Códigos Generales Sprint 0 PBIO

Generado por Doxygen 1.9.4

Chapter 1

JSON Schema $Ref Parser

1.0.0.1 Parse, Resolve, and Dereference JSON Schema $ref pointers

[](LICENSE)

1.1 The Problem:

You've got a JSON Schema with `$ref` pointers to other files and/or URLs. Maybe you know all the referenced files ahead of time. Maybe you don't. Maybe some are local files, and others are remote URLs. Maybe they are a mix of JSON and YAML format. Maybe some of the files contain cross-references to each other.

```
{
  "definitions": {
    "person": {
      // references an external file
      "$ref": "schemas/people/Bruce-Wayne.json"
    },
    "place": {
      // references a sub-schema in an external file
      "$ref": "schemas/places.yaml#/definitions/Gotham-City"
    },
    "thing": {
      // references a URL
      "$ref": "http://wayne-enterprises.com/things/batmobile"
    },
    "color": {
      // references a value in an external file via an internal reference
      "$ref": "#/definitions/thing/properties/colors/black-as-the-night"
    }
  }
}
```

1.2 The Solution:

JSON Schema $Ref Parser is a full `JSON Reference` and `JSON Pointer` implementation that crawls even the most complex `JSON Schemas` and gives you simple, straightforward JavaScript objects.

- Use **JSON** or **YAML** schemas — or even a mix of both!

- Supports `$ref` pointers to external files and URLs, as well as `custom sources` such as databases
- Can `bundle` multiple files into a single schema that only has *internal* `$ref` pointers
- Can `dereference` your schema, producing a plain-old JavaScript object that's easy to work with
- Supports `circular references`, nested references, back-references, and cross-references between files
- Maintains object reference equality — `$ref` pointers to the same value always resolve to the same object instance
- Tested in Node v10, v12, & v14, and all major web browsers on Windows, Mac, and Linux

1.3 Example

```
$RefParser.dereference(mySchema, (err, schema) => {
  if (err) {
    console.error(err);
  }
  else {
    // `schema` is just a normal JavaScript object that contains your entire JSON Schema,
    // including referenced files, combined into a single object
    console.log(schema.definitions.person.properties.firstName);
  }
})
```

Or use `async/await` syntax instead. The following example is the same as above:

```
try {
  let schema = await $RefParser.dereference(mySchema);
  console.log(schema.definitions.person.properties.firstName);
}
catch(err) {
  console.error(err);
}
```

For more detailed examples, please see the `API Documentation`

1.4 Installation

Install using `npm`:

```
npm install @apidevtools/json-schema-ref-parser
```

1.5 Usage

When using JSON Schema $Ref Parser in Node.js apps, you'll probably want to use **CommonJS** syntax:

```
const $RefParser = require("@apidevtools/json-schema-ref-parser");
```

When using a transpiler such as `Babel` or `TypeScript`, or a bundler such as `Webpack` or `Rollup`, you can use **ECMAScript modules** syntax instead:

```
import $RefParser from "@apidevtools/json-schema-ref-parser";
```

1.6 Browser support

JSON Schema $Ref Parser supports recent versions of every major web browser. Older browsers may require `Babel` and/or `polyfills`.

To use JSON Schema $Ref Parser in a browser, you'll need to use a bundling tool such as `Webpack`, `Rollup`, `Parcel`, or `Browserify`. Some bundlers may require a bit of configuration, such as setting `browser: true` in `rollup-plugin-resolve`.

1.7 API Documentation

Full API documentation is available `right here`

1.8 Contributing

I welcome any contributions, enhancements, and bug-fixes. `Open an issue` on GitHub and `submit a pull request`.

1.8.0.1 Building/Testing

To build/test the project locally on your computer:

1. **Clone this repo**
   `git clone https://github.com/APIDevTools/json-schema-ref-parser.git`
2. **Install dependencies**
   `npm install`
3. **Run the tests**
   `npm test`

1.9 License

JSON Schema $Ref Parser is 100% free and open-source, under the [MIT license](LICENSE). Use it however you want.

This package is `Treeware`. If you use it in production, then we ask that you **`buy the world a tree`** to thank us for our work. By contributing to the Treeware forest you'll be creating employment for local families and restoring wildlife habitats.

1.10 Big Thanks To

Thanks to these awesome companies for their support of Open Source developers

Chapter 2

Change Log

All notable changes will be documented in this file. OpenAPI Schemas adheres to `Semantic Versioning`.

2.1 <a href="https://github.com/APIDevTools/openapi-schemas/tree/v2.0.0" >v2.0.0</a> (2020-03-10)

- Moved OpenAPI Schemas to the `@APIDevTools scope` on NPM
- The "openapi-schemas" NPM package is now just a wrapper around the scoped "@apidevtools/openapi-schemas" package

`Full Changelog`

2.2 <a href="https://github.com/APIDevTools/openapi-schemas/tree/v1.0.0" >v1.0.0</a> (2019-06-22)

Initial release

Chapter 3

OpenAPI Specification Schemas

[](LICENSE)

This package contains **the official JSON Schemas** for every version of Swagger/OpenAPI Specification:

Version	Schema	Docs
Swagger 1.2	v1.2 schema	v1.2 docs
Swagger 2.0	v2.0 schema	v2.0 docs
OpenAPI 3.↩ 0.x	v3.0.x schema	v3.0.3 docs
OpenAPI 3.↩ 1.x	v3.1.x schema	v3.1.0 docs

All schemas are kept up-to-date with the latest official definitions via an automated CI/CD job.

3.1 Installation

You can install OpenAPI Schemas via npm.

```
npm install @apidevtools/openapi-schemas
```

3.2 Usage

The default export contains all OpenAPI Specification versions:

```
const openapi = require("@apidevtools/openapi-schemas");
console.log(openapi.v1);    // { $schema, id, properties, definitions, ...  }
console.log(openapi.v2);    // { $schema, id, properties, definitions, ...  }
console.log(openapi.v3);    // { $schema, id, properties, definitions, ...  }
console.log(openapi.v31);    // { $schema, id, properties, definitions, ...  }
```

Or you can import the specific version(s) that you need:

```
const { openapiV1, openapiV2, openapiV3, openapiV31 } = require("@apidevtools/openapi-schemas");
console.log(openapiV1);    // { $schema, id, properties, definitions, ...  }
console.log(openapiV2);    // { $schema, id, properties, definitions, ... }
console.log(openapiV3);    // { $schema, id, properties, definitions, ... }
console.log(openapiV31);    // { $schema, id, properties, definitions, ... }
```

You can use a JSON Schema validator such as `Z-Schema` or `AJV` to validate OpenAPI definitions against the specification.

```
const { openapiV31 } = require("@apidevtools/openapi-schemas");
const ZSchema = require("z-schema");
// Create a ZSchema validator
let validator = new ZSchema();
// Validate an OpenAPI definition against the OpenAPI v3.0 specification
validator.validate(openapiDefinition, openapiV31);
```

3.3 Contributing

Contributions, enhancements, and bug-fixes are welcome! `Open an issue` on GitHub and `submit a pull request`.

3.3.0.1 Building

To build the project locally on your computer:

1. **Clone this repo**
   `git clone https://github.com/APIDevTools/openapi-schemas.git`
2. **Install dependencies**
   `npm install`
3. **Build the code**
   `npm run build`
4. **Run the tests**
   `npm test`

3.4 License

OpenAPI Schemas is 100% free and open-source, under the [MIT license](LICENSE). Use it however you want.

This package is `Treeware`. If you use it in production, then we ask that you **`buy the world a tree`** to thank us for our work. By contributing to the Treeware forest you'll be creating employment for local families and restoring wildlife habitats.

3.5 Big Thanks To

Thanks to these awesome companies for their support of Open Source developers

Chapter 4

Swagger Specification JSON Schemas

The work on the JSON Schema for the Swagger Specification was donated to the community by `Francis Galiegue`!

Keep in mind that due to some JSON Schema limitations, not all constraints can be described. The missing constraints will be listed here in the future.

Chapter 5

OpenAPI Specification v2.0 JSON Schema

This is the JSON Schema file for the OpenAPI Specification version 2.0. Download and install it via NPM.

5.1 Install via NPM

```
npm install --save swagger-schema-official
```

5.2 License

Apache-2.0

Chapter 6

OpenAPI 3.0.X JSON Schema

Here you can find the JSON Schema for validating OpenAPI definitions of versions 3.0.X.

As a reminder, the JSON Schema is not the source of truth for the Specification. In cases of conflicts between the Specification itself and the JSON Schema, the Specification wins. Also, some Specification constraints cannot be represented with the JSON Schema so it's highly recommended to employ other methods to ensure compliance.

The iteration version of the JSON Schema can be found in the `id` field. For example, the value of `id: https://spec.openapis.org/oas/3.0/schema/2019-04-02` means this iteration was created on April 2nd, 2019.

To submit improvements to the schema, modify the schema.yaml file only.

The TSC will then:

- Run tests on the updated schema
- Update the iteration version
- Convert the schema.yaml to schema.json
- Publish the new version

Chapter 7

OpenAPI 3.1.X JSON Schema

Here you can find the JSON Schema for validating OpenAPI definitions of versions 3.1.X.

As a reminder, the JSON Schema is not the source of truth for the Specification. In cases of conflicts between the Specification itself and the JSON Schema, the Specification wins. Also, some Specification constraints cannot be represented with the JSON Schema so it's highly recommended to employ other methods to ensure compliance.

The iteration version of the JSON Schema can be found in the `$id` field. For example, the value of `$id: https://spec.openapis.org/oas/3.1/schema/2021-03-02` means this iteration was created on March 2nd, 2021.

The `schema.yaml` schema doesn't validate the JSON Schemas in your OpenAPI document because 3.1 allows you to use any JSON Schema dialect you choose. We have also included `schema-base.yaml` that extends the main schema to validate that all schemas use the default OAS base vocabulary.

7.1 Contributing

To submit improvements to the schema, modify the schema.yaml file only.

The TSC will then:

- Run tests on the updated schema
- Update the iteration version
- Convert the schema.yaml to schema.json
- Publish the new version

7.2 Tests

The test suite is included as a git submodule of `https://github.com/Mermade/openapi3-examples`.

```
npx mocha --recursive tests
```

You can also validate a document individually.

```
scripts/validate.js path/to/document/to/validate.yaml
```

Chapter 8

Change Log

All notable changes will be documented in this file. Swagger Methods adheres to `Semantic Versioning`.

8.1 <a href="https://github.com/APIDevTools/swagger-methods/tree/v3.0.0" >v3.0.0</a> (2020-03-10)

- Moved Swagger Methods to the `@APIDevTools scope` on NPM
- The "swagger-methods" NPM package is now just a wrapper around the scoped "@apidevtools/swagger-methods" package

`Full Changelog`

8.2 <a href="https://github.com/APIDevTools/swagger-methods/tree/v2.0.0" >v2.0.0</a> (2019-06-11)

8.2.1 Breaking Changes

- Dropped support for Node 0.10 through 6.0.0
- Converted to ES6 syntax

`Full Changelog`

8.3 <a href="https://github.com/APIDevTools/swagger-methods/tree/v1.0.0" >v1.0.0</a> (2015-09-07)

Initial release

Chapter 9

Swagger Methods

9.0.0.1 HTTP methods that are supported by Swagger 2.0

[](LICENSE)

This is an array of lower-case HTTP method names that are supported by the Swagger 2.0 spec.

This module is tested against the Swagger 2.0 schema

9.1 Installation

Install using npm:

```
npm install @apidevtools/swagger-methods
```

9.2 Usage

```
var methods = require('@apidevtools/swagger-methods');
methods.forEach(function(method) {
  console.log(method);
});
// get
// put
// post
// delete
// options
// head
// patch
```

9.3 Contributing

I welcome any contributions, enhancements, and bug-fixes. Open an issue on GitHub and submit a pull request.

9.3.0.1 Building/Testing

To build/test the project locally on your computer:

1. **Clone this repo**
   `git clone https://github.com/APIDevTools/swagger-methods.git`
2. **Install dev dependencies**
   `npm install`
3. **Run the unit tests**
   `npm test`

9.4 License

[MIT license](LICENSE). Use it however you want.

This package is `Treeware`. If you use it in production, then we ask that you **`buy the world a tree`** to thank us for our work. By contributing to the Treeware forest you'll be creating employment for local families and restoring wildlife habitats.

9.5 Big Thanks To

Thanks to these awesome companies for their support of Open Source developers

Chapter 10

Change Log

All notable changes will be documented in this file. Swagger Parser adheres to `Semantic Versioning`.

10.1 <a href="https://github.com/APIDevTools/swagger-parser/tree/v10.0.0" >v10.0.0</a> (2020-07-10)

10.1.0.1 Breaking Changes

- Removed the `YAML` export. We recommend using `@stoplight/yaml` instead

10.1.0.2 Other Changes

- Added a new `continueOnError option` that allows you to get all errors rather than just the first one

`Full Changelog`

10.2 <a href="https://github.com/APIDevTools/swagger-parser/tree/v9.0.0" >v9.0.0</a> (2020-03-14)

- Moved Swagger Parser to the `@APIDevTools scope` on NPM
- The "swagger-parser" NPM package is now just a wrapper around the scoped "@apidevtools/swagger-parser" package

`Full Changelog`

10.3 <a href="https://github.com/APIDevTools/swagger-parser/tree/v8.0.0">v8.0.0</a> (2019-06-22)

10.3.0.1 Potentially Breaking Changes

- The validate() function now uses the official JSON Schemas for Swagger 2.0 and OpenAPI 3.0 schema validation. We tested this change on over 1,500 real-world APIs and there were **no breaking changes**, but we're bumped the major version number just to be safe.

Full Changelog

10.4 <a href="https://github.com/APIDevTools/swagger-parser/tree/v7.0.0">v7.0.0</a> (2019-06-12)

10.4.0.1 Breaking Changes

- Dropped support for Node 6
- Updated all code to ES6+ syntax (async/await, template literals, arrow functions, etc.)
- No longer including a pre-built bundle in the package. such as Webpack, Rollup, Parcel, or Browserify to include Swagger Parser in your app

10.4.0.2 Other Changes

- Added TypeScript definitions

Full Changelog

10.5 <a href="https://github.com/APIDevTools/swagger-parser/tree/v6.0.0">v6.0.0</a> (2018-10-05)

- Dropped support for Bower, since it has been deprecated
- Removed the debug dependency

Full Changelog

10.6 <a href="https://github.com/APIDevTools/swagger-parser/tree/v5.0.0" >v5.0.0</a> (2018-05-25)

- After `months` and `months` of delays, initial support for OpenAPI 3.0 is finally here! A big "Thank You!" to `Leo Long` for doing the work and submitting `PR #88`.

`Full Changelog`

10.7 <a href="https://github.com/APIDevTools/swagger-parser/tree/v4.1.0" >v4.1.0</a> (2018-04-11)

- `@marcelstoer` submitted `PR #83` and `PR #84`, both of which improve the `validate() method`. It will now detect when a JSON Schema in your API definition has `required` properties that don't exist.

`Full Changelog`

10.8 <a href="https://github.com/APIDevTools/swagger-parser/tree/v4.0.0" >v4.0.0</a> (2017-10-19)

10.8.0.1 Breaking Changes

- Update the `Swagger 2.0 JSON schema`, so it's possible that an API that previously passed validation may no longer pass due to changes in the Swagger schema
- To reduce the size of this library, it no longer includes polyfills for `Promises` and `TypedArrays`, which are natively supported in the latest versions of Node and web browsers. If you need to support older browsers (such as IE9), then just use `this Promise polyfill` and `this TypedArray polyfill`.

10.8.0.2 Minor Changes

- `PR #74` - Fixes `an edge-case bug` with the `validate()` method and `x-` vendor extensions

`Full Changelog`

10.9 <a href="https://github.com/APIDevTools/swagger-parser/tree/v4.↩ 0.0-beta.2" >v4.0.0-beta.2</a> (2016-04-25)

10.9.0.1 Just one small fix

Fixed `issue #13`. You can now pass a URL *and* an object to any method.

```
SwaggerParser.validate("http://example.com/my-schema.json", mySchemaObject, {})
```

> **NOTE:** As shown in the example above, you *must* also pass an options object (even an empty object will work), otherwise, the method signature looks like you're just passing a URL and options.

`Full Changelog`

10.10 <a href="https://github.com/APIDevTools/swagger-parser/tree/v4.0.0-beta.1" >v4.0.0-beta.1</a> (2016-04-10)

10.10.0.1 Plug-ins !!!

That's the major new feature in this version. Originally requested in `PR #8`, and refined a few times over the past few months, the plug-in API is now finalized and ready to use. You can now define your own `resolvers` and `parsers`.

10.10.0.2 Breaking Changes

The available `options have changed`, mostly due to the new plug-in API. There's not a one-to-one mapping of old options to new options, so you'll have to read the docs and determine which options you need to set. If any. The out-of-the-box configuration works for most people.

All of the `caching options have been removed`. Instead, files are now cached by default, and the entire cache is reset for each new parse operation. Caching options may come back in a future release, if there is enough demand for it. If you used the old caching options, please open an issue and explain your use-case and requirements. I need a better understanding of what caching functionality is actually needed by users.

10.10.0.3 Bug Fixes

Lots of little bug fixes, and a couple major bug fixes:

- `completely rewrote the bundling logic` to fix `issue #16`
- Added support for `root-level $refs`

`Full Changelog`

10.11 <a href="https://github.com/APIDevTools/swagger-parser/tree/v3.3.0" >v3.3.0</a> (2015-10-02)

Updated to the latest version of `the Official Swagger 2.0 Schema`. The schema `hadn't been updated for six months`, so Swagger Parser was missing `several recent changes`.

`Full Changelog`

10.12 <a href="https://github.com/APIDevTools/swagger-parser/tree/v3.2.0" >v3.2.0</a> (2015-10-01)

Swagger Parser now uses `call-me-maybe` to support `promises or callbacks`.

`Full Changelog`

10.13 <a href="https://github.com/APIDevTools/swagger-parser/tree/v3.1.0" >v3.1.0</a> (2015-09-28)

Fixed several bugs with circular references, particularly with the `validate()` method.

Added a new `$refs.circular option` to determine how circular references are handled. Options are fully-dereferencing them (default), throwing an error, or ignoring them.

`Full Changelog`

10.14 <a href="https://github.com/APIDevTools/swagger-parser/tree/v3.0.0" >v3.0.0</a> (2015-09-25)

This is a **complete rewrite** of Swagger Parser. Major changes include:

Swagger 2.0 Compliant
Previous versions of Swagger Parser were based on early drafts of Swagger 2.0, and were not compliant with `the final version of the spec`. Swagger Parser v3.0 is now compliant with the final spec as well as related specs, such as `JSON Reference` and `JSON Pointer`

All-New API
The old API only had a single method: `parse()`. But depending on which options you passed it, the method did *much* more than its name implied. The new API has separate methods to make things a bit more intuitive. The most commonly used will be `validate()`, `bundle()`, and `dereference()`. The `parse()` and `resolve()` methods are also available, but these are mostly just for internal use by the other methods.

Asynchronous API
The old API was "asynchronous", but it relied on global state, so it did not support multiple simultaneous operations. The new API is truly asynchronous and supports both `ES6 Promises` and Node-style callbacks.

New JSON Schema Validator
Swagger Parser has switched from `tv4` to `Z-Schema`, which is `faster` and performs `more accurate validation`. This means that some APIs that previously passed validation will now fail. But that's a *good* thing!

`Full Changelog`

Chapter 11

Swagger 2.0 and OpenAPI 3.0 parser/validator

[](LICENSE)

11.1 Features

- Parses Swagger specs in **JSON** or **YAML** format
- Validates against the `Swagger 2.0 schema` or `OpenAPI 3.0 Schema`
- `Resolves` all `$ref` pointers, including external files and URLs
- Can `bundle` all your Swagger files into a single file that only has *internal* `$ref` pointers
- Can `dereference` all `$ref` pointers, giving you a normal JavaScript object that's easy to work with
- **`Tested`** in Node.js and all modern web browsers on Mac, Windows, and Linux
- Tested on **`over 1,500 real-world APIs`** from Google, Microsoft, Facebook, Spotify, etc.
- Supports `circular references`, nested references, back-references, and cross-references
- Maintains object reference equality — `$ref` pointers to the same value always resolve to the same object instance

11.2 Related Projects

- `Swagger CLI`
- `Swagger Express Middleware`

11.3 Example

```
SwaggerParser.validate(myAPI, (err, api) => {
  if (err) {
    console.error(err);
  }
  else {
    console.log("API name:  %s, Version:  %s", api.info.title, api.info.version);
  }
});
```

Or use `async/await` or `Promise` syntax instead. The following example is the same as above:

```
try {
  let api = await SwaggerParser.validate(myAPI);
  console.log("API name:  %s, Version:  %s", api.info.title, api.info.version);
}
catch(err) {
  console.error(err);
}
```

For more detailed examples, please see the `API Documentation`

11.4 Installation

Install using `npm`:

```
npm install @apidevtools/swagger-parser
```

11.5 Usage

When using Swagger Parser in Node.js apps, you'll probably want to use **CommonJS** syntax:

```
const SwaggerParser = require("@apidevtools/swagger-parser");
```

When using a transpiler such as `Babel` or `TypeScript`, or a bundler such as `Webpack` or `Rollup`, you can use **ECMAScript modules** syntax instead:

```
import SwaggerParser from "@apidevtools/swagger-parser";
```

11.6 Browser support

Swagger Parser supports recent versions of every major web browser. Older browsers may require `Babel` and/or `polyfills`.

To use Swagger Parser in a browser, you'll need to use a bundling tool such as `Webpack`, `Rollup`, `Parcel`, or `Browserify`. Some bundlers may require a bit of configuration, such as setting `browser: true` in `rollup-plugin-resolve`.

11.7 API Documentation

Full API documentation is available `right here`

11.8 Contributing

I welcome any contributions, enhancements, and bug-fixes. `Open an issue` on GitHub and `submit a pull request`.

11.8.0.1 Building/Testing

To build/test the project locally on your computer:

1. **Clone this repo**
   `git clone https://github.com/APIDevTools/swagger-parser.git`
2. **Install dependencies**
   `npm install`
3. **Run the build script**
   `npm run build`
4. **Run the tests**
   `npm test`

11.9 License

Swagger Parser is 100% free and open-source, under the [MIT license](LICENSE). Use it however you want.

This package is `Treeware`. If you use it in production, then we ask that you **`buy the world a tree`** to thank us for our work. By contributing to the Treeware forest you'll be creating employment for local families and restoring wildlife habitats.

11.10 Big Thanks To

Thanks to these awesome companies for their support of Open Source developers

Chapter 12

Change Log

All notable changes will be documented in this file. `ono` adheres to Semantic Versioning.

12.1 <a href="https://github.com/JS-DevTools/ono/tree/v7.1.0" >v7.1.0</a> (2020-03-03)

- Added a static `Ono.extend()` method that allows Ono to extend errors that were created outside of Ono. The extended error gets all the Ono functionality, including nested stack traces, custom properties, improved support for `JSON.stringify()`, etc.

Full Changelog

12.2 <a href="https://github.com/JS-DevTools/ono/tree/v7.0.0" >v7.0.0</a> (2020-02-16)

- Moved Ono to the @JSDevTools scope on NPM
- The "ono" NPM package is now just a wrapper around the scoped "@jsdevtools/ono" package

Full Changelog

12.3 <a href="https://github.com/JS-DevTools/ono/tree/v6.0.0" >v6.0.0</a> (2019-12-28)

12.3.1 Breaking Changes

- Dropped support for IE8 and other JavaScript engines that don't support `Object.getOwnPropertyDescriptor()`
- Removed `ono.formatter`. It has been replaced with the format option
- When using the default `ono()` function to wrap an error, it will now try to match the error's type, rather than simply using the base `Error` class.

12.3.2 New Features

- The `Ono constructor` now accepts an optional `options parameter`, which lets you customize the behavior of Ono
- The `concatMessages option` lets you control whether the original error's message is appended to your error message
- The `format option` lets you provide a custom function for replacing placeholders in error messages

`Full Changelog`

12.4 <a href="https://github.com/JS-DevTools/ono/tree/v5.1.0" >v5.1.0</a> (2019-09-10)

- Added a static `Ono.toJSON()` method that accepts any `Error` (even a non-Ono error) and returns a POJO that can be used with `JSON.stringify()`. Ono errors already have a built-in `toJSON()` method, but this exposes that enhanced functionality in a way that can be used with *any* error.

`Full Changelog`

12.5 <a href="https://github.com/JS-DevTools/ono/tree/v5.0.0" >v5.0.0</a> (2019-02-18)

12.5.1 Breaking Changes

12.5.1.1 in Node.js

- Ono errors previously included an `inspect()` method to support Node's `util.inspect() function`. As of Node v6.6.0, the `inspect()` method is deprecated in favor of a `util.inspect.↩ custom`. Ono has updated accordingly, so errors no longer have an `inspect()` method.

12.5.1.2 in Web Browsers

- We no longer automatically include a polyfill for `Node's util.format() function`. We recommend using `ES6 template strings` instead. Or you can import `a polyfill` yourself and assign it to `the ono.formatter property`.

12.5.2 New Features

- Completely rewritten in TypeScript.
- Ono is now completely dependency free.
- You can now create your own Ono functions for custom error classes. See `the docs` for details.
- Symbol-keyed properties are now supported. If the `originalError` and/or `props` objects has Symbol-keyed properties, they will be copied to the Ono error.

`Full Changelog`

12.6 <a href="https://github.com/JS-DevTools/ono/tree/v4.0.0" >v4.0.0</a> (2017-07-07)

The `err` parameter (see `the API docs`) can now be any type of object, not just an `instanceof Error`. This allows for errors that don't extend from the `Error` class, such as `DOMError`, `DOMException`, and custom error types.

> **NOTE:** This should **not** be a breaking change, but I'm bumping the major version number out of an abundance of caution.

`Full Changelog`

12.7 <a href="https://github.com/JS-DevTools/ono/tree/v3.1.0" >v3.1.0</a> (2017-06-01)

We removed the direct dependency on `Node's util.format()`, which was needlessly bloating the browser bundle. Instead, I now import `format-util`, which a much more `lightweight browser implementation`. There's no change when running in Node.js, because `format-util` simply `exports util.format()`.

`Full Changelog`

12.8 <a href="https://github.com/JS-DevTools/ono/tree/v3.0.0" >v3.0.0</a> (2017-06-01)

- Updated all dependencies and verified support for Node 8.0
- Ono no longer appears in error stack traces, so errors look like they came directly from your code

`Full Changelog`

12.9 <a href="https://github.com/JS-DevTools/ono/tree/v2.0.0" >v2.0.0</a> (2015-12-14)

- Did a major refactoring and code cleanup
- Support for various browser-specific `Error.prototype` properties (`fileName`, `lineNumber`, `sourceURL`, etc.)
- If you define a custom `toJSON()` method on an error object, Ono will no longer overwrite it
- Added support for Node's `util.inspect()`

`Full Changelog`

Chapter 13

ono (Oh No!)

13.0.1 Throw better errors.

[](LICENSE)

13.1 Features

- Wrap and re-throw an error *without* losing the original error's type, message, stack trace, and properties
- Add custom properties to errors — great for error numbers, status codes, etc.
- Use format strings for error messages — great for localization
- Enhanced support for `JSON.stringify()` and `util.inspect()` — great for logging
- Supports and enhances your own custom error classes
- Tested on Node.js and all modern web browsers on Mac, Windows, and Linux.

13.2 Example

```
const ono = require("@jsdevtools/ono");
// Throw an error with custom properties
throw ono({ code:  "NOT_FOUND", status:  404 }, `Resource not found:  ${url}`);
// Wrap an error without losing the original error’s stack and props
throw ono(originalError, "An error occurred while saving your changes");
// Wrap an error and add custom properties
throw ono(originalError, { code:  404, status:  "NOT_FOUND" });
// Wrap an error, add custom properties, and change the error message
throw ono(originalError, { code:  404, status:  "NOT_FOUND" }, `Resource not found:  ${url}`);
// Throw a specific Error subtype instead
// (works with any of the above signatures)
throw ono.range(...);                           // RangeError
throw ono.syntax(...);                          // SyntaxError
throw ono.reference(...);                       // ReferenceError
// Create an Ono method for your own custom error class
const { Ono } = require("@jsdevtools/ono");
class MyErrorClass extends Error {}
ono.myError = new Ono(MyErrorClass);
// And use it just like any other Ono method
throw ono.myError(...);                         // MyErrorClass
```

13.3 Installation

Install using npm:

```
npm install @jsdevtools/ono
```

13.4 Usage

When using Ono in Node.js apps, you'll probably want to use **CommonJS** syntax:

```
const ono = require("@jsdevtools/ono");
```

When using a transpiler such as Babel or TypeScript, or a bundler such as Webpack or Rollup, you can use **ECMAScript modules** syntax instead:

```
import ono from "@jsdevtools/ono";
```

13.5 Browser support

Ono supports recent versions of every major web browser. Older browsers may require Babel and/or polyfills.

To use Ono in a browser, you'll need to use a bundling tool such as Webpack, Rollup, Parcel, or Browserify. Some bundlers may require a bit of configuration, such as setting `browser: true` in rollup-plugin-resolve.

13.6 API

13.6.1 <tt>ono([originalError], [props], [message, ...])</tt>

Creates an Error object with the given properties.

- `originalError` - _(optional)_ The original error that occurred, if any. This error's message, stack trace, and properties will be copied to the new error. If this error's type is one of the known error types, then the new error will be of the same type.
- `props` - _(optional)_ An object whose properties will be copied to the new error. Properties can be anything, including objects and functions.
- `message` - _(optional)_ The error message string. If it contains placeholders, then pass each placeholder's value as an additional parameter. See the `format` option for more info.

13.6.2 Specific error types

The default `ono()` function may return an instance of the base Error class, or it may return a more specific sub-class, based on the type of the `originalError` argument. If you want to *explicitly* create a specific type of error, then you can use any of the following methods:

The method signatures and arguments are exactly the same as the default `ono()` function.

Method	Return Type
`ono.error()`	`Error`
`ono.eval()`	`EvalError`
`ono.range()`	`RangeError`
`ono.reference()`	`ReferenceError`
`ono.syntax()`	`SyntaxError`
`ono.type()`	`TypeError`
`ono.uri()`	`URIError`
`ono.yourCustomErrorHere()`	Add your own custom error classes to ono

13.6.3 <tt>Ono(Error, [options])</tt>

The `Ono` constructor is used to create your own custom \`ono\` methods for custom error types, or to change the default behavior of the built-in methods.

Warning: Be sure not to confuse `ono` (lowercase) and `Ono` (capitalized). The latter one is a class.

- `Error` - The Error sub-class that this Ono method will create instances of
- `options` - _(optional)_ An options object, which customizes the behavior of the Ono method

13.7 Options

The `Ono` constructor takes an optional options object as a second parameter. The object can have the following properties, all of which are optional:

Option	Type	Default	Description
`concatMessages`	boolean	`true`	When Ono is used to wrap an error, this setting determines whether the inner error's message is appended to the new error message.
`format`	function or boolean	`util.format()` in Node.js<br> `false` in web browsers	A function that replaces placeholders like in error messages with values. If set to `false`, then error messages will be treated as literals and no placeholder replacement will occur.

13.7.1 <tt>concatMessages</tt> Option

When wrapping an error, Ono's default behavior is to append the error's message to your message, with a newline between them. For example:

```
const ono = require("@jsdevtools/ono");
function createArray(length) {
  try {
    return new Array(length);
  }
  catch (error) {
    // Wrap and re-throw the error
    throw ono(error, "Sorry, I was unable to create the array.");
  }
}
// Try to create an array with a negative length
createArray(-5);
```

The above code produces the following error message:

```
Sorry, I was unable to create the array.
Invalid array length;
```

If you'd rather not include the original message, then you can set the `concatMessages` option to `false`. For example:

```
const { ono, Ono } = require("@jsdevtools/ono");
// Override the default behavior for the RangeError
ono.range = new Ono(RangeError, { concatMessages:  false });
function createArray(length) {
  try {
    return new Array(length);
  }
  catch (error) {
    // Wrap and re-throw the error
    throw ono(error, "Sorry, I was unable to create the array.");
  }
}
// Try to create an array with a negative length
createArray(-5);
```

Now the error only includes your message, not the original error message.

```
Sorry, I was unable to create the array.
```

13.7.2 <tt>format</tt> option

The `format` option let you set a format function, which replaces placeholders in error messages with values.

When running in Node.js, Ono uses `the util.format() function` by default, which lets you use placeholders such as s, d, and j. You can provide the values for these placeholders when calling any Ono method:

```
throw ono("%s is invalid.  Must be at least %d characters.", username, minLength);
```

Of course, the above example could be accomplished using `ES6 template literals` instead of format strings:

```
throw ono(`${username} is invalid.  Must be at least ${minLength} characters.`);
```

Format strings are most useful when you don't alrady know the values at the time that you're writing the string. A common scenario is localization. Here's a simplistic example:

```
const errorMessages {
  invalidLength:  {
    en:  "%s is invalid.  Must be at least %d characters.",
    es:  "%s no es válido.  Debe tener al menos %d caracteres.",
    zh:  "%s  %d",
  }
}
let lang = getCurrentUsersLanguage();
throw ono(errorMessages.invalidLength[lang], username, minLength);
```

13.7.2.1 The <tt>format</tt> option in web browsers

Web browsers don't have a built-in equivalent of Node's `util.format() function`, so format strings are only supported in Node.js by default. However, you can set the `format` option to any compatible polyfill library to enable this functionality in web browsers too.

Here are some compatible polyfill libraries:

- `format`
- `format-util`

13.7.2.2 Custom <tt>format</tt> implementation

If the standard `util.format()` functionality isn't sufficient for your needs, then you can set the `format` option to your own custom implementation. Here's a simplistic example:

```
const { ono, Ono } = require("@jsdevtools/ono");
// This is a simple formatter that replaces $0, $1, $2, ... with the corresponding argument
let options = {
  format(message, ...args) {
    for (let [index, arg] of args.entries()) {
      message = message.replace("$" + index, arg);
    }
    return message;
  }
};
// Use your custom formatter for all of the built-in error types
ono.error = new Ono(Error, options);
ono.eval = new Ono(EvalError, options);
ono.range = new Ono(RangeError, options);
ono.reference = new Ono(ReferenceError, options);
ono.syntax = new Ono(SyntaxError, options);
ono.type = new Ono(TypeError, options);
ono.uri = new Ono(URIError, options);
// Now all Ono functions support your custom formatter
throw ono("$0 is invalid.  Must be at least $1 characters.", username, minLength);
```

13.8 Custom Error Classes

There are two ways to use Ono with your own custom error classes. Which one you choose depends on what parameters your custom error class accepts, and whether you'd prefer to use `ono.myError()` syntax or `new MyError()` syntax.

13.8.1 Option 1: Standard Errors

Ono has built-in support for all of `the built-in JavaScript Error types`. For example, you can use `ono.reference()` to create a `ReferenceError`, or `ono.syntax()` to create a `SyntaxError`.

All of these built-in JavaScript Error types accept a single parameter: the error message string. If your own error classes also work this way, then you can create Ono methods for your custom error classes. Here's an example:

```
const { ono, Ono } = require("@jsdevtools/ono");
let counter = 0;
// A custom Error class that assigns a unique ID and timestamp to each error
class MyErrorClass extends Error {
  constructor(message) {
    super(message);
    this.id = ++counter;
    this.timestamp = new Date();
  }
}
// Create a new Ono method for your custom Error class
ono.myError = new Ono(MyErrorClass);
// You can use this method just like any other Ono method
throw ono.myError({ code:  404, status:  "NOT_FOUND" }, `Resource not found:  ${url}`);
```

The code above throws an instance of `MyErrorClass` that looks like this:

```
{
  "name":  "MyErrorClass",
  "message":  "Resource not found:  xyz.html",
  "id":  1,
  "timestamp":  "2019-01-01T12:30:00.456Z",
  "code":  404,
  "status":  "NOT_FOUND",
  "stack":  "MyErrorClass:  Resource not found:  xyz.html\n   at someFunction (index.js:24:5)",
}
```

13.8.2 Option 2: Enhanced Error Classes

If your custom error classes require more than just an error message string parameter, then you'll need to use Ono differently. Rather than creating a custom Ono method and using `ono.myError()` syntax, you'll use Ono *inside* your error class's constructor. This has a few benefits:

- Your error class can accept whatever parameters you want
- Ono is encapsulated within your error class
- You can use `new MyError()` syntax rather than `ono.myError()` syntax

```
const { ono, Ono } = require("@jsdevtools/ono");
// A custom Error class for 404 Not Found
class NotFoundError extends Error {
  constructor(method, url) {
    super(`404:  ${method} ${url} was not found`);
    // Add custom properties, enhance JSON.stringify() support, etc.
    Ono.extend(this, { statusCode:  404, method, url });
  }
}
// A custom Error class for 500 Server Error
class ServerError extends Error {
  constructor(originalError, method, url) {
    super(`500:  A server error occurred while responding to ${method} ${url}`);
    // Append the stack trace and custom properties of the original error,
    // and add new custom properties, enhance JSON.stringify() support, etc.
    Ono.extend(this, originalError, { statusCode:  500, method, url });
  }
}
```

13.9 Contributing

Contributions, enhancements, and bug-fixes are welcome! `Open an issue` on GitHub and `submit a pull request`.

13.9.0.1 Building/Testing

To build/test the project locally on your computer:

1. **Clone this repo**
   `git clone https://github.com/JS-DevTools/ono.git`
2. **Install dependencies**
   `npm install`
3. **Run the build script**
   `npm run build`
4. **Run the tests**
   `npm test`

13.10 License

Ono is 100% free and open-source, under the [MIT license](LICENSE). Use it however you want.

This package is `Treeware`. If you use it in production, then we ask that you **`buy the world a tree`** to thank us for our work. By contributing to the Treeware forest you'll be creating employment for local families and restoring wildlife habitats.

13.11 Big Thanks To

Thanks to these awesome companies for their support of Open Source developers

Chapter 14

Installation

```
npm install --save @types/geojson
```

14.1 Summary

This package contains type definitions for geojson (`https://geojson.org/`).

14.2 Details

Files were exported from `https://github.com/DefinitelyTyped/DefinitelyTyped/tree/master/types/g`

14.2.0.1 Additional Details

- Last updated: Tue, 30 Jan 2024 21:35:45 GMT
- Dependencies: none

14.3 Credits

These definitions were written by `Jacob Bruun`, `Arne Schubert`, `Jeff Jacobson`, `Ilia Choly`, and `Dan Vanderkam`.

Chapter 15

Installation

```
npm install --save @types/json-schema
```

15.1 Summary

This package contains type definitions for json-schema (https://github.com/kriszyp/json-schema).

15.2 Details

Files were exported from https://github.com/DefinitelyTyped/DefinitelyTyped/tree/master/types/j

15.2.0.1 Additional Details

- Last updated: Tue, 07 Nov 2023 03:09:37 GMT
- Dependencies: none

15.3 Credits

These definitions were written by Boris Cherny, Lucian Buzzo, Roland Groza, and Jason Kwok.

Chapter 16

Installation

```
npm install --save @types/node
```

16.1 Summary

This package contains type definitions for Node.js (`https://nodejs.org/`).

16.2 Details

Files were exported from `https://github.com/DefinitelyTyped/DefinitelyTyped/tree/master/types/n`

16.2.0.1 Additional Details

- Last updated: Wed, 15 Jun 2022 23:01:35 GMT
- Dependencies: none
- Global values: `AbortController`, `AbortSignal`, `__dirname`, `__filename`, `console`, `exports`, `gc`, `global`, `module`, `process`, `require`, `structuredClone`

16.3 Credits

These definitions were written by `Microsoft TypeScript`, `DefinitelyTyped`, `Alberto Schiabel`, `Alvis HT Tang`, `Andrew Makarov`, `Benjamin Toueg`, `Chigozirim C.`, `David Junger`, `Deividas Bakanas`, `Eugene Y. Q. Shen`, `Hannes Magnusson`, `Huw`, `Kelvin Jin`, `Klaus Meinhardt`, `Lishude`, `Mariusz Wiktorczyk`, `Mohsen Azimi`, `Nicolas Even`, `Nikita Galkin`, `Parambir Singh`, `Sebastian Silbermann`, `Simon Schick`, `Thomas den Hollander`, `Wilco Bakker`, `wwwy3y3`, `Samuel Ainsworth`, `Kyle Uehlein`, `Thanik Bhongbhibhat`, `Marcin Kopacz`, `Trivikram Kamat`, `Junxiao Shi`, `Ilia Baryshnikov`, `ExE Boss`, `Piotr Błażejewicz`, `Anna Henningsen`, `Victor Perin`, `Yongsheng Zhang`, `NodeJS Contributors`, `Linus Unnebäck`, and `wafuwafu13`.

Chapter 17

1.3.8 / 2022-02-02

- deps: mime-types2.1.34
  - deps: mime-db1.51.0
- deps: negotiator@0.6.3

17.1 1.3.7 / 2019-04-29

- deps: negotiator@0.6.2
  - Fix sorting charset, encoding, and language with extra parameters

17.2 1.3.6 / 2019-04-28

- deps: mime-types2.1.24
  - deps: mime-db1.40.0

17.3 1.3.5 / 2018-02-28

- deps: mime-types2.1.18
  - deps: mime-db1.33.0

17.4 1.3.4 / 2017-08-22

- deps: mime-types2.1.16
  - deps: mime-db1.29.0

17.5 1.3.3 / 2016-05-02

- deps: mime-types2.1.11
  - deps: mime-db1.23.0
- deps: negotiator@0.6.1
  - perf: improve `Accept` parsing speed
  - perf: improve `Accept-Charset` parsing speed
  - perf: improve `Accept-Encoding` parsing speed
  - perf: improve `Accept-Language` parsing speed

17.6 1.3.2 / 2016-03-08

- deps: mime-types2.1.10
  - Fix extension of `application/dash+xml`
  - Update primary extension for `audio/mp4`
  - deps: mime-db1.22.0

17.7 1.3.1 / 2016-01-19

- deps: mime-types2.1.9
  - deps: mime-db1.21.0

17.8 1.3.0 / 2015-09-29

- deps: mime-types2.1.7
  - deps: mime-db1.19.0
- deps: negotiator@0.6.0
  - Fix including type extensions in parameters in `Accept` parsing
  - Fix parsing `Accept` parameters with quoted equals
  - Fix parsing `Accept` parameters with quoted semicolons
  - Lazy-load modules from main entry point
  - perf: delay type concatenation until needed
  - perf: enable strict mode
  - perf: hoist regular expressions
  - perf: remove closures getting spec properties
  - perf: remove a closure from media type parsing
  - perf: remove property delete from media type parsing

17.9 1.2.13 / 2015-09-06

- deps: mime-types2.1.6
  - deps: mime-db1.18.0

17.10 1.2.12 / 2015-07-30

- deps: mime-types2.1.4
  - deps: mime-db1.16.0

17.11 1.2.11 / 2015-07-16

- deps: mime-types2.1.3
  - deps: mime-db1.15.0

17.12 1.2.10 / 2015-07-01

- deps: mime-types2.1.2
  - deps: mime-db1.14.0

17.13 1.2.9 / 2015-06-08

- deps: mime-types2.1.1
  - perf: fix deopt during mapping

17.14 1.2.8 / 2015-06-07

- deps: mime-types2.1.0
  - deps: mime-db1.13.0
- perf: avoid argument reassignment & argument slice
- perf: avoid negotiator recursive construction
- perf: enable strict mode
- perf: remove unnecessary bitwise operator

17.15 1.2.7 / 2015-05-10

- deps: negotiator@0.5.3
  - Fix media type parameter matching to be case-insensitive

17.16 1.2.6 / 2015-05-07

- deps: mime-types2.0.11
  - deps: mime-db1.9.1
- deps: negotiator@0.5.2
  - Fix comparing media types with quoted values
  - Fix splitting media types with quoted commas

17.17 1.2.5 / 2015-03-13

- deps: mime-types2.0.10
  - deps: mime-db1.8.0

17.18 1.2.4 / 2015-02-14

- Support Node.js 0.6
- deps: mime-types2.0.9
  - deps: mime-db1.7.0
- deps: negotiator@0.5.1
  - Fix preference sorting to be stable for long acceptable lists

17.19 1.2.3 / 2015-01-31

- deps: mime-types2.0.8
  - deps: mime-db1.6.0

17.20 1.2.2 / 2014-12-30

- deps: mime-types2.0.7
  - deps: mime-db1.5.0

17.21 1.2.1 / 2014-12-30

- deps: mime-types2.0.5
  - deps: mime-db1.3.1

17.22 1.2.0 / 2014-12-19

- deps: negotiator@0.5.0
  - Fix list return order when large accepted list
  - Fix missing identity encoding when q=0 exists
  - Remove dynamic building of Negotiator class

17.23 1.1.4 / 2014-12-10

- deps: mime-types2.0.4
  - deps: mime-db1.3.0

17.24 1.1.3 / 2014-11-09

- deps: mime-types2.0.3
  - deps: mime-db1.2.0

17.25 1.1.2 / 2014-10-14

- deps: negotiator@0.4.9
  - Fix error when media type has invalid parameter

17.26 1.1.1 / 2014-09-28

- deps: mime-types2.0.2
  - deps: mime-db1.1.0
- deps: negotiator@0.4.8
  - Fix all negotiations to be case-insensitive
  - Stable sort preferences of same quality according to client order

17.27 1.1.0 / 2014-09-02

- update `mime-types`

17.28 1.0.7 / 2014-07-04

- Fix wrong type returned from `type` when match after unknown extension

17.29 1.0.6 / 2014-06-24

- deps: negotiator@0.4.7

17.30 1.0.5 / 2014-06-20

- fix crash when unknown extension given

17.31 1.0.4 / 2014-06-19

- use `mime-types`

17.32 1.0.3 / 2014-06-11

- deps: negotiator@0.4.6
  - Order by specificity when quality is the same

17.33 1.0.2 / 2014-05-29

- Fix interpretation when header not in request
- deps: pin negotiator@0.4.5

17.34 1.0.1 / 2014-01-18

- Identity encoding isn't always acceptable
- deps: negotiator0.4.0

17.35 1.0.0 / 2013-12-27

- Genesis

Chapter 18

accepts

Higher level content negotiation based on negotiator. Extracted from koa for general use.

In addition to negotiator, it allows:

- Allows types as an array or arguments list, ie ‘(['text/html’, 'application/json'])as well as('text/html', 'application/json').
- Allows type shorthands such asjson.
- Returnsfalsewhen no types match
- Treats non-existent headers as*`

18.1 Installation

This is a Node.js module available through the npm registry. Installation is done using the npm install command:

```
$ npm install accepts
```

18.2 API

```
var accepts = require('accepts')
```

18.2.1 accepts(req)

Create a new `Accepts` object for the given `req`.

18.2.1.1 .charset(charsets)

Return the first accepted charset. If nothing in `charsets` is accepted, then `false` is returned.

18.2.1.2 .charsets()

Return the charsets that the request accepts, in the order of the client's preference (most preferred first).

18.2.1.3 .encoding(encodings)

Return the first accepted encoding. If nothing in `encodings` is accepted, then `false` is returned.

18.2.1.4 .encodings()

Return the encodings that the request accepts, in the order of the client's preference (most preferred first).

18.2.1.5 .language(languages)

Return the first accepted language. If nothing in `languages` is accepted, then `false` is returned.

18.2.1.6 .languages()

Return the languages that the request accepts, in the order of the client's preference (most preferred first).

18.2.1.7 .type(types)

Return the first accepted type (and it is returned as the same text as what appears in the `types` array). If nothing in `types` is accepted, then `false` is returned.

The `types` array can contain full MIME types or file extensions. Any value that is not a full MIME types is passed to ‘require('mime-types’).lookup`.

18.2.1.8 .types()

Return the types that the request accepts, in the order of the client's preference (most preferred first).

18.3 Examples

18.3.1 Simple type negotiation

This simple example shows how to use `accepts` to return a different typed respond body based on what the client wants to accept. The server lists it's preferences in order and will get back the best match between the client and server.

```
var accepts = require('accepts')
var http = require('http')
function app (req, res) {
  var accept = accepts(req)
  // the order of this list is significant; should be server preferred order
  switch (accept.type(['json', 'html'])) {
    case 'json':
      res.setHeader('Content-Type', 'application/json')
      res.write('{"hello":"world!"}')
      break
    case 'html':
      res.setHeader('Content-Type', 'text/html')
      res.write('<b>hello, world!</b>')
      break
    default:
      // the fallback is text/plain, so no need to specify it above
      res.setHeader('Content-Type', 'text/plain')
      res.write('hello, world!')
      break
  }
  res.end()
}
http.createServer(app).listen(3000)
```

You can test this out with the cURL program:

```
curl -I -H'Accept:  text/html' http://localhost:3000/
```

18.4 License

[MIT](LICENSE)

Chapter 19

Changelog

All notable changes to this project will be documented in this file.

The format is based on `Keep a Changelog`, and this project adheres to `Semantic Versioning`.

19.1 <a href="https://github.com/nodeca/argparse/compare/2.0.0...2.0.1" >2.0.1</a> - 2020-08-29

19.1.1 Fixed

- Fix issue with `process.argv` when used with interpreters (`coffee`, `ts-node`, etc.), #150.

19.2 <a href="https://github.com/nodeca/argparse/compare/1.0.10...2.0.0" >2.0.0</a> - 2020-08-14

19.2.1 Changed

- Full rewrite. Now port from python 3.9.0 & more precise following. See `doc` for difference and migration info.
- node.js 10+ required
- Removed most of local docs in favour of original ones.

19.3 <a href="https://github.com/nodeca/argparse/compare/1.0.9...1.0.10" >1.0.10</a> - 2018-02-15

19.3.1 Fixed

- Use .concat instead of + for arrays, #122.

19.4 <a href="https://github.com/nodeca/argparse/compare/1.0.8...1.0.9" >1.0.9</a> - 2016-09-29

19.4.1 Changed

- Rerelease after 1.0.8 - deps cleanup.

19.5 <a href="https://github.com/nodeca/argparse/compare/1.0.7...1.0.8" >1.0.8</a> - 2016-09-29

19.5.1 Changed

- Maintenance (deps bump, fix node 6.5+ tests, coverage report).

19.6 <a href="https://github.com/nodeca/argparse/compare/1.0.6...1.0.7" >1.0.7</a> - 2016-03-17

19.6.1 Changed

- Teach `addArgument` to accept string arg names. #97, @tomxtobin.

19.7 <a href="https://github.com/nodeca/argparse/compare/1.0.5...1.0.6" >1.0.6</a> - 2016-02-06

19.7.1 Changed

- Maintenance: moved to eslint & updated CS.

19.8 <a href="https://github.com/nodeca/argparse/compare/1.0.4...1.0.5" >1.0.5</a> - 2016-02-05

19.8.1 Changed

- Removed lodash dependency to significantly reduce install size. Thanks to @mourner.

19.9 <a href="https://github.com/nodeca/argparse/compare/1.0.3...1.0.4" >1.0.4</a> - 2016-01-17

19.9.1 Changed

- Maintenance: lodash update to 4.0.0.

19.10 <a href="https://github.com/nodeca/argparse/compare/1.0.2...1.0.3" >1.0.3</a> - 2015-10-27

19.10.1 Fixed

- Fix parse = in args: `--examplepath="C:\myfolder\env=x64"`. #84, @CatWithApple.

19.11 <a href="https://github.com/nodeca/argparse/compare/1.0.1...1.0.2" >1.0.2</a> - 2015-03-22

19.11.1 Changed

- Relaxed lodash version dependency.

19.12 <a href="https://github.com/nodeca/argparse/compare/1.0.0...1.0.1" >1.0.1</a> - 2015-02-20

19.12.1 Changed

- Changed dependencies to be compatible with ancient nodejs.

19.13 <a href="https://github.com/nodeca/argparse/compare/0.1.16...1.0.0" >1.0.0</a> - 2015-02-19

19.13.1 Changed

- Maintenance release.
- Replaced `underscore` with `lodash`.
- Bumped version to 1.0.0 to better reflect semver meaning.
- HISTORY.md -> CHANGELOG.md

19.14 <a href="https://github.com/nodeca/argparse/compare/0.1.15...0.1.16" >0.1.16</a> - 2013-12-01

19.14.1 Changed

- Maintenance release. Updated dependencies and docs.

19.15 <a href="https://github.com/nodeca/argparse/compare/0.1.14...0.1.15" >0.1.15</a> - 2013-05-13

19.15.1 Fixed

- Fixed #55, @trebor89

19.16 <a href="https://github.com/nodeca/argparse/compare/0.1.13...0.1.14" >0.1.14</a> - 2013-05-12

19.16.1 Fixed

- Fixed #62, @maxtaco

19.17 <a href="https://github.com/nodeca/argparse/compare/0.1.12...0.1.13" >0.1.13</a> - 2013-04-08

19.17.1 Changed

- Added `.npmignore` to reduce package size

19.18 <a href="https://github.com/nodeca/argparse/compare/0.1.11...0.1.12" >0.1.12</a> - 2013-02-10

19.18.1 Fixed

- Fixed conflictHandler (#46), @hpaulj

19.19 <a href="https://github.com/nodeca/argparse/compare/0.1.10...0.1.11" >0.1.11</a> - 2013-02-07

19.19.1 Added

- Added 70+ tests (ported from python), @hpaulj
- Added conflictHandler, @applepicke
- Added fromfilePrefixChar, @hpaulj

19.19.2 Fixed

- Multiple bugfixes, @hpaulj

19.20 <a href="https://github.com/nodeca/argparse/compare/0.1.9...0.1.10" >0.1.10</a> - 2012-12-30

19.20.1 Added

- Added `mutual exclusion` support, thanks to @hpaulj

19.20.2 Fixed

- Fixed options check for `storeConst` & `appendConst` actions, thanks to @hpaulj

19.21 <a href="https://github.com/nodeca/argparse/compare/0.1.8...0.1.9" >0.1.9</a> - 2012-12-27

19.21.1 Fixed

- Fixed option dest interferens with other options (issue #23), thanks to @hpaulj
- Fixed default value behavior with $*$ positionals, thanks to @hpaulj
- Improve `getDefault()` behavior, thanks to @hpaulj
- Improve negative argument parsing, thanks to @hpaulj

19.22 <a href="https://github.com/nodeca/argparse/compare/0.1.7...0.1.8" >0.1.8</a> - 2012-12-01

19.22.1 Fixed

- Fixed parser parents (issue #19), thanks to @hpaulj
- Fixed negative argument parse (issue #20), thanks to @hpaulj

19.23 <a href="https://github.com/nodeca/argparse/compare/0.1.6...0.1.7" >0.1.7</a> - 2012-10-14

19.23.1 Fixed

- Fixed 'choices' argument parse (issue #16)
- Fixed stderr output (issue #15)

19.24 <a href="https://github.com/nodeca/argparse/compare/0.1.5...0.1.6" >0.1.6</a> - 2012-09-09

19.24.1 Fixed

- Fixed check for conflict of options (thanks to @tomxtobin)

19.25 <a href="https://github.com/nodeca/argparse/compare/0.1.4...0.1.5" >0.1.5</a> - 2012-09-03

19.25.1 Fixed

- Fix parser #setDefaults method (thanks to @tomxtobin)

19.26 <a href="https://github.com/nodeca/argparse/compare/0.1.3...0.1.4" >0.1.4</a> - 2012-07-30

19.26.1 Fixed

- Fixed pseudo-argument support (thanks to @CGamesPlay)
- Fixed addHelp default (should be true), if not set (thanks to @benblank)

19.27 <a href="https://github.com/nodeca/argparse/compare/0.1.2...0.1.3" >0.1.3</a> - 2012-06-27

19.27.1 Fixed

- Fixed formatter api name: Formatter -> HelpFormatter

19.28 <a href="https://github.com/nodeca/argparse/compare/0.1.1...0.1.2" >0.1.2</a> - 2012-05-29

19.28.1 Fixed

- Removed excess whitespace in help
- Fixed error reporting, when parcer with subcommands called with empty arguments

19.28.2 Added

- Added basic tests

19.29 <a href="https://github.com/nodeca/argparse/compare/0.1.0...0.1.1" >0.1.1</a> - 2012-05-23

19.29.1 Fixed

- Fixed line wrapping in help formatter
- Added better error reporting on invalid arguments

19.30 <a href="https://github.com/nodeca/argparse/releases/tag/0.1.0" >0.1.0</a> - 2012-05-16

19.30.1 Added

- First release.

Chapter 20

argparse

CLI arguments parser for node.js, with `sub-commands` support. Port of python's `argparse` (version `3.9.0`).

Difference with original.

- JS has no keyword arguments support.
  - Pass options instead: ‘new ArgumentParser({ description: 'example’, add_help: true })`.
- `JS has no python's types`int, float`, ...`
  - `Use string-typed names:.add_argument('-b', { type: 'int', help: 'help' }). -rformat specifier usesrequire('util').inspect()`.`

`More details in` `doc`.

20.1 Example

`test.js file:`

```
#!/usr/bin/env node
'use strict';
const { ArgumentParser } = require('argparse');
const { version } = require('./package.json');
const parser = new ArgumentParser({
  description: 'Argparse example'
});
parser.add_argument('-v', '--version', { action: 'version', version });
parser.add_argument('-f', '--foo', { help: 'foo bar' });
parser.add_argument('-b', '--bar', { help: 'bar foo' });
parser.add_argument('--baz', { help: 'baz bar' });
console.dir(parser.parse_args());
```

`Display help:`

```
$ ./test.js -h
usage: test.js [-h] [-v] [-f FOO] [-b BAR] [--baz BAZ]
Argparse example
optional arguments:
  -h, --help         show this help message and exit
  -v, --version      show program's version number and exit
  -f FOO, --foo FOO  foo bar
  -b BAR, --bar BAR  bar foo
  --baz BAZ          baz bar
```

`Parse arguments:`

```
$ ./test.js -f=3 --bar=4 --baz 5
{ foo: '3', bar: '4', baz: '5' }
```

20.2 API docs

Since this is a port with minimal divergence, there's no separate documentation.
Use original one instead, with notes about difference.

1. Original doc.
2. Original tutorial.
3. Difference with python.

20.3 argparse for enterprise

Available as part of the Tidelift Subscription

The maintainers of argparse and thousands of other packages are working
with Tidelift to deliver commercial support and maintenance for the open
source dependencies you use to build your applications. Save time, reduce
risk, and improve code health, while paying the maintainers of the exact
dependencies you use. Learn more.

Chapter 21

Array Flatten

Flatten an array of nested arrays into a single flat array. Accepts an optional depth.

21.1 Installation

```
npm install array-flatten --save
```

21.2 Usage

```
var flatten = require('array-flatten')
flatten([1, [2, [3, [4, [5], 6], 7], 8], 9])
//=> [1, 2, 3, 4, 5, 6, 7, 8, 9]
flatten([1, [2, [3, [4, [5], 6], 7], 8], 9], 2)
//=> [1, 2, 3, [4, [5], 6], 7, 8, 9]
(function () {
  flatten(arguments) //=> [1, 2, 3]
})(1, [2, 3])
```

21.3 License

MIT

Chapter 22

AWS SSL Profiles

`AWS RDS` **SSL** Certificates Bundles.

Table of Contents

22.1 Installation

```
npm install --save aws-ssl-profiles
```

22.2 Usage

22.2.1 <a href="https://github.com/mysqljs/mysql" >mysqljs/mysql</a>

```
const mysql = require('mysql');
const awsCaBundle = require('aws-ssl-profiles');
// mysql connection
const connection = mysql.createConnection({
  //...
  ssl: awsCaBundle,
});
// mysql connection pool
const pool = mysql.createPool({
  //...
  ssl: awsCaBundle,
});
```

22.2.2 <a href="https://github.com/sidorares/node-mysql2" >MySQL2</a>

```
const mysql = require('mysql2');
const awsCaBundle = require('aws-ssl-profiles');
// mysql2 connection
const connection = mysql.createConnection({
  //...
  ssl: awsCaBundle,
});
// mysql2 connection pool
const pool = mysql.createPool({
  //...
  ssl: awsCaBundle,
});
```

22.2.3 <a href="https://github.com/brianc/node-postgres" >node-postgres</a>

```
const pg = require('pg');
const awsCaBundle = require('aws-ssl-profiles');
// pg connection
const client = new pg.Client({
  // ...
  ssl: awsCaBundle,
});
// pg connection pool
const pool = new pg.Pool({
  // ...
  ssl: awsCaBundle,
});
```

22.2.4 Custom <tt>ssl</tt> options

Using **AWS SSL Profiles** with custom `ssl` options:

```
{
  // ...
  ssl: {
    ...awsCaBundle,
    rejectUnauthorized: true,
    // ...
  }
}
{
  // ...
  ssl: {
    ca: awsCaBundle.ca,
    rejectUnauthorized: true,
    // ...
  }
}
```

22.2.5 Custom bundles

```
const { proxyBundle } = require('aws-ssl-profiles');
{
  // ...
  ssl: proxyBundle,
}
```

22.3 License

AWS SSL Profiles is under the `MIT License`.

22.4 Security

Please check the **SECURITY.md**.

22.5 Contributing

Please check the **CONTRIBUTING.md** for instructions.

22.6 Acknowledgements

`Contributors`.

Chapter 23

LICENSE

Chapter 24

balanced-match

Match balanced string pairs, like `{` and `}` or `<b>` and `</b>`. Supports regular expressions as well!

24.1 Example

Get the first matching pair of braces:

```
var balanced = require('balanced-match');
console.log(balanced('{', '}', 'pre{in{nested}}post'));
console.log(balanced('{', '}', 'pre{first}between{second}post'));
console.log(balanced(/\s+\{\s+/, /\s+\}\s+/, 'pre  {   in{nest}   }  post'));
```

The matches are:

```
$ node example.js
{ start: 3, end: 14, pre: 'pre', body: 'in{nested}', post: 'post' }
{ start: 3,
  end: 9,
  pre: 'pre',
  body: 'first',
  post: 'between{second}post' }
{ start: 3, end: 17, pre: 'pre', body: 'in{nest}', post: 'post' }
```

24.2 API

24.2.1 var m = balanced(a, b, str)

For the first non-nested matching pair of `a` and `b` in `str`, return an object with those keys:

- **start** the index of the first match of `a`
- **end** the index of the matching `b`
- **pre** the preamble, `a` and `b` not included
- **body** the match, `a` and `b` not included
- **post** the postscript, `a` and `b` not included

If there's no match, `undefined` will be returned.
If the `str` contains more `a` than `b` / there are unmatched pairs, the first match that was closed will be used. For example, `{{a}` will match '['{', 'a', '']and{a}}`will match`['', 'a', '}']`.

24.2.2 var r = balanced.range(a, b, str)

For the first non-nested matching pair of `a` and `b` in `str`, return an array with indexes: `[ <a index>, <b index> ]`.
If there's no match, `undefined` will be returned.
If the `str` contains more `a` than `b` / there are unmatched pairs, the first match that was closed will be used. For example, `{{a}` will match `[ 1, 3 ]` and `{a}}` will match `[0, 2]`.

24.3 Installation

With npm do:

```
npm install balanced-match
```

24.4 Security contact information

To report a security vulnerability, please use the Tidelift security contact. Tidelift will coordinate the fix and disclosure.

24.5 License

Chapter 25

1.20.3 / 2024-09-10

- deps: `qs@6.13`.0
- add `depth` option to customize the depth level in the parser
- IMPORTANT: The default `depth` level for parsing URL-encoded data is now `32` (previously was `Infinity`)

25.1 1.20.2 / 2023-02-21

- Fix strict json error message on Node.js 19+
- deps: content-type1.0.5
  - perf: skip value escaping when unnecessary
- deps: raw-body@2.5.2

25.2 1.20.1 / 2022-10-06

- deps: `qs@6.11`.0
- perf: remove unnecessary object clone

25.3 1.20.0 / 2022-04-02

- Fix error message for json parse whitespace in `strict`
- Fix internal error when inflated body exceeds limit
- Prevent loss of async hooks context
- Prevent hanging when request already read
- deps: depd@2.0.0
  - Replace internal `eval` usage with `Function` constructor
  - Use instance methods on `process` to check for listeners
- deps: http-errors@2.0.0
  - deps: depd@2.0.0
  - deps: statuses@2.0.1
- deps: on-finished@2.4.1
- deps: `qs@6.10`.3
- deps: raw-body@2.5.1
  - deps: http-errors@2.0.0

25.4 1.19.2 / 2022-02-15

- deps: bytes@3.1.2
- deps: qs@6.9.7
  - Fix handling of `__proto__` keys
- deps: raw-body@2.4.3
  - deps: bytes@3.1.2

25.5 1.19.1 / 2021-12-10

- deps: bytes@3.1.1
- deps: http-errors@1.8.1
  - deps: inherits@2.0.4
  - deps: toidentifier@1.0.1
  - deps: setprototypeof@1.2.0
- deps: qs@6.9.6
- deps: raw-body@2.4.2
  - deps: bytes@3.1.1
  - deps: http-errors@1.8.1
- deps: safe-buffer@5.2.1
- deps: type-is1.6.18

25.6 1.19.0 / 2019-04-25

- deps: bytes@3.1.0
  - Add petabyte (`pb`) support
- deps: http-errors@1.7.2
  - Set constructor name when possible
  - deps: setprototypeof@1.1.1
  - deps: statuses@'>= 1.5.0 < 2'
- deps: `iconv-lite@0.4.24`
  - Added encoding MIK
- deps: qs@6.7.0
  - Fix parsing array brackets after index
- deps: raw-body@2.4.0
  - deps: bytes@3.1.0
  - deps: http-errors@1.7.2
  - deps: `iconv-lite@0.4.24`
- deps: type-is1.6.17
  - deps: mime-types2.1.24
  - perf: prevent internal `throw` on invalid type

25.7 1.18.3 / 2018-05-14

- Fix stack trace for strict json parse error
- deps: depd1.1.2
  - perf: remove argument reassignment
- deps: http-errors1.6.3
  - deps: depd1.1.2
  - deps: setprototypeof@1.1.0
  - deps: statuses@'>= 1.3.1 < 2'
- deps: `iconv-lite@0.4.23`
  - Fix loading encoding with year appended
  - Fix deprecation warnings on Node.js 10+
- deps: qs@6.5.2
- deps: raw-body@2.3.3
  - deps: http-errors@1.6.3
  - deps: `iconv-lite@0.4.23`
- deps: type-is1.6.16
  - deps: mime-types2.1.18

25.8 1.18.2 / 2017-09-22

- deps: debug@2.6.9
- perf: remove argument reassignment

25.9 1.18.1 / 2017-09-12

- deps: content-type1.0.4
  - perf: remove argument reassignment
  - perf: skip parameter parsing when no parameters
- deps: `iconv-lite@0.4.19`
  - Fix ISO-8859-1 regression
  - Update Windows-1255
- deps: qs@6.5.1
  - Fix parsing & compacting very deep objects
- deps: raw-body@2.3.2
  - deps: `iconv-lite@0.4.19`

25.10 1.18.0 / 2017-09-08

- Fix JSON strict violation error to match native parse error
- Include the `body` property on verify errors
- Include the `type` property on all generated errors
- Use `http-errors` to set status code on errors
- deps: bytes@3.0.0
- deps: debug@2.6.8
- deps: depd1.1.1
  - Remove unnecessary `Buffer` loading
- deps: http-errors1.6.2
  - deps: depd@1.1.1
- deps: `iconv-lite@0.4.18`
  - Add support for React Native
  - Add a warning if not loaded as utf-8
  - Fix CESU-8 decoding in Node.js 8
  - Improve speed of ISO-8859-1 encoding
- deps: qs@6.5.0
- deps: raw-body@2.3.1
  - Use `http-errors` for standard emitted errors
  - deps: bytes@3.0.0
  - deps: `iconv-lite@0.4.18`
  - perf: skip buffer decoding on overage chunk
- perf: prevent internal `throw` when missing charset

25.11 1.17.2 / 2017-05-17

- deps: debug@2.6.7
  - Fix `DEBUG_MAX_ARRAY_LENGTH`
  - deps: ms@2.0.0
- deps: type-is1.6.15
  - deps: mime-types2.1.15

25.12 1.17.1 / 2017-03-06

- deps: qs@6.4.0
  - Fix regression parsing keys starting with `[`

25.13 1.17.0 / 2017-03-01

- deps: http-errors1.6.1
  - Make `message` property enumerable for `HttpError`s
  - deps: setprototypeof@1.0.3
- deps: qs@6.3.1
  - Fix compacting nested arrays

25.14 1.16.1 / 2017-02-10

- deps: debug@2.6.1
  - Fix deprecation messages in WebStorm and other editors
  - Undeprecate `DEBUG_FD` set to `1` or `2`

25.15 1.16.0 / 2017-01-17

- deps: debug@2.6.0
  - Allow colors in workers
  - Deprecated `DEBUG_FD` environment variable
  - Fix error when running under React Native
  - Use same color for same namespace
  - deps: ms@0.7.2
- deps: http-errors1.5.1
  - deps: inherits@2.0.3
  - deps: setprototypeof@1.0.2
  - deps: statuses@'>= 1.3.1 < 2'
- deps: `iconv-lite@0.4.15`
  - Added encoding MS-31J
  - Added encoding MS-932
  - Added encoding MS-936
  - Added encoding MS-949
  - Added encoding MS-950
  - Fix GBK/GB18030 handling of Euro character
- deps: qs@6.2.1
  - Fix array parsing from skipping empty values
- deps: raw-body2.2.0
  - deps: `iconv-lite@0.4.15`
- deps: type-is1.6.14
  - deps: mime-types2.1.13

25.16 1.15.2 / 2016-06-19

- deps: bytes@2.4.0
- deps: content-type1.0.2
  - perf: enable strict mode
- deps: http-errors1.5.0
  - Use `setprototypeof` module to replace __`proto`__ setting
  - deps: statuses@'>= 1.3.0 < 2'
  - perf: enable strict mode
- deps: qs@6.2.0
- deps: raw-body2.1.7
  - deps: bytes@2.4.0
  - perf: remove double-cleanup on happy path
- deps: type-is1.6.13
  - deps: mime-types2.1.11

25.17 1.15.1 / 2016-05-05

- deps: bytes@2.3.0
  - Drop partial bytes on all parsed units
  - Fix parsing byte string that looks like hex
- deps: raw-body2.1.6
  - deps: bytes@2.3.0
- deps: type-is1.6.12
  - deps: mime-types2.1.10

25.18 1.15.0 / 2016-02-10

- deps: http-errors1.4.0
  - Add `HttpError` export, for `err instanceof createError.HttpError`
  - deps: inherits@2.0.1
  - deps: statuses@'>= 1.2.1 < 2'
- deps: qs@6.1.0
- deps: type-is1.6.11
  - deps: mime-types2.1.9

25.19 1.14.2 / 2015-12-16

- deps: bytes@2.2.0
- deps: `iconv-lite@0.4.13`
- deps: qs@5.2.0
- deps: raw-body2.1.5
  - deps: bytes@2.2.0
  - deps: `iconv-lite@0.4.13`
- deps: type-is1.6.10
  - deps: mime-types2.1.8

25.20 1.14.1 / 2015-09-27

- Fix issue where invalid charset results in 400 when `verify` used
- deps: `iconv-lite@0.4.12`
  - Fix CESU-8 decoding in Node.js 4.x
- deps: raw-body2.1.4
  - Fix masking critical errors from `iconv-lite`
  - deps: `iconv-lite@0.4.12`
- deps: type-is1.6.9
  - deps: mime-types2.1.7

25.21 1.14.0 / 2015-09-16

- Fix JSON strict parse error to match syntax errors
- Provide static `require` analysis in `urlencoded` parser
- deps: depd1.1.0
  - Support web browser loading
- deps: qs@5.1.0
- deps: raw-body2.1.3
  - Fix sync callback when attaching data listener causes sync read
- deps: type-is1.6.8
  - Fix type error when given invalid type to match against
  - deps: mime-types2.1.6

25.22 1.13.3 / 2015-07-31

- deps: type-is1.6.6
  - deps: mime-types2.1.4

25.23 1.13.2 / 2015-07-05

- deps: `iconv-lite@0.4.11`
- deps: qs@4.0.0
  - Fix dropping parameters like `hasOwnProperty`
  - Fix user-visible incompatibilities from 3.1.0
  - Fix various parsing edge cases
- deps: raw-body2.1.2
  - Fix error stack traces to skip `makeError`
  - deps: `iconv-lite@0.4.11`
- deps: type-is1.6.4
  - deps: mime-types2.1.2
  - perf: enable strict mode
  - perf: remove argument reassignment

25.24 1.13.1 / 2015-06-16

- deps: qs@2.4.2
  - Downgraded from 3.1.0 because of user-visible incompatibilities

25.25 1.13.0 / 2015-06-14

- Add `statusCode` property on `Error`s, in addition to `status`
- Change `type` default to `application/json` for JSON parser
- Change `type` default to `application/x-www-form-urlencoded` for urlencoded parser
- Provide static `require` analysis
- Use the `http-errors` module to generate errors
- deps: bytes@2.1.0
  - Slight optimizations
- deps: `iconv-lite@0.4.10`
  - The encoding UTF-16 without BOM now defaults to UTF-16LE when detection fails
  - Leading BOM is now removed when decoding
- deps: on-finished2.3.0
  - Add defined behavior for HTTP `CONNECT` requests
  - Add defined behavior for HTTP `Upgrade` requests
  - deps: ee-first@1.1.1
- deps: qs@3.1.0
  - Fix dropping parameters like `hasOwnProperty`
  - Fix various parsing edge cases
  - Parsed object now has `null` prototype
- deps: raw-body2.1.1

  - Use `unpipe` module for unpiping requests
  - deps: `iconv-lite@0.4.10`
- deps: type-is1.6.3
  - deps: mime-types2.1.1
  - perf: reduce try block size
  - perf: remove bitwise operations
- perf: enable strict mode
- perf: remove argument reassignment
- perf: remove delete call

25.26 1.12.4 / 2015-05-10

- deps: debug2.2.0
- deps: qs@2.4.2
  - Fix allowing parameters like `constructor`
- deps: on-finished2.2.1
- deps: raw-body2.0.1
  - Fix a false-positive when unpiping in Node.js 0.8
  - deps: bytes@2.0.1
- deps: type-is1.6.2
  - deps: mime-types2.0.11

25.27 1.12.3 / 2015-04-15

- Slight efficiency improvement when not debugging
- deps: depd1.0.1
- deps: iconv-lite@0.4.8
  - Add encoding alias UNICODE-1-1-UTF-7
- deps: raw-body@1.3.4
  - Fix hanging callback if request aborts during read
  - deps: iconv-lite@0.4.8

25.28 1.12.2 / 2015-03-16

- deps: qs@2.4.1
  - Fix error when parameter `hasOwnProperty` is present

25.29 1.12.1 / 2015-03-15

- deps: debug2.1.3
  - Fix high intensity foreground color for bold
  - deps: ms@0.7.0
- deps: type-is1.6.1
  - deps: mime-types2.0.10

25.30 1.12.0 / 2015-02-13

- add `debug` messages
- accept a function for the `type` option
- use `content-type` to parse `Content-Type` headers
- deps: iconv-lite@0.4.7
  - Gracefully support enumerables on `Object.prototype`
- deps: raw-body@1.3.3
  - deps: iconv-lite@0.4.7
- deps: type-is1.6.0
  - fix argument reassignment
  - fix false-positives in `hasBody` `Transfer-Encoding` check
  - support wildcard for both type and subtype (`*/*`)
  - deps: mime-types2.0.9

25.31 1.11.0 / 2015-01-30

- make internal `extended: true` depth limit infinity
- deps: type-is1.5.6
  - deps: mime-types2.0.8

25.32 1.10.2 / 2015-01-20

- deps: iconv-lite@0.4.6
  - Fix rare aliases of single-byte encodings
- deps: raw-body@1.3.2
  - deps: iconv-lite@0.4.6

25.33 1.10.1 / 2015-01-01

- deps: on-finished2.2.0
- deps: type-is1.5.5
  - deps: mime-types2.0.7

25.34 1.10.0 / 2014-12-02

- make internal `extended: true` array limit dynamic

25.35 1.9.3 / 2014-11-21

- deps: iconv-lite@0.4.5
  - Fix Windows-31J and X-SJIS encoding support
- deps: qs@2.3.3
  - Fix `arrayLimit` behavior
- deps: raw-body@1.3.1
  - deps: iconv-lite@0.4.5
- deps: type-is1.5.3
  - deps: mime-types2.0.3

25.36 1.9.2 / 2014-10-27

- deps: qs@2.3.2
  - Fix parsing of mixed objects and values

25.37 1.9.1 / 2014-10-22

- deps: on-finished2.1.1
  - Fix handling of pipelined requests
- deps: qs@2.3.0
  - Fix parsing of mixed implicit and explicit arrays
- deps: type-is1.5.2
  - deps: mime-types2.0.2

25.38 1.9.0 / 2014-09-24

- include the charset in "unsupported charset" error message
- include the encoding in "unsupported content encoding" error message
- deps: depd1.0.0

25.39 1.8.4 / 2014-09-23

- fix content encoding to be case-insensitive

25.40 1.8.3 / 2014-09-19

- deps: qs@2.2.4
  - Fix issue with object keys starting with numbers truncated

25.41 1.8.2 / 2014-09-15

- deps: depd@0.4.5

25.42 1.8.1 / 2014-09-07

- deps: media-typer@0.3.0
- deps: type-is1.5.1

25.43 1.8.0 / 2014-09-05

- make empty-body-handling consistent between chunked requests
  - empty `json` produces `{}`
  - empty `raw` produces `new Buffer(0)`
  - empty `text` produces ‘"
  - `empty`urlencoded`produces`{}`
- deps: qs@2.2.3
  - Fix issue where first empty value in array is discarded
- deps: type-is1.5.0
  - fix `hasbody` to be true for `content-length: 0`

25.44 1.7.0 / 2014-09-01

- add `parameterLimit` option to `urlencoded` parser
- change `urlencoded` extended array limit to 100
- respond with 413 when over `parameterLimit` in `urlencoded`

25.45 1.6.7 / 2014-08-29

- deps: qs@2.2.2
  - Remove unnecessary cloning

25.46 1.6.6 / 2014-08-27

- deps: qs@2.2.0
  - Array parsing fix
  - Performance improvements

25.47 1.6.5 / 2014-08-16

- deps: on-finished@2.1.0

25.48 1.6.4 / 2014-08-14

- deps: qs@1.2.2

25.49 1.6.3 / 2014-08-10

- deps: qs@1.2.1

25.50 1.6.2 / 2014-08-07

- deps: qs@1.2.0
  - Fix parsing array of objects

25.51 1.6.1 / 2014-08-06

- deps: qs@1.1.0
  - Accept urlencoded square brackets
  - Accept empty values in implicit array notation

25.52 1.6.0 / 2014-08-05

- deps: qs@1.0.2
  - Complete rewrite
  - Limits array length to 20
  - Limits object depth to 5
  - Limits parameters to 1,000

25.53 1.5.2 / 2014-07-27

- deps: depd@0.4.4
  - Work-around v8 generating empty stack traces

25.54 1.5.1 / 2014-07-26

- deps: depd@0.4.3
  - Fix exception when global `Error.stackTraceLimit` is too low

25.55 1.5.0 / 2014-07-20

- deps: depd@0.4.2
  - Add `TRACE_DEPRECATION` environment variable
  - Remove non-standard grey color from color output
  - Support `--no-deprecation` argument
  - Support `--trace-deprecation` argument
- deps: iconv-lite@0.4.4
  - Added encoding UTF-7
- deps: raw-body@1.3.0
  - deps: iconv-lite@0.4.4
  - Added encoding UTF-7
  - Fix `Cannot switch to old mode now` error on Node.js 0.10+
- deps: type-is1.3.2

25.56 1.4.3 / 2014-06-19

- deps: type-is@1.3.1
  - fix global variable leak

25.57 1.4.2 / 2014-06-19

- deps: type-is@1.3.0
  - improve type parsing

25.58 1.4.1 / 2014-06-19

- fix urlencoded extended deprecation message

25.59 1.4.0 / 2014-06-19

- add `text` parser
- add `raw` parser
- check accepted charset in content-type (accepts utf-8)
- check accepted encoding in content-encoding (accepts identity)
- deprecate `bodyParser()` middleware; use `.json()` and `.urlencoded()` as needed
- deprecate `urlencoded()` without provided `extended` option
- lazy-load urlencoded parsers
- parsers split into files for reduced mem usage
- support gzip and deflate bodies
  - set `inflate: false` to turn off
- deps: raw-body@1.2.2
  - Support all encodings from `iconv-lite`

25.60 1.3.1 / 2014-06-11

- deps: type-is@1.2.1
  - Switch dependency from mime to mime-types@1.0.0

25.61 1.3.0 / 2014-05-31

- add `extended` option to urlencoded parser

25.62 1.2.2 / 2014-05-27

- deps: raw-body@1.1.6
  - assert stream encoding on node.js 0.8
  - assert stream encoding on node.js $<$ 0.10.6
  - deps: bytes@1

25.63 1.2.1 / 2014-05-26

- invoke `next(err)` after request fully read
  - prevents hung responses and socket hang ups

25.64 1.2.0 / 2014-05-11

- add `verify` option
- deps: type-is@1.2.0
  - support suffix matching

25.65 1.1.2 / 2014-05-11

- improve json parser speed

25.66 1.1.1 / 2014-05-11

- fix repeated limit parsing with every request

25.67 1.1.0 / 2014-05-10

- add `type` option
- deps: pin for safety and consistency

25.68 1.0.2 / 2014-04-14

- use `type-is` module

25.69 1.0.1 / 2014-03-20

- lower default limits to 100kb

Chapter 26

body-parser

Node.js body parsing middleware.

Parse incoming request bodies in a middleware before your handlers, available under the `req.body` property.

Note As `req.body`'s shape is based on user-controlled input, all properties and values in this object are untrusted and should be validated before trusting. For example, `req.body.foo.toString()` may fail in multiple ways, for example the `foo` property may not be there or may not be a string, and `toString` may not be a function and instead a string or other user input.

`Learn about the anatomy of an HTTP transaction in Node.js`.

This does not handle multipart bodies, due to their complex and typically large nature. For multipart bodies, you may be interested in the following modules:

- `busboy` and `connect-busboy`
- `multiparty` and `connect-multiparty`
- `formidable`
- `multer`

This module provides the following parsers:

- JSON body parser
- Raw body parser
- Text body parser
- URL-encoded form body parser

Other body parsers you might be interested in:

- `body`
- `co-body`

26.1 Installation

```
$ npm install body-parser
```

26.2 API

```
var bodyParser = require('body-parser')
```

The `bodyParser` object exposes various factories to create middlewares. All middlewares will populate the `req.body` property with the parsed body when the `Content-Type` request header matches the `type` option, or an empty object (`{}`) if there was no body to parse, the `Content-Type` was not matched, or an error occurred.

The various errors returned by this module are described in the errors section.

26.2.1 bodyParser.json([options])

Returns middleware that only parses `json` and only looks at requests where the `Content-Type` header matches the `type` option. This parser accepts any Unicode encoding of the body and supports automatic inflation of `gzip` and `deflate` encodings.
A new `body` object containing the parsed data is populated on the `request` object after the middleware (i.e. `req.body`).

26.2.1.1 Options

The `json` function takes an optional `options` object that may contain any of the following keys:

26.2.1.1.1 inflate When set to `true`, then deflated (compressed) bodies will be inflated; when `false`, deflated bodies are rejected. Defaults to `true`.

26.2.1.1.2 limit Controls the maximum request body size. If this is a number, then the value specifies the number of bytes; if it is a string, the value is passed to the `bytes` library for parsing. Defaults to "100kb".

26.2.1.1.3 reviver The `reviver` option is passed directly to `JSON.parse` as the second argument. You can find more information on this argument `in the MDN documentation about JSON.parse`.

26.2.1.1.4 strict When set to `true`, will only accept arrays and objects; when `false` will accept anything `JSON.parse` accepts. Defaults to `true`.

26.2.1.1.5 type The `type` option is used to determine what media type the middleware will parse. This option can be a string, array of strings, or a function. If not a function, `type` option is passed directly to the `type-is` library and this can be an extension name (like `json`), a mime type (like `application/json`), or a mime type with a wildcard (like `*/*` or `*/json`). If a function, the `type` option is called as `fn(req)` and the request is parsed if it returns a truthy value. Defaults to `application/json`.

26.2.1.1.6 verify The `verify` option, if supplied, is called as `verify(req, res, buf, encoding)`, where `buf` is a `Buffer` of the raw request body and `encoding` is the encoding of the request. The parsing can be aborted by throwing an error.

