Playwright Integration with Jenkins

How to Run a Local Playwright Project in Jenkins:

Step 1: Login to Jenkins

Open your Jenkins instance in a browser and log in using your credentials.

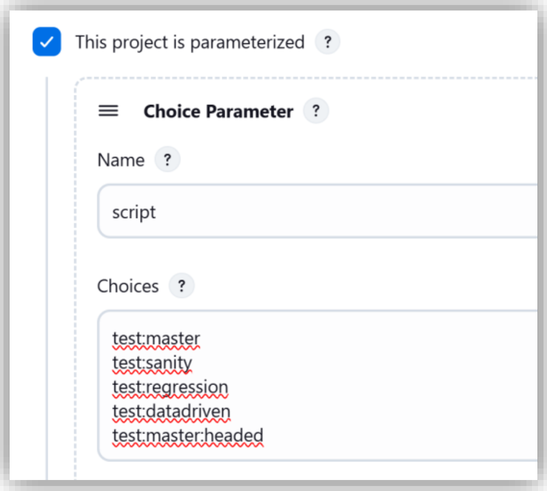
Step 2: Create a New Jenkins Job

- Click on **"New Item"**.
- Enter a name for your job.
- Select **"Freestyle project"**.
- Click **OK** to proceed.

Step 3: Configure the Job

Enable Parameters

- Check the box **"This project is parameterized"**.
- Click **Add Parameter → Choice Parameter**.
    - **Name:** script
    - **Choices:**
    - test:master
    - test:sanity
    - test:regression
    - test:master:headed



This project is parameterized

Choice Parameter

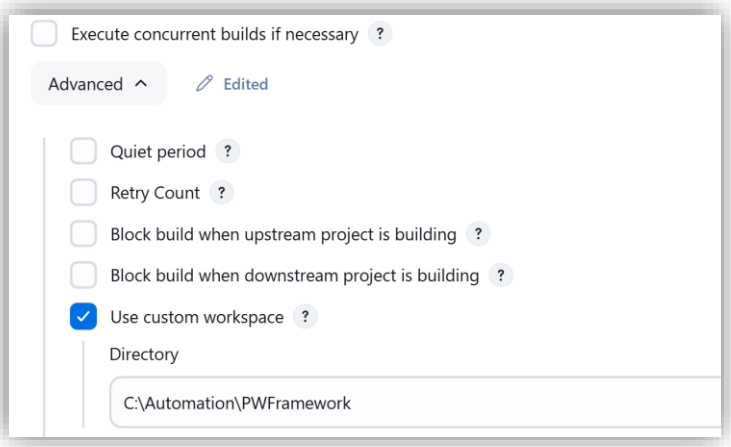
Name: script

Choices:

```
test:master
test:sanity
test:regression
test:datadriven
test:master:headed
```

Set Custom Workspace

- Check **“Use custom workspace”**.
- Enter the full path to your local Playwright project directory.



☐ Execute concurrent builds if necessary ?

Advanced ^ ✎ Edited

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

Directory

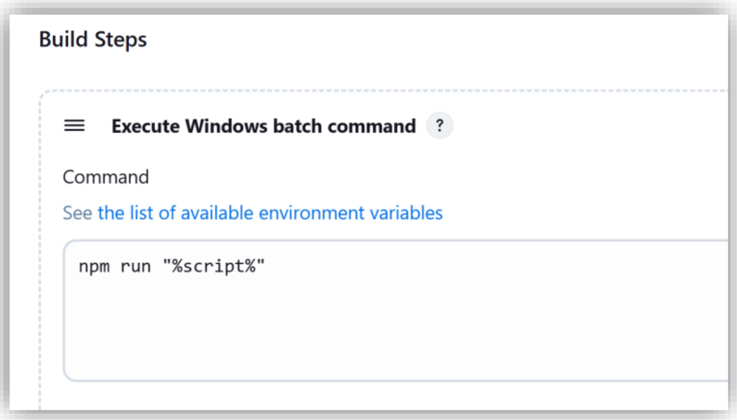
C:\Automation\PWFramework

Add Build Step

- Click on **“Add build step” → “Execute Windows batch command”** *(if on Windows)*.
    - **Command:**
    - npm run "%script%"

Note: If you're on **Mac/Linux**, select **“Execute shell”** and use:

npm run "$script"



Build Steps

Execute Windows batch command ?

