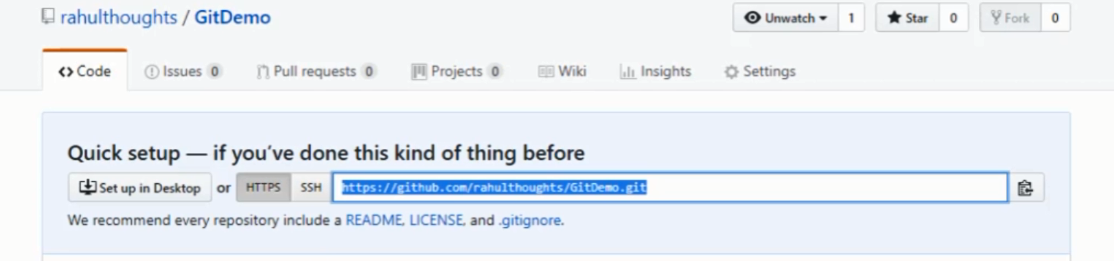
**Stash /staging** – commit looks only for the files that are in the stash and commits them.

We need to add the code in the stash and from there the commit will commit the code and then update on the git hub after that.

Next thing is the push, but before that we need to make the connection with the remote repository by giving the command

**git remote add origin <server>** - we need to do this for the first time and then after that the system remembers you.

The server address you will get from the git website when we make repository



Now, push the code in the master branch

Use the command - **git push origin master**

Then git will ask you for the credentials

When we use **git add \*** to push the project to the staging, then we get the commit along it and we can check the status, but we still have to push our code

You can add the **commit messages** to specify the commit update as **git commit -m “I am updating”**

**Git pull** is the command used for pulling in the command line interface

Another is git pull origin master

**Importance of Branching in GIT**

We can make a branch by **git checkout – b <branchname> only to create new branch we will use -b**

By this, it will make a copy of the repository and change its name in the branch

For example – developer branch and the master branch (for regression testing)

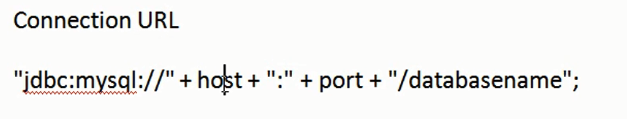
We need to specify the branch name when pulling the project

We can switch the branch from developer to master by simply

**Git checkout develop** – it will automatically switch to develop

Git merge develop – this command will merge the mentioned branch with the active branch which in our case is the master.

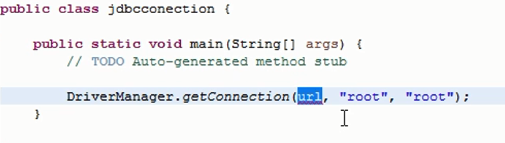
**Database connection to selenium test cases**



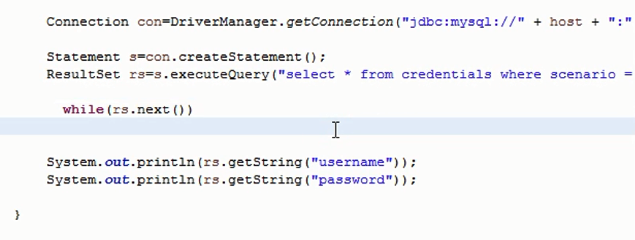
**Port no 3306** it’s the port that you will get when you install the database

Host is the **localhost** of the system

Database name is the database you want to connect



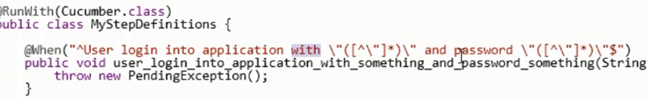
We need (rs.next())



**Cucumber Notes**

**TidyGherkin** plugin is used to generate the step definition file from the feature file. It automatically describes the function in the step definition file and the user just need to define the code in the function.