26.2.2 bodyParser.raw([options])

Returns middleware that parses all bodies as a `Buffer` and only looks at requests where the `Content-Type` header matches the `type` option. This parser supports automatic inflation of `gzip` and `deflate` encodings.
A new `body` object containing the parsed data is populated on the `request` object after the middleware (i.e. `req.body`). This will be a `Buffer` object of the body.

26.2.2.1 Options

The `raw` function takes an optional `options` object that may contain any of the following keys:

26.2.2.1.1 inflate When set to `true`, then deflated (compressed) bodies will be inflated; when `false`, deflated bodies are rejected. Defaults to `true`.

26.2.2.1.2 limit Controls the maximum request body size. If this is a number, then the value specifies the number of bytes; if it is a string, the value is passed to the `bytes` library for parsing. Defaults to "100kb".

26.2.2.1.3 type The `type` option is used to determine what media type the middleware will parse. This option can be a string, array of strings, or a function. If not a function, `type` option is passed directly to the `type-is` library and this can be an extension name (like `bin`), a mime type (like `application/octet-stream`), or a mime type with a wildcard (like `*/*` or `application/*`). If a function, the `type` option is called as `fn(req)` and the request is parsed if it returns a truthy value. Defaults to `application/octet-stream`.

26.2.2.1.4 verify The `verify` option, if supplied, is called as `verify(req, res, buf, encoding)`, where `buf` is a `Buffer` of the raw request body and `encoding` is the encoding of the request. The parsing can be aborted by throwing an error.

26.2.3 bodyParser.text([options])

Returns middleware that parses all bodies as a string and only looks at requests where the `Content-Type` header matches the `type` option. This parser supports automatic inflation of `gzip` and `deflate` encodings.
A new `body` string containing the parsed data is populated on the `request` object after the middleware (i.e. `req.body`). This will be a string of the body.

26.2.3.1 Options

The `text` function takes an optional `options` object that may contain any of the following keys:

26.2.3.1.1 defaultCharset Specify the default character set for the text content if the charset is not specified in the `Content-Type` header of the request. Defaults to `utf-8`.

26.2.3.1.2 inflate When set to `true`, then deflated (compressed) bodies will be inflated; when `false`, deflated bodies are rejected. Defaults to `true`.

26.2.3.1.3 limit Controls the maximum request body size. If this is a number, then the value specifies the number of bytes; if it is a string, the value is passed to the `bytes` library for parsing. Defaults to "100kb"`.

26.2.3.1.4 type The `type` option is used to determine what media type the middleware will parse. This option can be a string, array of strings, or a function. If not a function, `type` option is passed directly to the `type-is` library and this can be an extension name (like `txt`), a mime type (like `text/plain`), or a mime type with a wildcard (like `*/*` or `text/*`). If a function, the `type` option is called as `fn(req)` and the request is parsed if it returns a truthy value. Defaults to `text/plain`.

26.2.3.1.5 verify The `verify` option, if supplied, is called as `verify(req, res, buf, encoding)`, where `buf` is a `Buffer` of the raw request body and `encoding` is the encoding of the request. The parsing can be aborted by throwing an error.

26.2.4 bodyParser.urlencoded([options])

Returns middleware that only parses `urlencoded` bodies and only looks at requests where the `Content-Type` header matches the `type` option. This parser accepts only UTF-8 encoding of the body and supports automatic inflation of `gzip` and `deflate` encodings.
A new `body` object containing the parsed data is populated on the `request` object after the middleware (i.e. `req.body`). This object will contain key-value pairs, where the value can be a string or array (when `extended` is `false`), or any type (when `extended` is `true`).

26.2.4.1 Options

The `urlencoded` function takes an optional `options` object that may contain any of the following keys:

26.2.4.1.1 extended The `extended` option allows to choose between parsing the URL-encoded data with the `querystring` library (when `false`) or the `qs` library (when `true`). The "extended" syntax allows for rich objects and arrays to be encoded into the URL-encoded format, allowing for a JSON-like experience with URL-encoded. For more information, please `see the qs library`.
Defaults to `true`, but using the default has been deprecated. Please research into the difference between `qs` and `querystring` and choose the appropriate setting.

26.2.4.1.2 inflate When set to `true`, then deflated (compressed) bodies will be inflated; when `false`, deflated bodies are rejected. Defaults to `true`.

26.2.4.1.3 limit Controls the maximum request body size. If this is a number, then the value specifies the number of bytes; if it is a string, the value is passed to the `bytes` library for parsing. Defaults to "100kb"`.

26.2.4.1.4 parameterLimit The `parameterLimit` option controls the maximum number of parameters that are allowed in the URL-encoded data. If a request contains more parameters than this value, a 413 will be returned to the client. Defaults to `1000`.

26.2.4.1.5 type The `type` option is used to determine what media type the middleware will parse. This option can be a string, array of strings, or a function. If not a function, `type` option is passed directly to the `type-is` library and this can be an extension name (like `urlencoded`), a mime type (like `application/x-www-form-urlencoded`), or a mime type with a wildcard (like `*/x-www-form-urlencoded`). If a function, the `type` option is called as `fn(req)` and the request is parsed if it returns a truthy value. Defaults to `application/x-www-form-urlencoded`.

26.2.4.1.6 verify The `verify` option, if supplied, is called as `verify(req, res, buf, encoding)`, where `buf` is a `Buffer` of the raw request body and `encoding` is the encoding of the request. The parsing can be aborted by throwing an error.

26.2.4.2 depth

The `depth` option is used to configure the maximum depth of the `qs` library when `extended` is `true`. This allows you to limit the amount of keys that are parsed and can be useful to prevent certain types of abuse. Defaults to `32`. It is recommended to keep this value as low as possible.

26.3 Errors

The middlewares provided by this module create errors using the `http-errors module`. The errors will typically have a `status/statusCode` property that contains the suggested HTTP response code, an `expose` property to determine if the `message` property should be displayed to the client, a `type` property to determine the type of error without matching against the `message`, and a `body` property containing the read body, if available. The following are the common errors created, though any error can come through for various reasons.

26.3.1 content encoding unsupported

This error will occur when the request had a `Content-Encoding` header that contained an encoding but the "inflation" option was set to `false`. The `status` property is set to `415`, the `type` property is set to "encoding.↩ unsupported'`, and the`charset` property will be set to the encoding that is unsupported.

26.3.2 entity parse failed

This error will occur when the request contained an entity that could not be parsed by the middleware. The `status` property is set to `400`, the `type` property is set to "entity.parse.failed'`, and the`body` property is set to the entity value that failed parsing.

26.3.3 entity verify failed

This error will occur when the request contained an entity that could not be failed verification by the defined `verify` option. The `status` property is set to `403`, the `type` property is set to "entity.verify.failed'`, and the` body` property is set to the entity value that failed verification.

26.3.4 request aborted

This error will occur when the request is aborted by the client before reading the body has finished. The `received` property will be set to the number of bytes received before the request was aborted and the `expected` property is set to the number of expected bytes. The `status` property is set to `400` and `type` property is set to "request.↩ aborted"`.

26.3.5 request entity too large

This error will occur when the request body's size is larger than the "limit" option. The `limit` property will be set to the byte limit and the `length` property will be set to the request body's length. The `status` property is set to `413` and the `type` property is set to "entity.too.large"`.

26.3.6 request size did not match content length

This error will occur when the request's length did not match the length from the `Content-Length` header. This typically occurs when the request is malformed, typically when the `Content-Length` header was calculated based on characters instead of bytes. The `status` property is set to `400` and the `type` property is set to "request.size.invalid"`.

26.3.7 stream encoding should not be set

This error will occur when something called the `req.setEncoding` method prior to this middleware. This module operates directly on bytes only and you cannot call `req.setEncoding` when using this module. The `status` property is set to `500` and the `type` property is set to "stream.encoding.set"`.

26.3.8 stream is not readable

This error will occur when the request is no longer readable when this middleware attempts to read it. This typically means something other than a middleware from this module read the request body already and the middleware was also configured to read the same request. The `status` property is set to `500` and the `type` property is set to "stream.not.readable"`.

26.3.9 too many parameters

This error will occur when the content of the request exceeds the configured `parameterLimit` for the `urlencoded` parser. The `status` property is set to `413` and the `type` property is set to "parameters.too.many"`.

26.3.10 unsupported charset "BOGUS"

This error will occur when the request had a charset parameter in the `Content-Type` header, but the `iconv-lite` module does not support it OR the parser does not support it. The charset is contained in the message as well as in the `charset` property. The `status` property is set to `415`, the `type` property is set to "charset.unsupported', `and the`charset` property is set to the charset that is unsupported.

26.3.11 unsupported content encoding "bogus"

This error will occur when the request had a `Content-Encoding` header that contained an unsupported encoding. The encoding is contained in the message as well as in the `encoding` property. The `status` property is set to `415`, the `type` property is set to "encoding.unsupported', `and the`encoding` property is set to the encoding that is unsupported.

26.3.12 The input exceeded the depth

This error occurs when using `bodyParser.urlencoded` with the `extended` property set to `true` and the input exceeds the configured `depth` option. The `status` property is set to `400`. It is recommended to review the `depth` option and evaluate if it requires a higher value. When the `depth` option is set to `32` (default value), the error will not be thrown.

26.4 Examples

26.4.1 Express/Connect top-level generic

This example demonstrates adding a generic JSON and URL-encoded parser as a top-level middleware, which will parse the bodies of all incoming requests. This is the simplest setup.

```
var express = require('express')
var bodyParser = require('body-parser')
var app = express()
// parse application/x-www-form-urlencoded
app.use(bodyParser.urlencoded({ extended:  false }))
// parse application/json
app.use(bodyParser.json())
app.use(function (req, res) {
  res.setHeader('Content-Type', 'text/plain')
  res.write('you posted:\n')
  res.end(JSON.stringify(req.body, null, 2))
})
```

26.4.2 Express route-specific

This example demonstrates adding body parsers specifically to the routes that need them. In general, this is the most recommended way to use body-parser with Express.

```
var express = require('express')
var bodyParser = require('body-parser')
var app = express()
// create application/json parser
var jsonParser = bodyParser.json()
// create application/x-www-form-urlencoded parser
var urlencodedParser = bodyParser.urlencoded({ extended:  false })
// POST /login gets urlencoded bodies
app.post('/login', urlencodedParser, function (req, res) {
  res.send('welcome, ' + req.body.username)
})
// POST /api/users gets JSON bodies
app.post('/api/users', jsonParser, function (req, res) {
  // create user in req.body
})
```

26.4.3 Change accepted type for parsers

All the parsers accept a `type` option which allows you to change the `Content-Type` that the middleware will parse.

```
var express = require('express')
var bodyParser = require('body-parser')
var app = express()
// parse various different custom JSON types as JSON
app.use(bodyParser.json({ type:  'application/*+json' }))
// parse some custom thing into a Buffer
app.use(bodyParser.raw({ type:  'application/vnd.custom-type' }))
// parse an HTML body into a string
app.use(bodyParser.text({ type:  'text/html' }))
```

26.5 License

[MIT](LICENSE)

Chapter 27

Security Policies and Procedures

27.1 Reporting a Bug

The Express team and community take all security bugs seriously. Thank you for improving the security of Express. We appreciate your efforts and responsible disclosure and will make every effort to acknowledge your contributions. Report security bugs by emailing the current owner(s) of `body-parser`. This information can be found in the npm registry using the command `npm owner ls body-parser`. If unsure or unable to get the information from the above, open an issue in the `project issue tracker` asking for the current contact information.
To ensure the timely response to your report, please ensure that the entirety of the report is contained within the email body and not solely behind a web link or an attachment.
At least one owner will acknowledge your email within 48 hours, and will send a more detailed response within 48 hours indicating the next steps in handling your report. After the initial reply to your report, the owners will endeavor to keep you informed of the progress towards a fix and full announcement, and may ask for additional information or guidance.

Chapter 28

brace-expansion

Brace expansion, as known from sh/bash, in JavaScript.

28.1 Example

```
var expand = require('brace-expansion');
expand('file-{a,b,c}.jpg')
// => ['file-a.jpg', 'file-b.jpg', 'file-c.jpg']
expand('-v{,,}')
// => ['-v', '-v', '-v']
expand('file{0..2}.jpg')
// => ['file0.jpg', 'file1.jpg', 'file2.jpg']
expand('file-{a..c}.jpg')
// => ['file-a.jpg', 'file-b.jpg', 'file-c.jpg']
expand('file{2..0}.jpg')
// => ['file2.jpg', 'file1.jpg', 'file0.jpg']
expand('file{0..4..2}.jpg')
// => ['file0.jpg', 'file2.jpg', 'file4.jpg']
expand('file-{a..e..2}.jpg')
// => ['file-a.jpg', 'file-c.jpg', 'file-e.jpg']
expand('file{00..10..5}.jpg')
// => ['file00.jpg', 'file05.jpg', 'file10.jpg']
expand('{{A..C},{a..c}}')
// => ['A', 'B', 'C', 'a', 'b', 'c']
expand('ppp{,config,oe{,conf}}')
// => ['ppp', 'pppconfig', 'pppoe', 'pppoeconf']
```

28.2 API

```
var expand = require('brace-expansion');
```

28.2.1 var expanded = expand(str)

Return an array of all possible and valid expansions of `str`. If none are found, `[str]` is returned.
Valid expansions are:

```
/^(.*,)+(.+)?$/
// {a,b,...}
```

A comma separated list of options, like `{a,b}` or `{a,{b,c}}` or `{,a,}`.

```
/^-?\d+\.\.-?\d+(\.\.-?\d+)?$/
// {x..y[..incr]}
```

A numeric sequence from `x` to `y` inclusive, with optional increment. If `x` or `y` start with a leading `0`, all the numbers will be padded to have equal length. Negative numbers and backwards iteration work too.

```
/^-?\d+\.\.-?\d+(\.\.-?\d+)?$/
// {x..y[..incr]}
```

An alphabetic sequence from `x` to `y` inclusive, with optional increment. `x` and `y` must be exactly one character, and if given, `incr` must be a number.
For compatibility reasons, the string `${` is not eligible for brace expansion.

28.3 Installation

With npm do:

```
npm install brace-expansion
```

28.4 Contributors

- Julian Gruber
- Isaac Z. Schlueter

28.5 Sponsors

This module is proudly supported by my Sponsors!

Do you want to support modules like this to improve their quality, stability and weigh in on new features? Then please consider donating to my Patreon. Not sure how much of my modules you're using? Try feross/thanks!

28.6 License

Chapter 29

3.1.2 / 2022-01-27

- Fix return value for un-parsable strings

29.1 3.1.1 / 2021-11-15

- Fix "thousandsSeparator" incorrecting formatting fractional part

29.2 3.1.0 / 2019-01-22

- Add petabyte (`pb`) support

29.3 3.0.0 / 2017-08-31

- Change "kB" to "KB" in format output
- Remove support for Node.js 0.6
- Remove support for ComponentJS

29.4 2.5.0 / 2017-03-24

- Add option "unit"

29.5 2.4.0 / 2016-06-01

- Add option "unitSeparator"

29.6 2.3.0 / 2016-02-15

- Drop partial bytes on all parsed units
- Fix non-finite numbers to `.format` to return `null`
- Fix parsing byte string that looks like hex
- perf: hoist regular expressions

29.7 2.2.0 / 2015-11-13

- add option "decimalPlaces"
- add option "fixedDecimals"

29.8 2.1.0 / 2015-05-21

- add `.format` export
- add `.parse` export

29.9 2.0.2 / 2015-05-20

- remove map recreation
- remove unnecessary object construction

29.10 2.0.1 / 2015-05-07

- fix browserify require
- remove node.extend dependency

29.11 2.0.0 / 2015-04-12

- add option "case"
- add option "thousandsSeparator"
- return "null" on invalid parse input
- support proper round-trip: bytes(bytes(num)) === num
- units no longer case sensitive when parsing

29.12 1.0.0 / 2014-05-05

- add negative support. fixes #6

29.13 0.3.0 / 2014-03-19

- added terabyte support

29.14 0.2.1 / 2013-04-01

- add .component

29.15 0.2.0 / 2012-10-28

- bytes(200).should.eql('200b')

29.16 0.1.0 / 2012-07-04

- add bytes to string conversion [yields]

Chapter 30

Bytes utility

Utility to parse a string bytes (ex: `1TB`) to bytes (`1099511627776`) and vice-versa.

30.1 Installation

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install bytes
```

30.2 Usage

```
var bytes = require('bytes');
```

30.2.0.1 bytes(numberstring value, [options]): numberstringnull

Default export function. Delegates to either `bytes.format` or `bytes.parse` based on the type of `value`.
Arguments

Name	Type	Description
value	`numberstring`	Number value to format or string value to parse
options	`Object`	Conversion options for `format`

Returns

Name	Type	Description
results	`stringnumbernull`	Return null upon error. Numeric value in bytes, or string value otherwise.

Example

```
bytes(1024);
// output:  '1KB'
bytes('1KB');
// output:  1024
```

30.2.0.2 bytes.format(number value, [options]): stringnull

Format the given value in bytes into a string. If the value is negative, it is kept as such. If it is a float, it is rounded.
Arguments

Name	Type	Description
value	`number`	Value in bytes
options	`Object`	Conversion options

Options

Property	Type	Description
decimalPlaces	`numbernull`	Maximum number of decimal places to include in output. Default value to `2`.
fixedDecimals	`booleannull`	Whether to always display the maximum number of decimal places. Default value to `false`
thousandsSeparator	`stringnull`	Example of values: ‘ ’,’,’and’.’... `Default value to". \ilinebr </td> </tr> <tr class="markdown↩ TableRowEven"> <td class="markdown↩ TableBodyNone"> unit \ilinebr </td> <td class="markdownTableBodyNone">`stringnull`\ilinebr </td> <td class="markdownTableBody↩ None"> The unit in which the result will be returned (B/KB/MB/GB/TB). Default value to"(which means auto detect). \ilinebr </td> </tr> <tr class="markdownTable↩ RowOdd"> <td class="markdownTableBody↩ None"> unitSeparator \ilinebr </td> <td class="markdownTableBodyNone">`stringnull`\ilinebr </td> <td class="markdownTableBodyNone"> Separator to use between number and unit. Default value to"`.

Returns

Name	Type	Description
results	`stringnull`	Return null upon error. String value otherwise.

Example

```
bytes.format(1024);
// output:  '1KB'
bytes.format(1000);
// output:  '1000B'
bytes.format(1000, {thousandsSeparator:  '  '});
// output:  '1 000B'
bytes.format(1024 * 1.7, {decimalPlaces:  0});
// output:  '2KB'
bytes.format(1024, {unitSeparator:  '  '});
// output:  '1 KB'
```

30.2.0.3 bytes.parse(stringnumber value): numbernull

Parse the string value into an integer in bytes. If no unit is given, or `value` is a number, it is assumed the value is in bytes.
Supported units and abbreviations are as follows and are case-insensitive:

- `b` for bytes
- `kb` for kilobytes
- `mb` for megabytes
- `gb` for gigabytes
- `tb` for terabytes
- `pb` for petabytes

The units are in powers of two, not ten. This means 1kb = 1024b according to this parser.
Arguments

Name	Type	Description
value	`stringnumber`	String to parse, or number in bytes.

Returns

Name	Type	Description
results	`numbernull`	Return null upon error. Value in bytes otherwise.

Example

```
bytes.parse('1KB');
// output:  1024
bytes.parse('1024');
// output:  1024
bytes.parse(1024);
// output:  1024
```

30.3 License

[MIT](LICENSE)

Chapter 31

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

31.1 <a href="https://github.com/ljharb/call-bind/compare/v1.0.6...v1.0.7" >v1.0.7</a> - 2024-02-12

31.1.1 Commits

- [Refactor] use `es-define-property` `09b76a0`
- [Deps] update `get-intrinsic`, `set-function-length` `ad5136d`

31.2 <a href="https://github.com/ljharb/call-bind/compare/v1.0.5...v1.0.6" >v1.0.6</a> - 2024-02-05

31.2.1 Commits

- [Dev Deps] update `aud`, `npmignore`, `tape` `d564d5c`
- [Deps] update `get-intrinsic`, `set-function-length` `cfc2bdc`
- [Refactor] use `es-errors`, so things that only need those do not need `get-intrinsic` `64cd289`
- [meta] add missing `engines.node` `32a4038`

31.3 <a href="https://github.com/ljharb/call-bind/compare/v1.0.4...v1.0.5" >v1.0.5</a> - 2023-10-19

31.3.1 Commits

- [Fix] throw an error on non-functions as early as possible `f262408`
- [Deps] update `set-function-length` `3fff271`

31.4 <a href="https://github.com/ljharb/call-bind/compare/v1.0.3...v1.0.4" >v1.0.4</a> - 2023-10-19

31.5 <a href="https://github.com/ljharb/call-bind/compare/v1.0.2...v1.0.3" >v1.0.3</a> - 2023-10-19

31.5.1 Commits

- [actions] reuse common workflows `a994df6`
- [meta] use `npmignore` to autogenerate an npmignore file `eef3ef2`
- [readme] flesh out content `1845ccf`
- [actions] use `node/install` instead of `node/run`; use `codecov` action `5b47d53`
- [Refactor] use `set-function-length` `a0e165c`
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `tape` `9c50103`
- [meta] simplify "exports" `019c6d0`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog`, `safe-publish-latest`, `tape` `23bd718`
- [actions] update codecov uploader `62552d7`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog`, `tape` `ec81665`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `safe-publish-latest`, `tape` `35d67fc`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `tape` `0266d8d`
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `tape` `43a5b28`
- [Deps] update `define-data-property`, `function-bind`, `get-intrinsic` `780eb36`
- [Dev Deps] update `aud`, `tape` `90d50ad`
- [meta] use `prepublishOnly` script for npm 7+ `44c5433`
- [Deps] update `get-intrinsic` `86bfbfc`
- [Deps] update `get-intrinsic` `5c53354`
- [actions] update checkout action `4c393a8`
- [Deps] update `get-intrinsic` `4e70bde`
- [Deps] update `get-intrinsic` `55ae803`

31.6 <a href="https://github.com/ljharb/call-bind/compare/v1.0.1...v1.0.2" >v1.0.2</a> - 2021-01-11

31.6.1 Commits

- [Fix] properly include the receiver in the bound length `dbae7bc`

31.7 <a href="https://github.com/ljharb/call-bind/compare/v1.0.0...v1.0.1" >v1.0.1</a> - 2021-01-08

31.7.1 Commits

- [Tests] migrate tests to Github Actions `b6db284`
- [meta] do not publish github action workflow files `ec7fe46`
- [Fix] preserve original function's length when possible `adbceaa`
- [Tests] gather coverage data on every job `d69e23c`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `tape` `2fd3586`
- [Deps] update `get-intrinsic` `f23e931`
- [Deps] update `get-intrinsic` `72d9f44`
- [meta] fix FUNDING.yml `e723573`
- [eslint] ignore coverage output `15e76d2`
- [meta] add Automatic Rebase and Require Allow Edits workflows `8fa4dab`

31.8 v1.0.0 - 2020-10-30

31.8.1 Commits

- Initial commit `306cf98`
- Tests `e10d0bb`
- Implementation `43852ed`
- npm init `408f860`
- [meta] add Automatic Rebase and Require Allow Edits workflows `fb349b2`
- [meta] add `auto-changelog` `c4001fc`
- [meta] add "funding"; create `FUNDING.yml` `d4d6d29`
- [Tests] add `npm run lint` `dedfb98`
- Only apps should have lockfiles `54ac776`
- [meta] add `safe-publish-latest` `9ea8e43`

Chapter 32

call-bind <sup><a href="https://npmjs.org/package/call-bind" ><img src="https://versionbadg.es/ljharb/call-bind.svg" alt="Version Badge"/></a></sup>

[][license-url]

Robustly `.call.bind()` a function.

32.1 Getting started

```
npm install --save call-bind
```

32.2 Usage/Examples

```
const assert = require('assert');
const callBind = require('call-bind');
const callBound = require('call-bind/callBound');
function f(a, b) {
    assert.equal(this, 1);
    assert.equal(a, 2);
    assert.equal(b, 3);
    assert.equal(arguments.length, 2);
}
const fBound = callBind(f);
const slice = callBound('Array.prototype.slice');
delete Function.prototype.call;
delete Function.prototype.bind;
fBound(1, 2, 3);
assert.deepEqual(slice([1, 2, 3, 4], 1, -1), [2, 3]);
```

32.3 Tests

Clone the repo, `npm install`, and run `npm test`

Chapter 33

call-me-maybe <a href="https://github.com/limulus/call-me-maybe/actions/workflows/continuous-release.yaml" ><img src="https://github.com/limulus/call-me-maybe/actions/workflows/continuous-release.↩ yaml/badge.svg" alt="Continuous Release"/></a>

Let your JS API users either give you a callback or receive a promise.

33.1 Usage

```
var maybe = require("call-me-maybe")
module.exports = function asyncFunc (cb) {
  return maybe(cb, new Promise(function(resolve, reject) {
    // ...
  }))
}
```

33.2 API

33.2.1 maybe(cb, promise)

If the callback `cb` is truthy, returns `undefined` and will call `cb` when `promise` is settled. The parameters passed to `cb` are standard error-first:

- If `promise` is fulfilled, then it is called with the result of the promise: `cb(null, result)`
- If `promise` is rejected, then it is called with the rejection error: `cb(err)`

If `cb` is falsey, then `promise` is returned.

Chapter 34

Changelog

All notable changes to this project will be documented in this file.
The format is based on Keep a Changelog and this project adheres to Semantic Versioning. (Format adopted after v3.0.0.)

34.1 [6.2.0] (2020-10-25)

34.1.1 Added

- added 'tsx' file extension for stand-alone executable subcommands (#1368)
- documented second parameter to `.description()` to describe command arguments (#1353)
- documentation of special cases with options taking varying numbers of option-arguments (#1332)
- documentation for terminology (#1361)

34.1.2 Fixed

- add missing TypeScript definition for '.addHelpCommand()' (#1375)
- removed blank line after "Arguments:" in help, to match "Options:" and "Commands:" (#1360)

34.1.3 Changed

- update dependencies

34.2 [6.1.0] (2020-08-28)

34.2.1 Added

- include URL to relevant section of README for error for potential conflict between Command properties and option values (#1306)
- `.combineFlagAndOptionalValue(false)` to ease upgrade path from older versions of Commander (#1326)
- allow disabling the built-in help option using `.helpOption(false)` (#1325)
- allow just some arguments in `argumentDescription` to `.description()` (#1323)

34.2.2 Changed

- tidy async test and remove lint override (#1312)

34.2.3 Fixed

- executable subcommand launching when script path not known (#1322)

34.3 [6.0.0] (2020-07-21)

34.3.1 Added

- add support for variadic options (#1250)
- allow options to be added with just a short flag (#1256)
  - *Breaking* the option property has same case as flag. e.g. flag `-n` accessed as `opts().n` (previously uppercase)
- *Breaking* throw an error if there might be a clash between option name and a Command property, with advice on how to resolve (#1275)

34.3.2 Fixed

- Options which contain -no- in the middle of the option flag should not be treated as negatable. (#1301)

34.4 [6.0.0-0] (2020-06-20)

(Released in 6.0.0)

34.5 [5.1.0] (2020-04-25)

34.5.1 Added

- support for multiple command aliases, the first of which is shown in the auto-generated help (#531, #1236)
- configuration support in `addCommand()` for `hidden` and `isDefault` (#1232)

34.5.2 Fixed

- omit masked help flags from the displayed help (#645, #1247)
- remove old short help flag when change help flags using `helpOption` (#1248)

34.5.3 Changed

- remove use of `arguments` to improve auto-generated help in editors (#1235)
- rename `.command()` configuration `noHelp` to `hidden` (but not remove old support) (#1232)
- improvements to documentation
- update dependencies
- update tested versions of node
- eliminate lint errors in TypeScript (#1208)

34.6 [5.0.0] (2020-03-14)

34.6.1 Added

- support for nested commands with action-handlers ([#1] [#764] #1149)
- `.addCommand()` for adding a separately configured command ([#764] #1149)
- allow a non-executable to be set as the default command ([#742] #1149)
- implicit help command when there are subcommands (previously only if executables) (#1149)
- customise implicit help command with `.addHelpCommand()` (#1149)
- display error message for unknown subcommand, by default ([#432] [#1088] #1149)
- display help for missing subcommand, by default ([#1088] #1149)
- combined short options as single argument may include boolean flags and value flag and value (e.g. `-a -b -p 80` can be written as `-abp80`) (#1145)
- `.parseOption()` includes short flag and long flag expansions (#1145)
- `.helpInformation()` returns help text as a string, previously a private routine (#1169)
- `.parse()` implicitly uses `process.argv` if arguments not specified (#1172)
- optionally specify where `.parse()` arguments "from", if not following node conventions ([#512] #1172)
- suggest help option along with unknown command error (#1179)
- TypeScript definition for `commands` property of `Command` (#1184)
- export `program` property (#1195)
- `createCommand` factory method to simplify subclassing (#1191)

34.6.2 Fixed

- preserve argument order in subcommands ([#508] [#962] #1138)
- do not emit `command:*` for executable subcommands ([#809] #1149)
- action handler called whether or not there are non-option arguments ([#1062] #1149)
- combining option short flag and value in single argument now works for subcommands (#1145)
- only add implicit help command when it will not conflict with other uses of argument ([#1153] #1149)
- implicit help command works with command aliases ([#948] #1149)
- options are validated whether or not there is an action handler (#1149)

34.6.3 Changed

- *Breaking* `.args` contains command arguments with just recognised options removed ([#1032] #1138)
- *Breaking* display error if required argument for command is missing ([#995] #1149)
- tighten TypeScript definition of custom option processing function passed to `.option()` (#1119)
- *Breaking* `.allowUnknownOption()` ([#802] #1138)
  - unknown options included in arguments passed to command action handler
  - unknown options included in `.args`
- only recognised option short flags and long flags are expanded (e.g. `-ab` or `--foo=bar`) (#1145)

- *Breaking* `.parseOptions()` (#1138)
  - `args` in returned result renamed `operands` and does not include anything after first unknown option
  - `unknown` in returned result has arguments after first unknown option including operands, not just options and values
- *Breaking* ‘.on('command:∗', callback)`and other command events passed (changed) results from`.parseOptions`, i.e. operands and unknown (#1138)
- refactor Option from prototype to class (#1133)
- refactor Command from prototype to class (#1159)
- changes to error handling (#1165)
  - throw for author error, not just display message
  - preflight for variadic error
  - add tips to missing subcommand executable
- TypeScript fluent return types changed to be more subclass friendly, return `this` rather than `Command` (#1180)
- `.parseAsync` returns `Promise<this>` to be consistent with `.parse()` (#1180)
- update dependencies

34.6.4 Removed

- removed EventEmitter from TypeScript definition for Command, eliminating implicit peer dependency on `@types/node` (#1146)
- removed private function `normalize` (the functionality has been integrated into `parseOptions`) (#1145)
- `parseExpectedArgs` is now private (#1149)

34.6.5 Migration Tips

If you use ‘.on('command:∗')` or more complicated tests to detect an unrecognised subcommand, you may be able to delete the code and rely on the default behaviour.
If you use `program.args` or more complicated tests to detect a missing subcommand, you may be able to delete the code and rely on the default behaviour.
If you use ‘.command('∗')`to add a default command, you may be be able to switch to`isDefault:true` with a named command.
If you want to continue combining short options with optional values as though they were boolean flags, set `combineFlagAndOptionalValue(false)` to expand `-fb` to `-f -b` rather than `-f b`.

34.7 [5.0.0-4] (2020-03-03)

(Released in 5.0.0)

34.8 [5.0.0-3] (2020-02-20)

(Released in 5.0.0)

34.9 [5.0.0-2] (2020-02-10)

(Released in 5.0.0)

34.10 [5.0.0-1] (2020-02-08)

(Released in 5.0.0)

34.11 [5.0.0-0] (2020-02-02)

(Released in 5.0.0)

34.12 [4.1.1] (2020-02-02)

34.12.1 Fixed

- TypeScript definition for `.action()` should include Promise for async (#1157)

34.13 [4.1.0] (2020-01-06)

34.13.1 Added

- two routines to change how option values are handled, and eliminate name clashes with command properties ([#933] #1102)
  - see storeOptionsAsProperties and passCommandToAction in README
- `.parseAsync` to use instead of `.parse` if supply async action handlers ([#806] #1118)

34.13.2 Fixed

- Remove trailing blanks from wrapped help text (#1096)

34.13.3 Changed

- update dependencies
- extend security coverage for Commander 2.x to 2020-02-03
- improvements to README
- improvements to TypeScript definition documentation
- move old versions out of main CHANGELOG
- removed explicit use of `ts-node` in tests

34.14 [4.0.1] (2019-11-12)

34.14.1 Fixed

- display help when requested, even if there are missing required options (#1091)

34.15 [4.0.0] (2019-11-02)

34.15.1 Added

- automatically wrap and indent help descriptions for options and commands (#1051)
- `.exitOverride()` allows override of calls to `process.exit` for additional error handling and to keep program running (#1040)
- support for declaring required options with `.requiredOptions()` (#1071)

- GitHub Actions support (#1027)
- translation links in README

34.15.2 Changed

- dev: switch tests from Sinon+Should to Jest with major rewrite of tests (#1035)
- call default subcommand even when there are unknown options (#1047)
- *Breaking* Commander is only officially supported on Node 8 and above, and requires Node 6 (#1053)

34.15.3 Fixed

- *Breaking* keep command object out of program.args when action handler called (#1048)
  - also, action handler now passed array of unknown arguments
- complain about unknown options when program argument supplied and action handler (#1049)
  - this changes parameters to `command:*` event to include unknown arguments
- removed deprecated `customFds` option from call to `child_process.spawn` (#1052)
- rework TypeScript declarations to bring all types into imported namespace (#1081)

34.15.4 Migration Tips

34.15.4.1 Testing for no arguments

If you were previously using code like:
```
if (!program.args.length) ...
```
a partial replacement is:
```
if (program.rawArgs.length < 3) ...
```

34.16 <a href="https://github.com/tj/commander.js/compare/v4.0.0-0..↩ v4.0.0-1" >4.0.0-1</a> Prerelease (2019-10-08)

(Released in 4.0.0)

34.17 <a href="https://github.com/tj/commander.js/compare/v3.0.2...↩ v4.0.0-0" >4.0.0-0</a> Prerelease (2019-10-01)

(Released in 4.0.0)

34.18 Older versions

- 3.x
- 2.x
- 1.x
- 0.x

Chapter 35

Commander.js

The complete solution for `node.js` command-line interfaces.
Read this in other languages: English |

For information about terms used in this document see: terminology

35.1 Installation

```
npm install commander
```

35.2 Declaring <em>program</em> variable

Commander exports a global object which is convenient for quick programs. This is used in the examples in this README for brevity.

```
const { program } = require('commander');
program.version('0.0.1');
```

For larger programs which may use commander in multiple ways, including unit testing, it is better to create a local Command object to use.

```
const { Command } = require('commander');
const program = new Command();
program.version('0.0.1');
```

35.3 Options

Options are defined with the `.option()` method, also serving as documentation for the options. Each option can have a short flag (single character) and a long name, separated by a comma or space or vertical bar ('|').

The options can be accessed as properties on the Command object. Multi-word options such as "--template-engine" are camel-cased, becoming `program.templateEngine` etc. See also optional new behaviour to avoid name clashes.

Multiple short flags may optionally be combined in a single argument following the dash: boolean flags, followed by a single option taking a value (possibly followed by the value). For example `-a -b -p 80` may be written as `-ab -p80` or even `-abp80`.

You can use `--` to indicate the end of the options, and any remaining arguments will be used without being interpreted.

Options on the command line are not positional, and can be specified before or after other arguments.

35.3.1 Common option types, boolean and value

The two most used option types are a boolean option, and an option which takes its value from the following argument (declared with angle brackets like `--expect <value>`). Both are `undefined` unless specified on command line.

Example file: `options-common.js`

```
program
  .option('-d, --debug', 'output extra debugging')
  .option('-s, --small', 'small pizza size')
  .option('-p, --pizza-type <type>', 'flavour of pizza');
program.parse(process.argv);
if (program.debug) console.log(program.opts());
console.log('pizza details:');
if (program.small) console.log('- small pizza size');
if (program.pizzaType) console.log(`- ${program.pizzaType}`);
```

```
$ pizza-options -d
{ debug: true, small: undefined, pizzaType: undefined }
pizza details:
$ pizza-options -p
error: option '-p, --pizza-type <type>' argument missing
$ pizza-options -ds -p vegetarian
{ debug: true, small: true, pizzaType: 'vegetarian' }
pizza details:
- small pizza size
- vegetarian
$ pizza-options --pizza-type=cheese
pizza details:
- cheese
```

`program.parse(arguments)` processes the arguments, leaving any args not consumed by the program options in the `program.args` array.

35.3.2 Default option value

You can specify a default value for an option which takes a value.
Example file: `options-defaults.js`

```
program
  .option('-c, --cheese <type>', 'add the specified type of cheese', 'blue');
program.parse(process.argv);
console.log(`cheese: ${program.cheese}`);
$ pizza-options
cheese: blue
$ pizza-options --cheese stilton
cheese: stilton
```

35.3.3 Other option types, negatable boolean and boolean|value

You can define a boolean option long name with a leading `no-` to set the option value to false when used. Defined alone this also makes the option true by default.
If you define `--foo` first, adding `--no-foo` does not change the default value from what it would otherwise be. You can specify a default boolean value for a boolean option and it can be overridden on command line.
Example file: `options-negatable.js`

```
program
  .option('--no-sauce', 'Remove sauce')
  .option('--cheese <flavour>', 'cheese flavour', 'mozzarella')
  .option('--no-cheese', 'plain with no cheese')
  .parse(process.argv);
const sauceStr = program.sauce ? 'sauce' : 'no sauce';
const cheeseStr = (program.cheese === false) ? 'no cheese' : `${program.cheese} cheese`;
console.log(`You ordered a pizza with ${sauceStr} and ${cheeseStr}`);
$ pizza-options
You ordered a pizza with sauce and mozzarella cheese
$ pizza-options --sauce
error: unknown option '--sauce'
$ pizza-options --cheese=blue
You ordered a pizza with sauce and blue cheese
$ pizza-options --no-sauce --no-cheese
You ordered a pizza with no sauce and no cheese
```

You can specify an option which may be used as a boolean option but may optionally take an option-argument (declared with square brackets like `--optional [value]`).
Example file: `options-boolean-or-value.js`

```
program
  .option('-c, --cheese [type]', 'Add cheese with optional type');
program.parse(process.argv);
if (program.cheese === undefined) console.log('no cheese');
else if (program.cheese === true) console.log('add cheese');
else console.log(`add cheese type ${program.cheese}`);
$ pizza-options
no cheese
$ pizza-options --cheese
add cheese
$ pizza-options --cheese mozzarella
add cheese type mozzarella
```

For information about possible ambiguous cases, see options taking varying arguments.

35.3.4 Custom option processing

You may specify a function to do custom processing of option-arguments. The callback function receives two parameters, the user specified option-argument and the previous value for the option. It returns the new value for the option.
This allows you to coerce the option-argument to the desired type, or accumulate values, or do entirely custom processing.
You can optionally specify the default/starting value for the option after the function parameter.
Example file: `options-custom-processing.js`

```
function myParseInt(value, dummyPrevious) {
  // parseInt takes a string and an optional radix
  return parseInt(value);
}
function increaseVerbosity(dummyValue, previous) {
  return previous + 1;
}
function collect(value, previous) {
  return previous.concat([value]);
}
function commaSeparatedList(value, dummyPrevious) {
  return value.split(',');
}
program
  .option('-f, --float <number>', 'float argument', parseFloat)
  .option('-i, --integer <number>', 'integer argument', myParseInt)
  .option('-v, --verbose', 'verbosity that can be increased', increaseVerbosity, 0)
  .option('-c, --collect <value>', 'repeatable value', collect, [])
  .option('-l, --list <items>', 'comma separated list', commaSeparatedList)
;
program.parse(process.argv);
if (program.float !== undefined) console.log(`float:  ${program.float}`);
if (program.integer !== undefined) console.log(`integer:  ${program.integer}`);
if (program.verbose > 0) console.log(`verbosity:  ${program.verbose}`);
if (program.collect.length > 0) console.log(program.collect);
if (program.list !== undefined) console.log(program.list);
$ custom -f 1e2
float:  100
$ custom --integer 2
integer:  2
$ custom -v -v -v
verbose:  3
$ custom -c a -c b -c c
[ 'a', 'b', 'c' ]
$ custom --list x,y,z
[ 'x', 'y', 'z' ]
```

35.3.5 Required option

You may specify a required (mandatory) option using `.requiredOption`. The option must have a value after parsing, usually specified on the command line, or perhaps from a default value (say from environment). The method is otherwise the same as `.option` in format, taking flags and description, and optional default value or custom processing.

Example file: `options-required.js`

```
program
  .requiredOption('-c, --cheese <type>', 'pizza must have cheese');
program.parse(process.argv);
$ pizza
error:  required option '-c, --cheese <type>' not specified
```

35.3.6 Variadic option

You may make an option variadic by appending `...` to the value placeholder when declaring the option. On the command line you can then specify multiple option-arguments, and the parsed option value will be an array. The extra arguments are read until the first argument starting with a dash. The special argument `--` stops option processing entirely. If a value is specified in the same argument as the option then no further values are read.

Example file: `options-variadic.js`

```
program
  .option('-n, --number <numbers...>', 'specify numbers')
  .option('-l, --letter [letters...]', 'specify letters');
program.parse();
console.log('Options:  ', program.opts());
console.log('Remaining arguments:  ', program.args);
$ collect -n 1 2 3 --letter a b c
Options:    { number:  [ '1', '2', '3' ], letter:  [ 'a', 'b', 'c' ] }
Remaining arguments:    []
$ collect --letter=A -n80 operand
Options:    { number:  [ '80' ], letter:  [ 'A' ] }
Remaining arguments:    [ 'operand' ]
$ collect --letter -n 1 -n 2 3 -- operand
Options:    { number:  [ '1', '2', '3' ], letter:  true }
Remaining arguments:    [ 'operand' ]
```

For information about possible ambiguous cases, see options taking varying arguments.

35.3.7 Version option

The optional `version` method adds handling for displaying the command version. The default option flags are `-V` and `--version`, and when present the command prints the version number and exits.

```
program.version('0.0.1');
$ ./examples/pizza -V
0.0.1
```

You may change the flags and description by passing additional parameters to the `version` method, using the same syntax for flags as the `option` method.

