MAVEN AND GIT ASSIGNMENT

1. WHAT IS MAVEN LIFE CYCLE AND IT’S COMMANDS?

ANS: Maven is a powerful project management tool that is based on POM (project object model), used for projects build, dependency and documentation. It is a tool that can be used for building and managing any Java-based project. Maven makes the day-to-day work of Java developers easier and helps with the building and running of any Java-based project.

**Maven Commands:**

* **mvn clean:** Cleans the project and removes all files generated by the previous build.
* **mvn compile:** Compiles source code of the project.
* **mvn test-compile:** Compiles the test source code.
* **mvn test:** Runs tests for the project.
* **mvn package:** Creates JAR or WAR file for the project to convert it into a distributable format.
* **mvn install:** Deploys the packaged JAR/ WAR file to the local repository.
* **mvn deploy:** Copies the packaged JAR/ WAR file to the remote repository after compiling, running tests and building the project.

2.

* mvn --version : This command helps us in **knowing the current version of Maven that is installed.**
* **mvn --compile :**  It compiles the source code, converts the .java files to .class and stores the classes in target/classes folder.
* **mvn** --test : It runs unit tests for the project.
* **mvn --**install : This step installs the packaged code to the local Maven repository.
* **mvn –**clean : Cleans the project and removes all files generated by the previous build.

3.WHAT IS WEB SERVICES?

ANS:

* A web service is any piece of software that makes itself available over the internet and uses a standardized XML messaging system.
* XML is used to encode all communications to a web service. For example, a client invokes a web service by sending an XML message, then waits for a corresponding XML response.
* As all communication is in XML, web services are not tied to any one operating system or programming language—Java can talk with Perl; Windows applications can talk with Unix applications.

4.WHAT IS REST CONTROLLER?

ANS: RestController is used for making restful web services with the help of the @RestController annotation. This annotation is used at the class level and allows the class to handle the requests made by the client

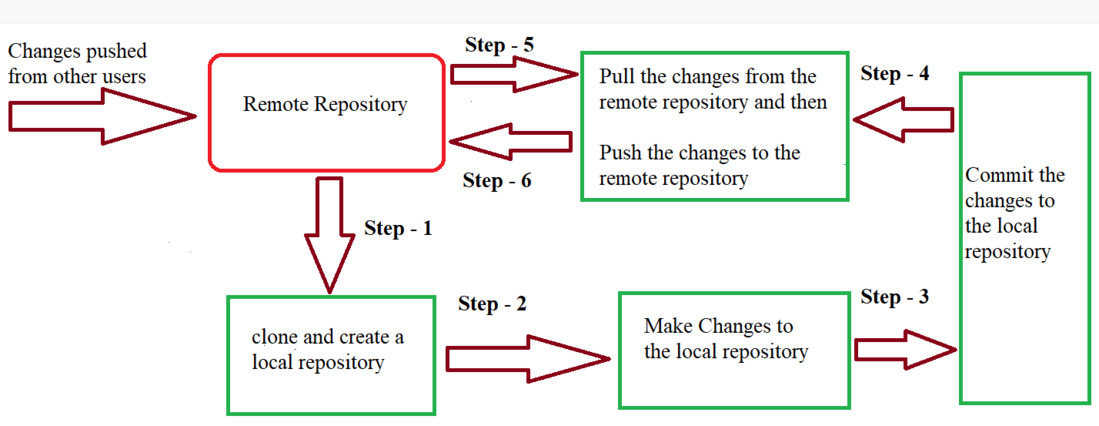
5.WEB SERVICES AND ITS METHOD.

ANS:

|  |  |
| --- | --- |
| 1. | GET - The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data |
| 2. | POST - A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms. |
| 3. | PUT - Replaces all current representations of the target resource with the uploaded content. |
| 4. | DELETE - Removes all current representations of the target resource given by a URI. |

6.WHAT IS GIT LIFECYCLE?

ANS : Git is used in our day-to-day work, we use git for keeping a track of our files, working in a collaboration with our team, to go back to our previous code versions if we face some error, Git helps us in many ways.



