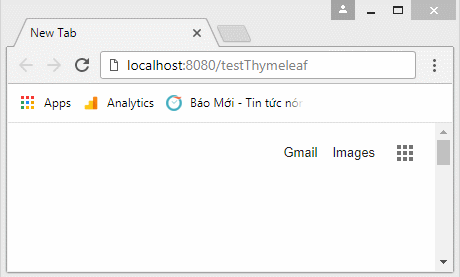
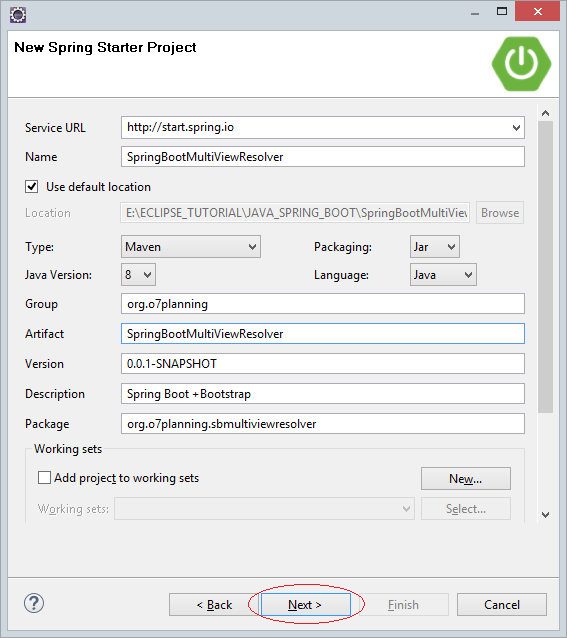
**Лабораторна робота № 23 «Використання кількох ViewResolver в Spring Boot»**

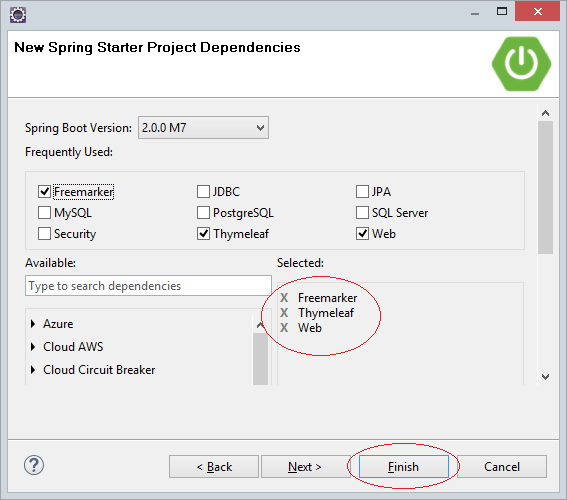
Зазвичай у прикладенні Spring вам потрібно використовувати лише одну технологію для рівня View , це може бути Thymeleaf , JSP або FreeMarker. Але ви так само можете використовувати різні технології одночасно для рівня View. У цій статті я покажу вам, як створити такий додаток.

Гаразд, ми створили додаток, використовуючи 3 одночасно технології Thymeleaf, JSP і FreeMarker для рівня View .



2- Створити проект Spring Boot





3- Конфігурація pom.xml

Щоб використовувати JSP, Thymeleaf, FreeMarker, вам потрібно мати наступні бібліотеки:

<!-- Thymeleaf -->

<**dependency**>

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

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

</**dependency**>

<!-- FreeMarker -->

<**dependency**>

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

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

</**dependency**>

<!-- Web -->

<**dependency**>

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

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

</**dependency**>

<!-- JSP -->

<**dependency**>

<**groupId**>org.apache.tomcat.embed</**groupId**>

<**artifactId**>tomcat-embed-jasper</**artifactId**>

</**dependency**>

<!-- JSTL -->

<**dependency**>

<**groupId**>javax.servlet</**groupId**>

<**artifactId**>jstl</**artifactId**>

</**dependency**>

Повний вміст файлу pom.xml :

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/xsd/maven-4.0.0.xsd">

<**modelVersion**>4.0.0</**modelVersion**>

<**groupId**>org.o7planning</**groupId**>

<**artifactId**>SpringBootMultiViewResolver</**artifactId**>

<**version**>0.0.1-SNAPSHOT</**version**>

<**packaging**>jar</**packaging**>

<**name**>SpringBootMultiViewResolver</**name**>

<**description**>Spring Boot + Multi ViewResolver</**description**>

<**parent**>

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

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

<**version**>2.0.0.RELEASE</**version**>

<**relativePath**/> <!-- lookup parent from repository -->

</**parent**>

<**properties**>

<**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>

<**project.reporting.outputEncoding**>UTF-8</**project.reporting.outputEncoding**>

