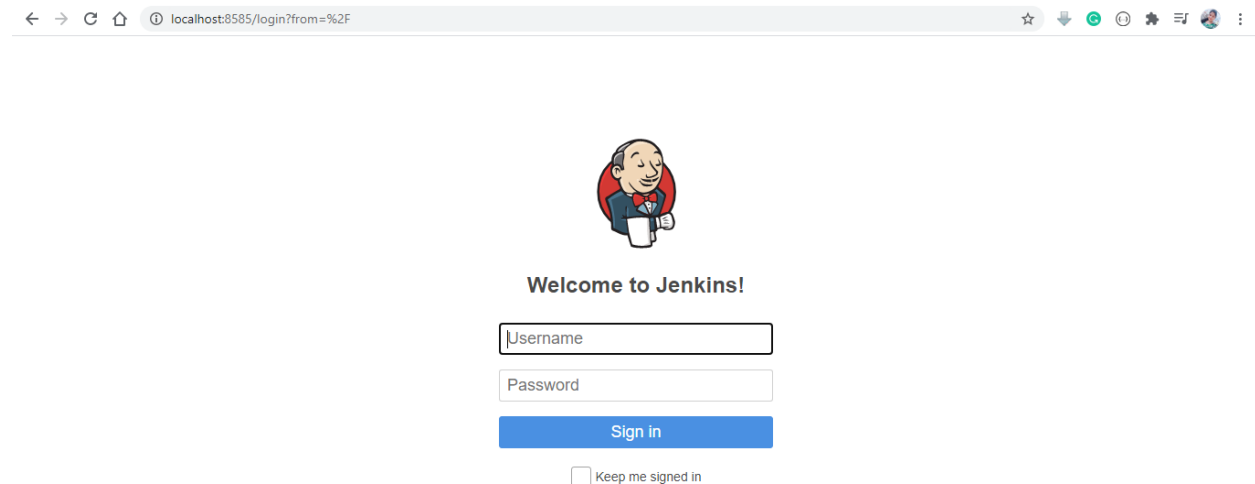


Step 1: Download & Install jenkins on your system.

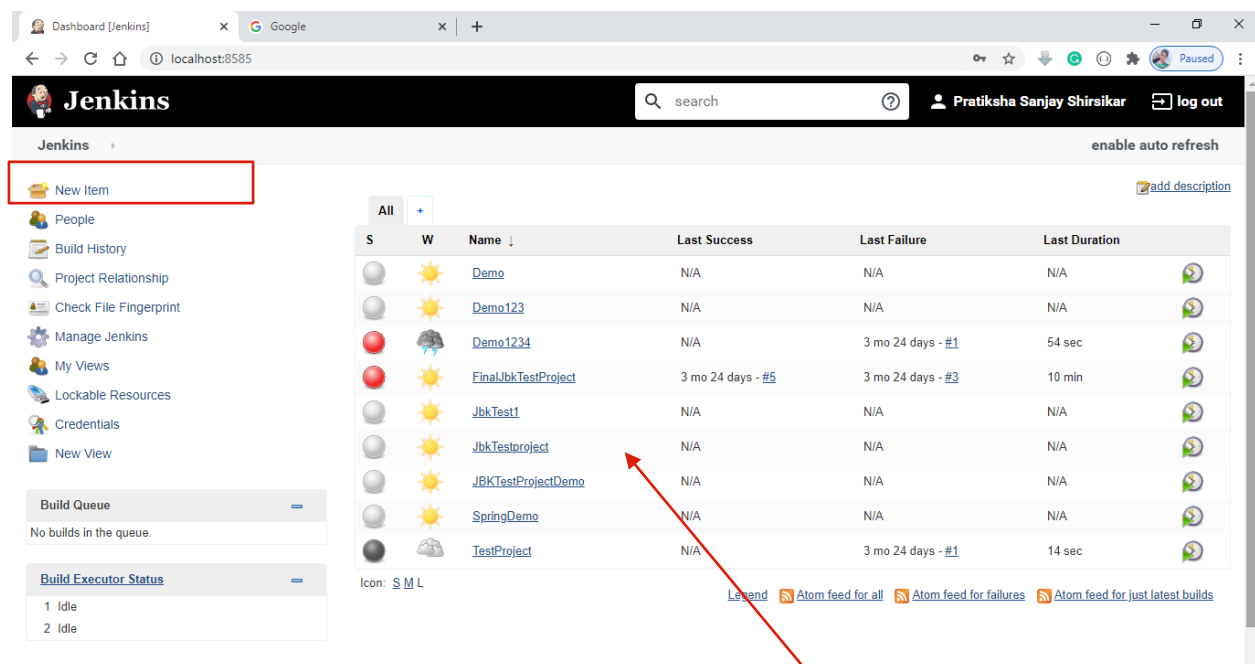
Step 2 :Start service of jenkins.

Step 3 Open jenkins dashboard with jenkins username & password credentials.



The image shows the Jenkins login page in a web browser. The address bar displays 'localhost:8585/login?from=%2F'. The page features the Jenkins logo (a cartoon man with a red bow tie) and the text 'Welcome to Jenkins!'. Below this, there are input fields for 'Username' and 'Password', followed by a blue 'Sign in' button. At the bottom, there is a checkbox labeled 'Keep me signed in'.

Step 4: Jenkin Dashboard will open.

