《Java EE核心框架实战》

图书基本信息

书名：《Java EE核心框架实战》

13位ISBN编号：9787115365717

出版时间：2014-9

作者：高洪岩

页数：614

《Java EE核心框架实战》

书籍目录

《Java EE核心框架实战》

精彩短评

1、.....spring mybatis搭框架那会看的
2、第一次收到同事送的书。慢慢学吧。
3、烂的一比。一个MyBatis，你告诉我怎么用就可以了，尼玛要说，怎么连接MySql，怎么连接Oracle，怎么连接Sqlserver，有必要吗？有必要吗？完全凑字数。坑钱货

《Java EE核心框架实战》

章节试读

1、《Java EE核心框架实战》的笔记-第3页

```
&lt;?xml version="1.0" encoding="UTF-8" ?&gt;&lt;!DOCTYPE configuration PUBLIC
"-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd"&gt;
&lt;configuration&gt;
  &lt;environments default="development"&gt;
    &lt;environment id="development"&gt;
      &lt;transactionManager type="JDBC"&gt;&lt;/transactionManager&gt;
      &lt;dataSource type="POOLED"&gt;
        &lt;property name="driver"
          value="com.microsoft.sqlserver.jdbc.SQLServerDriver" /&gt;
        &lt;property name="url"
          value="jdbc:sqlserver://localhost:1433;databaseName=j2eeTest" /&gt;
        &lt;property name="username" value="sa" /&gt;
        &lt;property name="password" value="Passw0rd" /&gt;
      &lt;/dataSource&gt;
    &lt;/environment&gt;
  &lt;/environments&gt;
  &lt;mappers&gt;
    &lt;mapper resource="com/byhard/mybatistest/orm/userinfoMapper.xml"&gt;&lt;/mapper&gt;
  &lt;/mappers&gt;
&lt;/configuration&gt;
```

2、《Java EE核心框架实战》的笔记-第58页

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
&lt;!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 2.1//EN"
"http://struts.apache.org/dtds/struts-2.1.dtd"&gt;
&lt;struts&gt;
  &lt;constant name="struts.devMode" value="true"&gt;&lt;/constant&gt;
  &lt;package name="com.byhard.Struts2Login" extends="struts-default"&gt;
    &lt;action name="login" class="com.byhard.Struts2Login.controller.Login"&gt;
      &lt;result name="toOKJSP"&gt;/ok.jsp&lt;/result&gt;
      &lt;result name="toNOJSP"&gt;/no.jsp&lt;/result&gt;
    &lt;/action&gt;
  &lt;/package&gt;
&lt;/struts&gt;
```

3、《Java EE核心框架实战》的笔记-第210页

```
    &lt;filter&gt;
    &lt;filter-name&gt;encodingFilter&lt;/filter-name&gt;
    &lt;filter-class&gt;
      org.springframework.web.filter.CharacterEncodingFilter
    &lt;/filter-class&gt;
    &lt;init-param&gt;
      &lt;param-name&gt;encoding&lt;/param-name&gt;
```

```
        &lt;param-value&gt;UTF-8&lt;/param-value&gt;
      &lt;/init-param&gt;
      &lt;init-param&gt;
        &lt;param-name&gt;forceEncoding&lt;/param-name&gt;
        &lt;param-value&gt;true&lt;/param-value&gt;
      &lt;/init-param&gt;
  &lt;/filter&gt;
  &lt;filter-mapping&gt;
    &lt;filter-name&gt;encodingFilter&lt;/filter-name&gt;
    &lt;url-pattern&gt;/*&lt;/url-pattern&gt;
  &lt;/filter-mapping&gt;
```

4、《Java EE核心框架实战》的笔记-第196页

Web.xml配置：

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
&lt;web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  version="2.5"&gt;

  &lt;servlet&gt;
    &lt;servlet-name&gt;springMVC&lt;/servlet-name&gt;
    &lt;servlet-class&gt;org.springframework.web.servlet.DispatcherServlet&lt;/servlet-class&gt;
    &lt;init-param&gt;
      &lt;param-name&gt;contextConfigLocation&lt;/param-name&gt;
      &lt;param-value&gt;classpath:spring-servlet.xml&lt;/param-value&gt;
    &lt;/init-param&gt;
    &lt;load-on-startup&gt;1&lt;/load-on-startup&gt;
  &lt;/servlet&gt;

  &lt;servlet-mapping&gt;
    &lt;servlet-name&gt;springMVC&lt;/servlet-name&gt;
    &lt;url-pattern&gt;*.spring&lt;/url-pattern&gt;
  &lt;/servlet-mapping&gt;

  &lt;welcome-file-list&gt;
    &lt;welcome-file&gt;index.jsp&lt;/welcome-file&gt;
  &lt;/welcome-file-list&gt;
&lt;/web-app&gt;
```