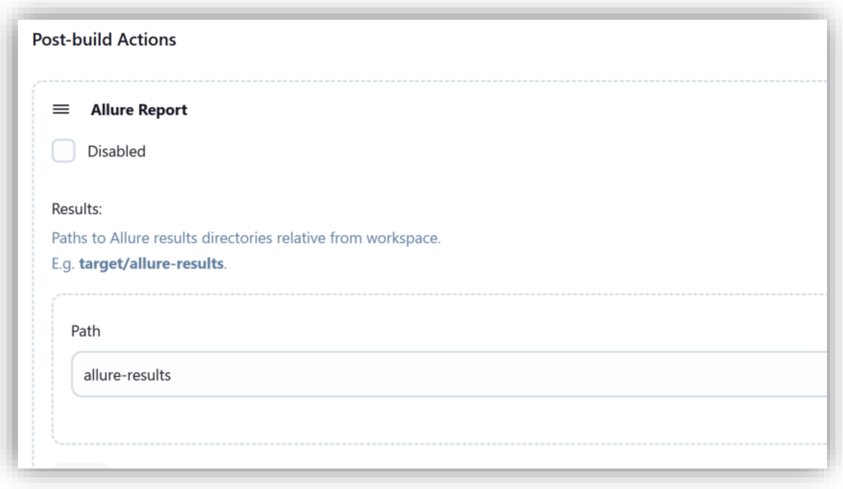
Command

See the list of available environment variables

```
npm run "%script%"
```

Step 4: Post-build Actions (Allure Reports)

- Scroll to **“Post-build Actions”**.
- Select **“Allure Report”**.
- In **“Results path”**, enter:
- allure-results


Post-build Actions

Allure Report

Disabled

Results:

Paths to Allure results directories relative from workspace.
E.g. **target/allure-results**.

Path

allure-results

Step 5: Save the Configuration

- Click **Apply** and then **Save** the job.

How to Run Github Playwright project in Jenkins

Step 1: Login to Jenkins

Open your Jenkins instance in a browser and log in using your credentials.

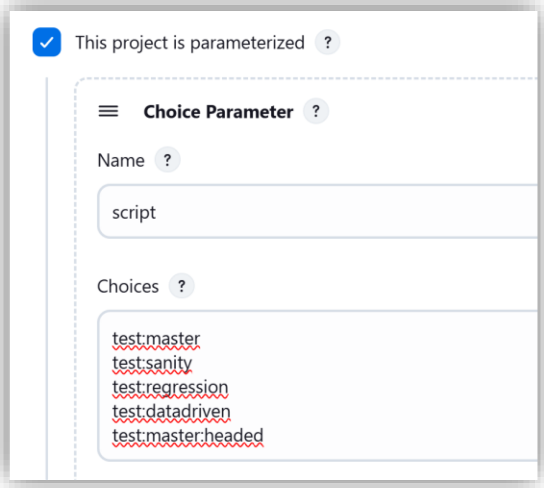
Step 2: Create a New Jenkins Job

- Click on **"New Item"**.
- Enter a name for your job.
- Select **"Freestyle project"**.
- Click **OK** to proceed.

Step 3: Configure the Job

Enable Parameters

- Check the box **"This project is parameterized"**.
- Click **Add Parameter → Choice Parameter**.
    - **Name:** script
    - **Choices:**
    - test:master
    - test:sanity
    - test:regression
    - test:master:headed



This project is parameterized [✓]

