

超文本传输协议(**http**)是目前互联网上极其普遍的传输协议, 它为构建功能丰富, 绚丽的网页提供了强大的支持。本次选择的项目即为对 **httpcomponents** 做简要分析。由于接近期中考试, 时间有限, 暂时只做出一些粗浅的分析。

Apache 开源组织中的 **HttpComponents**, 主要是提供对 **http** 服务器的访问功能, 目前已经是一个单独的项目, 可见 **http** 服务器的访问绝非易事。**HttpComponents** 项目就是专门设计来简化 **HTTP** 客户端与服务器进行各种通讯编程。通过它可以原来很头疼的事情现在轻松的解决, 例如你不再管是 **HTTP** 或者 **HTTPS** 的通讯方式, 告诉它你想使用 **HTTPS** 方式, 剩下的事情交给 **httpclient** 替你完成。

httpComponents 组件结构包括 **HttpComponents Core** 和 **HttpComponents Client**。**HttpComponents Core** 简称 **HttpCore**, 是一组底层 **Http** 传输协议组件, 支持两种 **I/O** 模型, 阻塞 **I/O** 模型和和非阻塞 **I/O** 模型。上层组件(**HttpComponents Client**, **HttpComponents AsyncClient**)依赖此组件实现数据传输。阻塞 **I/O** 模型基于基本的 **JAVA I/O** 实现, 非阻塞模型基于 **JAVA NIO** 实现。

而 **HttpComponents Client** 是建立在 **HttpCore** 之上的 **Http** 客户端管理组件。底层基于 **HttpCore** 阻塞 **I/O**。从 **Commons HttpClient 3.x** 继承而来, **Commons HttpClient** 原来是 **apache commons** 组建的一部分, 现在被 **HttpComponents Client** 所替代了。此外, 还有 **HttpComponents AsyncClient**, 它是建立在 **HttpCore NIO** 模型之上的 **Http** 客户端, 与基于阻塞 **I/O** 的 **HttpComponents Client** 形成互补, 由于底层使用的 **NIO** 非阻塞模型, 所以适用于高性能的应用场景。

Get、**Post** 是最常见的获取网页内容的请求形式, 当然, 返回内容并非必须是 **html** 代码, 任何的 **xml**、**json** 或文字字符串都可以作为返回内容。**HttpGet** 的实例就是一个 **get** 请求, 构造函数只有一个字符串参数, 即要获取的网页地址。另外一种构造形式是使用 **URI** 实例作为 **HttpGet** 的参数。**HttpComponents** 提供了 **URIUtils** 类, 它的 **createURI()** 返回一个 **URI** 实例, 将请求地址拆分构造不失为一种更加清晰的方式。

本次暂时分析这么多, 剩下的问题等期中考试之后再解决。