```
program.version('0.0.1', '-v, --vers', 'output the current version');
```

35.4 Commands

You can specify (sub)commands using `.command()` or `.addCommand()`. There are two ways these can be implemented: using an action handler attached to the command, or as a stand-alone executable file (described in more detail later). The subcommands may be nested (`example`).
In the first parameter to `.command()` you specify the command name and any command-arguments. The arguments may be `<required>` or `[optional]`, and the last argument may also be `variadic...`.
You can use `.addCommand()` to add an already configured subcommand to the program.
For example:

```
// Command implemented using action handler (description is supplied separately to `.command`)
// Returns new command for configuring.
program
  .command('clone <source> [destination]')
  .description('clone a repository into a newly created directory')
  .action((source, destination) => {
    console.log('clone command called');
  });
// Command implemented using stand-alone executable file (description is second parameter to `.command`)
// Returns `this` for adding more commands.
program
  .command('start <service>', 'start named service')
  .command('stop [service]', 'stop named service, or all if no name supplied');
// Command prepared separately.
// Returns `this` for adding more commands.
program
  .addCommand(build.makeBuildCommand());
```

Configuration options can be passed with the call to `.command()` and `.addCommand()`. Specifying `hidden: true` will remove the command from the generated help output. Specifying `isDefault: true` will run the subcommand if no other subcommand is specified (`example`).

35.4.1 Specify the argument syntax

You use `.arguments` to specify the expected command-arguments for the top-level command, and for subcommands they are usually included in the `.command` call. Angled brackets (e.g. `<required>`) indicate required command-arguments. Square brackets (e.g. `[optional]`) indicate optional command-arguments. You can optionally describe the arguments in the help by supplying a hash as second parameter to `.description()`.
Example file: `env`

```
program
  .version('0.1.0')
  .arguments('<cmd> [env]')
  .description('test command', {
    cmd: 'command to run',
    env: 'environment to run test in'
  })
  .action(function (cmd, env) {
    console.log('command:', cmd);
    console.log('environment:', env || 'no environment given');
  });
program.parse(process.argv);
```

The last argument of a command can be variadic, and only the last argument. To make an argument variadic you append `...` to the argument name. For example:

```
const { program } = require('commander');
program
  .version('0.1.0')
  .command('rmdir <dir> [otherDirs...]')
  .action(function (dir, otherDirs) {
    console.log('rmdir %s', dir);
    if (otherDirs) {
      otherDirs.forEach(function (oDir) {
        console.log('rmdir %s', oDir);
      });
    }
  });
program.parse(process.argv);
```

The variadic argument is passed to the action handler as an array.

35.4.2 Action handler (sub)commands

You can add options to a command that uses an action handler. The action handler gets passed a parameter for each argument you declared, and one additional argument which is the command object itself. This command argument has the values for the command-specific options added as properties.

```
const { program } = require('commander');
program
  .command('rm <dir>')
  .option('-r, --recursive', 'Remove recursively')
  .action(function (dir, cmdObj) {
    console.log('remove ' + dir + (cmdObj.recursive ? ' recursively' : ''))
  })
program.parse(process.argv)
```

You may supply an `async` action handler, in which case you call `.parseAsync` rather than `.parse`.

```
async function run() { /* code goes here */ }
async function main() {
  program
    .command('run')
    .action(run);
  await program.parseAsync(process.argv);
}
```

A command's options on the command line are validated when the command is used. Any unknown options will be reported as an error.

35.4.3 Stand-alone executable (sub)commands

When `.command()` is invoked with a description argument, this tells Commander that you're going to use stand-alone executables for subcommands. Commander will search the executables in the directory of the entry script (like `./examples/pm`) with the name `program-subcommand`, like `pm-install`, `pm-search`. You can specify a custom name with the `executableFile` configuration option.
You handle the options for an executable (sub)command in the executable, and don't declare them at the top-level.
Example file: `pm`

```
program
  .version('0.1.0')
  .command('install [name]', 'install one or more packages')
  .command('search [query]', 'search with optional query')
  .command('update', 'update installed packages', { executableFile: 'myUpdateSubCommand' })
  .command('list', 'list packages installed', { isDefault: true });
program.parse(process.argv);
```

If the program is designed to be installed globally, make sure the executables have proper modes, like `755`.

35.5 Automated help

The help information is auto-generated based on the information commander already knows about your program.
The default help option is `-h,--help`.
Example file: `pizza`

```
$ node ./examples/pizza --help
Usage: pizza [options]
An application for pizzas ordering
Options:
  -V, --version        output the version number
  -p, --peppers        Add peppers
  -c, --cheese <type>  Add the specified type of cheese (default: "marble")
  -C, --no-cheese      You do not want any cheese
  -h, --help           display help for command
```

A `help` command is added by default if your command has subcommands. It can be used alone, or with a subcommand name to show further help for the subcommand. These are effectively the same if the `shell` program has implicit help:

```
shell help
shell --help
shell help spawn
shell spawn --help
```

35.5.1 Custom help

You can display extra information by listening for "--help".
Example file: `custom-help`

```
program
  .option('-f, --foo', 'enable some foo');
// must be before .parse()
program.on('--help', () => {
  console.log('');
  console.log('Example call:');
  console.log('  $ custom-help --help');
});
```

Yields the following help output:

```
Usage:  custom-help [options]
Options:
  -f, --foo   enable some foo
  -h, --help  display help for command
Example call:
  $ custom-help --help
```

35.5.2 .usage and .name

These allow you to customise the usage description in the first line of the help. The name is otherwise deduced from the (full) program arguments. Given:

```
program
  .name("my-command")
  .usage("[global options] command")
```

The help will start with:

```
Usage:  my-command [global options] command
```

35.5.3 .help(cb)

Output help information and exit immediately. Optional callback cb allows post-processing of help text before it is displayed.

35.5.4 .outputHelp(cb)

Output help information without exiting. Optional callback cb allows post-processing of help text before it is displayed.

35.5.5 .helpInformation()

Get the command help information as a string for processing or displaying yourself. (The text does not include the custom help from `--help` listeners.)

35.5.6 .helpOption(flags, description)

Override the default help flags and description. Pass false to disable the built-in help option.

```
program
  .helpOption('-e, --HELP', 'read more information');
```

35.5.7 .addHelpCommand()

You can explicitly turn on or off the implicit help command with `.addHelpCommand()` and `.addHelp↩ Command(false).`
You can both turn on and customise the help command by supplying the name and description:

```
program.addHelpCommand('assist [command]', 'show assistance');
```

35.6 Custom event listeners

You can execute custom actions by listening to command and option events.

```
program.on('option:verbose', function () {
  process.env.VERBOSE = this.verbose;
});
program.on('command:*', function (operands) {
  console.error(`error:  unknown command '${operands[0]}'`);
  const availableCommands = program.commands.map(cmd => cmd.name());
  mySuggestBestMatch(operands[0], availableCommands);
  process.exitCode = 1;
});
```

35.7 Bits and pieces

35.7.1 .parse() and .parseAsync()

The first argument to `.parse` is the array of strings to parse. You may omit the parameter to implicitly use `process.argv`.

If the arguments follow different conventions than node you can pass a `from` option in the second parameter:

- 'node': default, `argv[0]` is the application and `argv[1]` is the script being run, with user parameters after that
- 'electron': `argv[1]` varies depending on whether the electron application is packaged
- 'user': all of the arguments from the user

For example:

```
program.parse(process.argv); // Explicit, node conventions
program.parse(); // Implicit, and auto-detect electron
program.parse(['-f', 'filename'], { from:  'user' });
```

35.7.2 Avoiding option name clashes

The original and default behaviour is that the option values are stored as properties on the program, and the action handler is passed a command object with the options values stored as properties. This is very convenient to code, but the downside is possible clashes with existing properties of Command.
There are two new routines to change the behaviour, and the default behaviour may change in the future:

- `storeOptionsAsProperties`: whether to store option values as properties on command object, or store separately (specify false) and access using `.opts()`
- `passCommandToAction`: whether to pass command to action handler, or just the options (specify false)

Example file: `storeOptionsAsProperties-action.js`

```
program
  .storeOptionsAsProperties(false)
  .passCommandToAction(false);
program
  .name('my-program-name')
  .option('-n,--name <name>');
program
  .command('show')
  .option('-a,--action <action>')
  .action((options) => {
    console.log(options.action);
  });
program.parse(process.argv);
const programOptions = program.opts();
console.log(programOptions.name);
```

35.7.3 TypeScript

The Commander package includes its TypeScript Definition file.
If you use `ts-node` and stand-alone executable subcommands written as `.ts` files, you need to call your program through node to get the subcommands called correctly. e.g.

```
node -r ts-node/register pm.ts
```

35.7.4 createCommand()

This factory function creates a new command. It is exported and may be used instead of using `new`, like:

```
const { createCommand } = require('commander');
const program = createCommand();
```

`createCommand` is also a method of the Command object, and creates a new command rather than a subcommand. This gets used internally when creating subcommands using `.command()`, and you may override it to customise the new subcommand (examples using `subclass` and `function`).

35.7.5 Import into ECMAScript Module

Commander is currently a CommonJS package, and the default export can be imported into an ES Module:

```
// index.mjs
import commander from 'commander';
const program = commander.program;
const newCommand = new commander.Command();
```

35.7.6 Node options such as <tt>--harmony</tt>

You can enable `--harmony` option in two ways:

- Use `#! /usr/bin/env node --harmony` in the subcommands scripts. (Note Windows does not support this pattern.)
- Use the `--harmony` option when call the command, like `node --harmony examples/pm publish`. The `--harmony` option will be preserved when spawning subcommand process.

35.7.7 Debugging stand-alone executable subcommands

An executable subcommand is launched as a separate child process.
If you are using the node inspector for `debugging` executable subcommands using `node --inspect` et al, the inspector port is incremented by 1 for the spawned subcommand.
If you are using VSCode to debug executable subcommands you need to set the `"autoAttachChild↩ Processes": true` flag in your launch.json configuration.

35.7.8 Override exit handling

By default Commander calls `process.exit` when it detects errors, or after displaying the help or version. You can override this behaviour and optionally supply a callback. The default override throws a `CommanderError`.
The override callback is passed a `CommanderError` with properties `exitCode` number, `code` string, and `message`. The default override behaviour is to throw the error, except for async handling of executable subcommand completion which carries on. The normal display of error messages or version or help is not affected by the override which is called after the display.

```
program.exitOverride();
try {
  program.parse(process.argv);
} catch (err) {
  // custom processing...
}
```

35.8 Examples

Example file: `deploy`

```
const { program } = require('commander');
program
  .version('0.1.0')
  .option('-C, --chdir <path>', 'change the working directory')
  .option('-c, --config <path>', 'set config path.  defaults to ./deploy.conf')
  .option('-T, --no-tests', 'ignore test hook');
program
  .command('setup [env]')
  .description('run setup commands for all envs')
  .option("-s, --setup_mode [mode]", "Which setup mode to use")
  .action(function(env, options){
    const mode = options.setup_mode || "normal";
    env = env || 'all';
    console.log('setup for %s env(s) with %s mode', env, mode);
  });
program
  .command('exec <cmd>')
  .alias('ex')
  .description('execute the given remote cmd')
  .option("-e, --exec_mode <mode>", "Which exec mode to use")
  .action(function(cmd, options){
    console.log('exec "%s" using %s mode', cmd, options.exec_mode);
  }).on('--help', function() {
    console.log('');
    console.log('Examples:');
    console.log('');
    console.log('  $ deploy exec sequential');
    console.log('  $ deploy exec async');
  });
program.parse(process.argv);
```

More Demos can be found in the `examples` directory.

35.9 Support

The current version of Commander is fully supported on Long Term Support versions of Node, and is likely to work with Node 6 but not tested. (For versions of Node below Node 6, use Commander 3.x or 2.x.)
The main forum for free and community support is the project `Issues` on GitHub.

35.9.1 Commander for enterprise

Available as part of the Tidelift Subscription
The maintainers of Commander and thousands of other packages are working with Tidelift to deliver commercial support and maintenance for the open source dependencies you use to build your applications. Save time, reduce risk, and improve code health, while paying the maintainers of the exact dependencies you use. `Learn more.`

Chapter 36

concat-map

Concatenative mapdashery.

36.1 example

```
var concatMap = require('concat-map');
var xs = [ 1, 2, 3, 4, 5, 6 ];
var ys = concatMap(xs, function (x) {
    return x % 2 ?  [ x - 0.1, x, x + 0.1 ] :  [];
});
console.dir(ys);
```

```
[ 0.9, 1, 1.1, 2.9, 3, 3.1, 4.9, 5, 5.1 ]
```

36.2 methods

```
var concatMap = require('concat-map')
```

36.2.1 concatMap(xs, fn)

Return an array of concatenated elements by calling `fn(x, i)` for each element `x` and each index `i` in the array `xs`.

When `fn(x, i)` returns an array, its result will be concatenated with the result array. If `fn(x, i)` returns anything else, that value will be pushed onto the end of the result array.

36.3 install

With `npm` do:

```
npm install concat-map
```

36.4 license

MIT

36.5 notes

This module was written while sitting high above the ground in a tree.

Chapter 37

0.5.4 / 2021-12-10

- deps: safe-buffer@5.2.1

37.1 0.5.3 / 2018-12-17

- Use `safe-buffer` for improved Buffer API

37.2 0.5.2 / 2016-12-08

- Fix `parse` to accept any linear whitespace character

37.3 0.5.1 / 2016-01-17

- perf: enable strict mode

37.4 0.5.0 / 2014-10-11

- Add `parse` function

37.5 0.4.0 / 2014-09-21

- Expand non-Unicode `filename` to the full ISO-8859-1 charset

37.6 0.3.0 / 2014-09-20

- Add `fallback` option
- Add `type` option

37.7 0.2.0 / 2014-09-19

- Reduce ambiguity of file names with hex escape in buggy browsers

37.8 0.1.2 / 2014-09-19

- Fix periodic invalid Unicode filename header

37.9 0.1.1 / 2014-09-19

- Fix invalid characters appearing in `filename*` parameter

37.10 0.1.0 / 2014-09-18

- Make the `filename` argument optional

37.11 0.0.0 / 2014-09-18

- Initial release

Chapter 38

content-disposition

Create and parse HTTP `Content-Disposition` header

38.1 Installation

```
$ npm install content-disposition
```

38.2 API

```
var contentDisposition = require('content-disposition')
```

38.2.1 contentDisposition(filename, options)

Create an attachment `Content-Disposition` header value using the given file name, if supplied. The `filename` is optional and if no file name is desired, but you want to specify `options`, set `filename` to `undefined`.

```
res.setHeader('Content-Disposition', contentDisposition(' maths.pdf'))
```

note HTTP headers are of the ISO-8859-1 character set. If you are writing this header through a means different from `setHeader` in Node.js, you'll want to specify the ''binary'' encoding in Node.js.

38.2.1.1 Options

`contentDisposition` accepts these properties in the options object.

38.2.1.1.1 fallback If the `filename` option is outside ISO-8859-1, then the file name is actually stored in a supplemental field for clients that support Unicode file names and a ISO-8859-1 version of the file name is automatically generated.
This specifies the ISO-8859-1 file name to override the automatic generation or disables the generation all together, defaults to `true`.

- A string will specify the ISO-8859-1 file name to use in place of automatic generation.
- `false` will disable including a ISO-8859-1 file name and only include the Unicode version (unless the file name is already ISO-8859-1).
- `true` will enable automatic generation if the file name is outside ISO-8859-1.

If the `filename` option is ISO-8859-1 and this option is specified and has a different value, then the `filename` option is encoded in the extended field and this set as the fallback field, even though they are both ISO-8859-1.

38.2.1.1.2 type Specifies the disposition type, defaults to `"attachment"`. This can also be `"inline"`, or any other value (all values except inline are treated like `attachment`, but can convey additional information if both parties agree to it). The type is normalized to lower-case.

38.2.2 contentDisposition.parse(string)

```
var disposition = contentDisposition.parse('attachment; filename="EURO rates.txt";
      filename*=UTF-8\'\'%e2%82%ac%20rates.txt')
```

Parse a `Content-Disposition` header string. This automatically handles extended ("Unicode") parameters by decoding them and providing them under the standard parameter name. This will return an object with the following properties (examples are shown for the string “attachment; filename="EURO rates.txt"; filename∗=UTF-8&rsquo;\'e2%82ac%20rates.txt`):

- `type`: The disposition type (always lower case). Example: “attachment`
- `parameters`: An object of the parameters in the disposition (name of parameter always lower case and extended versions replace non-extended versions). Example: `{filename: "€ rates.txt"}`

38.3 Examples

38.3.1 Send a file for download

```
var contentDisposition = require('content-disposition')
var destroy = require('destroy')
var fs = require('fs')
var http = require('http')
var onFinished = require('on-finished')
var filePath = '/path/to/public/plans.pdf'
http.createServer(function onRequest (req, res) {
  // set headers
  res.setHeader('Content-Type', 'application/pdf')
  res.setHeader('Content-Disposition', contentDisposition(filePath))
  // send file
  var stream = fs.createReadStream(filePath)
  stream.pipe(res)
  onFinished(res, function () {
    destroy(stream)
  })
})
```

38.4 Testing

```
$ npm test
```

38.5 References

- RFC 2616: Hypertext Transfer Protocol - HTTP/1.1
- RFC 5987: Character Set and Language Encoding for Hypertext Transfer Protocol (HTTP) Header Field Parameters
- RFC 6266: Use of the Content-Disposition Header Field in the Hypertext Transfer Protocol (HTTP)
- Test Cases for HTTP Content-Disposition header field (RFC 6266) and the Encodings defined in RFCs 2047, 2231 and 5987

38.6 License

[MIT](LICENSE)

Chapter 39

1.0.5 / 2023-01-29

- perf: skip value escaping when unnecessary

39.1 1.0.4 / 2017-09-11

- perf: skip parameter parsing when no parameters

39.2 1.0.3 / 2017-09-10

- perf: remove argument reassignment

39.3 1.0.2 / 2016-05-09

- perf: enable strict mode

39.4 1.0.1 / 2015-02-13

- Improve missing `Content-Type` header error message

39.5 1.0.0 / 2015-02-01

- Initial implementation, derived from `media-typer@0.3.0`

Chapter 40

content-type

Create and parse HTTP Content-Type header according to RFC 7231

40.1 Installation

```
$ npm install content-type
```

40.2 API

```
var contentType = require('content-type')
```

40.2.1 contentType.parse(string)

```
var obj = contentType.parse('image/svg+xml; charset=utf-8')
```

Parse a `Content-Type` header. This will return an object with the following properties (examples are shown for the string ‘‘image/svg+xml; charset=utf-8`‘):

- `type`: The media type (the type and subtype, always lower case). Example: ‘‘image/svg+xml’‘
- `parameters`: An object of the parameters in the media type (name of parameter always lower case). Example: ‘{charset: 'utf-8'}`

Throws a `TypeError` if the string is missing or invalid.

40.2.2 contentType.parse(req)

```
var obj = contentType.parse(req)
```

Parse the `Content-Type` header from the given `req`. Short-cut for ‘contentType.parse(req.headers['content-type'])`.
Throws a `TypeError` if the `Content-Type` header is missing or invalid.

40.2.3 contentType.parse(res)

```
var obj = contentType.parse(res)
```

Parse the `Content-Type` header set on the given `res`. Short-cut for ‘contentType.parse(res.getHeader('content-type'))`.
Throws a `TypeError` if the `Content-Type` header is missing or invalid.

40.2.4 contentType.format(obj)

```
var str = contentType.format({
  type: 'image/svg+xml',
  parameters: { charset: 'utf-8' }
})
```

Format an object into a `Content-Type` header. This will return a string of the content type for the given object with the following properties (examples are shown that produce the string ‘‘image/svg+xml; charset=utf-8`‘):

- `type`: The media type (will be lower-cased). Example: ''image/svg+xml'`
- `parameters`: An object of the parameters in the media type (name of the parameter will be lower-cased). Example: '{charset: 'utf-8'}`

Throws a `TypeError` if the object contains an invalid type or parameter names.

40.3 License

[MIT](LICENSE)

Chapter 41

1.0.6 / 2015-02-03

- use `npm test` instead of `make test` to run tests
- clearer assertion messages when checking input

41.1 1.0.5 / 2014-09-05

- add license to package.json

41.2 1.0.4 / 2014-06-25

- corrected avoidance of timing attacks (thanks @tenbits!)

41.3 1.0.3 / 2014-01-28

- [incorrect] fix for timing attacks

41.4 1.0.2 / 2014-01-28

- fix missing repository warning
- fix typo in test

41.5 1.0.1 / 2013-04-15

- Revert "Changed underlying HMAC algo. to sha512."
- Revert "Fix for timing attacks on MAC verification."

41.6 0.0.1 / 2010-01-03

- Initial release

Chapter 42

cookie-signature

Sign and unsign cookies.

42.1 Example

```
var cookie = require('cookie-signature');
var val = cookie.sign('hello', 'tobiiscool');
val.should.equal('hello.DGDUkGlIkCzPz+C0B064FNgHdEjox7ch8tOBGslZ5QI');
var val = cookie.sign('hello', 'tobiiscool');
cookie.unsign(val, 'tobiiscool').should.equal('hello');
cookie.unsign(val, 'luna').should.be.false;
```

42.2 License

Chapter 43

0.6.0 / 2023-11-06

- Add `partitioned` option

43.1 0.5.0 / 2022-04-11

- Add `priority` option
- Fix `expires` option to reject invalid dates
- perf: improve default decode speed
- perf: remove slow string split in parse

43.2 0.4.2 / 2022-02-02

- perf: read value only when assigning in parse
- perf: remove unnecessary regexp in parse

43.3 0.4.1 / 2020-04-21

- Fix `maxAge` option to reject invalid values

43.4 0.4.0 / 2019-05-15

- Add `SameSite=None` support

43.5 0.3.1 / 2016-05-26

- Fix `sameSite: true` to work with draft-7 clients
  - `true` now sends `SameSite=Strict` instead of `SameSite`

43.6 0.3.0 / 2016-05-26

- Add `sameSite` option
  - Replaces `firstPartyOnly` option, never implemented by browsers
- Improve error message when `encode` is not a function
- Improve error message when `expires` is not a `Date`

43.7 0.2.4 / 2016-05-20

- perf: enable strict mode
- perf: use for loop in parse
- perf: use string concatenation for serialization

43.8 0.2.3 / 2015-10-25

- Fix cookie `Max-Age` to never be a floating point number

43.9 0.2.2 / 2015-09-17

- Fix regression when setting empty cookie value
  - Ease the new restriction, which is just basic header-level validation
- Fix typo in invalid value errors

43.10 0.2.1 / 2015-09-17

- Throw on invalid values provided to `serialize`
  - Ensures the resulting string is a valid HTTP header value

43.11 0.2.0 / 2015-08-13

- Add `firstPartyOnly` option
- Throw better error for invalid argument to parse
- perf: hoist regular expression

43.12 0.1.5 / 2015-09-17

- Fix regression when setting empty cookie value
  - Ease the new restriction, which is just basic header-level validation
- Fix typo in invalid value errors

43.13 0.1.4 / 2015-09-17

- Throw better error for invalid argument to parse
- Throw on invalid values provided to `serialize`
  - Ensures the resulting string is a valid HTTP header value

43.14 0.1.3 / 2015-05-19

- Reduce the scope of try-catch deopt
- Remove argument reassignments

43.15 0.1.2 / 2014-04-16

- Remove unnecessary files from npm package

43.16 0.1.1 / 2014-02-23

- Fix bad parse when cookie value contained a comma
- Fix support for `maxAge` of `0`

43.17 0.1.0 / 2013-05-01

- Add `decode` option
- Add `encode` option

43.18 0.0.6 / 2013-04-08

- Ignore cookie parts missing =

43.19 0.0.5 / 2012-10-29

- Return raw cookie value if value unescape errors

43.20 0.0.4 / 2012-06-21

- Use encode/decodeURIComponent for cookie encoding/decoding
  - Improve server/client interoperability

43.21 0.0.3 / 2012-06-06

- Only escape special characters per the cookie RFC

43.22 0.0.2 / 2012-06-01

- Fix `maxAge` option to not throw error

43.23 0.0.1 / 2012-05-28

- Add more tests

43.24 0.0.0 / 2012-05-28

- Initial release

Chapter 44

cookie

Basic HTTP cookie parser and serializer for HTTP servers.

44.1 Installation

This is a Node.js module available through the npm registry. Installation is done using the npm install command:

```
$ npm install cookie
```

44.2 API

```
var cookie = require('cookie');
```

44.2.1 cookie.parse(str, options)

Parse an HTTP `Cookie` header string and returning an object of all cookie name-value pairs. The `str` argument is the string representing a `Cookie` header value and `options` is an optional object containing additional parsing options.

```
var cookies = cookie.parse('foo=bar; equation=E%3Dmc%5E2');
// { foo: 'bar', equation: 'E=mc^2' }
```

44.2.1.1 Options

`cookie.parse` accepts these properties in the options object.

44.2.1.1.1 decode Specifies a function that will be used to decode a cookie's value. Since the value of a cookie has a limited character set (and must be a simple string), this function can be used to decode a previously-encoded cookie value into a JavaScript string or other object.

The default function is the global `decodeURIComponent`, which will decode any URL-encoded sequences into their byte representations.

note if an error is thrown from this function, the original, non-decoded cookie value will be returned as the cookie's value.

44.2.2 cookie.serialize(name, value, options)

Serialize a cookie name-value pair into a `Set-Cookie` header string. The `name` argument is the name for the cookie, the `value` argument is the value to set the cookie to, and the `options` argument is an optional object containing additional serialization options.

```
var setCookie = cookie.serialize('foo', 'bar');
// foo=bar
```

44.2.2.1 Options

`cookie.serialize` accepts these properties in the options object.

44.2.2.1.1 domain Specifies the value for the `Domain Set-Cookie attribute`. By default, no domain is set, and most clients will consider the cookie to apply to only the current domain.

44.2.2.1.2 encode Specifies a function that will be used to encode a cookie's value. Since value of a cookie has a limited character set (and must be a simple string), this function can be used to encode a value into a string suited for a cookie's value.
The default function is the global `encodeURIComponent`, which will encode a JavaScript string into UTF-8 byte sequences and then URL-encode any that fall outside of the cookie range.

44.2.2.1.3 expires Specifies the `Date` object to be the value for the `Expires Set-Cookie attribute`. By default, no expiration is set, and most clients will consider this a "non-persistent cookie" and will delete it on a condition like exiting a web browser application.
note the `cookie storage model specification` states that if both `expires` and `maxAge` are set, then `maxAge` takes precedence, but it is possible not all clients by obey this, so if both are set, they should point to the same date and time.

44.2.2.1.4 httpOnly Specifies the `boolean` value for the `HttpOnly Set-Cookie attribute`. When truthy, the `HttpOnly` attribute is set, otherwise it is not. By default, the `HttpOnly` attribute is not set.
note be careful when setting this to `true`, as compliant clients will not allow client-side JavaScript to see the cookie in `document.cookie`.

44.2.2.1.5 maxAge Specifies the `number` (in seconds) to be the value for the `Max-Age Set-Cookie attribute`. The given number will be converted to an integer by rounding down. By default, no maximum age is set.
note the `cookie storage model specification` states that if both `expires` and `maxAge` are set, then `maxAge` takes precedence, but it is possible not all clients by obey this, so if both are set, they should point to the same date and time.

44.2.2.1.6 partitioned Specifies the `boolean` value for the [`Partitioned` `Set-Cookie`](rfc-cutler-httpbis-partitioned-cookies) attribute. When truthy, the `Partitioned` attribute is set, otherwise it is not. By default, the `Partitioned` attribute is not set.
note This is an attribute that has not yet been fully standardized, and may change in the future. This also means many clients may ignore this attribute until they understand it.
More information about can be found in `the proposal`.

44.2.2.1.7 path Specifies the value for the `Path Set-Cookie attribute`. By default, the path is considered the `"default path"`.

44.2.2.1.8 priority Specifies the `string` to be the value for the `Priority Set-Cookie attribute`.

- "low'`will set the`Priority`attribute to`Low`. -`'medium'`will set the`Priority`attribute to`Medium`, the default priority when not set. -`'high'`will set the`Priority`attribute to`High`.

More information about the different priority levels can be found in `the specification`.
note This is an attribute that has not yet been fully standardized, and may change in the future. This also means many clients may ignore this attribute until they understand it.

44.2.2.1.9 sameSite Specifies the `boolean` or `string` to be the value for the `SameSite Set-Cookie attribute`.

- `true` will set the `SameSite` attribute to `Strict` for strict same site enforcement.
- `false` will not set the `SameSite` attribute.
- "lax'`will set the`SameSite`attribute to`Lax`for lax same site enforcement. -`'none'`will set the`SameSite`attribute to`None`for an explicit cross-site cookie. -`'strict'`will set the`SameSite`attribute to`Strict` for strict same site enforcement.

More information about the different enforcement levels can be found in `the specification`.
note This is an attribute that has not yet been fully standardized, and may change in the future. This also means many clients may ignore this attribute until they understand it.

44.2.2.1.10 secure Specifies the `boolean` value for the `Secure Set-Cookie attribute`. When truthy, the `Secure` attribute is set, otherwise it is not. By default, the `Secure` attribute is not set.
note be careful when setting this to `true`, as compliant clients will not send the cookie back to the server in the future if the browser does not have an HTTPS connection.

44.3 Example

The following example uses this module in conjunction with the Node.js core HTTP server to prompt a user for their name and display it back on future visits.

```
var cookie = require('cookie');
var escapeHtml = require('escape-html');
var http = require('http');
var url = require('url');
function onRequest(req, res) {
  // Parse the query string
  var query = url.parse(req.url, true, true).query;
  if (query && query.name) {
    // Set a new cookie with the name
    res.setHeader('Set-Cookie', cookie.serialize('name', String(query.name), {
      httpOnly: true,
      maxAge: 60 * 60 * 24 * 7 // 1 week
    }));
    // Redirect back after setting cookie
    res.statusCode = 302;
    res.setHeader('Location', req.headers.referer || '/');
    res.end();
    return;
  }
  // Parse the cookies on the request
  var cookies = cookie.parse(req.headers.cookie || ”);
  // Get the visitor name set in the cookie
  var name = cookies.name;
  res.setHeader('Content-Type', 'text/html; charset=UTF-8');
  if (name) {
    res.write('<p>Welcome back, <b>' + escapeHtml(name) + '</b>!</p>');
  } else {
    res.write('<p>Hello, new visitor!</p>');
  }
  res.write('<form method="GET">');
  res.write('<input placeholder="enter your name" name="name"> <input type="submit" value="Set Name">');
  res.end('</form>');
}
http.createServer(onRequest).listen(3000);
```

44.4 Testing

```
$ npm test
```

44.5 Benchmark

```
$ npm run bench
> cookie@0.5.0 bench
> node benchmark/index.js
  node@18.18.2
  acorn@8.10.0
  ada@2.6.0
  ares@1.19.1
  brotli@1.0.9
  cldr@43.1
  icu@73.2
  llhttp@6.0.11
  modules@108
  napi@9
  nghttp2@1.57.0
  nghttp3@0.7.0
  ngtcp2@0.8.1
  openssl@3.0.10+quic
  simdutf@3.2.14
  tz@2023c
  undici@5.26.3
  unicode@15.0
  uv@1.44.2
  uvwasi@0.0.18
  v8@10.2.154.26-node.26
  zlib@1.2.13.1-motley
> node benchmark/parse-top.js
  cookie.parse - top sites
  14 tests completed.
  parse accounts.google.com x 2,588,913 ops/sec ±0.74% (186 runs sampled)
  parse apple.com           x 2,370,002 ops/sec ±0.69% (186 runs sampled)
  parse cloudflare.com      x 2,213,102 ops/sec ±0.88% (188 runs sampled)
  parse docs.google.com     x 2,194,157 ops/sec ±1.03% (184 runs sampled)
  parse drive.google.com    x 2,265,084 ops/sec ±0.79% (187 runs sampled)
  parse en.wikipedia.org    x   457,099 ops/sec ±0.81% (186 runs sampled)
  parse linkedin.com        x   504,407 ops/sec ±0.89% (186 runs sampled)
  parse maps.google.com     x 1,230,959 ops/sec ±0.98% (186 runs sampled)
  parse microsoft.com       x   926,294 ops/sec ±0.88% (184 runs sampled)
  parse play.google.com     x 2,311,338 ops/sec ±0.83% (185 runs sampled)
  parse support.google.com  x 1,508,850 ops/sec ±0.86% (186 runs sampled)
  parse www.google.com      x 1,022,582 ops/sec ±1.32% (182 runs sampled)
  parse youtu.be            x   332,136 ops/sec ±1.02% (185 runs sampled)
  parse youtube.com         x   323,833 ops/sec ±0.77% (183 runs sampled)
> node benchmark/parse.js
  cookie.parse - generic
  6 tests completed.
  simple      x 3,214,032 ops/sec ±1.61% (183 runs sampled)
  decode      x   587,237 ops/sec ±1.16% (187 runs sampled)
  unquote     x 2,954,618 ops/sec ±1.35% (183 runs sampled)
  duplicates  x   857,008 ops/sec ±0.89% (187 runs sampled)
  10 cookies  x   292,133 ops/sec ±0.89% (187 runs sampled)
  100 cookies x    22,610 ops/sec ±0.68% (187 runs sampled)
```

44.6 References

- `RFC 6265: HTTP State Management Mechanism`
- `Same-site Cookies`

44.7 License

[MIT](LICENSE)

Chapter 45

Security Policies and Procedures

45.1 Reporting a Bug

The `cookie` team and community take all security bugs seriously. Thank you for improving the security of the project. We appreciate your efforts and responsible disclosure and will make every effort to acknowledge your contributions.
Report security bugs by emailing the current owner(s) of `cookie`. This information can be found in the npm registry using the command `npm owner ls cookie`. If unsure or unable to get the information from the above, open an issue in the `project issue tracker` asking for the current contact information.
To ensure the timely response to your report, please ensure that the entirety of the report is contained within the email body and not solely behind a web link or an attachment.
At least one owner will acknowledge your email within 48 hours, and will send a more detailed response within 48 hours indicating the next steps in handling your report. After the initial reply to your report, the owners will endeavor to keep you informed of the progress towards a fix and full announcement, and may ask for additional information or guidance.

Chapter 46

2.6.9 / 2017-09-22

- remove ReDoS regexp in o formatter (#504)

46.1 2.6.8 / 2017-05-18

- Fix: Check for undefined on browser globals (#462, @marbemac)

46.2 2.6.7 / 2017-05-16

- Fix: Update ms to 2.0.0 to fix regular expression denial of service vulnerability (#458, @hubdotcom)
- Fix: Inline extend function in node implementation (#452, @dougwilson)
- Docs: Fix typo (#455, @msasad)

46.3 2.6.5 / 2017-04-27

- Fix: null reference check on window.documentElement.style.WebkitAppearance (#447, @thebigredgeek)
- Misc: clean up browser reference checks (#447, @thebigredgeek)
- Misc: add npm-debug.log to .gitignore (@thebigredgeek)

46.4 2.6.4 / 2017-04-20

- Fix: bug that would occure if process.env.DEBUG is a non-string value. (#444, @LucianBuzzo)
- Chore: ignore bower.json in npm installations. (#437, @joaovieira)
- Misc: update "ms" to v0.7.3 (@tootallnate)

46.5 2.6.3 / 2017-03-13

- Fix: Electron reference to `process.env.DEBUG` (#431, @paulcbetts)
- Docs: Changelog fix (@thebigredgeek)

46.6 2.6.2 / 2017-03-10

- Fix: DEBUG_MAX_ARRAY_LENGTH (#420, @slavaGanzin)
- Docs: Add backers and sponsors from Open Collective (#422, @piamancini)
- Docs: Add Slackin invite badge (@tootallnate)

46.7 2.6.1 / 2017-02-10

- Fix: Module's `export default` syntax fix for IE8 `Expected identifier` error
- Fix: Whitelist DEBUG_FD for values 1 and 2 only (#415, @pi0)
- Fix: IE8 "Expected identifier" error (#414, @vgoma)
- Fix: Namespaces would not disable once enabled (#409, @musikov)

46.8 2.6.0 / 2016-12-28

- Fix: added better null pointer checks for browser useColors (@thebigredgeek)
- Improvement: removed explicit `window.debug` export (#404, @tootallnate)
- Improvement: deprecated `DEBUG_FD` environment variable (#405, @tootallnate)

46.9 2.5.2 / 2016-12-25

- Fix: reference error on window within webworkers (#393, @KlausTrainer)
- Docs: fixed README typo (#391, @lurch)
- Docs: added notice about v3 api discussion (@thebigredgeek)

46.10 2.5.1 / 2016-12-20

- Fix: babel-core compatibility

46.11 2.5.0 / 2016-12-20

- Fix: wrong reference in bower file (@thebigredgeek)
- Fix: webworker compatibility (@thebigredgeek)
- Fix: output formatting issue (#388, @kribblo)
- Fix: babel-loader compatibility (#383, @escwald)
- Misc: removed built asset from repo and publications (@thebigredgeek)
- Misc: moved source files to /src (#378, @yamikuronue)
- Test: added karma integration and replaced babel with browserify for browser tests (#378, @yamikuronue)
- Test: coveralls integration (#378, @yamikuronue)
- Docs: simplified language in the opening paragraph (#373, @yamikuronue)

46.12 2.4.5 / 2016-12-17

- Fix: `navigator` undefined in Rhino (#376, @jochenberger)
- Fix: custom log function (#379, @hsiliev)
- Improvement: bit of cleanup + linting fixes (@thebigredgeek)
- Improvement: rm non-maintainted `dist/` dir (#375, @freewil)
- Docs: simplified language in the opening paragraph. (#373, @yamikuronue)

46.13 2.4.4 / 2016-12-14

- Fix: work around debug being loaded in preload scripts for electron (#368, @paulcbetts)

46.14 2.4.3 / 2016-12-14

- Fix: navigation.userAgent error for react native (#364, @escwald)

46.15 2.4.2 / 2016-12-14

- Fix: browser colors (#367, @tootallnate)
- Misc: travis ci integration (@thebigredgeek)
- Misc: added linting and testing boilerplate with sanity check (@thebigredgeek)

46.16 2.4.1 / 2016-12-13

- Fix: typo that broke the package (#356)

46.17 2.4.0 / 2016-12-13

- Fix: bower.json references unbuilt src entry point (#342, @justmatt)
- Fix: revert "handle regex special characters" (@tootallnate)
- Feature: configurable util.inspect()`options for NodeJS (#327, @tootallnate)`
- `Feature: ` `O`(big O) pretty-prints objects (#322, @tootallnate)
- Improvement: allow colors in workers (#335, @botverse)
- Improvement: use same color for same namespace. (#338, @lchenay)

46.18 2.3.3 / 2016-11-09

- Fix: Catch `JSON.stringify()` errors (#195, Jovan Alleyne)
- Fix: Returning `localStorage` saved values (#331, Levi Thomason)
- Improvement: Don't create an empty object when no `process` (Nathan Rajlich)

46.19 2.3.2 / 2016-11-09

- Fix: be super-safe in index.js as well (@TooTallNate)
- Fix: should check whether process exists (Tom Newby)

46.20 2.3.1 / 2016-11-09

- Fix: Added electron compatibility (#324, @paulcbetts)
- Improvement: Added performance optimizations (@tootallnate)
- Readme: Corrected PowerShell environment variable example (#252, @gimre)
- Misc: Removed yarn lock file from source control (#321, @fengmk2)

46.21 2.3.0 / 2016-11-07

- Fix: Consistent placement of ms diff at end of output (#215, @gorangajic)
- Fix: Escaping of regex special characters in namespace strings (#250, @zacronos)
- Fix: Fixed bug causing crash on react-native (#282, @vkarpov15)
- Feature: Enabled ES6+ compatible import via default export (#212 @bucaran)
- Feature: Added O formatter to reflect Chrome's console.log capability (#279, @oncletom)
- Package: Update "ms" to 0.7.2 (#315, @DevSide)
- Package: removed superfluous version property from bower.json (#207 @kkirsche)
- Readme: fix USE_COLORS to DEBUG_COLORS
- Readme: Doc fixes for format string sugar (#269, @mlucool)
- Readme: Updated docs for DEBUG_FD and DEBUG_COLORS environment variables (#232, @mattlyons0)
- Readme: doc fixes for PowerShell (#271 #243, @exoticknight @unreadable)
- Readme: better docs for browser support (#224, @matthewmueller)
- Tooling: Added yarn integration for development (#317, @thebigredgeek)
- Misc: Renamed History.md to CHANGELOG.md (@thebigredgeek)
- Misc: Added license file (#226 #274, @CantemoInternal @sdaitzman)
- Misc: Updated contributors (@thebigredgeek)

46.22 2.2.0 / 2015-05-09

- package: update "ms" to v0.7.1 (#202, @dougwilson)
- README: add logging to file example (#193, @DanielOchoa)
- README: fixed a typo (#191, @amir-s)
- browser: expose `storage` (#190, @stephenmathieson)
- Makefile: add a `distclean` target (#189, @stephenmathieson)

46.23 2.1.3 / 2015-03-13

- Updated stdout/stderr example (#186)
- Updated example/stdout.js to match debug current behaviour
- Renamed example/stderr.js to stdout.js
- Update Readme.md (#184)
- replace high intensity foreground color for bold (#182, #183)

46.24 2.1.2 / 2015-03-01

- dist: recompile
- update "ms" to v0.7.0
- package: update "browserify" to v9.0.3
- component: fix "ms.js" repo location
- changed bower package name
- updated documentation about using debug in a browser
- fix: security error on safari (#167, #168, @yields)

46.25 2.1.1 / 2014-12-29

- browser: use `typeof` to check for `console` existence
- browser: check for `console.log` truthiness (fix IE 8/9)
- browser: add support for Chrome apps
- Readme: added Windows usage remarks
- Add `bower.json` to properly support bower install

46.26 2.1.0 / 2014-10-15

- node: implement `DEBUG_FD` env variable support
- package: update "browserify" to v6.1.0
- package: add "license" field to package.json (#135, @panuhorsmalahti)

46.27 2.0.0 / 2014-09-01

- package: update "browserify" to v5.11.0
- node: use stderr rather than stdout for logging (#29, @stephenmathieson)

46.28 1.0.4 / 2014-07-15

- dist: recompile
- example: remove `console.info()` log usage
- example: add "Content-Type" UTF-8 header to browser example
- browser: place c marker after the space character
- browser: reset the "content" color via `color:  inherit`
- browser: add colors support for Firefox >= v31
- debug: prefer an instance `log()` function over the global one (#119)
- Readme: update documentation about styled console logs for FF v31 (#116, @wryk)

46.29 1.0.3 / 2014-07-09

- Add support for multiple wildcards in namespaces (#122, @seegno)
- browser: fix lint

46.30 1.0.2 / 2014-06-10

- browser: update color palette (#113, @gscottolson)
- common: make console logging function configurable (#108, @timoxley)
- node: fix o colors on old node <= 0.8.x
- Makefile: find node path using shell/which (#109, @timoxley)

46.31 1.0.1 / 2014-06-06

- browser: use `removeItem()` to clear localStorage
- browser, node: don't set DEBUG if namespaces is undefined (#107, @leedm777)
- package: add "contributors" section
- node: fix comment typo
- README: list authors

46.32 1.0.0 / 2014-06-04

- make ms diff be global, not be scope
- debug: ignore empty strings in enable()
- node: make DEBUG_COLORS able to disable coloring
- $*$: export the `colors` array
- npmignore: don't publish the `dist` dir
- Makefile: refactor to use browserify
- package: add "browserify" as a dev dependency
- Readme: add Web Inspector Colors section
- node: reset terminal color for the debug content
- node: map "%o" to `util.inspect()`
- browser: map "%j" to `JSON.stringify()`
- debug: add custom "formatters"
- debug: use "ms" module for humanizing the diff
- Readme: add "bash" syntax highlighting
- browser: add Firebug color support
- browser: add colors for WebKit browsers
- node: apply log to `console`
- rewrite: abstract common logic for Node & browsers
- add .jshintrc file

46.33 0.8.1 / 2014-04-14

- package: re-add the "component" section

46.34 0.8.0 / 2014-03-30

- add `enable()` method for nodejs. Closes #27
- change from stderr to stdout
- remove unnecessary index.js file

46.35 0.7.4 / 2013-11-13

- remove "browserify" key from package.json (fixes something in browserify)

46.36 0.7.3 / 2013-10-30

- fix: catch localStorage security error when cookies are blocked (Chrome)
- add debug(err) support. Closes #46
- add .browser prop to package.json. Closes #42

46.37 0.7.2 / 2013-02-06

- fix package.json
- fix: Mobile Safari (private mode) is broken with debug
- fix: Use unicode to send escape character to shell instead of octal to work with strict mode javascript

46.38 0.7.1 / 2013-02-05

- add repository URL to package.json
- add DEBUG_COLORED to force colored output
- add browserify support
- fix component. Closes #24

46.39 0.7.0 / 2012-05-04

- Added .component to package.json
- Added debug.component.js build

46.40 0.6.0 / 2012-03-16

- Added support for "-" prefix in DEBUG [Vinay Pulim]
- Added `.enabled` flag to the node version [TooTallNate]

46.41 0.5.0 / 2012-02-02

- Added: humanize diffs. Closes #8
- Added `debug.disable()` to the CS variant
- Removed padding. Closes #10
- Fixed: persist client-side variant again. Closes #9

46.42 0.4.0 / 2012-02-01

- Added browser variant support for older browsers [TooTallNate]
- Added ‘debug.enable('project:∗')` to browser variant [TooTallNate]
- Added padding to diff (moved it to the right)

46.43 0.3.0 / 2012-01-26

- Added millisecond diff when isatty, otherwise UTC string

46.44 0.2.0 / 2012-01-22

- Added wildcard support

46.45 0.1.0 / 2011-12-02

- Added: remove colors unless stderr isatty [TooTallNate]

46.46 0.0.1 / 2010-01-03

- Initial release

Chapter 47

debug

![OpenCollective](https://opencollective.com/debug/backers/badge.svg) ![OpenCollective](https://opencollective.com/debug/spo
A tiny node.js debugging utility modelled after node core's debugging technique.
Discussion around the V3 API is under way here

47.1 Installation

```
$ npm install debug
```

47.2 Usage

`debug` exposes a function; simply pass this function the name of your module, and it will return a decorated version of `console.error` for you to pass debug statements to. This will allow you to toggle the debug output for different parts of your module as well as the module as a whole.
Example *app.js*:

```
var debug = require('debug')('http')
  , http = require('http')
  , name = 'My App';
// fake app
debug('booting %s', name);
http.createServer(function(req, res){
  debug(req.method + ' ' + req.url);
  res.end('hello\n');
}).listen(3000, function(){
  debug('listening');
});
// fake worker of some kind
require('./worker');
```

Example *worker.js*:

```
var debug = require('debug')('worker');
setInterval(function(){
  debug('doing some work');
}, 1000);
```

The **DEBUG** environment variable is then used to enable these based on space or comma-delimited names. Here are some examples:

47.2.0.1 Windows note

On Windows the environment variable is set using the `set` command.