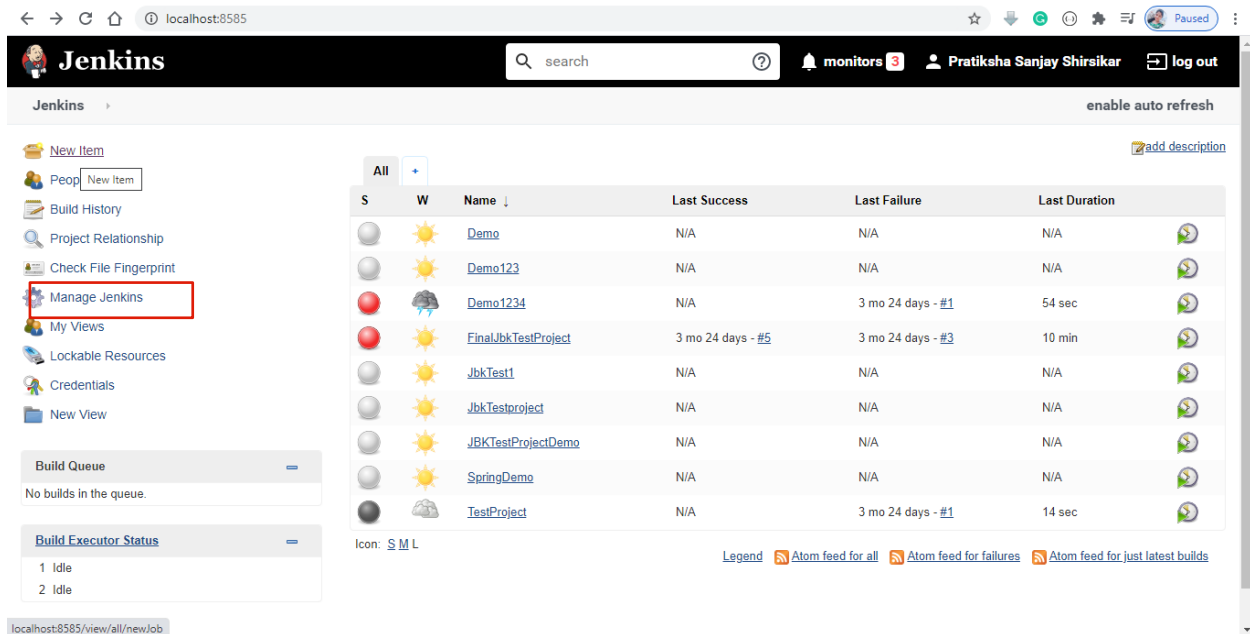


The image shows the Jenkins Dashboard in a web browser. The address bar displays 'localhost:8585'. The dashboard has a dark header with the Jenkins logo, a search bar, and the user 'Pratiksha Sanjay Shirsikar' with a 'log out' button. On the left, there is a sidebar with various links, including 'New Item' which is highlighted with a red box. The main area displays a table of jobs. A red arrow points to the 'JbkTestProject' job in the table.

S	W	Name	Last Success	Last Failure	Last Duration
		Demo	N/A	N/A	N/A
		Demo123	N/A	N/A	N/A
		Demo1234	N/A	3 mo 24 days - #1	54 sec
		FinalJbkTestProject	3 mo 24 days - #5	3 mo 24 days - #3	10 min
		JbkTest1	N/A	N/A	N/A
		JbkTestProject	N/A	N/A	N/A
		JBKTestProjectDemo	N/A	N/A	N/A
		SpringDemo	N/A	N/A	N/A
		TestProject	N/A	3 mo 24 days - #1	14 sec

List of jobs you have created earlier.

Step 5: Select Manage Jenkins for managing configuration of jdk, git and maven.

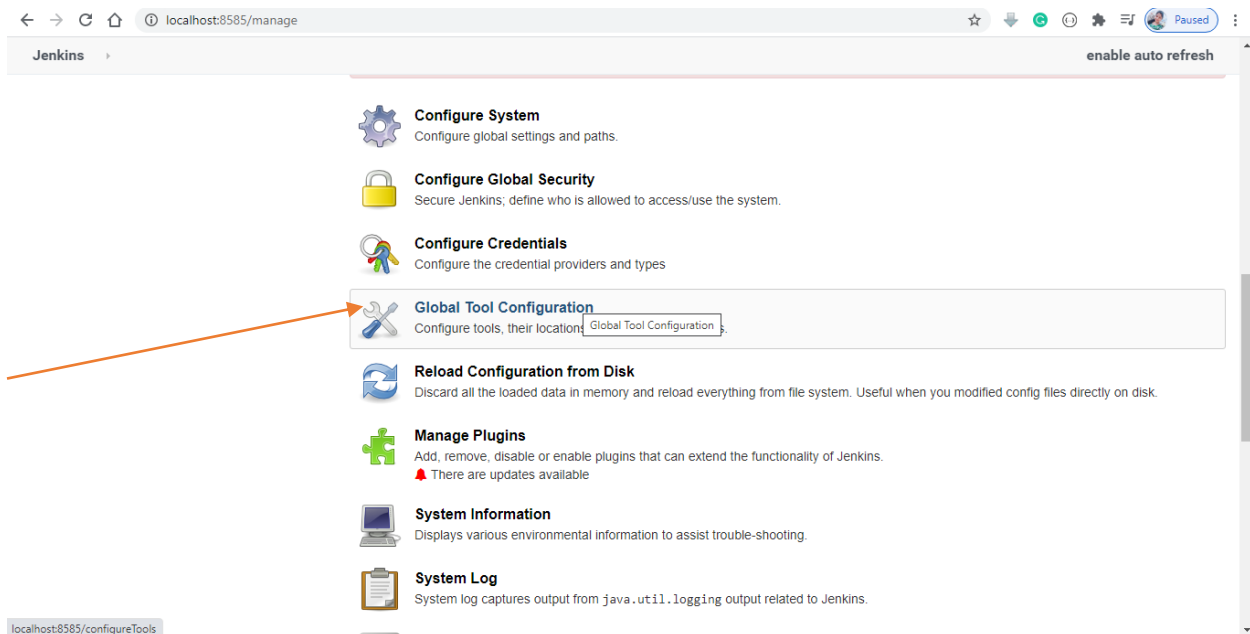


The screenshot shows the Jenkins dashboard at localhost:8585. The left sidebar contains a list of navigation options: New Item, People, Build History, Project Relationship, Check File Fingerprint, **Manage Jenkins** (highlighted with a red box), My Views, Lockable Resources, Credentials, and New View. Below the sidebar, there are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main content area displays a table of jobs with columns: S, W, Name, Last Success, Last Failure, and Last Duration. The table lists several jobs including Demo, Demo123, Demo1234, FinalJbkTestProject, JbkTest1, JbkTestproject, JBKTestProjectDemo, SpringDemo, and TestProject. At the bottom of the sidebar, the URL localhost:8585/view/all/newJob is visible.

S	W	Name	Last Success	Last Failure	Last Duration
		Demo	N/A	N/A	N/A
		Demo123	N/A	N/A	N/A
		Demo1234	N/A	3 mo 24 days - #1	54 sec
		FinalJbkTestProject	3 mo 24 days - #5	3 mo 24 days - #3	10 min
		JbkTest1	N/A	N/A	N/A
		JbkTestproject	N/A	N/A	N/A
		JBKTestProjectDemo	N/A	N/A	N/A
		SpringDemo	N/A	N/A	N/A
		TestProject	N/A	3 mo 24 days - #1	14 sec

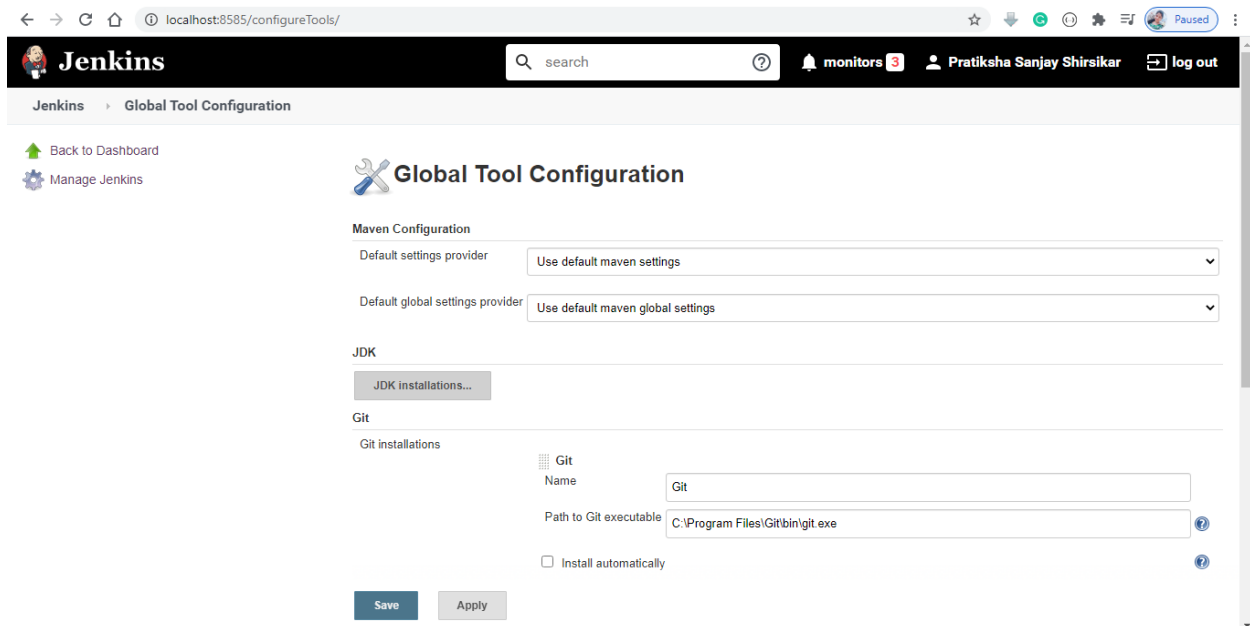
Step 6: After that select Manage plugins to add new plugin.

Step 7: Select Global tool Configuration.



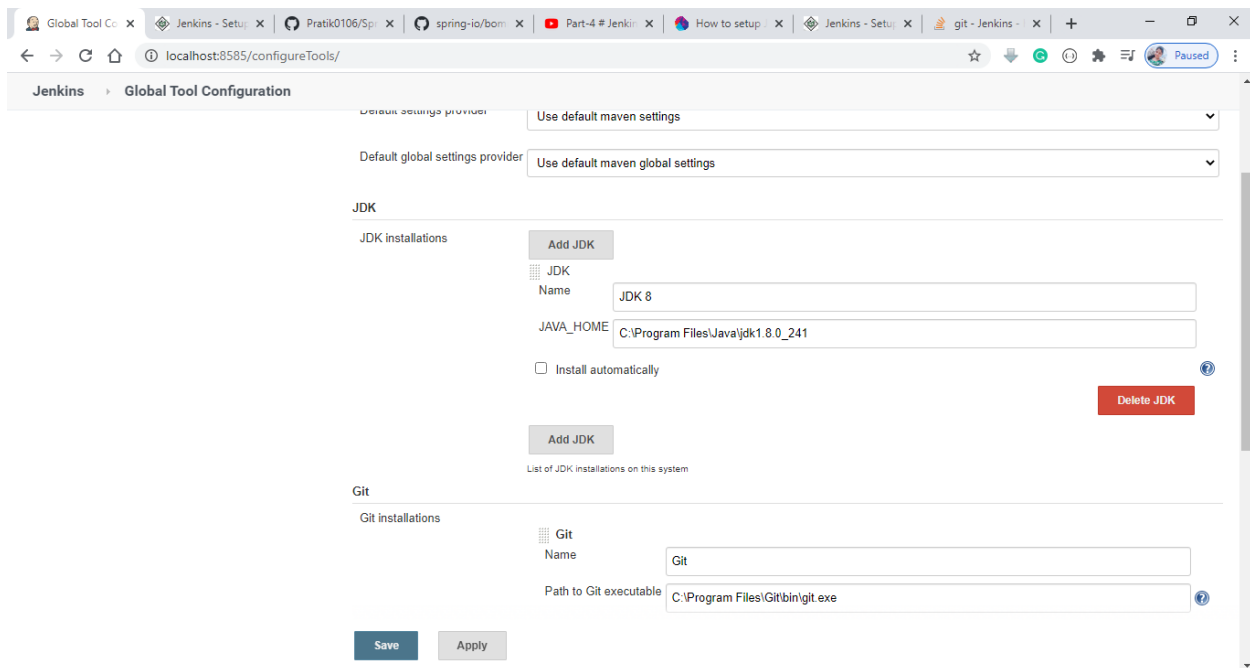
The screenshot shows the 'Manage Jenkins' page at localhost:8585/manage. The page lists several configuration options: Configure System, Configure Global Security, Configure Credentials, **Global Tool Configuration** (highlighted with an orange arrow), Reload Configuration from Disk, Manage Plugins, System Information, and System Log. The 'Global Tool Configuration' option is selected, and its sub-page 'Global Tool Configuration' is visible in the breadcrumb trail.

