

利用 Spring MVC 实现数据分页显示处理

丁振凡

(华东交通大学 信息工程学院, 南昌 330013)

摘要: 结合资源查询需求, 讨论了用 Spring MVC 实现数据分页显示编程处理思路与技巧。在介绍 Spring MVC 的工作过程的基础上, 分析讨论了资源查询分页处理的设计过程。控制器中通过 URL 参数传递分页信息, 在业务逻辑中则通过方法参数传递分页信息, 通过 SQL 语句的关联查询处理, 可限制只从数据库读取当前页的数据。视图显示中, 通过 EL 表达式获取来自模型的数据。文中介绍的方法具有较大通用性, 实现了分页查询处理的高效化。

关键词: Spring 3.x; MVC; REST; 数据分页显示; JDBC 模板

中图分类号: TP393

文献标识码: A

文章编号: 2095-2163(2012)05-0020-03

Using Spring MVC to Realize Data Pagination Display Processing

DING Zhenfan

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

Abstract: Combined with resource query demand, the paper discusses the ideas and skills of using Spring MVC to realize data paging display programming. Based on the introduction of the Spring MVC process, the paper analyzes design process of resource query paging processing. Controller could transfer paging information through URL parameter, the business logic could pass the paging information through the method parameter, and through the SQL statement, relational query processing could be restricted only from the database to read the current page data. In addition, in view display, data could be obtained through the EL expression from the model. The method proposed in this paper has great versatility and good efficiency in realizing paging query processing.

Key words: Spring 3.x; MVC; REST; Data paging display; JdbcTemplate

0 引言

实际应用中, 显示数据量较大时, 经常需要进行分页浏览显示, 有些研究者已经开发了相关组件^[1], 但在使用上却存在着一定局限。结合资源查询的分页显示, 本文介绍了利用 Spring MVC 实现数据分页显示的处理方法。

MVC 框架是一个受大众欢迎的应用开发模式。通过控制器、模型和视图的配合, 可有效提高应用开发效率。Spring 3.0 的 MVC 框架提供了注解符的表示形式, 为满足 MVC 的工作需要, 可进行相应的配置。首先, 在 web.xml 配置文件中需要设定 DispatcherServlet 作为 Servlet 控制器; 其次, 在对应的 Servlet 配置文件中定义视图解析器, 用 mvc 命名空间的“annotation-driven”标记定义注解驱动, 以及通过 context 命名空间的“component-scan”标记定义扫描 Spring 部件的包路径, 具体控制器的源代码安排在该包路径下。Spring MVC 的工作过程分析如下。

(1) Spring 通过特殊的控制器 DispatcherServlet 处理用户的请求;

(2) 由 DispatcherServlet 控制器根据控制器注解的 Mapping 映射查找与 URL 请求处理相对应的具体控制器, 实现控制分派;

(3) 通过执行具体控制器的方法运行业务逻辑, 并设置模型和选择视图;

(4) 控制器将模型和视图名传递给视图解析器;

(5) 通过视图解析器分析视图文件和访问模型, 完成

解析处理;

(6) 将视图解析的网页结果通过 HTTP 响应返回给客户浏览器。

将上述工作过程做以形象化表示, 如图 1 所示^[2]。为了实现分页显示处理, 必须传递分页信息。具体实现时, 有三处与页码有关:

- (1) 访问请求的 URL 参数;
- (2) 业务逻辑中方法参数;
- (3) SQL 语句中取指定页的数据。

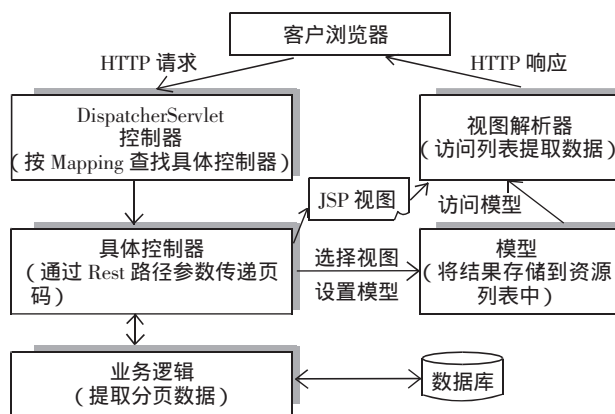


图 1 分页显示处理工作过程

1 数据对象及业务逻辑 Bean 配置

首先, 对要查询处理的数据对象进行封装, 在后续数据

收稿日期: 2012-09-04

作者简介: 丁振凡(1965-) 男, 江西丰城人, 学士, 教授, 硕士生导师, 主要研究方向: 语义 Web、分布式计算、计算机辅助教学。

处理中要访问其属性和方法。

```
public class MyResource {
    int resourceID; // 资源 ID
    String titleName; // 资源标题
    String description; // 资源描述
    .....// 其他属性及各属性的 getter 和 setter 方法为节省篇幅略
}
```

