

Struts2 面试题分析

1. 简述 Struts2 的工作流程:

- ①. 请求发送给 StrutsPrepareAndExecuteFilter
- ②. StrutsPrepareAndExecuteFilter 判定该请求是否是一个 Struts2 请求
- ③. 若该请求是一个 Struts2 请求, 则 StrutsPrepareAndExecuteFilter 把请求的处理交给 ActionProxy
- ④. ActionProxy 创建一个 ActionInvocation 的实例, 并进行初始化
- ⑤. ActionInvocation 实例在调用 Action 的过程前后, 涉及到相关拦截器 (Interceptor) 的调用。
- ⑥. Action 执行完毕, ActionInvocation 负责根据 struts.xml 中的配置找到对应的返回结果。调用结果的 execute 方法, 渲染结果。
- ⑦. 执行各个拦截器 invocation.invoke() 之后的代码
- ⑧. 把结果发送到客户端

2. Struts2 拦截器 和 过滤器 的区别:

- ①、过滤器依赖于 Servlet 容器, 而拦截器不依赖于 Servlet 容器。
- ②、Struts2 拦截器只能对 Action 请求起作用, 而过滤器则可以对几乎所有请求起作用。
- ③、拦截器可以访问 Action 上下文(ActionContext)、值栈里的对象(ValueStack), 而过滤器不能。
- ④、在 Action 的生命周期中, 拦截器可以多次调用, 而过滤器只能在容器初始化时被调用一次。

3. 为什么要使用 Struts2 & Struts2 的优点:

- ①. 基于 MVC 架构, 框架结构清晰。
- ②. 使用 OGNL: OGNL 可以快捷的访问值栈中的数据、调用值栈中对象的方法
- ③. 拦截器: Struts2 的拦截器是一个 Action 级别的 AOP, Struts2 中的许多特性都是通过拦截器来实现的, 例如异常处理, 文件上传, 验证等。拦截器是可配置与重用的
- ④. 多种表现层技术. 如: JSP、FreeMarker、Velocity 等

4. Struts2 如何访问 HttpServletRequest、HttpSession、ServletContext 三个域对象 ?

- ①. 与 Servlet API 解耦的访问方式

- > 通过 `ActionContext` 访问域对象对应的 `Map` 对象
- > 通过实现 `Aware` 接口使 `Struts2` 注入对应的 `Map` 对象

②. 与 `Servlet API` 耦合的访问方式

- > 通过 `ServletActionContext` 直接获取 `Servlet API` 对象
- > 通过实现 `ServletXxxAware` 接口的方式使 `Struts2` 注入对应的对象

5. `Struts2` 中的默认包 `struts-default` 有什么作用？

①. `struts-default` 包是 `struts2` 内置的，它定义了 `struts2` 内部的众多拦截器和 `Result` 类型，而 `Struts2` 很多核心的功能都是通过这些内置的拦截器实现，如：从请求中把请求参数封装到 `action`、文件上传和数据验证等等都是通过拦截器实现的。当包继承了 `struts-default` 包才能使用 `struts2` 为我们提供的这些功能。

②. `.struts-default` 包是在 `struts-default.xml` 中定义，`struts-default.xml` 也是 `Struts2` 默认配置文件。`Struts2` 每次都会自动加载 `struts-default.xml` 文件。

③. 通常每个包都应该继承 `struts-default` 包。

6. 说出 `struts2` 中至少 5 个的默认拦截器

`exception`; `fileUpload`; `i18n`; `modelDriven`; `params`; `prepare`; `token`; `tokenSession`; `validation` 等

7. 谈谈 `ValueStack`:

①. `ValueStack` 贯穿整个 `Action` 的生命周期，保存在 `request` 域中，所以 `ValueStack` 和 `request` 的生命周期一样。当 `Struts2` 接受一个请求时，会

迅速创建 `ActionContext`, `ValueStack`, `Action`. 然后把 `Action` 存放在 `ValueStack`, 所以 `Action` 的实例变量可以被 `OGNL` 访问。请求来的时候，`Action`、`ValueStack` 的生命开始；请求结束，`Action`、`ValueStack` 的生命结束

②. 值栈是多实例的，因为 `Action` 是多例的(和 `Servlet` 不一样，`Servlet` 是单例的)，而每个 `Action` 都有一个对应的值栈，`Action` 对象默认保存在栈顶；

③. `ValueStack` 本质上就是一个 `ArrayList`(查看源代码得到)；

④. 使用 `OGNL` 访问值栈的内容时，不需要#号，而访问 `request`、`session`、`application`、`attr` 时，需要加#号；

⑤. `Struts2` 重写了 `request` 的 `getAttribute` 方法，所以可以使用 `EL` 直接访问值栈中的内容

8. `ActionContext`、`ServletContext`、`pageContext` 的区别？

- ①. ActionContext Struts2 的 API: 是当前的 Action 的上下文环境
- ②. ServletContext 和 PageContext 是 Servlet 的 API

9. Struts2 有哪几种结果类型？

参看 struts-default.xml 中的相关配置: dispatcher、chain、redirect 等.

10. 拦截器的生命周期与工作过程？

每个拦截器都是需要实现 Interceptor 接口

> init(): 在拦截器被创建后立即被调用, 它在拦截器的生命周期内只被调用一次. 可以在该方法中对相关资源进行必要的初始化;

> intercept(ActionInvocation invocation): 每拦截一个动作请求, 该方法就会被调用一次;

> destroy: 该方法将在拦截器被销毁之前被调用, 它在拦截器的生命周期内也只被调用一次;

11. 如何在 Struts2 中使用 Ajax 功能？

- ①. JSON plugin
- ②. DOJO plugin
- ③. DWR plugin
- ④. 使用 Stream 结果类型.