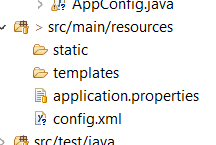
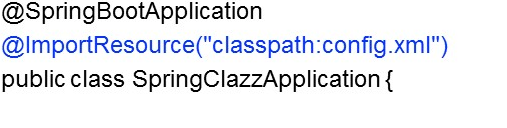
Spring boot:

1. Tạo pro vào start.spring.io, vào serch for depen chọn web: dùng mvn clean install để cài và mvn:spring-boot:run để chạy.
2. Cách thêm dispatcher:

Tạo 1 file config trong



Trong application main thêm import resource:



Cách 2 : tạo class app config:

package edu.hanoi.springclass;

import org.springframework.beans.factory.ListableBeanFactory;

import org.springframework.beans.factory.ObjectProvider;

import org.springframework.boot.autoconfigure.http.HttpMessageConverters;

import org.springframework.boot.autoconfigure.web.ResourceProperties;

import org.springframework.boot.autoconfigure.web.servlet.WebMvcAutoConfiguration.WebMvcAutoConfigurationAdapter;

import org.springframework.boot.autoconfigure.web.servlet.WebMvcProperties;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.servlet.DispatcherServlet;

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

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

import org.springframework.web.servlet.config.annotation.ViewControllerRegistry;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

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

import org.springframework.web.servlet.view.InternalResourceViewResolver;

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

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

import org.springframework.web.servlet.view.tiles3.TilesViewResolver;

import edu.hanoi.controller.HomeController;

@Configuration

@EnableWebMvc

public class AppConfig implements WebMvcConfigurer {

@Bean

public InternalResourceViewResolver viewResolver() {

InternalResourceViewResolver viewResolver = new InternalResourceViewResolver();

viewResolver.setPrefix("/WEB-INF/jsp/");

viewResolver.setSuffix(".jsp");

return viewResolver;

}

@Bean

public HomeController homeController(){

return new HomeController();

}

@Bean(name = "tilesConfigurer")

public TilesConfigurer getTilesConfigurer() {

TilesConfigurer configurer = new TilesConfigurer();

configurer.setDefinitions("/tiles/definitions.xml");

return configurer;

}

@Bean

public TilesViewResolver tilesViewResolver() {

final TilesViewResolver resolver = new TilesViewResolver();

resolver.setViewClass(TilesView.class);

return resolver;

}

@Override

public void configureDefaultServletHandling(

DefaultServletHandlerConfigurer configurer) {

configurer.enable();

}

}

1. Dùng tiles:

Thêm thư viện:

<dependency>

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

<artifactId>tiles-extras</artifactId>

<version>3.0.5</version>

</dependency>

Tạo definition trong tiles:

<?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">

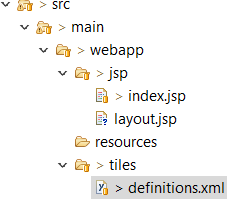
<tiles-definitions>

<definition name="template" template="/WEB-INF/jsp/layout.jsp">

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

</definition>

</tiles-definitions>



Tạo layout jsp:

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<table border="1" cellpadding="2" cellspacing="2" align=center>

<tr>

<td height="30"> Welcome to Java Spring Clazz</td>

</tr>

<tr>

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

</tr>

</table>

</body>

</html>

Tạo index jsp:

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

<tiles:insertDefinition name="template">

<tiles:putAttribute name="body">

${message}

</tiles:putAttribute>

</tiles:insertDefinition>

Thêm vào appconfig:

@Bean(name = "tilesConfigurer")

public TilesConfigurer getTilesConfigurer() {

TilesConfigurer configurer = new TilesConfigurer();

configurer.setDefinitions("/tiles/definitions.xml");

return configurer;

}

1. Biên dịch ra file war:

Tạo appliecation.properties trong target:

server.contextPath=/

server.port=333

vào tar get và chạy java –jar <tên pro>-0.0.1-snapshot.war

tùy vào chạy ở đâu nó sẽ lấy application.properties ở đó

1. Log:

private final static Logger logger = Logger.getLogger(HomeController.class);

với dòng log:

logger.info("Da truy cap");

1. Log:

Thêm depen:

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

<exclusions>

<exclusion>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-logging</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-log4j</artifactId>

<version>1.3.8.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

Dùng lệnh:

private final static Logger logger = Logger.getLogger(HomeController.class);

@RequestMapping("/")

public ModelAndView home(){

ModelAndView mv = new ModelAndView("index");

mv.addObject("message","hello");

logger.info("Da truy cap");

return mv;

}

Tạo file log4j.properties trong main/resource:

# Root logger option

log4j.rootLogger=DEBUG, stdout, file

# Redirect log messages to console

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.Target=System.out

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n

# Redirect log messages to a log file

log4j.appender.file=org.apache.log4j.RollingFileAppender

#outputs to Tomcat home

log4j.appender.file.File=F:/myapp.log

log4j.appender.file.MaxFileSize=5MB

log4j.appender.file.MaxBackupIndex=10

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n