* **Step 1 -** We first clone any of the code residing in the remote repository to make our own local repository.
* **Step 2 -**  we edit the files that we have cloned in our local repository and make the necessary changes in it.
* **Step 3** - we commit our changes by first adding them to our staging area and committing them with a commit message.
* **Step 4 and Step 5 -** we first check whether there are any of the changes done in the remote repository by some other users and we first pull that changes.
* If there are no changes we directly proceed with **Step 6** in which we push our changes to the remote repository and we are done with our work.

7.GIT COMMANDS:

* GIT INIT COMMAND - This command is used to create a local repository.
* GIT ADD COMMAND - This command is used to add one or more files to staging (Index) area.
* GIT COMMIT COMMAND - Commit command is used in two scenarios. They are as follows.

**Git commit -m -** This command commits any files added in the repository with git add and also commits any files you've changed since then.

**Git commit -a -** This command commits any files added in the repository with git add and also commits any files you've changed since then.

* GIT PUSH COMMAND - It is used to upload local repository content to a remote repository. Pushing is an act of transfer commits from your local repository to a remote repository.

Git push command can be used as follows:

**Git push origin master -** This command sends the changes made on the master branch, to your remote repository.

**Git push -all -** This command pushes all the branches to the server repository.

* GIT PULL COMMAND - Pull command is used to receive data from GitHub. It fetches and merges changes on the remote server to your working directory.
* GIT LOG COMMAND - This command is used to check the commit history.
* GIT MERGE COMMAND - This command is used to merge the specified branch?s history into the current branch.
* GIT BRANCH COMMAND - This command lists all the branches available in the repository.
* GIT CHECKOUT COMMAND - It checks the branches and updates the files in the working directory to match the version already available in that branch, and it forwards the updates to Git to save all new commit in that branch.
* GIT REVERT COMMAND - It is an undo type command. It can be useful for tracking bugs in the project.
* GIT RESET COMMAND - The git reset command is used to reset the changes.
* GIT REBASE COMMAND -  It is used to apply a sequence of commits from distinct branches into a final commit. It is an alternative of git merge command.

8.CENTRALISED AND DISTRIBUTED VERSION CONTROL

|  |  |
| --- | --- |
| CENTRALISED VERSION CONTROL | DISTRIBUTED VERSION CONTROL |
| * In centralized source control, there is a server and a client.The server is the master repository that contains all of the versions of the code. * To work on any project, firstly user or client needs to get the code from the master repository or server. So the client communicates with the server and pulls all the code or current version of the code from the server to their local machine. * In other terms we can say, you need to take an update from the master repository and then you get the local copy of the code in your system. * So once you get the latest version of the code, you start making your own changes in the code and after that, you simply need to commit those changes straight forward into the master repository. * Committing a change simply means merging your own code into the master repository or making a new version of the source code. So everything is centralized in this model. * There will be just one repository and that will contain all the history or version of the code and different branches of the code. * So the basic workflow involves in the centralized source control is getting the latest version of the code from a central repository that will contain other people’s code as well, making your own changes in the code, and then committing or merging those changes into the central repository. | * In distributed version control most of the mechanism or model applies the same as centralized. * The only major difference you will find here is, instead of one single repository which is the server, here every single developer or client has their own server and they will have a copy of the entire history or version of the code and all of its branches in their local server or machine. * Basically, every client or user can work locally and disconnected which is more convenient than centralized source control and that’s why it is called distributed. * You don’t need to rely on the central server, you can clone the entire history or copy of the code to your hard drive. * So when you start working on a project, you clone the code from the master repository in your own hard drive, then you get the code from your own repository to make changes and after doing changes, * you commit your changes to your local repository and at this point, your local repository will have ‘change sets‘ but it is still disconnected with the master repository (master repository will have different ‘**sets of changes**‘ from each and every individual developer’s repository), * so to communicate with it, you issue a request to the master repository and push your local repository code to the master repository. |

