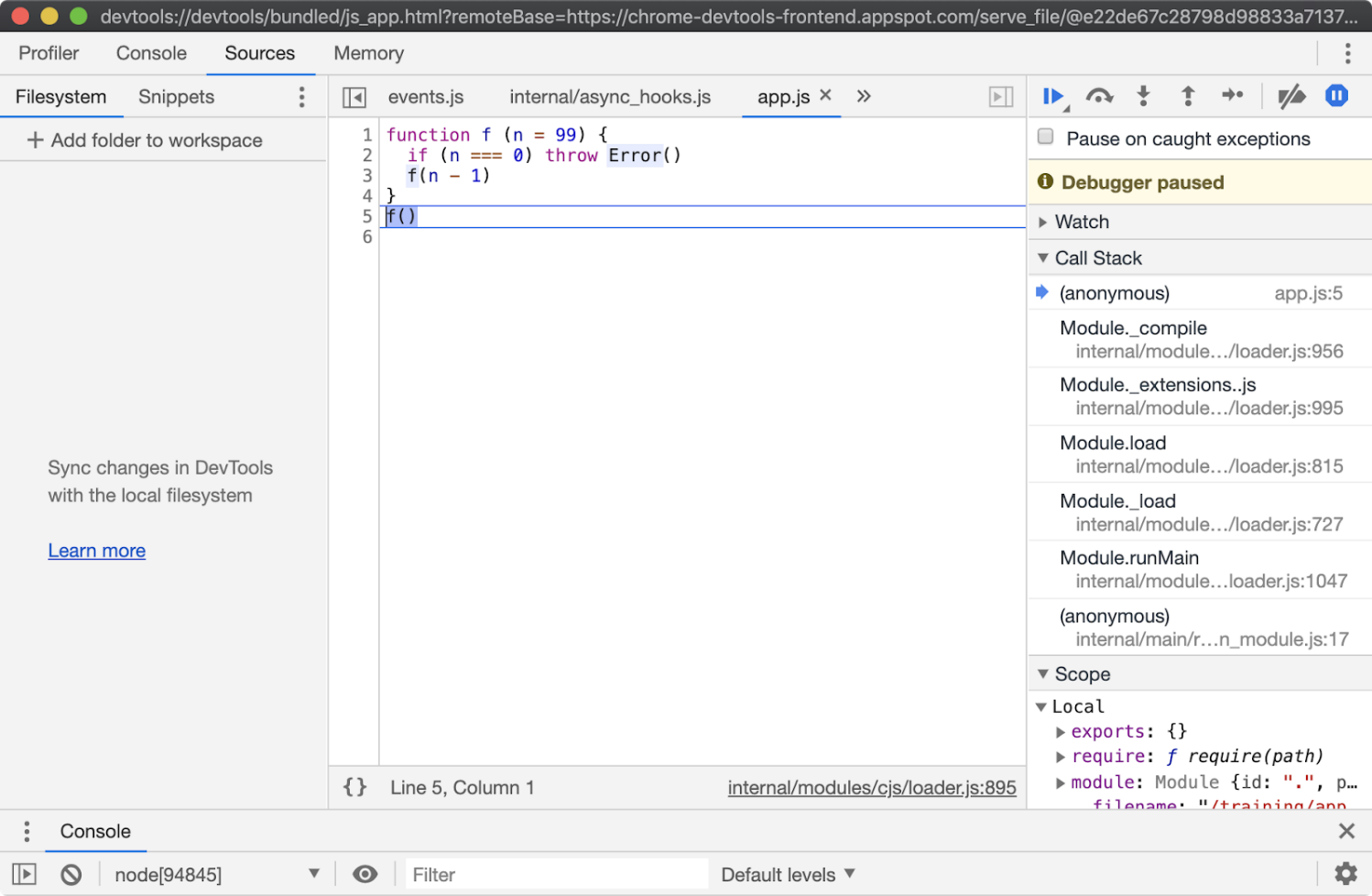
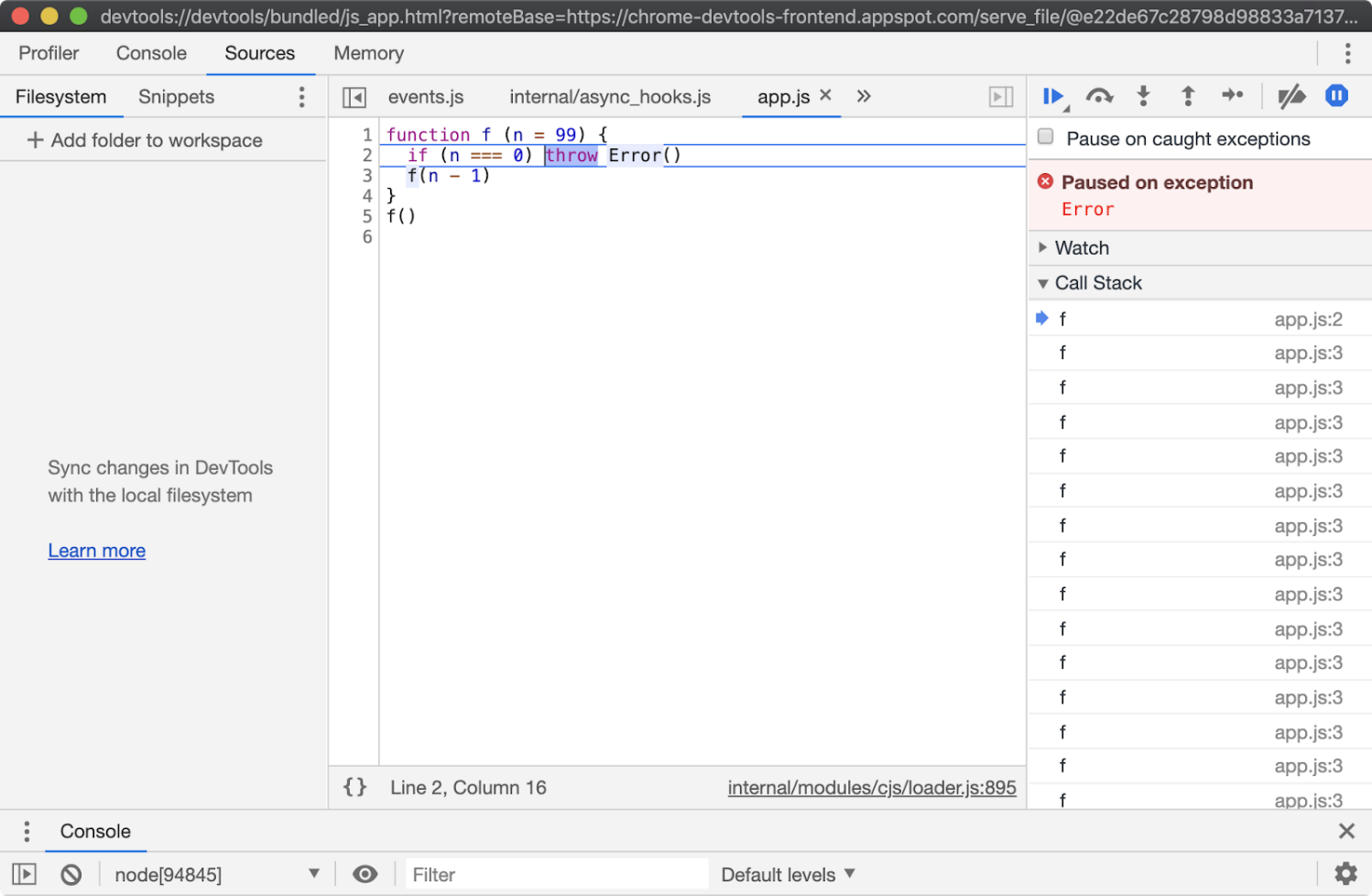
**Breaking on Error in Devtools**

Once a Node.js process has been started in inspect mode and connected to from a debugging client, in this case Chrome Devtools, we can start to try out the debugger features. The **app.js** file will throw when **n** is equal to 0. The "Pause on exceptions" feature can be used to automatically set a breakpoint at the line where an error is thrown.

To activate this behaviour, start **app.js** in Inspect Break mode (**--inspect-brk**), connect Chrome Devtools, ensure that the "Sources" tab is selected and then click the pause button in the top right. The pause button should turn from gray to blue:



Ensure that the "Pause on caught exceptions" checkbox is unchecked and then press the play button. The process should then pause on line 2, where the error is thrown:



From here the Call Stack can be explored over in the right hand column and state can be analyzed by hovering over any local variables and looking in the Scope panel of the right hand column, located beneath the Call Stack panel.

Sometimes a program will throw in far less obvious ways. In these scenarios the "Pause on exceptions" feature can be a useful tool for locating the source of an exception.