

自定义拦截器：

步骤：

1、编写一个类，选择实现Interceptor接口

选择继承AbstractInterceptor抽象类

选择继承MethodFilterInterceptor抽象类 推荐的方式

2、提供抽象类中的方法实现

3、配置拦截器：

声明拦截器

```
<interceptors>
```

```
    <interceptor name="checkLogin"  
class="com.itheima.web.interceptors.CheckLoginInterceptor"></interceptor>
```

```
</interceptors>
```

引用拦截器

```
<interceptor-ref name="checkLogin"></interceptor-ref>
```

拦截器的放行：

```
invocation.invoke();
```

拦截器和过滤器一样：

访问资源之前会拦截，访问资源之后也会拦截

它和过滤器的区别：

过滤器是javaweb中Servlet规范的

拦截器是struts2独有的。

过滤器可以拦截任何资源：比如访问jsp,css,html,image等等都可以拦截

拦截器只能拦截访问的动作方法。

注意事项： 并且非常重要！！！！！！！！

当我们的action配置时，在内部不写任何拦截器时，默认由一个叫做defaultStack的拦截器栈为我们工作。

但是当我们引用了任何一个拦截器，默认的拦截器栈就失效了。

## 在struts2主配置文件中配置拦截器

```
<interceptors>

  <interceptor name="checkLogin"
class="com.itheima.web.interceptors.CheckLoginInterceptor"></interceptor>

  <!-- 定义一个拦截器栈 -->

  <interceptor-stack name="myDefaultStack">

    <interceptor-ref name="checkLogin">

      <!-- 传入参数：告知拦截器对登录方法不拦截 -->

      <param name="excludeMethods">login</param>

    </interceptor-ref>

    <interceptor-ref name="defaultStack"></interceptor-ref>

  </interceptor-stack>

</interceptors>

<!-- 把myDefaultStack声明为默认拦截器栈 -->

<default-interceptor-ref name="myDefaultStack"></default-interceptor-ref>
```

## 在动作配置中引用

```
<interceptor-ref name="checkLogin"></interceptor-ref>
```

过滤器是javaweb中Servlet规范的在Web.xml中配置