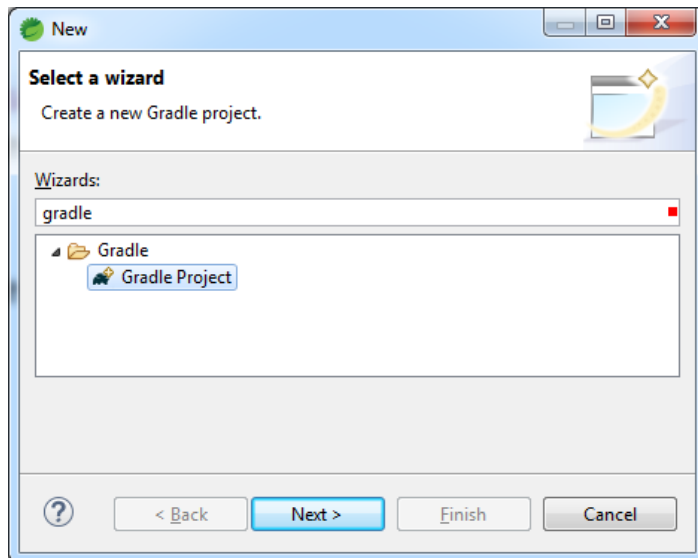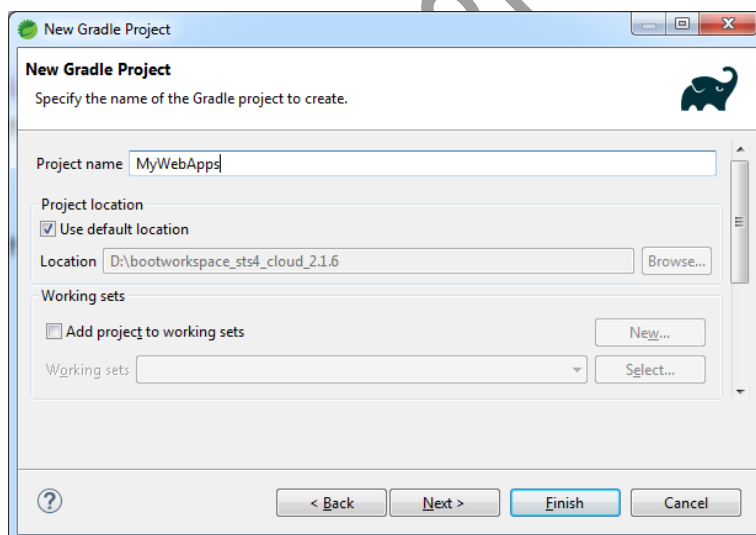# Gradle Servlets/JSP Web Application

## 1. CREATE GRADLE PROJECT

\>File>New>Other...



Enter the name of the project: MyWebApp



\> Next > Finish

## 2. CONFIGURING GRADLE:-

You need to add the configuration for your application to become "WEB Application". And can be run directly on Eclipse + Tomcat Plugin.

```
apply plugin: 'java'
apply plugin: 'war'
apply plugin: 'com.bmuschko.tomcat'
apply plugin: 'eclipse-wtp'

repositories {
   jcenter()
}

sourceCompatibility = 1.8
targetCompatibility = 1.8

dependencies {
   compile 'com.google.guava:guava:23.0'
   testCompile 'junit:junit:4.12'
   providedCompile "javax.servlet:javax.servlet-api:3.1.0"

}

dependencies {
   def tomcatVersion = '8.0.53'

   tomcat "org.apache.tomcat.embed:tomcat-embed-core:${tomcatVersion}",
   "org.apache.tomcat.embed:tomcat-embed-logging-
juli:${tomcatVersion}",
      "org.apache.tomcat.embed:tomcat-embed-jasper:${tomcatVersion}"
}

buildscript {

   repositories {
      jcenter()
   }
```
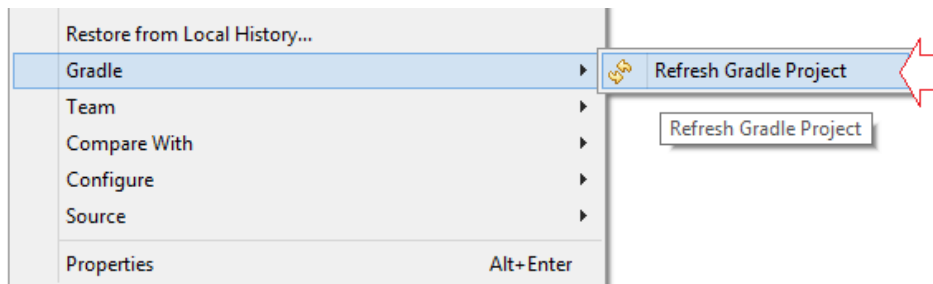
```
dependencies {
    classpath 'com.bmuschko:gradle-tomcat-plugin:2.5'


    }
}
```