```
set DEBUG=*,-not_this
```

Note that PowerShell uses different syntax to set environment variables.

```
$env:DEBUG = "*,-not_this"
```

Then, run the program to be debugged as usual.

47.3 Millisecond diff

When actively developing an application it can be useful to see when the time spent between one `debug()` call and the next. Suppose for example you invoke `debug()` before requesting a resource, and after as well, the "+NNNms" will show you how much time was spent between calls.

When stdout is not a TTY, `Date#toUTCString()` is used, making it more useful for logging the debug information as shown below:

47.4 Conventions

If you're using this in one or more of your libraries, you *should* use the name of your library so that developers may toggle debugging as desired without guessing names. If you have more than one debuggers you *should* prefix them with your library name and use ":" to separate features. For example "bodyParser" from Connect would then be "connect:bodyParser".

47.5 Wildcards

The * character may be used as a wildcard. Suppose for example your library has debuggers named "connect↩ :bodyParser", "connect:compress", "connect:session", instead of listing all three with `DEBUG=connect:body↩ Parser,connect:compress,connect:session`, you may simply do `DEBUG=connect:*`, or to run everything using this module simply use `DEBUG=*`.
You can also exclude specific debuggers by prefixing them with a "-" character. For example, `DEBUG=*,-connect↩ :*` would include all debuggers except those starting with "connect:".

47.6 Environment Variables

When running through Node.js, you can set a few environment variables that will change the behavior of the debug logging:

Name	Purpose
`DEBUG`	Enables/disables specific debugging namespaces.
`DEBUG_COLORS`	Whether or not to use colors in the debug output.
`DEBUG_DEPTH`	Object inspection depth.
`DEBUG_SHOW_HIDDEN`	Shows hidden properties on inspected objects.

Note: The environment variables beginning with `DEBUG_` end up being converted into an Options object that gets used with `o`/`O` formatters. See the Node.js documentation for `util.inspect()` for the complete list.

47.7 Formatters

Debug uses `printf-style` formatting. Below are the officially supported formatters:

Formatter	Representation
`O`	Pretty-print an Object on multiple lines.
`o`	Pretty-print an Object all on a single line.
`s`	String.
`d`	Number (both integer and float).
`j`	JSON. Replaced with the string '[Circular]' if the argument contains circular references.
`%%`	Single percent sign (''). This does not consume an argument.

47.7.1 Custom formatters

You can add custom formatters by extending the `debug.formatters` object. For example, if you wanted to add support for rendering a Buffer as hex with `h`, you could do something like:

```
const createDebug = require('debug')
createDebug.formatters.h = (v) => {
  return v.toString('hex')
}
// ...elsewhere
```

```
const debug = createDebug('foo')
debug('this is hex:  %h', new Buffer('hello world'))
//   foo this is hex:  68656c6c6f20776f726c6421 +0ms
```

47.8 Browser support

You can build a browser-ready script using `browserify`, or just use the `browserify-as-a-service build`, if you don't want to build it yourself.
Debug's enable state is currently persisted by `localStorage`. Consider the situation shown below where you have `worker:a` and `worker:b`, and wish to debug both. You can enable this using `localStorage.debug`:

```
localStorage.debug = 'worker:*'
```

And then refresh the page.

```
a = debug('worker:a');
b = debug('worker:b');
setInterval(function(){
  a('doing some work');
}, 1000);
setInterval(function(){
  b('doing some work');
}, 1200);
```

47.8.0.1 Web Inspector Colors

Colors are also enabled on "Web Inspectors" that understand the `c` formatting option. These are WebKit web inspectors, Firefox (`since version 31`) and the Firebug plugin for Firefox (any version).
Colored output looks something like:

47.9 Output streams

By default `debug` will log to stderr, however this can be configured per-namespace by overriding the `log` method:
Example *stdout.js*:

```
var debug = require('debug');
var error = debug('app:error');
// by default stderr is used
error('goes to stderr!');
var log = debug('app:log');
// set this namespace to log via console.log
log.log = console.log.bind(console); // don't forget to bind to console!
log('goes to stdout');
error('still goes to stderr!');
// set all output to go via console.info
// overrides all per-namespace log settings
debug.log = console.info.bind(console);
error('now goes to stdout via console.info');
log('still goes to stdout, but via console.info now');
```

47.10 Authors

- TJ Holowaychuk
- Nathan Rajlich
- Andrew Rhyne

47.11 Backers

Support us with a monthly donation and help us continue our activities. [`Become a backer`]

47.12 Sponsors

Become a sponsor and get your logo on our README on Github with a link to your site. [`Become a sponsor`]

47.13 License

Chapter 48

Changelog

All notable changes to this project will be documented in this file.
The format is based on Keep a Changelog and this project adheres to Semantic Versioning.

48.1 <a href="https://github.com/ljharb/define-data-property/compare/v1.1.3...v1.1.4" >v1.1.4</a> - 2024-02-13

48.1.1 Commits

- [Refactor] use `es-define-property` 90f2f4c
- [Dev Deps] update `@types/object.getownpropertydescriptors` cd929d9

48.2 <a href="https://github.com/ljharb/define-data-property/compare/v1.1.2...v1.1.3" >v1.1.3</a> - 2024-02-12

48.2.1 Commits

- [types] hand-write d.ts instead of emitting it 0cbc988
- [meta] simplify `exports` 690781e
- [Dev Deps] update `hasown`; clean up DT packages 6cdfd1c
- [actions] cleanup 3142bc6
- [meta] add `funding` 8474423
- [Deps] update `get-intrinsic` 3e9be00

48.3 <a href="https://github.com/ljharb/define-data-property/compare/v1.1.1...v1.1.2" >v1.1.2</a> - 2024-02-05

48.3.1 Commits

- [Dev Deps] update @types packages, `object-inspect`, `tape`, `typescript` df41bf8
- [Dev Deps] update DT packages, `aud`, `npmignore`, `tape`, typescript [fab0e4e] (https↩ ://github.com/ljharb/define-data-property/commit/fab0e4ec709ee02b79f42d6db3ee5f26e0a
- `[Dev Deps] use`hasown`instead of`has [aa51ef9] (https://github.com/ljharb/define-data-prope
- `[Refactor] use`es-errors`, so things that only need those do not need`get-intrinsic [d89be50] (https://github.com/ljharb/define-data-property/commit/d89be50571175888d3912386051226

- `[Deps] update`has-property-descriptors`[`7af887c`] (` `https://github.com/ljharb/define-data-propert`
- `[Deps] update`get-intrinsic`[`bb8728e`](` `https://github.com/ljharb/define-data-property/commit/`

48.4 <a href="https://github.com/ljharb/define-data-property/compare/v1.1.0...v1.1.1" >v1.1.1</a> - 2023-10-12

48.4.1 Commits

- [Tests] fix tests in ES3 engines 5c6920e
- [Dev Deps] update `@types/es-value-fixtures`, `@types/for-each`, `@types/gopd`, `@types/has-property-descriptors`, `tape`, `typescript` 7d82dfc
- [Fix] IE 8 has a broken `Object.defineProperty` 0672e1a
- [meta] emit types on prepack 73acb1f
- [Dev Deps] update `tape`, `typescript` 9489a77

48.5 <a href="https://github.com/ljharb/define-data-property/compare/v1.0.1...v1.1.0" >v1.1.0</a> - 2023-09-13

48.5.1 Commits

- [New] add `loose` arg 155235a
- [New] allow `null` to be passed for the non* args 7d2fa5f

48.6 <a href="https://github.com/ljharb/define-data-property/compare/v1.0.0...v1.0.1" >v1.0.1</a> - 2023-09-12

48.6.1 Commits

- [meta] add TS types 43d763c
- [Dev Deps] update `@types/tape`, `typescript` f444985
- [meta] add `safe-publish-latest`, 172bb10

48.7 v1.0.0 - 2023-09-12

48.7.1 Commits

- Initial implementation, tests, readme 5b43d6b
- Initial commit 35e577a
- npm init 82a0a04
- Only apps should have lockfiles 96df244
- [meta] use `npmignore` to autogenerate an npmignore file a87ff18

Chapter 49

define-data-property <sup><a href="https://npmjs.org/package/define-data-property" ><img src="https://versionbadg.es/ljharb/define-data-property.svg" alt="Version Badge"/></a></sup>

[][license-url]

Define a data property on an object. Will fall back to assignment in an engine without descriptors.
The three `non*` argument can also be passed `null`, which will use the existing state if available.
The `loose` argument will mean that if you attempt to set a non-normal data property, in an environment without descriptor support, it will fall back to normal assignment.

49.1 Usage

```
var defineDataProperty = require('define-data-property');
var assert = require('assert');
var obj = {};
defineDataProperty(obj, 'key', 'value');
defineDataProperty(
    obj,
    'key2',
    'value',
    true, // nonEnumerable, optional
    false, // nonWritable, optional
    true, // nonConfigurable, optional
    false // loose, optional
);
assert.deepEqual(
    Object.getOwnPropertyDescriptors(obj),
    {
        key:  {
            configurable:  true,
            enumerable:  true,
            value:  'value',
            writable:  true,
        },
        key2:  {
            configurable:  false,
            enumerable:  false,
            value:  'value',
            writable:  true,
        },
    }
);
```

Chapter 50

2.1.0

- fix: issue where `clear()` is still keeping references to the elements (#47)
- refactor: performance optimizations for growth and array copy (#43)
- refactor: performance optimizations for toArray and fromArray (#46)
- test: add additional benchmarks for queue growth and `toArray` (#45)

50.1 2.0.1

- fix(types): incorrect return type on `size()`

50.2 2.0.0

- fix!: `push` & `unshift` now accept `undefined` values to match behaviour of `Array` (fixes #25) (#35)
  - This is only a **BREAKING** change if you are currently expecting `push(undefined)` and `unshift(undefined)` to do nothing - the new behaviour now correctly adds undefined values to the queue.
  - **Note**: behaviour of `push()` & `unshift()` (no arguments) remains unchanged (nothing gets added to the queue).
  - **Note**: If you need to differentiate between `undefined` values in the queue and the return value of `pop()` then check the queue `.length` before popping.
- fix: incorrect methods in types definition file

50.3 1.5.1

- perf: minor performance tweak when growing queue size (#29)

50.4 1.5.0

- feat: adds capacity option for circular buffers (#27)

Chapter 51

README

Denque

Denque is a well tested, extremely fast and lightweight `double-ended queue` implementation with zero dependencies and includes TypeScript types.
Double-ended queues can also be used as a:

- `Stack`
- `Queue`

This implementation is currently the fastest available, even faster than `double-ended-queue`, see the `benchmarks`.
Every queue operation is done at a constant `O(1)` - including random access from `.peekAt(index)`.
Works on all node versions >= v0.10

51.0.1 Quick Start

Install the package:
```
npm install denque
```
Create and consume a queue:
```
const Denque = require("denque");
const denque = new Denque([1,2,3,4]);
denque.shift(); // 1
denque.pop(); // 4
```
See the `API reference documentation` for more examples.

51.0.2 Who's using it?

- `Kafka Node.js client`
- `MariaDB Node.js client`
- `MongoDB Node.js client`
- `MySQL Node.js client`
- `Redis Node.js clients`

... and `many more`.

51.0.3 License

- See `LICENSE`

Built and maintained by `Invertase`.

Chapter 52

2.0.0 / 2018-10-26

- Drop support for Node.js 0.6
- Replace internal `eval` usage with `Function` constructor
- Use instance methods on `process` to check for listeners

52.1 1.1.2 / 2018-01-11

- perf: remove argument reassignment
- Support Node.js 0.6 to 9.x

52.2 1.1.1 / 2017-07-27

- Remove unnecessary `Buffer` loading
- Support Node.js 0.6 to 8.x

52.3 1.1.0 / 2015-09-14

- Enable strict mode in more places
- Support io.js 3.x
- Support io.js 2.x
- Support web browser loading
  - Requires bundler like Browserify or webpack

52.4 1.0.1 / 2015-04-07

- Fix `TypeError`s when under "use strict`` code
- Fix useless type name on auto-generated messages
- Support io.js 1.x
- Support Node.js 0.12

52.5 1.0.0 / 2014-09-17

- No changes

52.6 0.4.5 / 2014-09-09

- Improve call speed to functions using the function wrapper
- Support Node.js 0.6

52.7 0.4.4 / 2014-07-27

- Work-around v8 generating empty stack traces

52.8 0.4.3 / 2014-07-26

- Fix exception when global `Error.stackTraceLimit` is too low

52.9 0.4.2 / 2014-07-19

- Correct call site for wrapped functions and properties

52.10 0.4.1 / 2014-07-19

- Improve automatic message generation for function properties

52.11 0.4.0 / 2014-07-19

- Add `TRACE_DEPRECATION` environment variable
- Remove non-standard grey color from color output
- Support `--no-deprecation` argument
- Support `--trace-deprecation` argument
- Support `deprecate.property(fn, prop, message)`

52.12 0.3.0 / 2014-06-16

- Add `NO_DEPRECATION` environment variable

52.13 0.2.0 / 2014-06-15

- Add `deprecate.property(obj, prop, message)`
- Remove `supports-color` dependency for node.js 0.8

52.14 0.1.0 / 2014-06-15

- Add `deprecate.function(fn, message)`
- Add ‘process.on('deprecation’, fn)`emitter`
- `Automatically generate message when omitted from`deprecate()`

52.15 0.0.1 / 2014-06-15

- Fix warning for dynamic calls at singe call site

52.16 0.0.0 / 2014-06-15

- Initial implementation

Chapter 53

depd

Deprecate all the things

> With great modules comes great responsibility; mark things deprecated!

53.1 Install

This module is installed directly using `npm`:

```
$ npm install depd
```

This module can also be bundled with systems like `Browserify` or `webpack`, though by default this module will alter it's API to no longer display or track deprecations.

53.2 API

```
var deprecate = require('depd')('my-module')
```

This library allows you to display deprecation messages to your users. This library goes above and beyond with deprecation warnings by introspection of the call stack (but only the bits that it is interested in).

Instead of just warning on the first invocation of a deprecated function and never again, this module will warn on the first invocation of a deprecated function per unique call site, making it ideal to alert users of all deprecated uses across the code base, rather than just whatever happens to execute first.

The deprecation warnings from this module also include the file and line information for the call into the module that the deprecated function was in.

NOTE this library has a similar interface to the `debug` module, and this module uses the calling file to get the boundary for the call stacks, so you should always create a new `deprecate` object in each file and not within some central file.

53.2.1 depd(namespace)

Create a new deprecate function that uses the given namespace name in the messages and will display the call site prior to the stack entering the file this function was called from. It is highly suggested you use the name of your module as the namespace.

53.2.2 deprecate(message)

Call this function from deprecated code to display a deprecation message. This message will appear once per unique caller site. Caller site is the first call site in the stack in a different file from the caller of this function.

If the message is omitted, a message is generated for you based on the site of the `deprecate()` call and will display the name of the function called, similar to the name displayed in a stack trace.

53.2.3 deprecate.function(fn, message)

Call this function to wrap a given function in a deprecation message on any call to the function. An optional message can be supplied to provide a custom message.

53.2.4 deprecate.property(obj, prop, message)

Call this function to wrap a given property on object in a deprecation message on any accessing or setting of the property. An optional message can be supplied to provide a custom message.
The method must be called on the object where the property belongs (not inherited from the prototype).
If the property is a data descriptor, it will be converted to an accessor descriptor in order to display the deprecation message.

53.2.5 process.on('deprecation', fn)

This module will allow easy capturing of deprecation errors by emitting the errors as the type "deprecation" on the global `process`. If there are no listeners for this type, the errors are written to STDERR as normal, but if there are any listeners, nothing will be written to STDERR and instead only emitted. From there, you can write the errors in a different format or to a logging source.
The error represents the deprecation and is emitted only once with the same rules as writing to STDERR. The error has the following properties:

- `message` - This is the message given by the library
- `name` - This is always ‘‘DeprecationError’−namespace`− This is the namespace the deprecation came from −`stack` - This is the stack of the call to the deprecated thing

Example `error.stack` output:
```
DeprecationError:  my-cool-module deprecated oldfunction
    at Object.<anonymous> ([eval]-wrapper:6:22)
    at Module._compile (module.js:456:26)
    at evalScript (node.js:532:25)
    at startup (node.js:80:7)
    at node.js:902:3
```

53.2.6 process.env.NO_DEPRECATION

As a user of modules that are deprecated, the environment variable `NO_DEPRECATION` is provided as a quick solution to silencing deprecation warnings from being output. The format of this is similar to that of `DEBUG`:
```
$ NO_DEPRECATION=my-module,othermod node app.js
```
This will suppress deprecations from being output for "my-module" and "othermod". The value is a list of comma-separated namespaces. To suppress every warning across all namespaces, use the value $*$ for a namespace.
Providing the argument `--no-deprecation` to the `node` executable will suppress all deprecations (only available in Node.js 0.8 or higher).
NOTE This will not suppress the deperecations given to any "deprecation" event listeners, just the output to STDERR.

53.2.7 process.env.TRACE_DEPRECATION

As a user of modules that are deprecated, the environment variable `TRACE_DEPRECATION` is provided as a solution to getting more detailed location information in deprecation warnings by including the entire stack trace. The format of this is the same as `NO_DEPRECATION`:
```
$ TRACE_DEPRECATION=my-module,othermod node app.js
```
This will include stack traces for deprecations being output for "my-module" and "othermod". The value is a list of comma-separated namespaces. To trace every warning across all namespaces, use the value $*$ for a namespace.
Providing the argument `--trace-deprecation` to the `node` executable will trace all deprecations (only available in Node.js 0.8 or higher).
NOTE This will not trace the deperecations silenced by `NO_DEPRECATION`.

53.3 Display

When a user calls a function in your library that you mark deprecated, they will see the following written to STDERR (in the given colors, similar colors and layout to the `debug` module):
```
bright cyan    bright yellow
|              |          reset       cyan
|              |          |           |

my-cool-module deprecated oldfunction [eval]-wrapper:6:22
```

```
|               |          |           |
namespace       |          |           location of mycoolmod.oldfunction() call
                |          deprecation message
                the word "deprecated"
```

If the user redirects their STDERR to a file or somewhere that does not support colors, they see (similar layout to the `debug` module):

```
Sun, 15 Jun 2014 05:21:37 GMT my-cool-module deprecated oldfunction at [eval]-wrapper:6:22

|                             |              |          |              |
timestamp of message          namespace      |          |             location of mycoolmod.oldfunction()
      call
                                             |          deprecation message
                                             the word "deprecated"
```

53.4 Examples

53.4.1 Deprecating all calls to a function

This will display a deprecated message about "oldfunction" being deprecated from "my-module" on STDERR.

```
var deprecate = require('depd')('my-cool-module')
// message automatically derived from function name
// Object.oldfunction
exports.oldfunction = deprecate.function(function oldfunction () {
  // all calls to function are deprecated
})
// specific message
exports.oldfunction = deprecate.function(function () {
  // all calls to function are deprecated
}, 'oldfunction')
```

53.4.2 Conditionally deprecating a function call

This will display a deprecated message about "weirdfunction" being deprecated from "my-module" on STDERR when called with less than 2 arguments.

```
var deprecate = require('depd')('my-cool-module')
exports.weirdfunction = function () {
  if (arguments.length < 2) {
    // calls with 0 or 1 args are deprecated
    deprecate('weirdfunction args < 2')
  }
}
```

When calling `deprecate` as a function, the warning is counted per call site within your own module, so you can display different deprecations depending on different situations and the users will still get all the warnings:

```
var deprecate = require('depd')('my-cool-module')
exports.weirdfunction = function () {
  if (arguments.length < 2) {
    // calls with 0 or 1 args are deprecated
    deprecate('weirdfunction args < 2')
  } else if (typeof arguments[0] !== 'string') {
    // calls with non-string first argument are deprecated
    deprecate('weirdfunction non-string first arg')
  }
}
```

53.4.3 Deprecating property access

This will display a deprecated message about "oldprop" being deprecated from "my-module" on STDERR when accessed. A deprecation will be displayed when setting the value and when getting the value.

```
var deprecate = require('depd')('my-cool-module')
exports.oldprop = 'something'
// message automatically derives from property name
deprecate.property(exports, 'oldprop')
// explicit message
deprecate.property(exports, 'oldprop', 'oldprop >= 0.10')
```

53.5 License

[MIT](LICENSE)

Chapter 54

destroy

![License][license-image]
Destroy a stream.
This module is meant to ensure a stream gets destroyed, handling different APIs and Node.js bugs.

54.1 API

```
var destroy = require('destroy')
```

54.1.1 destroy(stream [, suppress])

Destroy the given stream, and optionally suppress any future `error` events.
In most cases, this is identical to a simple `stream.destroy()` call. The rules are as follows for a given stream:

1. If the `stream` is an instance of `ReadStream`, then call `stream.destroy()` and add a listener to the `open` event to call `stream.close()` if it is fired. This is for a Node.js bug that will leak a file descriptor if `.destroy()` is called before `open`.
2. If the `stream` is an instance of a zlib stream, then call `stream.destroy()` and close the underlying zlib handle if open, otherwise call `stream.close()`. This is for consistency across Node.js versions and a Node.js bug that will leak a native zlib handle.
3. If the `stream` is not an instance of `Stream`, then nothing happens.
4. If the `stream` has a `.destroy()` method, then call it.

The function returns the `stream` passed in as the argument.

54.2 Example

```
var destroy = require('destroy')
var fs = require('fs')
var stream = fs.createReadStream('package.json')
// ...  and later
destroy(stream)
```

Chapter 55

CHANGELOG

v3.0.0 - November 9, 2018

- 0b5a8c7 Breaking: drop support for Node $<$ 6 (#223) (Kai Cataldo)
- a05e9f2 Upgrade: eslint-release@1.0.0 (#220) (Teddy Katz)
- 36ed027 Chore: upgrade coveralls to ^3.0.1 (#213) (Teddy Katz)
- 8667e34 Upgrade: eslint-release@^0.11.1 (#210) (Kevin Partington)

v2.1.0 - January 6, 2018

- 827f314 Update: support node ranges (fixes #89) (#190) (Teddy Katz)

v2.0.2 - November 25, 2017

- 5049ee3 Fix: Remove redundant LICENSE/README names from files (#203) (Kevin Partington)

v2.0.1 - November 10, 2017

- 009f33d Fix: Making sure union type stringification respects compact flag (#199) (Mitermayer Reis)
- 19da935 Use native String.prototype.trim instead of a custom implementation. (#201) (Rouven Weßling)
- e3a011b chore: add mocha.opts to restore custom mocha config (Jason Kurian)
- d888200 chore: adds nyc and a newer version of mocha to accurately report coverage (Jason Kurian)
- 6b210a8 fix: support type expression for @this tag (fixes #181) (#182) (Frédéric Junod)
- 1c4a4c7 fix: Allow array indexes in names (#193) (Tom MacWright)
- 9aed54d Fix incorrect behavior when arrow functions are used as default values (#189) (Gaurab Paul)
- 9efb6ca Upgrade: Use Array.isArray instead of isarray package (#195) (medanat)

v2.0.0 - November 15, 2016

- 7d7c5f1 Breaking: Re-license to Apache 2 (fixes #176) (#178) (Nicholas C. Zakas)
- 5496132 Docs: Update license copyright (Nicholas C. Zakas)

v1.5.0 - October 13, 2016

- e33c6bb Update: Add support for BooleanLiteralType (#173) (Erik Arvidsson)

v1.4.0 - September 13, 2016

- d7426e5 Update: add ability to parse optional properties in typedefs (refs #5) (#174) (ikokostya)

v1.3.0 - August 22, 2016

- 12c7ad9 Update: Add support for numeric and string literal types (fixes #156) (#172) (Andrew Walter)

v1.2.3 - August 16, 2016

- b96a884 Build: Add CI release script (Nicholas C. Zakas)
- 8d9b3c7 Upgrade: Upgrade esutils to v2.0.2 (fixes #170) (#171) (Emeegeemee)

v1.2.2 - May 19, 2016

- ebe0b08 Fix: Support case insensitive tags (fixes #163) (#164) (alberto)
- 8e6d81e Chore: Remove copyright and license from headers (Nicholas C. Zakas)
- 79035c6 Chore: Include jQuery Foundation copyright (Nicholas C. Zakas)
- 06910a7 Fix: Preserve whitespace in default param string values (fixes #157) (Kai Cataldo)

v1.2.1 - March 29, 2016

- 1f54014 Fix: allow hyphens in names (fixes #116) (Kai Cataldo)
- bbee469 Docs: Add issue template (Nicholas C. Zakas)

v1.2.0 - February 19, 2016

- 18136c5 Build: Cleanup build system (Nicholas C. Zakas)
- b082f85 Update: Add support for slash in namepaths (fixes #100) (Ryan Duffy)
- def53a2 Docs: Fix typo in option lineNumbers (Daniel Tschinder)
- e2cbbc5 Update: Bump isarray to v1.0.0 (Shinnosuke Watanabe)
- ae07aa8 Fix: Allow whitespace in optional param with default value (fixes #141) (chris)

v1.1.0 - January 6, 2016

- Build: Switch to Makefile.js (Nicholas C. Zakas)
- New: support name expression for @this tag (fixes #143) (Tim Schaub)
- Build: Update ESLint settings (Nicholas C. Zakas)

v1.0.0 - December 21, 2015

- New: parse caption tags in examples into separate property. (fixes #131) (Tom MacWright)

v0.7.2 - November 27, 2015

- Fix: Line numbers for some tags (fixes #138) Fixing issue where input was not consumed via advance() but was skipped when parsing tags resulting in sometimes incorrect reported lineNumber. (TEHEK)
- Build: Add missing linefix package (Nicholas C. Zakas)

v0.7.1 - November 13, 2015

- Update: Begin switch to Makefile.js (Nicholas C. Zakas)
- Fix: permit return tag without type (fixes #136) (Tom MacWright)
- Fix: package.json homepage field (Bogdan Chadkin)
- Fix: Parse array default syntax. Fixes #133 (Tom MacWright)
- Fix: Last tag always has
  in the description (fixes #87) (Burak Yigit Kaya)

- Docs: Add changelog (Nicholas C. Zakas)

v0.7.0 - September 21, 2015

- Docs: Update README with new info (fixes #127) (Nicholas C. Zakas)
- Fix: Parsing fix for param with arrays and properties (fixes #111) (Gyandeep Singh)
- Build: Add travis build (fixes #123) (Gyandeep Singh)
- Fix: Parsing of parameter name without a type (fixes #120) (Gyandeep Singh)
- New: added preserveWhitespace option (Aleks Totic)
- New: Add "files" entry to only deploy select files (Rob Loach)
- New: Add support and tests for typedefs. Refs #5 (Tom MacWright)

Chapter 56

README

56.1 Doctrine

Doctrine is a JSDoc parser that parses documentation comments from JavaScript (you need to pass in the comment, not a whole JavaScript file).

56.1.1 Installation

You can install Doctrine using npm:

```
$ npm install doctrine --save-dev
```

Doctrine can also be used in web browsers using Browserify.

56.1.2 Usage

Require doctrine inside of your JavaScript:

```
var doctrine = require("doctrine");
```

56.1.2.1 parse()

The primary method is `parse()`, which accepts two arguments: the JSDoc comment to parse and an optional options object. The available options are:

- `unwrap` - set to `true` to delete the leading `/**`, any `*` that begins a line, and the trailing `*/` from the source text. Default: `false`.
- `tags` - an array of tags to return. When specified, Doctrine returns only tags in this array. For example, if `tags` is `["param"]`, then only `@param` tags will be returned. Default: `null`.
- `recoverable` - set to `true` to keep parsing even when syntax errors occur. Default: `false`.
- `sloppy` - set to `true` to allow optional parameters to be specified in brackets (`@param {string} [foo]`). Default: `false`.
- `lineNumbers` - set to `true` to add `lineNumber` to each node, specifying the line on which the node is found in the source. Default: `false`.
- `range` - set to `true` to add `range` to each node, specifying the start and end index of the node in the original comment. Default: `false`.

Here's a simple example:

```
var ast = doctrine.parse(
    [
        "/**",
        " * This function comment is parsed by doctrine",
        " * @param {{ok:String}} userName",
        "*/"
    ].join('\n'), { unwrap:  true });
```

This example returns the following AST:

```
{
    "description": "This function comment is parsed by doctrine",
    "tags": [
        {
            "title": "param",
            "description": null,
            "type": {
                "type": "RecordType",
                "fields": [
                    {
                        "type": "FieldType",
                        "key": "ok",
                        "value": {
                            "type": "NameExpression",
                            "name": "String"
                        }
                    }
                ]
            },
            "name": "userName"
        }
    ]
}
```

See the demo page more detail.

56.1.3 Team

These folks keep the project moving and are resources for help:

- Nicholas C. Zakas (@nzakas) - project lead
- Yusuke Suzuki (@constellation) - reviewer

56.1.4 Contributing

Issues and pull requests will be triaged and responded to as quickly as possible. We operate under the ESLint Contributor Guidelines, so please be sure to read them before contributing. If you're not sure where to dig in, check out the issues.

56.1.5 Frequently Asked Questions

56.1.5.1 Can I pass a whole JavaScript file to Doctrine?

No. Doctrine can only parse JSDoc comments, so you'll need to pass just the JSDoc comment to Doctrine in order to work.

56.1.5.2 License

56.1.5.2.3 closure-compiler some of extensions is derived from closure-compiler
Apache License Version 2.0, January 2004 `http://www.apache.org/licenses/`

56.1.5.3 Where to ask for help?

Join our `Chatroom`

Chapter 57

EE First

![License][license-image]
Get the first event in a set of event emitters and event pairs, then clean up after itself.

57.1 Install

```
$ npm install ee-first
```

57.2 API

```
var first = require('ee-first')
```

57.2.1 first(arr, listener)

Invoke `listener` on the first event from the list specified in `arr`. `arr` is an array of arrays, with each array in the format `[ee, ...event]`. `listener` will be called only once, the first time any of the given events are emitted. If `error` is one of the listened events, then if that fires first, the `listener` will be given the `err` argument.
The `listener` is invoked as `listener(err, ee, event, args)`, where `err` is the first argument emitted from an `error` event, if applicable; `ee` is the event emitter that fired; `event` is the string event name that fired; and `args` is an array of the arguments that were emitted on the event.

```
var ee1 = new EventEmitter()
var ee2 = new EventEmitter()
first([
  [ee1, 'close', 'end', 'error'],
  [ee2, 'error']
], function (err, ee, event, args) {
  // listener invoked
})
```

57.2.1.1 .cancel()

The group of listeners can be cancelled before being invoked and have all the event listeners removed from the underlying event emitters.

```
var thunk = first([
  [ee1, 'close', 'end', 'error'],
  [ee2, 'error']
], function (err, ee, event, args) {
  // listener invoked
})
// cancel and clean up
thunk.cancel()
```

Chapter 58

Encode URL

Encode a URL to a percent-encoded form, excluding already-encoded sequences.

58.1 Installation

```
npm install encodeurl
```

58.2 API

```
var encodeUrl = require('encodeurl')
```

58.2.1 encodeUrl(url)

Encode a URL to a percent-encoded form, excluding already-encoded sequences.
This function accepts a URL and encodes all the non-URL code points (as UTF-8 byte sequences). It will not encode the "%" character unless it is not part of a valid sequence (`%20` will be left as-is, but `foo` will be encoded as `%25foo`).
This encode is meant to be "safe" and does not throw errors. It will try as hard as it can to properly encode the given URL, including replacing any raw, unpaired surrogate pairs with the Unicode replacement character prior to encoding.

58.3 Examples

58.3.1 Encode a URL containing user-controlled data

```
var encodeUrl = require('encodeurl')
var escapeHtml = require('escape-html')
http.createServer(function onRequest (req, res) {
  // get encoded form of inbound url
  var url = encodeUrl(req.url)
  // create html message
  var body = '<p>Location ' + escapeHtml(url) + ' not found</p>'
  // send a 404
  res.statusCode = 404
  res.setHeader('Content-Type', 'text/html; charset=UTF-8')
  res.setHeader('Content-Length', String(Buffer.byteLength(body, 'utf-8')))
  res.end(body, 'utf-8')
})
```

58.3.2 Encode a URL for use in a header field

```
var encodeUrl = require('encodeurl')
var escapeHtml = require('escape-html')
var url = require('url')
http.createServer(function onRequest (req, res) {
  // parse inbound url
  var href = url.parse(req)
  // set new host for redirect
  href.host = 'localhost'
  href.protocol = 'https:'
  href.slashes = true
  // create location header
  var location = encodeUrl(url.format(href))
  // create html message
  var body = '<p>Redirecting to new site: ' + escapeHtml(location) + '</p>'
  // send a 301
  res.statusCode = 301
  res.setHeader('Content-Type', 'text/html; charset=UTF-8')
  res.setHeader('Content-Length', String(Buffer.byteLength(body, 'utf-8')))
  res.setHeader('Location', location)
  res.end(body, 'utf-8')
})
```

58.4 Similarities

This function is *similar* to the intrinsic function `encodeURI`. However, it will not encode:

- The \, ^, or | characters
- The `%` character when it's part of a valid sequence
- `[` and `]` (for IPv6 hostnames)
- Replaces raw, unpaired surrogate pairs with the Unicode replacement character

As a result, the encoding aligns closely with the behavior in the `WHATWG URL specification`. However, this package only encodes strings and does not do any URL parsing or formatting.
It is expected that any output from `new URL(url)` will not change when used with this package, as the output has already been encoded. Additionally, if we were to encode before `new URL(url)`, we do not expect the before and after encoded formats to be parsed any differently.

58.5 Testing

```
$ npm test
$ npm run lint
```

58.6 References

- `RFC 3986: Uniform Resource Identifier (URI): Generic Syntax`
- `WHATWG URL Living Standard`

58.7 License

[MIT](LICENSE)

Chapter 59

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

59.1 v1.0.0 - 2024-02-12

59.1.1 Commits

- Initial implementation, tests, readme, types `3e154e1`
- Initial commit `07d98de`
- npm init `c4eb634`
- Only apps should have lockfiles `7af86ec`

Chapter 60

es-define-property <sup><a href="https://npmjs.org/package/es-define-property" ><img src="https://versionbadg.es/ljharb/es-define-property.svg" alt="Version Badge"/></a></sup>

[][license-url]

`Object.defineProperty`, but not IE 8's broken one.

60.1 Example

```
const assert = require('assert');
const $defineProperty = require('es-define-property');
if ($defineProperty) {
    assert.equal($defineProperty, Object.defineProperty);
} else if (Object.defineProperty) {
    assert.equal($defineProperty, false, 'this is IE 8');
} else {
    assert.equal($defineProperty, false, 'this is an ES3 engine');
}
```

60.2 Tests

Simply clone the repo, `npm install`, and run `npm test`

60.3 Security

Please email `@ljharb` or see `https://tidelift.com/security` if you have a potential security vulnerability to report.

Chapter 61

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

61.1 <a href="https://github.com/ljharb/es-errors/compare/v1.2.1...v1.3.0" >v1.3.0</a> - 2024-02-05

61.1.1 Commits

- [New] add `EvalError` and `URIError` `1927627`

61.2 <a href="https://github.com/ljharb/es-errors/compare/v1.2.0...v1.2.1" >v1.2.1</a> - 2024-02-04

61.2.1 Commits

- [Fix] add missing `exports` entry `5bb5f28`

61.3 <a href="https://github.com/ljharb/es-errors/compare/v1.1.0...v1.2.0" >v1.2.0</a> - 2024-02-04

61.3.1 Commits

- [New] add `ReferenceError` `6d8cf5b`

61.4 <a href="https://github.com/ljharb/es-errors/compare/v1.0.0...v1.1.0" >v1.1.0</a> - 2024-02-04

61.4.1 Commits

- [New] add base Error `2983ab6`

61.5 v1.0.0 - 2024-02-03

61.5.1 Commits

- Initial implementation, tests, readme, type `8f47631`
- Initial commit `ea5d099`
- npm init `6f5ebf9`
- Only apps should have lockfiles `e1a0aeb`
- [meta] add `sideEffects` flag `a9c7d46`

Chapter 62

es-errors <sup><a href="https://npmjs.org/package/es-errors" ><img src="https://versionbadg.es/ljharb/es-errors.svg" alt="Version Badge"/></a></sup>

[][license-url]

A simple cache for a few of the JS Error constructors.

62.1 Example

```
const assert = require('assert');
const Base = require('es-errors');
const Eval = require('es-errors/eval');
const Range = require('es-errors/range');
const Ref = require('es-errors/ref');
const Syntax = require('es-errors/syntax');
const Type = require('es-errors/type');
const URI = require('es-errors/uri');
assert.equal(Base, Error);
assert.equal(Eval, EvalError);
assert.equal(Range, RangeError);
assert.equal(Ref, ReferenceError);
assert.equal(Syntax, SyntaxError);
assert.equal(Type, TypeError);
assert.equal(URI, URIError);
```

62.2 Tests

Simply clone the repo, `npm install`, and run `npm test`

62.3 Security

Please email `@ljharb` or see `https://tidelift.com/security` if you have a potential security vulnerability to report.

Chapter 63

escape-html

Escape string for use in HTML

63.1 Example

```
var escape = require('escape-html');
var html = escape('foo & bar');
// -> foo &amp; bar
```

63.2 Benchmark

```
$ npm run-script bench
> escape-html@1.0.3 bench nodejs-escape-html
> node benchmark/index.js
  http_parser@1.0
  node@0.10.33
  v8@3.14.5.9
  ares@1.9.0-DEV
  uv@0.10.29
  zlib@1.2.3
  modules@11
  openssl@1.0.1j
  1 test completed.
  2 tests completed.
  3 tests completed.
  no special characters    x 19,435,271 ops/sec ±0.85% (187 runs sampled)
  single special character x  6,132,421 ops/sec ±0.67% (194 runs sampled)
  many special characters  x  3,175,826 ops/sec ±0.65% (193 runs sampled)
```

63.3 License

MIT

Chapter 64

esutils <a href="http://travis-ci.org/estools/esutils" ><img src="https://secure.travis-ci.↩ org/estools/esutils.svg" alt="Build Status"/></a>

esutils (`esutils`) is utility box for ECMAScript language tools.

64.0.1 API

64.0.2 ast

64.0.2.1 ast.isExpression(node)

Returns true if `node` is an Expression as defined in ECMA262 edition 5.1 section `11`.

64.0.2.2 ast.isStatement(node)

Returns true if `node` is a Statement as defined in ECMA262 edition 5.1 section `12`.

64.0.2.3 ast.isIterationStatement(node)

Returns true if `node` is an IterationStatement as defined in ECMA262 edition 5.1 section `12.6`.

64.0.2.4 ast.isSourceElement(node)

Returns true if `node` is a SourceElement as defined in ECMA262 edition 5.1 section `14`.

64.0.2.5 ast.trailingStatement(node)

Returns `Statement?` if `node` has trailing `Statement`.

