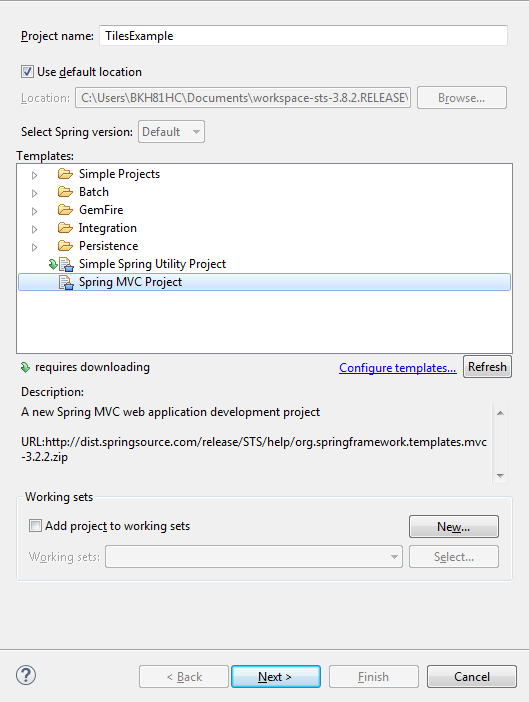
# Spring MVC with Apache Tiles 3.x.

## Create new MVC Spring project:

File -> new -> Spring Legacy project.

Choose Spring MVC Project and fill in Project name as below:

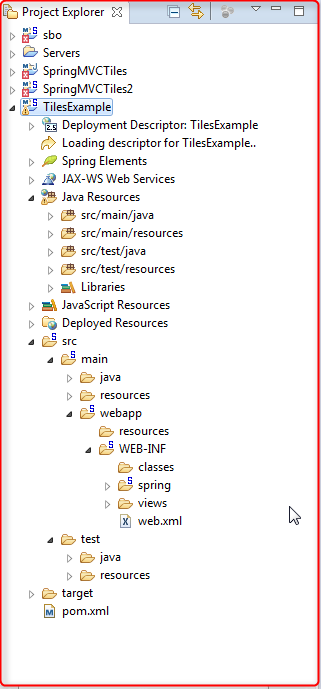


Click next.

**Enter top-level package, for example:**

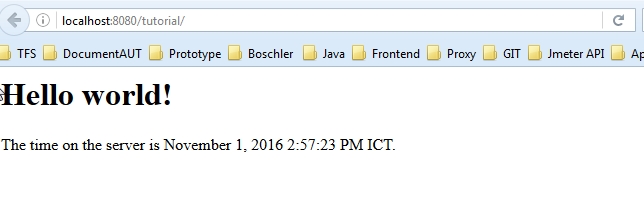
Com.khanh.tutorial.example

**After Project is created, upgrade project to Maven Project. Then you will have the structure as follow:**



**Under the src/main if there is no java/resources folder, then manually create them.**

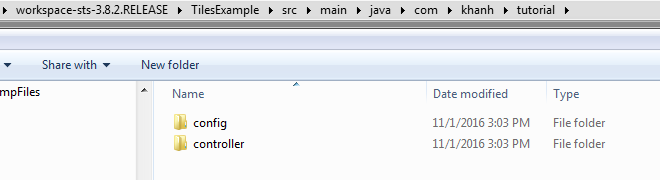
**The sample example should looks like this after run on server:**



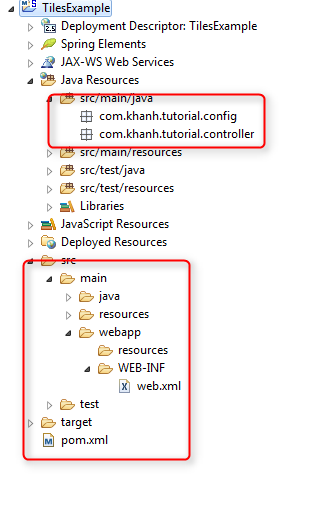
**Ok, now delete these file(if exsits):**

* Src/main/webapp/WEB-INF/classes
* Src/main/webapp/WEB-INF/spring
* Src/main/webapp/WEB-INF/views
* Src/main/java/com.khanh.tutorial/HomeController.java