Note that each time there is a change in build.gradle you need to update the project, using the tool of Gradle.

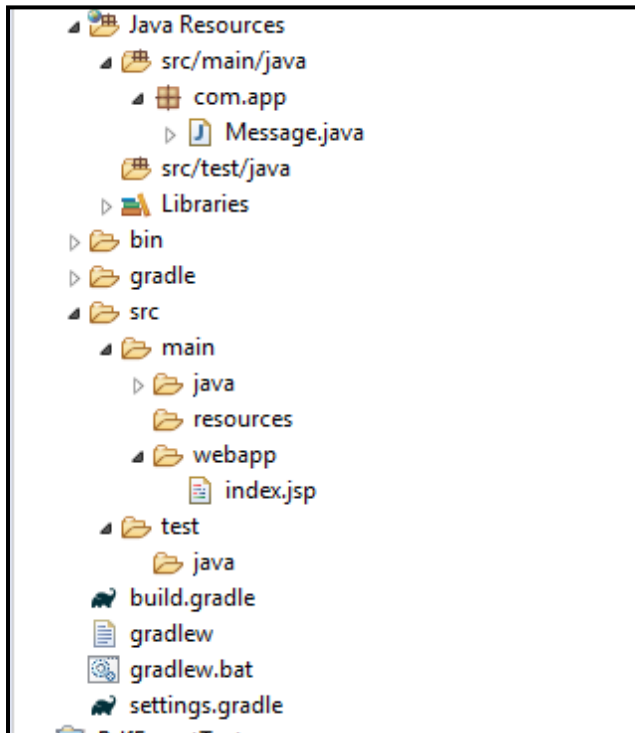

## 3. ADD FOLDER TO APPLICATION:-

In "src/main" folder, you need to create 2 sub folders
are "resources" and "webapp".
src/main/java: This folder has java sources.

src/main/resources: This folder can hold property files and other resources

src/main/webapp: This folder holds jsp and other web application content.

## 4. Write Code in Project



**Servlet class:**

```java
package com.app;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/home")
public class Message extends HttpServlet{

    private static final long serialVersionUID = -4633811587840173475L;

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
                    throws ServletException, IOException {
        PrintWriter out=resp.getWriter();
```
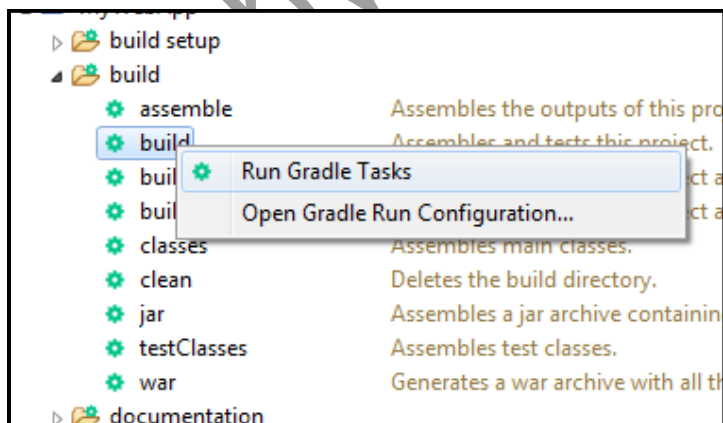
```java
            out.println("Hello App");
    }
}
```