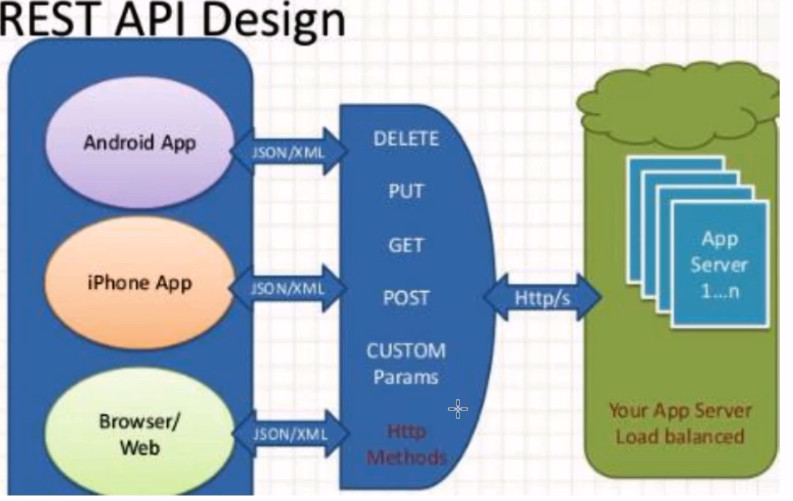
The regex function are used to accept the different values



If the sentence is defined in the” ” and then it can accept the anything after the with and password in the above commands

REST API





API layer is independent of the front end as seen in the diagram

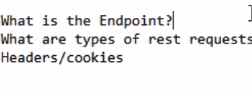
Example

Place search API – google map rest api

XML/Json – input and the radius

As an API tester we work on the validation of responses for API calls

Three questions we need to ask before the API Testing



Endpoint: Base URL/resource?Parameters

The resource and the parameter can be separated by a ‘?’ symbol

Qaclickacademy.com/course-detail?id=wev

Base URI –QaClickAcademy - if we need to talk to rest api we need to give the base url just like no in a telephone call

Resource

Parameters

Every webpage call is a rest API call. For example, when we search a term on the google map, it uses the map API to build the REST API call request and access the API to get a response

The URL below is an example of the rest API call consisting of the base url/resource? Parameter or id

https://www.google.com/search?source=hp&ei=HJGTXIbuA5CHgge77bc4&q=maps&btnK=Google+Search&oq=maps&gs\_l=psy-ab.3..0i131l2j0l8.1064.1453..1619...0.0..0.71.232.4......0....1..gws-wiz.....0.ErlIgycKAZM&safe=active&ssui=on

**What are the types of rest request?**

GET, POST, PUT, DELETE

**GET** – If you hit the rest request with a **get type**, then the system will understand that the need of the request is to **retrieve something from the server.**

In this, parameters needs to be send as part of request URL

**POST** – we are trying to add something like course title, course description, we can add or update something.

Parameters should be in the form of XML/JSON in the separate body but not in the current URL

Request body/Payload – in xml/json format

**PUT** – to update or change the data, this request is used, it will be in form of POST and will be sent using the payload or request body

Course name =

Description =

Trainer details =

**DELETE** -

Request Body/Payload

Course name =

Description =

Trainer details =

What are headers and cookies in the REST request?

Headers

1. **Key**-(For authentication) value to unlock the server, without the key server cannot recognise any API request (for security). Server reads the header value to authenticate a particular user
2. **Content – type** (another type of header) = Json