Step 8: Click on jdk installation for jdk configuration.



The screenshot shows the Jenkins 'Global Tool Configuration' page. The 'Maven Configuration' section has two dropdown menus: 'Default settings provider' and 'Default global settings provider', both set to 'Use default maven settings'. The 'JDK' section has a button labeled 'JDK installations...'. The 'Git' section has a 'Git' table with one entry: 'Name' is 'Git' and 'Path to Git executable' is 'C:\Program Files\Git\bin\git.exe'. There is an unchecked checkbox for 'Install automatically'. At the bottom are 'Save' and 'Apply' buttons.

Step 9: Paste java jdk path to the JAVA_HOME and give name to it.



The screenshot shows the Jenkins 'Global Tool Configuration' page with the 'JDK' section expanded. An 'Add JDK' button is visible. Below it, a table shows a new entry: 'Name' is 'JDK 8' and 'JAVA_HOME' is 'C:\Program Files\Java\jdk1.8.0_241'. There is an unchecked checkbox for 'Install automatically' and a red 'Delete JDK' button. Below the table is another 'Add JDK' button and a note 'List of JDK installations on this system'. The 'Git' section is also visible at the bottom.

Step 10: If git is already installed your machine then u can just provide the path to Git executable path otherwise u can select git to install automatically.

The screenshot shows the Jenkins 'Global Tool Configuration' page. The browser address bar indicates the URL is `localhost:8585/configureTools/`. The page title is 'Jenkins > Global Tool Configuration'. The main content area is divided into sections for different tools:

- JDK:** At the top right is a red 'Delete JDK' button. Below it is a grey 'Add JDK' button. A note says 'List of JDK installations on this system'.
- Git:** This section has a 'Git installations' header. It contains a table with one entry: 'Git'. The 'Name' field is 'Git' and the 'Path to Git executable' field is 'C:\Program Files\Git\bin\git.exe'. There is an unchecked checkbox for 'Install automatically'. To the right of the table is a red 'Delete Git' button. Below the table is a grey 'Add Git' button with a dropdown arrow.
- Gradle:** This section has a 'Gradle installations' header. It contains a grey 'Add Gradle' button. A note says 'List of Gradle installations on this system'.
- Ant:** This section has an 'Ant installations' header. It contains a grey 'Add Ant' button.

At the bottom of the page are two buttons: 'Save' (dark blue) and 'Apply' (grey).

Step 11: Click on add maven and choose version to get download automatically.

Step 12: Click on Apply and Save button.

The screenshot shows the Jenkins 'Global Tool Configuration' page for Maven. The 'Maven' section is active, showing a list of Maven installations. A new installation is being added with the name 'Maven'. The 'Install automatically' checkbox is checked. The 'Install from Apache' section shows the version '3.3.9'. There are buttons for 'Add Ant', 'Add Maven', 'Delete Installer', and 'Delete Maven'. At the bottom, there are 'Save' and 'Apply' buttons.

Ant

Ant Installations

Add Ant

List of Ant installations on this system

Maven

Maven Installations

Add Maven

Maven

Name

Maven

☒ Install automatically

☐ Install from Apache

Version

3.3.9

Delete Installer

Add Installer

Add Maven

List of Maven installations on this system

Save

Apply

Step 13: Create new job by selecting New Item menu.

The screenshot shows the Jenkins main dashboard. The 'New Item' menu is highlighted in the left sidebar. The main area displays a table of jobs with columns for Status (S), Weather (W), Name, Last Success, Last Failure, and Last Duration. The table lists several jobs, including 'Demo', 'Demo123', 'Demo1234', 'FinalJbkTestProject', 'JbkTest1', 'JbkTestProject', 'JbkTestProjectDemo', 'SpringDemo', and 'TestProject'. The 'TestProject' job is highlighted. The bottom of the page shows the 'Build Queue' and 'Build Executor Status' sections.

Jenkins

enable auto refresh

New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Lockable Resources

Credentials

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

All

S	W	Name	Last Success	Last Failure	Last Duration
		Demo	N/A	N/A	N/A
		Demo123	N/A	N/A	N/A
		Demo1234	N/A	3 mo 24 days - #1	54 sec
		FinalJbkTestProject	3 mo 24 days - #5	3 mo 24 days - #3	10 min
		JbkTest1	N/A	N/A	N/A
		JbkTestProject	N/A	N/A	N/A
		JbkTestProjectDemo	N/A	N/A	N/A
		SpringDemo	N/A	N/A	N/A
		TestProject	N/A	3 mo 24 days - #1	14 sec

