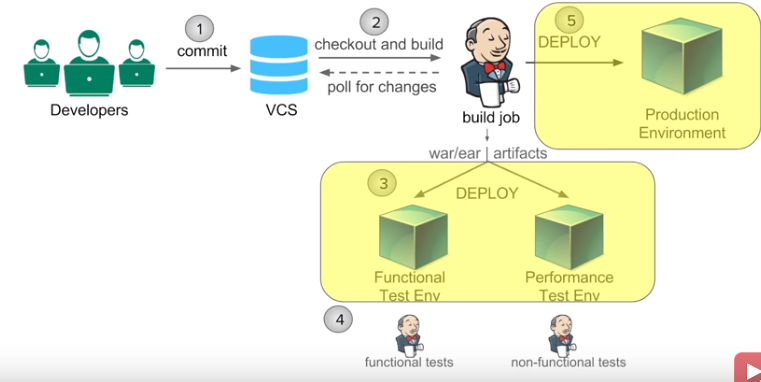
Automated Deployment:

Automated Deployment is the process of Automating the deployment process in a Continuous Delivery System.

Build > Deploy > Test > Release.



Jenkins :

1. Running Jenkins standalone (Jetty/Winstone) v/s Running Jenkins on Tomcat

* We can run Jenkins inside the Tomcat apache server.

Localhost:8080/Jenkins

Pre-requisite \*\*\*Tomcat 5 or above\*\*\* \*\*\*java(7 or above) should be available\*\*\* ([https://www.youtube.com/watch?v=FqpmH...](https://www.youtube.com/watch?v=FqpmH8MVO6A&list=PLhW3qG5bs-L_qj1L5hnHvJYeFpQ_g4UuU)) Step 1 : Download Tomcat Step 2 : Unzip and place tomcat folder at any location Step 3 : Copy/Place the jenkins.war file inside tomcat/webapps folder Step 4 : Goto commandpromt (cmd) - windows | terminal - mac - goto tomcat/bin directory - make all files executable : chmod +x \*.sh In case of windows this command will not work. This step might not be needed if you are on windows. Else you can right click on the file/folder goto security tab and change the permissions.

* Localhost:8080/systeminfo -> To see all the information.
* **Java -jar Jenkins.war - -httpPort=9090** -> to start Jenkins on other port
* To work on Jenkins in CLI mode: <http://localhost:8080/cli/> -> Gives the details.
* GIT(SCM) – Source code Management System
* GIT and GIT HUB tutorial by Raghav PAL – Automation
* Manage plugins -> Install Git Plugin, Role based Authorization strategy.

New Item -> Adding details:

**Configure the project**:

**Source code Management**

**Build Triggers** -> Build periodically -> Need to give an expression here.

Trigger Builds remotely: localhost:9090/job/TestProject1/build?token=1234 (this command can be executed from other system as well and changing localhost to IP address)

This will build remotely and show **Build Queue 1.**

**Console Output:**

Building in workspace C:\Users\Admin\.jenkins\workspace\TestProject1

**Build** -> Execute windows batch command

Build projects in chain job executions -> One build after other based on condition like stable or unstable.

Jenkins Beginner Tutorial 8 - Jenkins integration with GIT (SCM)

