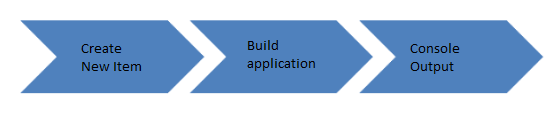
**What is a Jenkins freestyle project?**

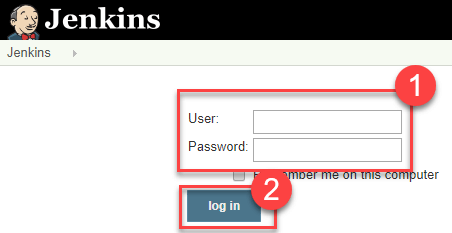
A Jenkins project is a repeatable build job which contains steps and post-build actions. The types of actions you can perform in a build step or post-build action are quite limited. There are many standard plugins available within a Jenkins freestyle project to help you overcome this problem. They allow you to configure build triggers and offers project-based security for your Jenkins project.

[](https://www.guru99.com/images/1/091318_0458_HowtoCreate1.png)

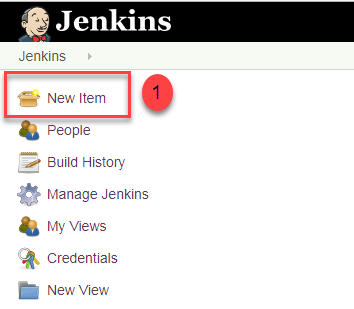
**Creating a Freestyle Build Job**

The freestyle build job is a highly flexible and easy-to-use option. You can use it for any type of project; it is easy to set up, and many of its options appear in other build jobs.

**Step 1)**To create a Jenkins freestyle job, log on to your Jenkins dashboard by visiting your Jenkins installation path. Usually, it will be hosted on localhost at [http://localhost:8080](http://localhost:8080/) If you have installed Jenkins in another path, use the appropriate URL to access your dashboard.

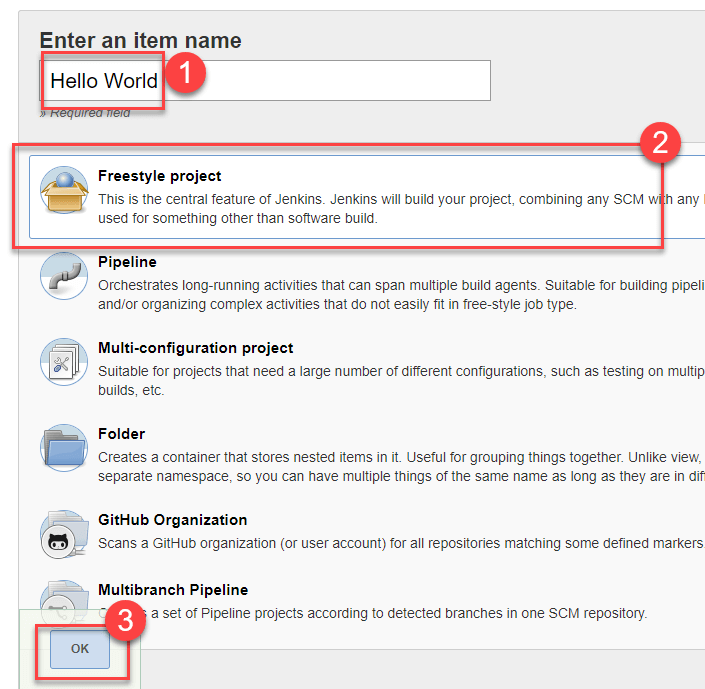
[](https://www.guru99.com/images/1/091318_0458_HowtoCreate2.png)