spring-servlet.xml配置：

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
&lt;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"
```

xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd
http://www.springframework.org/schema/cache
http://www.springframework.org/schema/cache/spring-cache-4.0.xsd">

<context:component-scan
base-package="com.byhard.springlogin.controller"></context:component-scan>
</beans>

5、《Java EE核心框架实战》的笔记-第215页

415错误需要参考文章：

http://blog.csdn.net/yixiaoping/article/details/45281721

重要配置：

spring 4.x要配 jackson2.x

jackson-annotations-2.6.2.jar
jackson-core-2.6.2.jar
jackson-databind-2.6.2.jar

spring-servlet.xml配置：

```
&lt;bean class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter"&gt;
  &lt;property name="messageConverters"&gt;
    &lt;list&gt;
      &lt;ref bean="jsonHttpMessageConverter" /&gt;
    &lt;/list&gt;
  &lt;/property&gt;
&lt;/bean&gt;

&lt;bean id="jsonHttpMessageConverter"
class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter"&gt;
  &lt;property name="supportedMediaTypes"&gt;
    &lt;list&gt;
      &lt;value&gt;application/json;charset=UTF-8&lt;/value&gt;
    &lt;/list&gt;
  &lt;/property&gt;
&lt;/bean&gt;
```

mvc配置：

```
    &lt;mvc:annotation-driven /&gt;
```

jsp中ajax请求：

```
    function userinfo(username, password){
        this.username = username;
        this.password = password;
    }

   function sendAjax2(){
      var userinfoRef = new userinfo('高洪岩new123', '123new');
      var jsonStringRef = JSON.stringify(userinfoRef);
      $.ajax({
         type: 'POST',
         data: jsonStringRef,
         url: 'createJSONObjectURL.spring?t=' + new Date().getTime(),
         dataType:'json',
         contentType: 'application/json;charset=UTF-8'
      });
   }
```

controller测试文件：

```
    @RequestMapping(value = "/createJSONObjectURL", method = RequestMethod.POST, consumes =
"application/json")
    public String createJSON2(@RequestBody Userinfo userinfo) {
        java.util.Date date = new java.util.Date();
        System.out.println(date.toGMTString() + ":username value=" + userinfo.getUsername());
        System.out.println(date.toGMTString() + ":password value=" + userinfo.getPassword());
        return "index.jsp";
    }
```

6、《Java EE核心框架实战》的笔记-第12页

```
       &lt;?xml version="1.0" encoding="UTF-8" ?&gt;&lt;!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD
Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd"&gt;
&lt;mapper namespace="com.byhard.mybatistest.orm"&gt;
    &lt;resultMap type="com.byhard.mybatistest.orm.Userinfo" id="userinfoResultMap"&gt;
        &lt;id property="id" column="id" /&gt;
        &lt;result property="username" column="username" /&gt;
        &lt;result property="password" column="password" /&gt;
    &lt;/resultMap&gt;

    &lt;insert id="insertUserinfo" parameterType="com.byhard.mybatistest.orm.Userinfo"&gt;
        insert into
        userinfo(username, password)
        values(#{username},#{password})
```

```
    &lt;/insert&gt;

    &lt;!-- 查询用户，根据id --&gt;
    &lt;select id="getUser" parameterType="int" resultType="com.byhard.mybatistest.orm.Userinfo"
        resultMap="userinfoResultMap"&gt;
        SELECT * from userinfo ui WHERE ui.id = #{id}
    &lt;/select&gt;