1. Create a java program and run it through command line

2. Create a Jenkins job to run the java program

3. Add this program/project to Git

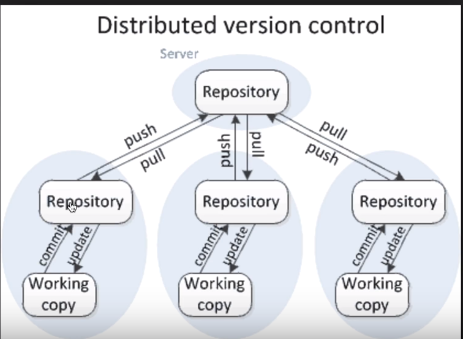
4. Jenkins - add git plugin

5. Configure Jenkins job to trigger the execution when a change is pushed to GitHub

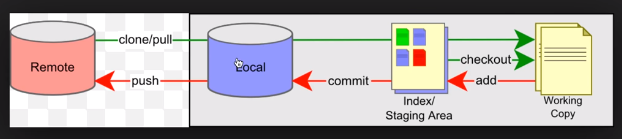
GIT

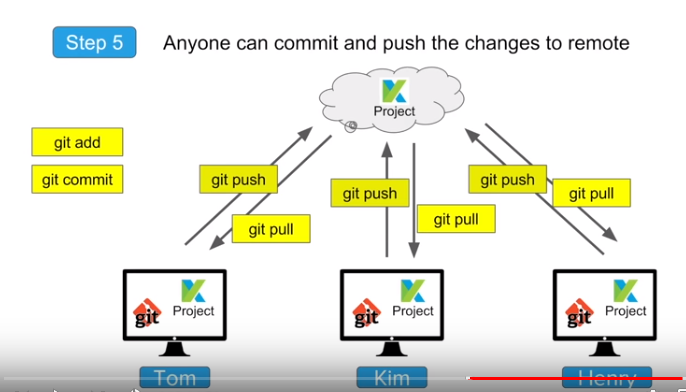
Git download link: <https://git-scm.com/download/win>

* GIT is a Distributed Version Control System(VCS) and it is a Open Source.



A Work Flow:





GIT Commands:

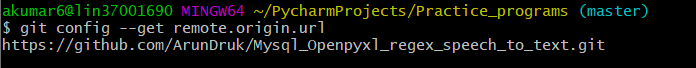
* git init (Go to the folder path in cmd where the project is present and enter git init)

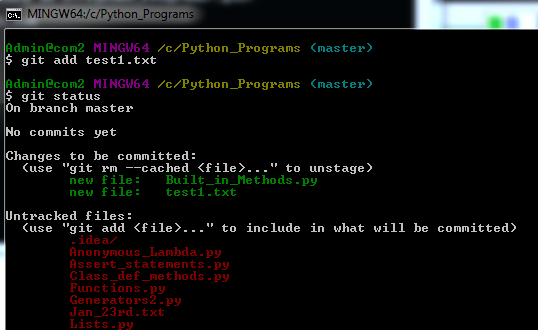
git init -> Is to initialize tracking of files and folders.

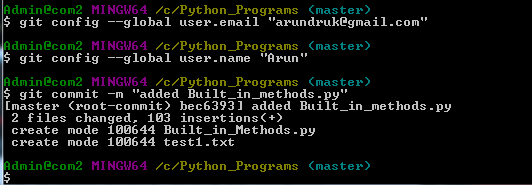
* git status
* git add test1.txt
* git add . (This adds all the files present in the folder to the git)
* git \*.csv (This adds all the files present in the folder with extension .csv to the git)
* git commit –m “new files added” (-m is the message we are displaying)
* git restore –staged filename (This restores the file which is earlier added to git and doesn’t go to git commit)
* git push –u origin master

GIT command to get the active remote URL:

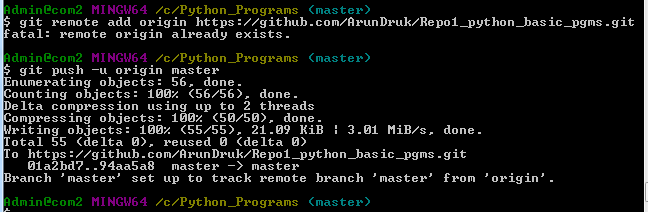
git config --get remote.origin.url

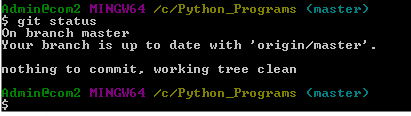






* **git remote add origin** [**https://github.com/ArunDruk/Repo1\_python\_basic\_pgms.git**](https://github.com/ArunDruk/Repo1_python_basic_pgms.git) **(**This folder or path tells where the new file or project needs to be pushed)
* git push -u origin master

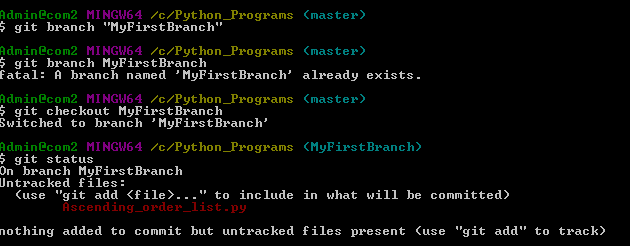




* git log
* git –help

Creating Branch and checkout :

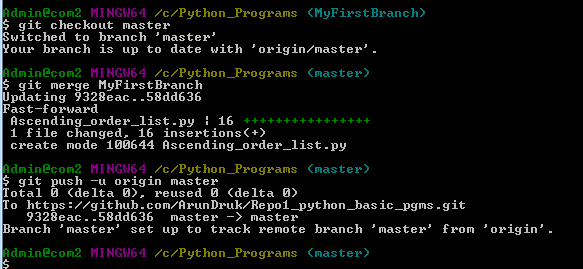
* git branch branch\_name
* git checkout branch\_name
* git push –u origin branch\_name



