Debugging

In this lab, you will debug a very simple JavaScript program to see one of the issues of debugging asynchronous applications.

# Objectives

In this lab, you will learn how essential breakpoints are when debugging in node.js.

# Open the file in your IDE and debug

1. Open debug.js in your IDE. Set a breakpoint on the first console.log() call in the file and then debug the file as a node.js application.

# Step through the code

2. Try stepping through the code one statement at a time. When you hit the setTimeout() call, you will find yourself deep in node.js code.

# Set a breakpoint

3. Kill the debugging session and set a breakpoint in the second console.log() call that is inside the callback for the setTimeout() call. Restart the debug session and now you can use "continue" to get past all of the node.js internals and into the body of the callback.

# Optional: Use the node.js debugging client

4. Open up a terminal and navigate to the folder with the debug.js file in it. Enter the command node debug debug.js. The debugger will show the top of the file. Enter help at the prompt for a list of available commands.