**Web application Performance Improvement activities**

Below are the steps to be followed to improve the speed of web applications

1. Minification of JS, CSS and HTML files
2. Compression of Request and Response Objects
3. Usage of HTTP/2 protocol over HTTP/1

**Minification:**

While deploying JSS, CSS and HTML (Optional) files to production environment we need to minify these files so that the size required to transfer over the network will reduce, which in turn increases the speed over the network.

You can use the below URL to minify your JS and CSS files.

<https://www.minifier.org/>

Use the minified files to deploy in the server.

**Note:** kindly follow JS and CSS coding standards before Minification, otherwise the files will throw errors after Minification.

**Compression:**

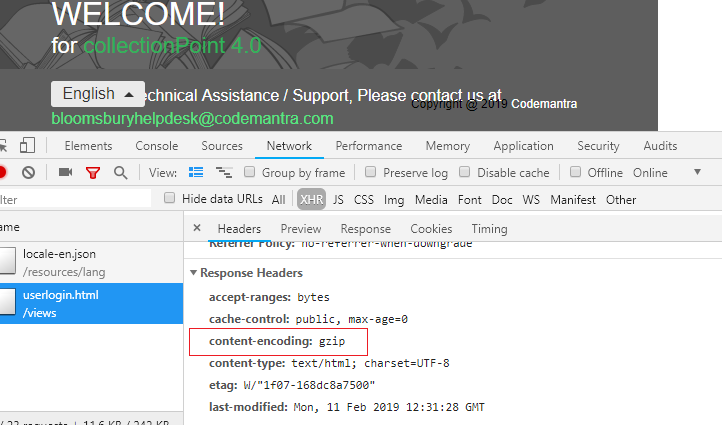
Compression is a built in node feature with express which will compress request and response objects over the network which again reduces the size.

This can be achieved in node by following the below steps.

1. Include "**compression": "^1.7.3**" module in your package.json.
2. Include the below steps in server.js

|  |
| --- |
| var compression = require('compression');  app = express(); **(Avoid this line if already using express)**  app.use(compression()); |

1. Once you include the above steps and run the node server if you see the below line in your response headers of the application Developer Tools, then compression is working in your application.



**HTTP/2 Protocol:**

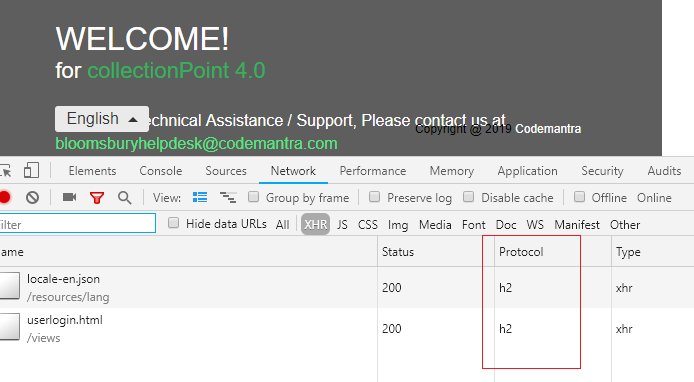
If you are using express framework in your application, then you can enable HTTP/2 using “spdy” module of node js

Below are the steps to follow:

1. Include "spdy": "^4.0.0", in your package.json
2. For HTTP/2, selfSigned certificate (https) is mandatory even in your local environment. You can get selfSigned Certificate using openSSL tools.
3. Instead of creating server using express, we need to create server using spdy.
4. Include below lines in server-config.js (If you have separate js file to run server or include in server.js file)

|  |
| --- |
| const options = {  key : fs.readFileSync('ca.key'),  cert : fs.readFileSync('ca.crt')  }  const server = spdy.createServer(options, app);  server.listen(“your port number”); |

1. Include ca.key and ca.crt certificates in the root folder.
2. Once you run the server you can see change in protocol in Developer Tools.



1. You can use hapi framework if you are not using express which gives better compatibility and built in application logics over express.

Once the above techniques are implemented in the application, we can see better performance