**index.jsp file**

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Welcome to All</h1>
</body>
</html>
```
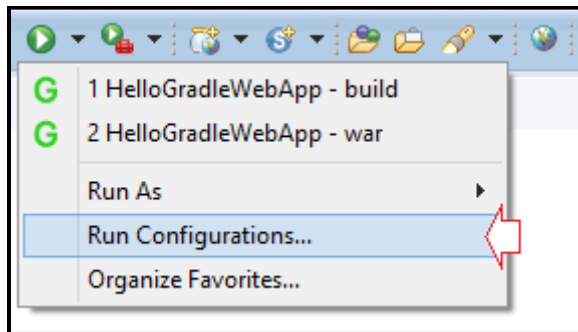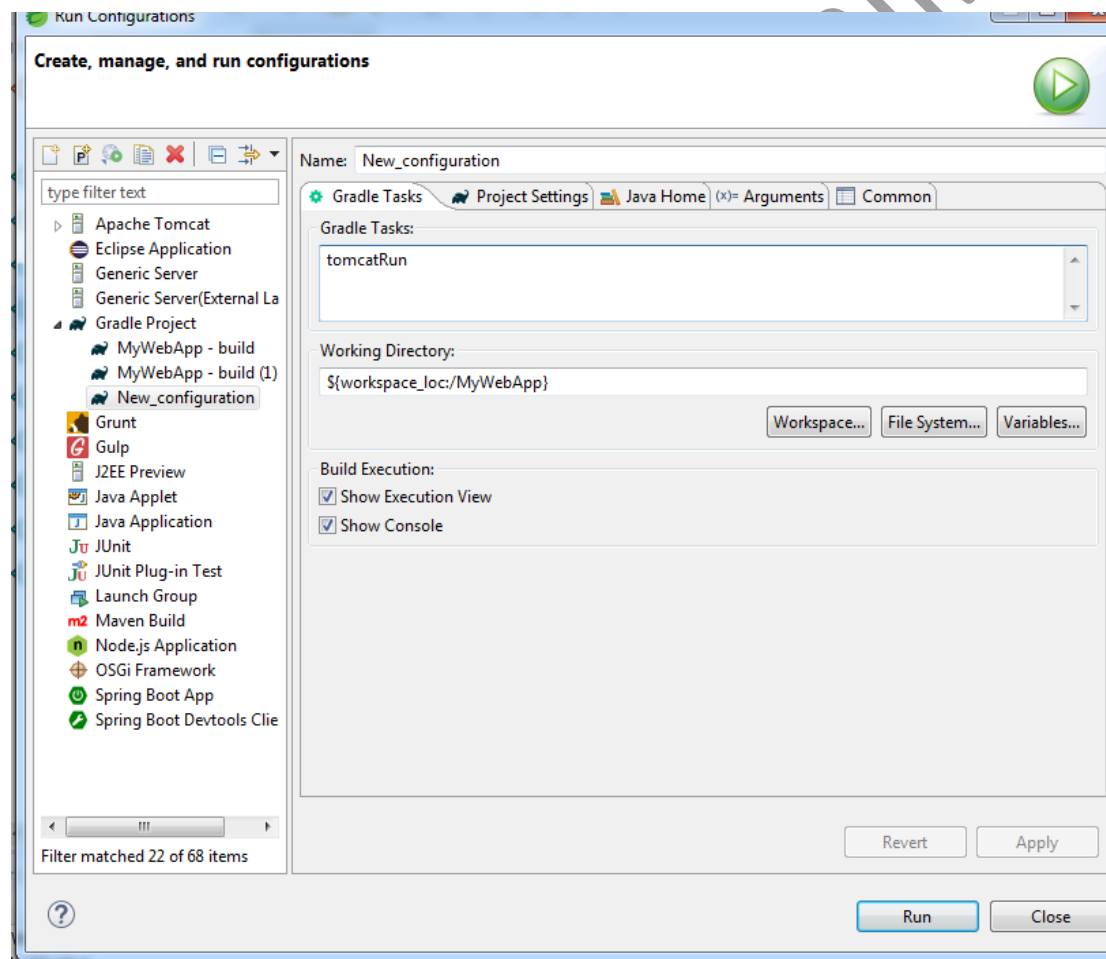
# 5. EXECUTE GRADLE BUILD TASK

> Open "Gradle Task" view > Expand Project > choose build > Run Gradle Tasks