```
if (cond)
    consequent;
```

When taking this `IfStatement`, returns `consequent;` statement.

64.0.2.6 ast.isProblematicIfStatement(node)

Returns true if `node` is a problematic IfStatement. If `node` is a problematic `IfStatement`, `node` cannot be represented as an one on one JavaScript code.

```
{
    type: 'IfStatement',
    consequent: {
        type: 'WithStatement',
        body: {
            type: 'IfStatement',
            consequent: {type: 'EmptyStatement'}
        }
    },
    alternate: {type: 'EmptyStatement'}
}
```

The above node cannot be represented as a JavaScript code, since the top level `else` alternate belongs to an inner `IfStatement`.

64.0.3 code

64.0.3.1 code.isDecimalDigit(code)

Return true if provided code is decimal digit.

64.0.3.2 code.isHexDigit(code)

Return true if provided code is hexadecimal digit.

64.0.3.3 code.isOctalDigit(code)

Return true if provided code is octal digit.

64.0.3.4 code.isWhiteSpace(code)

Return true if provided code is white space. White space characters are formally defined in ECMA262.

64.0.3.5 code.isLineTerminator(code)

Return true if provided code is line terminator. Line terminator characters are formally defined in ECMA262.

64.0.3.6 code.isIdentifierStart(code)

Return true if provided code can be the first character of ECMA262 Identifier. They are formally defined in ECMA262.

64.0.3.7 code.isIdentifierPart(code)

Return true if provided code can be the trailing character of ECMA262 Identifier. They are formally defined in ECMA262.

64.0.4 keyword

64.0.4.1 keyword.isKeywordES5(id, strict)

Returns `true` if provided identifier string is a Keyword or Future Reserved Word in ECMA262 edition 5.1. They are formally defined in ECMA262 sections `7.6.1.1` and `7.6.1.2`, respectively. If the `strict` flag is truthy, this function additionally checks whether `id` is a Keyword or Future Reserved Word under strict mode.

64.0.4.2 keyword.isKeywordES6(id, strict)

Returns `true` if provided identifier string is a Keyword or Future Reserved Word in ECMA262 edition 6. They are formally defined in ECMA262 sections `11.6.2.1` and `11.6.2.2`, respectively. If the `strict` flag is truthy, this function additionally checks whether `id` is a Keyword or Future Reserved Word under strict mode.

64.0.4.3 keyword.isReservedWordES5(id, strict)

Returns `true` if provided identifier string is a Reserved Word in ECMA262 edition 5.1. They are formally defined in ECMA262 section `7.6.1`. If the `strict` flag is truthy, this function additionally checks whether `id` is a Reserved Word under strict mode.

64.0.4.4 keyword.isReservedWordES6(id, strict)

Returns `true` if provided identifier string is a Reserved Word in ECMA262 edition 6. They are formally defined in ECMA262 section `11.6.2`. If the `strict` flag is truthy, this function additionally checks whether `id` is a Reserved Word under strict mode.

64.0.4.5 keyword.isRestrictedWord(id)

Returns `true` if provided identifier string is one of `eval` or `arguments`. They are restricted in strict mode code throughout ECMA262 edition 5.1 and in ECMA262 edition 6 section `12.1.1`.

64.0.4.6 keyword.isIdentifierNameES5(id)

Return true if provided identifier string is an IdentifierName as specified in ECMA262 edition 5.1 section `7.6`.

64.0.4.7 keyword.isIdentifierNameES6(id)

Return true if provided identifier string is an IdentifierName as specified in ECMA262 edition 6 section `11.6`.

64.0.4.8 keyword.isIdentifierES5(id, strict)

Return true if provided identifier string is an Identifier as specified in ECMA262 edition 5.1 section `7.6`. If the `strict` flag is truthy, this function additionally checks whether `id` is an Identifier under strict mode.

64.0.4.9 keyword.isIdentifierES6(id, strict)

Return true if provided identifier string is an Identifier as specified in ECMA262 edition 6 section `12.1`. If the `strict` flag is truthy, this function additionally checks whether `id` is an Identifier under strict mode.

64.0.5 License

Chapter 65

1.8.1 / 2017-09-12

- perf: replace regular expression with substring

65.1 1.8.0 / 2017-02-18

- Use SHA1 instead of MD5 for ETag hashing
  - Improves performance for larger entities
  - Works with FIPS 140-2 OpenSSL configuration

65.2 1.7.0 / 2015-06-08

- Always include entity length in ETags for hash length extensions
- Generate non-Stats ETags using MD5 only (no longer CRC32)
- Improve stat performance by removing hashing
- Remove base64 padding in ETags to shorten
- Use MD5 instead of MD4 in weak ETags over 1KB

65.3 1.6.0 / 2015-05-10

- Improve support for JXcore
- Remove requirement of `atime` in the stats object
- Support "fake" stats objects in environments without `fs`

65.4 1.5.1 / 2014-11-19

- deps: crc@3.2.1
  - Minor fixes

65.5 1.5.0 / 2014-10-14

- Improve string performance
- Slightly improve speed for weak ETags over 1KB

65.6 1.4.0 / 2014-09-21

- Support "fake" stats objects
- Support Node.js 0.6

65.7 1.3.1 / 2014-09-14

- Use the (new and improved) `crc` for crc32

65.8 1.3.0 / 2014-08-29

- Default strings to strong ETags
- Improve speed for weak ETags over 1KB

65.9 1.2.1 / 2014-08-29

- Use the (much faster) `buffer-crc32` for crc32

65.10 1.2.0 / 2014-08-24

- Add support for file stat objects

65.11 1.1.0 / 2014-08-24

- Add fast-path for empty entity
- Add weak ETag generation
- Shrink size of generated ETags

65.12 1.0.1 / 2014-08-24

- Fix behavior of string containing Unicode

65.13 1.0.0 / 2014-05-18

- Initial release

Chapter 66

etag

Create simple HTTP ETags
This module generates HTTP ETags (as defined in RFC 7232) for use in HTTP responses.

66.1 Installation

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install etag
```

66.2 API

```
var etag = require('etag')
```

66.2.1 etag(entity, [options])

Generate a strong ETag for the given entity. This should be the complete body of the entity. Strings, `Buffer`s, and `fs.Stats` are accepted. By default, a strong ETag is generated except for `fs.Stats`, which will generate a weak ETag (this can be overwritten by `options.weak`).

```
res.setHeader('ETag', etag(body))
```

66.2.1.1 Options

`etag` accepts these properties in the options object.

66.2.1.1.1 weak Specifies if the generated ETag will include the weak validator mark (that is, the leading `W/`). The actual entity tag is the same. The default value is `false`, unless the `entity` is `fs.Stats`, in which case it is `true`.

66.3 Testing

```
$ npm test
```

66.4 Benchmark

```
$ npm run-script bench
> etag@1.8.1 bench nodejs-etag
> node benchmark/index.js
  http_parser@2.7.0
  node@6.11.1
  v8@5.1.281.103
  uv@1.11.0
  zlib@1.2.11
  ares@1.10.1-DEV
  icu@58.2
  modules@48
  openssl@1.0.2k
> node benchmark/body0-100b.js
  100B body
  4 tests completed.
  buffer - strong x 258,647 ops/sec ±1.07% (180 runs sampled)
  buffer - weak   x 263,812 ops/sec ±0.61% (184 runs sampled)
  string - strong x 259,955 ops/sec ±1.19% (185 runs sampled)
  string - weak   x 264,356 ops/sec ±1.09% (184 runs sampled)
> node benchmark/body1-1kb.js
  1KB body
  4 tests completed.
  buffer - strong x 189,018 ops/sec ±1.12% (182 runs sampled)
  buffer - weak   x 190,586 ops/sec ±0.81% (186 runs sampled)
  string - strong x 144,272 ops/sec ±0.96% (188 runs sampled)
  string - weak   x 145,380 ops/sec ±1.43% (187 runs sampled)
> node benchmark/body2-5kb.js
  5KB body
  4 tests completed.
  buffer - strong x 92,435 ops/sec ±0.42% (188 runs sampled)
  buffer - weak   x 92,373 ops/sec ±0.58% (189 runs sampled)
  string - strong x 48,850 ops/sec ±0.56% (186 runs sampled)
  string - weak   x 49,380 ops/sec ±0.56% (190 runs sampled)
> node benchmark/body3-10kb.js
  10KB body
  4 tests completed.
  buffer - strong x 55,989 ops/sec ±0.93% (188 runs sampled)
  buffer - weak   x 56,148 ops/sec ±0.55% (190 runs sampled)
  string - strong x 27,345 ops/sec ±0.43% (188 runs sampled)
  string - weak   x 27,496 ops/sec ±0.45% (190 runs sampled)
> node benchmark/body4-100kb.js
  100KB body
  4 tests completed.
  buffer - strong x 7,083 ops/sec ±0.22% (190 runs sampled)
  buffer - weak   x 7,115 ops/sec ±0.26% (191 runs sampled)
  string - strong x 3,068 ops/sec ±0.34% (190 runs sampled)
  string - weak   x 3,096 ops/sec ±0.35% (190 runs sampled)
> node benchmark/stats.js
  stat
  4 tests completed.
  real - strong x 871,642 ops/sec ±0.34% (189 runs sampled)
  real - weak   x 867,613 ops/sec ±0.39% (190 runs sampled)
  fake - strong x 401,051 ops/sec ±0.40% (189 runs sampled)
  fake - weak   x 400,100 ops/sec ±0.47% (188 runs sampled)
```

66.5 License

[MIT](LICENSE)

Chapter 67

4.21.0 / 2024-09-11

- Deprecate `res.location("back")` and `res.redirect("back")` magic string
- deps: `serve-static@1.16`.2
  - includes `send@0.19`.0
- deps: finalhandler@1.3.1
- deps: `qs@6.13`.0

67.1 4.20.0 / 2024-09-10

- deps: `serve-static@0.16`.0
  - Remove link renderization in html while redirecting
- deps: `send@0.19`.0
  - Remove link renderization in html while redirecting
- deps: body-parser@0.6.0
  - add `depth` option to customize the depth level in the parser
  - IMPORTANT: The default `depth` level for parsing URL-encoded data is now `32` (previously was `Infinity`)
- Remove link renderization in html while using `res.redirect`
- deps: `path-to-regexp@0.1.10`
  - Adds support for named matching groups in the routes using a regex
  - Adds backtracking protection to parameters without regexes defined
- deps: encodeurl2.0.0
  - Removes encoding of \, |, and ^ to align better with URL spec
- Deprecate passing `options.maxAge` and `options.expires` to `res.clearCookie`
  - Will be ignored in v5, clearCookie will set a cookie with an expires in the past to instruct clients to delete the cookie

67.2 4.19.2 / 2024-03-25

- Improved fix for open redirect allow list bypass

67.3 4.19.1 / 2024-03-20

- Allow passing non-strings to res.location with new encoding handling checks

67.4 4.19.0 / 2024-03-20

- Prevent open redirect allow list bypass due to encodeurl
- deps: cookie@0.6.0

67.5 4.18.3 / 2024-02-29

- Fix routing requests without method
- deps: `body-parser@1.20`.2
  - Fix strict json error message on Node.js 19+
  - deps: content-type1.0.5
  - deps: raw-body@2.5.2
- deps: cookie@0.6.0
  - Add `partitioned` option

67.6 4.18.2 / 2022-10-08

- Fix regression routing a large stack in a single route
- deps: `body-parser@1.20`.1
  - deps: `qs@6.11`.0
  - perf: remove unnecessary object clone
- deps: `qs@6.11`.0

67.7 4.18.1 / 2022-04-29

- Fix hanging on large stack of sync routes

67.8 4.18.0 / 2022-04-25

- Add "root" option to `res.download`
- Allow `options` without `filename` in `res.download`
- Deprecate string and non-integer arguments to `res.status`
- Fix behavior of `null`/`undefined` as `maxAge` in `res.cookie`
- Fix handling very large stacks of sync middleware
- Ignore `Object.prototype` values in settings through `app.set`/`app.get`
- Invoke `default` with same arguments as types in `res.format`
- Support proper 205 responses using `res.send`
- Use `http-errors` for `res.format` error
- deps: `body-parser@1.20`.0

  - Fix error message for json parse whitespace in `strict`
  - Fix internal error when inflated body exceeds limit
  - Prevent loss of async hooks context
  - Prevent hanging when request already read
  - deps: depd@2.0.0
  - deps: http-errors@2.0.0
  - deps: on-finished@2.4.1
  - deps: `qs@6.10`.3
  - deps: raw-body@2.5.1
- deps: cookie@0.5.0
  - Add `priority` option
  - Fix `expires` option to reject invalid dates
- deps: depd@2.0.0
  - Replace internal `eval` usage with `Function` constructor
  - Use instance methods on `process` to check for listeners
- deps: finalhandler@1.2.0
  - Remove set content headers that break response
  - deps: on-finished@2.4.1
  - deps: statuses@2.0.1
- deps: on-finished@2.4.1
  - Prevent loss of async hooks context
- deps: `qs@6.10`.3
- deps: `send@0.18`.0
  - Fix emitted 416 error missing headers property
  - Limit the headers removed for 304 response
  - deps: depd@2.0.0
  - deps: destroy@1.2.0
  - deps: http-errors@2.0.0
  - deps: on-finished@2.4.1
  - deps: statuses@2.0.1
- deps: `serve-static@1.15`.0
  - deps: `send@0.18`.0
- deps: statuses@2.0.1
  - Remove code 306
  - Rename `425 Unordered Collection` to standard `425 Too Early`

67.9 4.17.3 / 2022-02-16

- deps: accepts1.3.8
  - deps: mime-types2.1.34
  - deps: negotiator@0.6.3
- deps: `body-parser@1.19`.2
  - deps: bytes@3.1.2
  - deps: qs@6.9.7
  - deps: raw-body@2.4.3
- deps: cookie@0.4.2
- deps: qs@6.9.7
  - Fix handling of __`proto`__ keys
- pref: remove unnecessary regexp for trust proxy

67.10 4.17.2 / 2021-12-16

- Fix handling of `undefined` in `res.jsonp`
- Fix handling of `undefined` when `"json escape"` is enabled
- Fix incorrect middleware execution with unanchored `RegExp`s
- Fix `res.jsonp(obj, status)` deprecation message
- Fix typo in `res.is` JSDoc
- deps: `body-parser@1.19`.1
  - deps: bytes@3.1.1
  - deps: http-errors@1.8.1
  - deps: qs@6.9.6
  - deps: raw-body@2.4.2
  - deps: safe-buffer@5.2.1
  - deps: type-is1.6.18
- deps: content-disposition@0.5.4
  - deps: safe-buffer@5.2.1
- deps: cookie@0.4.1
  - Fix `maxAge` option to reject invalid values
- deps: proxy-addr2.0.7
  - Use `req.socket` over deprecated `req.connection`
  - deps: forwarded@0.2.0
  - deps: ipaddr.js@1.9.1
- deps: qs@6.9.6
- deps: safe-buffer@5.2.1
- deps: `send@0.17`.2
  - deps: http-errors@1.8.1

  - deps: ms@2.1.3
  - pref: ignore empty http tokens
- deps: `serve-static@1.14`.2
  - deps: `send@0.17`.2
- deps: setprototypeof@1.2.0

67.11 4.17.1 / 2019-05-25

- Revert "Improve error message for \`null\`/\`undefined\` to \`res.status\`"

67.12 4.17.0 / 2019-05-16

- Add `express.raw` to parse bodies into `Buffer`
- Add `express.text` to parse bodies into string
- Improve error message for non-strings to `res.sendFile`
- Improve error message for `null/undefined` to `res.status`
- Support multiple hosts in `X-Forwarded-Host`
- deps: accepts1.3.7
- deps: `body-parser@1.19`.0
  - Add encoding MIK
  - Add petabyte (`pb`) support
  - Fix parsing array brackets after index
  - deps: bytes@3.1.0
  - deps: http-errors@1.7.2
  - deps: `iconv-lite@0.4.24`
  - deps: qs@6.7.0
  - deps: raw-body@2.4.0
  - deps: type-is1.6.17
- deps: content-disposition@0.5.3
- deps: cookie@0.4.0
  - Add `SameSite=None` support
- deps: finalhandler1.1.2
  - Set stricter `Content-Security-Policy` header
  - deps: parseurl1.3.3
  - deps: statuses1.5.0
- deps: parseurl1.3.3
- deps: proxy-addr2.0.5
  - deps: ipaddr.js@1.9.0
- deps: qs@6.7.0
  - Fix parsing array brackets after index

- deps: range-parser1.2.1
- deps: send@0.17.1
  - Set stricter CSP header in redirect & error responses
  - deps: http-errors1.7.2
  - deps: mime@1.6.0
  - deps: ms@2.1.1
  - deps: range-parser1.2.1
  - deps: statuses1.5.0
  - perf: remove redundant `path.normalize` call
- deps: serve-static@1.14.1
  - Set stricter CSP header in redirect response
  - deps: parseurl1.3.3
  - deps: send@0.17.1
- deps: setprototypeof@1.1.1
- deps: statuses1.5.0
  - Add `103 Early Hints`
- deps: type-is1.6.18
  - deps: mime-types2.1.24
  - perf: prevent internal `throw` on invalid type

67.13 4.16.4 / 2018-10-10

- Fix issue where `"Request aborted"` may be logged in `res.sendfile`
- Fix JSDoc for `Router` constructor
- deps: body-parser@1.18.3
  - Fix deprecation warnings on Node.js 10+
  - Fix stack trace for strict json parse error
  - deps: depd1.1.2
  - deps: http-errors1.6.3
  - deps: iconv-lite@0.4.23
  - deps: qs@6.5.2
  - deps: raw-body@2.3.3
  - deps: type-is1.6.16
- deps: proxy-addr2.0.4
  - deps: ipaddr.js@1.8.0
- deps: qs@6.5.2
- deps: safe-buffer@5.1.2

67.14 4.16.3 / 2018-03-12

- deps: accepts1.3.5
  - deps: mime-types2.1.18
- deps: depd1.1.2
  - perf: remove argument reassignment
- deps: encodeurl1.0.2
  - Fix encoding `%` as last character
- deps: finalhandler@1.1.1
  - Fix 404 output for bad / missing pathnames
  - deps: encodeurl1.0.2
  - deps: statuses1.4.0
- deps: proxy-addr2.0.3
  - deps: ipaddr.js@1.6.0
- deps: `send@0.16`.2
  - Fix incorrect end tag in default error & redirects
  - deps: depd1.1.2
  - deps: encodeurl1.0.2
  - deps: statuses1.4.0
- deps: `serve-static@1.13`.2
  - Fix incorrect end tag in redirects
  - deps: encodeurl1.0.2
  - deps: `send@0.16`.2
- deps: statuses1.4.0
- deps: type-is1.6.16
  - deps: mime-types2.1.18

67.15 4.16.2 / 2017-10-09

- Fix `TypeError` in `res.send` when given `Buffer` and `ETag` header set
- perf: skip parsing of entire `X-Forwarded-Proto` header

67.16 4.16.1 / 2017-09-29

- deps: `send@0.16`.1
- deps: `serve-static@1.13`.1
  - Fix regression when `root` is incorrectly set to a file
  - deps: `send@0.16`.1

67.17 4.16.0 / 2017-09-28

- Add `"json escape"` setting for `res.json` and `res.jsonp`
- Add `express.json` and `express.urlencoded` to parse bodies
- Add `options` argument to `res.download`
- Improve error message when autoloading invalid view engine
- Improve error messages when non-function provided as middleware
- Skip `Buffer` encoding when not generating ETag for small response
- Use `safe-buffer` for improved Buffer API
- deps: accepts1.3.4
  - deps: mime-types2.1.16
- deps: content-type1.0.4
  - perf: remove argument reassignment
  - perf: skip parameter parsing when no parameters
- deps: etag1.8.1
  - perf: replace regular expression with substring
- deps: finalhandler@1.1.0
  - Use `res.headersSent` when available
- deps: parseurl1.3.2
  - perf: reduce overhead for full URLs
  - perf: unroll the "fast-path" `RegExp`
- deps: proxy-addr2.0.2
  - Fix trimming leading / trailing OWS in `X-Forwarded-For`
  - deps: forwarded0.1.2
  - deps: ipaddr.js@1.5.2
  - perf: reduce overhead when no `X-Forwarded-For` header
- deps: qs@6.5.1
  - Fix parsing & compacting very deep objects
- deps: `send@0.16`.0
  - Add 70 new types for file extensions
  - Add `immutable` option
  - Fix missing `</html>` in default error & redirects
  - Set charset as "UTF-8" for .js and .json
  - Use instance methods on steam to check for listeners
  - deps: mime@1.4.1
  - perf: improve path validation speed
- deps: `serve-static@1.13`.0
  - Add 70 new types for file extensions
  - Add `immutable` option

  - Set charset as "UTF-8" for .js and .json
  - deps: `send@0.16`.0

- deps: setprototypeof@1.1.0
- deps: utils-merge@1.0.1
- deps: vary1.1.2
  - perf: improve header token parsing speed
- perf: re-use options object when generating ETags
- perf: remove dead `.charset` set in `res.jsonp`

67.18 4.15.5 / 2017-09-24

- deps: debug@2.6.9
- deps: finalhandler1.0.6
  - deps: debug@2.6.9
  - deps: parseurl1.3.2
- deps: fresh@0.5.2
  - Fix handling of modified headers with invalid dates
  - perf: improve ETag match loop
  - perf: improve `If-None-Match` token parsing
- deps: `send@0.15`.6
  - Fix handling of modified headers with invalid dates
  - deps: debug@2.6.9
  - deps: etag1.8.1
  - deps: fresh@0.5.2
  - perf: improve `If-Match` token parsing
- deps: `serve-static@1.12`.6
  - deps: parseurl1.3.2
  - deps: `send@0.15`.6
  - perf: improve slash collapsing

67.19 4.15.4 / 2017-08-06

- deps: debug@2.6.8
- deps: depd1.1.1
  - Remove unnecessary `Buffer` loading
- deps: finalhandler1.0.4
  - deps: debug@2.6.8
- deps: proxy-addr1.1.5
  - Fix array argument being altered
  - deps: ipaddr.js@1.4.0

- deps: qs@6.5.0
- deps: `send@0.15`.4
  - deps: debug@2.6.8
  - deps: depd1.1.1
  - deps: http-errors1.6.2
- deps: `serve-static@1.12`.4
  - deps: `send@0.15`.4

67.20 4.15.3 / 2017-05-16

- Fix error when `res.set` cannot add charset to `Content-Type`
- deps: debug@2.6.7
  - Fix `DEBUG_MAX_ARRAY_LENGTH`
  - deps: ms@2.0.0
- deps: finalhandler1.0.3
  - Fix missing `</html>` in HTML document
  - deps: debug@2.6.7
- deps: proxy-addr1.1.4
  - deps: ipaddr.js@1.3.0
- deps: `send@0.15`.3
  - deps: debug@2.6.7
  - deps: ms@2.0.0
- deps: `serve-static@1.12`.3
  - deps: `send@0.15`.3
- deps: type-is1.6.15
  - deps: mime-types2.1.15
- deps: vary1.1.1
  - perf: hoist regular expression

67.21 4.15.2 / 2017-03-06

- deps: qs@6.4.0
  - Fix regression parsing keys starting with `[`

67.22 4.15.1 / 2017-03-05

- deps: `send@0.15`.1
  - Fix issue when `Date.parse` does not return `NaN` on invalid date
  - Fix strict violation in broken environments
- deps: `serve-static@1.12`.1
  - Fix issue when `Date.parse` does not return `NaN` on invalid date
  - deps: `send@0.15`.1

67.23 4.15.0 / 2017-03-01

- Add debug message when loading view engine
- Add `next("router")` to exit from router
- Fix case where `router.use` skipped requests routes did not
- Remove usage of `res._headers` private field
  - Improves compatibility with Node.js 8 nightly
- Skip routing when `req.url` is not set
- Use `o` in path debug to tell types apart
- Use `Object.create` to setup request & response prototypes
- Use `setprototypeof` module to replace `__proto__` setting
- Use `statuses` instead of `http` module for status messages
- deps: debug@2.6.1
  - Allow colors in workers
  - Deprecated `DEBUG_FD` environment variable set to `3` or higher
  - Fix error when running under React Native
  - Use same color for same namespace
  - deps: ms@0.7.2
- deps: etag1.8.0
  - Use SHA1 instead of MD5 for ETag hashing
  - Works with FIPS 140-2 OpenSSL configuration
- deps: finalhandler1.0.0
  - Fix exception when `err` cannot be converted to a string
  - Fully URL-encode the pathname in the 404
  - Only include the pathname in the 404 message
  - Send complete HTML document
  - Set ‘Content-Security-Policy: default-src 'self'` header
  - deps: debug@2.6.1
- deps: fresh@0.5.0
  - Fix false detection of `no-cache` request directive
  - Fix incorrect result when `If-None-Match` has both `*` and ETags
  - Fix weak `ETag` matching to match spec
  - perf: delay reading header values until needed
  - perf: enable strict mode
  - perf: hoist regular expressions
  - perf: remove duplicate conditional
  - perf: remove unnecessary boolean coercions
  - perf: skip checking modified time if ETag check failed
  - perf: skip parsing `If-None-Match` when no `ETag` header
  - perf: use `Date.parse` instead of `new Date`
- deps: qs@6.3.1

  - Fix array parsing from skipping empty values
  - Fix compacting nested arrays
- deps: send@0.15.0
  - Fix false detection of `no-cache` request directive
  - Fix incorrect result when `If-None-Match` has both * and ETags
  - Fix weak `ETag` matching to match spec
  - Remove usage of `res._headers` private field
  - Support `If-Match` and `If-Unmodified-Since` headers
  - Use `res.getHeaderNames()` when available
  - Use `res.headersSent` when available
  - deps: debug@2.6.1
  - deps: etag1.8.0
  - deps: fresh@0.5.0
  - deps: http-errors1.6.1
- deps: serve-static@1.12.0
  - Fix false detection of `no-cache` request directive
  - Fix incorrect result when `If-None-Match` has both * and ETags
  - Fix weak `ETag` matching to match spec
  - Remove usage of `res._headers` private field
  - Send complete HTML document in redirect response
  - Set default CSP header in redirect response
  - Support `If-Match` and `If-Unmodified-Since` headers
  - Use `res.getHeaderNames()` when available
  - Use `res.headersSent` when available
  - deps: send@0.15.0
- perf: add fast match path for * route
- perf: improve `req.ips` performance

67.24 4.14.1 / 2017-01-28

- deps: content-disposition@0.5.2
- deps: finalhandler@0.5.1
  - Fix exception when `err.headers` is not an object
  - deps: statuses1.3.1
  - perf: hoist regular expressions
  - perf: remove duplicate validation path
- deps: proxy-addr1.1.3
  - deps: ipaddr.js@1.2.0
- deps: send@0.14.2
  - deps: http-errors1.5.1
  - deps: ms@0.7.2
  - deps: statuses1.3.1

- deps: serve-static1.11.2
  - deps: `send@0.14`.2
- deps: type-is1.6.14
  - deps: mime-types2.1.13

67.25 4.14.0 / 2016-06-16

- Add `acceptRanges` option to `res.sendFile`/`res.sendfile`
- Add `cacheControl` option to `res.sendFile`/`res.sendfile`
- Add `options` argument to `req.range`
  - Includes the `combine` option
- Encode URL in `res.location`/`res.redirect` if not already encoded
- Fix some redirect handling in `res.sendFile`/`res.sendfile`
- Fix Windows absolute path check using forward slashes
- Improve error with invalid arguments to `req.get()`
- Improve performance for `res.json`/`res.jsonp` in most cases
- Improve `Range` header handling in `res.sendFile`/`res.sendfile`
- deps: accepts1.3.3
  - Fix including type extensions in parameters in `Accept` parsing
  - Fix parsing `Accept` parameters with quoted equals
  - Fix parsing `Accept` parameters with quoted semicolons
  - Many performance improvements
  - deps: mime-types2.1.11
  - deps: negotiator@0.6.1
- deps: content-type1.0.2
  - perf: enable strict mode
- deps: cookie@0.3.1
  - Add `sameSite` option
  - Fix cookie `Max-Age` to never be a floating point number
  - Improve error message when `encode` is not a function
  - Improve error message when `expires` is not a `Date`
  - Throw better error for invalid argument to parse
  - Throw on invalid values provided to `serialize`
  - perf: enable strict mode
  - perf: hoist regular expression
  - perf: use for loop in parse
  - perf: use string concatenation for serialization
- deps: finalhandler@0.5.0
  - Change invalid or non-numeric status code to 500
  - Overwrite status message to match set status code

  - Prefer `err.statusCode` if `err.status` is invalid
  - Set response headers from `err.headers` object
  - Use `statuses` instead of `http` module for status messages
- deps: proxy-addr1.1.2
  - Fix accepting various invalid netmasks
  - Fix IPv6-mapped IPv4 validation edge cases
  - IPv4 netmasks must be contiguous
  - IPv6 addresses cannot be used as a netmask
  - deps: ipaddr.js@1.1.1
- deps: qs@6.2.0
  - Add `decoder` option in `parse` function
- deps: range-parser1.2.0
  - Add `combine` option to combine overlapping ranges
  - Fix incorrectly returning -1 when there is at least one valid range
  - perf: remove internal function
- deps: `send@0.14`.1
  - Add `acceptRanges` option
  - Add `cacheControl` option
  - Attempt to combine multiple ranges into single range
  - Correctly inherit from `Stream` class
  - Fix `Content-Range` header in 416 responses when using `start/end` options
  - Fix `Content-Range` header missing from default 416 responses
  - Fix redirect error when `path` contains raw non-URL characters
  - Fix redirect when `path` starts with multiple forward slashes
  - Ignore non-byte `Range` headers
  - deps: http-errors1.5.0
  - deps: range-parser1.2.0
  - deps: statuses1.3.0
  - perf: remove argument reassignment
- deps: serve-static1.11.1
  - Add `acceptRanges` option
  - Add `cacheControl` option
  - Attempt to combine multiple ranges into single range
  - Fix redirect error when `req.url` contains raw non-URL characters
  - Ignore non-byte `Range` headers
  - Use status code 301 for redirects
  - deps: `send@0.14`.1
- deps: type-is1.6.13
  - Fix type error when given invalid type to match against
  - deps: mime-types2.1.11
- deps: vary1.1.0
  - Only accept valid field names in the `field` argument
- perf: use strict equality when possible

67.26 4.13.4 / 2016-01-21

- deps: content-disposition@0.5.1
  - perf: enable strict mode
- deps: cookie@0.1.5
  - Throw on invalid values provided to `serialize`
- deps: depd1.1.0
  - Support web browser loading
  - perf: enable strict mode
- deps: escape-html1.0.3
  - perf: enable strict mode
  - perf: optimize string replacement
  - perf: use faster string coercion
- deps: finalhandler@0.4.1
  - deps: escape-html1.0.3
- deps: merge-descriptors@1.0.1
  - perf: enable strict mode
- deps: methods1.1.2
  - perf: enable strict mode
- deps: parseurl1.3.1
  - perf: enable strict mode
- deps: proxy-addr1.0.10
  - deps: ipaddr.js@1.0.5
  - perf: enable strict mode
- deps: range-parser1.0.3
  - perf: enable strict mode
- deps: `send@0.13`.1
  - deps: depd1.1.0
  - deps: destroy1.0.4
  - deps: escape-html1.0.3
  - deps: range-parser1.0.3
- deps: serve-static1.10.2
  - deps: escape-html1.0.3
  - deps: parseurl1.3.0
  - deps: `send@0.13`.1

67.27 4.13.3 / 2015-08-02

- Fix infinite loop condition using `mergeParams: true`
- Fix inner numeric indices incorrectly altering parent `req.params`

67.28 4.13.2 / 2015-07-31

- deps: accepts1.2.12
  - deps: mime-types2.1.4
- deps: array-flatten@1.1.1
  - perf: enable strict mode
- deps: path-to-regexp@0.1.7
  - Fix regression with escaped round brackets and matching groups
- deps: type-is1.6.6
  - deps: mime-types2.1.4

67.29 4.13.1 / 2015-07-05

- deps: accepts1.2.10
  - deps: mime-types2.1.2
- deps: qs@4.0.0
  - Fix dropping parameters like `hasOwnProperty`
  - Fix various parsing edge cases
- deps: type-is1.6.4
  - deps: mime-types2.1.2
  - perf: enable strict mode
  - perf: remove argument reassignment

67.30 4.13.0 / 2015-06-20

- Add settings to debug output
- Fix `res.format` error when only `default` provided
- Fix issue where ‘next('route')`in`app.param`would incorrectly skip values`
- `Fix hiding platform issues with`decodeURIComponent
  - `OnlyURIErrors are a 400`
- `Fix using*` before params in routes`
- `Fix using capture groups before params in routes`
- `Simplify res.cookie to call res.append`
- `Use array-flatten module for flattening arrays`
- `deps: accepts1.2.9`
  - `deps: mime-types2.1.1`
  - `perf: avoid argument reassignment & argument slice`
  - `perf: avoid negotiator recursive construction`
  - `perf: enable strict mode`
  - `perf: remove unnecessary bitwise operator`

- deps: cookie@0.1.3
  - perf: deduce the scope of try-catch deopt
  - perf: remove argument reassignments
- deps: escape-html@1.0.2
- deps: etag1.7.0
  - Always include entity length in ETags for hash length extensions
  - Generate non-Stats ETags using MD5 only (no longer CRC32)
  - Improve stat performance by removing hashing
  - Improve support for JXcore
  - Remove base64 padding in ETags to shorten
  - Support "fake" stats objects in environments without fs
  - Use MD5 instead of MD4 in weak ETags over 1KB
- deps: finalhandler@0.4.0
  - Fix a false-positive when unpiping in Node.js 0.8
  - Support statusCode property on Error objects
  - Use unpipe module for unpiping requests
  - deps: escape-html@1.0.2
  - deps: on-finished2.3.0
  - perf: enable strict mode
  - perf: remove argument reassignment
- deps: fresh@0.3.0
  - Add weak ETag matching support
- deps: on-finished2.3.0
  - Add defined behavior for HTTP CONNECT requests
  - Add defined behavior for HTTP Upgrade requests
  - deps: ee-first@1.1.1
- deps: path-to-regexp@0.1.6
- deps: send@0.13.0
  - Allow Node.js HTTP server to set Date response header
  - Fix incorrectly removing Content-Location on 304 response
  - Improve the default redirect response headers
  - Send appropriate headers on default error response
  - Use http-errors for standard emitted errors
  - Use statuses instead of http module for status messages
  - deps: escape-html@1.0.2
  - deps: etag1.7.0
  - deps: fresh@0.3.0
  - deps: on-finished2.3.0
  - perf: enable strict mode
  - perf: remove unnecessary array allocations
- deps: serve-static1.10.0

  - Add fallthrough option
  - Fix reading options from options prototype
  - Improve the default redirect response headers
  - Malformed URLs now next() instead of 400
  - deps: escape-html@1.0.2
  - deps: send@0.13.0
  - perf: enable strict mode
  - perf: remove argument reassignment
- deps: type-is1.6.3
  - deps: mime-types2.1.1
  - perf: reduce try block size
  - perf: remove bitwise operations
- perf: enable strict mode
- perf: isolate app.render try block
- perf: remove argument reassignments in application
- perf: remove argument reassignments in request prototype
- perf: remove argument reassignments in response prototype
- perf: remove argument reassignments in routing
- perf: remove argument reassignments in View
- perf: skip attempting to decode zero length string
- perf: use saved reference to http.STATUS_CODES

67.31 4.12.4 / 2015-05-17

- deps: accepts1.2.7
  - deps: mime-types2.0.11
  - deps: negotiator@0.5.3
- deps: debug2.2.0
  - deps: ms@0.7.1
- deps: depd1.0.1
- deps: etag1.6.0
  - Improve support for JXcore
  - Support "fake" stats objects in environments without fs
- deps: finalhandler@0.3.6
  - deps: debug2.2.0
  - deps: on-finished2.2.1
- deps: on-finished2.2.1
  - Fix isFinished(req) when data buffered
- deps: proxy-addr1.0.8

  - deps: ipaddr.js@1.0.1
- deps: qs@2.4.2
  - Fix allowing parameters like constructor
- deps: send@0.12.3
  - deps: debug2.2.0
  - deps: depd1.0.1
  - deps: etag1.6.0
  - deps: ms@0.7.1
  - deps: on-finished2.2.1
- deps: serve-static1.9.3
  - deps: send@0.12.3
- deps: type-is1.6.2
  - deps: mime-types2.0.11

67.32 4.12.3 / 2015-03-17

- deps: accepts1.2.5
  - deps: mime-types2.0.10
- deps: debug2.1.3
  - Fix high intensity foreground color for bold
  - deps: ms@0.7.0
- deps: finalhandler@0.3.4
  - deps: debug2.1.3
- deps: proxy-addr1.0.7
  - deps: ipaddr.js@0.1.9
- deps: qs@2.4.1
  - Fix error when parameter hasOwnProperty is present
- deps: send@0.12.2
  - Throw errors early for invalid extensions or index options
  - deps: debug2.1.3
- deps: serve-static1.9.2
  - deps: send@0.12.2
- deps: type-is1.6.1
  - deps: mime-types2.0.10

67.33 4.12.2 / 2015-03-02

- Fix regression where "Request aborted" is logged using res.sendFile

67.34 4.12.1 / 2015-03-01

- Fix constructing application with non-configurable prototype properties
- Fix ECONNRESET errors from res.sendFile usage
- Fix req.host when using "trust proxy" hops count
- Fix req.protocol/req.secure when using "trust proxy" hops count
- Fix wrong code on aborted connections from res.sendFile
- deps: merge-descriptors@1.0.0

67.35 4.12.0 / 2015-02-23

- Fix "trust proxy" setting to inherit when app is mounted
- Generate ETags for all request responses
  - No longer restricted to only responses for GET and HEAD requests
- Use content-type to parse Content-Type headers
- deps: accepts1.2.4
  - Fix preference sorting to be stable for long acceptable lists
  - deps: mime-types2.0.9
  - deps: negotiator@0.5.1
- deps: cookie-signature@1.0.6
- deps: send@0.12.1
  - Always read the stat size from the file
  - Fix mutating passed-in options
  - deps: mime@1.3.4
- deps: serve-static1.9.1
  - deps: send@0.12.1
- deps: type-is1.6.0
  - fix argument reassignment
  - fix false-positives in hasBody Transfer-Encoding check
  - support wildcard for both type and subtype (*/*)
  - deps: mime-types2.0.9

67.36 4.11.2 / 2015-02-01

- Fix res.redirect double-calling res.end for HEAD requests
- deps: accepts1.2.3
  - deps: mime-types2.0.8
- deps: proxy-addr1.0.6
  - deps: ipaddr.js@0.1.8
- deps: type-is1.5.6
  - deps: mime-types2.0.8

67.37 4.11.1 / 2015-01-20

- deps: send@0.11.1
  - Fix root path disclosure
- deps: serve-static1.8.1
  - Fix redirect loop in Node.js 0.11.14
  - Fix root path disclosure
  - deps: send@0.11.1

67.38 4.11.0 / 2015-01-13

- Add res.append(field, val) to append headers
- Deprecate leading : in name for app.param(name, fn)
- Deprecate req.param() - use req.params, req.body, or req.query instead
- Deprecate app.param(fn)
- Fix OPTIONS responses to include the HEAD method properly
- Fix res.sendFile not always detecting aborted connection
- Match routes iteratively to prevent stack overflows
- deps: accepts1.2.2
  - deps: mime-types2.0.7
  - deps: negotiator@0.5.0
- deps: send@0.11.0
  - deps: debug2.1.1
  - deps: etag1.5.1
  - deps: ms@0.7.0
  - deps: on-finished2.2.0
- deps: serve-static1.8.0
  - deps: send@0.11.0

67.39 4.10.8 / 2015-01-13

- Fix crash from error within OPTIONS response handler
- deps: proxy-addr1.0.5
  - deps: ipaddr.js@0.1.6

67.40 4.10.7 / 2015-01-04

- Fix Allow header for OPTIONS to not contain duplicate methods
- Fix incorrect "Request aborted" for res.sendFile when HEAD or 304
- deps: debug2.1.1
- deps: finalhandler@0.3.3
  - deps: debug2.1.1
  - deps: on-finished2.2.0
- deps: methods1.1.1
- deps: on-finished2.2.0
- deps: serve-static1.7.2
  - Fix potential open redirect when mounted at root
- deps: type-is1.5.5
  - deps: mime-types2.0.7

67.41 4.10.6 / 2014-12-12

- Fix exception in req.fresh/req.stale without response headers

67.42 4.10.5 / 2014-12-10

- Fix res.send double-calling res.end for HEAD requests
- deps: accepts1.1.4
  - deps: mime-types2.0.4
- deps: type-is1.5.4
  - deps: mime-types2.0.4

67.43 4.10.4 / 2014-11-24

- Fix res.sendfile logging standard write errors

67.44 4.10.3 / 2014-11-23

- Fix res.sendFile logging standard write errors
- deps: etag1.5.1
- deps: proxy-addr1.0.4
  - deps: ipaddr.js@0.1.5
- deps: qs@2.3.3
  - Fix arrayLimit behavior

67.45 4.10.2 / 2014-11-09

- Correctly invoke async router callback asynchronously
- deps: accepts1.1.3
  - deps: mime-types2.0.3
- deps: type-is1.5.3
  - deps: mime-types2.0.3

67.46 4.10.1 / 2014-10-28

- Fix handling of URLs containing :// in the path
- deps: qs@2.3.2
  - Fix parsing of mixed objects and values

