

[Skip to content](#)



[Personal](#) [Open source](#) [Business](#) [Explore](#)
[Sign up](#) [Sign in](#)
[Pricing](#) [Blog](#) [Support](#) [Search GitHub](#)

This repository

- [Watch 19](#)
- [Star 330](#)
- [Fork 37](#)

[plasma-umass/browsix](#)

[Code](#) ⓘ [Issues 17](#) [Pull requests 1](#) [Projects 0](#) [Wiki](#) [Pulse](#) [Graphs](#)

Browsix is a Unix-like operating system for the browser.

- [634 commits](#)
- [4 branches](#)
- [0 releases](#)
- [7 contributors](#)
- [MIT](#)

1. [JavaScript 75.8%](#)
2. [TypeScript 23.0%](#)
3. Other 1.2%

[JavaScript](#) [TypeScript](#) Other

Clone with HTTPS ⓘ

Use Git or checkout with SVN using the web URL.



[Open in Desktop](#) [Download ZIP](#)
[Find file](#)

✕ Switch branches/tags

- [Branches](#)
- [Tags](#)

✓ [master](#) ✓ [_new-term](#) ✓ [__sync-syscalls](#) ✓ [_term3](#)











































Nothing to show

Nothing to show

Latest commit [46e686d](#) on Oct 12 [bpowers](#) [kernel: allow closing stdin/stdout/stderr](#)
[Permalink](#)

⚠ Failed to load latest commit information.

app	wait4, wait3, waitpid, wait	7 months ago
bench	move bench stuff out of /test	7 months ago
doc	doc: add meme example to README	6 months ago
examples	examples/go-curl: trailing newline	3 months ago
gulp-plugins	build: add shebang to output commands (breaks all tests)	10 months ago
node-modified/lib	Revert "kernel: http_parser converted to TypeScript"	5 months ago
node	node/lib/buffer: fix Buffer() when called w/ ArrayBuffer (Firefox 48)	4 months ago
src	kernel: allow closing stdin/stdout/stderr	2 months ago
test	test: make xargs less picky	3 months ago

  typings	build: tsconfig, typings	5 months ago
  .gitignore	build: ignore latex example bin dir	6 months ago
  .gitmodules	build: update systemgo submodule location	3 months ago
  AUTHORS	Emery is an author	10 months ago
  LICENSE	build/doc: update readme + licenses	10 months ago
  LICENSE.acorn	build/doc: update readme + licenses	10 months ago
  LICENSE.browserfs	build/doc: update readme + licenses	10 months ago
  LICENSE.browserify	build/doc: update readme + licenses	10 months ago
  LICENSE.node	build/doc: update readme + licenses	10 months ago
  Makefile	Makefile: systemctl target	3 months ago
  README.md	README: added a 'recursive' flag to clone instructions	3 months ago
  analyze.sh	remove 'conf' from tsv, it kills R (doesn't read col in numeric	7 months ago
  benchmark.sh	match up benchmarks between native + browser	7 months ago
  bower.json	build: update bower deps	8 months ago
  gulpfile.js	build: don't pull BrowserFS into browser-node	3 months ago
  karma.conf.js	build: use /lib-dist instead of /dist in preparation for the shell	10 months ago
  package.json	build: pull latest BrowserFS	4 months ago
  tsconfig.json	build: tsconfig, typings	5 months ago
  tslint.json	build: fix tslint errors with latest tslint	9 months ago
  typings.json	build: tsconfig, typings	5 months ago
  xhrfs-index	build: don't omit dotfiles in xhrfs-index	7 months ago

README.md

Browsix - Bringing Unix to the Browser

npm package 0.9.2
dependencies up to date
devDependencies up to date

While standard operating systems like Unix make it relatively simple to build complex applications, web browsers lack the features that make this possible. This project is Browsix, a JavaScript-only framework that brings the essence of Unix to the browser. Browsix makes core Unix features available to web applications (including pipes, processes, signals, sockets, and a shared file system) and extends JavaScript runtimes for C, C++, Go, and Node.js programs so they can run in a Unix-like environment within the browser. Browsix also provides a POSIX-like shell that makes it easy to compose applications together for parallel data processing via pipes.

For more details, check out our [tech report \(PDF\)](#).

