MyTroubleshooting.com

Saturday, June 30, 2018

Java Spring Boot MVC And JSP With Maven And Visual Studio Code

In this tutorial I will explain step by step how to create from scratch a working Spring Boot MVC application, with JSP, using Maven and configure Visual Studio Code in order to develop and debug while using Hot Swap feature.

Highlight Of Versions Used In This Tutorial:

Java version: 1.8
Spring version: 2.1

Install Extensions

Open Visual Studio Code
Install extensions:

- Spring Boot Extension Pack (Pivotal)
- Java Extension Pack (Microsoft)

This extensions will install all other extensions needed for Spring and Java.
Reload VSCode to load installed extensions.
Wait until it ends importing Java Maven Project (look at the status bar)

Generate A Spring Boot Project

- Press F1 from VSCode
- Select "Spring Initializr: Generate a Maven Project" and press Enter
- Select "Java" and press Enter
- Type the group Id, we will use the default "com.example" and press Enter
- Type the artifact Id, we will use the default "demo" and press Enter
- Select the spring version, we will use "2.1.0 (SNAPSHOT)" and press Enter
- Select the required dependencies for this project: "Web, DevTools", I suggest to add also "Lombok, JPA, H2" to use the same project later for further examples and press Enter.
- Select a folder to save the project and press Enter

VS Code will ask if you want to move to the created folder, answer YES.

Add Support For JSP-EL

Edit pom.xml and add dependencies:

```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
</dependency>
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
</dependency>
```

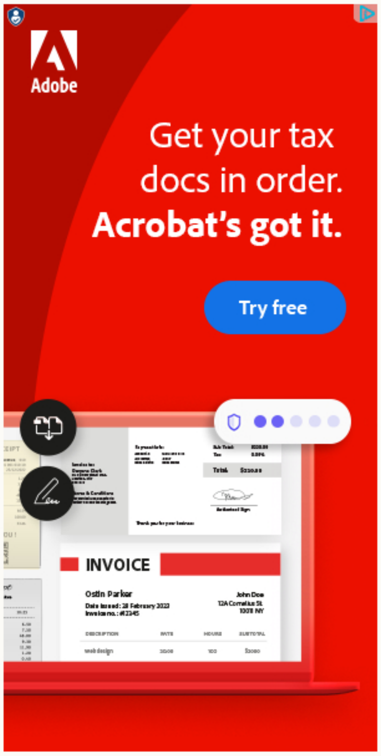
Be sure that in `.vscode/settings.json` has been added:

```
{
"java.configuration.updateBuildConfiguration": "interactive"
}
```

Edit main/resources/application.properties, adding the following:

Search This Blog

Search



Popular Posts

Java Spring Boot MVC and JSP with Maven and Visual Studio Code
In this tutorial I will explain step by step how to create from scratch a working Spring Boot MVC application, with JSP, using Maven an...

Echo server using python and sockets
This post explains how to create a create a simple echo server using python and sockets. In the first example we will create a simple...

Oracle Procedure to Write Log into a File
Sometimes you have to write in Oracle a long running time PL/SQL script and you want to log to file what is happening during the execution...

Solve ORA-01843: not a valid month error in TO_DATE function
It happens while importing some SQL script to have a ORA-01843: not a valid month error, when you try to execute a TO_DATE function in you...

Configure JNDI Mail Session with WebLogic and Spring, Gmail
In this tutorial I will explain how to configure a JNDI mail session using Oracle WebLogic 12c and Spring Framework and how to send email ...

Hibernate - Cannot insert explicit value for identity column in table 'MYTABLE' when IDENTITY_INSERT is set to OFF
When using Hibernate targeting an SQL Server database could happen to have the error "Cannot insert explicit value for identity co...

Maven copy configuration file or resource in target classes directory
Although building a maven project you have to use a specific directory for resources, there are some cases that you have to keep a resourc...

```
spring.mvc.view.prefix: /WEB-INF/views/
spring.mvc.view.suffix: .jsp
# logging.level.org.springframework: TRACE
# logging.level.com: TRACE
welcome.message: Hello
```

The first two rows are needed to set up JSP view resolver.
You can uncomment logging levels if you want to trace what is happening.
The third row is just put as example of how to use a property in the code, we will use it later.