Merging the branch to the master:

Before merging need to checkout master :

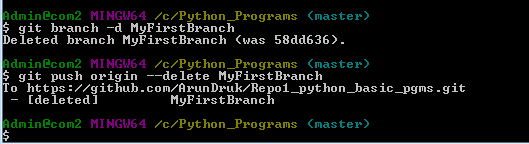
* git checkout master
* git merge MyFirstBranch
* git push –u origin master



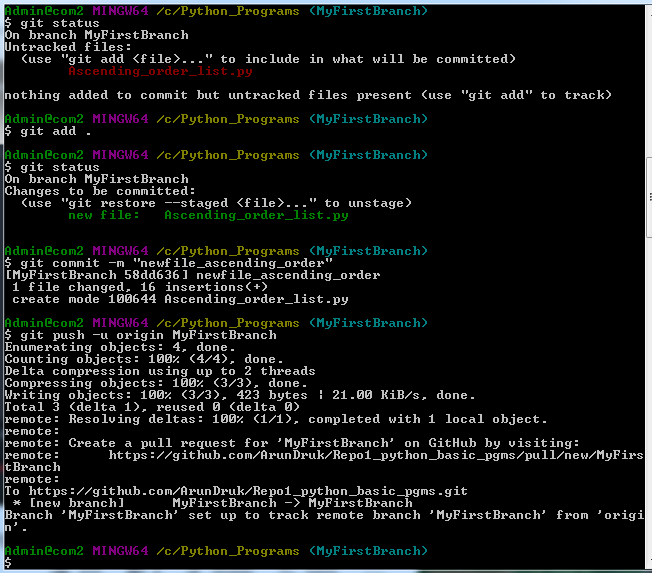
* Deleting Branch:
* Once the branch is created, worked on it and finally merged to the master, then this branch is no longer needed.
* So we can delete this branch.

Below commands to delete branch:

* git branch –d branch\_name 🡪 Deletes from Local
* git push origin –delete branch\_name 🡪 Deletes from Remote.



Below is the Sequence of git commands to commit and push files:



GIT Hub:

* Website to upload your repositories online.
* GIT and GIT Hub are different.

Tags in Git:

* Whenever you want to create a release point for a stable version of your code.
* To create a historic point for your code/data that you can refer at any future time(to restore your data)

Steps for creating tags:

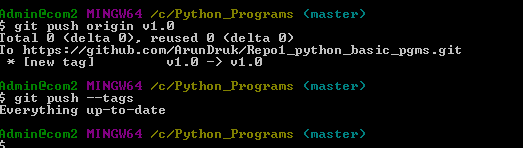
* Checkout the branch where you want to create the tag

git checkout <branch\_name>

* Create the tag : **git tag <tag\_name>**
* To create annotated tag: **git tag –a v1.1 –m “tag for release ver 1.1”** (This gives some information to tag
* **git tag** (This displays the list of tags created)
* **git show v1.0**
* **git tag –l “v1.\*”** (This displays all the tags starting with v1)

Below is pushing tags to remote**:**

* git push origin v1.0
* git push origin --tags
* git push --tags (To push all tags at once)

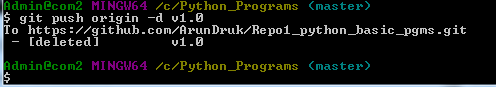


Below is Deleting tags from local:

* git tag --d v1.0 (To delete multiple tags once : git tag --d v1.0 v1.1)
* git tag --delete v1.0
* git tag (To check the tags are deleted)

Below is Deleting tags from remote:

* git push origin –d v1.0 (To delete multiple tags once : git push origin --d v1.0 v1.1)
* git push origin –delete v1.0
* git push origin:v1.0

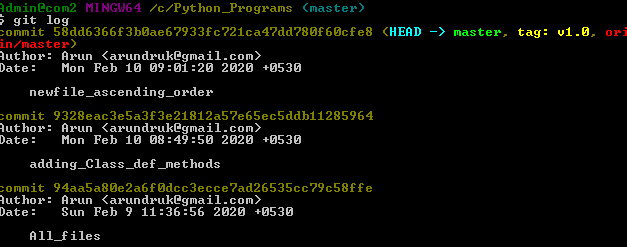


Creating a Branch from a tag :

