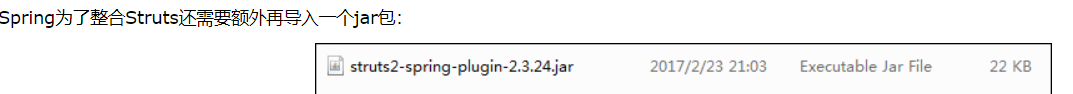
c3p0就是一个开放源代码的JDBC数据库连接池， 并且管理这些连接，不用每次访问数据库都重新发出数据库连接请求，只需要在c3p0的连接池中拿一个过来就可以了，这样可以大大的提高性能

ref代表的是引用这个对象



**过程**

**Dao和service都有接口和实现类（接口引用实现类）**

**Jsp请求—Action（包含service对象）--service对象完成数据库操作--service实现类调用dao方--dao（调用session完成数据库具体操作）**

**spring.xml(applicationContext.xml)**

1. **引入约束 xmlns~~~**
2. **扫描包下注解（action、dao、service）**
3. **加载jdbc配置文件db.properties**
4. **建立c3p0数据源**
5. **配置SessionFactory（省略hibernate的配置文件\*.cfg.xml）注入数据源，扫描实体及映射文件**
6. **配置注解事务管理器并加入注解事务管理器**

**Struts.xml**

<!-- 声明对象是由 Spring 创建的 -->

<constant name="struts.objectFactory" value="spring"></constant>

**Web.xml**

<context-param>

<param-name>contextConfigLocation</param-name>

定义要装入的applicationContext 配置文件位置

<param-value>classpath:applicationContext.xml</param-value>

</context-param>

<listener>

ContextLoaderListener监听器的作用是

启动Web容器时，自动装配applicationContext的配置信息

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

<filter-name>struts2</filter-name>

<filter-class>指明过滤器类所在的包路径

<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>

<url-pattern>匹配规则既不是简单的通配，也不是正则表达式，而是特定的规则

当servlet容器接收到浏览器发起的一个url请求后，容器会用url减去当前应用的上下文路径，以剩余的字符串作为servlet映射

假如url是<http://localhost:8080/appDemo/index.html>

其应用上下文是appDemo，

容器会将http://localhost:8080/appDemo去掉

用剩下的/index.html部分拿来做servlet的映射匹配

<url-pattern>\*.jsp</url-pattern>

<url-pattern>\*.action</url-pattern>

则任何扩展名为jsp或action的url请求都会匹配

"/\*"可以匹配所有url （路径型的和后缀型的url）

"/"能拦截“.js”，“.css”，".png"等静态资源的访问