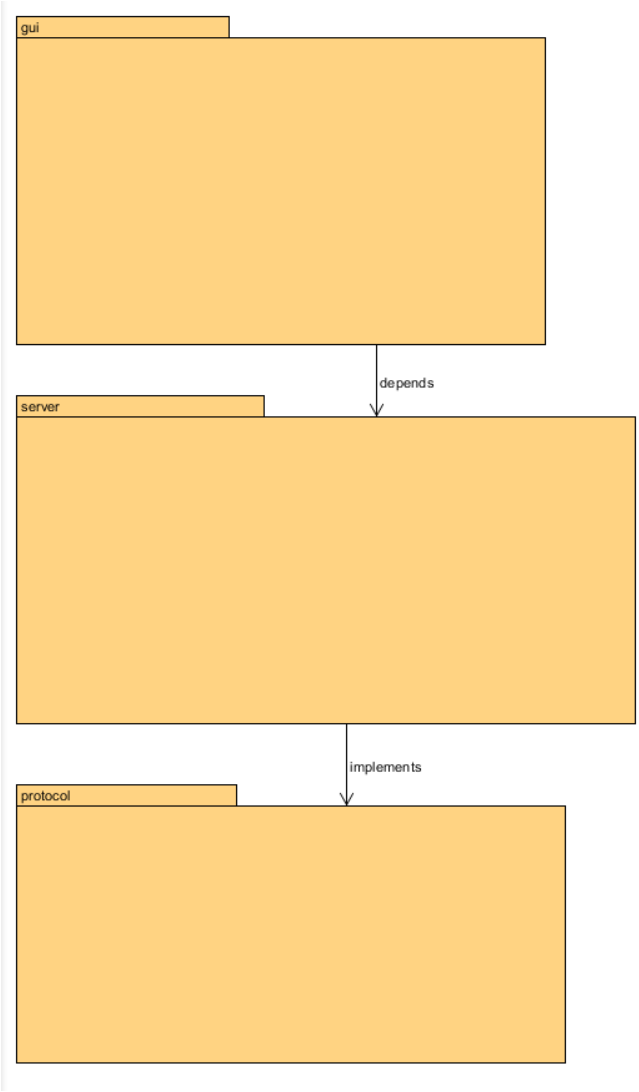
Web Server Architecture

No changes were made to the original server architecture. All changes were made within the modules themselves.

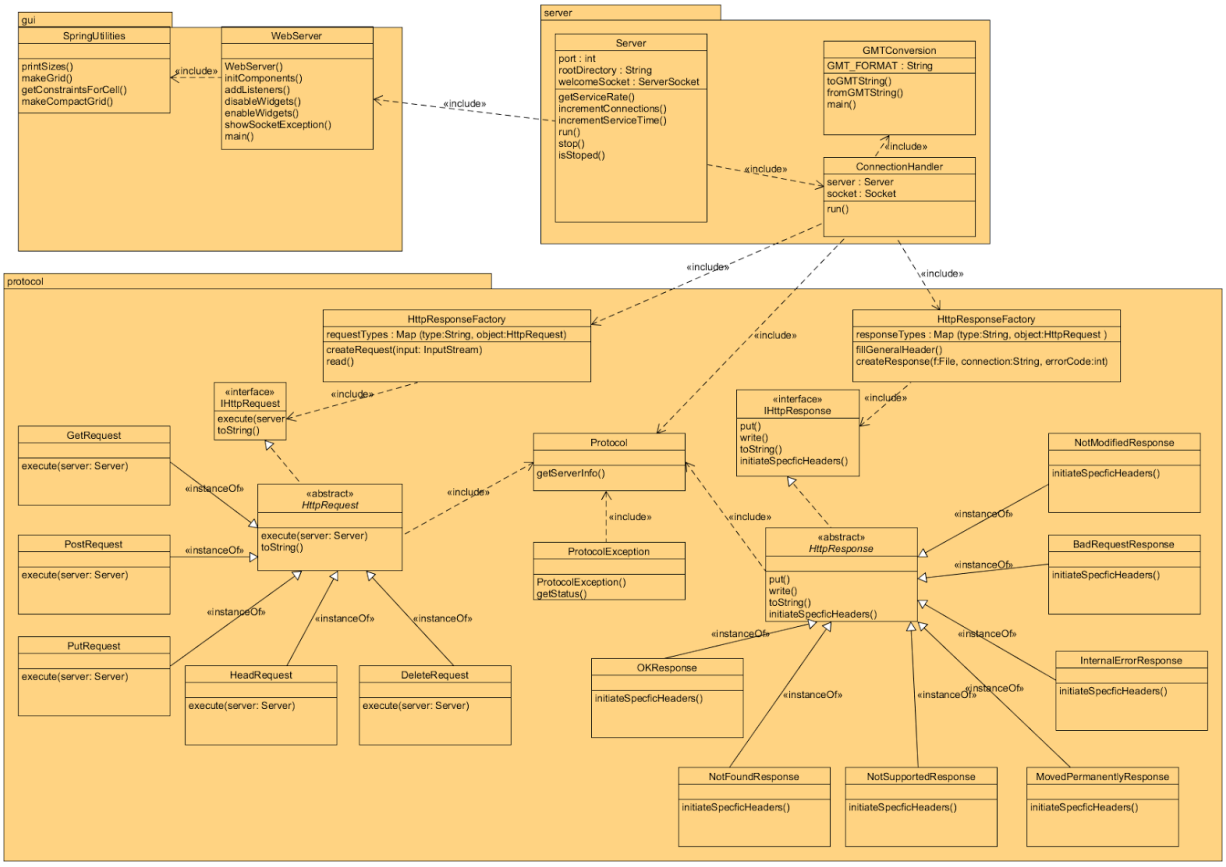


Web Server Design

Below can be found the architecture of the web server. It now implements interfaces for requests and responses instead of a generic HttpRequest and HttpResponse allowing for increased modifiability as long as the requirements of the interfaces are met.
We added a HttpRequestFactory to facilitate additional request types. This factory uses a map of the request types and their associated objects. To allow additional types of requests to be easily added, we created the IHttpRequest interface and the HttpRequest abstract class. All of the existing requests make use of this. We also moved the execution of the specific requests to the request objects themselves. This will allow new requests to write their functionality without editing the existing connectionHandler. To add an additional request type,

a user would simply have to have that request implement the abstract class, add the string command to the protocol, and add that string and the new object to the map in HttpRequestFactory.

In addition to adding the already present methods in the original HttpResponse to the IHttpResponse interface, an additional method for generating response specific headers was added and the HttpResponse was made into an abstract class implementing IHttpResponse. This is currently only used in the 200 OkResponse. This is because the HttpResponseFactory originally added additional headers about the file returned in the response. In addition the individual create responses that were contained within the HttpResponseFactory were removed and a generic createResponse method was added that takes the response code in addition to the original parameters of the individual create methods. It uses a map which uses the string value of the response code to create the correct IHttpResponse object and returns it. Since the map does not have access to the file yet an additional method was added to the interface and abstract classes which allows the file to be set after initiation of the class. Each individual response type has a blank constructor. These constructors automatically call the super constructor with the correct parameters. In addition to these changes a MovedPermanentlyResponse and an InternalErrorResponse were added and their codes and text were added to the Protocol. The HttpResponseFactory will now generate an InternalErrorResponse whenever an attempt is made to send an unsupported response.



Design Patterns

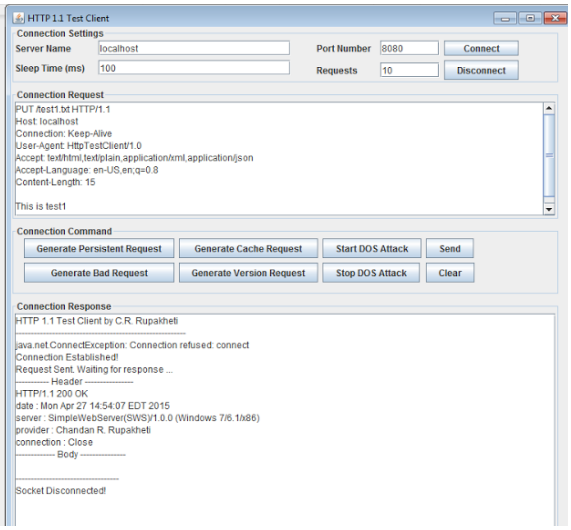
We made use of several design patterns in our new design. We created abstractions of the HttpRequest and HttpResponse to allow these classes to be treated generically. This also allows for extensibility of the existing request and response types. We added a HttpRequestFactory and improved on the existing HttpResponseFactory. The Factory design pattern allows us to easily create specific instances of the abstract response and requests types.

