Peter Robe
12/1/2017

Design/Implementation

HttpServer receives incoming connections, creates a socket and starts a new ResponseServer thread for each connection. Each instance of ResponseServer creates a PrintWriter and a BufferedReader for input and output. ResponseServer first parses the request using regex Patterns. Only the first line of the http header is required in this case, so I run 3 regular expressions on it (method, file path, and file-type) using the Pattern and Matcher classes. I determine if the request is valid: the file does not exist, the method is not implemented, or it is a bad request. Finally, it sends back a http response with the indicated file in the body. I added support for PNG files since it only involved adding two lines of code. I use a PrintWriter and a BufferedReader in this project.

Troubleshooting

The biggest thing I struggled on was BufferedReader.readLine(). When used with a socket it blocks for input, so it took me a while to find the BufferedReader.ready() function.