# 6. CONFIGURE TO RUN APPLICATION
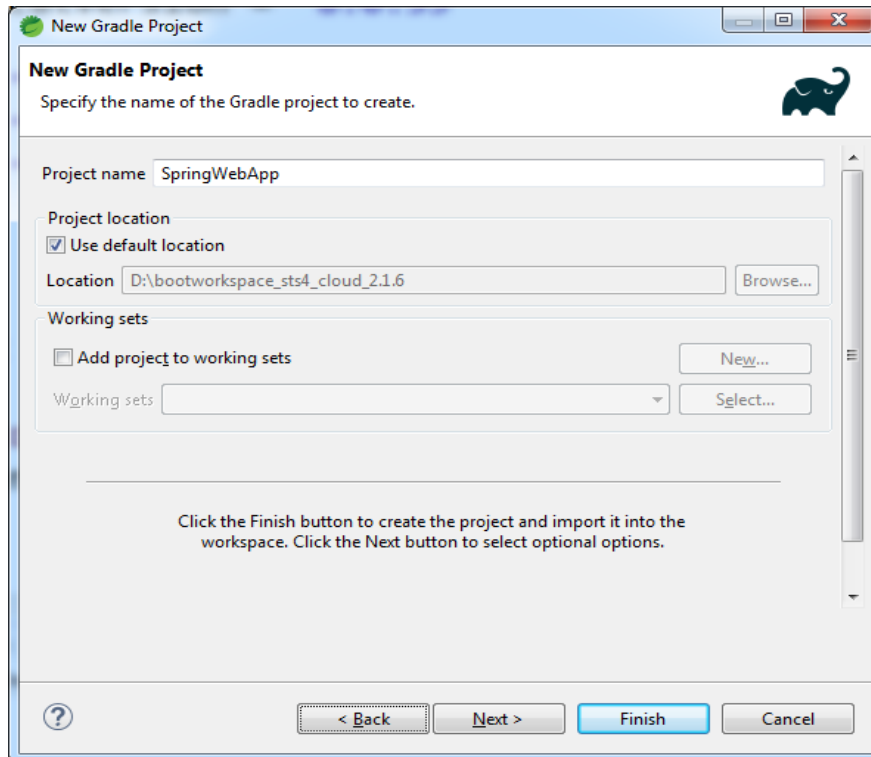


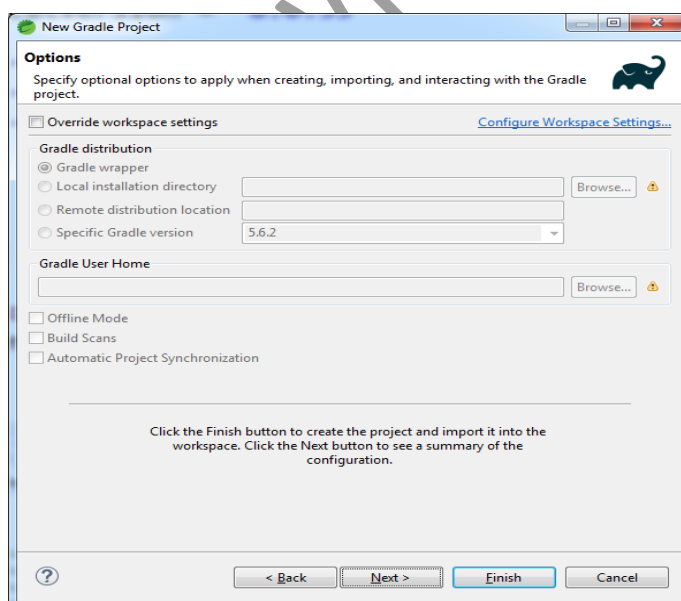Enter: tomcatRun  > Apply > Run



Goto browser and enter URL:

http://localhost:8080/MyWebApp/home
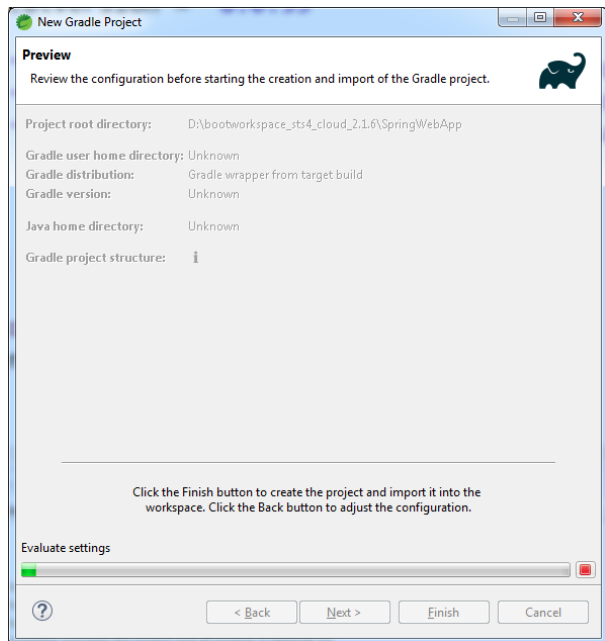