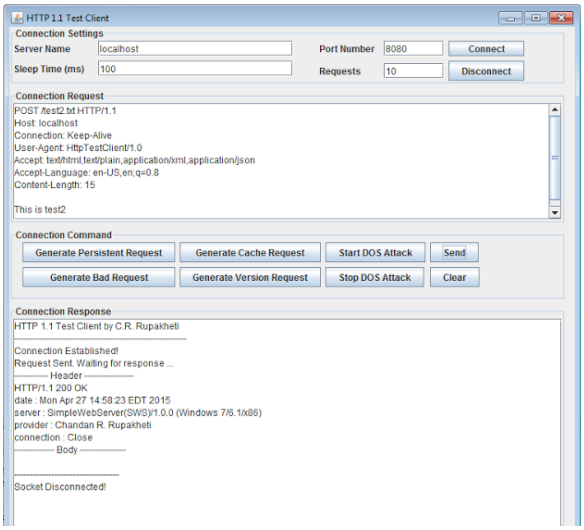
Improvements

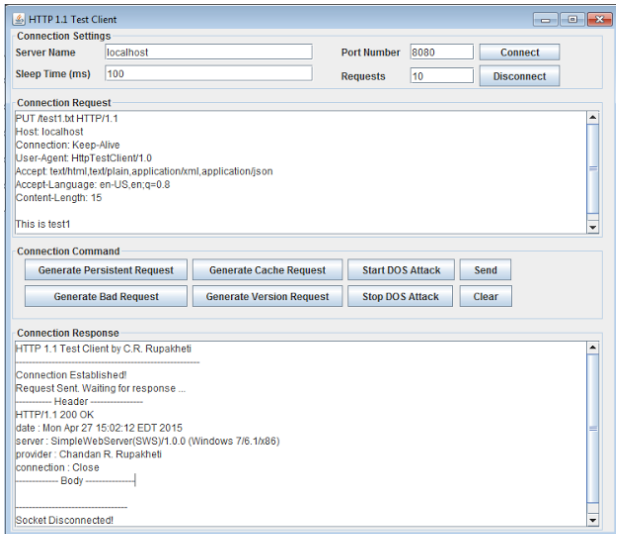
With our current implementation, all of the request types have similar execute methods. Some of the functionality and basic error checking could be abstracted to the HttpRequest class. The put and post request in particular are very similar; the only major difference being whether they overwrite the existing file. These requests could be combined into a file writer superclass, which both requests inherit. An additional improvement could include extracting the requests and responses out into individual plugins and automatically adding them to the maps in the factory classes to allow for drag and drop of jar files to update code.