**Next, manually go into com.khanh.tutorial.example and create 2 folders: config and controller.**



**Go back to your STS, refresh the project, now your project directory should looks like this:**



**Config your pom.xml:**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.khanh</groupId>

<artifactId>tutorial</artifactId>

<name>TilesExample</name>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<!-- Servlet Library -->

<!-- http://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>3.1.0</version>

<scope>provided</scope>

</dependency>

<!-- JSP API -->

<!-- http://mvnrepository.com/artifact/javax.servlet.jsp/jsp-api -->

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>2.2</version>

<scope>provided</scope>

</dependency>

<!-- Servlet JSP JSTL API -->

<!-- http://mvnrepository.com/artifact/javax.servlet.jsp.jstl/javax.servlet.jsp.jstl-api -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>javax.servlet.jsp.jstl</groupId>

<artifactId>javax.servlet.jsp.jstl-api</artifactId>

<version>1.2.1</version>

</dependency>

<!-- Spring dependencies -->

<!-- http://mvnrepository.com/artifact/org.springframework/spring-core -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>4.3.3.RELEASE</version>

</dependency>

<!-- http://mvnrepository.com/artifact/org.springframework/spring-web -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>4.3.3.RELEASE</version>

</dependency>

<!-- http://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>4.3.3.RELEASE</version>

</dependency>

<!-- Tiles API -->

<!-- http://mvnrepository.com/artifact/org.apache.tiles/tiles-api -->

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-api</artifactId>

<version>3.0.7</version>

</dependency>

<!-- Tiles Core -->

<!-- http://mvnrepository.com/artifact/org.apache.tiles/tiles-core -->

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-core</artifactId>

<version>3.0.7</version>

</dependency>

<!-- Tiles Servlet -->

<!-- http://mvnrepository.com/artifact/org.apache.tiles/tiles-servlet -->

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-servlet</artifactId>

<version>3.0.7</version>

</dependency>

<!-- Tiles JSP -->

<!-- http://mvnrepository.com/artifact/org.apache.tiles/tiles-jsp -->

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-jsp</artifactId>

<version>3.0.7</version>

</dependency>

</dependencies>

<build>

<finalName>SpringMVCTiles2</finalName>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.3</version>

<!-- Must update Maven Project -->

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<!-- Config: Maven Tomcat Plugin -->

<!-- http://mvnrepository.com/artifact/org.apache.tomcat.maven/tomcat7-maven-plugin -->

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<version>2.2</version>

<!-- Config: contextPath and Port (Default - /SpringMVCTiles : 8080) -->

<!--

<configuration>

<path>/</path>

<port>8899</port>

</configuration>

-->

</plugin>

</plugins>

</build>

</project>

**Your web.xml:**

**Configure Spring MVC:**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://java.sun.com/xml/ns/javaee"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee*

*http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"*

id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>SpringMVCTiles</display-name>

<servlet-mapping>

<servlet-name>default</servlet-name>

<url-pattern>/resource/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>default</servlet-name>

<url-pattern>\*.js</url-pattern>

<url-pattern>\*.css</url-pattern>

<url-pattern>\*.png</url-pattern>

<url-pattern>\*.gif</url-pattern>

<url-pattern>\*.jpg</url-pattern>

<url-pattern>\*.jpeg</url-pattern>

<url-pattern>\*.eot</url-pattern>

<url-pattern>\*.svg</url-pattern>

<url-pattern>\*.ttf</url-pattern>

<url-pattern>\*.woff</url-pattern>

</servlet-mapping>

</web-app>

Create 3 java classes as below:

**SpringWebAppInitializer.java**

**package** com.khanh.tutorial.config;

**import** javax.servlet.FilterRegistration;

**import** javax.servlet.ServletContext;

**import** javax.servlet.ServletException;

**import** javax.servlet.ServletRegistration;

**import** org.springframework.web.WebApplicationInitializer;

**import** org.springframework.web.context.support.AnnotationConfigWebApplicationContext;

**import** org.springframework.web.filter.CharacterEncodingFilter;

**import** org.springframework.web.servlet.DispatcherServlet;

**public** **class** SpringWebAppInitializer **implements** WebApplicationInitializer {

@Override

**public** **void** onStartup(ServletContext servletContext) **throws** ServletException {

AnnotationConfigWebApplicationContext appContext = **new** AnnotationConfigWebApplicationContext();

appContext.register(ApplicationContextConfig.**class**);

