# ****QA Automation Groovy Script for Jenkins CI/CD Pipeline****

**(Selenium + TestNG + Maven + Parallel Execution + Reporting)**

This Jenkins pipeline script automates:  
✅ **Git Checkout** → Pull latest code  
✅ **Dependency Installation** → Maven + WebDriver  
✅ **Parallel Test Execution** → Cross-browser testing  
✅ **Test Reporting** → JUnit + ExtentReports  
✅ **Artifact Archiving** → Screenshots & Logs  
✅ **Slack Notifications** → Build status alerts

## ****1. Jenkinsfile (Groovy Pipeline Script)****

groovy

Copy

Download

pipeline {

agent any

environment {

*// Configurable parameters*

REPO\_URL = 'https://github.com/your-repo/qa-automation.git'

BRANCH = 'main'

TESTNG\_SUITE = 'testng.xml'

BROWSERS = 'chrome,firefox' *// Comma-separated for parallel runs*

}

stages {

*// Stage 1: Checkout Code*

stage('Checkout') {

steps {

script {

echo "📦 Checking out code from ${REPO\_URL}..."

git branch: "${BRANCH}", url: "${REPO\_URL}"

}

}

}

*// Stage 2: Install Dependencies*

stage('Setup') {

steps {

script {

echo "🔧 Installing dependencies..."

sh 'mvn clean install -DskipTests'

*// Auto-download browsers using WebDriverManager*

sh '''

java -cp target/test-classes:target/classes \

com.utilities.BrowserSetup

'''

}

}

}

*// Stage 3: Parallel Test Execution*

stage('Test') {

parallel {

stage('Chrome Tests') {

when { expression { env.BROWSERS.contains('chrome') } }

steps {

script {

echo "🚀 Running Chrome tests..."

sh "mvn test -DsuiteXmlFile=${TESTNG\_SUITE} -Dbrowser=chrome"

}

}

}

stage('Firefox Tests') {

when { expression { env.BROWSERS.contains('firefox') } }

steps {

script {

echo "🚀 Running Firefox tests..."

sh "mvn test -DsuiteXmlFile=${TESTNG\_SUITE} -Dbrowser=firefox"

}

}

}

}

}

*// Stage 4: Reporting*

stage('Reports') {

steps {

script {

echo "📊 Generating reports..."

*// JUnit Report*

junit 'target/surefire-reports/\*.xml'

*// ExtentReports HTML*

publishHTML target: [

allowMissing: true,

alwaysLinkToLastBuild: true,

keepAll: true,

reportDir: 'test-output',

reportFiles: 'ExtentReport.html',

reportName: 'QA Test Report'

]

*// Archive screenshots on failure*

archiveArtifacts artifacts: 'test-output/screenshots/\*.png', allowEmptyArchive: true

}

}

}

}

post {

always {

script {

echo "🧹 Cleaning up workspace..."

cleanWs()

}

}

success {

slackSend (

color: '#36a64f',

message: "✅ QA Automation PASSED - ${env.JOB\_NAME} (#${env.BUILD\_NUMBER})",

channel: '#qa-alerts'

)

}

failure {

slackSend (

color: '#ff0000',

message: "❌ QA Automation FAILED - ${env.JOB\_NAME} (#${env.BUILD\_NUMBER})",

channel: '#qa-alerts'

)

}

}}

## ****2. Supporting Java Class (BrowserSetup.java)****

(Optional: For auto-downloading browsers in Jenkins agents)

java

Copy

Download

package com.utilities;

import io.github.bonigarcia.wdm.WebDriverManager;

public class BrowserSetup {

public static void main(String[] args) {

WebDriverManager.chromedriver().setup();

WebDriverManager.firefoxdriver().setup();

WebDriverManager.edgedriver().setup();

}}

## ****3. Pipeline Features Explained****

### ****A. Parallel Cross-Browser Testing****

Runs **Chrome + Firefox** tests simultaneously

Configurable via BROWSERS environment variable

### ****B. Smart Reporting****

**JUnit Reports** → Integrated with Jenkins

**ExtentReports** → Interactive HTML dashboard

**Artifact Archiving** → Screenshots on failure

### ****C. Slack Notifications****

Sends alerts to #qa-alerts channel

Different colors for **pass/fail**

### ****D. Self-Healing Setup****

Auto-downloads browser drivers using WebDriverManager

## ****4. Jenkins Setup Requirements****

**Install Plugins**:

[Pipeline](https://plugins.jenkins.io/workflow-aggregator/" \t "https://chat.deepseek.com/a/chat/s/_blank)

[JUnit](https://plugins.jenkins.io/junit/" \t "https://chat.deepseek.com/a/chat/s/_blank)

[HTML Publisher](https://plugins.jenkins.io/htmlpublisher/" \t "https://chat.deepseek.com/a/chat/s/_blank)

[Slack Notification](https://plugins.jenkins.io/slack/" \t "https://chat.deepseek.com/a/chat/s/_blank)

**Configure System**:

Add **Slack credentials** in Manage Jenkins > Configure System

Set **JDK + Maven** paths in global tools

**Create Pipeline Job**:

Select Pipeline script from SCM → Point to your Git repo

## ****5. How to Run****

**Manual Trigger**:

bash

Copy

Download

*# Override parameters if needed*BRANCH="dev" BROWSERS="chrome,edge" TESTNG\_SUITE="smoke-tests.xml"

**Scheduled Run (Cron)**:

groovy

Copy

Download

triggers {

cron('H 22 \* \* \*') *// Nightly at 10 PM*}

**Git Webhook**: Auto-trigger on git push

## ****6. Expected Output****

| **Stage** | **Result** |
| --- | --- |
| Checkout | ✅ Git repo cloned |
| Setup | ✅ Maven + WebDriver installed |
| Test (Parallel) | 🔄 Chrome + Firefox tests running |
| Reports | 📊 JUnit + ExtentReports generated |
| Slack Alert | 💬 Notification sent to #qa-alerts |

## ****7. Pro Tips****

🔹 **Dockerize Tests**:

groovy

Copy

Download

agent {

docker {

image 'maven:3.8.4-openjdk-11'

args '-v /tmp/.X11-unix:/tmp/.X11-unix' *// For GUI browsers*

}}

🔹 **Dynamic Parameters**:

groovy

Copy

Download

parameters {

choice(

name: 'BROWSERS',

choices: ['chrome', 'firefox', 'chrome,firefox'],

description: 'Select browsers'

)}

🔹 **Retry Failed Tests**:

groovy

Copy

Download

sh 'mvn test -Dsurefire.rerunFailingTestsCount=2'

### ****Final Notes****

This pipeline:  
✔ **Reduces feedback time** with parallel execution  
✔ **Improves visibility** with rich reports  
✔ **Requires zero manual intervention**  
✔ **Scales easily** for new browsers/tests

Would you like me to add **API testing integration** or **Dockerized Selenium Grid** support? 🚀