|  |  |
| --- | --- |
| 30-10-2017  Hướng dẫn sử dụng Spring mvc - Java Giảng viên: Ths. Phạm Minh Tú | Cài đặt IntelliJ Idea────Sử dụng Web server Tomcat────Tạo project Maven────Cấu hình Spring MVC────Tạo Model - View - Controller Trường Đại học Khoa học Tự nhiên  Bộ môn Hệ thống thông tin |

**1. Cài đặt Java, IntelliJ Idea, Tomcat**

Tải tại link: <https://www.jetbrains.com/idea/download/#section=windows>

Lưu ý: Trước khi cài đặt IntelliJ Idea, máy của bạn phải cài đặt trước JDK Java từ 7.0 trở lên.

Tải và cài đặt JDK (Java): <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Các bước cài đặt JDK và IntelliJ Idea khá đơn giản, phạm vi bài này không đi chi tiết vào các bước cài đặt.

Tomcat là web server xử lý các request từ người dùng, quản lý chương trình web application trên môi trường java như: Servlet, Jsp,…

Tải tại link: <https://tomcat.apache.org/download-80.cgi>

Giải nén và không cần cài đặt.

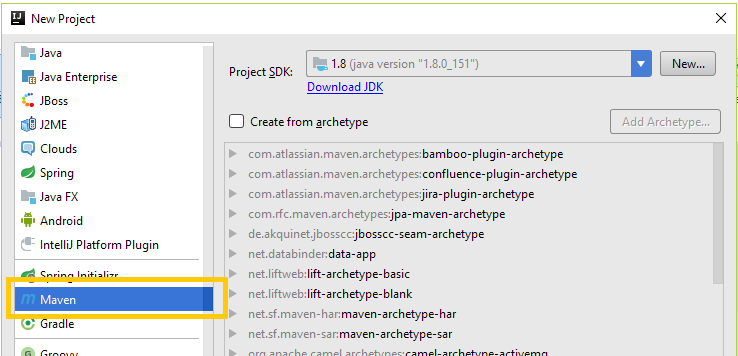
Phần sau sẽ hướng dẫn cách tích hợp Tomcat vào IntelliJ Idea.

**2. Tạo project Maven**

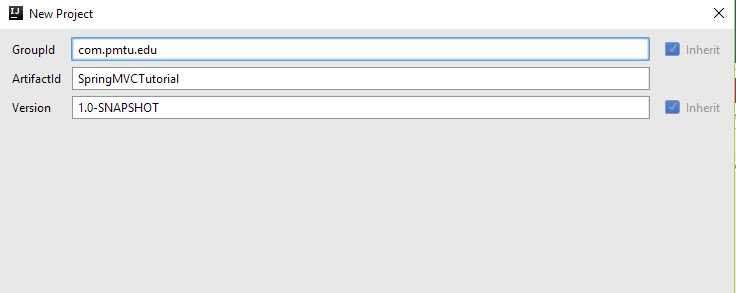
Maven là kho lưu trữ tất cả các thư viện cần thiết trong ứng dụng web Java. Thật ra, Maven còn làm nhiều thứ hơn nữa, ví dụ: biên dịch thư viện, lưu trữ, đóng vai trò quan trọng trong phát triển dự án trong nhóm.

Các bước tạo project Maven

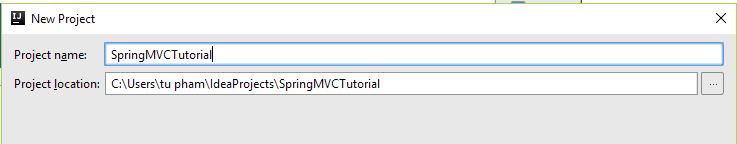
Bước 1: Khi tạo project chọn mục Maven



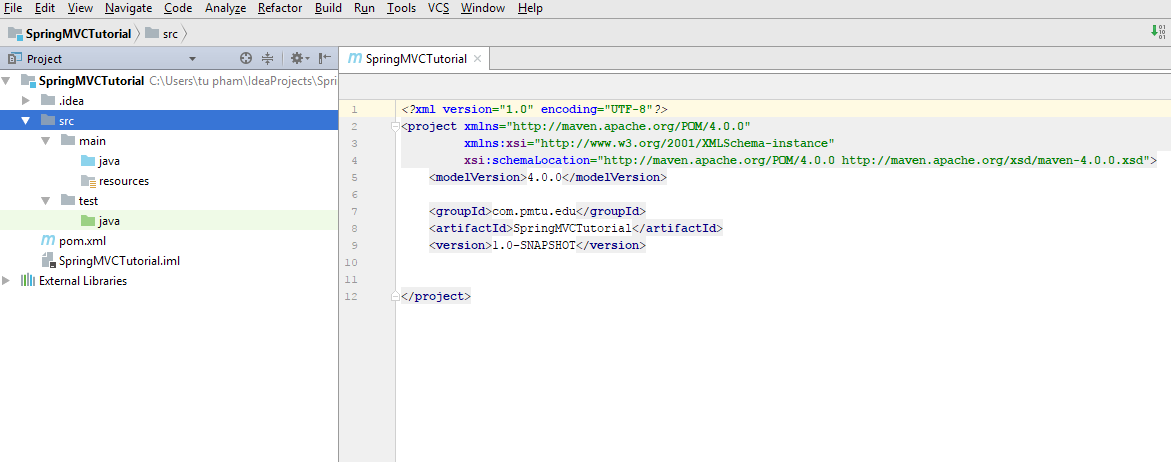
Bước 2: Cung cấp thông tin cơ bản project



Nhấn Next



Bước 3: Nhấn các nút Finish để hoàn thành tạo project



Lưu ý: Để ý file **pom.xml**, chính là nơi cấu hình các thư viện cần dùng trong Project, nó tương tự như **nuget package manager** trong Visual Studio.

**3. Cấu hình Spring MVC**

Phần này giới thiệu các bước cấu hình Spring MVC, nói cách khác các bước để có thể dùng được framework MVC dùng Spring MVC

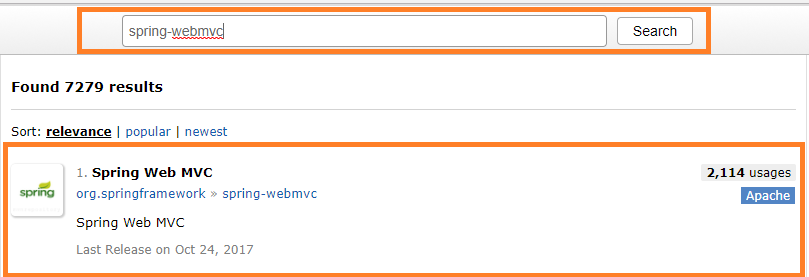
Bước 1: Khai báo các thư viện dùng trong Spring MVC, chúng ta vào trang maven và tìm kiếm các thư viện như sau:

* spring-webmvc
* spring-core
* javax.servlet-api
* spring-web

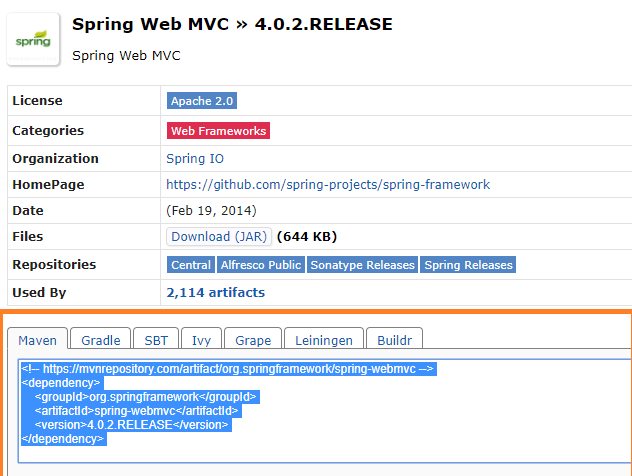
Đầu tiên search spring-webmvc trên kho dữ liệu maven

Kho: <https://mvnrepository.com/>

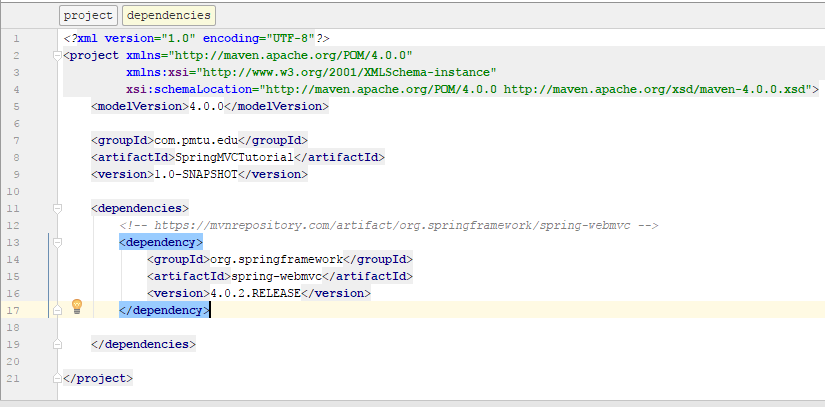
Search:



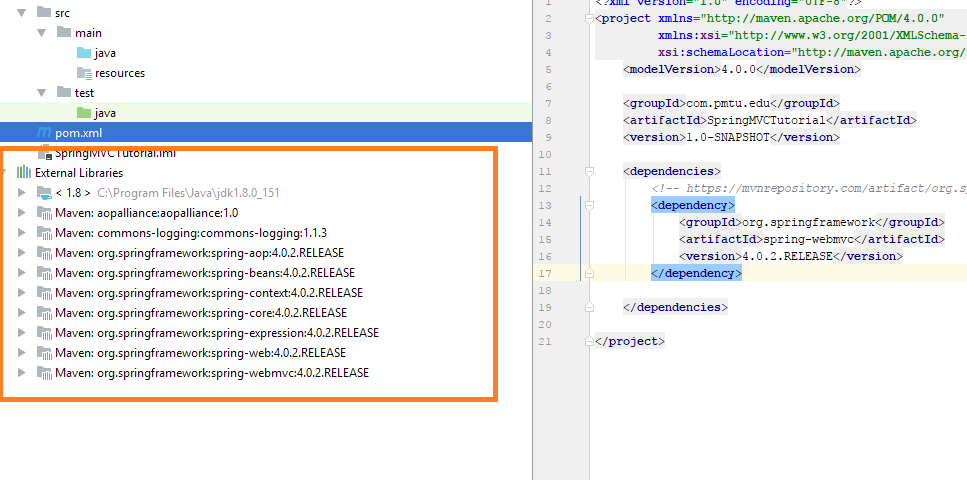
Copy thông tin vào trong file pom.xml. Lưu ý nhớ chọn thông tin phiên bản, trong bài này chúng ta dùng phiên bản: **4.0.2**



Paste vào file pom.xml. Tự động maven sẽ tải thư viện này vào máy chúng ta.



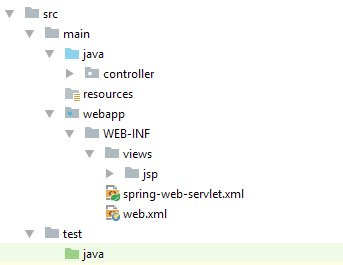
Lưu ý: Cần kết nối internet và chờ trong giây lát, nhanh hay chậm tùy dung lượng thư viện và tốc độ mạng.



Hình trên minh họa maven đã tải các thư viện xuống máy local. Tương tự thực hiện các bước này cho các thư viện còn lại.

Bước 2: Sau khi đã tải các thư viện cần thiết cho Project. Tiếp tục cấu hình MVC dùng XML Config. Bước này chúng ta làm việc nhiều với file XML, mỗi thông tin được cấu hình theo hướng dẫn của tài liệu Spring. Trong bài này chỉ cấu hình những thông tin cơ bản nhất, ngoài ra còn các thông tin cấu hình khác có thể xem thêm tại đây: <https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html>

Hãy tạo cẩn thận các thư mục trong Project theo hình bên dưới:



Với file web.xml, hãy copy và paste nội dung như bên dưới:

|  |
| --- |
| <**web-app xmlns="http://java.sun.com/xml/ns/javaee"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"  version="2.5"**>   <**display-name**>Spring MVC Application</**display-name**>   <**servlet**>  <**servlet-name**>spring-web</**servlet-name**>  <**servlet-class**> org.springframework.web.servlet.DispatcherServlet  </**servlet-class**>  <**load-on-startup**>1</**load-on-startup**>  </**servlet**>   <**servlet-mapping**>  <**servlet-name**>spring-web</**servlet-name**>  <**url-pattern**>/</**url-pattern**>  </**servlet-mapping**>  </**web-app**> |

Trong đó:

<**servlet-class**> org.springframework.web.servlet.DispatcherServlet

</**servlet-class**>

Đây là nơi nhận các request từ người dùng, Spring sẽ quản lý việc này thông qua DispatcherServlet, vì vậy hãy nhập chính xác thông tin này.

<**servlet-mapping**>  
 <**servlet-name**>spring-web</**servlet-name**>  
 <**url-pattern**>/</**url-pattern**>  
</**servlet-mapping**>

Tên servlet cũng nên lưu ý, cuối cùng **url-pattern** quy định mẫu request ví dụ như: /abc.com/…. Trong ví dụ này, mẫu request đơn giản là tại đường dẫn gốc.

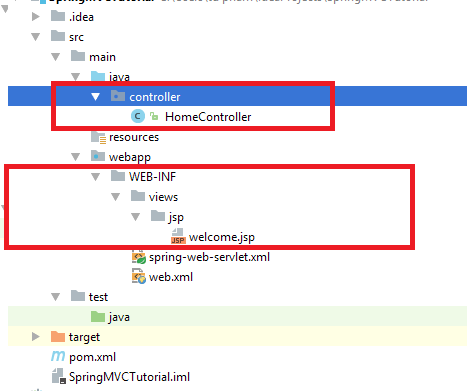
<http://localhost:8080/>

Tiếp theo là cấu hình Spring Servlet

Copy nội dung và paste vào file spring-web-servlet.xml:

|  |
| --- |
| <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:context="http://www.springframework.org/schema/context"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xsi:schemaLocation="  http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans-3.2.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context-3.2.xsd"**>   <**context:component-scan base-package="controller"** />   <**bean  class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>  <**property name="prefix"**>  <**value**>/WEB-INF/views/jsp/</**value**>  </**property**>  <**property name="suffix"**>  <**value**>.jsp</**value**>  </**property**>  </**bean**>  <**mvc:annotation-driven** />  </**beans**> |

Thông tin trong file này chủ yếu lưu 2 thông tin quan trọng. Nơi **Controller** và **View** được xác định. Trong trường hợp này, Controller được xác định là các class nằm trong package controller, và View được xác định bởi các trang jsp tại thư mục WEB-INF/views/jsp/.



Trong Project, có nhiều class **controller** và nhiều trang jsp là **view.** Tất cả xác định đầy đủ 3 thành phần Controller, View, Model. Model là đối tượng sẳn có trong Spring MVC hoặc những class java thông thường và trong class này phải có phương thức getter và setter tương ứng với các thuộc tính.

Hãy tạo class HomeController như trong hình và đặt nội dung như sau:

|  |
| --- |
| @Controller **public class** HomeController {   @RequestMapping(value = **"/"**, method = RequestMethod.***GET***)  **public** String printWelcome(Model model) {   model.addAttribute(**"message"**, **"This is a spring mvc tutorial"**);  **return "welcome"**;   } } |

Trong class này, phương thức printWelcome() trả về View tên là **welcome** với Model là cặp (**key**,**value**) là (message, This is a spring mvc tutorial). Khi View truy xuất thông tin Model, View chỉ cần truy xuất giá trị thông qua **key** của Model.

Ví dụ: ${key}

<**h1**>**${**message**}**</**h1**>

Hãy tạo trang JSP như hình trên và copy nội dung bên dưới:

|  |
| --- |
| <**html**>  <**head**>  <**title**>welcome</**title**>  </**head**>  <**body**>  <**h1**>**${**message**}**</**h1**>  </**body**>  </**html**> |

Lúc này, có thể hoàn thành các bước cơ bản cấu hình Spring MVC. Làm thế nào để Compile và Run ứng dụng này. Ta chỉ cần một Web Server như Tomcat, và chạy ứng dụng trên Tomcat này. May mắn là ItelliJ Idea cung cấp các công cụ cấu hình Tomcat rất tốt, chúng ta không mất nhiều công sức để làm việc này.

Hãy thực hiện theo các bước sau:

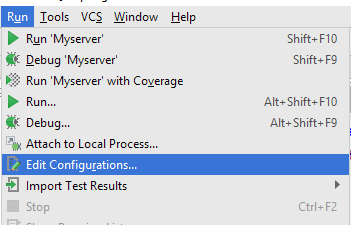
Đầu tiên hãy chú ý file pom.xml, web application trong Java phải được compile với định dạng war. Vì vậy hãy thêm câu lệnh cấu hình sau vào pom.xml.

<**packaging**>war</**packaging**>

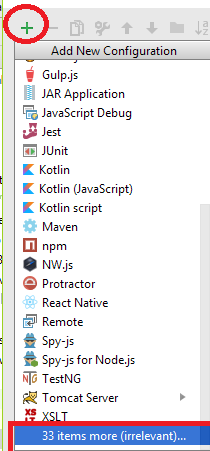
Toàn bộ cấu hình trong file pom.xml như bên dưới:

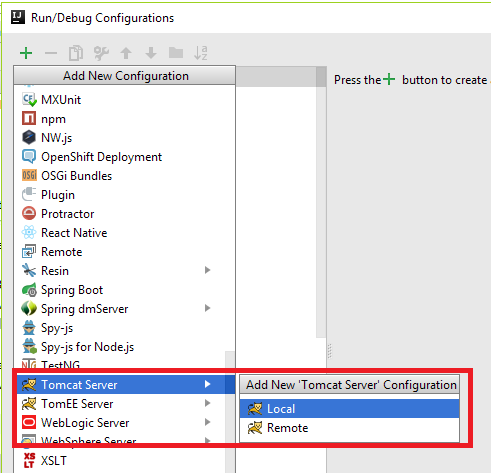
|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  <**modelVersion**>4.0.0</**modelVersion**>   <**groupId**>com.pmtu.edu</**groupId**>  <**artifactId**>SpringMVCTutorial</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  <**packaging**>war</**packaging**>  <**dependencies**>  *<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->* <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-webmvc</**artifactId**>  <**version**>4.0.2.RELEASE</**version**>  </**dependency**>  *<!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->* <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-core</**artifactId**>  <**version**>4.0.2.RELEASE</**version**>  </**dependency**>  *<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->* <**dependency**>  <**groupId**>javax.servlet</**groupId**>  <**artifactId**>javax.servlet-api</**artifactId**>  <**version**>4.0.0</**version**>  <**scope**>provided</**scope**>  </**dependency**>  *<!-- https://mvnrepository.com/artifact/org.springframework/spring-web -->* <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-web</**artifactId**>  <**version**>4.0.2.RELEASE</**version**>  </**dependency**>   </**dependencies**>  </**project**> |