Google place search API

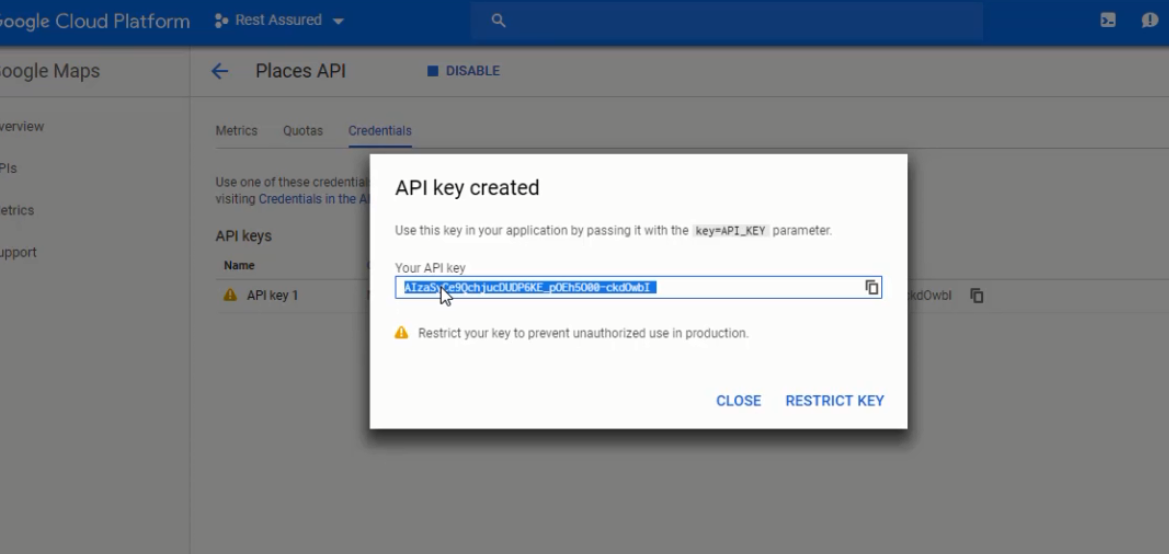
In this the required parameters are defined by the google are:

1. Key - which is the application’s API key. This identifies your application for the purposes of quota management and so the places added from your application are made immediately available.
2. Location
3. Radius – defines the distance within which to return place results. The max allowed radius is 50 000 meters.
4. If rankby= distance

Example

https://maps.googleapis.com/maps/api/place/findplacefromtext/json?input=%2B61293744000&inputtype=phonenumber&fields=place\_id&key=YOUR\_API\_KEY

How to generate an API key for google API



Google cloud platform> create a new project> from the API selection page select the places API >Create credentials> API Key

API Key

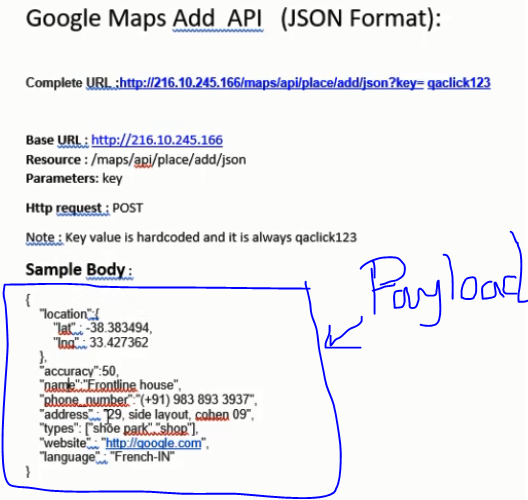
AIzaSyDhnnyJi5J-2bCn\_COe1Qr14GS4VAeXjNU

POST MAN

Rest Client

SOAP UI

Jmeter



We can send JSON as a payload in the POST body by selecting the option the body select raw option and using drop down to make it JSON

**Complete URL :[http://216.10.245.166/maps/api/place/add/json?key=](http://216.10.245.166/maps/api/place/add/json?key=qaclick123)****[qaclick123](http://216.10.245.166/maps/api/place/add/json?key=qaclick123)**

**Base URL** : <http://216.10.245.166>  
**Resource** : /maps/api/place/add/json  
**Parameters**: key

**Http request** : POST

Note : Key value is hardcoded and it is always qaclick123

Sample body

{

"location":{

"lat" : -38.383494,

"lng" : 33.427362

},

"accuracy":50,

"name":"Frontline house",

"phone\_number":"(+91) 983 893 3937",

"address" : "29, side layout, cohen 09",

"types": ["shoe park","shop"],

"website" : "http://google.com",

"language" : "French-IN"

}

The developer or the client has to provide this body as they specify how the payload is to be sent across for the request

The response message

{

"status": "OK",

"place\_id": "2a0b96fc13222cacf16a8def06f42cfb",

"scope": "APP",

"reference": "a68c4b6e31210bd6912df82b5f0ef24ca68c4b6e31210bd6912df82b5f0ef24c",

"id": "a68c4b6e31210bd6912df82b5f0ef24c"

}

The difference between GET and POST request is by the GET request we specify parameters and retrieve the information while in the POST request, we are sending the payload in the body either in the JSON.