

Jetty HTTP Client

Jetty HTTP Client

Improved Tutorial at Eclipse

Jetty 6 and Jetty 7 HTTP clients are very similar, almost identical. You may want to follow a detailed tutorial [here](#) for Jetty 7, which will be mostly valid for Jetty 6 too.

The [HttpClient](#) is an implementation of the client API. It makes it easy to conduct exchanges with a server, and to process the responses. It can use either blocking or non-blocking connectors, the default is to use non-blocking. The HttpClient can also do SSL, use proxies and authentication.

By nature it is asynchronous. To begin a request, create an [HttpExchange](#) instance (or a subclass of it). HttpExchange provides a number of callback methods that you implement in order to receive reply. You can be notified of the receipt of particular parts of the response, such as the headers or just the content. To start the request, call [HttpClient.send\(HttpExchange\)](#) method. A single HttpClient can have many exchanges outstanding.

Asynchronous Request Example

```
HttpClient client = new HttpClient();
client.setConnectorType(HttpClient.CONNECTOR
_SELECT_CHANNEL);
try
{
    client.start();
}
catch (Exception e)
{
    throw new ServletException(e);
}

// create the exchange object, which lets
you define where you want to go
// and what you want to do once you get a
response
```

```
ContentExchange exchange = new
ContentExchange()
{
    // define the callback method to process
the response when you get it back
    protected void onResponseComplete() throws
IOException
    {
        super.onResponseComplete();
        String responseContent =
this.getResponseContent();

        // do something with the response
content
        ...
    }
};

exchange.setMethod("GET");
exchange.setURL("http://www.example.com/");
```

```
// start the exchange
client.send(exchange);
```

Synchronous Request Example

You can perform a synchronous request by using the [HttpExchange.waitForDone\(\)](#) method.

```
public static void main(String[] args)
{
    HttpClient httpClient = new HttpClient();
        //set up httpClient
        httpClient.start();
    ContentExchange contentExchange = new
ContentExchange();

    httpClient.setConnectorType(HttpClient.CONNE
CTOR_SELECT_CHANNEL);

    contentExchange.setURL("http://slashdot.org"
);

    httpClient.send(contentExchange);

    contentExchange.waitForDone();
    System.err.println("Response status:
"+contentExchange.getResponseStatus());
}
```