

# 基于 Spring MVC 的网络存储系统<sup>①</sup>

丁振凡

(华东交通大学 信息工程学院, 南昌 330013)

**摘要:** 给用户提供的网络存储是许多应用所需要的功能. 讨论了利用 Spring MVC 实现该功能的解决办法. 可以让用户在网络空间下自由建立子目录、上传文件, 实现目录和文档的在线管理. 有效解决了目录名和文件名的汉字编码转换问题, 以及当前目录路径标识中斜杠符导致的 URI 匹配问题, 该应用已用于网络教学系统中学生个人空间和小组空间的信息保存, 为小组协作学习提供了基础.

**关键词:** Spring MVC; 文件上传; 文件下载; 用户网络文档空间; REST

## Network Storage System Based on Spring MVC

DING Zhen-Fan

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

**Abstract:** Network storage provided for user is the useful function for many application. The paper discuss the solution of Spring MVC. Users can create sub directory freely, upload files and manage directories as well as files on line. The system effectively solves the problem of the Chinese characters coding conversion about directory name and file name, as well as it solves the URI matching problem caused by the slash in current path. The application is used for information save at student personal space and group space in network teaching system, it provide the basics for group collaborative learning.

**Key words:** spring MVC; file upload; file download; user network document space; REST

在许多网络应用中可给用户提供一个网络存储空间. 用户可以将文件上传到服务器上自己的文件夹下面, 可以在自己的空间下建子目录.

本应用中假设所有用户的文档放在 d:\user 文件夹下. 每个用户有一个自己的根目录路径, 这个根目录和用户的登录名一致. 系统自动为用户建立根目录. 如图 1 所示.

Spring MVC 框架是一个受大众欢迎的应用开发模式<sup>[1]</sup>. Spring3.0 的 MVC 框架提供了注解符的表示形式, Spring 控制器的请求和处理风格符合 REST 架构的设计, 通过控制器、模型和视图的配合, 可有效提高应用开发效率<sup>[2]</sup>. Spring MVC 的工作过程如图 2 所示. ①Spring 通过 DispatcherServlet 这个特殊的控制器处理用户的请求; ②由该控制器根据配置信息查找对应

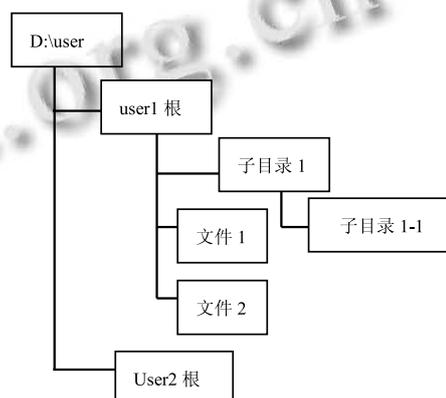


图 1 用户可在各自空间下建子目录及上传文件

的控制器, 实现控制分派; ③通过执行具体控制器的方法设置模型和视图; ④控制器将模型和视图名传递

① 收稿时间:2012-07-06;收到修改稿时间:2012-08-29

给视图解析器; ⑤通过视图解析器定位到视图文件进行解析处理; ⑥将结果通过 HTTP 响应返回给客户浏览器. 其中:

1) 模型(Model)用来表达应用的业务逻辑, Spring 通常用 HashMap 存储模型的处理结果.

2) 视图(View)用来表达应用界面, Spring 整合了多种视图层技术, 如 JSP、JSTL、FreeMarker 等.

3) 控制器(Controller)主要是接收用户请求, 依据不同的请求, 执行对应业务逻辑, 获取执行结果, 选择对应的视图返回给用户.

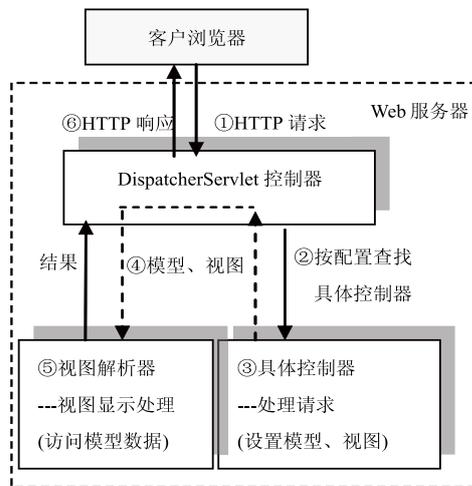


图 2 Spring MVC 工作过程

## 1 控制器的设计

### 1.1 控制器设计中的几个难点

#### 1.1.1 url 匹配问题

Spring MVC 支持 REST 风格 Web 服务的定义<sup>[3]</sup>. 为了显示当前目录下用户的子目录和文件列表, 一个基本问题是要记住当前所处的目录路径. 实现当前路径传递的较好办法是通过 URL 参数. 但由于当前路径中包含的斜杠分隔符在 REST 风格的 URI 识别中是路径匹配识别的重要标记. 在这种情况下, 对 REST 风格的 URI 模板要进行特殊设计. 控制器的映射定义采用在路径映射后加“\*”的方式, 表示匹配所有情形. 例如: /filedel\*, 其对应的功能是删除当前目录下文件. 被删文件和当前目录等通过 URL 参数传递.

