Debugging issues on the Web

The tips below can help you debug Web issues quicker and file more readable Github issues.

Chrome DevTools

Dart Dev Tools are not available when running an app in profile or release mode. However, you can still debug your app using Chrome DevTools. When Flutter builds your app it also produces *source maps*. Source maps is a browser technology that allows you to debug applications that are compiled from any language to JavaScript. Dart, TypeScript, CoffeeScript, and even JavaScript minifiers use it for debugging.

To open Chrome DevTools enter Ctrl + Shift + J (on Linux or Windows) or Command + Options + J (on Mac).

Chrome DevTools provide network request/response information, performance profiling tools, a memory profiler, HTML element debugger, a console with REPL (JavaScript-only), and more.

Getting readable stack traces and profile traces

When building in release mode the dart2js compiler *minifies* your code to reduce the size of the app's JavaScript bundle. This process results in obfuscated class, method, and function names. An exception may look like this:

```
Uncaugh TypeError: Cannot read property 'c' of undefined js_helper.dart:1234
    at a3R.$0 (app.dart:123)
    at afG.b (util.dart:321)
```

To get more readable stack-traces in release builds, you can build and run your app in `--profile` mode. While the compiler does optimize the app in profile mode, it does not obfuscate the symbols from the original source code, making stack traces and profile traces much more readable.

Here's an example of a profile trace recorded in Chrome DevTools in profile mode:

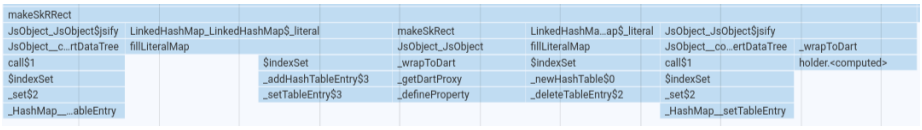

Figure 1: Profile trace sample

You can see that symbols appear unchanged, e.g. "makeSkRRect", and "fillLiteralMap".