IonicaBizau / **node-cobol**

COBOL bridge for NodeJS which allows you to run COBOL code from NodeJS.

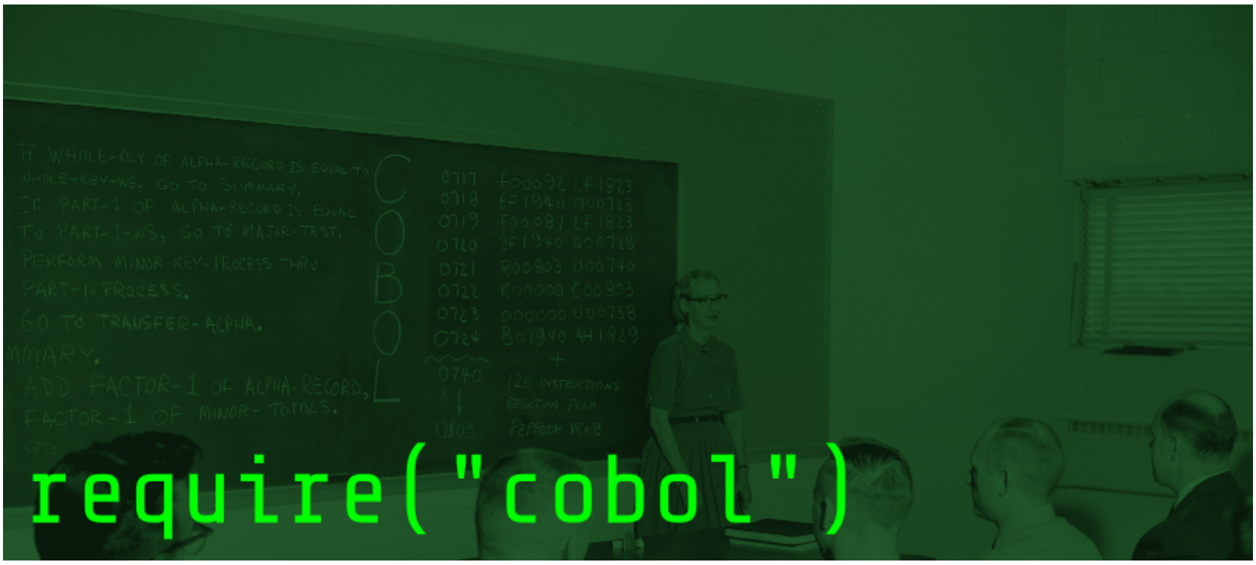
#mad-science

88 commits | **2** branches | **16** releases | **6** contributors | MIT

Branch: master ▾ | New pull request | Create new file | Upload files | Find file | Clone or download ▾

IonicaBizau 1.6.2		Latest commit 75ae1b5 on 14 Jun
cobol-samples	Moving cobol examples in cobol-samples	2 years ago
examples	1.6.0	a year ago
lib	1.6.0	a year ago
.gitignore	Updated docs	3 months ago
CONTRIBUTING.md	Updated docs	a year ago
DOCUMENTATION.md	Updated docs	a year ago
LICENSE	Updated docs	5 months ago
README.md	Updated docs	3 months ago
package.json	1.6.2	3 months ago

README.md



cobol

Patreon | Buy me a book | Donate | npm v1.6.1 | downloads 3k

> COBOL bridge for NodeJS which allows you to run COBOL code from NodeJS.

Can I use this on production?

Of course, you can! It's production ready! If you ever did such a thing, ping me (@IonicaBizau).

Installation

Currently GNUCobol is required. If you are using a debian-based distribution you can install it using:

```
$ sudo apt-get install open-cobol
```

💡 It would be interesting to fallback into a COBOL compiler written in NodeJS. Contributions are welcome! 😁

