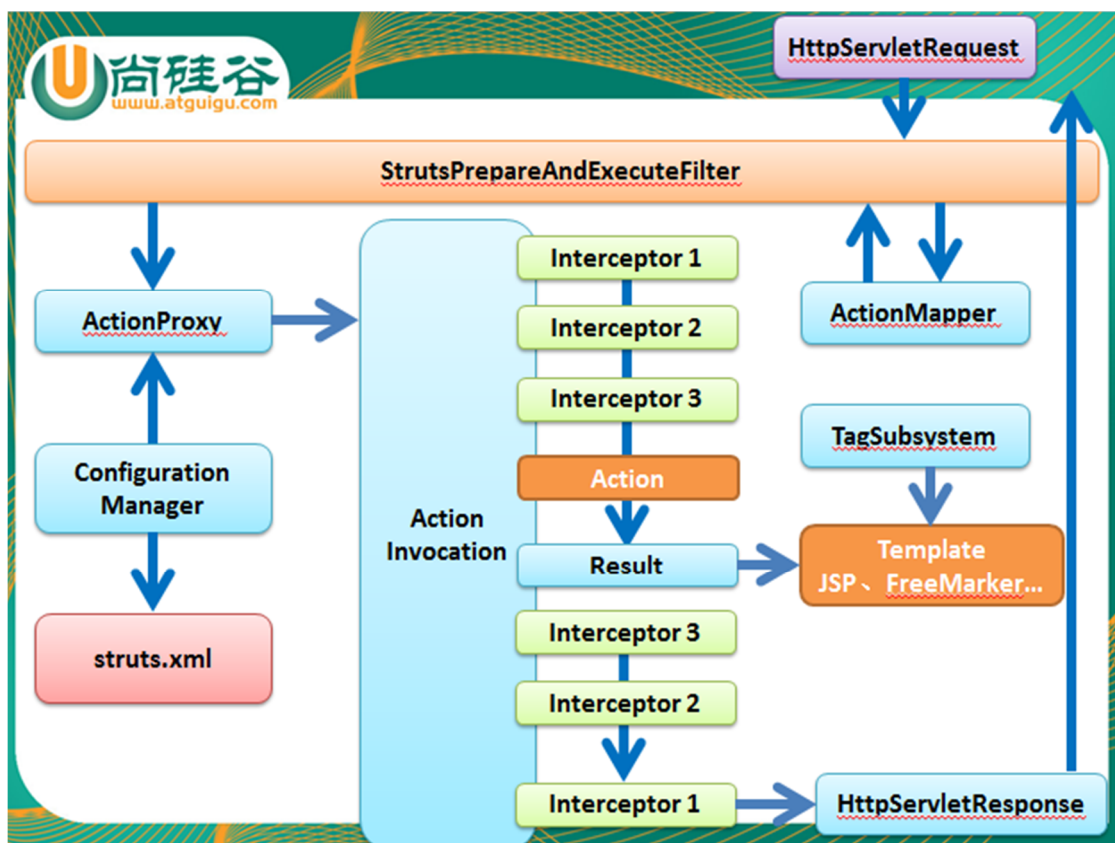


Struts2 运行流程分析

1. Struts2 运行流程如下：

- 1). 请求发送给 [StrutsPrepareAndExecuteFilter](#)
- 2). [StrutsPrepareAndExecuteFilter](#) 询问 [ActionMapper](#)：该请求是否是一个 Struts2 请求（即是否返回一个非空的 [ActionMapping](#) 对象）
- 3). 若 [ActionMapper](#) 认为该请求是一个 Struts2 请求，则 [StrutsPrepareAndExecuteFilter](#) 把请求的处理交给 [ActionProxy](#)
- 4). [ActionProxy](#) 通过 [ConfigurationManager](#) 询问框架的配置文件，确定需要调用的 Action 类及 Action 方法
- 5). [ActionProxy](#) 创建一个 [ActionInvocation](#) 的实例，并进行初始化
- 6). [ActionInvocation](#) 实例在调用 Action 的过程前后，涉及到相关拦截器（[Interceptor](#)）的调用。
- 7). 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。在渲染的过程中可以使用 Struts2 框架中的标签。
- 8). 执行各个拦截器 [invocation.invoke\(\)](#) 之后的代码
- 9). 把结果发送到客户端

2. 流程图：



3. 相关 API:

- 1). **ActionMapping**: Simple class that holds the action mapping information used to invoke a Struts action. The name and namespace are required
- 2). **ActionMapper**: When given an HttpServletRequest, the ActionMapper may return null if no action invocation request matches, or it may return an ActionMapping that describes an action invocation for the framework to try
- 3). **ActionProxy**: ActionProxy is an extra layer between XWork and the action so that different proxies are possible.
- 4). **ActionInvocation**: An ActionInvocation represents the execution state of an Action. It holds the Interceptors and the Action instance. By repeated re-entrant execution of the invoke() method, initially by the ActionProxy, then by the Interceptors, the Interceptors are all executed, and then the Action and the Result.

4. 时序图:

