

Aim – Executing java sequence programming and integration of GitHub with Jenkins.

Theory –

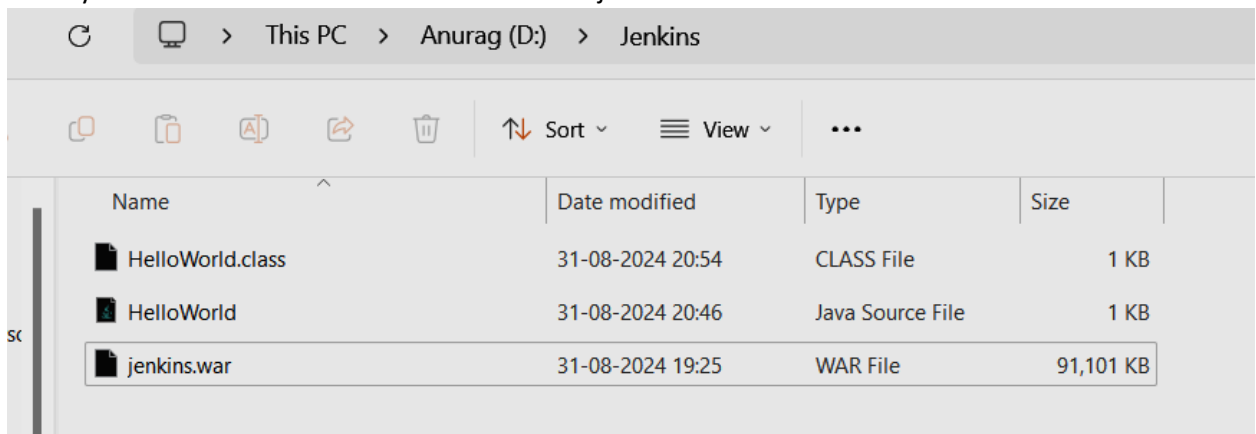
Jenkins comes with a pretty basic setup, so you will need to install the required plugins to enable respective third-party application support.

GitHub is a web-based repository of code which plays a major role in DevOps. It provides a common platform for multiple developers working on the same code/project to upload and retrieve updated code, thereby facilitating continuous integration.

Jenkins needs to have GitHub plugin installed to be able to pull code from the GitHub repository.

Steps –

1. Login to Jenkins.
2. Go to your Jenkins folder and create “HelloWorld.java” file.







3. In Jenkins Create a Job or Item, Enter item name as Hello World and select “Freestyle Project”.

New Item

Enter an item name

Select an item type

-  **Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
-  **Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
-  **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
-  **Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different

4. Write a description.

General

Enabled 






Description

This is a java Hello World Program.

Plain text [Preview](#)

5. Go to advanced tab of “General configure” and select “Use Custom Workspace” and copy the path of your java file directory and paste it here.

Advanced   Edited

- ☐ Quiet period 
- ☐ Retry Count 
- ☐ Block build when upstream project is building 
- ☐ Block build when downstream project is building 
- ☒ Use custom workspace 

Directory

D:\Jenkins


Display Name 

Save

Apply

6. Go to “Build Steps” in Configure and add a “Execute Windows Batch Command”.

Build Steps

Add build step 

 Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to “pending” on GitHub commit

7. In command type “javac HelloWorld.java” and “java HelloWorld” and click Save.

Build Steps

≡ Execute Windows batch command ?

Command

See [the list of available environment variables](#)

```
javac HelloWorld.java
java HelloWorld
```

Advanced ▾

8. A new page opens. Execute “Hello java” by clicking on “Build Now” link. If the job is successful, then it will show Blue ball or Green tick otherwise Red ball under “Build History”.

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Hello World

This is a java Hello World Program.

Permalinks

9. In Build History go to “Console Output” to get the output of your build.

Status

Changes

Console Output

Edit Build Information

Delete build #1

Timings

✓ Console Output

Download Copy View as plain text

```
Started by user Anurag Tiwari
Running as SYSTEM
Building in workspace D:\Jenkins
[Jenkins] $ cmd /c call C:\Users\anura\AppData\Local\Temp\jenkins3616656777399973236.bat

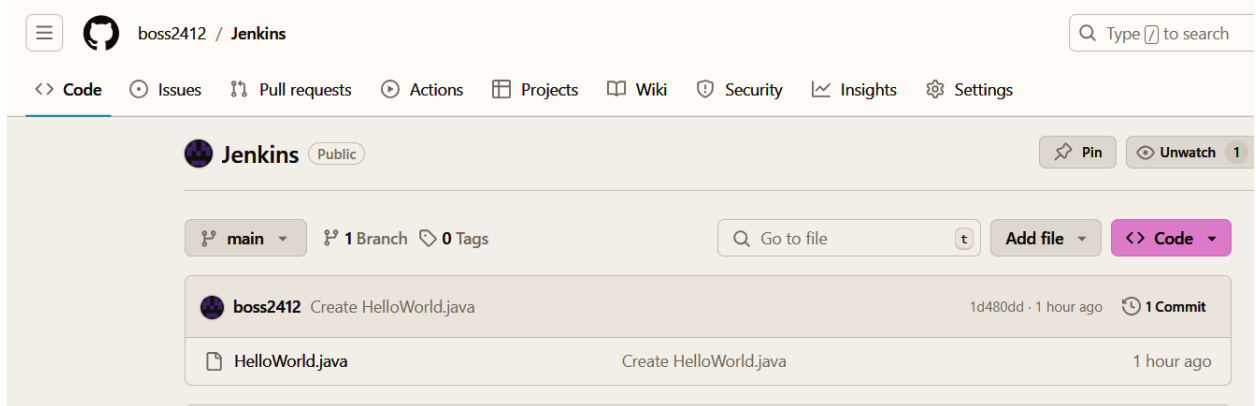
D:\Jenkins>javac HelloWorld.java

D:\Jenkins>java HelloWorld
Hello, World!

D:\Jenkins>exit 0
Finished: SUCCESS
```

Integrate Github with Jenkins Steps –

1. Login to Jenkins.
2. In Jenkins Create a Job or Item, Enter item name as Hello and select “Freestyle Project”.
3. Write a description.
4. Create a New Repository in Github and create “HelloWorld.java” file in it containing the code.



