zaach / jison

Watch 104 | Star 2,066 | Fork 228

Bison in JavaScript. http://jison.org

625 commits | 4 branches | 54 releases | 33 contributors

Branch: master | jison / +

zaach Merge pull request #305 from rumblesan/strictfix ··· Latest commit a9e58a a day ago

examples	Fix spelling error	a year ago
lib	Fixing strict mode function decleration error	14 days ago
ports	Remove path from C# class name.	2 years ago
tests	Fix this.lexer undefined, fixes #246	8 months ago
web	Update docs.html	2 years ago
.gitignore	reverts ignoring log directory produced by jsdoc	2 years ago
.npmignore	reverts ignoring log directory produced by jsdoc	2 years ago
.travis.yml	disable node 0.8 testing	a year ago
CONTRIBUTING.md	add CONTRIBUTING.md with guidelines	8 months ago
Makefile	Use local uglifyjs	8 months ago
README.md	add CONTRIBUTING.md with guidelines	8 months ago
entry.js	switch to esprima and escodegen so browserify will work	3 years ago
package.json	add a license property to package.json	6 months ago

Code
Issues 82
Pull requests 9
Wiki
Pulse
Graphs

SSH clone URL

git@github.com:z

You can clone with HTTPS, SSH, or Subversion.

Download ZIP

README.md

Jison

- issues
- discuss

build passing

An API for creating parsers in JavaScript

Jison generates bottom-up parsers in JavaScript. Its API is similar to Bison's, hence the name. It supports many of Bison's major features, plus some of its own. If you are new to parser generators such as Bison, and Context-free Grammars in general, a good introduction is found in the Bison manual. If you already know Bison, Jison should be easy to pickup.

Briefly, Jison takes a JSON encoded grammar or Bison style grammar and outputs a JavaScript file capable of parsing the language described by that grammar. You can then use the generated script to parse inputs and accept, reject, or perform actions based on the input.

Installation

Jison can be installed for Node using `npm`

Using npm:

```
npm install jison -g
```

Usage from the command line

Clone the github repository for examples:

```
git clone git://github.com/zaach/jison.git
cd jison/examples
```

Now you're ready to generate some parsers:

```
jison calculator.jison
```

This will generate `calculator.js` in your current working directory. This file can be used to parse an input file, like so:

```
echo "2^32 / 1024" > testcalc
node calculator.js testcalc
```

This will print out `4194304` .

Full cli option list:

```
Usage: jison [file] [lexfile] [options]

file        file containing a grammar
lexfile     file containing a lexical grammar

Options:
   -j, --json                    force jison to expect a grammar in JSON format
   -o FILE, --outfile FILE       Filename and base module name of the generated parser
   -t, --debug                   Debug mode
   -m TYPE, --module-type TYPE   The type of module to generate (commonjs, amd, js)
   -p TYPE, --parser-type TYPE   The type of algorithm to use for the parser (lr0, slr,
   -V, --version                 print version and exit
```

Usage from a CommonJS module

You can generate parsers programatically from JavaScript as well. Assuming Jison is in your commonjs environment's load path:

```
// mygenerator.js
var Parser = require("jison").Parser;

// a grammar in JSON
var grammar = {
    "lex": {
        "rules": [
           ["\\s+", "/* skip whitespace */"],
           ["[a-f0-9]+", "return 'HEX';"]
        ]
    },

    "bnf": {
        "hex_strings" :[ "hex_strings HEX",
                         "HEX" ]
    }
};

// `grammar` can also be a string that uses jison's grammar format
var parser = new Parser(grammar);

// generate source, ready to be written to disk
var parserSource = parser.generate();

// you can also use the parser directly from memory

// returns true
parser.parse("adfe34bc e82a");

// throws lexical error
parser.parse("adfe34bc zxg");
```

More Documentation

For more information on creating grammars and using the generated parsers, read the documentation.

How to contribute

See CONTRIBUTING.md for contribution guidelines, how to run the tests, etc.

Projects using Jison

View them on the wiki, or add your own.

Contributors

Githubbers

Special thanks to Jarred Ligatti, Manuel E. Bermúdez

License

> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.