ServletRegistration.Dynamic dispatcher = servletContext.addServlet("SpringDispatcher",

**new** DispatcherServlet(appContext));

dispatcher.setLoadOnStartup(1);

dispatcher.addMapping("/");

// UtF8 Charactor Filter.

FilterRegistration.Dynamic fr = servletContext.addFilter("encodingFilter", CharacterEncodingFilter.**class**);

fr.setInitParameter("encoding", "UTF-8");

fr.setInitParameter("forceEncoding", "true");

fr.addMappingForUrlPatterns(**null**, **true**, "/\*");

}

}

**ApplicationContextConfig.java:**

**package** com.khanh.tutorial.config;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.web.servlet.ViewResolver;

**import** org.springframework.web.servlet.view.UrlBasedViewResolver;

**import** org.springframework.web.servlet.view.tiles3.TilesConfigurer;

**import** org.springframework.web.servlet.view.tiles3.TilesView;

@Configuration

@ComponentScan("com.khanh.tutorial.\*")

**public** **class** ApplicationContextConfig {

@Bean(name = "viewResolver")

**public** ViewResolver getViewResolver() {

UrlBasedViewResolver viewResolver = **new** UrlBasedViewResolver();

// TilesView 3

viewResolver.setViewClass(TilesView.**class**);

**return** viewResolver;

}

@Bean(name = "tilesConfigurer")

**public** TilesConfigurer getTilesConfigurer() {

TilesConfigurer tilesConfigurer = **new** TilesConfigurer();

// TilesView 3

tilesConfigurer.setDefinitions("/WEB-INF/tiles.xml");

**return** tilesConfigurer;

}

}

**WebMvcConfig.java**

**package** com.khanh.tutorial.config;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.web.servlet.config.annotation.DefaultServletHandlerConfigurer;

**import** org.springframework.web.servlet.config.annotation.EnableWebMvc;

**import** org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;

**import** org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

@Configuration

@EnableWebMvc

**public** **class** WebMvcConfig **extends** WebMvcConfigurerAdapter {

@Override

**public** **void** addResourceHandlers(ResourceHandlerRegistry registry) {

// Default..

}

@Override

**public** **void** configureDefaultServletHandling(DefaultServletHandlerConfigurer configurer) {

configurer.enable();

}

}

**Configure the Apache Tiles (tiles.xml):**

Create apache Tiles xml: src/main/webapp/WEB-INF/tiles.xml

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!DOCTYPE tiles-definitions PUBLIC

"-//Apache Software Foundation//DTD Tiles Configuration 3.0//EN"