<**java.version**>1.8</**java.version**>

</**properties**>

<**dependencies**>

<!-- Thymeleaf -->

<**dependency**>

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

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

</**dependency**>

<!-- FreeMarker -->

<**dependency**>

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

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

</**dependency**>

<!-- Web -->

<**dependency**>

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

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

</**dependency**>

<!-- JSP -->

<**dependency**>

<**groupId**>org.apache.tomcat.embed</**groupId**>

<**artifactId**>tomcat-embed-jasper</**artifactId**>

</**dependency**>

<!-- JSTL -->

<**dependency**>

<**groupId**>javax.servlet</**groupId**>

<**artifactId**>jstl</**artifactId**>

</**dependency**>

<**dependency**>

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

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

<**scope**>test</**scope**>

</**dependency**>

</**dependencies**>

<**build**>

<**plugins**>

<**plugin**>

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

<**artifactId**>spring-boot-maven-plugin</**artifactId**>

</**plugin**>

</**plugins**>

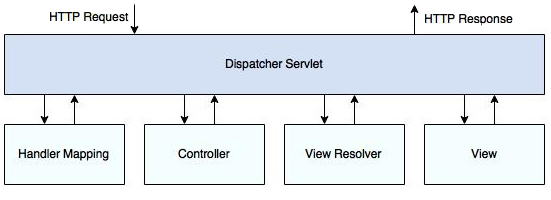
</**build**>

</**project**>

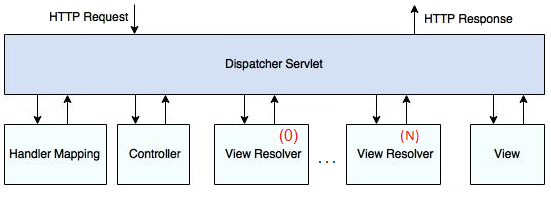
4- Конфігурація ViewResolver

При розробці проекту Spring Boot ви зазвичай використовуєте лише одну технологію для рівня View ( JSP, Thymeleaf, ..), Spring Boot автоматично налаштовує для вас ViewResolver, щоб ви працювали з цією технологією. Але у випадку, якщо ви використовуєте багато технологій для рівня View , вам потрібно самим конфігурувати всі потрібні ViewResolver .

Це зображення потоку (Flow) додаток Spring у випадку, коли ви використовуєте 1 ViewResolver .



У випадку, якщо ви використовуєте багато технологій для рівня View , буде багато ViewResolver , які беруть участь у потоці (потоку) додатків. ViewResolver встановлено за пріоритетами (0, 1, 2, ..). Якщо ViewResolver (0) не знайдено потрібний *"View Name"* , буде використано ViewResolver (1).



Конфігурація Thymeleaf ViewResolver .

ThymeleafViewResolverConfig.java

**package** org.o7planning.sbmultiviewresolver.config;

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

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

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

**import** org.thymeleaf.spring5.SpringTemplateEngine;

**import** org.thymeleaf.spring5.templateresolver.SpringResourceTemplateResolver;

**import** org.thymeleaf.spring5.view.ThymeleafViewResolver;

**import** org.thymeleaf.templatemode.TemplateMode;

**import** org.thymeleaf.templateresolver.ClassLoaderTemplateResolver;

**import** org.thymeleaf.templateresolver.ITemplateResolver;

@Configuration

**public** **class** **ThymeleafViewResolverConfig** {

@Bean

**public** ViewResolver **thymeleafViewResolver**() {

ThymeleafViewResolver viewResolver = **new** **ThymeleafViewResolver**();

viewResolver.setTemplateEngine(thymeleafTemplateEngine());

viewResolver.setCharacterEncoding("UTF-8");

viewResolver.setOrder(0);

// Important!!