Icon: S M L

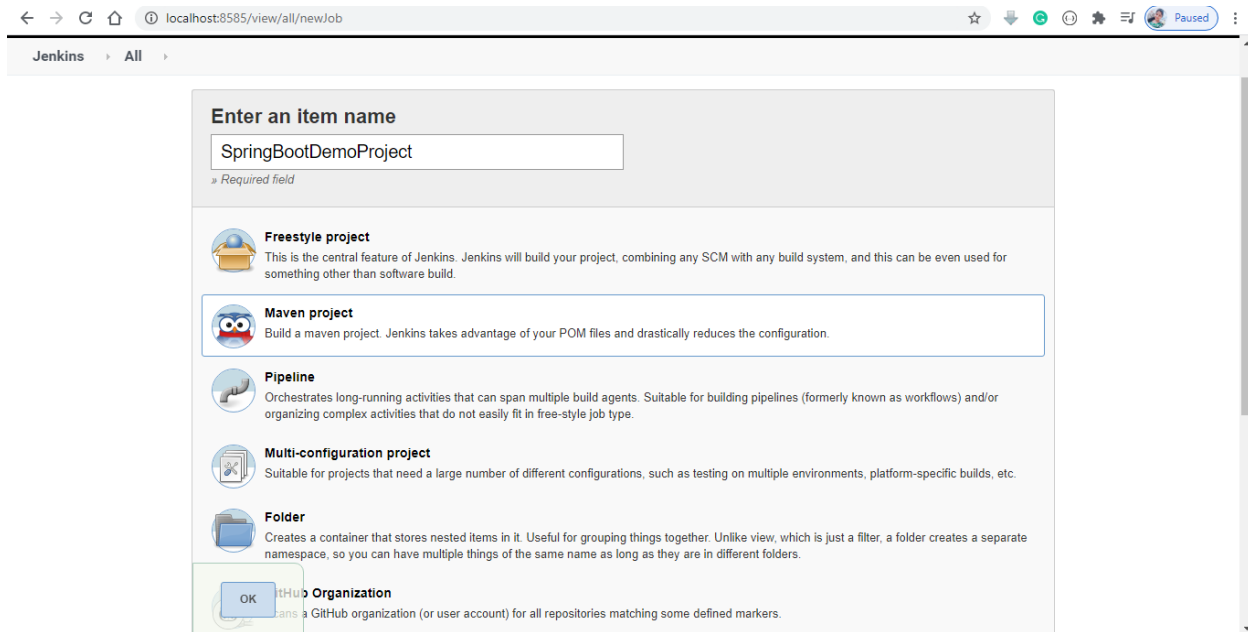
Legend

Atom feed for all

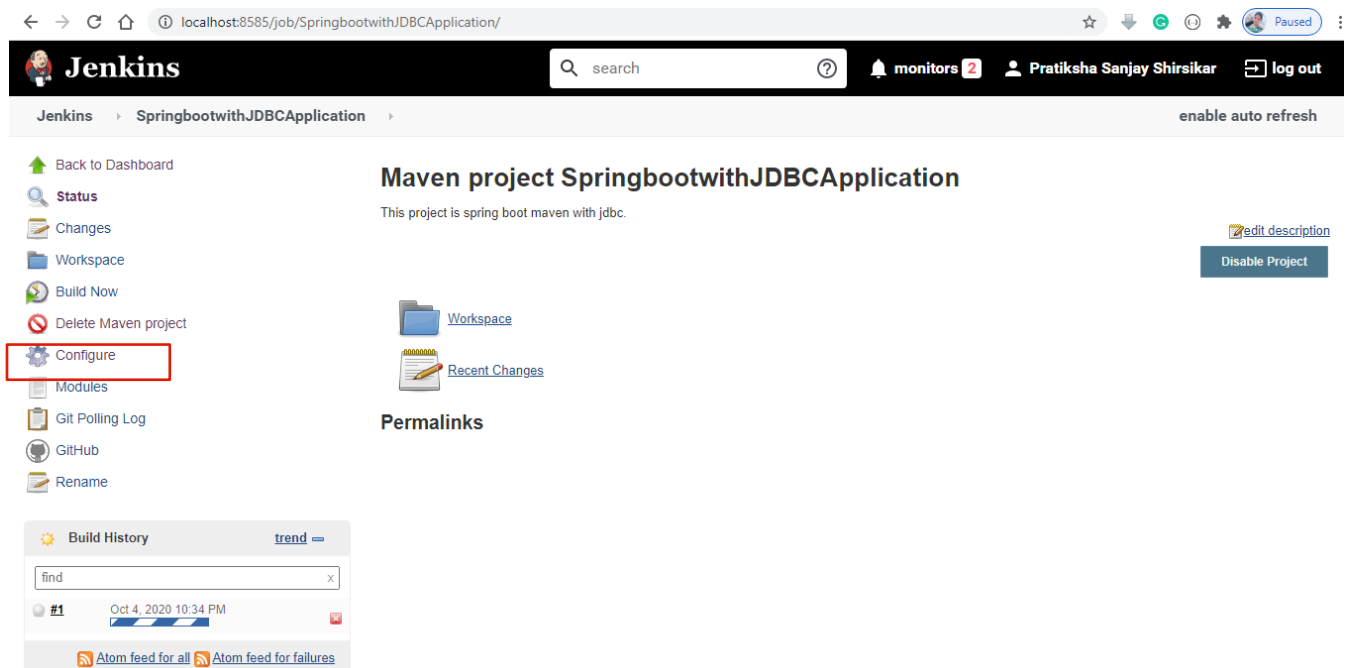
Atom feed for failures

Atom feed for just latest builds

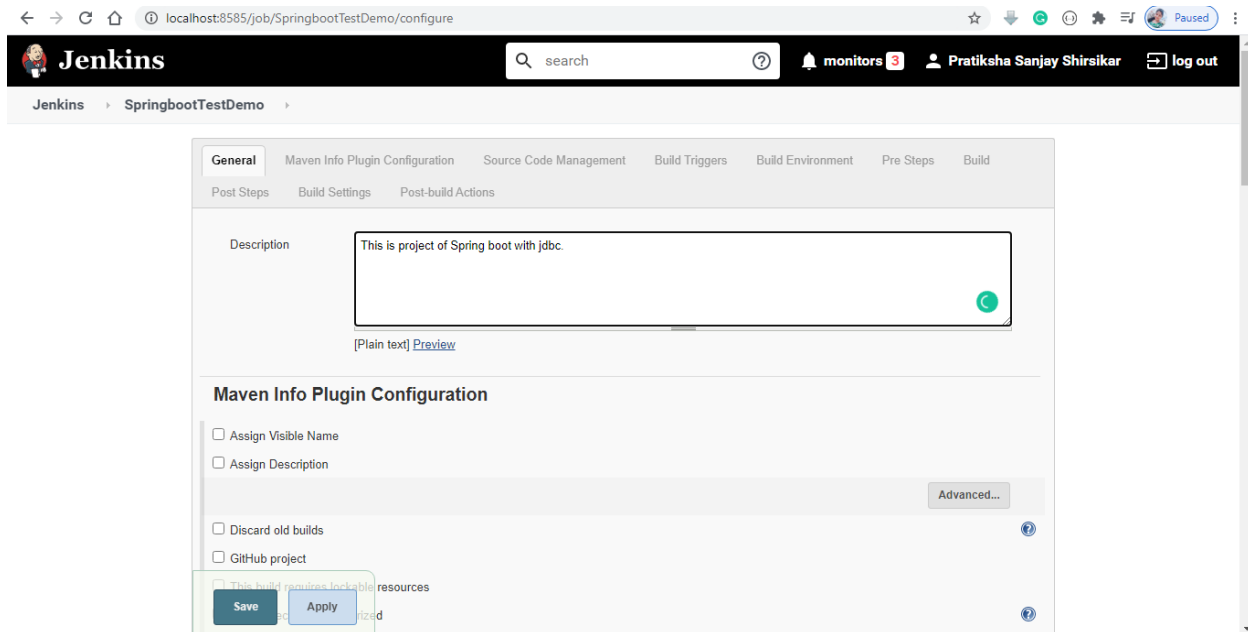
Step 14: Give name to your job. E,g SpringbootDemoProject and choose Maven project and click on Ok.



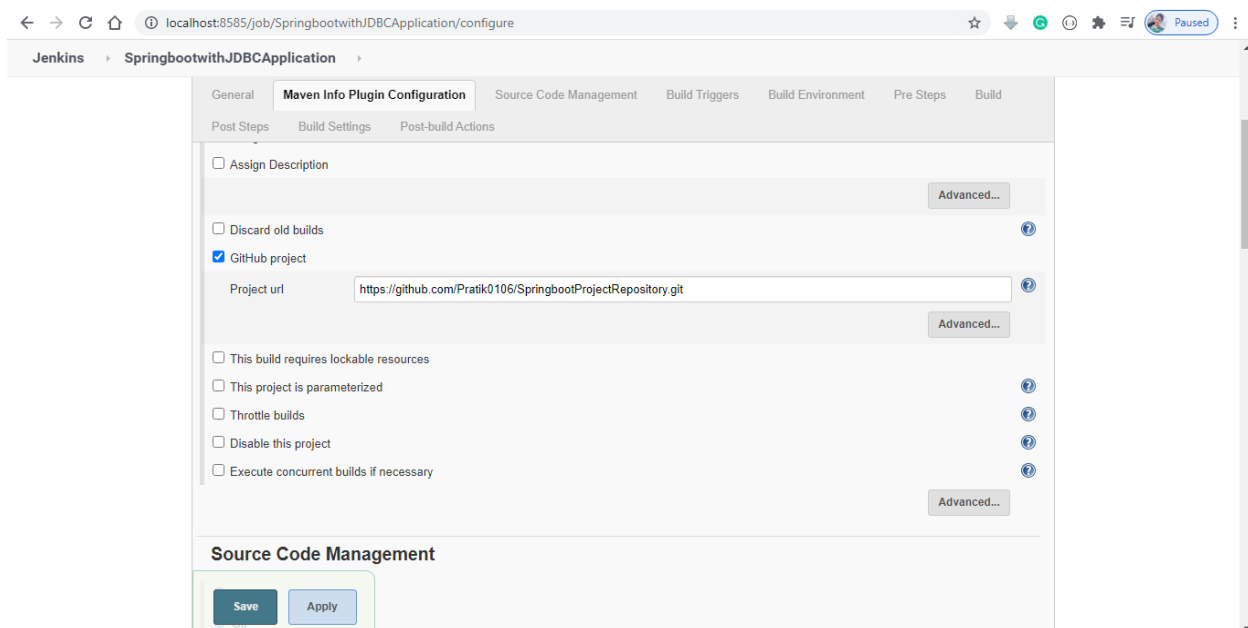
Step 15: Now configure your project by clicking on configure menu.



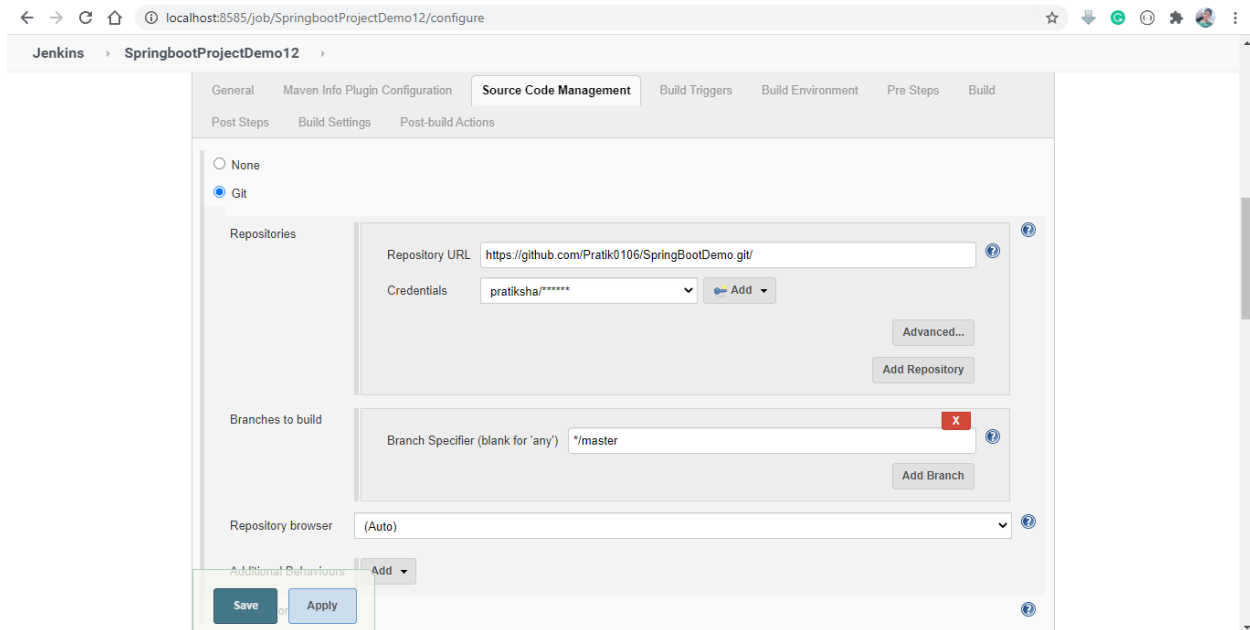
Step 16: Add project description into description area.



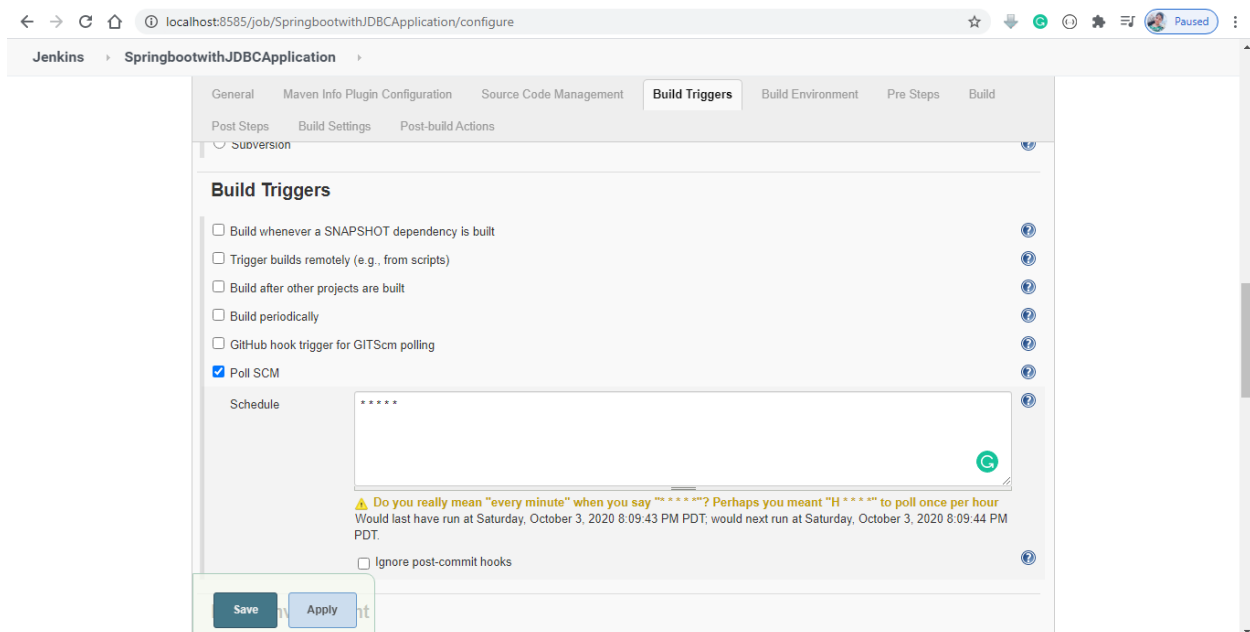
Step 17: Add your github repository path into Project url.



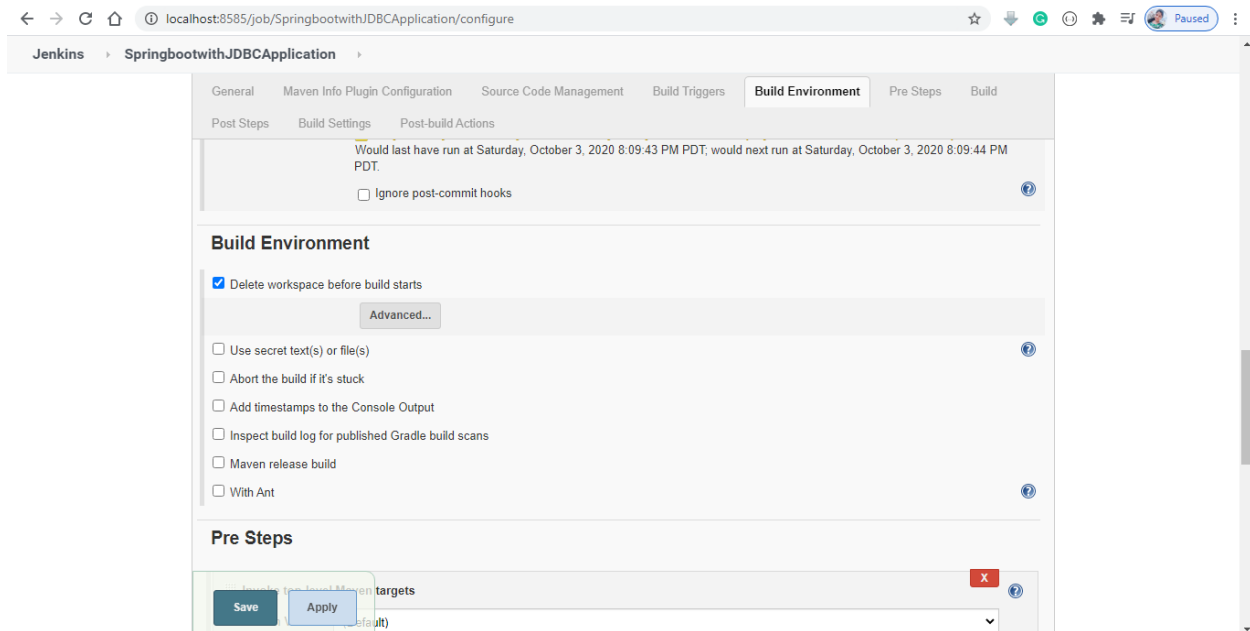
Step 18: At the Source code management section . Select git button and add repository url and add credentials of jenkins.



Step 19: At the Build Triggers select on Poll SCM and give 5 * to schedule for every minute.



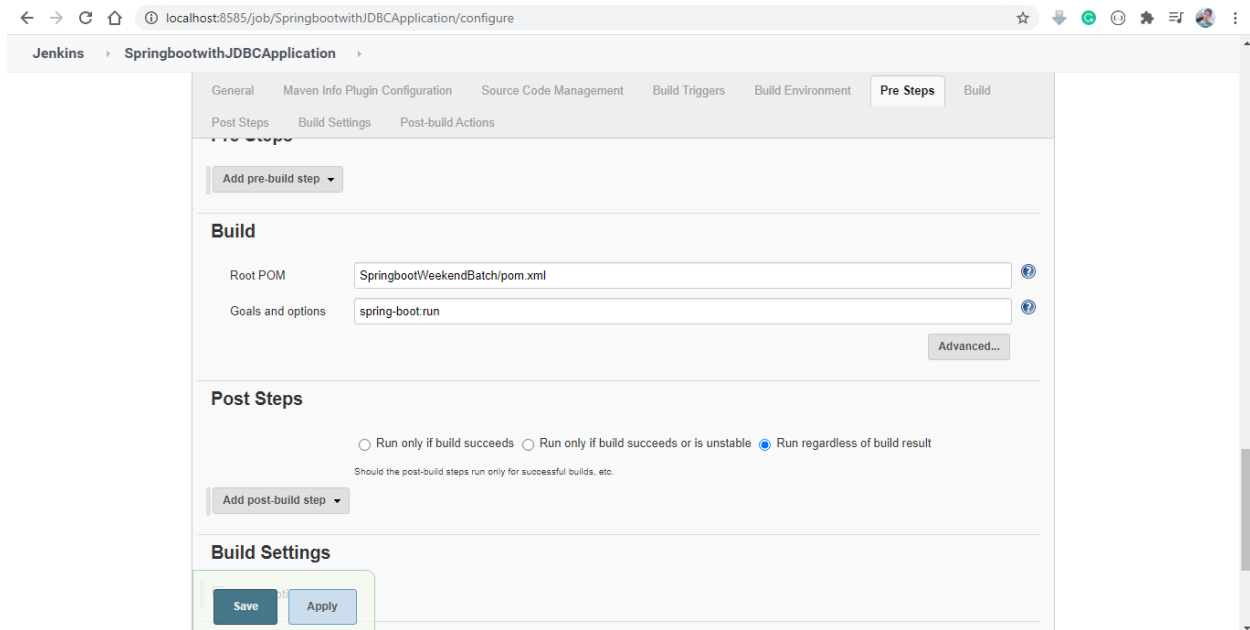
Step 20: Select Delete workspace before build starts at Build environment.



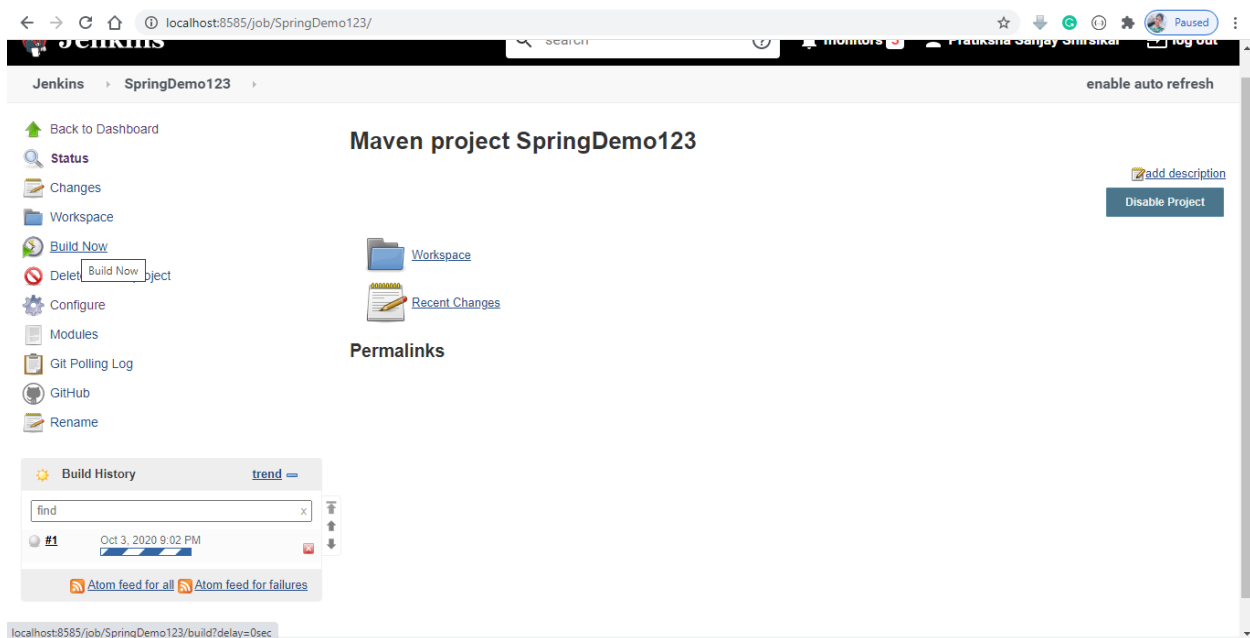
Step 21 At the build section, Root pom give your pom.xml path to create jar or war of project according to pom.xml. (You have to provide your project name to resolve error like. No such 'pom.xml')

Step 22: Give spring-boot:run command at Goals to create jar of your project. And run project.

Step 23: Click on Apply and Save.



Step 24: Click on build now.



Step 26: You can check your output by selecting console output option.

Spring boot project running on 9999 port

So the Output of first api will be :-

```
▼ [
  {
    "id": 1,
    "name": "ABC",
    "departmentit": "IT",
    "phoneno": "78965471258",
    "createddtm": "2020-01-21",
    "createdby": "kiran",
    "updateddtm": "2020-01-21",
    "updatedby": "kiran",
    "status": "active",
    "cid": 1
  },
  {
    "id": 2,
    "name": "pqr",
    "departmentit": "IT",
    "phoneno": "78965471258",
    "createddtm": "2020-01-20",
    "createdby": "kiran",
    "updateddtm": "2020-01-20",
    "updatedby": "kiran",
    "status": "inactive",
    "cid": 2
  },
  {
    "id": 3,
    "name": "Richa",
    "departmentit": "IT",
    "phoneno": "78965471258",
    "createddtm": "2020-09-30",
    "createdby": "kiran",
    "updateddtm": "2020-01-20",
    "updatedby": "kiran",
    "status": "active",
    "cid": 1
  }
]
```

Raw Parsed