There have been countless changes and additions to Java over the years as different versions are released. For this week’s post I will be focusing on the changes that occurred in Java 9.

Java 9 saw several changes and improvements including language modifications, process APIs and more. One improvement in particular that I found interesting is the new, completely rebuilt, HTTP client.

An HTTP client manages requests and responses from a webserver. This is part of what allows us to build and navigate to web pages. Many programming languages have some functionality to allow for an HTTP client including JavaScript, Python, Ruby and of course Java.

Part of this overhaul included a modernization to allow for the support of HTTP/2, which is a newer version of HTTP aimed at reducing latency on web pages. Additional changes of note include, request and response bodies, WebSocket support and a modern API.

The logistics of releasing a new version of something as huge and widely used as Java is mindboggling to me. I suppose the first step is sifting through user feedback and determining what people are asking for the most.

Another driving factor, I would think, would be considering how the language has aged with new and emerging technologies. Just like operating systems, or any long term use technology for that matter, languages like Java need to be refreshed every so often in order to stay on the cutting edge.

Special thanks to Lindsey for introducing me to baeldung.com, the website I used as a source for this post.

<https://www.jetbrains.com/help/idea/http-client-in-product-code-editor.html>

<https://www.baeldung.com/new-java-9>