**Step 2)**Click on "**New Item**" at the top left-hand side of your dashboard.

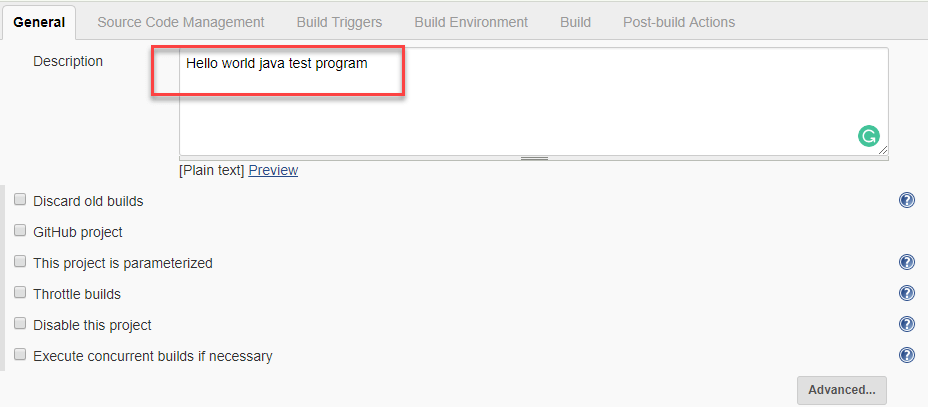
[](https://www.guru99.com/images/1/091318_0458_HowtoCreate3.png)

**Step 3)**In the next screen,

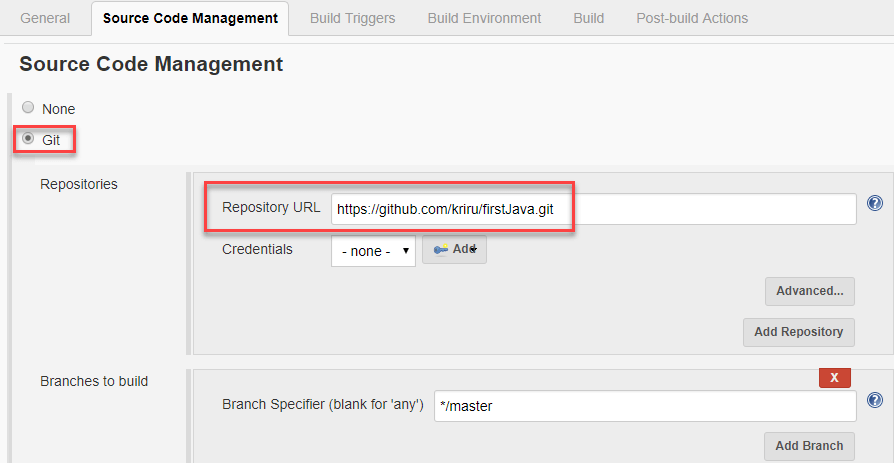
1. Enter the name of the item you want to create. We shall use the "Hello world" for this demo.
2. Select Freestyle project
3. Click Okay

[](https://www.guru99.com/images/1/091318_0458_HowtoCreate4.png)

**Step 4)**Enter the details of the project you want to test.

[](https://www.guru99.com/images/1/091318_0458_HowtoCreate5.png)

**Step 5)**Under Source Code Management, Enter your repository URL

[](https://www.guru99.com/images/1/091318_0458_HowtoCreate6.png)

It is also possible for you to use a local repository.

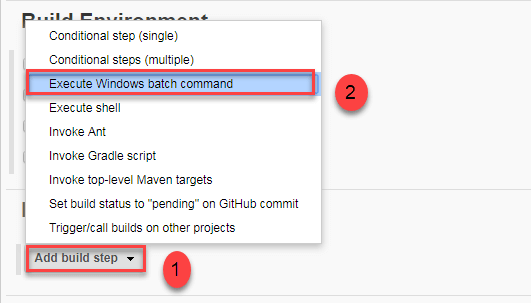
If your GitHub repository is private, Jenkins will first validate your login credentials with GitHub and only then pull the source code from your GitHub repository.

**Step 6)**Now that you have provided all the details, it's time to build the code. Tweak the settings under the **build** section to build the code at the time you want. You can even schedule the build to happen periodically, at set times.

Under **build**,

1. Click on "**Add build step**"

2. Click on "**Execute Windows batch command**" and add the commands you want to execute during the build process.

[](https://www.guru99.com/images/1/091318_0458_HowtoCreate7.png)

Here, I have added the java commands to compile the java code.