Bước tiếp theo, cấu hình web server Tomcat. Chọn menu -> Run -> Edit Configuration

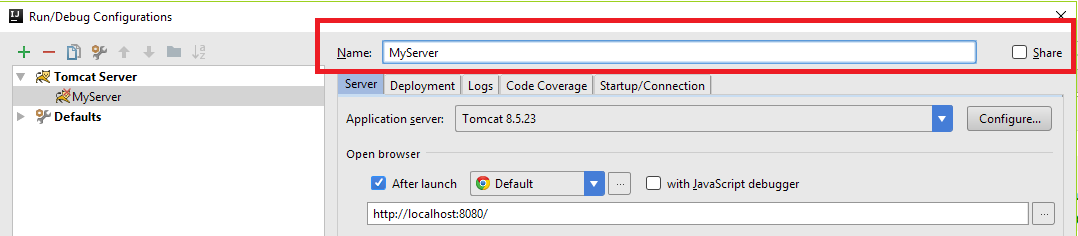


Nhấn biểu tượng + cộng để thêm một cấu hình Tomcat.

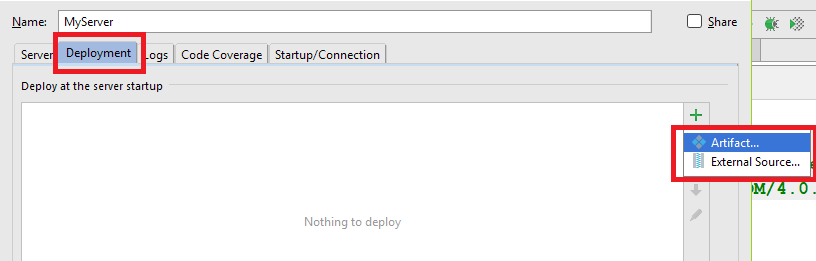


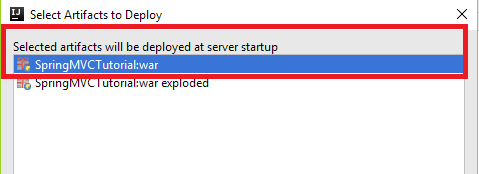


Lưu tên cấu hình là MyServer

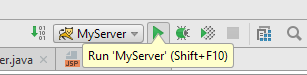


Tiếp theo chọn thẻ Deployment để chọn ứng dụng war triển khai trên Tomcat là có thể biên dịch và chạy chương trình.

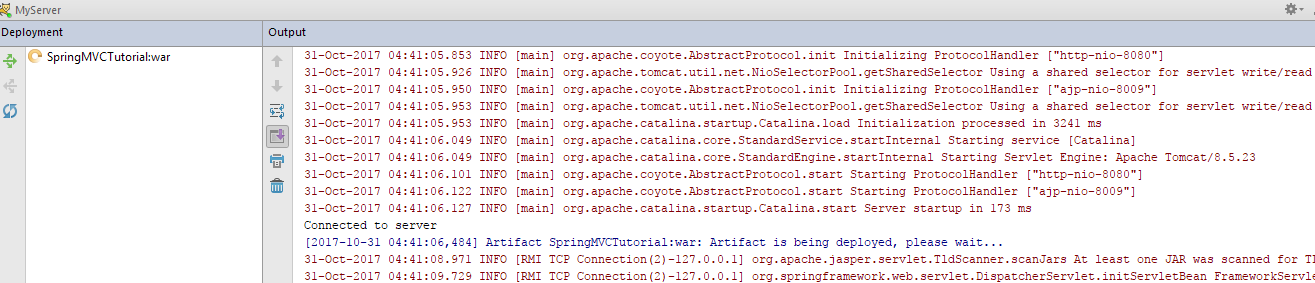




Cuối cùng nhấn OK và Run chương trình.



Hình ảnh lúc biên dịch và triển khai.



Kết quả hoàn thành.