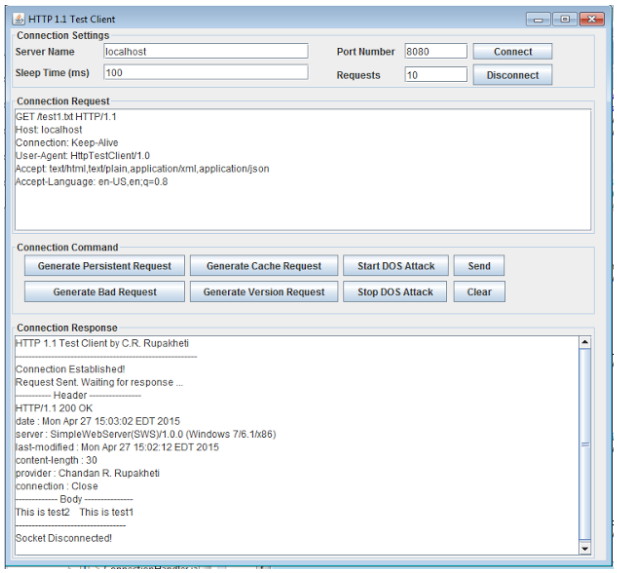
Test Report

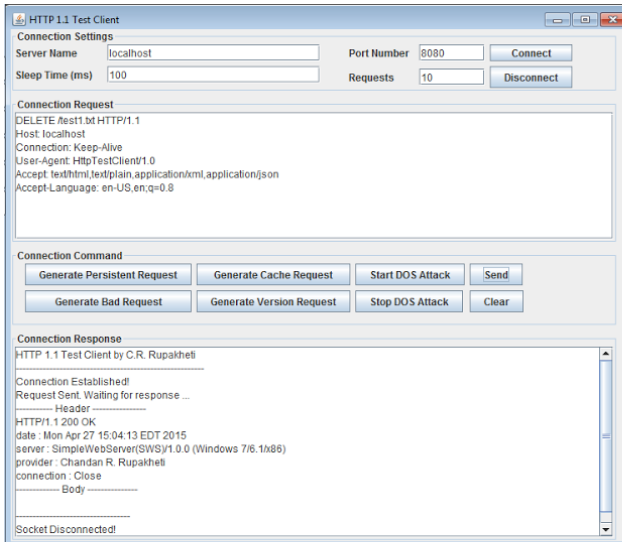
The following few pages show our testing results. First we create a file using PUT. Then we overwrite the contents of that file using POST. Next we DELETE the file. Finally we try a final GET method, which returns a 404, as the file has been deleted.

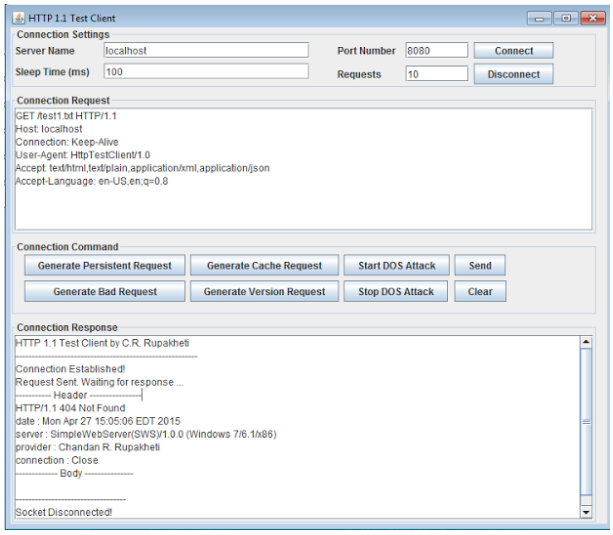

HTTP 1.1 Test Client
Connection Settings
Server Name
localhost
Port Number
8080
Connect
Sleep Time (ms)
100
Requests
10
Disconnect
Connection Request
PUT /test1.txt HTTP/1.1
Host: localhost
Connection: Keep-Alive
User-Agent: HttpTestClient/1.0
Accept: text/html,text/plain,application/xml,application/json
Accept-Language: en-US,en;q=0.8
Content-Length: 15
This is test1
Connection Command
Generate Persistent Request
Generate Cache Request
Start DOS Attack
Send
Generate Bad Request
Generate Version Request
Stop DOS Attack
Clear
Connection Response
HTTP 1.1 Test Client by C.R. Rupakheti
java.net.ConnectException: Connection refused: connect
Connection Established!
Request Sent. Waiting for response ...
---------- Header ----------
HTTP/1.1 200 OK
date : Mon Apr 27 14:54:07 EDT 2015
server : SimpleWebServer(SWS)/1.0.0 (Windows 7/6.1/x86)
provider : Chandan R. Rupakheti
connection : Close
---------- Body ----------
Socket Disconnected!


HTTP 1.1 Test Client
Connection Settings
Server Name
localhost
Port Number
8080
Connect
Sleep Time (ms)
100
Requests
10
Disconnect
Connection Request
POST /test2.txt HTTP/1.1
Host: localhost
Connection: Keep-Alive
User-Agent: HttpTestClient/1.0
Accept: text/html,text/plain,application/xml,application/json
Accept-Language: en-US,en;q=0.8
Content-Length: 15
This is test2
Connection Command
Generate Persistent Request
Generate Cache Request
Start DOS Attack
Send
Generate Bad Request
Generate Version Request
Stop DOS Attack
Clear
Connection Response
HTTP 1.1 Test Client by C.R. Rupakheti
Connection Established!
Request Sent. Waiting for response ...
---------- Header ----------
HTTP/1.1 200 OK
date : Mon Apr 27 14:58:23 EDT 2015
server : SimpleWebServer(SWS)/1.0.0 (Windows 7/6.1/x86)
provider : Chandan R. Rupakheti
connection : Close
---------- Body ----------
Socket Disconnected!