Choice Parameter

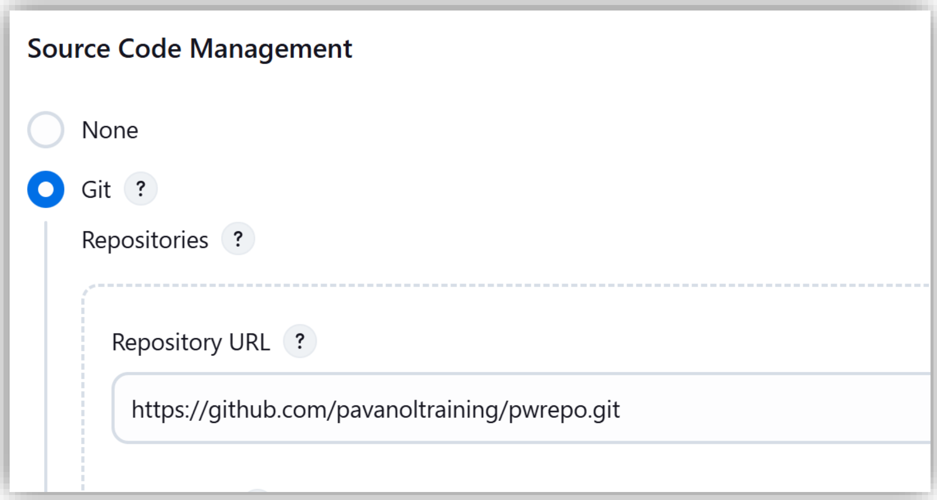
Name: script

Choices:

```
test:master
test:sanity
test:regression
test:datadriven
test:master:headed
```

Set GitHub Repository URL

- Go to '**Source Code Management'** section.
- Check **"Git"**.
- Provide GitHub repository URL.



Source Code Management

None

[✓ Git]

Repositories

Repository URL: https://github.com/pavanoltraining/pwrepo.git

Add Build Steps

- Click on **"Add build step" → "Execute Windows batch command"** *(if on Windows)*.
    - **Command:**
    - setup_env.bat
- **Note:** We need to create **seup_env.bat** file inside the project prior to run the tests. This file contains commands to setup environment to run playwright tests.
    - npm install
    - npm install -D allure-playwright
    - npm install -g allure-commandline --force

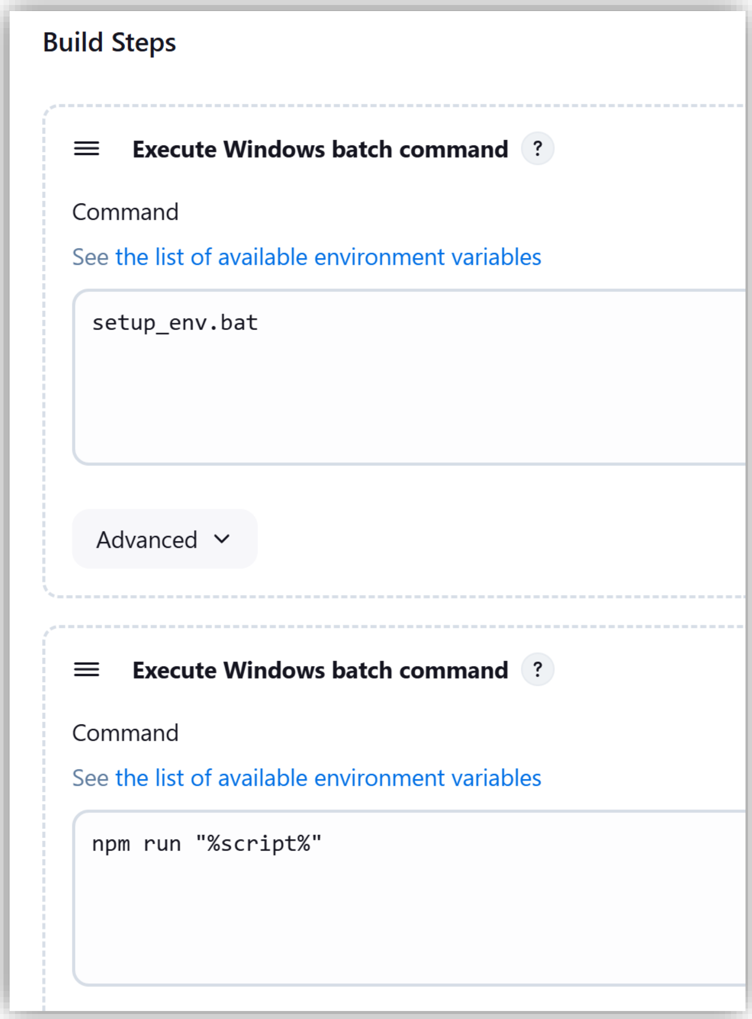
- npx playwright install

If you are using Mac then setup_env.sh file Should be created with the same commands.

