

Node.js Debug

五月君-Nodejs技术栈

使用 Chrome Devtools 调试您的 Node.js 程序



五月君

艺术源于生活而高于生活，编程亦是如此。

关注

21 人赞同了该文章

俗话说：“工欲善其事，必先利其器”，调试是每一个开发人员都要遇到的问题，选择一个合适的调试工具也尤为重要。在 Node.js 开发过程中除了万能的 `console.log` 之外，本节介绍一个 Node.js 与 Chrome Devtools 结合的调试工具，以后你可以选择使用浏览器来调试 Node.js 应用程序了。

启动调试器

创建测试代码

```
const fs = require('fs');
const path = require('path');
const filePath = path.resolve(__dirname, 'hello.txt')

console.log('filePath: ', filePath);

fs.readFile(filePath, (err, res) => {
  console.log(err, res.toString());
});
```

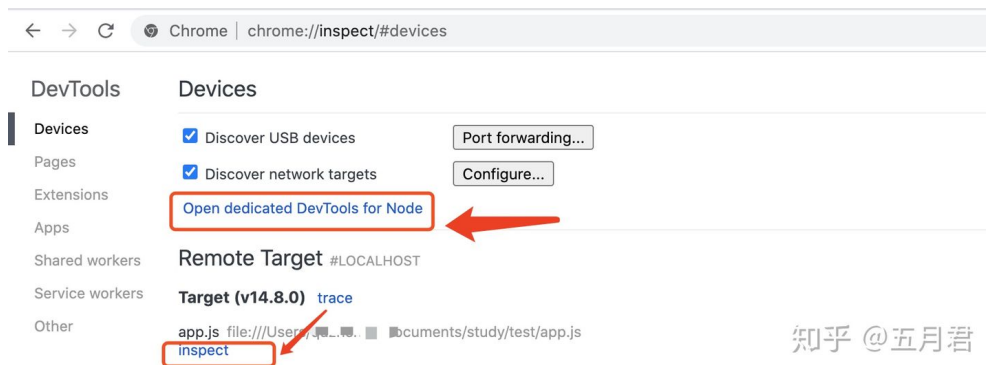
运行带有 `--inspect-brk` 标志的 node

启动时在 node 后面加上 `--inspect-brk` 标志，Node.js 将监听调试客户端，默认情况下监听在 127.0.0.1:9229 地址，也可以显示指定地址 `--inspect-brk=_host:port_`

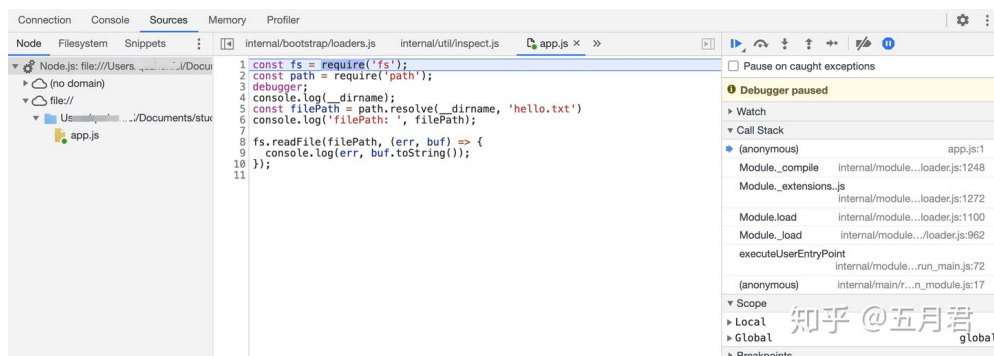
```
$ node --inspect-brk app.js
Debugger listening on ws://127.0.0.1:9229/c7a51e5a-d9be-4506-83fb-0a9340d2b9ba
For help, see: https://nodejs.org/en/docs/inspector
```

注意 node `--inspect` 与 node `--inspect-brk` 的区别：`--inspect` 不会终断，`--inspect-brk` 在用户代码启动之前会终断，也就是代码在第一行就会暂停执行。

浏览器地址栏输入 **chrome://inspect/** 按回车键，如下所示：



Remote Target 下展示了当前运行的 Node.js 版本号，打开 **inspect** 或 **Open dedicated Devtools for Node** 链接，如下所示：