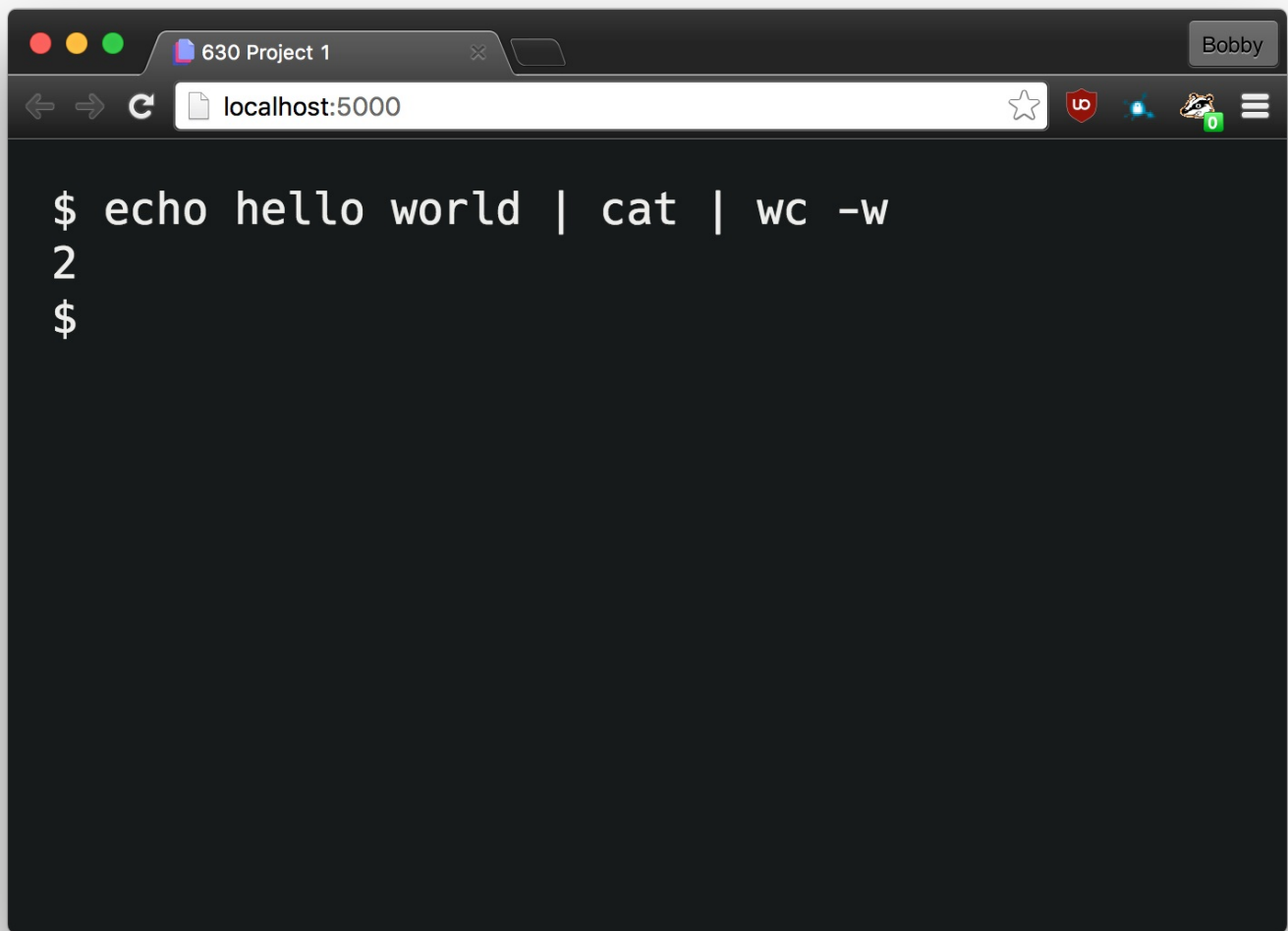
Another way to think about this is that modern web applications are multi-process by nature - the client and some of the application logic lives in the browser, and some of it lives in the cloud, often implemented as [microservices](#).

Browsix lets you rethink the boundary between code executing in the browser vs. server-side, while taking advantage of the multi-core nature of modern computing devices.

Browsix enables you to compose the in-browser part of your web applications out of processes. Processes behave as you would expect coming from [Unix](#): they run in parallel with the main browser thread, can communicate over pipes, sockets, or the filesystem, and can create subprocesses. This process model is implemented on top of existing browser APIs, like [web workers](#), so it works in all modern browsers. Browsix applications can be served statically or over a [CDN](#).

