



JENKINS SHARED LIBRARIES

Mithun Technologies, +91 99809 23226, devopstrainingblr@gmail.com

A large, faint watermark of the Mithun Technologies logo is centered on the page. The logo consists of a stylized blue 'T' shape with a bird-like figure perched on top. The text 'Mithun Technologies' is written in a light blue font across the middle of the 'T' and below it.

Mithun
Technologies

Mithun
Technologies

MITHUN SOFTWARE SOLUTIONS
Bangalore

Mithun Technologies +91-9980923226	Jenkins Shared Libraries info@mithuntechnologies.com	Author Web site	Mithun Technologies http://mithuntechnologies.com
-----------------------------------------------------	-----------------------------------------------------------------------	----------------------------------	--------------------------------------------------------------------

Jenkins Shared Libraries

Jenkins Shared library is the concept of having a common pipeline code in the version control system that can be used by any number of pipelines just by referencing it. In fact, multiple teams can use the same library for their pipelines.

Step 1

Directory structure

The directory structure of a Shared Library repository is as follows:

(root)

```

+- src                # Groovy source files

| +- com
|
| +- mss
|
| +- deploy.groovy   # for com.mss.deploy class

+- vars

| +- build.groovy     # for global 'build' variable
| +- build.txt        # help for 'build' variable

+- resources          # resource files (external libraries only)

| +- com
|
| +- mss
|
| +- help.json        # static helper data for org.foo.Bar

```

src: The src directory should look like standard Java source directory structure. This directory is added to the classpath when executing Pipelines.

vars: This directory holds all the global shared library code that can be called from a pipeline. It has all the library files with a .groovy extension.

resources: A resources directory allows the libraryResource step to be used from an external library to load associated non-Groovy files. Currently this feature is not supported for internal libraries.

Step 2:

Add GitHub Shared Library Repository to Jenkins

Go to Manage Jenkins → Configure System, then

Find the **Global Pipeline Libraries** section and add your shared libraries repo details and configurations as shown below.

Global Pipeline Libraries

Sharable libraries available to any Pipeline jobs running on this system. These libraries will be trusted, meaning they run without “sandbox” restrictions and may use @Grab.

Library

Name

mssharedlibs — Give Any Name

Default version

main — Branch Name

Currently maps to revision: ea0ab539778be200dbea836cc3ddb3b9044b9

☐ Load implicitly

☒ Allow default version to be overridden

☒ Include @Library changes in job recent changes

Retrieval method

☒ Modern SCM

Source Code Management

☒ Git

Project Repository

https://github.com/MithunTechnologiesDevOps/JenkinsSharedLibraries.git

Credentials

devopstrainingblr/***** (GitHub Credentials)

Add

Behaviors

Within Repository

Discover branches

Additional

Add

☐ GitHub

☐ Legacy SCM

Mithun Technologies +91-9980923226	Jenkins Shared Libraries info@mithuntechnologies.com	Author	Mithun Technologies
		Web site	http://mithuntechnologies.com

Step 3:

To access the shared libraries, in the Jenkinsfile (declarative pipeline) needs to add the @Library annotation, specifying the library's name as follows.

```
@Library('msssharedlibs') _
/* Accessing multiple libraries with one statement */
@Library(['msssharedlibs', 'msssharedlibs1']) _

pipeline {
    agent any
    stages{

        stage('CheckoutCode')
        {
            steps
            {
                git branch: 'development', credentialsId: '6773dc34-a3fd-44ec-blee-3d2f0aa9f346', url: 'https://github.com/MithunTechnologiesDevOps/maven-web-application.git'
            }
        }

        stage("Build"){
            steps{
                stages("Build")
            }
        }

        stage("Execute SonarQube Report"){
            steps{
                stages("SonarQube Report")
            }
        }

        stage("Upload Artifacts Into Nexus"){
            steps{
                stages("Upload Into Nexus")
            }
        }

    } // Stages Close
} // Pipeline Close
```

To get the build parameters...

```
parameters {
    string(name: 'GoalName', defaultValue: 'Package', description: 'Pass the Maven Goals')
    choice(name: 'BranchName', choices: ['master', 'development', 'stage'], description:
'Pick the Branch Name.')
```

Mithun Technologies +91-9980923226	Jenkins Shared Libraries info@mithuntechnologies.com	Author	Mithun Technologies
		Web site	http://mithuntechnologies.com

```
}
```

```
stages{
```

```
    stage('CheckouCode')
```

```
{
```

```
    steps
```

```
{
```

```
        git branch: "${params.BranchName}", credentialsId: '6773dc34-a3fd-44ec-b1ee-3d2f0aa9f346', url: 'https://github.com/MithunTechnologiesDevOps/maven-web-application.git'
```

```
    }
```

```
}
```

Note: Here you must use the "" quotes.

References

<https://www.jenkins.io/doc/book/pipeline/shared-libraries/>

<https://www.jenkins.io/doc/book/pipeline/syntax/>