05_SS整合

笔记本: Spring&SpringMVC
创建时间: 2019/9/8 9:05
作者: 韩延兵

1.创建一个Maven版的Web工程，通过依赖导入以下jar包

```
    <dependencies>
        <!-- 只需要将context的jar包的依赖配置上，其他依赖的jar包会自动导入 -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>4.0.0.RELEASE</version>
        </dependency>
        <!-- 只需要导入spring-webmvc的jar包，Maven会自动将它依赖的spring-web的jar导入 -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>4.0.0.RELEASE</version>
        </dependency>
    </dependencies>
```

2.web.xml中的配置

- 1）配置过滤器CharacterEncodingFilter，目的是为了解决请求和响应的乱码问题
- 2）配置DispatcherServlet，快捷键是Alt+/ 选择倒数第二项
- 3）配置ContextLoaderListener，快捷键是Alt+/ 选择倒数第三项

```
<?xml version="1.0" encoding="UTF-8"?>
<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/javaeehttp://java.sun.com/xml/ns/jav
app_2_5.xsd"
    id="WebApp_ID" version="2.5">
    <filter>
        <filter-name>CharacterEncodingFilter</filter-name>
        <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
        <init-param>
            <param-name>encoding</param-name>
            <param-value>UTF-8</param-value>
        </init-param>
        <init-param>
            <param-name>forceEncoding</param-name>
            <param-value>true</param-value>
        </init-param>
    </filter>
    <filter-mapping>
        <filter-name>CharacterEncodingFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>

    <servlet>
        <servlet-name>springDispatcherServlet</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>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>springDispatcherServlet</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

    <!-- 配置ContextLoaderListener，快捷键：Alt + / 选择倒数第三项 -->
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath:beans.xml</param-value>
    </context-param>
    <listener>
        <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>

</web-app>
```

3.SpringMVC的配置文件springmvc.xml中的配置

```
<?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"
    xsi:schemaLocation="http://www.springframework.org/schema/beanshttp://www.springfra
beans.xsd
http://www.springframework.org/schema/contexthttp://www.springframework.org/schema/cont
context-4.0.xsd">
    <!-- 设置自动扫描的包 -->
    <context:component-scan base-package="com.atguigu.ss" use-default-
filters="false">
        <!-- 配置自扫描Controller -->
        <context:include-filter type="annotation"
expression="org.springframework.stereotype.Controller"/>
    </context:component-scan>

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

</beans>
```

4.Spring的配置文件beans.xml中的配置

```
<?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"
    xsi:schemaLocation="http://www.springframework.org/schema/beanshttp://www.springfra
beans.xsd
```

```
http://www.springframework.org/schema/contexthttp://www.springframework.org/schema/cont
context-4.0.xsd">
    <!-- 配置自动扫描的包 -->
    <context:component-scan base-package="com.atguigu.ss">
        <!-- 配置不扫描Controller -->
        <context:exclude-filter type="annotation"
expression="org.springframework.stereotype.Controller"/>
    </context:component-scan>
</beans>
```

5.创建业务逻辑层的EmployeeService的接口和实现

- EmployeeServiceImpl

```
@Service("employeeService")
public class EmployeeServiceImpl implements EmployeeService {


    public EmployeeServiceImpl() {
        System.out.println("EmployeeServiceImpl对象被创建");
    }

    @Override
    public Employee getEmployeeById(Integer id) {
        // 根据员工的id获取该员工的信息
        Employee employee = new Employee(id, "韩总", "hybing@atguigu.com",
null);
        return employee;
    }
}
```

6.创建表现层控制器EmployeeController

- EmployeeController

```
@Controller
public class EmployeeController {
    public EmployeeController() {
        System.out.println("EmployeeController对象被创建");
    }

    @Autowired
    private EmployeeService employeeService;

    /*
     * Spring与SpringMVC整合：
     *  不整合
     *      1) 将所有的配置都配置到SpringMVC的配置文件中
     *      2) 使用import标签将其他配置文件导入到SpringMVC的配置文件中
     *  整合
     *      Spring负责管理Service、Dao、数据源以及与其他第三方框架的整合
     *      SpringMVC负责管理Controller、视图解析器、处理静态资源等
     *  出现的问题：
     *      1.IOC容器如何初始化？
     *          Java工程：new ClassPathXmlApplicationContext("beans.xml");
     *          Web工程：在web.xml配置文件中配置一个监听器ContextLoaderListener
     *      2.Controller和Service被创建了两次
     *          让Spring不扫描Controller
     *          让SpringMVC只扫描Controller
     *
     */
    @RequestMapping("/getEmployee")
    public String getEmployee(@RequestParam("id") Integer id,Map<String ,
Object> map) {
        //调用EmployeeService获取员工信息
        Employee employee = employeeService.getEmployeeById(id);
        //将员工信息放到map中
        map.put("emp", employee);
        return "list";
    }
}
```

7.创建首页index.jsp页面并再WEB-INF目录下创建pages目录及list.jsp页面

- index.jsp页面

```
<a href="getEmployee?id=1">获取一个员工</a>
```