

| | | | |
|---------------------------------------|--|--------------------|--|
| Landmark Technologies +14372152483 | Jenkins Shared Libraries info@mylandmarktech.com | Author Web site | Landmark Technologies http://mylandmarktech.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

LandmarkSS-sharedlibs — Anyname

Default version

master — Branch Name

Cannot validate default version until after saving and reconfiguring.

☐ Load implicitly

☒ Allow default version to be overridden

☒ Include @Library changes in job recent changes

☐ Cache fetched versions on master for quick retrieval

Retrieval method

Modern SCM

Loads a library from an SCM plugin using newer interfaces optimized for this purpose. The recommended option when available.

Source Code Management

Git

Project Repository

https://github.com/LandmakTechnology/JenkinsSharedLibraries

Credentials

legah2045/***** (GitHubCredentials)

Add

Behaviours

Discover branches

Add

Library Path (optional)

Delete

| | | | |
|---------------------------------------|--|--------------------|--|
| Landmark Technologies +14372152483 | Jenkins Shared Libraries info@mylandmarktech.com | Author Web site | Landmark Technologies http://mylandmarktech.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('LandmarkSS-sharedlibs') _

pipeline {
    agent any
    stages{

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

| | | | |
|--|--|--|--|
| Landmark Technologies +14372152483 | Jenkins Shared Libraries info@mylandmarktech.com | Author Web site | Landmark Technologies http://mylandmarktech.com |
| <pre> stages{ stage('CheckouCode') { steps { git branch: "\${params.BranchName}", credentialsId: '6773dc34-a3fd-44ec-b1ee-3d2f0aa9f346', url: ' https://github.com/LandmakTechnology/maven-web-application.git' } } } </pre> <p>Note: Here you must use the "" quotes.</p> | | | |
| © All rights Reserved by Landmark Technologies | | mylanmarktech@gmail.com | |