# Spring Web Application using Gradle

## STEP#1 CREATE NEW GRADLE PROJECT

> File > New > Gradle Project > Enter Project Name : SpringWebApp
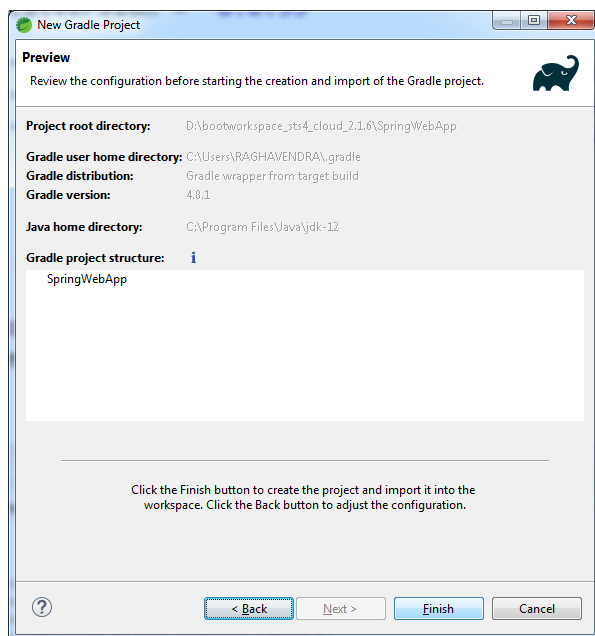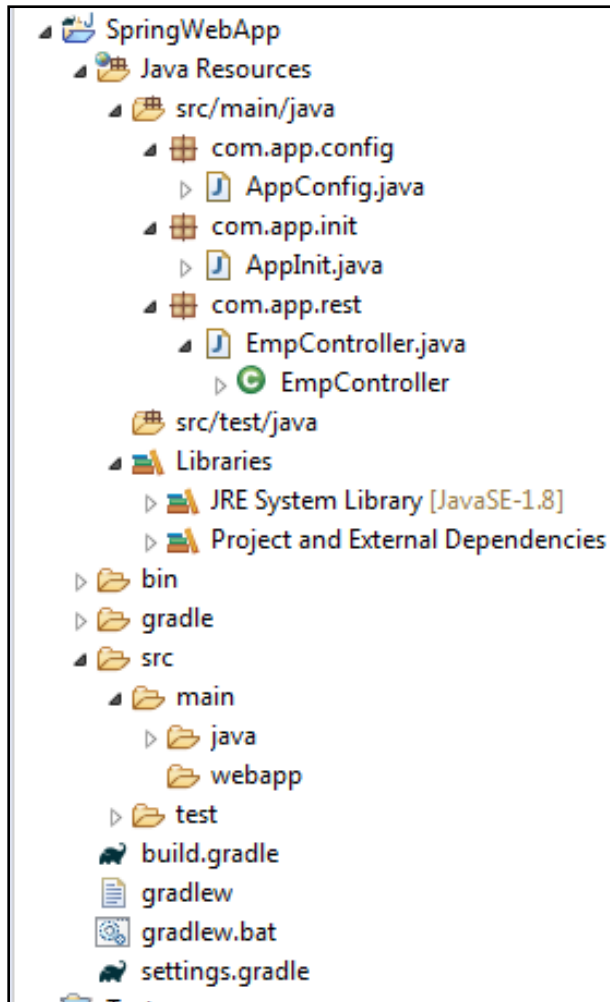


Click on Next Button

Click on Finish Button



# Folder System:-

# STEP#2 PROVIDE CODE IN PROJECT

**build.gradle**