// th\_page1.html, th\_page2.html, ...

viewResolver.setViewNames(**new** **String**[] { "th\_\*" });

**return** viewResolver;

}

// Thymeleaf template engine with Spring integration

@Bean

**public** SpringTemplateEngine **thymeleafTemplateEngine**() {

SpringTemplateEngine templateEngine = **new** **SpringTemplateEngine**();

templateEngine.setTemplateResolver(thymeleafTemplateResolver());

templateEngine.setEnableSpringELCompiler(true);

**return** templateEngine;

}

@Bean

**public** SpringResourceTemplateResolver **springResourceTemplateResolver**() {

**return** **new** **SpringResourceTemplateResolver**();

}

// Thymeleaf template resolver serving HTML 5

@Bean

**public** ITemplateResolver **thymeleafTemplateResolver**() {

ClassLoaderTemplateResolver templateResolver = **new** **ClassLoaderTemplateResolver**();

templateResolver.setPrefix("templates/");

templateResolver.setCacheable(false);

templateResolver.setSuffix(".html");

templateResolver.setTemplateMode(TemplateMode.HTML);

templateResolver.setCharacterEncoding("UTF-8");

**return** templateResolver;

}

}

У цьому додатку ми конфігуруємо Thymeleaf ViewResolver з найвищим пріоритетом (order = 0).

Конфігурація FreeMarker ViewResolver .

FreeMarkerViewResolverConfig.java

**package** org.o7planning.sbmultiviewresolver.config;

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

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

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

**import** org.springframework.web.servlet.view.freemarker.FreeMarkerConfigurer;

**import** org.springframework.web.servlet.view.freemarker.FreeMarkerViewResolver;

@Configuration

**public** **class** **FreeMarkerViewResolverConfig** {

@Bean(name = "viewResolver")

**public** ViewResolver **getViewResolver**() {

FreeMarkerViewResolver viewResolver = **new** **FreeMarkerViewResolver**();

viewResolver.setCache(true);

viewResolver.setPrefix("/freemarker/");

viewResolver.setSuffix(".ftl");

viewResolver.setOrder(1);

**return** viewResolver;

}

@Bean(name = "freemarkerConfig")

**public** FreeMarkerConfigurer **getFreemarkerConfig**() {

FreeMarkerConfigurer config = **new** **FreeMarkerConfigurer**();

// Folder containing FreeMarker templates.

// 1 - "/WEB-INF/views/"

// 2 - "classpath:/templates"

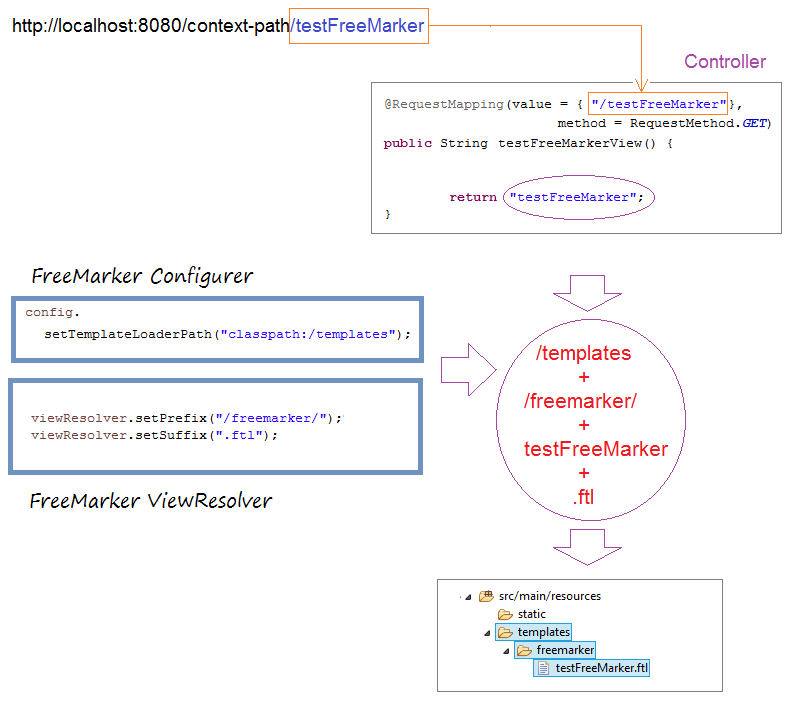
config.setTemplateLoaderPath("classpath:/templates");

**return** config;

}

}

Принцип роботи FreeMarker ViewResolver :



Конфігурація JSP ViewResolver .

*Примітка: JSP ViewResolver повинен бути представлений у найнижчому пріоритеті. Це прокоментовано в документах у Spring .*

JspViewResolverConfig.java

**package** org.o7planning.sbmultiviewresolver.config;

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

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

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

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

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

@Configuration

**public** **class** **JspViewResolverConfig** {

@Bean

**public** ViewResolver **jspViewResolver**() {

InternalResourceViewResolver viewResolver = **new** **InternalResourceViewResolver**();

viewResolver.setViewClass(JstlView.class);

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

viewResolver.setSuffix(".jsp");

viewResolver.setContentType("text/html");

// Make sure > Thymeleaf order & FreeMarker order.