The Browsix Shell

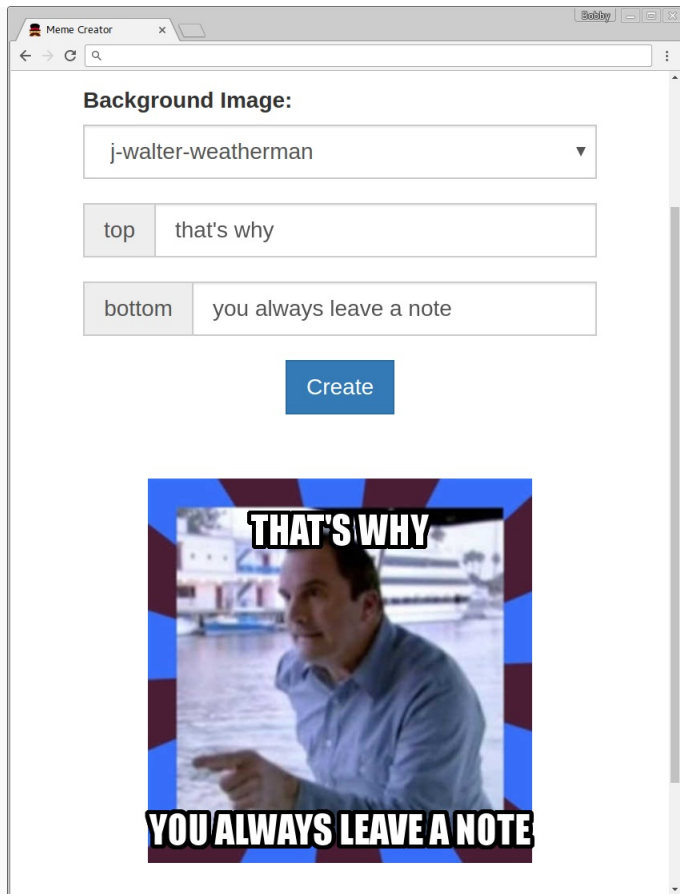
As a proof of concept, we've implemented a POSIX-like shell on top of Browsix, along with an implementation of a number of standard Unix utilities (cat, tee, echo, sha1sum, and friends). The utilities are all standard node programs that will run directly under node, or in the browser under Browsix. Individual commands are executed in their own workers, and piping works as expected:



Try it out here: [live demo!](#)

Meme creator

Browsix is useful for more than web terminals. With Browsix, you can run Go microservices directly in the browser! As an example, we have implemented a meme creator, that lets you create memes (sometimes known as image macros) with (hopefully) humorous text on top of several images. We wrote this as a standard REST service in Go, accepting the text and image type as parameters, and returning a PNG. We used [our modified GopherJS](#) compiler to compile the Go service (including all dependencies, such as the TrueType font renderer and image manipulation libraries) to JavaScript, and Browsix to run this JavaScript as a process in a background Web Worker. We then dynamically route requests to either this in-browser server or a remote server depending on user agent and network connectivity.



Details

Browsix currently supports running node.js, Go, and C/C++ programs. It supports Go with a [modified GopherJS compiler](#), and C/C++ with [modifications to Emscripten](#).

Using Browsix

There are two parts to Browsix: build tooling (the modified Go + C compilers) and runtime support (the kernel + Browsix APIs).

Get browsix through npm:

```
$ npm install --save browsix
```

Building & Testing

Browsix requires **nodejs 4.3.0** or later, which is more recent than the version packaged in Ubuntu Wily. To get a recent version of node, follow the instructions on the [node.js website](#). If you don't know whether you should choose node 4.x or 5.x, choose 4.x (it is the long-term support branch).

Browsix has three other dependencies: git, npm (usually installed along with node), and make, and builds on OSX and Linux systems. Once you have those dependencies:

```
$ git clone --recursive https://github.com/plasma-umass/browsix
$ cd browsix
$ make test-once serve
```

This will pull the dependencies, build the runtime and all the utilities, run a number of tests in either Firefox or Chrome, and then launch a copy of the shell served locally.

In-browser node limitations

Browsix's browser-node implementation has an important to understand limitation: **you must explicitly call `process.exit()`**. Without this, utilities will work under real-node, but appear to hang under browser-node. This is not an intrinsic limitation, but it is a hairy implementation detail -- node exits when the event loop is empty, and there are no active timers or network callbacks. For us to do the same thing means we need to hook `setTimeout` and any other functions that take callbacks to ensure we don't exit early.

Documentation

For a high-level overview of the system design and architecture, please see [this document](#).

🔗Contributing

You're interested in contributing? That's great!

The process is similar to other open-source projects hosted on github:

- Fork the repository
- Make some changes
- Commit your changes with a descriptive commit message
- Open a pull request

🔗Contact

If you have questions or problems, please [open an issue](#) on this repository (plasma-umass/browsix).

🔗Open Source

This project is licensed under the MIT license, but also incorporates code from other sources.

Browsix uses [BrowserFS](#) for its filesystem, which is [primarily MIT licensed](#).

browser-node's [nextTick](#) implementation comes from the [acorn](#) project, released under [the MIT license](#).

A large portion of browser-node is the [node](#) standard library, which is [MIT licensed](#).

Functions to convert buffers to utf-8 strings and back are derivative of [browserify](#) implementations (ported to TypeScript), [MIT licensed](#) as well.

- [Contact GitHub](#)
- [API](#)
- [Training](#)
- [Shop](#)
- [Blog](#)
- [About](#)



- © 2016 GitHub, Inc.
- [Terms](#)
- [Privacy](#)
- [Security](#)
- [Status](#)
- [Help](#)

⚠️ You can't perform that action at this time.

⚠️ You signed in with another tab or window. [Reload](#) to refresh your session. You signed out in another tab or window. [Reload](#) to refresh your session.