- One more time Click on **“Add build step” → “Execute Windows batch command”** *(if on Windows)*.
    - **Command:**
    - npm run "%script%"

Note: If you're on **Mac/Linux**, select **“Execute shell”** and use:

```
npm run "$script"
```



Build Steps

Execute Windows batch command
?
Command
See the list of available environment variables

```
setup_env.bat
```

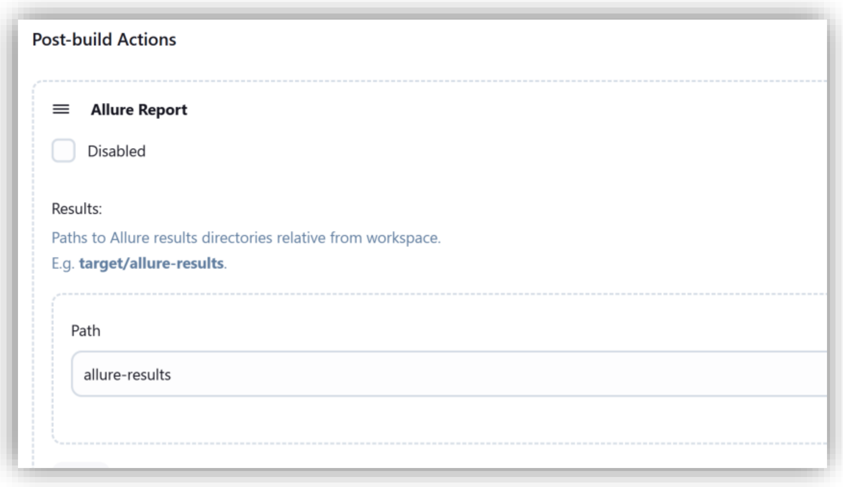
Advanced
Execute Windows batch command
?
Command
See the list of available environment variables

```
npm run "%script%"
```

Step 4: Post-build Actions (Allure Reports)

- Scroll to **“Post-build Actions”**.
- Select **“Allure Report”**.
- In **“Results path”**, enter:

- allure-results



Post-build Actions

Allure Report

Disabled

Results:

Paths to Allure results directories relative from workspace.

E.g. **target/allure-results**.

Path

allure-results

Step 5: Save the Configuration

- Click **Apply** and then **Save** the job.

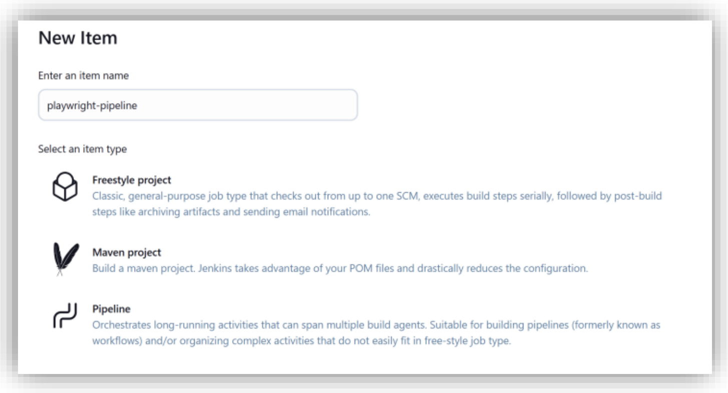
Run Github Playwright project in Jenkins with Pipeline

Step 1: Login to Jenkins

Open your Jenkins instance in a browser and log in using your credentials.

Step 2: Create a New Jenkins Job

- Click on **“New Item”**.
- Enter a name for your job.
- Select **“Pipeline project”**. (If you could not see this , then you need to install Pipeline plugin from manage Jenkins→plugins).
- Click **OK** to proceed.



New Item

Enter an item name

playwright-pipeline

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.

