**Project Report #2: Building a Multi-Threaded Web Server**

Daniel Hanna

Computer Networking (CSC 360)

The College of New Jersey

Department of Computer Science



This project involved having a Web Server listen to two different port numbers, 8888 and 5555. The intention is to have requests from port 8888 be serviced (and to actually load the content that is requested) and to have requests from 5555 be serviced with a Moved Permanently response. When doing this, it is required to use ServerSocketChannel because SocketChannel can only be used to listen to one port number. The below screenshot demonstrates using ServerSocketChannel to listen to both ports.

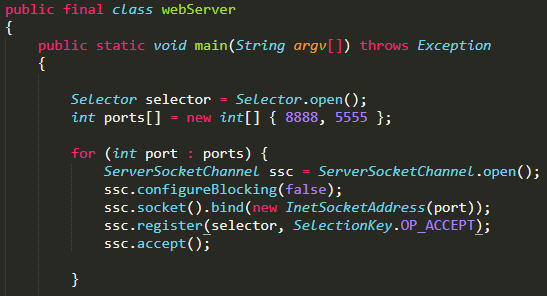


Figure 1: Using ServerSocketChannel to Listen to Both Ports

Selector is then used to select the appropriate port number to be serviced. This is done in an infinite loop, since the Web Server will need to be always listening for requests which may come at any time.

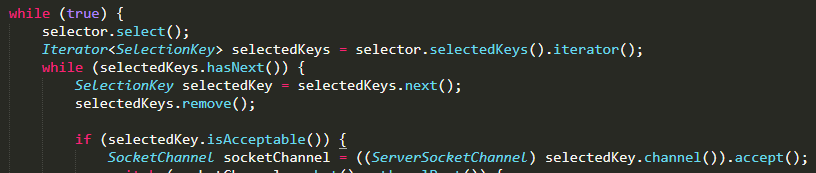


Figure 2: Using Selector in an Infinite Loop

Depending on which port is selected, either an HttpRequest or a MovedRequest will need to be created. This request will be utilized in a new thread in order to make the Web Server be multi-threaded.

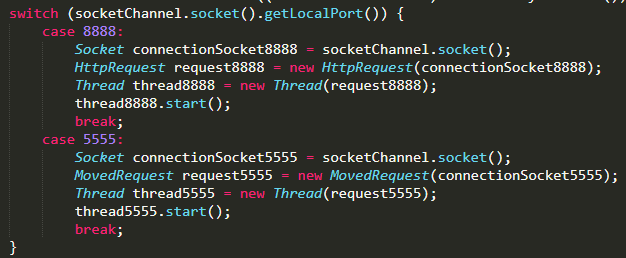


Figure 3: Servicing Requests from Port 8888 and Port 5555

As shown above, in the event where a request is received from port 8888, a new HttpRequest object is created. The HttpRequest class defines the nature of this object.

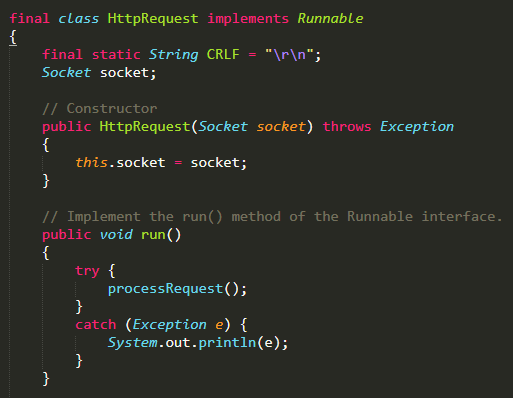


Figure 4: The HttpRequest Class

The HttpRequest class will also attempt to process the request that was received. This includes obtaining the input stream, output stream, and utilizing a buffered reader to find out what file was requested.

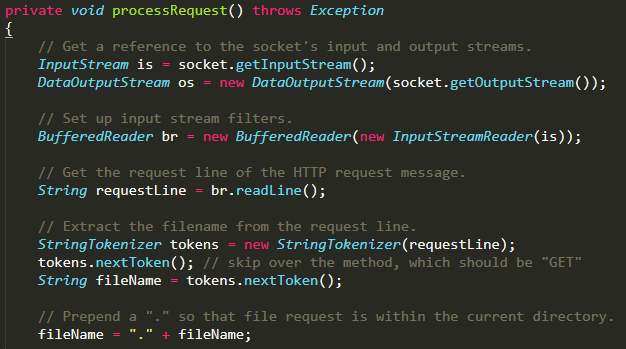


Figure 5: Finding What the Desired File is

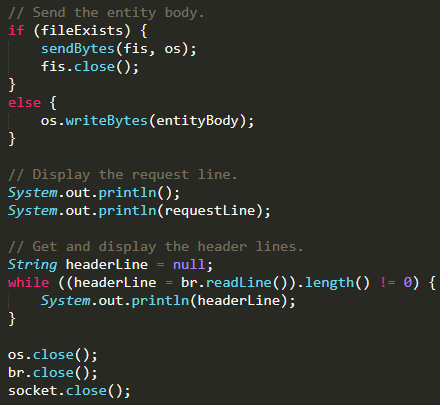
 In the advent that the file exists, an OK status line can be generated and the file can be sent to the requester. However, if the file does not exist, then a Not Found status must be sent instead.

Figure 6: Sending the Appropriate Response to the Request

A buffer is also utilized for sending the data.

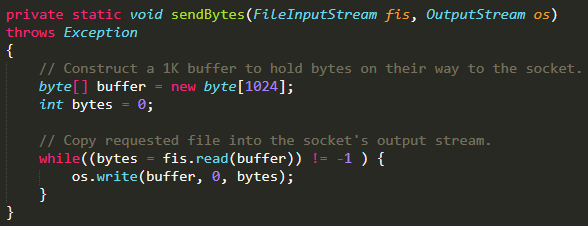


Figure 7: Buffering the Data that is to be Sent

The type of data that is to be sent is determined by the filename’s ending.



Figure 8: Defining the Type of Content that is Being Sent

The MovedRequest class is almost an identical copy to the HttpRequest class. However, this class will only be utilized if the request that is received is from port 5555. In that event, a Moved Permanently response will be sent, regardless of whether the requested file exists or not.

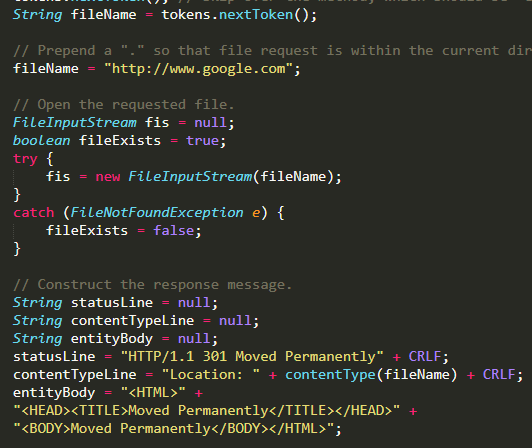
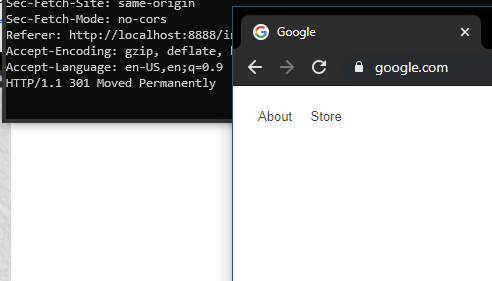


Figure 9: Responding with a 301 Moved Permanently

Testing the program will show that navigating to localhost:8888/index.html will load the appropriate content. Navigating to localhost:5555/index.html will redirect to [www.google.com](http://www.google.com). Thus, the desired behavior is obtained.