```
apply plugin: 'java'
apply plugin: 'war'
apply plugin: 'com.bmuschko.tomcat'
apply plugin: 'eclipse-wtp'

sourceCompatibility = 1.8
targetCompatibility = 1.8

repositories {
  jcenter()
}

dependencies {
```

```groovy
  providedCompile 'javax.servlet:javax.servlet-api:3.1.0'
  compile 'org.springframework:spring-webmvc:5.1.8.RELEASE'
  compile 'com.fasterxml.jackson.core:jackson-databind:2.9.5'
  runtime 'javax.servlet:jstl:1.2'

  def tomcatVersion = '8.0.53'

  tomcat "org.apache.tomcat.embed:tomcat-embed-core:${tomcatVersion}",
      "org.apache.tomcat.embed:tomcat-embed-logging-
juli:${tomcatVersion}",
      "org.apache.tomcat.embed:tomcat-embed-jasper:${tomcatVersion}"
}
buildscript {

  repositories {
    jcenter()
  }

  dependencies {
    classpath 'com.bmuschko:gradle-tomcat-plugin:2.5'

  }
}
```

**AppInit.java**

```java
package com.app.init;

import
org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcher
ServletInitializer;

import com.app.config.AppConfig;

public class AppInit extends
AbstractAnnotationConfigDispatcherServletInitializer {

    @Override
    protected Class<?>[] getRootConfigClasses() {
```

```java
                    return null;
        }

        @Override
        protected Class<?>[] getServletConfigClasses() {
                return new Class[] {AppConfig.class};
        }

        @Override
        protected String[] getServletMappings() {
                return new String[] {"/*"};
        }

}
```

## AppConfig.java

```java
package com.app.config;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;

@Configuration
@ComponentScan("com.app")
@EnableWebMvc
public class AppConfig {


}
```
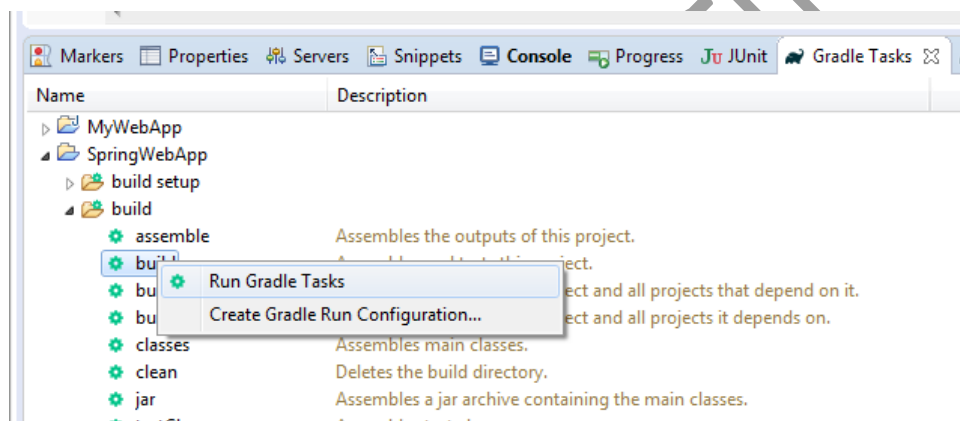
## RestController

```java
package com.app.rest;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
@RequestMapping("/emp")
public class EmpController {