"http://tiles.apache.org/dtds/tiles-config\_3\_0.dtd">

<tiles-definitions>

<!-- Base Define -->

<definition name=*"base.definition"* template=*"/WEB-INF/layouts/classic.jsp"*>

<put-attribute name=*"title"* value=*""* />

<put-attribute name=*"header"* value=*"/WEB-INF/basefragments/\_header.jsp"* />

<put-attribute name=*"menu"* value=*"/WEB-INF/basefragments/\_menu.jsp"* />

<put-attribute name=*"body"* value=*""* />

<put-attribute name=*"footer"* value=*"/WEB-INF/basefragments/\_footer.jsp"* />

<put-list-attribute name=*"stylesheets"*>

<add-attribute value=*"/resources/css/style.css"* />

</put-list-attribute>

</definition>

<!-- Home Page -->

<definition name=*"homePage"* extends=*"base.definition"*>

<put-attribute name=*"title"* value=*"Home Page"* />

<put-attribute name=*"body"* value=*"/WEB-INF/bodyfragments/\_home.jsp"* />

</definition>

<!-- ContactUs Page -->

<definition name=*"contactusPage"* extends=*"base.definition"*>

<put-attribute name=*"title"* value=*"Contact Us"* />

<put-attribute name=*"body"* value=*"/WEB-INF/bodyfragments/\_contactus.jsp"* />

<put-list-attribute name=*"stylesheets"* inherit=*"true"*>

<add-attribute value=*"/resources/css/contactUs.css"* />

</put-list-attribute>

</definition>

<!-- About Khanh -->

<definition name=*"aboutPage"* extends=*"base.definition"*>

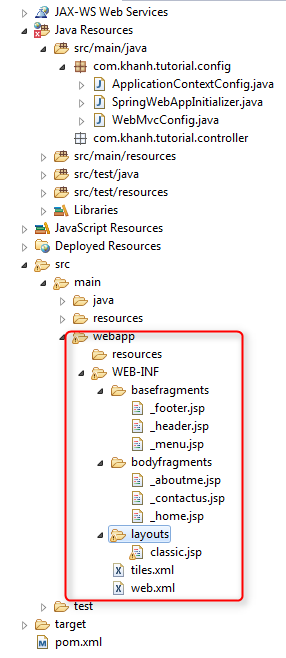
<put-attribute name=*"title"* value=*"About Me"* />

<put-attribute name=*"body"* value=*"/WEB-INF/bodyfragments/\_aboutme.jsp"* />

</definition>

</tiles-definitions>

**Create template structure:**



**Now edit page content:**

\_footer.jsp

<div

style="background: *#E0E0E0*; text-align: *center*; padding: *5px*; margin-top: *10px*;">

@Copyright Khanh Bui

</div>

\_header.jsp

<div style="background: *#E0E0E0*; height: *55px*; padding: *5px*;">

<div style="float: *left*">

<h1>My Site</h1>

</div>

<div class=*"colorRed"* style="float: *right*; padding: *10px*; text-align: *right*;">

Search <input name=*"search"*>

</div>

</div>

\_menu.jsp

<ul>

<li><a href=*"*${pageContext.request.contextPath}*/"*>Home</a></li>

<li><a href=*"*${pageContext.request.contextPath}*/contactus"*>Contact Us</a></li>

<li><a href=*"*${pageContext.request.contextPath}*/aboutme"*>About Me</a></li>

</ul>

\_aboutme.jsp

Name: ${name}

<br>

Address: ${address}

<br>

Phone: ${phone}

<br>

Email: ${email}

\_contactus.jsp

Contact Us: Khanh Bui

<br>

Address: ${address}

<br>

<div class=*"contactUs"*>Phone: ${phone}</div>

<br>

Email: ${email}

\_home.jsp

<h2>Hi All</h2>

<div class=*"colorRed"*>

This is Home Page

</div>

Layout.jsp

<%@taglib uri=*"http://tiles.apache.org/tags-tiles"* prefix=*"tiles"*%>

<%@taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"* %>

<tiles:importAttribute name=*"stylesheets"*/>

<html>

<head>

<title><tiles:getAsString name=*"title"* /></title>

<c:forEach var=*"css"* items=*"*${stylesheets}*"*>

<link rel=*"stylesheet"* type=*"text/css"* href=*"*<c:url value="${css}"/>*"*>

</c:forEach>

</head>

<body>

<table style="width:*100%*">

<tr>

<td colspan=*"2"*>

<tiles:insertAttribute name=*"header"* />

</td>

</tr>

<tr>

<td width=*"20%"* nowrap=*"nowrap"*>

<tiles:insertAttribute name=*"menu"* />

</td>

<td width=*"80%"*>

<tiles:insertAttribute name=*"body"* />

</td>

</tr>

<tr>

<td colspan=*"2"*>

<tiles:insertAttribute name=*"footer"* />

</td>

</tr>

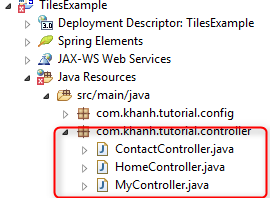
</table>

</body>

</html>

**Create Page Controllers:**

You will have to create 3 controller as bellow:



ContactController.java

package com.khanh.tutorial.controller;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