HTTP 1.1 Test Client
Connection Settings
Server Name
localhost
Port Number
8080
Connect
Sleep Time (ms)
100
Requests
10
Disconnect
Connection Request
PUT /test1.txt HTTP/1.1
Host: localhost
Connection: Keep-Alive
User-Agent: HttpTestClient/1.0
Accept: text/html,text/plain,application/xml,application/json
Accept-Language: en-US,en;q=0.8
Content-Length: 15
This is test1
Connection Command
Generate Persistent Request
Generate Cache Request
Start DOS Attack
Send
Generate Bad Request
Generate Version Request
Stop DOS Attack
Clear
Connection Response
HTTP 1.1 Test Client by C.R. Rupakheti
Connection Established!
Request Sent. Waiting for response ...
---------- Header ----------
HTTP/1.1 200 OK
date : Mon Apr 27 15:02:12 EDT 2015
server : SimpleWebServer(SWS)/1.0.0 (Windows 7/6.1/x86)
provider : Chandan R. Rupakheti
connection : Close
---------- Body ----------
Socket Disconnected!


HTTP 1.1 Test Client
Connection Settings
Server Name
localhost
Port Number
8080
Connect
Sleep Time (ms)
100
Requests
10
Disconnect
Connection Request
GET /test1.txt HTTP/1.1
Host: localhost
Connection: Keep-Alive
User-Agent: HttpTestClient/1.0
Accept: text/html,text/plain,application/xml,application/json
Accept-Language: en-US,en;q=0.8
Connection Command
Generate Persistent Request
Generate Cache Request
Start DOS Attack
Send
Generate Bad Request
Generate Version Request
Stop DOS Attack
Clear
Connection Response
HTTP 1.1 Test Client by C.R. Rupakheti
Connection Established!
Request Sent. Waiting for response ...
---------- Header ----------
HTTP/1.1 200 OK
date : Mon Apr 27 15:03:02 EDT 2015
server : SimpleWebServer(SWS)/1.0.0 (Windows 7/6.1/x86)
last-modified : Mon Apr 27 15:02:12 EDT 2015
content-length : 30
provider : Chandan R. Rupakheti
connection : Close
---------- Body ----------
This is test2 This is test1
Socket Disconnected!


HTTP 1.1 Test Client
Connection Settings
Server Name
localhost
Port Number
8080
Connect
Sleep Time (ms)
100
Requests
10
Disconnect
Connection Request
DELETE /test1.txt HTTP/1.1
Host: localhost
Connection: Keep-Alive
User-Agent: HttpTestClient/1.0
Accept: text/html,text/plain,application/xml,application/json
Accept-Language: en-US,en;q=0.8
Connection Command
Generate Persistent Request
Generate Cache Request
Start DOS Attack
Send
Generate Bad Request
Generate Version Request
Stop DOS Attack
Clear
Connection Response
HTTP 1.1 Test Client by C.R. Rupakheti
Connection Established!
Request Sent. Waiting for response ...
---------- Header ----------
HTTP/1.1 200 OK
date : Mon Apr 27 15:04:13 EDT 2015
server : SimpleWebServer(SWS)/1.0.0 (Windows 7/6.1/x86)
provider : Chandan R. Rupakheti
connection : Close
---------- Body ----------
Socket Disconnected!


HTTP 1.1 Test Client
Connection Settings
Server Name
localhost
Port Number
8080
Connect
Sleep Time (ms)
100
Requests
10
Disconnect
Connection Request
GET /test1.txt HTTP/1.1
Host: localhost
Connection: Keep-Alive
User-Agent: HttpTestClient/1.0
Accept: text/html,text/plain,application/xml,application/json
Accept-Language: en-US,en;q=0.8
Connection Command
Generate Persistent Request
Generate Cache Request
Start DOS Attack
Send
Generate Bad Request
Generate Version Request
Stop DOS Attack
Clear
Connection Response
HTTP 1.1 Test Client by C.R. Rupakheti
Connection Established!
Request Sent. Waiting for response ...
---------- Header ----------
HTTP/1.1 404 Not Found
date : Mon Apr 27 15:05:06 EDT 2015
server : SimpleWebServer(SWS)/1.0.0 (Windows 7/6.1/x86)
provider : Chandan R. Rupakheti
connection : Close
---------- Body ----------
Socket Disconnected!