* We can create a branch from a tag and we can check out and start working on the branch.
* Command: git checkout –b <branch\_name> <tag\_name>

Creating a tag from past commit:

* The commit has the checksum, 40 digits, we can use part of the digits.
* Command: git tag <tag\_name> <reference of commit>
* **git tag v1.1 9328eac3e**

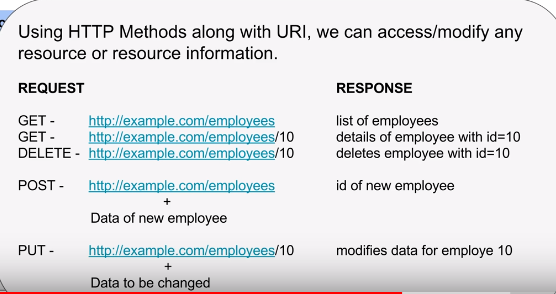


WEB Services:

* Web Service enables communication between applications over the web.
* Provides a standard format/protocol for communication.
* Platform independent communication.
* Third party applications exposed their webservices in the form of API’s and companies like MakemyTrip/ ClearTrip uses to these API to fetch the data in real time.
* SOAP and REST are two WebServices.
* **SOAP** -> Simple Object Access Protocol, Medium is HTTP(POST) and the format is XML .
* A web service that complies/follows to the **SOAP Web Services Specifications** is a SOAP Web Service.
* **W3C -> World wide web Consortium ->** A international Community that defines the standards.
* **SOAP** XML message consist of: **Envelope, Header and a Body**
* **REST -> RE**presentational **S**tate **T**ransfer, Medium is HTTP(POST, GET, PUT. DELETE) and the format is XML/JSON/TEXT…
* **REST** is an architectural style.
* **Uniform Resource Identifier** (URI) is a string of characters used to identify a resource either by location (URL), or a name (URN), or both (URL and URN). A URI has two specializations known as URL (Uniform Resource Locator) and URN (Uniform Resource Name)
* Any information that can be named can be a resource: a document or image and so on
* Any resource or a data can be accessed by a URI.
* **Web Services Description Language – WSDL**

Service provider publishes an interface for his web services that describes all attributes of the web services. This XML basaed interface and is called WSDL.

* A web service provider publishes his web service (through wsdl) on an online directory from where consumers can query and search the web services. This online registry/directory is called **UDDI** stands for **Universal Description, Discovery and Integration**.
* C = CREATE = **POST**
* R= READ=**GET**
* U= UPDATE=**PUT**
* D=DELETE=**DELETE**



* **REST is flexible, light weight and easy compared to SOAP**

# Soap UI:

The link for the WSDL which gives information on the country:

Below is for the SOAP project:

<http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?wsdl>

1. Copy the above URI and paste it in the new SOAP project.

Project >Test Suite > Test case > Test Step

Below is for the REST Project:

URI: <https://restcountries.eu/rest/v2/all>

Properties :

* Properties can be used as variables to store values that can be referred in testing.

${#Project#PropertyName}

POSTMAN tool:

* Can be installed as a Chrome plugin or Download and install as an application
* Install the KB2533623 on windows 7 if you get kernel dl issue.
* Link of KB : <https://www.youtube.com/redirect?v=FUmzPZYF3Yk&redir_token=n532VXcjNtDh3nAkxGmI8KRLPpN8MTU4MjAwNDA1MkAxNTgxOTE3NjUy&event=video_description&q=https%3A%2F%2Fwww.microsoft.com%2Fen-us%2Fdownload%2Fdetails.aspx%3Fid%3D26764>
* Collections is like a Test suite, which contains Requests and tests.
* Workspace lets you to create our path and directory and our requests will be stored in that.
* URL (Uniform resource Locator) = Domain + URI (Uniform resource identifier)

Domain = [www.google.com](http://www.google.com)

URI=search?sxsrf=ACYBGNSldKGB46bqFr-7M5uXL5oB3jUEtg%3A1581942122640&source=hp&ei=aoVKXtnpJLbXsPrWYoAQ&q=selenium&

* Before starting our testing we need - Webservice specification document
* URL is called the Endpoint which is Domain + URI

Page Object Model OR Page chaining Model :

* It is a Design Pattern and an approach used with Data driven Framework.

This is the parent class which contains: driver object initialization, opening browser, maximize window, get url, implicit wait(),delete cookies

Page Layer - l Test Layer – ll

Pytest (Unit testing)

All the classes Loginpage, HomePage, Test\_LoginPage is all child class

Test\_LoginPage

Login Page

er

wwe

Config.properties Class. (iv)

All Environment variables will be defined here, like URL, username, passwd, browser properties etc.

Test\_Homepage

Home Page

Test\_SearchPage

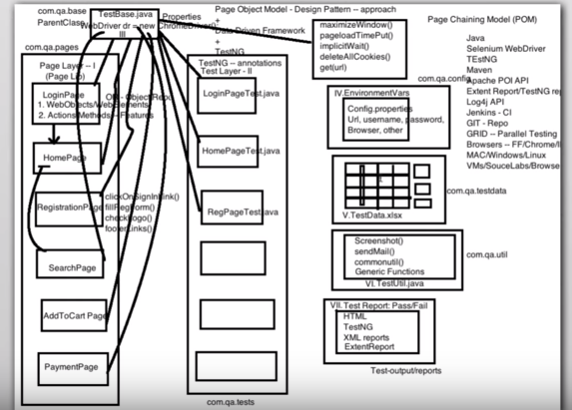
Search Page

Test report generation:Pass, fail, HTML report (VII)

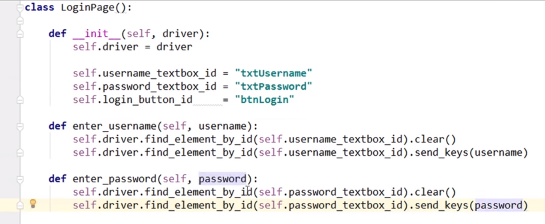
TestUtility class: Here all the common utilities like taking screenshot, sendmail(), Generic functions (VI)

TestData.xls : To define test data in a excel sheet. (v)

* In the abv diagram, Once we do login, the page will be directed to Home Page and from Home Page using Search element we can go to Search Page and all the pages are interconnected, so it is also called as Page Chaining model.
* In the application we’ll be having different pages and creating a Class for each page, suppose there are 100 pages in an application we need to create 100 Classes.
* Whatever you see on a webpage is called a web element in selenium
* Different click or links is Actions/Methods called as Features.
* Different methods in the Class say: clickOnSignInLink(), fillRegForm(), checkLogo(), footerLinks()
* First create Page Layer indicating how many pages are there in the application which is class, then create all the webElements for that class or webpage
* Collection of all the web elements or web objects is called as Object Repository (OR)



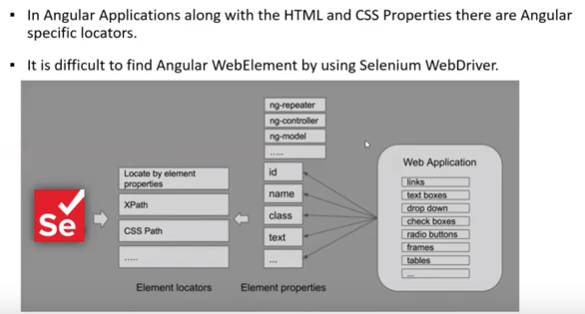
* Create a Class for Login page and in the constructor method define the entire web elements in that login page.
* Then Create a function for each actions like “Enter Username”, “Enter Password” and “Click\_Login”





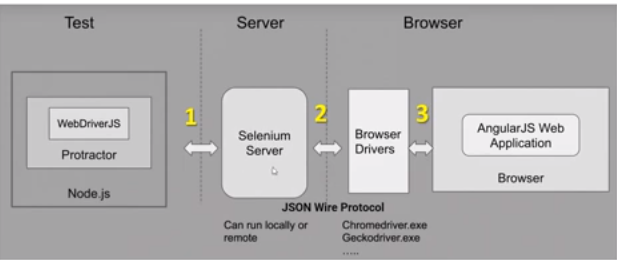
Protractor:

* Protractor is an End-End testing framework, used for automating web applications developed on Angular and Angular JS
* Angular is developed in Sep 2016 by Google and Angular JS was developed in 2010
* ng-model, ng-controller these locators can’t be found using selenium webdriver if the web application is developed in Angular / Angular JS
* Protractor is a wrapper over Selenium webdriverJS and provides support to find angular specific elements.



* How Protractor works ??

Below diagram explain that flow.



JMeter:

Step 1 : Unzip and keep Jmeter folder at any location

Step 2 : Start Jmeter Windows - jmeter/bin - jmeter.bat