Application Debugging

Whenever your Electron application is not behaving the way you wanted it to, an array of debugging tools might help you find coding errors, performance bottlenecks, or optimization opportunities.

Renderer Process

The most comprehensive tool to debug individual renderer processes is the Chromium Developer Toolset. It is available for all renderer processes, including instances of `BrowserWindow`, `BrowserView`, and `WebView`. You can open them programmatically by calling the `openDevTools()` API on the `webContents` of the instance:

```
const { BrowserWindow } = require('electron')

const win = new BrowserWindow()
win.webContents.openDevTools()
```

Google offers excellent documentation for their developer tools. We recommend that you make yourself familiar with them - they are usually one of the most powerful utilities in any Electron Developer's tool belt.

Main Process

Debugging the main process is a bit trickier, since you cannot open developer tools for them. The Chromium Developer Tools can be used to debug Electron's main process thanks to a closer collaboration between Google / Chrome and Node.js, but you might encounter oddities like `require` not being present in the console.

For more information, see the Debugging the Main Process documentation.

V8 Crashes

If the V8 context crashes, the DevTools will display this message.

`DevTools was disconnected from the page. Once page is reloaded, DevTools will automatically reconnect.`

Chromium logs can be enabled via the `ELECTRON_ENABLE_LOGGING` environment variable. For more information, see the environment variables documentation.

Alternatively, the command line argument `--enable-logging` can be passed. More information is available in the command line switches documentation.