67.47 4.10.0 / 2014-10-23

- Add support for ‘app.set('views’, array)`
  - Views are looked up in sequence in array of directories
- Fix res.send(status) to mention res.sendStatus(status)
- Fix handling of invalid empty URLs
- Use content-disposition module for res.attachment/res.download
  - Sends standards-compliant Content-Disposition header
  - Full Unicode support
- Use path.resolve in view lookup
- deps: debug2.1.0
  - Implement DEBUG_FD env variable support
- deps: depd1.0.0
- deps: etag1.5.0
  - Improve string performance
  - Slightly improve speed for weak ETags over 1KB
- deps: finalhandler@0.3.2
  - Terminate in progress response only on error
  - Use on-finished to determine request status
  - deps: debug2.1.0
  - deps: on-finished2.1.1
- deps: on-finished2.1.1
  - Fix handling of pipelined requests
- deps: qs@2.3.0
  - Fix parsing of mixed implicit and explicit arrays
- deps: send@0.10.1

  - deps: debug2.1.0
  - deps: depd1.0.0
  - deps: etag1.5.0
  - deps: on-finished2.1.1
- deps: serve-static1.7.1
  - deps: send@0.10.1

67.48 4.9.8 / 2014-10-17

- Fix res.redirect body when redirect status specified
- deps: accepts1.1.2
  - Fix error when media type has invalid parameter
  - deps: negotiator@0.4.9

67.49 4.9.7 / 2014-10-10

- Fix using same param name in array of paths

67.50 4.9.6 / 2014-10-08

- deps: accepts1.1.1
  - deps: mime-types2.0.2
  - deps: negotiator@0.4.8
- deps: serve-static1.6.4
  - Fix redirect loop when index file serving disabled
- deps: type-is1.5.2
  - deps: mime-types2.0.2

67.51 4.9.5 / 2014-09-24

- deps: etag1.4.0
- deps: proxy-addr1.0.3
  - Use forwarded npm module
- deps: send@0.9.3
  - deps: etag1.4.0
- deps: serve-static1.6.3
  - deps: send@0.9.3

67.52 4.9.4 / 2014-09-19

- deps: qs@2.2.4
  - Fix issue with object keys starting with numbers truncated

67.53 4.9.3 / 2014-09-18

- deps: proxy-addr1.0.2
  - Fix a global leak when multiple subnets are trusted
  - deps: ipaddr.js@0.1.3

67.54 4.9.2 / 2014-09-17

- Fix regression for empty string path in app.use
- Fix router.use to accept array of middleware without path
- Improve error message for bad app.use arguments

67.55 4.9.1 / 2014-09-16

- Fix app.use to accept array of middleware without path
- deps: depd@0.4.5
- deps: etag1.3.1
- deps: send@0.9.2
  - deps: depd@0.4.5
  - deps: etag1.3.1
  - deps: range-parser1.0.2
- deps: serve-static1.6.2
  - deps: send@0.9.2

67.56 4.9.0 / 2014-09-08

- Add res.sendStatus
- Invoke callback for sendfile when client aborts
  - Applies to res.sendFile, res.sendfile, and res.download
  - err will be populated with request aborted error
- Support IP address host in req.subdomains
- Use etag to generate ETag headers
- deps: accepts1.1.0
  - update mime-types
- deps: cookie-signature@1.0.5
- deps: debug2.0.0
- deps: finalhandler@0.2.0
  - Set X-Content-Type-Options: nosniff header
  - deps: debug2.0.0
- deps: fresh@0.2.4
- deps: media-typer@0.3.0

  - Throw error when parameter format invalid on parse
- deps: qs@2.2.3
  - Fix issue where first empty value in array is discarded
- deps: range-parser1.0.2
- deps: send@0.9.1
  - Add lastModified option
  - Use etag to generate ETag header
  - deps: debug2.0.0
  - deps: fresh@0.2.4
- deps: serve-static1.6.1
  - Add lastModified option
  - deps: send@0.9.1
- deps: type-is1.5.1
  - fix hasbody to be true for content-length: 0
  - deps: media-typer@0.3.0
  - deps: mime-types2.0.1
- deps: vary1.0.0
  - Accept valid Vary header string as field

67.57 4.8.8 / 2014-09-04

- deps: send@0.8.5
  - Fix a path traversal issue when using root
  - Fix malicious path detection for empty string path
- deps: serve-static1.5.4
  - deps: send@0.8.5

67.58 4.8.7 / 2014-08-29

- deps: qs@2.2.2
  - Remove unnecessary cloning

67.59 4.8.6 / 2014-08-27

- deps: qs@2.2.0
  - Array parsing fix
  - Performance improvements

67.60 4.8.5 / 2014-08-18

- deps: send@0.8.3
  - deps: destroy@1.0.3
  - deps: on-finished@2.1.0
- deps: serve-static1.5.3
  - deps: send@0.8.3

67.61 4.8.4 / 2014-08-14

- deps: qs@1.2.2
- deps: send@0.8.2
  - Work around fd leak in Node.js 0.10 for fs.ReadStream
- deps: serve-static1.5.2
  - deps: send@0.8.2

67.62 4.8.3 / 2014-08-10

- deps: parseurl1.3.0
- deps: qs@1.2.1
- deps: serve-static1.5.1
  - Fix parsing of weird req.originalUrl values
  - deps: parseurl1.3.0
  - deps: utils-merge@1.0.0

67.63 4.8.2 / 2014-08-07

- deps: qs@1.2.0
  - Fix parsing array of objects

67.64 4.8.1 / 2014-08-06

- fix incorrect deprecation warnings on res.download
- deps: qs@1.1.0
  - Accept urlencoded square brackets
  - Accept empty values in implicit array notation

67.65 4.8.0 / 2014-08-05

- add res.sendFile
  - accepts a file system path instead of a URL
  - requires an absolute path or root option specified
- deprecate res.sendfile - use res.sendFile instead

- support mounted app as any argument to app.use()
- deps: qs@1.0.2
  - Complete rewrite
  - Limits array length to 20
  - Limits object depth to 5
  - Limits parameters to 1,000
- deps: send@0.8.1
  - Add extensions option
- deps: serve-static1.5.0
  - Add extensions option
  - deps: send@0.8.1

67.66 4.7.4 / 2014-08-04

- fix res.sendfile regression for serving directory index files
- deps: send@0.7.4
  - Fix incorrect 403 on Windows and Node.js 0.11
  - Fix serving index files without root dir
- deps: serve-static1.4.4
  - deps: send@0.7.4

67.67 4.7.3 / 2014-08-04

- deps: send@0.7.3
  - Fix incorrect 403 on Windows and Node.js 0.11
- deps: serve-static1.4.3
  - Fix incorrect 403 on Windows and Node.js 0.11
  - deps: send@0.7.3

67.68 4.7.2 / 2014-07-27

- deps: depd@0.4.4
  - Work-around v8 generating empty stack traces
- deps: send@0.7.2
  - deps: depd@0.4.4
- deps: serve-static1.4.2

67.69 4.7.1 / 2014-07-26

- deps: depd@0.4.3
  - Fix exception when global Error.stackTraceLimit is too low
- deps: send@0.7.1
  - deps: depd@0.4.3
- deps: serve-static1.4.1

67.70 4.7.0 / 2014-07-25

- fix req.protocol for proxy-direct connections
- configurable query parser with ‘app.set('query parser’, parser)-app.↩ set('query parser', 'extended')parse with "qs" module -app.set('query parser', 'simple')parse with "querystring" core module -app.set('query parser', false)disable query string parsing -app.set('query parser', true)enable simple parsing
- deprecateres.json(status, obj)-- useres.status(status).json(obj)instead
- deprecateres.jsonp(status, obj)-- useres.status(status).jsonp(obj)instead
- deprecateres.send(status, body)-- useres.status(status).send(body)` instead
- deps: debug@1.0.4
- deps: depd@0.4.2
  - Add TRACE_DEPRECATION environment variable
  - Remove non-standard grey color from color output
  - Support --no-deprecation argument
  - Support --trace-deprecation argument
- deps: finalhandler@0.1.0
  - Respond after request fully read
  - deps: debug@1.0.4
- deps: parseurl1.2.0
  - Cache URLs based on original value
  - Remove no-longer-needed URL mis-parse work-around
  - Simplify the "fast-path" RegExp
- deps: send@0.7.0
  - Add dotfiles option
  - Cap maxAge value to 1 year
  - deps: debug@1.0.4
  - deps: depd@0.4.2
- deps: serve-static1.4.0
  - deps: parseurl1.2.0
  - deps: send@0.7.0
- perf: prevent multiple Buffer creation in res.send

67.71 4.6.1 / 2014-07-12

- fix subapp.mountpath regression for app.use(subapp)

67.72 4.6.0 / 2014-07-11

- accept multiple callbacks to app.use()
- add explicit "Rosetta Flash JSONP abuse" protection
  - previous versions are not vulnerable; this is just explicit protection
- catch errors in multiple req.param(name, fn) handlers
- deprecate res.redirect(url, status) - use res.redirect(status, url) instead
- fix res.send(status, num) to send num as json (not error)
- remove unnecessary escaping when res.jsonp returns JSON response
- support non-string path in app.use(path, fn)
  - supports array of paths
  - supports RegExp
- router: fix optimization on router exit
- router: refactor location of try blocks
- router: speed up standard app.use(fn)
- deps: debug@1.0.3
  - Add support for multiple wildcards in namespaces
- deps: finalhandler@0.0.3
  - deps: debug@1.0.3
- deps: methods@1.1.0
  - add CONNECT
- deps: parseurl1.1.3
  - faster parsing of href-only URLs
- deps: path-to-regexp@0.1.3
- deps: send@0.6.0
  - deps: debug@1.0.3
- deps: serve-static1.3.2
  - deps: parseurl1.1.3
  - deps: send@0.6.0
- perf: fix arguments reassign deopt in some res methods

67.73 4.5.1 / 2014-07-06

- fix routing regression when altering req.method

67.74 4.5.0 / 2014-07-04

- add deprecation message to non-plural req.accepts*
- add deprecation message to res.send(body, status)
- add deprecation message to res.vary()
- add headers option to res.sendfile
  - use to set headers on successful file transfer
- add mergeParams option to Router
  - merges req.params from parent routes
- add req.hostname - correct name for what req.host returns
- deprecate things with depd module
- deprecate req.host - use req.hostname instead
- fix behavior when handling request without routes
- fix handling when route.all is only route
- invoke router.param() only when route matches
- restore req.params after invoking router
- use finalhandler for final response handling
- use media-typer to alter content-type charset
- deps: accepts1.0.7
- deps: send@0.5.0
  - Accept string for maxage (converted by ms)
  - Include link in default redirect response
- deps: serve-static1.3.0
  - Accept string for maxAge (converted by ms)
  - Add setHeaders option
  - Include HTML link in redirect response
  - deps: send@0.5.0
- deps: type-is1.3.2

67.75 4.4.5 / 2014-06-26

- deps: cookie-signature@1.0.4
  - fix for timing attacks

67.76 4.4.4 / 2014-06-20

- fix res.attachment Unicode filenames in Safari
- fix "trim prefix" debug message in express:router
- deps: accepts1.0.5
- deps: buffer-crc32@0.2.3

67.77 4.4.3 / 2014-06-11

- fix persistence of modified req.params[name] from app.param()
- deps: accepts@1.0.3
  - deps: negotiator@0.4.6
- deps: debug@1.0.2
- deps: send@0.4.3
  - Do not throw uncatchable error on file open race condition
  - Use escape-html for HTML escaping
  - deps: debug@1.0.2
  - deps: finished@1.2.2
  - deps: fresh@0.2.2
- deps: serve-static@1.2.3
  - Do not throw uncatchable error on file open race condition
  - deps: send@0.4.3

67.78 4.4.2 / 2014-06-09

- fix catching errors from top-level handlers
- use vary module for res.vary
- deps: debug@1.0.1
- deps: proxy-addr@1.0.1
- deps: send@0.4.2
  - fix "event emitter leak" warnings
  - deps: debug@1.0.1
  - deps: finished@1.2.1
- deps: serve-static@1.2.2
  - fix "event emitter leak" warnings
  - deps: send@0.4.2
- deps: type-is@1.2.1

67.79 4.4.1 / 2014-06-02

- deps: methods@1.0.1
- deps: send@0.4.1
  - Send max-age in Cache-Control in correct format
- deps: serve-static@1.2.1
  - use escape-html for escaping
  - deps: send@0.4.1

67.80 4.4.0 / 2014-05-30

- custom etag control with ‘app.set('etag’, val)-app.set('etag', function(body, encoding){ return '"etag"' })custom etag generation -app.set('etag', 'weak')weak tag -app.set('etag', 'strong')strong etag -app.set('etag', false)turn off -app.set('etag', true)standard etag
- markres.send` ETag as weak and reduce collisions
- update accepts to 1.0.2
  - Fix interpretation when header not in request
- update send to 0.4.0
  - Calculate ETag with md5 for reduced collisions
  - Ignore stream errors after request ends
  - deps: debug@0.8.1
- update serve-static to 1.2.0
  - Calculate ETag with md5 for reduced collisions
  - Ignore stream errors after request ends
  - deps: send@0.4.0

67.81 4.3.2 / 2014-05-28

- fix handling of errors from router.param() callbacks

67.82 4.3.1 / 2014-05-23

- revert "fix behavior of multiple `app.VERB` for the same path"
  - this caused a regression in the order of route execution

67.83 4.3.0 / 2014-05-21

- add req.baseUrl to access the path stripped from req.url in routes
- fix behavior of multiple app.VERB for the same path
- fix issue routing requests among sub routers
- invoke router.param() only when necessary instead of every match
- proper proxy trust with ‘app.set('trust proxy’, trust)-app.set('trust proxy', 1)trust first hop -app.set('trust proxy', 'loopback')trust loopback addresses -app.set('trust proxy', '10.0.0.1')trust single IP -app.↩ set('trust proxy', '10.0.0.1/16')trust subnet -app.set('trust proxy', '10.0.0.1, 10.0.0.2')trust list -app.set('trust proxy', false)turn off -app.set('trust proxy', true)trust everything
- set propercharsetinContent-Typeforres.send`
- update type-is to 1.2.0
  - support suffix matching

67.84 4.2.0 / 2014-05-11

- deprecate `app.del()` - use `app.delete()` instead
- deprecate `res.json(obj, status)` - use `res.json(status, obj)` instead
    - the edge-case `res.json(status, num)` requires `res.status(status).json(num)`
- deprecate `res.jsonp(obj, status)` - use `res.jsonp(status, obj)` instead
    - the edge-case `res.jsonp(status, num)` requires `res.status(status).jsonp(num)`
- fix `req.next` when inside router instance
- include `ETag` header in `HEAD` requests
- keep previous `Content-Type` for `res.jsonp`
- support `PURGE` method
    - add `app.purge`
    - add `router.purge`
    - include `PURGE` in `app.all`
- update debug to 0.8.0
    - add `enable()` method
    - change from stderr to stdout
- update methods to 1.0.0
    - add `PURGE`

67.85 4.1.2 / 2014-05-08

- fix `req.host` for IPv6 literals
- fix `res.jsonp` error if callback param is object

67.86 4.1.1 / 2014-04-27

- fix package.json to reflect supported node version

67.87 4.1.0 / 2014-04-24

- pass options from `res.sendfile` to `send`
- preserve casing of headers in `res.header` and `res.set`
- support unicode file names in `res.attachment` and `res.download`
- update accepts to 1.0.1
    - deps: negotiator@0.4.0
- update cookie to 0.1.2
    - Fix for `maxAge == 0`
    - made compat with `expires` field
- update send to 0.3.0
    - Accept API options in options object

  - Coerce option types
  - Control whether to generate etags
  - Default directory access to 403 when index disabled
  - Fix sending files with dots without root set
  - Include file path in etag
  - Make "Can't set headers after they are sent." catchable
  - Send full entity-body for multi range requests
  - Set etags to "weak"
  - Support "If-Range" header
  - Support multiple index paths
  - deps: mime@1.2.11
- update serve-static to 1.1.0
  - Accept options directly to send module
  - Resolve relative paths at middleware setup
  - Use parseurl to parse the URL from request
  - deps: send@0.3.0
- update type-is to 1.1.0
  - add non-array values support
  - add multipart as a shorthand

67.88 4.0.0 / 2014-04-09

- remove:
  - node 0.8 support
  - connect and connect's patches except for charset handling
  - express(1) - moved to express-generator
  - express.createServer() - it has been deprecated for a long time. Use express()
  - app.configure - use logic in your own app code
  - app.router - is removed
  - req.auth - use basic-auth instead
  - req.accepted* - use req.accepts*() instead
  - res.location - relative URL resolution is removed
  - res.charset - include the charset in the content type when using res.set()
  - all bundled middleware except static
- change:
  - app.route -> app.mountpath when mounting an express app in another express app
  - json spaces no longer enabled by default in development
  - req.accepts* -> req.accepts*s - i.e. req.acceptsEncoding -> req.↩ acceptsEncodings
  - req.params is now an object instead of an array
  - res.locals is no longer a function. It is a plain js object. Treat it as such.

  - res.headerSent -> res.headersSent to match node.js ServerResponse object

- refactor:
  - req.accepts* with accepts
  - req.is with type-is
  - path-to-regexp
- add:
  - app.router() - returns the app Router instance
  - app.route() - Proxy to the app's Router#route() method to create a new route
  - Router & Route - public API

67.89 3.21.2 / 2015-07-31

- deps: connect@2.30.2
  - deps: body-parser1.13.3
  - deps: compression1.5.2
  - deps: errorhandler1.4.2
  - deps: method-override2.3.5
  - deps: serve-index1.7.2
  - deps: type-is1.6.6
  - deps: vhost3.0.1
- deps: vary1.0.1
  - Fix setting empty header from empty field
  - perf: enable strict mode
  - perf: remove argument reassignments

67.90 3.21.1 / 2015-07-05

- deps: basic-auth1.0.3
- deps: connect@2.30.1
  - deps: body-parser1.13.2
  - deps: compression1.5.1
  - deps: errorhandler1.4.1
  - deps: morgan1.6.1
  - deps: pause@0.1.0
  - deps: qs@4.0.0
  - deps: serve-index1.7.1
  - deps: type-is1.6.4

67.91 3.21.0 / 2015-06-18

- deps: basic-auth@1.0.2
  - perf: enable strict mode
  - perf: hoist regular expression
  - perf: parse with regular expressions
  - perf: remove argument reassignment
- deps: connect@2.30.0
  - deps: body-parser1.13.1
  - deps: bytes@2.1.0
  - deps: compression1.5.0
  - deps: cookie@0.1.3
  - deps: cookie-parser1.3.5
  - deps: csurf1.8.3
  - deps: errorhandler1.4.0
  - deps: express-session1.11.3
  - deps: finalhandler@0.4.0
  - deps: fresh@0.3.0
  - deps: morgan1.6.0
  - deps: serve-favicon2.3.0
  - deps: serve-index1.7.0
  - deps: serve-static1.10.0
  - deps: type-is1.6.3
- deps: cookie@0.1.3
  - perf: deduce the scope of try-catch deopt
  - perf: remove argument reassignments
- deps: escape-html@1.0.2
- deps: etag1.7.0
  - Always include entity length in ETags for hash length extensions
  - Generate non-Stats ETags using MD5 only (no longer CRC32)
  - Improve stat performance by removing hashing
  - Improve support for JXcore
  - Remove base64 padding in ETags to shorten
  - Support "fake" stats objects in environments without fs
  - Use MD5 instead of MD4 in weak ETags over 1KB
- deps: fresh@0.3.0
  - Add weak ETag matching support
- deps: mkdirp@0.5.1
  - Work in global strict mode
- deps: send@0.13.0
  - Allow Node.js HTTP server to set Date response header
  - Fix incorrectly removing Content-Location on 304 response

  - Improve the default redirect response headers
  - Send appropriate headers on default error response
  - Use http-errors for standard emitted errors
  - Use statuses instead of http module for status messages
  - deps: escape-html@1.0.2
  - deps: etag1.7.0
  - deps: fresh@0.3.0
  - deps: on-finished2.3.0
  - perf: enable strict mode
  - perf: remove unnecessary array allocations

67.92 3.20.3 / 2015-05-17

- deps: connect@2.29.2
  - deps: body-parser1.12.4
  - deps: compression1.4.4
  - deps: connect-timeout1.6.2
  - deps: debug2.2.0
  - deps: depd1.0.1
  - deps: errorhandler1.3.6
  - deps: finalhandler@0.3.6
  - deps: method-override2.3.3
  - deps: morgan1.5.3
  - deps: qs@2.4.2
  - deps: response-time2.3.1
  - deps: serve-favicon2.2.1
  - deps: serve-index1.6.4
  - deps: serve-static1.9.3
  - deps: type-is1.6.2
- deps: debug2.2.0
  - deps: ms@0.7.1
- deps: depd1.0.1
- deps: proxy-addr1.0.8
  - deps: ipaddr.js@1.0.1
- deps: send@0.12.3
  - deps: debug2.2.0
  - deps: depd1.0.1
  - deps: etag1.6.0
  - deps: ms@0.7.1
  - deps: on-finished2.2.1

67.93 3.20.2 / 2015-03-16

- deps: connect@2.29.1
  - deps: body-parser1.12.2
  - deps: compression1.4.3
  - deps: connect-timeout1.6.1
  - deps: debug2.1.3
  - deps: errorhandler1.3.5
  - deps: express-session1.10.4
  - deps: finalhandler@0.3.4
  - deps: method-override2.3.2
  - deps: morgan1.5.2
  - deps: qs@2.4.1
  - deps: serve-index1.6.3
  - deps: serve-static1.9.2
  - deps: type-is1.6.1
- deps: debug2.1.3
  - Fix high intensity foreground color for bold
  - deps: ms@0.7.0
- deps: merge-descriptors@1.0.0
- deps: proxy-addr1.0.7
  - deps: ipaddr.js@0.1.9
- deps: send@0.12.2
  - Throw errors early for invalid extensions or index options
  - deps: debug2.1.3

67.94 3.20.1 / 2015-02-28

- Fix req.host when using "trust proxy" hops count
- Fix req.protocol/req.secure when using "trust proxy" hops count

67.95 3.20.0 / 2015-02-18

- Fix "trust proxy" setting to inherit when app is mounted
- Generate ETags for all request responses
  - No longer restricted to only responses for GET and HEAD requests
- Use content-type to parse Content-Type headers
- deps: connect@2.29.0
  - Use content-type to parse Content-Type headers
  - deps: body-parser1.12.0
  - deps: compression1.4.1
  - deps: connect-timeout1.6.0

  - deps: cookie-parser1.3.4
  - deps: cookie-signature@1.0.6
  - deps: csurf1.7.0
  - deps: errorhandler1.3.4
  - deps: express-session1.10.3
  - deps: http-errors1.3.1
  - deps: response-time2.3.0
  - deps: serve-index1.6.2
  - deps: serve-static1.9.1
  - deps: type-is1.6.0
- deps: cookie-signature@1.0.6
- deps: send@0.12.1
  - Always read the stat size from the file
  - Fix mutating passed-in options
  - deps: mime@1.3.4

67.96 3.19.2 / 2015-02-01

- deps: connect@2.28.3
  - deps: compression1.3.1
  - deps: csurf1.6.6
  - deps: errorhandler1.3.3
  - deps: express-session1.10.2
  - deps: serve-index1.6.1
  - deps: type-is1.5.6
- deps: proxy-addr1.0.6
  - deps: ipaddr.js@0.1.8

67.97 3.19.1 / 2015-01-20

- deps: connect@2.28.2
  - deps: body-parser1.10.2
  - deps: serve-static1.8.1
- deps: send@0.11.1
  - Fix root path disclosure

67.98 3.19.0 / 2015-01-09

- Fix OPTIONS responses to include the HEAD method property
- Use readline for prompt in express(1)
- deps: commander@2.6.0
- deps: connect@2.28.1

  - deps: body-parser1.10.1
  - deps: compression1.3.0
  - deps: connect-timeout1.5.0
  - deps: csurf1.6.4
  - deps: debug2.1.1
  - deps: errorhandler1.3.2
  - deps: express-session1.10.1
  - deps: finalhandler@0.3.3
  - deps: method-override2.3.1
  - deps: morgan1.5.1
  - deps: serve-favicon2.2.0
  - deps: serve-index1.6.0
  - deps: serve-static1.8.0
  - deps: type-is1.5.5
- deps: debug2.1.1
- deps: methods1.1.1
- deps: proxy-addr1.0.5
  - deps: ipaddr.js@0.1.6
- deps: send@0.11.0
  - deps: debug2.1.1
  - deps: etag1.5.1
  - deps: ms@0.7.0
  - deps: on-finished2.2.0

67.99 3.18.6 / 2014-12-12

- Fix exception in req.fresh/req.stale without response headers

67.100 3.18.5 / 2014-12-11

- deps: connect@2.27.6
  - deps: compression1.2.2
  - deps: express-session1.9.3
  - deps: http-errors1.2.8
  - deps: serve-index1.5.3
  - deps: type-is1.5.4

67.101 3.18.4 / 2014-11-23

- deps: connect@2.27.4
  - deps: body-parser1.9.3
  - deps: compression1.2.1
  - deps: errorhandler1.2.3
  - deps: express-session1.9.2

  - deps: qs@2.3.3
  - deps: serve-favicon2.1.7
  - deps: serve-static1.5.1
  - deps: type-is1.5.3
- deps: etag1.5.1
- deps: proxy-addr1.0.4
  - deps: ipaddr.js@0.1.5

67.102 3.18.3 / 2014-11-09

- deps: connect@2.27.3
  - Correctly invoke async callback asynchronously
  - deps: csurf1.6.3

67.103 3.18.2 / 2014-10-28

- deps: connect@2.27.2
  - Fix handling of URLs containing :// in the path
  - deps: body-parser1.9.2
  - deps: qs@2.3.2

67.104 3.18.1 / 2014-10-22

- Fix internal utils.merge deprecation warnings
- deps: connect@2.27.1
  - deps: body-parser1.9.1
  - deps: express-session1.9.1
  - deps: finalhandler@0.3.2
  - deps: morgan1.4.1
  - deps: qs@2.3.0
  - deps: serve-static1.7.1
- deps: send@0.10.1
  - deps: on-finished2.1.1

67.105 3.18.0 / 2014-10-17

- Use content-disposition module for res.attachment/res.download
  - Sends standards-compliant Content-Disposition header
  - Full Unicode support
- Use etag module to generate ETag headers
- deps: connect@2.27.0
  - Use http-errors module for creating errors
  - Use utils-merge module for merging objects

  - deps: body-parser1.9.0
  - deps: compression1.2.0
  - deps: connect-timeout1.4.0
  - deps: debug2.1.0
  - deps: depd1.0.0
  - deps: express-session1.9.0
  - deps: finalhandler@0.3.1
  - deps: method-override2.3.0
  - deps: morgan1.4.0
  - deps: response-time2.2.0
  - deps: serve-favicon2.1.6
  - deps: serve-index1.5.0
  - deps: serve-static1.7.0
- deps: debug2.1.0
  - Implement DEBUG_FD env variable support
- deps: depd1.0.0
- deps: send@0.10.0
  - deps: debug2.1.0
  - deps: depd1.0.0
  - deps: etag1.5.0

67.106 3.17.8 / 2014-10-15

- deps: connect@2.26.6
  - deps: compression1.1.2
  - deps: csurf1.6.2
  - deps: errorhandler1.2.2

67.107 3.17.7 / 2014-10-08

- deps: connect@2.26.5
  - Fix accepting non-object arguments to logger
  - deps: serve-static1.6.4

67.108 3.17.6 / 2014-10-02

- deps: connect@2.26.4
  - deps: morgan1.3.2
  - deps: type-is1.5.2

67.109 3.17.5 / 2014-09-24

- deps: connect@2.26.3
  - deps: body-parser1.8.4
  - deps: serve-favicon2.1.5
  - deps: serve-static1.6.3
- deps: proxy-addr1.0.3
  - Use forwarded npm module
- deps: send@0.9.3
  - deps: etag1.4.0

67.110 3.17.4 / 2014-09-19

- deps: connect@2.26.2
  - deps: body-parser1.8.3
  - deps: qs@2.2.4

67.111 3.17.3 / 2014-09-18

- deps: proxy-addr1.0.2
  - Fix a global leak when multiple subnets are trusted
  - deps: ipaddr.js@0.1.3

67.112 3.17.2 / 2014-09-15

- Use crc instead of buffer-crc32 for speed
- deps: connect@2.26.1
  - deps: body-parser1.8.2
  - deps: depd@0.4.5
  - deps: express-session1.8.2
  - deps: morgan1.3.1
  - deps: serve-favicon2.1.3
  - deps: serve-static1.6.2
- deps: depd@0.4.5
- deps: send@0.9.2
  - deps: depd@0.4.5
  - deps: etag1.3.1
  - deps: range-parser1.0.2

67.113 3.17.1 / 2014-09-08

- Fix error in req.subdomains on empty host

67.114 3.17.0 / 2014-09-08

- Support X-Forwarded-Host in req.subdomains
- Support IP address host in req.subdomains
- deps: connect@2.26.0
  - deps: body-parser1.8.1
  - deps: compression1.1.0
  - deps: connect-timeout1.3.0
  - deps: cookie-parser1.3.3
  - deps: cookie-signature@1.0.5
  - deps: csurf1.6.1
  - deps: debug2.0.0
  - deps: errorhandler1.2.0
  - deps: express-session1.8.1
  - deps: finalhandler@0.2.0
  - deps: fresh@0.2.4
  - deps: media-typer@0.3.0
  - deps: method-override2.2.0
  - deps: morgan1.3.0
  - deps: qs@2.2.3
  - deps: serve-favicon2.1.3
  - deps: serve-index1.2.1
  - deps: serve-static1.6.1
  - deps: type-is1.5.1
  - deps: vhost3.0.0
- deps: cookie-signature@1.0.5
- deps: debug2.0.0
- deps: fresh@0.2.4
- deps: media-typer@0.3.0
  - Throw error when parameter format invalid on parse
- deps: range-parser1.0.2
- deps: send@0.9.1
  - Add lastModified option
  - Use etag to generate ETag header
  - deps: debug2.0.0
  - deps: fresh@0.2.4
- deps: vary1.0.0
  - Accept valid Vary header string as field

67.115 3.16.10 / 2014-09-04

- deps: connect@2.25.10
  - deps: serve-static1.5.4
- deps: send@0.8.5
  - Fix a path traversal issue when using root
  - Fix malicious path detection for empty string path

67.116 3.16.9 / 2014-08-29

- deps: connect@2.25.9
  - deps: body-parser1.6.7
  - deps: qs@2.2.2

67.117 3.16.8 / 2014-08-27

- deps: connect@2.25.8
  - deps: body-parser1.6.6
  - deps: csurf1.4.1
  - deps: qs@2.2.0

67.118 3.16.7 / 2014-08-18

- deps: connect@2.25.7
  - deps: body-parser1.6.5
  - deps: express-session1.7.6
  - deps: morgan1.2.3
  - deps: serve-static1.5.3
- deps: send@0.8.3
  - deps: destroy@1.0.3
  - deps: on-finished@2.1.0

67.119 3.16.6 / 2014-08-14

- deps: connect@2.25.6
  - deps: body-parser1.6.4
  - deps: qs@1.2.2
  - deps: serve-static1.5.2
- deps: send@0.8.2
  - Work around fd leak in Node.js 0.10 for fs.ReadStream

67.120 3.16.5 / 2014-08-11

- deps: connect@2.25.5
  - Fix backwards compatibility in logger

67.121 3.16.4 / 2014-08-10

- Fix original URL parsing in res.location
- deps: connect@2.25.4
  - Fix query middleware breaking with argument
  - deps: body-parser1.6.3
  - deps: compression1.0.11
  - deps: connect-timeout1.2.2
  - deps: express-session1.7.5
  - deps: method-override2.1.3
  - deps: on-headers1.0.0
  - deps: parseurl1.3.0
  - deps: qs@1.2.1
  - deps: response-time2.0.1
  - deps: serve-index1.1.6
  - deps: serve-static1.5.1
- deps: parseurl1.3.0

67.122 3.16.3 / 2014-08-07

- deps: connect@2.25.3
  - deps: multiparty@3.3.2

67.123 3.16.2 / 2014-08-07

- deps: connect@2.25.2
  - deps: body-parser1.6.2
  - deps: qs@1.2.0

67.124 3.16.1 / 2014-08-06

- deps: connect@2.25.1
  - deps: body-parser1.6.1
  - deps: qs@1.1.0

67.125 3.16.0 / 2014-08-05

- deps: connect@2.25.0
  - deps: body-parser1.6.0
  - deps: compression1.0.10
  - deps: csurf1.4.0
  - deps: express-session1.7.4
  - deps: qs@1.0.2
  - deps: serve-static1.5.0
- deps: send@0.8.1
  - Add extensions option

67.126 3.15.3 / 2014-08-04

- fix res.sendfile regression for serving directory index files
- deps: connect@2.24.3
  - deps: serve-index1.1.5
  - deps: serve-static1.4.4
- deps: send@0.7.4
  - Fix incorrect 403 on Windows and Node.js 0.11
  - Fix serving index files without root dir

67.127 3.15.2 / 2014-07-27

- deps: connect@2.24.2
  - deps: body-parser1.5.2
  - deps: depd@0.4.4
  - deps: express-session1.7.2
  - deps: morgan1.2.2
  - deps: serve-static1.4.2
- deps: depd@0.4.4
  - Work-around v8 generating empty stack traces
- deps: send@0.7.2
  - deps: depd@0.4.4

67.128 3.15.1 / 2014-07-26

- deps: connect@2.24.1
  - deps: body-parser1.5.1
  - deps: depd@0.4.3
  - deps: express-session1.7.1
  - deps: morgan1.2.1
  - deps: serve-index1.1.4
  - deps: serve-static1.4.1
- deps: depd@0.4.3
  - Fix exception when global Error.stackTraceLimit is too low
- deps: send@0.7.1
  - deps: depd@0.4.3

67.129 3.15.0 / 2014-07-22

- Fix req.protocol for proxy-direct connections
- Pass options from res.sendfile to send
- deps: connect@2.24.0
  - deps: body-parser1.5.0
  - deps: compression1.0.9
  - deps: connect-timeout1.2.1
  - deps: debug@1.0.4
  - deps: depd@0.4.2
  - deps: express-session1.7.0
  - deps: finalhandler@0.1.0
  - deps: method-override2.1.2
  - deps: morgan1.2.0
  - deps: multiparty@3.3.1
  - deps: parseurl1.2.0
  - deps: serve-static1.4.0
- deps: debug@1.0.4
- deps: depd@0.4.2
  - Add TRACE_DEPRECATION environment variable
  - Remove non-standard grey color from color output
  - Support --no-deprecation argument
  - Support --trace-deprecation argument
- deps: parseurl1.2.0
  - Cache URLs based on original value
  - Remove no-longer-needed URL mis-parse work-around
  - Simplify the "fast-path" RegExp
- deps: send@0.7.0
  - Add dotfiles option
  - Cap maxAge value to 1 year
  - deps: debug@1.0.4
  - deps: depd@0.4.2

67.130 3.14.0 / 2014-07-11

- add explicit "Rosetta Flash JSONP abuse" protection
  - previous versions are not vulnerable; this is just explicit protection
- deprecate res.redirect(url, status) - use res.redirect(status, url) instead
- fix res.send(status, num) to send num as json (not error)
- remove unnecessary escaping when res.jsonp returns JSON response
- deps: basic-auth@1.0.0

    - support empty password
    - support empty username
- deps: connect@2.23.0
    - deps: debug@1.0.3
    - deps: express-session1.6.4
    - deps: method-override2.1.0
    - deps: parseurl1.1.3
    - deps: serve-static1.3.1

deps: debug@1.0.3

- Add support for multiple wildcards in namespaces

deps: methods@1.1.0

- add CONNECT

deps: parseurl1.1.3

- faster parsing of href-only URLs

67.131 3.13.0 / 2014-07-03

- add deprecation message to app.configure
- add deprecation message to req.auth
- use basic-auth to parse Authorization header
- deps: connect@2.22.0
    - deps: csurf1.3.0
    - deps: express-session1.6.1
    - deps: multiparty@3.3.0
    - deps: serve-static1.3.0
- deps: send@0.5.0
    - Accept string for maxage (converted by ms)
    - Include link in default redirect response

67.132 3.12.1 / 2014-06-26

- deps: connect@2.21.1
    - deps: cookie-parser@1.3.2
    - deps: cookie-signature@1.0.4
    - deps: express-session1.5.2
    - deps: type-is1.3.2
- deps: cookie-signature@1.0.4
    - fix for timing attacks

67.133 3.12.0 / 2014-06-21

- `use media-typer to alter content-type charset`
- `deps: connect@2.21.0`
  - `deprecate connect(middleware) - use app.use(middleware) instead`
  - `deprecate connect.createServer() - use connect() instead`
  - `fix res.setHeader() patch to work with get -> append -> set pattern`
  - `deps: compression1.0.8`
  - `deps: errorhandler1.1.1`
  - `deps: express-session1.5.0`
  - `deps: serve-index1.1.3`

67.134 3.11.0 / 2014-06-19

- `deprecate things with depd module`
- `deps: buffer-crc32@0.2.3`
- `deps: connect@2.20.2`
  - `deprecate verify option to json - use body-parser npm module instead`
  - `deprecate verify option to urlencoded - use body-parser npm module instead`
  - `deprecate things with depd module`
  - `use finalhandler for final response handling`
  - `use media-typer to parse content-type for charset`
  - `deps: body-parser@1.4.3`
  - `deps: connect-timeout@1.1.1`
  - `deps: cookie-parser@1.3.1`
  - `deps: csurf@1.2.2`
  - `deps: errorhandler@1.1.0`
  - `deps: express-session@1.4.0`
  - `deps: multiparty@3.2.9`
  - `deps: serve-index@1.1.2`
  - `deps: type-is@1.3.1`
  - `deps: vhost@2.0.0`

67.135 3.10.5 / 2014-06-11

- `deps: connect@2.19.6`
  - `deps: body-parser@1.3.1`
  - `deps: compression@1.0.7`
  - `deps: debug@1.0.2`
  - `deps: serve-index@1.1.1`
  - `deps: serve-static@1.2.3`
- `deps: debug@1.0.2`
- `deps: send@0.4.3`

  - Do not throw uncatchable error on file open race condition
  - Use escape-html for HTML escaping
  - deps: debug@1.0.2
  - deps: finished@1.2.2
  - deps: fresh@0.2.2

67.136 3.10.4 / 2014-06-09

- deps: connect@2.19.5
  - fix "event emitter leak" warnings
  - deps: csurf@1.2.1
  - deps: debug@1.0.1
  - deps: serve-static@1.2.2
  - deps: type-is@1.2.1
- deps: debug@1.0.1
- deps: send@0.4.2
  - fix "event emitter leak" warnings
  - deps: finished@1.2.1
  - deps: debug@1.0.1

67.137 3.10.3 / 2014-06-05

- use vary module for res.vary
- deps: connect@2.19.4
  - deps: errorhandler@1.0.2
  - deps: method-override@2.0.2
  - deps: serve-favicon@2.0.1
- deps: debug@1.0.0

67.138 3.10.2 / 2014-06-03

- deps: connect@2.19.3
  - deps: compression@1.0.6

67.139 3.10.1 / 2014-06-03

- deps: connect@2.19.2
  - deps: compression@1.0.4
- deps: proxy-addr@1.0.1

67.140 3.10.0 / 2014-06-02

- deps: connect@2.19.1
  - deprecate methodOverride() - use method-override npm module instead
  - deps: body-parser@1.3.0
  - deps: method-override@2.0.1
  - deps: multiparty@3.2.8
  - deps: response-time@2.0.0
  - deps: serve-static@1.2.1
- deps: methods@1.0.1
- deps: send@0.4.1
  - Send max-age in Cache-Control in correct format

67.141 3.9.0 / 2014-05-30

- custom etag control with ‘app.set('etag’, val)-app.set('etag', function(body, encoding){ return '"etag"' })custom etag generation -app.set('etag', 'weak')weak tag -app.set('etag', 'strong')strong etag -app.set('etag', false)turn off -app.set('etag', true)` standard etag
- Include ETag in HEAD requests
- mark res.send ETag as weak and reduce collisions
- update connect to 2.18.0
  - deps: compression@1.0.3
  - deps: serve-index@1.1.0
  - deps: serve-static@1.2.0
- update send to 0.4.0
  - Calculate ETag with md5 for reduced collisions
  - Ignore stream errors after request ends
  - deps: debug@0.8.1

67.142 3.8.1 / 2014-05-27

- update connect to 2.17.3
  - deps: body-parser@1.2.2
  - deps: express-session@1.2.1
  - deps: method-override@1.0.2

67.143 3.8.0 / 2014-05-21

- keep previous Content-Type for res.jsonp
- set proper charset in Content-Type for res.send
- update connect to 2.17.1
  - fix res.charset appending charset when content-type has one
  - deps: express-session@1.2.0
  - deps: morgan@1.1.1
  - deps: serve-index@1.0.3

67.144 3.7.0 / 2014-05-18

- proper proxy trust with 'app.set('trust proxy', trust)-app.set('trust proxy', 1)trust first hop -app.set('trust proxy', 'loopback')trust loopback addresses -app.set('trust proxy', '10.0.0.1')trust single IP -app.↩ set('trust proxy', '10.0.0.1/16')trust subnet -app.set('trust proxy', '10.0.0.1, 10.0.0.2')trust list -app.set('trust proxy', false)turn off -app.set('trust proxy', true)` trust everything
- update connect to 2.16.2
  - deprecate res.headerSent - use res.headersSent
  - deprecate res.on("header") - use on-headers module instead
  - fix edge-case in res.appendHeader that would append in wrong order
  - json: use body-parser
  - urlencoded: use body-parser
  - dep: bytes@1.0.0
  - dep: cookie-parser@1.1.0
  - dep: csurf@1.2.0
  - dep: express-session@1.1.0
  - dep: method-override@1.0.1

67.145 3.6.0 / 2014-05-09

- deprecate app.del() - use app.delete() instead
- deprecate res.json(obj, status) - use res.json(status, obj) instead
  - the edge-case res.json(status, num) requires res.status(status).json(num)
- deprecate res.jsonp(obj, status) - use res.jsonp(status, obj) instead
  - the edge-case res.jsonp(status, num) requires res.status(status).jsonp(num)
- support PURGE method
  - add app.purge
  - add router.purge
  - include PURGE in app.all
- update connect to 2.15.0
  - Add res.appendHeader
  - Call error stack even when response has been sent
  - Patch res.headerSent to return Boolean
  - Patch res.headersSent for node.js 0.8
  - Prevent default 404 handler after response sent
  - dep: compression@1.0.2
  - dep: connect-timeout@1.1.0
  - dep: debug@^0.8.0
  - dep: errorhandler@1.0.1
  - dep: express-session@1.0.4
  - dep: morgan@1.0.1
  - dep: serve-favicon@2.0.0

    - dep: serve-index@1.0.2
- update debug to 0.8.0
    - add enable() method
    - change from stderr to stdout
- update methods to 1.0.0
    - add PURGE
- update mkdirp to 0.5.0

67.146 3.5.3 / 2014-05-08

- fix req.host for IPv6 literals
- fix res.jsonp error if callback param is object

67.147 3.5.2 / 2014-04-24

- update connect to 2.14.5
- update cookie to 0.1.2
- update mkdirp to 0.4.0
- update send to 0.3.0

67.148 3.5.1 / 2014-03-25

- pin less-middleware in generated app

67.149 3.5.0 / 2014-03-06

- bump deps

67.150 3.4.8 / 2014-01-13

- prevent incorrect automatic OPTIONS responses #1868 @dpatti
- update binary and examples for jade 1.0 #1876 @yossi, #1877 @reqshark, #1892 @matheusazzi
- throw 400 in case of malformed paths @rlidwka

67.151 3.4.7 / 2013-12-10

- update connect

67.152 3.4.6 / 2013-12-01

- update connect (raw-body)

67.153 3.4.5 / 2013-11-27

- update connect
- res.location: remove leading ./ #1802 @kapouer
- res.redirect: fix ‘res.redirect('toString’) #1829 @michaelficarra
- res.send: always send ETag when content-length > 0
- router: add Router.all() method

67.154 3.4.4 / 2013-10-29

- update connect
- update supertest
- update methods
- express(1): replace bodyParser() with urlencoded() and json() #1795 @chirag04

67.155 3.4.3 / 2013-10-23

- update connect

67.156 3.4.2 / 2013-10-18

- update connect
- downgrade commander

67.157 3.4.1 / 2013-10-15

- update connect
- update commander
- jsonp: check if callback is a function
- router: wrap encodeURIComponent in a try/catch #1735 (@lxe)
- res.format: now includes charset @1747 (@sorribas)
- res.links: allow multiple calls @1746 (@sorribas)

67.158 3.4.0 / 2013-09-07

- add res.vary(). Closes #1682
- update connect

67.159 3.3.8 / 2013-09-02

- update connect

67.160 3.3.7 / 2013-08-28

- update connect

67.161 3.3.6 / 2013-08-27

- Revert "remove charset from json responses. Closes #1631" (causes issues in some clients)
- add: req.accepts take an argument list

67.162 3.3.4 / 2013-07-08

- update send and connect

67.163 3.3.3 / 2013-07-04

- update connect

67.164 3.3.2 / 2013-07-03

- update connect
- update send
- remove .version export

67.165 3.3.1 / 2013-06-27

- update connect

67.166 3.3.0 / 2013-06-26

- update connect
- add support for multiple X-Forwarded-Proto values. Closes #1646
- change: remove charset from json responses. Closes #1631
- change: return actual booleans from req.accept* functions
- fix jsonp callback array throw

67.167 3.2.6 / 2013-06-02

- update connect

67.168 3.2.5 / 2013-05-21

- update connect
- update node-cookie
- add: throw a meaningful error when there is no default engine
- change generation of ETags with res.send() to GET requests only. Closes #1619

67.169 3.2.4 / 2013-05-09

- fix req.subdomains when no Host is present
- fix req.host when no Host is present, return undefined

67.170 3.2.3 / 2013-05-07

- update connect / qs

67.171 3.2.2 / 2013-05-03

- update qs

67.172 3.2.1 / 2013-04-29

- add app.VERB() paths array deprecation warning
- update connect
- update qs and remove all ~ semver crap
- fix: accept number as value of Signed Cookie

67.173 3.2.0 / 2013-04-15

- add "view" constructor setting to override view behaviour
- add req.acceptsEncoding(name)
- add req.acceptedEncodings
- revert cookie signature change causing session race conditions
- fix sorting of Accept values of the same quality

67.174 3.1.2 / 2013-04-12

- add support for custom Accept parameters
- update cookie-signature

67.175 3.1.1 / 2013-04-01

- add X-Forwarded-Host support to req.host
- fix relative redirects
- update mkdirp
- update buffer-crc32
- remove legacy app.configure() method from app template.