Create the directory:
`main/webapp/WEB-INF/views`
and create a file `index.jsp`
Put this content in the file:

```
<!DOCTYPE HTML>
<html>
<head>
    <title>Main page</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
    <p>${greeting}</p>
</body>
</html>
```

Add A Controller

Create the directory:
`com/example/demo/controllers`

And create a file `HomeController.java`
Fill the file with this content:

```
package com.example.demo.controllers;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class HomeController {

    // inject via application.properties
    @Value("${welcome.message}")
    private String message = "Hello World";

    @RequestMapping("/")
    public String index(Model model) {
        model.addAttribute("greeting", message);
        return "index";
    }
}
```

Debug The Application

Press F5 to launch debug configuration.

Select "**Java**" from the list to create a configuration to launch, a `.vscode/launch.json` file will be created.
This procedure will happen only the first time, when file launch.json is missing.

Press F5 again to execute the debugging.

Go to http://localhost:8080 and verify that it works.

While debugging, try changing the code, for example by adding something to model:

```
model.addAttribute("greeting", message + "!");
```

Reload the page and verify that changes are directly reflected and hot swap works correctly.
This is permitted by Spring Tools dependency and the configuration in settings.json.

Sql Server Procedure to Write Log into a File

Sometimes you have to write a long running time procedure and you want to log to file what is happening during the execution, in ord...

How to set up a windows proxy to redirect traffic using netsh

Suppose you want to expose a service running at a specified address/port, to another address/port, configuring a windows machine for actin...

Configure Visual Studio Code for Python and Django framework

Visual Studio Code is a great, versatile, cross platform IDE, customizable with lot of plugins. Unfortunately there is nothing working...

Labels

- bitbucket
- database
- django
- eclipse
- email
- git
- iis
- java
- javascript
- liferay
- maven
- network
- oracle
- portlet
- python
- spring
- sql server
- tomcat
- vaadin
- visual studio
- visual studio code
- weblogic
- windows

In the same way you can add controllers and new methods and hot swap still will do the job.

Happy coding!

Pubblicato da RedPelle a 2:53 PM

Etichette: java, maven, spring, visual studio code

9 comments:

Unknown February 12, 2019 at 8:27 PM

This application has no explicit mapping for /error, so you are seeing this as a fallback.

I get an error.
How can I solve this?

Reply

Replies

RedPelle March 29, 2020 at 2:38 PM

More likely your controller cannot be found. Be sure that:

1) During project creation you selected "Spring Web" (not "Spring Reactive Web" or "Spring Web Services"
2) your full controller path is:
src/main/java/com/example/demo/controllers/HomeController.java

Keep in mind that every spring component (and the package containing it) should be put under:
src/main/java/com/example/demo
in order to be recognized by Spring.

Reply

JULIO GPE. ALEMAN January 9, 2020 at 1:06 AM

Very nice!!!

thanks

Reply

Anonymous April 6, 2020 at 5:01 AM

in eclipse most of the file/folder created by eclipse itself , but in vscode we have to create those file/folder, nicely explained thanks, god bless !

Reply

Anonymous April 6, 2020 at 5:09 AM

i too want to ask how do we configuration registers and initializes the DispatcherServlet in vscode ?

Reply

Replies

RedPelle April 13, 2020 at 1:34 PM

Configuration is done automatically by making use of the application.properties file (you can just modify or add properties).
However, if you need more freedom or additional configurations, you can just create a new class, i.e. WebConfig.java and put it in the folder com/example/demo (the starting folder for autoscan):
This class should be annotated with @Configuration annotation and implements the WebMvcConfigurer interface, like this:

```
@Configuration
public class WebConfig implements WebMvcConfigurer {
// ...
// implement needed methods here
// ...
}
```

This class will be automatically recognized by Spring (as soon as it is put in the right folder)

Personal Informations

RedPelle

View my complete profile

This class will be automatically recognized by Spring (as soon as it is put in the right folder)
You can implement also the ServletContextAware interface if you need the servlet context too.

Reply

unknown December 29, 2021 at 7:44 AM

My code is not run , I am getting the index word, not getting index.jsp page value.

Reply

micky January 9, 2022 at 11:56 AM

Thanks this worked. I was just about to download eclipse and gave a last try to google spring mvc with vscode and found your tutorial. its totally working and i'm really glad you made it in such detail.

Reply

micky January 9, 2022 at 11:57 AM

Thank you for such a detailed tutorial. It really helped.

Reply

Enter Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)