I have added the following windows commands:

javac HelloWorld.java

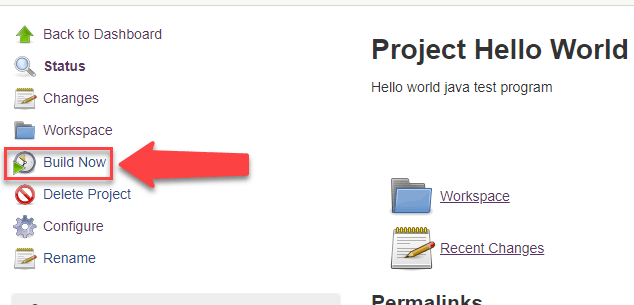
java HelloWorld

[](https://www.guru99.com/images/1/091318_0458_HowtoCreate8.png)

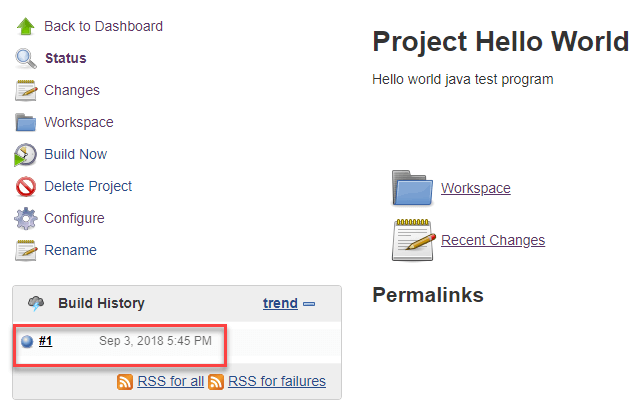
**Step 7)** When you have entered all the data,

1. Click **Apply**
2. **Save**the project.

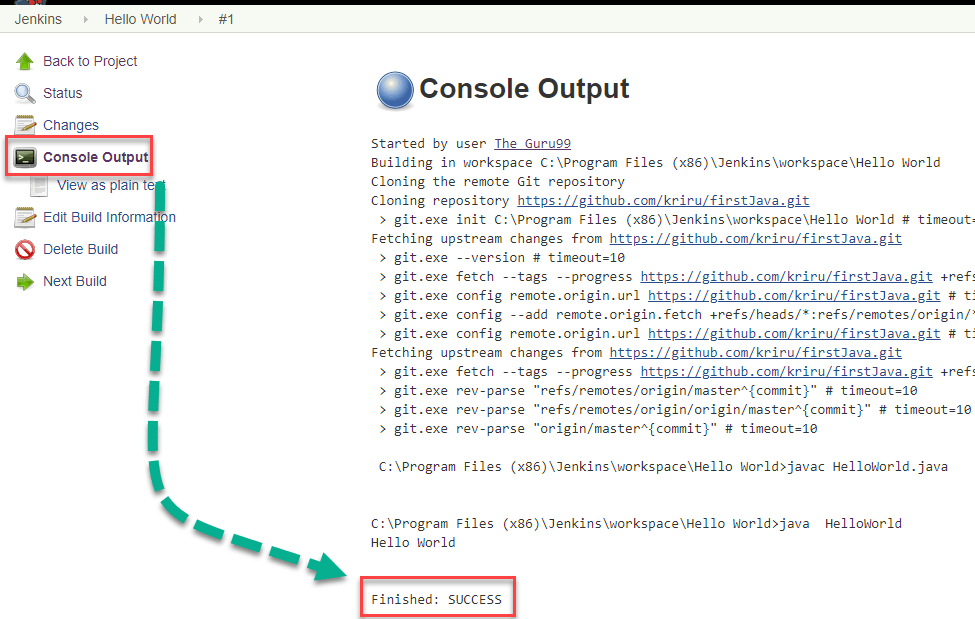
**Step 8)**Now, in the main screen, Click the **Build Now**button on the left-hand side to build the source code.

[](https://www.guru99.com/images/1/091318_0458_HowtoCreate9.png)

**Step 9)**After clicking on **Build now,**you can see the status of the build you run under **Build History**.

[](https://www.guru99.com/images/1/091318_0458_HowtoCreate10.png)

**Step 10)**Click on the **build number**andthenClick on **console output**to see the status of the build you run. It should show you a success message, provided you have followed the setup properly.

[](https://www.guru99.com/images/1/091318_0458_HowtoCreate11.png)