

# Spring

---

IoC 和 AOP

Web Spring Web MVC

数据库 Spring Data、JdbcTemplate

安全校验 Spring Security

分布式 Spring Boot

微服务 Spring Cloud

## Spring MVC

---

Spring Framework 提供的 Web 组件，专门用来处理 Web 开发的组件

Servlet + JSP

Spring MVC 取代传统的基于 Servlet 的开发模式，为开发者提供更加便利的 Web 开发机制。

Spring MVC 流程

### 1、pom.xml 添加依赖

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>5.3.15</version>
</dependency>
```

### 2、web.xml 中配置 DispatcherServlet

```
<servlet>
  <servlet-name>springmvc</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:springmvc.xml</param-value>
  </init-param>
</servlet>

<servlet-mapping>
  <servlet-name>springmvc</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
```

### 3、创建 springmvc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

xmlns:context="http://www.springframework.org/schema/context"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd">

<!-- 自动扫描 -->
<context:component-scan base-package="com.southwind.handler">
</context:component-scan>

<!-- 配置视图解析器 -->
<bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/"></property>
    <property name="suffix" value=".jsp"></property>
</bean>

</beans>

```

#### 4、创建 Handler

```

package com.southwind.handler;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("/index")
public class IndexHandler {

    @RequestMapping("/test")
    public String test(){
        System.out.println("执行了test方法");
        return "test";
    }

}

```

#### 5、创建 jsp

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    <h1>test</h1>
</body>
</html>

```

给方法添加 @ResponseBody 注解，直接将方法的返回值返回到页面，而不通过视图解析器  
视图数据返回， ModelAndView

```

package com.southwind.handler;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

import java.util.Arrays;
import java.util.List;

@Controller
@RequestMapping("/index")
public class IndexHandler {

    @RequestMapping("/test")
    public ModelAndView test(){
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.setViewName("test");
        List<String> list = Arrays.asList("Java","Spring","Spring MVC");
        modelAndView.addObject("list",list);
        return modelAndView;
    }

}

```

```

<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page isELIgnored="false" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    <h1>test</h1>
    <c:forEach items="${list}" var="str">
        <h1>${str}</h1>
    </c:forEach>
</body>
</html>

```

## Spring MVC 数据校验

### 1、pom.xml

```

<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>7.0.0.Alpha4</version>
</dependency>
<dependency>
    <groupId>javax.validation</groupId>
    <artifactId>validation-api</artifactId>
    <version>2.0.1.Final</version>
</dependency>
<dependency>
    <groupId>org.jboss.logging</groupId>

```

```
<artifactId>jboss-logging</artifactId>
<version>3.4.3.Final</version>
</dependency>
```

## 2、通过注解在实体类 User 上进行标注

```
package com.southwind.entity;

import lombok.Data;
import org.hibernate.validator.constraints.NotEmpty;

@Data
public class User {
    @NotEmpty(message = "用户名不能为空")
    private Integer id;
    @NotEmpty(message = "姓名不能为空")
    private String name;
    @NotEmpty(message = "姓名不能为空")
    private Integer age;
}
```

```
@RequestMapping("/user")
public ModelAndView user(@Valid User user, BindingResult bindingResult){
    if(bindingResult.hasErrors()){
        List<ObjectError> allErrors = bindingResult.getAllErrors();
        for (ObjectError allError : allErrors) {
            System.out.println(allError.getDefaultMessage());
        }
    }
    System.out.println(user);
    ModelAndView modelAndView = new ModelAndView();
    modelAndView.setViewName("test");
    List<String> list = Arrays.asList("Java", "Spring", "Spring MVC");
    modelAndView.addObject("list", list);
    return modelAndView;
}
```