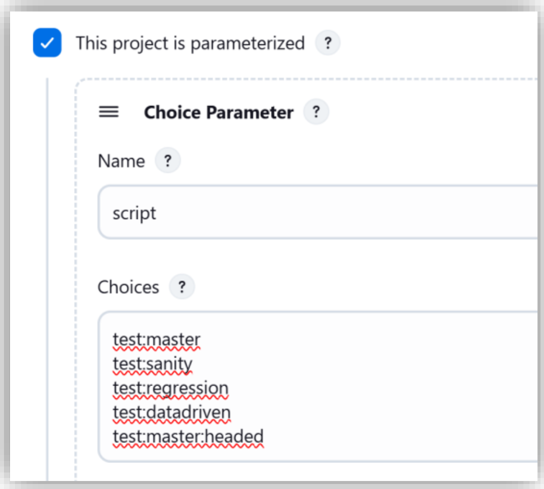
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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.

Step 3: Configure the Job

Enable Parameters

- Check the box **“This project is parameterized”**.
- Click **Add Parameter → Choice Parameter**.
    - **Name:** script
    - **Choices:**
    - test:master
    - test:sanity
    - test:regression
    - test:master:headed



This project is parameterized

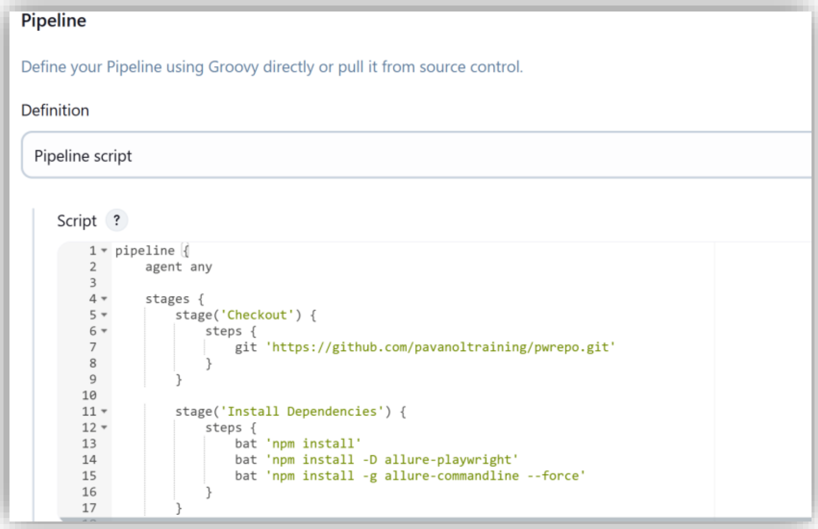
Choice Parameter

Name: script

Choices:

```
test:master
test:sanity
test:regression
test:datadriven
test:master:headed
```

Goto Pipline section and add Pipeline script.



Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script

Script

```
pipeline {
    agent any

    stages {
        stage('Checkout') {
            steps {
                git 'https://github.com/pavanoltraining/pwrepo.git'
            }
        }

        stage('Install Dependencies') {
            steps {
                bat 'npm install'
                bat 'npm install -D allure-playwright'
                bat 'npm install -g allure-commandline --force'
            }
        }
```

<u>Pipeline Script:</u>

```
pipeline {
    agent any

    stages {
        stage('Checkout') {
            steps {
                git 'https://github.com/pavanoltraining/pwrepo.git'
            }
        }

        stage('Install Dependencies') {
            steps {
                bat 'npm install'
                bat 'npm install -D allure-playwright'
                bat 'npm install -g allure-commandline --force'
            }
        }

        stage('Run Tests') {
            steps {
                bat 'npx playwright install'
                //bat 'npx playwright test --grep=@master'
                bat 'npm run "%script%"'


            }
        }

        stage('Generate Allure Report') {
            steps {
                bat 'allure generate ./allure-results --clean -o ./allure-report'
            }
        }
    }

    post {
        always {
            // Archive test results (if any)
            archiveArtifacts artifacts: 'test-results/**/*', allowEmptyArchive:
true

            // Publish Allure results (if plugin is installed)
            allure includeProperties: false,
                   jdk: '',
                   results: [[path: 'allure-results']]
        }
    }
}
```

Step 4: Save the Configuration

- Click **Apply** and then **Save** the job.