

Github intergrate + Maven build

Go to repo setting - webhooks:

<https://github.com/<account>/<repo>/settings/hooks>

choose setting below:

payload url: your url

Content type: application/json

if your url is https, choose enable ssl

Add webhook

Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub apps

Email notifications

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information [documentation](#).

Payload URL *

Content type

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

☒ Enable SSL verification ☐ Disable (not recommended)

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me everything.

☐ Let me select individual events.

☒ Active

We will deliver event details when this hook is triggered.

Add webhook

Create new pipeline and check "GitHub hook trigger for GITScm polling" it will trigger git and automate build and trigger jenkins .

Configure

General

Enabled 

-  General
-  Advanced Project Options
-  Pipeline

Description

[Plain text] [Preview](#)

- ☐ Discard old builds [?](#)
- ☐ Do not allow concurrent builds
- ☐ Do not allow the pipeline to resume if the controller restarts
- ☐ GitHub project
- ☐ Pipeline speed/durability override [?](#)
- ☐ Preserve stashes from completed builds [?](#)
- ☐ This project is parameterized [?](#)
- ☐ Throttle builds [?](#)

Build Triggers

- ☐ Build after other projects are built [?](#)
- ☐ Build periodically [?](#)
- ☒ GitHub hook trigger for GITScm polling [?](#)

Scroll down and type your repo and branch. apply and build

Definition

Pipeline script from SCM

SCM [?](#)

Git

Repositories [?](#)Repository URL [?](#)<https://github.com/hiep98/devsecops>Credentials [?](#)

- none -

+ Add

Advanced...

Add Repository

Branches to build [?](#)Branch Specifier (blank for 'any') [?](#)

*/main

Add Branch

Save

Apply

When you update code in git, git will call jenkins whenever a push event has been done.

We have file source code below:

```
pipeline {
  agent any
  stages {
    stage('Build Artifact') {
      steps {
        sh "mvn clean package -DskipTests=true"
        archive 'target/*.jar' //so that they can be downloaded later
      }
    }
  }
}
```

The Jenkins pipeline above is designed to create an artifact, or executable, in this case a *.jar file. First, it will run a command to execute the `mvn clean package -DskipTests=true` command, which cleans and packages the code. Then, it will store the built *.jar files in the `target` directory and archive them using the `archive 'target/*.jar'` command. This makes the files easier to share with other users and keeps them safe from accidental changes.