5. Go to “Source Code Management” and select “Git” option.



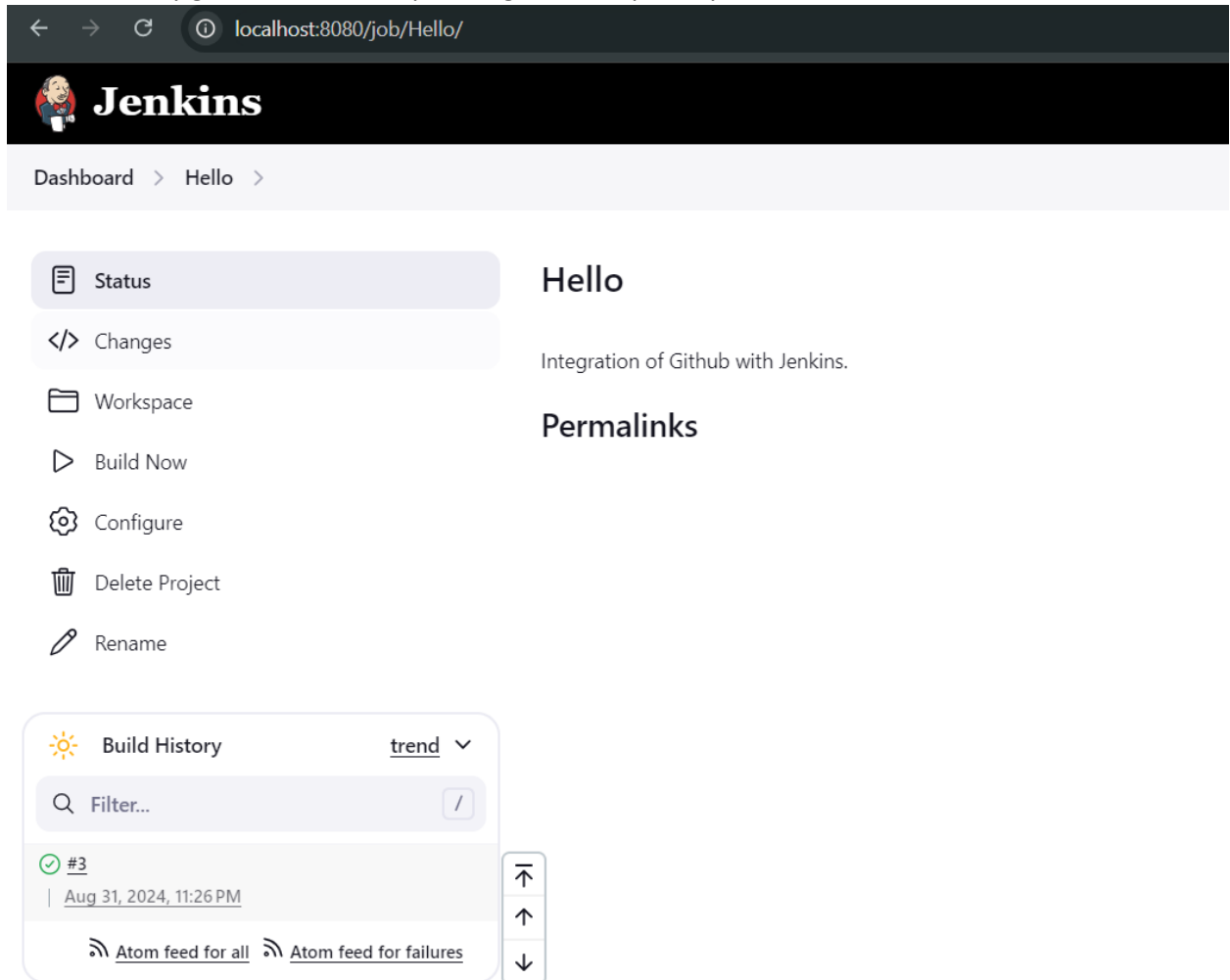
6. In Repository URL, paste the URL of the newly created repository and select the branch to build from your repository.

A screenshot of the Jenkins 'Repositories' configuration section. It shows a form with the following fields:

- Repository URL**: A text input field containing 'https://github.com/boss2412/Jenkins.git'.
- Credentials**: A dropdown menu showing '- none -'.
- Advanced**: A dropdown menu.
- Branches to build**: A section with a 'Branch Specifier (blank for 'any')' field containing '*/main'.

At the bottom, there are 'Save' and 'Apply' buttons.

7. Go to “Build Steps” in Configure and add a “Execute Windows Batch Command”.
8. In command type “javac HelloWorld.java” and “java HelloWorld” and click Save.
9. A new page opens. Execute “Hello java” by clicking on “Build Now” link. If the job is successful, then it will show Blue ball or Green tick otherwise Red ball under “Build History”.
10. In Build History go to “Console Output” to get the output of your build.



Conclusion –

Hence, successfully executed java sequence program and integrated GitHub with Jenkins.