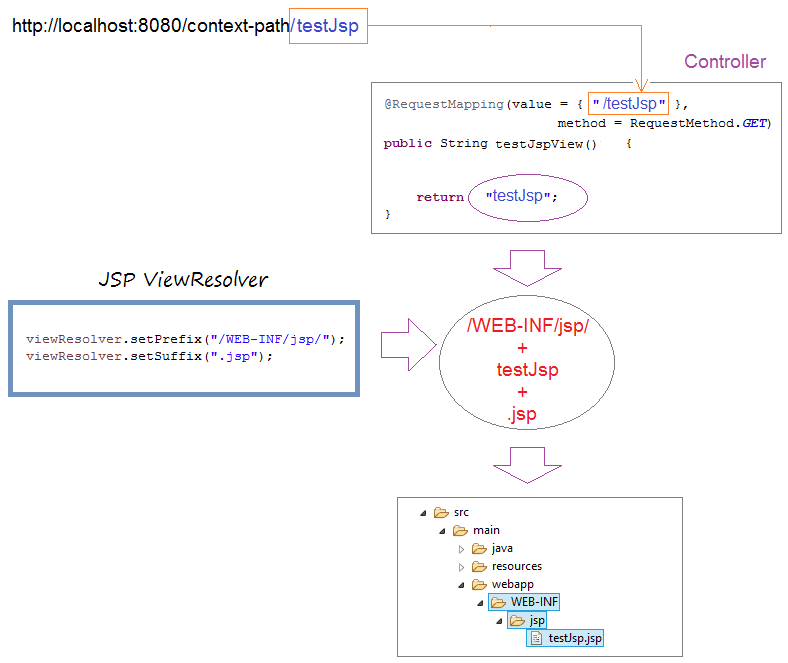
viewResolver.setOrder(1000);

**return** viewResolver;

}

}

Принцип роботи JSP ViewResolver :



5- Контролер

MainController.java

**package** org.o7planning.sbmultiviewresolver.controller;

**import** org.springframework.stereotype.Controller;

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

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

@Controller

**public** **class** **MainController** {

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

**public** String **testJspView**() {

**return** "testJsp";

}

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

**public** String **testThymeleafView**() {

**return** "th\_page1";

}

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

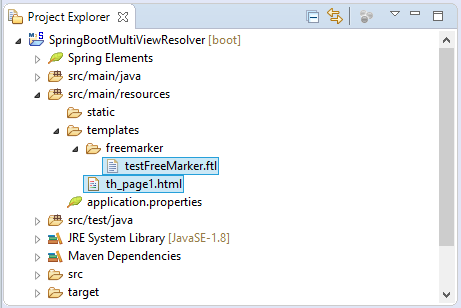
**public** String **testFreeMarkerView**() {

**return** "testFreeMarker";

}

}

6- Перегляди



th\_page1.html (Thymeleaf)

<!DOCTYPE **html**>

<**html** lang="en">

<**head**>

<**title**>Thymeleaf</**title**>

</**head**>

<**body**>

<**h2**>Thymeleaf Page</**h2**>

<**p**>templates/th\_page1.html</**p**>

</**body**>

</**html**>

freemaker/testFreeMarker.ftl (FreeMarker)

<!DOCTYPE **html**>

<**html** lang="en">

<**head**>

<**title**>FreeMarker</**title**>

</**head**>

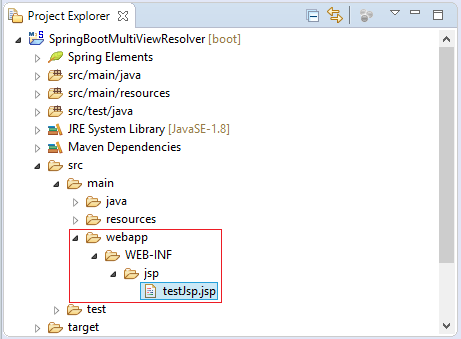
<**body**>

<**h2**>FreeMarker Page</**h2**>

<**p**>templates/freemarker/testFreeMarker.ftl</**p**>

</**body**>

</**html**>



testJsp.jsp

<!DOCTYPE **html**>

<**html** lang="en">

<**head**>

<**title**>JSP</**title**>

</**head**>

<**body**>

<**h2**>JSP Page</**h2**>

<**p**>WEB-INF/jsp/testJsp.jsp</**p**>

</**body**>

</**html**>