@Controller

public class ContactController {

@RequestMapping(value = { "/contactus" }, method = RequestMethod.GET)

public String contactusPage(Model model) {

model.addAttribute("address", "Vietnam");

model.addAttribute("phone", "...");

model.addAttribute("email", "...");

return "contactusPage";

}

}

HomeController.java

package com.khanh.tutorial.controller;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

@Controller

public class HomeController {

@RequestMapping(value = { "/", "/home" }, method = RequestMethod.GET)

public String homePage(Model model) {

return "homePage";

}

}

MyController.java

package com.khanh.tutorial.controller;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

@Controller

public class MyController {

@RequestMapping(value = { "/aboutme" }, method = RequestMethod.GET)

public String aboutmePage(Model model) {

model.addAttribute("name", "Bui Nguyen Thien Khanh");

model.addAttribute("Address", "Ho Chi Minh, Viet Nam");

model.addAttribute("email", "example@gmail.com");

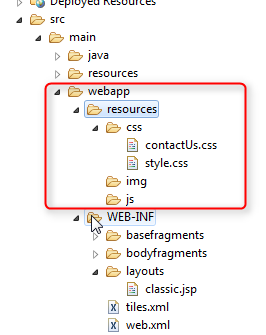
return "aboutPage";

}

}

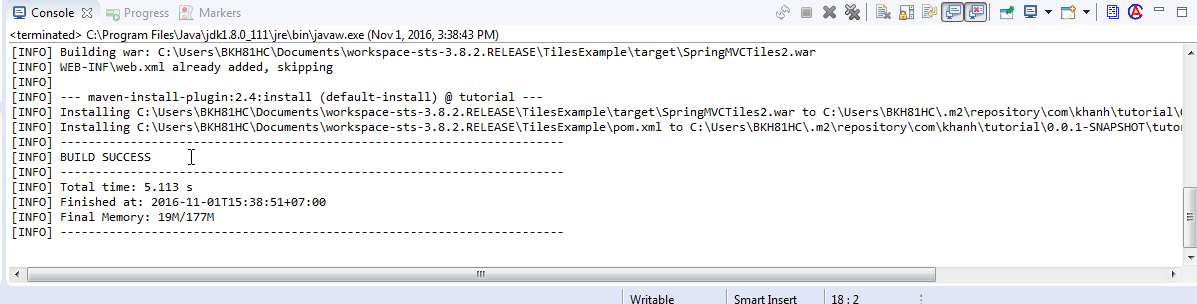
**CSS/JS/IMG Components:**

For CSS JS files will be stored in folder resources with is same directory with WEB-INF

