



NguyenDuong @duongichi

Follow

★ 987 👤 58 🛠 39

Published Sep 4th, 2016 1:06 PM - 6 min read

👁 3.0K 💬 0 🔖 3

TABLE OF CONTENTS

The
DispatcherServlet
Required
Configuration
Defining
a
Controller
Creating
JSP
Views

SUGGESTED ORGANIZATIONS

Sun* Cebu Branch /



Spring - MVC Framework Tutorial (Part 1)

Java Spring

...

Nguồn : http://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm

Spring MVC cung cấp kiến trúc model-view-controller và các thành phần để có thể được

một kết nối giữa các yếu tố này.

Model đóng gói dữ liệu ứng dụng và nói chung họ sẽ bao gồm các POJO.

Tầng View chịu trách nhiệm hiển thị các dữ liệu Model và nói chung nó tạo ra HTML mà trình duyệt hiển thị ra.

Controller chịu trách nhiệm xử lý yêu cầu người sử dụng và xây dựng Model phù hợp và chuyển nó qua tầng View để hiển thị.

The DispatcherServlet

MVC framework được thiết kế xoay quanh DispatcherServlet - cho phép xử lý tất cả các HTTP request và response. Sơ đồ dưới đây giải thích flow xử lý request của Spring Web MVC DispatcherServlet.

HTTP Request

HTTP Response



WELCOME TO VIBLO!

Let register a Viblo Account to get more interesting Posts from **NguyenDuong!**

Join Us

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up

1
Handler
Mapping

2
Controller

3
View Resolver

4
View

Đây là chuỗi các sự kiện tương ứng với một yêu cầu HTTP đến DispatcherServlet:

1. Sau khi nhận được một yêu cầu HTTP, DispatcherServlet chỉ định cho HandlerMapping gọi Controller thích hợp.
2. Controller sẽ nhận yêu cầu và gọi các Service tương ứng thích hợp dựa trên GET được sử dụng hoặc phương thức POST. Các phương thức service này sẽ thiết lập một nhóm các dữ liệu Model được định nghĩa theo logic business và trả về tên View cho DispatcherServlet.
3. Các DispatcherServlet sẽ được các ViewResolver hỗ trợ để chọn được View đã định nghĩa tương ứng với Request.



WELCOME TO VIBLO!

Let register a Viblo Account to get more interesting Posts from **NguyenDuong!**

Join Us

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up

số tính năng bổ sung cần thiết cho các ứng dụng web.

Required Configuration

Bạn cần một sơ đồ request mà bạn muốn DispatcherServlet xử lý, bằng cách sử dụng một ánh xạ URL trong file web.xml. Sau đây là một ví dụ thể hiện cho việc triển khai, lập bản đồ cho HelloWorld DispatcherServlet:

```
<web-app id="WebApp_ID" version="2.4"
xmlns="http://java.sun.com/xml/ns/j2ee"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
```

```
<display-name>Spring MVC Application</display-name>
```

```
<servlet>
  <servlet-name>HelloWeb</servlet-name>
  <servlet-class>
    org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```



WELCOME TO VIBLO!

Let [register a Viblo Account](#) to get more interesting Posts from [NguyenDuong!](#)

Join Us

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up

```
</web-app>
```



Các tập tin web.xml sẽ được lưu trong thư mục WebContent/WEB-INF trong application của bạn. Khi khởi tạo HelloWeb DispatcherServlet, framework sẽ load nội dung từ file [servlet-name]-servlet.xml đặt trong WebContent/WEB-INF. Trong trường hợp này thì file sẽ là HelloWeb-servlet.xml.

Tiếp theo, <servlet-mapping> cho thấy những URL sẽ được xử lý trong DispatcherServlet. Ở đây tất cả các request HTTP kết thúc với .jsp sẽ được xử lý bởi HelloWeb DispatcherServlet.

Nếu bạn ko muốn dùng default filename kiểu [servlet-name]-servlet.xml và vị trí là WebContent/WEB-INF, bạn có thể customize file name và đường dẫn bằng cách cho thêm servlet listener ContextLoaderListener trong web.xml như dưới đây :

```
<web-app...>
```



WELCOME TO VIBLO!

Let [register a Viblo Account](#) to get more interesting Posts from [NguyenDuong!](#)

Join Us

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up



```
<listener>
  <listener-class>
    org.springframework.web.context.ContextLoaderListener
  </listener-class>
</listener>
```

</web-app>

Hãy kiểm tra phần cấu hình cần thiết cho file HelloWorld-servlet.xml đặt trong WebContent/WEB-INF :

```
<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"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.0.xsd">

  <context:component-scan base-package="com.tutorialspoint" />

  <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
```



WELCOME TO VIBLO!

Let [register a Viblo Account](#) to get more interesting Posts from [NguyenDuong!](#)

Join Us

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up

Sau đây là những điểm quan trọng về file HelloWorld-servlet.xml:

1. Các file [servlet-name]-servlet.xml sẽ được sử dụng để tạo ra beans, override các định nghĩa của bất kỳ beans nào được định nghĩa ở scope global.
2. context:component-scan... sẽ được sử dụng để kích hoạt Spring MVC annotation quét cho phép sử dụng các chú thích như [@Controller](#) và [@RequestMapping](#) v.v...
3. Các InternalResourceViewResolver sẽ có những quy định để giải quyết các tên View. Theo các quy tắc định nghĩa ở trên, một View tên là Hello được /WEB-INF/jsp/hello.jsp.

Phần kế tiếp sẽ hướng dẫn bạn tạo các components như Controller, Model, View.

Defining a Controller

DispatcherServlet sẽ chuyển một request đến controllers để xử lý những chức năng được yêu cầu trong đó. annotation [@Controller](#) cho thấy rằng một class đặc biệt phục vụ một vai trò của một Controller. [@RequestMapping](#) được sử dụng để map một URL



WELCOME TO VIBLO!

Let [register a Viblo Account](#) to get more interesting Posts from [NguyenDuong!](#)

Join Us

```
@RequestMapping(method = RequestMethod.GET)
public String printHello(ModelMap model) {
    model.addAttribute("message", "Hello Spring MVC Framework!");
    return "hello";
}

}
```

@Controller định nghĩa một class như một controller Spring MVC. Ở đây, @RequestMapping cho thấy tất cả các methods trong controller này liên quan đến path /hello. @RequestMapping(method = RequestMethod.GET) được sử dụng để khai báo printHello() như là method service default của controller để xử lý HTTP GET request. Bạn có thể định nghĩa method khác để xử lý POST trên cùng một URL.

Bạn có thể viết controller ở trên trong form khác nơi bạn có thể cho thêm một vài thuộc tính bổ sung trong @RequestMapping như dưới đây :

```
@Controller
public class HelloController{
```



WELCOME TO VIBLO!

Let register a Viblo Account to get more interesting Posts from **NguyenDuong!**

[Join Us](#)

Thuộc tính giá trị cho thấy URL mà theo đó các method sẽ được map tương ứng và thuộc tính method định nghĩa service method để xử lý request POST GET. Có một vài điểm quan trọng cần chú ý về controller được định nghĩa ở trên :

1. Bạn sẽ định nghĩa logic business cần thiết trong một service method. Bạn có thể call một method khác bên trong method này.
2. Dựa trên logic business được định nghĩa, bạn có thể tạo nên một Model trong method này. Bạn có thể tạo thêm các thuộc tính model và các thuộc tính này có thể được access từ View để hiển thị kết quả cuối cùng. Đây là ví dụ tạo một Model với thuộc tính "message".
3. Method server có thể trả lại một String - chứa tên của View để sử dụng cho việc render Model. Đây là ví dụ trả lại view tên là "hello" .

Creating JSP Views



WELCOME TO VIBLO!

Let register a Viblo Account to get more interesting Posts from [NguyenDuong](#)!

Join Us

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up



0



```
<html>
<head>
<title>Hello Spring MVC</title>
</head>
<body>
<h2>${message}</h2>
</body>
</html>
```

`${message}` là một thuộc tính được setup trong Controller. Bạn có thể thực hiện hiển thị nhiều thuộc tính trong phía View.



Related

MVP Pattern for Android

Quân Nguyễn

Android Architecture: MVP or MVVM or Something Else

Chau Bao Long

Retrofit Android Tutorial to Get JSON from Server

Hoa Le Quang

Các Architecture Patterns phổ biến trong lập trình...

Nguyen Minh Duong



WELCOME TO VIBLO!

Let register a Viblo Account to get more interesting Posts from [NguyenDuong](#)!

Join Us

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up

Machine learning : K-means Clustering

NguyenDuong

12 min read

👁 1799 🗨 1 💬 0 ⬆ 5

Linear Regression - Hồi quy tuyến tính trong Machine...

NguyenDuong

19 min read

👁 12671 🗨 5 💬 0 ⬆ 5

Spring IoC, Spring Bean Example Tutorial

NguyenDuong

8 min read

👁 5132 🗨 4 💬 0 ⬆ 0

Java Dependency Injection – DI Design Pattern Example...

NguyenDuong

8 min read

👁 2641 🗨 7 💬 1 ⬆ 1

Comments

🗨 Login to comment

RESOURCES


[Posts](#)


[Organizations](#)

[Questions](#)

[Tags](#)

SERVICES

 [Viblo Code](#)

 [Viblo CV](#)

MOBILE APP