#### 1.1.2 汉字编码转换问题

通过超链发送的请求是以 get 方式发送 HTTP 请求, 这类请求的缺省编码方式是 iso8859-1, 为了正确获取汉字路径信息, 需要进行编码转换. 具体处理办

法如下:

首先, 请求的 JSP 文件的 URL 超链设计中, 对传送的 URL 参数用 URLEncoder 进行编码. 例如, 以下对当前目录路径进行编码处理.

```
<a href="<%=request.getContextPath()+"/docs?currentpath="+URLEncoder.encode(mydir,"utf-8")%>
```

其次, 在控制器的相应代码中需要两步处理, 首先对字节串数据进行转码, 其次是对串用 URLEncoder 进行解码. 转码需要将 iso-8859-1 转为 UTF-8. 例如:

```
dir=URLDecoder.decode(new String(dir.getBytes("ISO-8859-1"),"UTF-8"),"UTF-8");
```

### 1.2 控制器 url 的具体设计

按照 MVC 设计规则, 用户请求将提交给控制器处理, Spring 通过 DispatcherServlet 处理用户的请求, 它将寻找匹配的 URI, 从而转向具体的控制器进行业务处理. 该应用设计了如下 REST 风格的 URI 映射.

1) 某目录的资源浏览(/docs\*): 类似资源管理器, 可自由浏览目录和文件列表. 通过参数 current path 传递当前目录信息. 改变参数可进入下级子目录或返回上级目录.

2) 在当前路径下创建子目录(/createdir): 由请求表单的参数传递要创建的子目录名和当前目录路径.

3) 上传文件到用户当前目录(/fileupload): 由表单参数传递当前目录和上传的文件.

4) 删除用户当前目录下文件(/filedel\*): 由 URL 参数传递当前目录和要删除的文件.

5) 删除用户当前目录下子目录(/dirdel\*): 由 URL 参数传递上级目录和要删除的子目录.

6) 文件下载(/downfile\*): 由 URL 参数传递文件路径和文件名.

程序清单 1 为针对某目录的资源浏览的功能相关的控制器部分代码. 其中仅列出了显示当前目录下资源列表的实现代码, 利用 Java 的 File 类的方法可容易获取目录下文件和子目录<sup>[4]</sup>.

```
【程序清单 1】文件为 MydocController.java
@Controller
public class MydocController {
    @RequestMapping(value="/docs*", method=RequestMethod.GET)
    public ModelAndView list(
```



```
<%=myfile.getName()%></a></td>
.....
</tr>
<%= } %>
</table>
```

### 3 文件上传的实现

Spring MVC 文件上传是由 spring 内置的 Commons MultipartResolver 解析器来实现的. 实现文件上传的具体编程处理步骤如下:

首先, 在工程的 lib 路径引入从 Apache 下载的“commons-fileupload-1.2.2.jar”和“commons-io-2.0.1.jar”两个 JAR 文件.

其次, 在 web 应用程序上下文配置文件中定义如下解析器:

```
<bean id="multipartResolver"
class="org.springframework.web.multipart.common
s.CommonsMultipartResolver">
<!-- 以字节为单位的最大上传文件的大小 -->
<property name="maxUploadSize" value="100000" />
</bean>
```

第三, 设置页面中请求表单的 enctype 属性为“multipart/form-data”, 在表单中通过类型为“file”的输入元素选择上传文件; 表单的提交方法为“POST”.

第四, 在处理上传请求的控制器中, 通过 CommonsMultipartFile 类型的参数对象获取上传文件数据信息. 其中, getBytes()方法可获取上传文件的数据内容, getOriginalFilename()方法可获取上传文件的文件名.

### 4 文件下载的实现

文件下载处理是利用文件输入流访问文件读取字节数据, 利用 Servlet 的响应输出流将字节数据送客户端. 程序中用到 2 个 URL 请求参数, file 为文件名, currentpath 为当前目录. 以下为关键处理步骤和代码:

第 1 步: 文件和目录路径的解码, 得到文件名 file.

第 2 步: 根据帐户、目录路径和文件名计算文件的物理路径, 创建文件对象 fileobj.

第 3 步: 给 HTTP 响应头添加文件附件信息. 文件下载的附件名称在生成时需要用 UTF8 进行编码处理, 否则在客户浏览器端附件文件名将显示为乱码.

```
response.setHeader("Content-Disposition","attachm
ent; filename=\"" + URLEncoder.encode(file, "UTF-8")
+ "\"");
```

// 对文件名进行编码处理

```
response.addHeader("Content-Length", "" + fileobj.
length());
```

第 4 步: 输出文件内容给浏览器.

```
response.setContentType("application/octet-stream;c
harset=UTF-8");
```

```
OutputStream outputStream = new Buffered
OutputStream(response.getOutputStream());
```

... //以字节流方式读文件数据写入 outputStream

### 5 结语

文章介绍用 Spring MVC 实现用户网络文档空间的具体方案. 有效地解决了因文件目录路径所带来的传递 REST 路径匹配以及文件名中汉字的编码处理等问题. 各用户的空间通过用户标识区分, 每个用户可对各自的空间进行文档管理. 用户可在自己的空间下建立子目录,并在各级目录路径中实现文件的上传和下载, 以及文件和目录的删除管理等, 达到了网络虚拟硬盘的效果.

#### 参考文献

- 1 翟高粤.基于 Spring MVC 校园管理系统的设计与实现.电脑编程技巧与维护,2010,17:9-12.
- 2 张宇,王映辉,张翔南.基于 Spring 的 MVC 框架设计与实现.计算机工程,2010,36(4):59-62.
- 3 丁振凡.Spring REST 风格 Web 服务的 Json 消息封装及解析研究.智能计算机与应用,2012,2(2):9-10.
- 4 丁振凡.Java 语言程序设计.北京:清华大学出版社,2010. 225-228.
- 5 薛峰,梁锋,徐书勋,王彪任.基于 Spring MVC 框架的 Web 研究与应用.合肥工业大学学报(自然科学版),2012,35(3): 337-340.