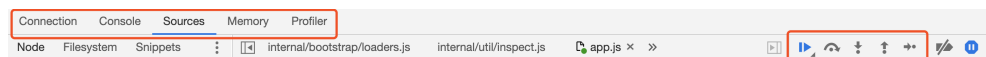


断点调试

调试工具窗口介绍

上方展示与服务器端调试需要的 5 个面板，和 Chrome 开发工具中的是基本相同的，可以理解为“服务端的定制版”

- Connection：链接
- Console：控制台
- Sources：源代码调试（本节主要讲的）
- Memory：内存，查找影响性能的内存问题，包括内存泄漏、内存膨胀和频繁的垃圾回收
- Profiler：性能



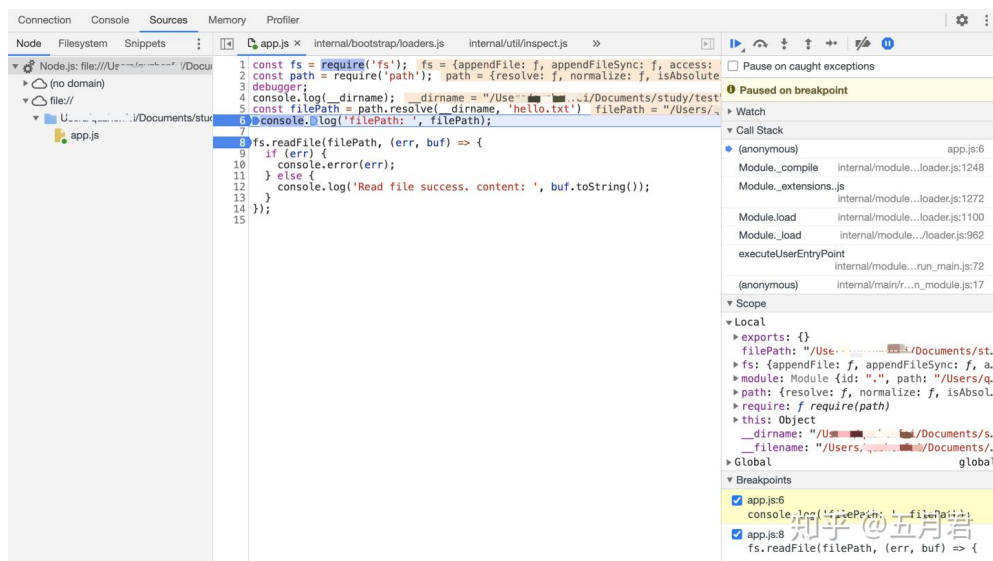
右上角的五個图表，从左至右依次分别表示：

- Resume script execution(F8): 恢复脚本执行，每一次都会自动执行到断点处。
- Step over next function call(F10): 跳过下一个函数调用，执行当前代码行，在当前代码行的下一行处停止，是一步一步的操作。
- Step into next function call(F11): 单步进入下一个函数调用。
- Step out next function call(F11): 单步退出下一个函数调用。
- Step(F9): 执行当前代码行并在下一行处停止。

设置断点

第一种设置断点的方式，是在程序里加入 `debugger` 命令。

第二种设置断点的方式是在编辑器窗口中单击要设置的代码行，此时编辑器窗口中该行会处于被选中状态，还有一个右侧的小箭头。另外右下方 Breakpoints 面板中也展示出了我们设置的断点。



取消断点，再次单击选中的代码行左侧，起到切换作用，或者右键选择 `Remove breakpoint`



欲了解更多断点调试相关内容，参考了解 [Chrome DevTools 更多信息](#)，参考 [使用断点暂停代码](#)

对已启动 Node.js 进程做调试

如果一个 Node.js 进程启动时没有加 `--inspect-brk` 标志，但是我们又不想重启进程来调试，这个时候怎么办？以下两种方式任何一种都可以：

方式一：process._debugProcess(PID)