    &lt;!-- 查询用户列表 --&gt;
    &lt;select id="getUserAll" resultType="com.byhard.mybatistest.orm.Userinfo"
        resultMap="userinfoResultMap"&gt;
        SELECT * from userinfo
    &lt;/select&gt;

    &lt;!-- 更新用户密码 --&gt;
    &lt;update id="updateUserPassword" parameterType="com.byhard.mybatistest.orm.Userinfo"&gt;
        update userinfo
        set password = #{password}
        where id = #{id};
    &lt;/update&gt;

    &lt;!-- 根据用户id删除用户记录 --&gt;
    &lt;update id="deleteUserById" parameterType="int"&gt;
        delete from userinfo where id = #{id};
    &lt;/update&gt;

    &lt;update id="findPwd" statementType="CALLABLE" parameterType="hashmap"&gt;
      &lt;![CDATA[
        { call P_J2EE_GetValue(#{username, mode=IN, jdbcType=NVARCHAR}, #{pwd, mode=OUT,
jdbcType=NVARCHAR}) }
      ]]&gt;
    &lt;/update&gt;

&lt;/mapper&gt;
```

7、《Java EE核心框架实战》的笔记-第213页

方式1：

```
    @RequestMapping(value="/getJSONString")
    public String getJSONString(@RequestParam("jsonString") String jsonString){
        ObjectMapper om = new ObjectMapper();
        try {
            @SuppressWarnings("rawtypes")
            Map map = om.readValue(jsonString, Map.class);
            System.out.println(map.get("username"));
            System.out.println(map.get("password"));
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
```

```
        }
        return "index.jsp";
    }
```

方式2：

8、《Java EE核心框架实战》的笔记-第55页

```
    &lt;?xml version="1.0" encoding="UTF-8"?&gt;
&lt;web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    version="2.5"&gt;

    &lt;filter&gt;
        &lt;filter-name&gt;Struts2&lt;/filter-name&gt;
        &lt;filter-class&gt;org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter&lt;/filter-class&gt;
    &lt;/filter&gt;
    &lt;filter-mapping&gt;
        &lt;filter-name&gt;Struts2&lt;/filter-name&gt;
        &lt;url-pattern&gt;*.jsp&lt;/url-pattern&gt;
    &lt;/filter-mapping&gt;
    &lt;filter-mapping&gt;
        &lt;filter-name&gt;Struts2&lt;/filter-name&gt;
        &lt;url-pattern&gt;*.js&lt;/url-pattern&gt;
    &lt;/filter-mapping&gt;
    &lt;filter-mapping&gt;
        &lt;filter-name&gt;Struts2&lt;/filter-name&gt;
        &lt;url-pattern&gt;*.action&lt;/url-pattern&gt;
    &lt;/filter-mapping&gt;
    &lt;welcome-file-list&gt;
        &lt;welcome-file&gt;index.jsp&lt;/welcome-file&gt;
    &lt;/welcome-file-list&gt;
&lt;/web-app&gt;
```

9、《Java EE核心框架实战》的笔记-第197页

Url中同名的参数将要自动传递给控制层方法中同名的参数，并且不再需要@RequestParam("username")注解。

指定请求路径与请求方法：

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

控制层重定向到控制层，带参数：

```
    @RequestMapping(value = "/login", method = RequestMethod.POST)
    public String login(@RequestParam("username") String username,
```

```
        @RequestParam("password") String password, Model model){
    if(username.equals("byhard") &amp;amp;&amp;amp; password.equals("123")) {
        model.addAttribute("username", username);
        //return "ok.jsp";
        return "redirect:/listUsername.spring";
    } else {
        //return "no.jsp";
        return "redirect:/listUsername2.spring?username=" + username;
    }
}
```

匹配url路径，执行指定的controller

```
@RequestMapping("/test1/{userId}")
public String test1(@PathVariable String userId){
    System.out.println("run test1 userId=" + userId);
    return "/index.jsp";
}

@RequestMapping("/test2/{userId}")
public String test2(@PathVariable("userId") String userIdParam){
    System.out.println("run test2 userId=" + userIdParam);
    return "/index.jsp";
}
```

《Java EE核心框架实战》

版权说明