业务逻辑中通过 JdbcTemplate 访问数据库^[3], 在实现业务逻辑的 resourceManger 类中通过定义 JdbcTemplate 属性, 然后通过属性注入相关的对象。以下为具体配置, 本文采用的数据源是 Access 数据库, 数据库的文件名和路径为“f:/data.mdb”。

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    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">
    <bean id="dataSource"
        class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <property name="driverClassName" value="sun.jdbc.odbc.JdbcOdbcDriver" />
        <property name="url"
            value="jdbc:odbc:driver={Microsoft Access Driver (*.mdb)};DBQ=f:/data.mdb" />
    </bean>
    <bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="dataSource" />
    </bean>
    <bean id="resourceBean" class="ecjtu.resourceManger">
        <property name="jdbcTemplate">
            <ref local="jdbcTemplate" />
        </property>
    </bean>
</beans>
```

2 在业务逻辑中获取分页数据

编程时, 有两种实现提取指定页记录的办法: 方法一是先查询得到该类别资源的所有记录, 然后通过记录指针的定位, 提取指定页的那些记录; 方法二则直接通过 SQL 语句一次性限制将指定页的记录提取出来。显然, 第二种办法更为高效, 没有返回任何多余数据。方法一由于在数据量大时, 效率明显下降, 所以, 已有研究者提出使用缓存策略来进行优化^[4]。

方法二的关键是 SQL 语句的书写, 利用 top 关键词可从查询中获取顶部的若干条记录, 首先注意限制过滤掉 page 页之前的那些记录, 然后, 从剩下的记录中取出头 8 条记录。这里, 假定每页的大小固定为 8 条数据记录。

首先, 定义业务逻辑服务接口 resourceService。在定义的方法中, 与分页显示相关的方法包括查找某类资源的 list 方法和获取某类资源总记录条数的 getAmount 方法。

以下 resourceManger 类给出了资源访问处理的实现。

```
public class resourceManger implements resourceService {
    public List<MyResource> list (int classifyID,int page) { // 取第 page 页
        List<MyResource> m=jdbcTemplate.query(
            "select top 8 * from resource where classifyID ="
            +classifyID+
            " and resourceID not in ( select " + " top "+(1+
            8*(page-1)) +
            "resourceID from resource where classifyID="+classifyID+" )",
            new RowMapper<MyResource>() {
                public MyResource mapRow (ResultSet rs, int
                rowNum)
                    throws SQLException {
                        MyResource r = new MyResource();
                        r.setResourceID (rs.getInt("resourceID"));
                        // 设置资源 ID
                        r.setTitleName (rs.getString("titleName"));
                        // 设置资源标题
                        .....
                    }
            });
        return m;
    }
}
```

/* 为了便于计算总页数, 提供获取某类资源的总记录条数的方法 */

```
public int getAmount (int classifyID) { // classifyID 为资源的类别
    return jdbcTemplate.queryForInt ("select count (*) from resource where classifyID="
        +classifyID);
}
```

3 控制器设计

Spring3 MVC 控制器支持注解的定义形式。每个控制器对应有 REST 风格的 Mapping 路径映射定义, DispatcherServlet 根据请求 URL 寻找匹配路径映射模板的控制器。为满足分页查找的要求, 在控制器路径映射设计中要传递页码信息, 为简单起见, 不妨将页的大小固定为 8。出于业务需要, 控制器的 REST 路径参数中还传递资源类别的信息。控制器的处理方法中会将调用业务逻辑对象获取的资源信息以及分页信息写入到类型为 ModelMap 的模型中, 返

回的 ModelAndView 对象中包装有模型和选择的显示视图。
以下为具体代码：