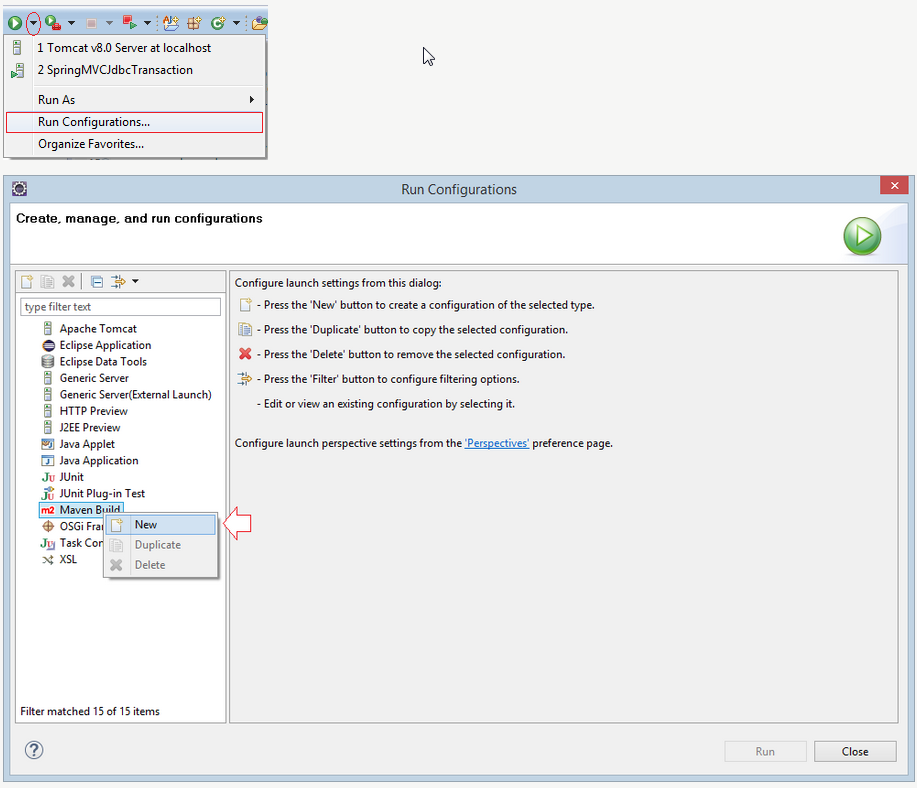


**Run Spring Application:**

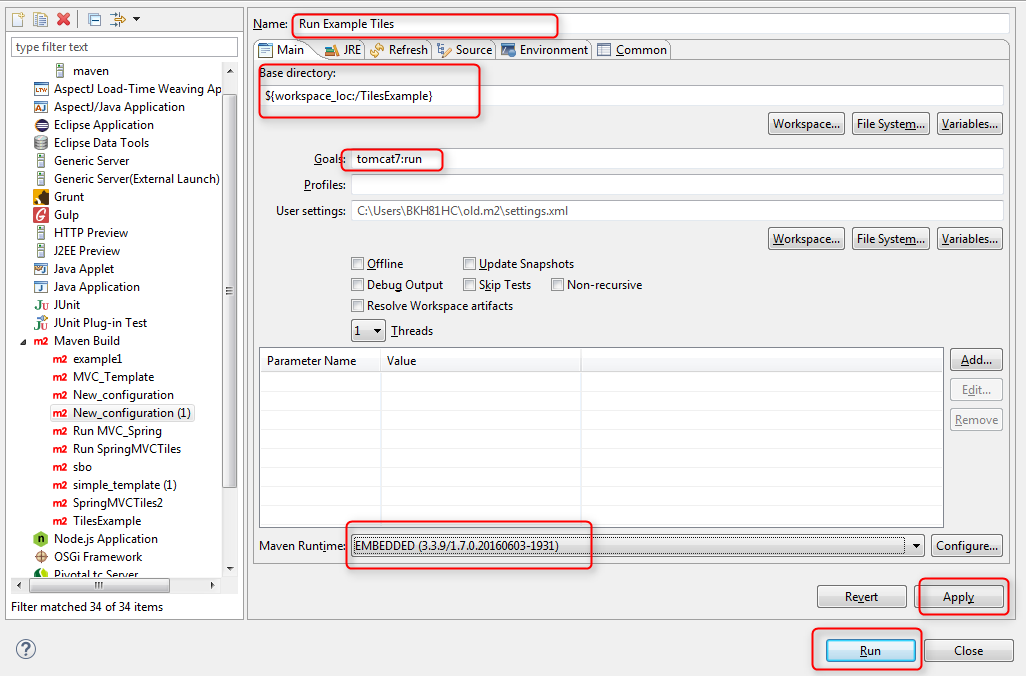
Right-click on the project then Run as -> Maven Install.



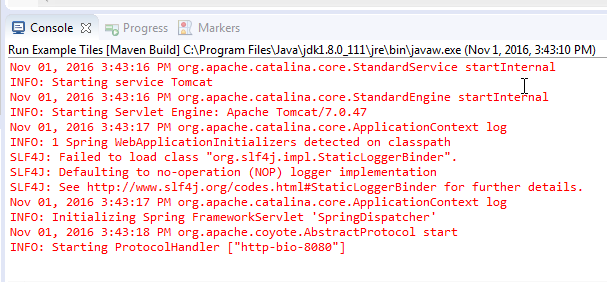
**Run configuration:**



**Config the Maven build like this:**



Then click apply -> run



Access: <http://localhost:8080/tutorial/>

Enjoy your first MVC Tiles template:

