[项目结构](http://blog.csdn.net/caib1109/article/details/51495932#项目结构)

[Controller](http://blog.csdn.net/caib1109/article/details/51495932#controller)

[Service](http://blog.csdn.net/caib1109/article/details/51495932#service)

[Web配置](http://blog.csdn.net/caib1109/article/details/51495932#web配置)

[第一步准备包](http://blog.csdn.net/caib1109/article/details/51495932#第一步准备包)

[第二步在eclipse工程中建Dynamic Web project](http://blog.csdn.net/caib1109/article/details/51495932#第二步在eclipse工程中建dynamic-web-project)

项目结构

[spring](http://lib.csdn.net/base/javaee) 3.2.4 + jdk6 + tomcat7

src

|--com.abc

|--controller

| |--AbcdefgController.java

|--service

|--Service1.java

|--Service2.java

WEB-INF

|--web.xml #listener, server, context

|--abcdefg-servlet.xml #spring相关

Controller

@Controller

public class 项目名Controller {

@Autowired

Service1 service1; // 使用@Autowired标签必须规范命名 - 除首字母外和类名一致

@Autowired

Service2 service2;

@ResponseBody // 返回值以http response的正文形式返回给浏览器

@RequestMapping(value="/helloworld",method=RequestMethod.POST)

public String getHelloworld(@RequestParam(value="city") String city,

@RequestParam(value="project") String project) {

return service1.doSomething();

}

@ResponseBody

@RequestMapping(value="/hello",method=RequestMethod.GET)

public String getHello() {

return service2.doSomething();

}

}

Service

Service类负责具体业务, 最好是有个Service接口, 再实现这个接口. 面向接口编程的好处无需赘述.

public interface Service{

doSomething();

}

@Service

public class Service1 implements Service{

public String doSomething(){

// your business logic

Web配置

转自:<http://www.open-open.com/lib/view/open1330670747561.html>   
Spring MVC 3 最简单的入门示例   
这两天再看spring mvc，想再网上找一个demo，感觉国内的demo都太复杂了。后来在国外网站上发现了一个比较简单，应该说是最简单的spring mvc的demo了，在此做个记录，给需要的人了解一下。

第一步：准备包

日志相关包   
jcl-over-slf4j-1.6.1.jar   
logback-classic-0.9.29.jar   
logback-core-0.9.29.jar   
slf4j-api-1.6.1.jar

jstl包(jsp标签包)   
jstl-1.2.jar

spring 相关包

<!-- ===============================

ivy.xml

${spring.version}=3.2.4.RELEASE

=============================== -->

<dependency org="org.springframework" name="spring-aop" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-aspects" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-beans" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-context" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-context-support" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-core" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-expression" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-jdbc" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-tx" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-web" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-webmvc" rev="${spring.version}"/>

<dependency org="org.springframework" name="spring-test" rev="${spring.version}"/>

第二步：在eclipse工程中建Dynamic Web project

向导窗口，一路next，最后得到一个web工程。

在WebContent\WEB-INF目录下生成一个jsp文件夹和两个配置文件：spring-servlet.xml、web.xml

web.xml内容为

<?xml version="1.0" encoding="UTF-8"?>

<web-app version="2.5" 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">

<display-name>Spring3MVC</display-name>

<servlet>

<servlet-name>spring</servlet-name>

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

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>spring</servlet-name>

<url-pattern>\*.html</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

</web-app>

spring-servlet.xml内容为

<?xml version="1.0" encoding="UTF-8"?>

<beans default-autowire="byName"

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:tx="http://www.springframework.org/schema/tx"

xmlns:task="http://www.springframework.org/schema/task" xmlns:mvc="http://www.springframework.org/schema/mvc"

xmlns:util="http://www.springframework.org/schema/util"

xsi:schemaLocation="

http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.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.xsd

http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd">

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

</beans>

到此就完成了所有的代码和配置，所有的文档和配置加在一起只有60行左右，比较简单把。

完成后就可以放在tomcat等容器中运行一下，然后在web中打开地址。

在index.jsp上点击Say Hello，就会转向<http://localhost:8080/springmvcdemo/hello.html>   
页面显示Hello World, Spring 3.0!