找到当前启动的 Node 进程 ID，之后使用 `node -e 'process._debugProcess(26853)'` 会建立进程 26853 与调试工具的链接。

```
$ ps ax | grep app.js
26864 s001 S+    0:00.01 grep app.js
26853 s002 S+    0:00.09 node app.js

$ node -e 'process._debugProcess(26853)'
SIGUSR1
```

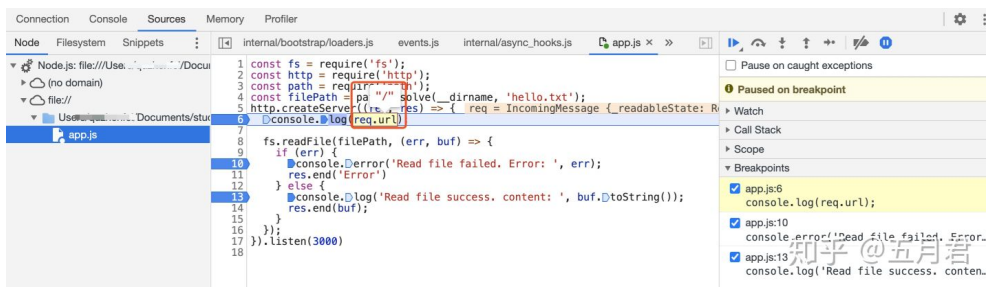
方式二：SIGUSR1 信号

```
$ kill -SIGUSR1 26853
```

测试

以下对 Demo 做了修改，创建一个 HTTP Server 每收到一个请求读取文件，如果按照以上方式开启调试工具后，在浏览器输入 `http://localhost:3000` 回车后，会自动跳转到调试界面，现在你可以设置断点，向上面的方式一样进行调试。

****Tips:** **当前程序运行在断点第 6 行，鼠标移动到 `req.url` 上之后会看到该属性对应的值。



远程调试

如果是在服务器上调试，建议不要让调试器监听公共 IP 地址，这可能会引起远程访问的安全风险，但我们又想用本地的方式调试该怎么办呢？

如果要允许远程调试链接，建议是使用 SSL 隧道的方式，假设我们的服务运行在服务器 `debug.nodejs.red` 上，首先启动服务，和上面的方式一样。

```
$ node --inspect-brk app.js
```

设置 SSH 隧道

在本地计算机上设置 SSH 隧道，这将使本地计算机上端口为 9221 接收的链接转换到服务器 `debug.nodejs.red` 上的 9229 端口。

```
$ ssh -L 9221:localhost:9229 user@debug.nodejs.red
```

Chrome DevTools 调试器的 Connection 中增加链接

默认情况下，Connection 下只有一个 `localhost:9229`，在添加 1 个 `localhost:9221` 之后我们就可以向在本地运行 Node.js 程序一样进行调试。



Specify network endpoint and DevTools will connect to it automatically. Read [Node.js debugging guide](#) to learn more.

localhost:9229

localhost:9221

Add Cancel 3

Add connection

知乎 @五月君

Reference

- [chrome-devtools](#)
- [debugging-getting-started](#)

发布于 2020-12-20 19:32

[Node.js](#) [JS 调试](#) [JavaScript](#)

文章被以下专栏收录

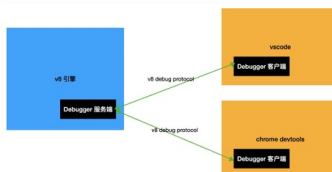


Nodejs技术栈
专注于 Node.js 技术栈分享

推荐阅读

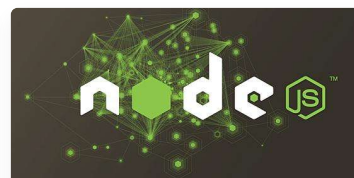
【译】使用 ndb 调试 node 应用

原文链接: [Debugging Node.js Application Using ndb](#) Google Chrome 实验室发布了一款新的 node debug 工具来提升开发者体验, 本文将会全面介绍 ndb 这款 node 工具 熟悉 node 的人可能...
cpselvis



让你 nodejs 水平暴增的 debugger 技巧

神说要有光



翻译计划-用node.js开发一个可交互的命令行应用

也树

发表于前端进击者



灵活使用 con 简单

Funde...

还没有评论

写下你的评论...

