

CircleCI configuration

Circle CI uses docker underneath.

CircleCI is a continuous integration and continuous delivery platform that can be used to implement DevOps practices.

Go to circleCI and click on to make new project and select maven.

For maven...

It pulls its own idk from docker image. Check this in config.yml.

Copy this config.yml and put it under .circleci folder under working directory.

For docker image of java 8 update this to 11

Image: circleci/openjdk:11-browsers-legacy

We can also check MAVEN_OPTS -> bumping up the memory of maven.

It also sets up restore cache for dependencies (internally in circleci env) and saves it in ~/.m2. It also downloads the dependencies so that we can work offline.

Mvn dependency:go-offline

Lastly it runs the integration test.

Now click on build under the project in circleci.

Actually you have to link your project to circleci using GitHub and then follow the steps...

The build does the same steps as mentioned in the config.yml file.

Build will run inside the docker container.

Maven version is actually coming from docker image.

```
# Java Maven CircleCI 2.0 configuration file
```

```
#
```

```
# Check https://circleci.com/docs/2.0/language-java/ for more details
```

```
#
```

```
version: 2
```

```
jobs:
```

```
  build:
```

```
    docker:
```

```
      # specify the version you desire here
```

```
      - image: circleci/openjdk:11-browsers-legacy
```

```
      # Specify service dependencies here if necessary
```

```
      # CircleCI maintains a library of pre-built images
```

```
      # documented at https://circleci.com/docs/2.0/circleci-images/
```

```
      # - image: circleci/postgres:9.4
```

```
working_directory: ~/repo
```

environment:

```
# Customize the JVM maximum heap limit
MAVEN_OPTS: -Xmx3200m
```

steps:

- checkout

```
# Download and cache dependencies
```

- restore_cache:

 - keys:

 - v1-dependencies-{{ checksum "pom.xml" }}

 - # fallback to using the latest cache if no exact match is found*

 - v1-dependencies-

- run: mvn dependency:go-offline

- save_cache:

 - paths:

 - ~/.m2

 - key: v1-dependencies-{{ checksum "pom.xml" }}

- run: git config user.email "john@springframework.guru"

- run: git config user.name "Via CircleCI Build"

```
# run tests!
```

- run: mvn --batch-mode release:prepare release:perform -s .circleci/settings.xml

You have to put settings.xml inside the .circleci folder

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://
maven.apache.org/xsd/settings-1.0.0.xsd">
```

```
<servers>
```

```
<server>
```

```
<id>packagecloud.release</id>
```

```
<password>${env.PCLOUD}</password>
```

```
</server>
```

```
<server>
```

```
<id>packagecloud.snapshot</id>
```

```
<password>${env.PCLOUD}</password>
```

```
</server>
```

```
</servers>
```

```
    <id>github</id>
    <username>springframeworkguru</username>
    <password>${env.GIT_PWD}</password>
  </server>
</servers>
</settings>
```