·Struts2大多数核心功能是通过拦截器实现的，每个拦截器完成某项功能

数据转移、类型转换、数据校验

·拦截器方法在Action执行之前或者之后执行

·拦截器栈

·从结构上，拦截器栈相当于多个拦截器的组合。

·从功能上看，拦截器栈也是拦截器。

·拦截器的执行过程是一个递归的过程。

·方式一：实现Interceptor接口

·void init()：初始化拦截器所需资源

·void destroy()：释放在init()中分配的资源

·String intercept(ActionInvocation ai) throw Exception

·实现拦截器功能

·利用ActionInvocation参数获取Action状态

·返回result字符串作为逻辑视图

·方式二：继承AbstractInterceptor类

·提供了init()和destroy()方法的空实现

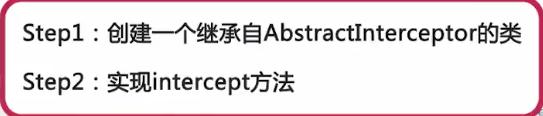
·只需要实现intercept方法即可

·计算Action执行时间

·实现步骤

·创建拦截器

·在配置文件中定义拦截器并引用它

·

Step3：注册拦截器

Step4：在action中引用拦截器

·Struts2内建拦截器

·params拦截器

·负责将请求参数设置为Action属性

·staticParams拦截器

·将配置文件中action元素的子元素param参数设置为Action属性

·servletConfig拦截器

·将源于Servlet API的各种对象注入到Action，必须实现对应接口

·fileUpload拦截器（内部使用了CommonsFileUpload组件）

·对文件上传提供支持，将文件和元数据设置到对应的Action属性

·exception拦截器

·捕获异常，并且将异常映射到用户自定义的错误页面

·默认拦截器栈

·在·struts-default.xml中定义一个defaultStack拦截器栈，并将其指定为默认拦截器。

·只要在定义包的过程中继承struts-default包，那么热defaultStack将是默认的拦截器。

·当为包中的某个action显式指定了某个拦截器，则默认拦截器不会起作用。

·拦截器栈中的各个拦截器的顺序很重要。