

Ctrl+shift+T快速查找需要的类

一，演示问题

1, action调用service, service调用dao

①相关jar包



freemarker-2.3.19.jar
909.34KB



javassist-3.11.0.GA.jar
599.8KB



log4j-api-2.11.2.jar
260.04KB



log4j-core-2.11.2.jar
1.55MB



mysql-connector-java-5.1.40.jar
2.18MB



ognl-3.0.6.jar
222.65KB



spring-aop-5.0.0.RELEASE.jar
352.16KB



spring-aspects-5.0.0.RELEASE.jar
40.16KB



spring-beans-5.0.0...E.jar
639.36KB



spring-context-5.0....E.jar
1.02MB



spring-core-5.0.0.R...E.jar
1.15MB



spring-expression-...E.jar
256.53KB



struts2-core-2.3.15.3.jar
783.92KB



xwork-core-2.3.15.3.jar
625.63KB



aopalliance.jar
4.36KB



asm-3.3.jar
42.55KB



asm-commons-3.3.jar
37.37KB



asm-tree-3.3.jar
20.99KB



aspectj-1.9.4.jar
16.51MB



aspectjweaver.jar
1.94MB



com.springsource.... 0.jar
4.5KB



com.springsource.... E.jar
1.56MB



commons-dbc-1.4.jar
156.75KB



commons-dbutils-1.4.jar
48.41KB



commons-fileuploa...3.jar
67.01KB



commons-io-2.0.1.jar
155.77KB



commons-lang3-3.1.jar
308.4KB



commons-logging-...3.jar
60.59KB



commons-logging-1.2.jar
60.37KB



commons-pool-1.6.jar
108.51KB

2, 遇见的问题: 每一次访问action的时候, 都会加载spring配置文件

```
六月 11, 2019 1:15:30 下午 org.springframework.context.support.ClassP
信息: Refreshing org.springframework.context.support.ClassPathXmlAp
六月 11, 2019 1:15:30 下午 org.springframework.beans.factory.xml.XmlB
信息: Loading XML bean definitions from class path resource [applic
service=====
dao
action=====
六月 11, 2019 1:15:40 下午 org.springframework.context.support.ClassP
信息: Refreshing org.springframework.context.support.ClassPathXmlAp
六月 11, 2019 1:15:40 下午 org.springframework.beans.factory.xml.XmlB
信息: Loading XML bean definitions from class path resource [applic
service=====
dao
action=====
六月 11, 2019 1:15:43 下午 org.springframework.context.support.ClassP
信息: Refreshing org.springframework.context.support.ClassPathXmlAp
六月 11, 2019 1:15:43 下午 org.springframework.beans.factory.xml.XmlB
信息: Loading XML bean definitions from class path resource [applic
service=====
dao
```

3, 解决方案

- (1) 在服务器启动的时候, 创建对象加载加载配置文件
- (2) 底层使用监听器、servletContext对象

4, 在spring里面不需要我们自己写代码实现, spring已经帮我们封装好了。

(1) 封装一个监听器，只需要配置监听器就可以了。

在web.xml文件中，

```
<!-- 配置监听器 -->
<listener>
<listener-class>
org.springframework.web.context.ContextLoaderListener
</listener-class>
</listener>
```

(2) 在配置监听器之前，需要先导入spring整合web项目的jar包。



spring-web-5.0.0.R...E.jar
1.15MB

(3) 指定加载spring配置文件的位置

```
<!-- 指定spring配置文件的位置 -->
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>classpath:applicationContext.xml</param-value>
</context-param>
```

二，具体的代码

1, userAction实体类

```
package cn.java.action;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.opensymphony.xwork2.ActionSupport;
import cn.java.service.UserService;

public class UserAction extends ActionSupport {
    @Override
    public String execute() throws Exception {
        System.out.println("action=====");
        // TODO Auto-generated method stub
        ApplicationContext context = new
ClassPathXmlApplicationContext("applicationContext.xml");
        UserService userService = (UserService)context.getBean("userService");
        userService.add();
        return NONE;
    }
}
```

2, userService实体类

```
package cn.java.service;
import javax.annotation.Resource;
import cn.java.dao.UserDao;
public class UserService {
```

```

private UserDao userDao;

public UserDao getUserDao() {
    return userDao;
}

public void setUserDao(UserDao userDao) {
    this.userDao = userDao;
}

public void add() {
    System.out.println("service=====");
    userDao.add();
}
}

```

3, userDao实体类

```

package cn.java.dao;
public class UserDao {
public void add(){
    System.out.println("dao");
}
}

```

4, applicationContext.xml配置文件

```

<?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"
    xmlns:jdbc="http://www.springframework.org/schema/jdbc"
    xmlns:jee="http://www.springframework.org/schema/jee"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:util="http://www.springframework.org/schema/util"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-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/jdbc
        http://www.springframework.org/schema/jdbc/spring-jdbc-4.0.xsd
        http://www.springframework.org/schema/jee
        http://www.springframework.org/schema/jee/spring-jee-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/util
        http://www.springframework.org/schema/util/spring-util-4.0.xsd
        http://www.springframework.org/schema/task
        http://www.springframework.org/schema/task/spring-task-4.0.xsd"
    default-lazy-init="false">

```

```

    <bean id="userService" class="cn.java.service.UserService">
    <!-- property中的name属性写的是实体userdao在userservice中的属性名字，ref写
    的是配置的userdao的id值 -->
        <property name="userDao" ref="userDao"> </property>
    </bean>
    <bean id="userDao" class="cn.java.dao.UserDao"> </bean>

```

```

</beans>

```

5, web.xml配置文件

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0"
    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_3_0.xsd">
    <!-- 配置Struts的核心控制器，就是一个过滤器 -->
    <filter>
        <filter-name>struts2</filter-name>
        <filter-
class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-
class>
        </filter>
        <filter-mapping>
            <filter-name>struts2</filter-name>
            <url-pattern>/*</url-pattern>
        </filter-mapping>
    <!-- 配置监听器 -->
    <listener>
    <listener-class>
    org.springframework.web.context.ContextLoaderListener
    </listener-class>
    </listener>
    <!-- 指定spring配置文件的位置 -->
    <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:applicationContext.xml</param-value>
    </context-param>

    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>default.html</welcome-file>
    </welcome-file-list>
</web-app>

```

6, struts.xml配置文件

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
    "http://struts.apache.org/dtds/struts-2.3.dtd">
<struts>
    <constant name="struts.devMode" value="true" />
    <package name="p1" extends="struts-default">
        <action name="userAction" class="cn.java.action.UserAction" >
            </action>
        </package>
    </struts>
```