Debugging with JavaScript

Overview

- expectations
- debugging in an IDE
- debugging client code in the browser
- debugging server code in the browser

pdf

Expectations

This short extra assumes :

- that you know how to debug a program e.g. in Java in Eclipse
- that you know the concept of break points, stepping through a program line by line
- that you can look at local variables and the call stack

If you do not, go there to learn or get a refresh.

Debugging in an IDE

If you develop your code in Eclipse or a similar IDE, then you can debug JS code just like you would debug a Java project. I use IntelliJ IDEA. I hear Visual Code is also good. . .

- You need to create a JavaScript project.
- You need to create a Run configuration for your server.
- Then you run your server in the debugger after having set breakpoints in the “conflict zone” :D

Debugging client code in the browser

All modern browsers know have similar JavaScript debugging environments :

- a JavaScript console
- a JS source editor where you can set breakpoints
- a window to display the call stack
- a window to display variables
- a control panel to run-step over-step in-step out. . .
- a window on HTTP requests and responses, including timing

The name of the environment varies : Web Inspector, Inspector, Developer Tools. . .

Debugging server code in the browser

If you do not have experience with debugging JavaScript in an IDE, but already know how to debug JS code in the browser, then you can use the browser debugger as a remote debugger for your server.

Run the server with “-inspect” option :

```
node --inspect server.js 8000
```

Then run Chrome on the same computer and go to URL : `chrome://inspect`

- In that page, you should see all instances of a node interpreter launched with the `--inspect` option. Choose the relevant one.
- Another window appears with the developer tools
- Choose the source folder (the one where server.js is)
- Then you can use the Chrome developer tools as you would on client side JS code.

Other tools

To debug a server, you can use your favorite browser.

Or you can use a simple command line tool named `curl`

- `curl http://localhost:8000/foo` requests the written url with GET
- `curl -i http://localhost:8000/foo` requests the written url and prints the response headers
- `curl -I http://localhost:8000/foo` requests the written url with method HEAD and prints just the headers
- `curl -X METHOD http://localhost:8000/foo/bar/3 ...` does the request with method METHOD (POST, PUT, DELETE, . . .)

curl exists by default on linux and macs, but here it is for windows.