Then, install the `cobol` package.

```
$ npm i cobol
```

📋 Example

```
// Dependencies
var Cobol = require("cobol");

// Execute some COBOL snippets
Cobol(function () {/*
                   IDENTIFICATION DIVISION.
                   PROGRAM-ID. HELLO.
                   ENVIRONMENT DIVISION.
                   DATA DIVISION.
                   PROCEDURE DIVISION.
                   PROGRAM-BEGIN.
                   DISPLAY "Hello world".
                   PROGRAM-DONE.
                   STOP RUN.
                   */}, function (err, data) {
    console.log(err || data);
});
// => "Hello World"

Cobol(__dirname + "/args.cbl", {
    args: ["Alice"]
}, function (err, data) {
    console.log(err || data);
});
// => "Your name is: Alice"

// This will read data from stdin
Cobol(function () {/*
                   IDENTIFICATION DIVISION.
                   PROGRAM-ID. APP.
                   *> http://stackoverflow.com/q/938760/1420197
                   ENVIRONMENT DIVISION.
                   INPUT-OUTPUT SECTION.
                   FILE-CONTROL.
                   SELECT SYSIN ASSIGN TO KEYBOARD ORGANIZATION LINE SEQUENTIAL.
                   DATA DIVISION.
                   FILE SECTION.
                   FD SYSIN.
                   01 ln PIC X(64).
                   88 EOF VALUE HIGH-VALUES.
                   WORKING-STORAGE SECTION.
                   PROCEDURE DIVISION.
                   DISPLAY "Write something and then press the <Enter> key"
                   OPEN INPUT SYSIN
                   READ SYSIN
                   AT END SET EOF TO TRUE
                   END-READ
                   PERFORM UNTIL EOF
                   DISPLAY "You wrote: ", ln
                   DISPLAY "------------"
                   READ SYSIN
                   AT END SET EOF TO TRUE
                   END-READ
                   END-PERFORM
                   CLOSE SYSIN
                   STOP RUN.
                   */}, {
    stdin: process.stdin,
    stdout: process.stdout
}, function (err) {
    if (err) {
        console.log(err);
    }
});
// => Write something and then press the <Enter> key
// <= Hi there!
// => You wrote: Hi there!
// => ------------
```

❓ Get Help

There are few ways to get help:

1. Please post questions on Stack Overflow. You can open issues with questions, as long you add a link to your Stack Overflow question.
2. For bug reports and feature requests, open issues. 🐛
3. For direct and quick help, you can use Codementor. 🚀

📝 Documentation

`Cobol(input, options, callback)`

Runs COBOL code from Node.JS side.

Params

- **Function|String|Path** `input` : A function containing a comment with inline COBOL code, the cobol code itself or a path to a COBOL file.
- **Object** `options` : An object containing the following fields:
- `cwd` (String): Where the COBOL code will run (by default in the current working directory)
- `args` (Array): An array of strings to pass to the COBOL process.
- `free` (Boolean): Use free option while compiling with GnuCobol
- `stdin` (Stream): An optional stdin stream used to pipe data to the stdin stream of the COBOL process.
- `stderr` (Stream): An optional stderr stream used to pipe data to the stdin stream of the COBOL process.
- `stdeout` (Stream): An optional stdout stream used to pipe data to the stdin stream of the COBOL process.
- `remove` (Boolean): Should the compiled executable be removed after running, default is true.
- `precompiled` (Boolean): Run the precompiled executable instead of re-compiling, default is false.
- **Function** `callback` : The callback function called with `err` , `stdout` and `stderr` .

📰 Press Highlights

This project has become more popular than I expected. 😋 If you wrote or found an article about this project that is not in the list below, please add it.

- Calling 1959 from your Web code: A COBOL bridge for Node.js (ArsTechnica, by Sean Gallagher)
- Cobol -- yes, Cobol -- gets a bridge to Node.js (InfoWorld, by Paul Krill)
- Ein COBOL-Plug-in für Node.js (Heise Online, by Alexander Neumann)
- Dit krijg je als je Node.js met COBOL kruist (Webwereld, by Chris Koenis)
- COBOL for Node.js (I Programmer, by Kay Ewbank)
- Nagyon durva: Már COBOL-ból is lehet programozni a Node.js-t (prog.hu, by Sting)
- Micro Focus Updates COBOL for Visual Studio 2015 (eWeek, by Darryl K. Taft)
- Sur GitHub, un projet relie Cobol et Node.js (LeMondeInformatique, by Maryse Gros avec IDG News Service)
- 3 open source projects for modern COBOL development (OpenSource.com, by Joshua Allen Holm)

😋 How to contribute

Have an idea? Found a bug? See how to contribute.

💖 Support my projects

I open-source almost everything I can, and I try to reply everyone needing help using these projects. Obviously, this takes time. You can integrate and use these projects in your applications *for free*! You can even change the source code and redistribute (even resell it).

However, if you get some profit from this or just want to encourage me to continue creating stuff, there are few ways you can do it:

- Starring and sharing the projects you like 🚀
- PayPal—You can make one-time donations via PayPal. I'll probably buy a ~~coffee~~ tea. 🍵
- Patreon—Set up a recurring monthly donation and you will get interesting news about what I'm doing (things that I don't share with everyone).
- **Bitcoin**—You can send me bitcoins at this address (or scanning the code below): `1P9BRsmazNQcuyTxEqveUsnf5CERdq35V6`

Thanks! ♡

💫 Where is this library used?

If you are using this library in one of your projects, add it in this list. ✨

- `cobol-promises` —COBOL bridge for NodeJS with promises support.

✨ Related

- `node.cobol` —Node.js bridge for COBOL which allows you to run Node.js code from COBOL.
- `fortran` —Fortran bridge for Node.js which allows you to run Fortran code from Node.js.
- `node.fortran` —Execute Node.js in your Fortran programs.

📜 License