```
@Controller
public class ResourceAccess {
    @RequestMapping ( value = "/resource/class/{classID}/
page/{pageNo}", method = RequestMethod.GET)
    public ModelAndView pagelist(
        @PathVariable("classID") int classID, // 资源类别
        @PathVariable("pageNo") int pageNo, // 页码
        HttpServletRequest request)
    {
        ApplicationContext applicationContext=
            RequestContextUtils.getWebApplicationContext ( requ-
            est ); // 应用环境
        resourceService rs=( resourceService )
            applicationContext.getBean ( "resourceBean" ); // 获
            取业务逻辑 Bean
        int size=rs.getAmount ( classID ); // 访问业务逻辑获取
            该类资源的总数量
        List<MyResource> rlist=rs.list ( classID,pageNo ); //获取
            某页的资源目录列表
        ModelMap modelMap= new ModelMap ( );
        modelMap.put ( "classID", classID ); //将相关信息保存
            到模型中
        modelMap.put ( "totalpage", ( int ) ( Math.ceil ( size / 8.0 )
            ) ); // 计算总页数
        modelMap.put ( "page", pageNo );
        modelMap.put ( "resources", rlist );
        return new ModelAndView ( "/listres", modelMap ); } }
```

4 视图设计

采用 JSP 作为视图文件(文件名为 listres.jsp) 程序中
使用 JSTL 的“forEach” 标签控制循环流程, 用 EL 表达式获
取模型数据。在整个显示页面的底部显示分页操作超链等。
分页超链显示时使用了 JSTL 的“if” 标签进行判定, 当前页
为第 1 页时无“上一页”, 当前页为最后页时无“下一页”。

```
<%@page contentType="text/html; charset=UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" pre-
fix="c" %>
<html><body>
<table>
<c:forEach items="${resources}" var="res">
<tr><td height="28" width="90%">
&nbsp;&nbsp;&nbsp;<a href="${pageContext.request.
contextPath}/resource/detail/
${res.resourceID}"> ${res.titleName}</a>
</td> </tr>
</c:forEach>
```

```
</table>
<p align="center">目前是第 ${page} 页 &nbsp;&nbsp;&nbsp;<a href="
${pageContext.request.contextPath}/resource/class/${classID}/
page/1">第 1 页
</a>&nbsp;&nbsp;&nbsp;
<c:if test="${page>1}">
<a href=
"${pageContext.request.contextPath}/resource/class/${c-
lassID}/page/${page-1}">
    上一页</a>&nbsp;&nbsp;&nbsp;
</c:if>
<c:if test="${page<totalpage}">
<a href=
"${pageContext.request.contextPath}/resource/class/${
classID}/page/${page+1}"> 下一页</a>&nbsp;&nbsp;&nbsp;
</c:if>
&nbsp;&nbsp;&nbsp;<a href=
"${pageContext.request.contextPath}/resource/class/${class-
sID}/as/page/${totalpage}">最后页</a>&nbsp;&nbsp;&nbsp;共有 ${total-
page} 页</p>
</body></html>
```

要访问某页资源列表可发送如下形式的 URL 请求, 访
问结果如图 2 所示。

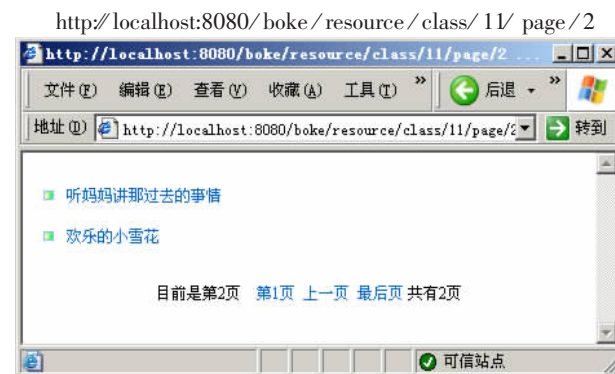


图 2 分页查询

5 结束语

本文结合资源查询的应用设计, 介绍了用 Spring MVC
进行数据分页显示处理的具体设计过程。包括 应用配置、业
务逻辑的设计、控制器的设计、视图的设计等。本文假定每页
的大小固定为 8 条数据记录, 如果还要页的大小可变, 则
URL 参数还需要传递页的大小信息。

参考文献：

- [1] 张俐. 基于 MVC 模式的分页组件应用[J]. 计算机工程, 2011, 37 (21): 255-257.
- [2] 丁振凡, 吴根斌. Spring 3.x MVC 模型的数据校验国际化处理 [J]. 计算机时代, 2012, 8: 26-27
- [3] 丁振凡, 李馨梅. 基于 JdbcTemplate 的数据库访问处理[J]. 智能计算机与应用, 2012, 2(3): 29-32.
- [4] 李辉, 王瑞波. 多条件分页查询优化的设计方法[J]. 计算机工程, 2010, 36(2): 51-55.