AsyncHttpClient

- what is it?
  The AsyncHttpClient (also known as AHC) is a client designed for applications that need to assemble responses from content retrieved from multiple HTTP servers. Using AHC, the distributed content can be gathered in parallel using non-blocking I/O without requiring a thread per request. Greater application throughput is possible because fewer resources are required for each application request.

- configure your application to use AHC
  To use AHC in your application, you just need to package the AHC jar file in the WEB_INF/lib directory of your application. AHC uses the Mina package to manage the underlying I/O, so the Mina jar files will also need to be included. Typically, this type of packaging is handled automatically by maven dependency management.

- sample
  Using AHC to make requests is very simple. To begin, you first create an AHC instance that will be used to make requests.

```
AsyncHttpClient client = new AsyncHttpClient();
```

Once you have the client created, the client is used to request content from different URLs:

```
HttpRequestMessage request = new HttpRequestMessage(url, callback);client.sendRequest(request);
```

That's all there is to send a request. The callback argument is an object that implements the AsyncHttpClientCallback interface. The callback handler will be notified of successful completion of the request and given the returned response. The callback will also be notified of connection closings, request timeouts, or any other request failure situation.

A single client instance can process multiple requests at one time and can be reused after requests complete. If it is anticipated that multiple requests will be made from target servers that support keep alive connections, a SessionCache can be used with the AHC client to allow live connections to be reused.

```
sessionCache = new SessionCache();
// enable connection reuse
client.setSessionCache(sessionCache);
```

* javadoc