

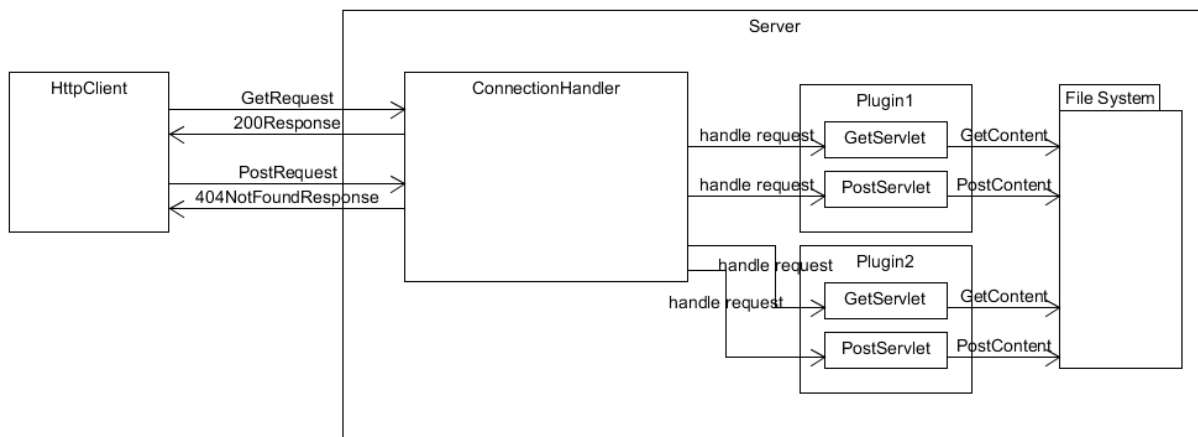
Andrew Davidson, Noah Miller

Milestone 2

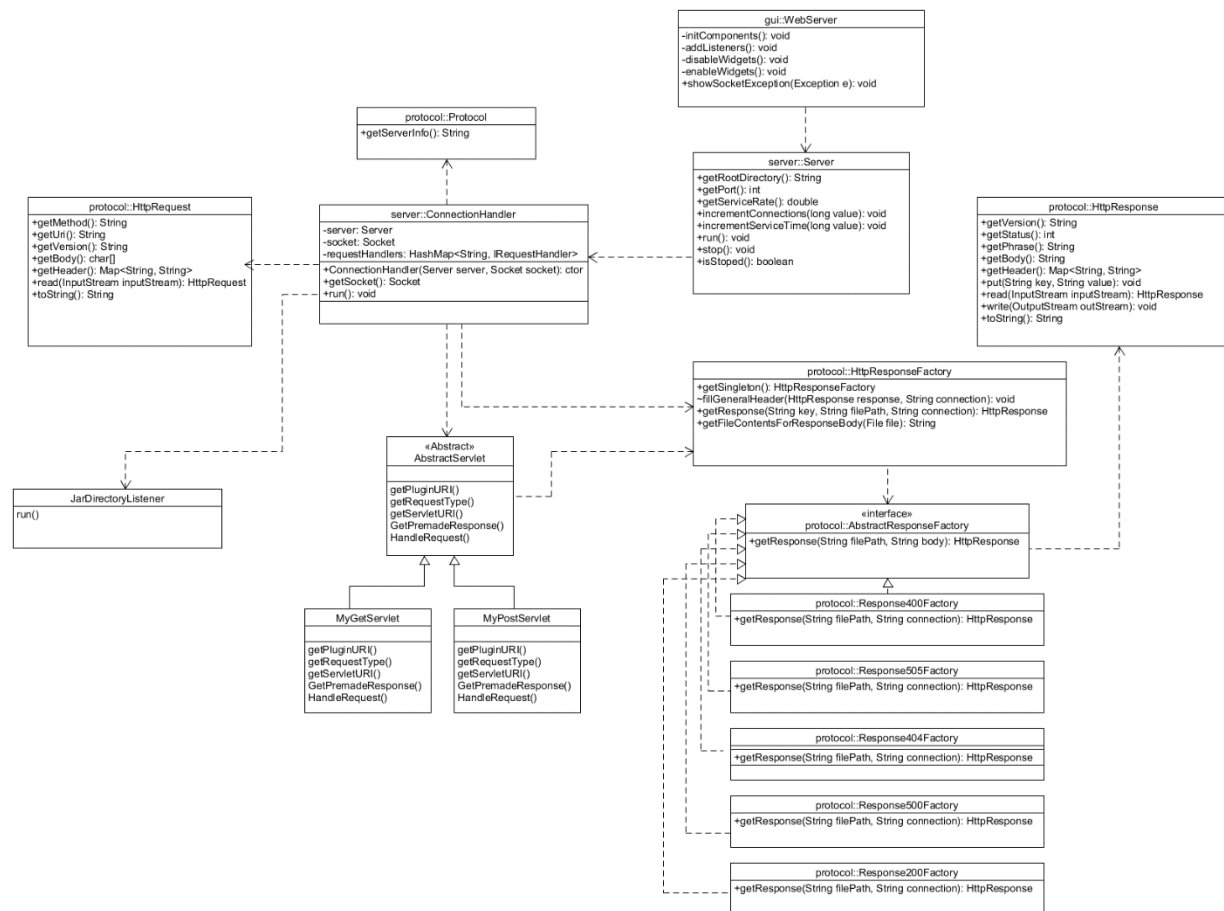
1. Change History:

- We refactored our abstract response factory methods to not take in a filePath as a parameter. This was already unnecessary for most methods, and our new design did not require it at all.
- We added a setBody() method to HttpResponse. This is called in our request handlers to set the body of the response only if needed.
- We added a pluggable architecture. Developers can create request handlers by creating a jar file. Each class within the jar represents a servlet and should extend our AbstractPluginServlet class. Classes that do not extend the abstract class will not be loaded or used.
- We added a directory listener for plugins. Dragging and dropping a plugin will be loaded at runtime. Removing the plugin will also be loaded at runtime. Dragging and dropping a plugin with the same name of an existing plugin will only load servlets from the new plugin that have not already been loaded. The original plugin will still be useable.
- We moved all of our request handlers to servlet classes. All request handlers are now loaded as plugins.

2. Updated Architecture Diagram:



3. Updated Detailed Design:



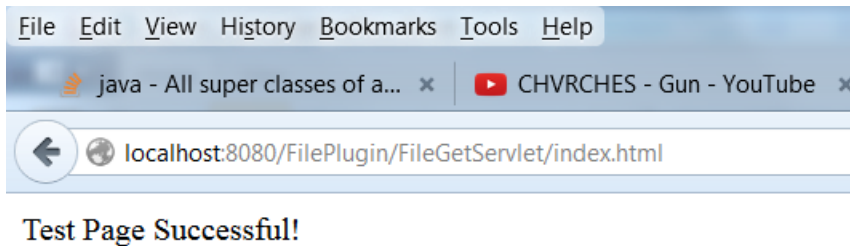
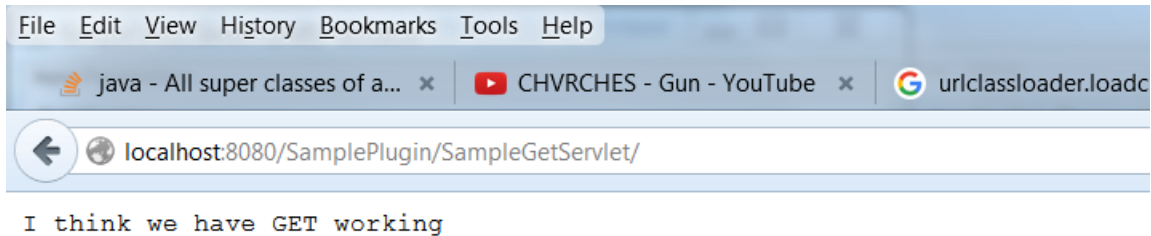
The new design allows developers to create and add request handlers. Adding a handler obeys the open closed principle by using a pluggable architecture. Developers can create “servlets” that extend our AbstractPluginServlet class, but they do not need to modify any of the existing code in order for a user to handle a request with the developer’s servlet. Instead, they just drop their plugin into our “plugins” folder. In addition, this can be done at runtime which eliminates the need to shut down the server for loading different types of requests. This allows developers to specify different behaviors for the same types of requests. When a user makes a request, they supply a URI for a plugin and a URI for a servlet to handle the request. If the servlet URI is mapped to an existing servlet, the request will be handled by the servlet.

4. Feature Listing:

- Refactoring abstract response methods – **Andrew Davidson**
- Creating Response servlets – **Andrew Davidson**
- Testing – **Andrew Davidson**
- Dragging and Dropping plugins – **Noah Miller and Andrew Davidson**
- Loading plugins and servlets – **Noah Miller**

5. Test Report:

We have added unit tests to our project which use a mock WebServer to enable us to test our project. We have tested both of our implementations of servlets. Below is a picture of both of our get methods tested:



6. Further Improvements:

We would like to refactor the `connectionHandler` code. At the current moment, it is somewhat disorganized and hard to follow. We would also like to make similar changes to our `read` and `write` methods in `HttpResponse` and `HttpRequest`.