        @GetMapping("/msg")
        public String show() {
                return "Hello R-APP";
        }
}
```
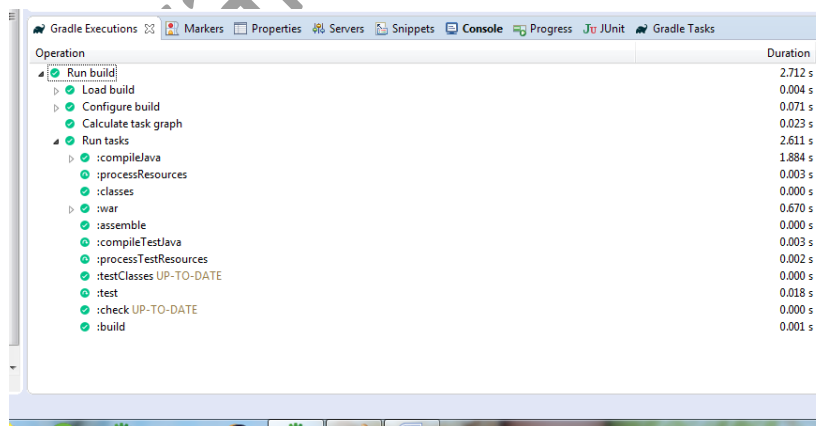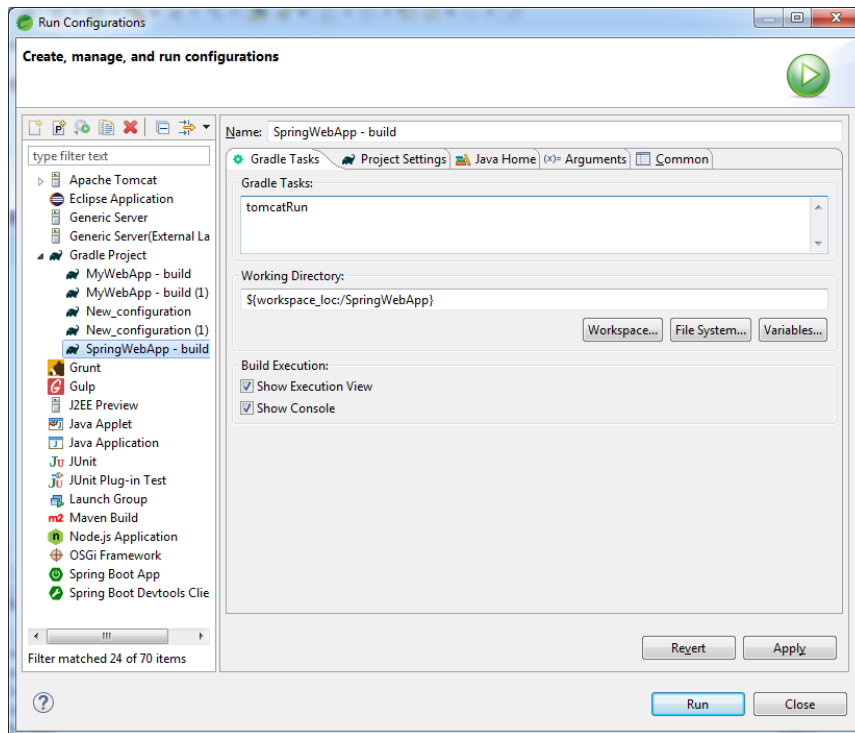
> Refresh Project

# STEP#3 RUN GRADLE TASK



> See Success Message Here

## STEP#4 RUN APPLICATION IN TOMCAT

> Right click on Project > Run As > enter tomcatRun > Apply and Run



## STEP#5 ENTER URL IN BROWSER

http://localhost:8080/SpringWebApp/emp/msg

FB:

**https://www.facebook.com/groups/thejavatemple/**