67.176 3.1.0 / 2013-01-25

- `add support for leading "." in "view engine" setting`
- `add array support to res.set()`
- `add node 0.8.x to travis.yml`
- `add "subdomain offset" setting for tweaking req.subdomains`
- `add res.location(url) implementing res.redirect()-like setting of Location`
- `use app.get() for x-powered-by setting for inheritance`
- `fix colons in passwords for req.auth`

67.177 3.0.6 / 2013-01-04

- `add http verb methods to Router`
- `update connect`
- `fix mangling of the res.cookie() options object`
- `fix jsonp whitespace escape.  Closes #1132`

67.178 3.0.5 / 2012-12-19

- `add throwing when a non-function is passed to a route`
- `fix:  explicitly remove Transfer-Encoding header from 204 and 304 responses`
- `revert "add 'etag' option"`

67.179 3.0.4 / 2012-12-05

- `add 'etag' option to disable res.send() Etags`
- `add escaping of urls in text/plain in res.redirect() for old browsers interpreting as html`
- `change crc32 module for a more liberal license`
- `update connect`

67.180 3.0.3 / 2012-11-13

- `update connect`
- `update cookie module`
- `fix cookie max-age`

67.181 3.0.2 / 2012-11-08

- `add OPTIONS to cors example.  Closes #1398`
- `fix route chaining regression.  Closes #1397`

67.182 3.0.1 / 2012-11-01

- update connect

67.183 3.0.0 / 2012-10-23

- add make clean
- add "Basic" check to req.auth
- add req.auth test coverage
- add cb && cb(payload) to res.jsonp(). Closes #1374
- add backwards compat for res.redirect() status. Closes #1336
- add support for res.json() to retain previously defined Content-Types. Closes #1349
- update connect
- change res.redirect() to utilize a pathname-relative Location again. Closes #1382
- remove non-primitive string support for res.send()
- fix view-locals example. Closes #1370
- fix route-separation example

67.184 3.0.0rc5 / 2012-09-18

- update connect
- add redis search example
- add static-files example
- add "x-powered-by" setting (‘app.disable('x-powered-by’)`)
- add "application/octet-stream" redirect Accept test case. Closes #1317

67.185 3.0.0rc4 / 2012-08-30

- add res.jsonp(). Closes #1307
- add "verbose errors" option to error-pages example
- add another route example to express(1) so people are not so confused
- add redis online user activity tracking example
- update connect dep
- fix etag quoting. Closes #1310
- fix error-pages 404 status
- fix jsonp callback char restrictions
- remove old OPTIONS default response

67.186 3.0.0rc3 / 2012-08-13

- update connect dep
- fix signed cookies to work with connect.cookieParser() ("s:" prefix was missing) [tnydwrds]
- fix res.render() clobbering of "locals"

67.187 3.0.0rc2 / 2012-08-03

- add CORS example
- update connect dep
- deprecate .createServer() & remove old stale examples
- fix: escape res.redirect() link
- fix vhost example

67.188 3.0.0rc1 / 2012-07-24

- add more examples to view-locals
- add scheme-relative redirects (res.redirect("//foo.com")) support
- update cookie dep
- update connect dep
- update send dep
- fix express(1) -h flag, use -H for hogan. Closes #1245
- fix res.sendfile() socket error handling regression

67.189 3.0.0beta7 / 2012-07-16

- update connect dep for send() root normalization regression

67.190 3.0.0beta6 / 2012-07-13

- add err.view property for view errors. Closes #1226
- add "jsonp callback name" setting
- add support for "/foo/:bar*" non-greedy matches
- change res.sendfile() to use send() module
- change res.send to use "response-send" module
- remove app.locals.use and res.locals.use, use regular middleware

67.191 3.0.0beta5 / 2012-07-03

- add "make check" support
- add route-map example
- add res.json(obj, status) support back for BC
- add "methods" dep, remove internal methods module
- update connect dep
- update auth example to utilize cores pbkdf2
- updated tests to use "supertest"

67.192 3.0.0beta4 / 2012-06-25

- Added req.auth
- Added req.range(size)
- Added res.links(obj)
- Added res.send(body, status) support back for backwards compat
- Added .default() support to res.format()
- Added 2xx / 304 check to req.fresh
- Revert "Added + support to the router"
- Fixed res.send() freshness check, respect res.statusCode

67.193 3.0.0beta3 / 2012-06-15

- Added hogan --hjs to express(1) [nullfirm]
- Added another example to content-negotiation
- Added fresh dep
- Changed: res.send() always checks freshness
- Fixed: expose connects mime module. Closes #1165

67.194 3.0.0beta2 / 2012-06-06

- Added + support to the router
- Added req.host
- Changed req.param() to check route first
- Update connect dep

67.195 3.0.0beta1 / 2012-06-01

- Added res.format() callback to override default 406 behaviour
- Fixed res.redirect() 406. Closes #1154

67.196 3.0.0alpha5 / 2012-05-30

- Added `req.ip`
- Added `{ signed:  true }` option to `res.cookie()`
- Removed `res.signedCookie()`
- Changed: dont reverse `req.ips`
- Fixed "trust proxy" setting check for `req.ips`

67.197 3.0.0alpha4 / 2012-05-09

- Added: allow `[]` in jsonp callback. Closes #1128
- Added PORT env var support in generated template. Closes #1118 [benatkin]
- Updated: connect 2.2.2

67.198 3.0.0alpha3 / 2012-05-04

- Added public `app.routes`. Closes #887
- Added *view-locals* example
- Added *mvc* example
- Added `res.locals.use()`. Closes #1120
- Added conditional-GET support to `res.send()`
- Added: coerce `res.set()` values to strings
- Changed: moved `static()` in generated apps below router
- Changed: `res.send()` only set ETag when not previously set
- Changed connect 2.2.1 dep
- Changed: make test now runs unit / acceptance tests
- Fixed req/res proto inheritance

67.199 3.0.0alpha2 / 2012-04-26

- Added make benchmark back
- Added `res.send()` support for String objects
- Added client-side data exposing example
- Added `res.header()` and `req.header()` aliases for BC
- Added `express.createServer()` for BC
- Perf: memoize parsed urls
- Perf: connect 2.2.0 dep
- Changed: make `expressInit()` middleware self-aware
- Fixed: use `app.get()` for all core settings
- Fixed redis session example
- Fixed session example. Closes #1105
- Fixed generated express dep. Closes #1078

67.200 3.0.0alpha1 / 2012-04-15

- Added app.locals.use(callback)
- Added app.locals object
- Added app.locals(obj)
- Added res.locals object
- Added res.locals(obj)
- Added res.format() for content-negotiation
- Added app.engine()
- Added res.cookie() JSON cookie support
- Added "trust proxy" setting
- Added req.subdomains
- Added req.protocol
- Added req.secure
- Added req.path
- Added req.ips
- Added req.fresh
- Added req.stale
- Added comma-delimited / array support for req.accepts()
- Added debug instrumentation
- Added res.set(obj)
- Added res.set(field, value)
- Added res.get(field)
- Added app.get(setting). Closes #842
- Added req.acceptsLanguage()
- Added req.acceptsCharset()
- Added req.accepted
- Added req.acceptedLanguages
- Added req.acceptedCharsets
- Added "json replacer" setting
- Added "json spaces" setting
- Added X-Forwarded-Proto support to res.redirect(). Closes #92
- Added --less support to express(1)
- Added express.response prototype
- Added express.request prototype
- Added express.application prototype
- Added app.path()

- Added app.render()
- Added res.type() to replace res.contentType()
- Changed: res.redirect() to add relative support
- Changed: enable "jsonp callback" by default
- Changed: renamed "case sensitive routes" to "case sensitive routing"
- Rewrite of all tests with mocha
- Removed "root" setting
- Removed 'res.redirect('home')support
- Removedreq.notify()
- Removedapp.register()
- Removedapp.redirect()
- Removedapp.is()
- Removedapp.helpers()
- Removedapp.dynamicHelpers()
- Fixedres.sendfile()` with non-GET. Closes #723
- Fixed express(1) public dir for windows. Closes #866

67.201 2.5.9/ 2012-04-02

- Added support for PURGE request method [pbuyle]
- Fixed express(1) generated app app.address() before listening [mmalecki]

67.202 2.5.8 / 2012-02-08

- Update mkdirp dep. Closes #991

67.203 2.5.7 / 2012-02-06

- Fixed app.all duplicate DELETE requests [mscdex]

67.204 2.5.6 / 2012-01-13

- Updated hamljs dev dep. Closes #953

67.205 2.5.5 / 2012-01-08

- Fixed: set filename on cached templates [matthewleon]

67.206 2.5.4 / 2012-01-02

- Fixed express(1) eol on 0.4.x. Closes #947

67.207 2.5.3 / 2011-12-30

- `Fixed req.is() when a charset is present`

67.208 2.5.2 / 2011-12-10

- `Fixed: express(1) LF -> CRLF for windows`

67.209 2.5.1 / 2011-11-17

- `Changed: updated connect to 1.8.x`
- `Removed sass.js support from express(1)`

67.210 2.5.0 / 2011-10-24

- `Added ./routes dir for generated app by default`
- `Added npm install reminder to express(1) app gen`
- `Added 0.5.x support`
- `Removed make test-cov since it wont work with node 0.5.x`
- `Fixed express(1) public dir for windows. Closes #866`

67.211 2.4.7 / 2011-10-05

- `Added mkdirp to express(1). Closes #795`
- Added simple *json-config* example
- `Added shorthand for the parsed request's pathname via req.path`
- `Changed connect dep to 1.7.x to fix npm issue...`
- Fixed res.redirect() **HEAD** support. [reported by xerox]
- `Fixed req.flash(), only escape args`
- `Fixed absolute path checking on windows. Closes #829 [reported by andrewpmckenzie]`

67.212 2.4.6 / 2011-08-22

- Fixed multiple param callback regression. Closes #824 [reported by TroyGoode]

67.213 2.4.5 / 2011-08-19

- `Added support for routes to handle errors. Closes #809`
- `Added app.routes.all(). Closes #803`
- Added "basepath" setting to work in conjunction with reverse proxies etc.
- `Refactored Route to use a single array of callbacks`

- Added support for multiple callbacks for app.param(). Closes #801 Closes #805
- Changed: removed .call(self) for route callbacks
- Dependency: qs >= 0.3.1
- Fixed res.redirect() on windows due to join() usage. Closes #808

67.214 2.4.4 / 2011-08-05

- Fixed res.header() intention of a set, even when undefined
- Fixed *, value no longer required
- Fixed res.send(204) support. Closes #771

67.215 2.4.3 / 2011-07-14

- Added docs for status option special-case. Closes #739
- Fixed options.filename, exposing the view path to template engines

67.216 2.4.2. / 2011-07-06

- Revert "removed jsonp stripping" for XSS

67.217 2.4.1 / 2011-07-06

- Added res.json() JSONP support. Closes #737
- Added *extending-templates* example. Closes #730
- Added "strict routing" setting for trailing slashes
- Added support for multiple envs in app.configure() calls. Closes #735
- Changed: res.send() using res.json()
- Changed: when cookie path === null don't default it
- Changed; default cookie path to "home" setting. Closes #731
- Removed *pids/logs* creation from express(1)

67.218 2.4.0 / 2011-06-28

- Added chainable res.status(code)
- Added res.json(), an explicit version of res.send(obj)
- Added simple web-service example

67.219 2.3.12 / 2011-06-22

- `#express is now on freenode! come join!`
- `Added req.get(field, param)`
- `Added links to Japanese documentation, thanks @hideyukisaito!`
- `Added; the express(1) generated app outputs the env`
- `Added content-negotiation example`
- `Dependency: connect >= 1.5.1 < 2.0.0`
- `Fixed view layout bug. Closes #720`
- `Fixed; ignore body on 304. Closes #701`

67.220 2.3.11 / 2011-06-04

- `Added npm test`
- `Removed generation of dummy test file from express(1)`
- `Fixed; express(1) adds express as a dep`
- `Fixed; prune on prepublish`

67.221 2.3.10 / 2011-05-27

- `Added req.route, exposing the current route`
- `Added` *package.json* `generation support to express(1)`
- `Fixed call to app.param() function for optional params. Closes #682`

67.222 2.3.9 / 2011-05-25

- `Fixed bug-ish with '../' in res.partial() calls`

67.223 2.3.8 / 2011-05-24

- `Fixed app.options()`

67.224 2.3.7 / 2011-05-23

- `Added route Collection, ex: 'app.get('/user/:id').remove();`
- `Added support forapp.param(fn)to define param logic`
- ``Removedapp.param()` support for callback with return value``
- `Removed module.parent check from express(1) generated app. Closes #670`
- `Refactored router. Closes #639`

67.225 2.3.6 / 2011-05-20

- Changed; using devDependencies instead of git submodules
- Fixed redis session example
- Fixed markdown example
- Fixed view caching, should not be enabled in development

67.226 2.3.5 / 2011-05-20

- Added export .view as alias for .View

67.227 2.3.4 / 2011-05-08

- Added ./examples/say
- Fixed res.sendfile() bug preventing the transfer of files with spaces

67.228 2.3.3 / 2011-05-03

- Added "case sensitive routes" option.
- Changed; split methods supported per rfc [slaskis]
- Fixed route-specific middleware when using the same callback function several times

67.229 2.3.2 / 2011-04-27

- Fixed view hints

67.230 2.3.1 / 2011-04-26

- Added app.match() as app.match.all()
- Added app.lookup() as app.lookup.all()
- Added app.remove() for app.remove.all()
- Added app.remove.VERB()
- Fixed template caching collision issue. Closes #644
- Moved router over from connect and started refactor

67.231 2.3.0 / 2011-04-25

- Added options support to res.clearCookie()
- Added res.helpers() as alias of res.locals()
- Added; json defaults to UTF-8 with res.send(). Closes #632. [Daniel * Dependency connect >= 1.4.0
- Changed; auto set Content-Type in res.attachement [Aaron Heckmann]
- Renamed "cache views" to "view cache". Closes #628

- Fixed caching of views when using several apps. Closes #637
- Fixed gotcha invoking app.param() callbacks once per route middleware. Closes #638
- Fixed partial lookup precedence. Closes #631 Shaw]

67.232 2.2.2 / 2011-04-12

- Added second callback support for res.download() connection errors
- Fixed filename option passing to template engine

67.233 2.2.1 / 2011-04-04

- Added layout(path) helper to change the layout within a view. Closes #610
- Fixed partial() collection object support. Previously only anything with .length would work. When .length is present one must still be aware of holes, however now '{ collection: {foo: 'bar'}}is valid, exposes keyInCollectionandkeysInCollection`.
- Performance improved with better view caching
- Removed request and response locals
- Changed; errorHandler page title is now Express instead of Connect

67.234 2.2.0 / 2011-03-30

- Added app.lookup.VERB(), ex 'app.lookup.put('/user/:id'). Closes #606
- Addedapp.match.VERB(), exapp.match.put('/user/12'). Closes #606
- Addedapp.VERB(path)as alias ofapp.lookup.VERB().
- Dependencyconnect >= 1.2.0`

67.235 2.1.1 / 2011-03-29

- Added; expose err.view object when failing to locate a view
- Fixed res.partial() call next(err) when no callback is given [reported by aheckmann]
- Fixed; res.send(undefined) responds with 204 [aheckmann]

67.236 2.1.0 / 2011-03-24

- Added <root>/_?<name> partial lookup support. Closes #447
- Added request, response, and app local variables
- Added settings local variable, containing the app's settings
- Added req.flash() exception if req.session is not available
- Added res.send(bool) support (json response)
- Fixed stylus example for latest version
- Fixed; wrap try/catch around res.render()

67.237 2.0.0 / 2011-03-17

- Fixed up index view path alternative.
- Changed; res.locals() without object returns the locals

67.238 2.0.0rc3 / 2011-03-17

- Added res.locals(obj) to compliment res.local(key, val)
- Added res.partial() callback support
- Fixed recursive error reporting issue in res.render()

67.239 2.0.0rc2 / 2011-03-17

- Changed; partial() "locals" are now optional
- Fixed SlowBuffer support. Closes #584 [reported by tyrda01]
- Fixed .filename view engine option [reported by drudge]
- Fixed blog example
- Fixed {req,res}.app reference when mounting [Ben Weaver]

67.240 2.0.0rc / 2011-03-14

- Fixed; expose HTTPSServer constructor
- Fixed express(1) default test charset. Closes #579 [reported by secoif]
- Fixed; default charset to utf-8 instead of utf8 for lame IE [reported by NickP]

67.241 2.0.0beta3 / 2011-03-09

- Added support for res.contentType() literal The original ‘res.content↩ Type('.json'), res.contentType('application/json'), andres.content↩ Type('json')` will work now.
- Added res.render() status option support back
- Added charset option for res.render()
- Added .charset support (via connect 1.0.4)
- Added view resolution hints when in development and a lookup fails
- Added layout lookup support relative to the page view. For example while rendering ./views/user/index.jade if you create ./views/user/layout.jade it will be used in favour of the root layout.
- Fixed res.redirect(). RFC states absolute url [reported by unlink]
- Fixed; default res.send() string charset to utf8
- Removed Partial constructor (not currently used)

67.242 2.0.0beta2 / 2011-03-07

- Added res.render() .locals support back to aid in migration process
- Fixed flash example

67.243 2.0.0beta / 2011-03-03

- Added HTTPS support
- Added res.cookie() maxAge support
- Added req.header() *Referrer* / *Referer* special-case, either works
- Added mount support for res.redirect(), now respects the mount-point
- Added union() util, taking place of merge(clone()) combo
- Added stylus support to express(1) generated app
- Added secret to session middleware used in examples and generated app
- Added res.local(name, val) for progressive view locals
- Added default param support to req.param(name, default)
- Added app.disabled() and app.enabled()
- Added app.register() support for omitting leading ".", either works
- Added res.partial(), using the same interface as partial() within a view. Closes #539
- Added app.param() to map route params to async/sync logic
- Added; aliased app.helpers() as app.locals(). Closes #481
- Added extname with no leading "." support to res.contentType()
- Added cache views setting, defaulting to enabled in "production" env
- Added index file partial resolution, eg: partial('user') may try *views/user/index.↵ jade*.
- Added req.accepts() support for extensions
- Changed; res.download() and res.sendfile() now utilize Connect's static file server connect.static.send().
- Changed; replaced connect.utils.mime() with npm *mime* module
- Changed; allow req.query to be pre-defined (via middleware or other parent
- Changed view partial resolution, now relative to parent view
- Changed view engine signature. no longer engine.render(str, options, callback), now engine.compile(str, options) -> Function, the returned function accepts fn(locals).
- Fixed req.param() bug returning Array.prototype methods. Closes #552
- Fixed; using Stream#pipe() instead of sys.pump() in res.sendfile()
- Fixed; using *qs* module instead of *querystring*
- Fixed; strip unsafe chars from jsonp callbacks
- Removed "stream threshold" setting

67.244 1.0.8 / 2011-03-01

- Allow req.query to be pre-defined (via middleware or other parent app)
- "connect": ">= 0.5.0 < 1.0.0". Closes #547
- Removed the long deprecated **EXPRESS_ENV** support

67.245 1.0.7 / 2011-02-07

- Fixed render() setting inheritance. Mounted apps would not inherit "view engine"

67.246 1.0.6 / 2011-02-07

- Fixed view engine setting bug when period is in dirname

67.247 1.0.5 / 2011-02-05

- Added secret to generated app session() call

67.248 1.0.4 / 2011-02-05

- Added qs dependency to *package.json*
- Fixed namespaced require()s for latest connect support

67.249 1.0.3 / 2011-01-13

- Remove unsafe characters from JSONP callback names [Ryan Grove]

67.250 1.0.2 / 2011-01-10

- Removed nested require, using connect.router

67.251 1.0.1 / 2010-12-29

- Fixed for middleware stacked via createServer() previously the foo middleware passed to createServer(foo) would not have access to Express methods such as res.send() or props like req.query etc.

67.252 1.0.0 / 2010-11-16

- Added; deduce partial object names from the last segment. For example by default 'partial('forum/post', postObject)` will give you the *post* object, providing a meaningful default.
- Added http status code string representation to res.redirect() body
- Added; res.redirect() supporting *text/plain* and *text/html* via **Accept.**
- Added req.is() to aid in content negotiation
- Added partial local inheritance [suggested by masylum]. Closes #102 providing access to parent template locals.

- Added *-s, -session[s]* flag to express(1) to add session related middleware
- Added *-template* flag to express(1) to specify the template engine to use.
- Added *-css* flag to express(1) to specify the stylesheet engine to use (or just plain css by default).
- Added app.all() support [thanks aheckmann]
- Added partial direct object support. You may now `partial('user', user)providing the "user" local, vs previouslypartial('user', { object: user })`.
- Added *route-separation* example since many people question ways to do this with CommonJS modules. Also view the *blog* example for an alternative.
- Performance; caching view path derived partial object names
- Fixed partial local inheritance precedence. [reported by Nick Poulden] Closes #454
- Fixed jsonp support; *text/javascript* as per mailinglist discussion

67.253 1.0.0rc4 / 2010-10-14

- Added *NODE_ENV* support, *EXPRESS_ENV* is deprecated and will be removed in 1.0.0
- Added route-middleware support (very helpful, see the docs)
- Added *jsonp callback* setting to enable/disable jsonp autowrapping [Dav Glass]
- Added callback query check on response.send to autowrap JSON objects for simple webservice implementations [Dav Glass]
- Added partial() support for array-like collections. Closes #434
- Added support for swappable querystring parsers
- Added session usage docs. Closes #443
- Added dynamic helper caching. Closes #439 [suggested by maritz]
- Added authentication example
- Added basic Range support to res.sendfile() (and res.download() etc)
- Changed; express(1) generated app using 2 spaces instead of 4
- Default env to "development" again [aheckmann]
- Removed *context* option is no more, use "scope"
- Fixed; exposing _./support_ libs to examples so they can run without installs
- Fixed mvc example

67.254 1.0.0rc3 / 2010-09-20

- Added confirmation for express(1) app generation. Closes #391
- Added extending of flash formatters via app.flashFormatters
- Added flash formatter support. Closes #411
- Added streaming support to res.sendfile() using sys.pump() when >= "stream threshold"
- Added *stream threshold* setting for res.sendfile()
- Added res.send() **HEAD** support
- Added res.clearCookie()
- Added res.cookie()
- Added res.render() headers option
- Added res.redirect() response bodies
- Added res.render() status option support. Closes #425 [thanks aheckmann]
- Fixed res.sendfile() responding with 403 on malicious path
- Fixed res.download() bug; when an error occurs remove *Content-Disposition*
- Fixed; mounted apps settings now inherit from parent app [aheckmann]
- Fixed; stripping Content-Length / Content-Type when 204
- Fixed res.send() 204. Closes #419
- Fixed multiple *Set-Cookie* headers via res.header(). Closes #402
- Fixed bug messing with error handlers when listenFD() is called instead of listen(). [thanks guillermo]

67.255 1.0.0rc2 / 2010-08-17

- Added app.register() for template engine mapping. Closes #390
- Added res.render() callback support as second argument (no options)
- Added callback support to res.download()
- Added callback support for res.sendfile()
- Added support for middleware access via express.middlewareName() vs connect.middlewareName()
- Added "partials" setting to docs
- Added default expresso tests to express(1) generated app. Closes #384
- Fixed res.sendfile() error handling, defer via next()
- Fixed res.render() callback when a layout is used [thanks guillermo]
- Fixed; make install creating ~/.node_libraries when not present
- Fixed issue preventing error handlers from being defined anywhere. Closes #387

67.256 1.0.0rc / 2010-07-28

- Added mounted hook. Closes #369
- Added connect dependency to *package.json*
- Removed "reload views" setting and support code development env never caches, production always caches.
- Removed *param* in route callbacks, signature is now simply (req, res, next), previously (req, res, params, next). Use *req.params* for path captures, *req.query* for GET params.
- Fixed "home" setting
- Fixed middleware/router precedence issue. Closes #366
- Fixed; *configure()* callbacks called immediately. Closes #368

67.257 1.0.0beta2 / 2010-07-23

- Added more examples
- Added; exporting Server constructor
- Added Server#helpers() for view locals
- Added Server#dynamicHelpers() for dynamic view locals. Closes #349
- Added support for absolute view paths
- Added; *home* setting defaults to Server#route for mounted apps. Closes #363
- Added Guillermo Rauch to the contributor list
- Added support for "as" for non-collection partials. Closes #341
- Fixed *install.sh*, ensuring _~/.node_libraries_ exists. Closes #362 [thanks jf]
- Fixed res.render() exceptions, now passed to next() when no callback is given [thanks guillermo]
- Fixed instanceof Array checks, now Array.isArray()
- Fixed express(1) expansion of public dirs. Closes #348
- Fixed middleware precedence. Closes #345
- Fixed view watcher, now async [thanks aheckmann]

67.258 1.0.0beta / 2010-07-15

- Re-write
  - much faster
  - much lighter
  - Check ExpressJS.com for migration guide and updated docs

67.259 0.14.0 / 2010-06-15

- Utilize relative requires
- Added Static bufferSize option [aheckmann]
- Fixed caching of view and partial subdirectories [aheckmann]
- Fixed mime.type() comments now that ".ext" is not supported
- Updated haml submodule
- Updated class submodule
- Removed bin/express

67.260 0.13.0 / 2010-06-01

- Added node v0.1.97 compatibility
- Added support for deleting cookies via Request::cookie('key', null)
- Updated haml submodule
- Fixed not-found page, now using charset utf-8
- Fixed show-exceptions page, now using charset utf-8
- Fixed view support due to fs.readFile Buffers
- Changed; mime.type() no longer accepts ".type" due to node extname() changes

67.261 0.12.0 / 2010-05-22

- Added node v0.1.96 compatibility
- Added view helpers export which act as additional local variables
- Updated haml submodule
- Changed ETag; removed inode, modified time only
- Fixed LF to CRLF for setting multiple cookies
- Fixed cookie compilation; values are now urlencoded
- Fixed cookies parsing; accepts quoted values and url escaped cookies

67.262 0.11.0 / 2010-05-06

- Added support for layouts using different engines
  - this.render('page.html.haml', { layout: 'super-cool-layout.html.↩ ejs' })
  - this.render('page.html.haml', { layout: 'foo' }) // assumes 'foo.↩ html.haml'
  - this.render('page.html.haml', { layout: false }) // no layout
- Updated ext submodule
- Updated haml submodule
- Fixed EJS partial support by passing along the context. Issue #307

67.263 0.10.1 / 2010-05-03

- Fixed binary uploads.

67.264 0.10.0 / 2010-04-30

- Added charset support via Request::charset (automatically assigned to 'UTF-8' when respond()'s encoding is set to 'utf8' or 'utf-8').
- Added "encoding" option to Request::render(). Closes #299
- Added "dump exceptions" setting, which is enabled by default.
- Added simple ejs template engine support
- Added error response support for text/plain, application/json. Closes #297
- Added callback function param to Request::error()
- Added Request::sendHead()
- Added Request::stream()
- Added support for Request::respond(304, null) for empty response bodies
- Added ETag support to Request::sendfile()
- Added options to Request::sendfile(), passed to fs.createReadStream()
- Added filename arg to Request::download()
- Performance enhanced due to pre-reversing plugins so that plugins.↩ reverse() is not called on each request
- Performance enhanced by preventing several calls to toLowerCase() in Router::match()
- Changed; Request::sendfile() now streams
- Changed; Renamed Request::halt() to Request::respond(). Closes #289
- Changed; Using sys.inspect() instead of JSON.encode() for error output
- Changed; run() returns the http.Server instance. Closes #298
- Changed; Defaulting Server::host to null (INADDR_ANY)
- Changed; Logger "common" format scale of 0.4f
- Removed Logger "request" format
- Fixed; Catching ENOENT in view caching, preventing error when "views/partials" is not found
- Fixed several issues with http client
- Fixed Logger Content-Length output
- Fixed bug preventing Opera from retaining the generated session id. Closes #292

67.265 0.9.0 / 2010-04-14

- Added DSL level error() route support
- Added DSL level notFound() route support
- Added Request::error()
- Added Request::notFound()
- Added Request::render() callback function. Closes #258
- Added "max upload size" setting
- Added "magic" variables to collection partials (__index__, __length__, __isFirst__, __isLast__). Closes #254
- Added haml.js submodule; removed haml-js
- Added callback function support to Request::halt() as 3rd/4th arg
- Added preprocessing of route param wildcards using param(). Closes #251
- Added view partial support (with collections etc.)
- Fixed bug preventing falsey params (such as ?page=0). Closes #286
- Fixed setting of multiple cookies. Closes #199
- Changed; view naming convention is now NAME.TYPE.ENGINE (for example page.html.haml)
- Changed; session cookie is now httpOnly
- Changed; Request is no longer global
- Changed; Event is no longer global
- Changed; "sys" module is no longer global
- Changed; moved Request::download to Static plugin where it belongs
- Changed; Request instance created before body parsing. Closes #262
- Changed; Pre-caching views in memory when "cache view contents" is enabled. Closes #253
- Changed; Pre-caching view partials in memory when "cache view partials" is enabled
- Updated support to node -version 0.1.90
- Updated dependencies
- Removed set("session cookie") in favour of use(Session, { cookie: { ... }})
- Removed utils.mixin(); use Object::mergeDeep()

67.266 0.8.0 / 2010-03-19

- Added coffeescript example app. Closes #242
- Changed; cache api now async friendly. Closes #240
- Removed deprecated 'express/static' support. Use 'express/plugins/static'

67.267 0.7.6 / 2010-03-19

- Added Request::isXHR. Closes #229
- Added make install (for the executable)
- Added express executable for setting up simple app templates
- Added "GET /public/*" to Static plugin, defaulting to <root>/public
- Added Static plugin
- Fixed; Request::render() only calls cache.get() once
- Fixed; Namespacing View caches with "view:"
- Fixed; Namespacing Static caches with "static:"
- Fixed; Both example apps now use the Static plugin
- Fixed set("views"). Closes #239
- Fixed missing space for combined log format
- Deprecated Request::sendfile() and 'express/static'
- Removed Server::running

67.268 0.7.5 / 2010-03-16

- Added Request::flash() support without args, now returns all flashes
- Updated ext submodule

67.269 0.7.4 / 2010-03-16

- Fixed session reaper
- Changed; class.js replacing js-oo Class implementation (quite a bit faster, no browser cruft)

67.270 0.7.3 / 2010-03-16

- Added package.json
- Fixed requiring of haml / sass due to kiwi removal

67.271 0.7.2 / 2010-03-16

- Fixed GIT submodules (HAH!)

67.272 0.7.1 / 2010-03-16

- Changed; Express now using submodules again until a PM is adopted
- Changed; chat example using millisecond conversions from ext

67.273 0.7.0 / 2010-03-15

- Added Request::pass() support (finds the next matching route, or the given path)
- Added Logger plugin (default "common" format replaces CommonLogger)
- Removed Profiler plugin
- Removed CommonLogger plugin

67.274 0.6.0 / 2010-03-11

- Added seed.yml for kiwi package management support
- Added HTTP client query string support when method is GET. Closes #205
- Added support for arbitrary view engines. For example "foo.engine.↩ html" will now require('engine'), the exports from this module are cached after the first require().
- Added async plugin support
- Removed usage of RESTful route funcs as http client get() etc, use http.↩ get() and friends
- Removed custom exceptions

67.275 0.5.0 / 2010-03-10

- Added ext dependency (library of js extensions)
- Removed extname() / basename() utils. Use path module
- Removed toArray() util. Use arguments.values
- Removed escapeRegexp() util. Use RegExp.escape()
- Removed process.mixin() dependency. Use utils.mixin()
- Removed Collection
- Removed ElementCollection
- Shameless self promotion of ebook "Advanced JavaScript" (http://dev-mag.↩ com) ;)

67.276 0.4.0 / 2010-02-11

- Added flash() example to sample upload app
- Added high level restful http client module (express/http)
- Changed; RESTful route functions double as HTTP clients. Closes #69
- Changed; throwing error when routes are added at runtime
- Changed; defaulting render() context to the current Request. Closes #197
- Updated haml submodule

67.277 0.3.0 / 2010-02-11

- Updated haml / sass submodules. Closes #200
- Added flash message support. Closes #64
- Added accepts() now allows multiple args. fixes #117
- Added support for plugins to halt. Closes #189
- Added alternate layout support. Closes #119
- Removed Route::run(). Closes #188
- Fixed broken specs due to use(Cookie) missing

67.278 0.2.1 / 2010-02-05

- Added "plot" format option for Profiler (for gnuplot processing)
- Added request number to Profiler plugin
- Fixed binary encoding for multipart file uploads, was previously defaulting to UTF8
- Fixed issue with routes not firing when not files are present. Closes #184
- Fixed process.Promise -> events.Promise

67.279 0.2.0 / 2010-02-03

- Added parseParam() support for name[] etc. (allows for file inputs with "multiple" attr) Closes #180
- Added Both Cache and Session option "reapInterval" may be "reapEvery". Closes #174
- Added expiration support to cache api with reaper. Closes #133
- Added cache Store.Memory::reap()
- Added Cache; cache api now uses first class Cache instances
- Added abstract session Store. Closes #172
- Changed; cache Memory.Store::get() utilizing Collection
- Renamed MemoryStore -> Store.Memory
- Fixed use() of the same plugin several time will always use latest options. Closes #176

67.280 0.1.0 / 2010-02-03

- Changed; Hooks (before / after) pass request as arg as well as evaluated in their context
- Updated node support to 0.1.27 Closes #169
- Updated dirname(__filename) -> __dirname
- Updated libxmljs support to v0.2.0

- Added session support with memory store / reaping
- Added quick uid() helper
- Added multi-part upload support
- Added Sass.js support / submodule
- Added production env caching view contents and static files
- Added static file caching. Closes #136
- Added cache plugin with memory stores
- Added support to StaticFile so that it works with non-textual files.
- Removed dirname() helper
- Removed several globals (now their modules must be required)

67.281 0.0.2 / 2010-01-10

- Added view benchmarks; currently haml vs ejs
- Added Request::attachment() specs. Closes #116
- Added use of node's parseQuery() util. Closes #123
- Added make init for submodules
- Updated Haml
- Updated sample chat app to show messages on load
- Updated libxmljs parseString -> parseHtmlString
- Fixed make init to work with older versions of git
- Fixed specs can now run independent specs for those who can't build deps. Closes #127
- Fixed issues introduced by the node url module changes. Closes 126.
- Fixed two assertions failing due to Collection::keys() returning strings
- Fixed faulty Collection::toArray() spec due to keys() returning strings
- Fixed make test now builds libxmljs.node before testing

67.282 0.0.1 / 2010-01-03

- Initial release

Chapter 68

Readme

Fast, unopinionated, minimalist web framework for `Node.js`.
This project has a `Code of Conduct`.

68.0.1 Table of contents

- Installation
- Features
- Docs & Community
- Quick Start
- Running Tests
- Philosophy
- Examples
- Contributing to Express
- TC (Technical Committee)
- Triagers
- License

```
const express = require('express')
const app = express()
app.get('/', function (req, res) {
  res.send('Hello World')
})
app.listen(3000)
```

68.0.2 Installation

This is a `Node.js` module available through the `npm registry`.
Before installing, `download and install Node.js`. Node.js 0.10 or higher is required.
If this is a brand new project, make sure to create a `package.json` first with the `npm init command`.
Installation is done using the `npm install command`:

```
$ npm install express
```

Follow `our installing guide` for more information.

68.0.3 Features

- Robust routing
- Focus on high performance
- Super-high test coverage
- HTTP helpers (redirection, caching, etc)
- View system supporting 14+ template engines
- Content negotiation
- Executable for generating applications quickly

68.0.4 Docs & Community

- `Website and Documentation` - [`website repo`]
- `#express` on `Libera Chat` IRC
- `GitHub Organization` for Official Middleware & Modules
- Visit the `Wiki`
- `Google Group` for discussion
- `Gitter` for support and discussion

PROTIP Be sure to read `Migrating from 3.x to 4.x` as well as `New features in 4.x`.

68.0.5 Quick Start

The quickest way to get started with express is to utilize the executable `express(1)` to generate an application as shown below:
Install the executable. The executable's major version will match Express's:
```
$ npm install -g express-generator@4
```
Create the app:
```
$ express /tmp/foo && cd /tmp/foo
```
Install dependencies:
```
$ npm install
```
Start the server:
```
$ npm start
```
View the website at: `http://localhost:3000`

68.0.6 Philosophy

The Express philosophy is to provide small, robust tooling for HTTP servers, making it a great solution for single page applications, websites, hybrids, or public HTTP APIs.
Express does not force you to use any specific ORM or template engine. With support for over 14 template engines via `Consolidate.js`, you can quickly craft your perfect framework.

68.0.7 Examples

To view the examples, clone the Express repo and install the dependencies:
```
$ git clone https://github.com/expressjs/express.git --depth 1
$ cd express
$ npm install
```
Then run whichever example you want:
```
$ node examples/content-negotiation
```

68.0.8 Contributing

The Express.js project welcomes all constructive contributions. Contributions take many forms, from code for bug fixes and enhancements, to additions and fixes to documentation, additional tests, triaging incoming pull requests and issues, and more!
See the Contributing Guide for more technical details on contributing.

68.0.8.1 Security Issues

If you discover a security vulnerability in Express, please see Security Policies and Procedures.

68.0.8.2 Running Tests

To run the test suite, first install the dependencies, then run `npm test`:
```
$ npm install
$ npm test
```

68.0.9 People

The original author of Express is `TJ Holowaychuk`
`List of all contributors`

68.0.9.1 TC (Technical Committee)

- `UlisesGascon` - **Ulises Gascón** (he/him)
- `jonchurch` - **Jon Church**
- `wesleytodd` - **Wes Todd**
- `LinusU` - **Linus Unnebäck**
- `blakeembrey` - **Blake Embrey**
- `sheplu` - **Jean Burellier**
- `crandmck` - **Rand McKinney**
- `ctcpip` - **Chris de Almeida**

TC emeriti members

68.0.9.1.1 TC emeriti members

- `dougwilson` - **Douglas Wilson**
- `hacksparrow` - **Hage Yaapa**
- `jonathanong` - **jongleberry**
- `niftylettuce` - **niftylettuce**
- `troygoode` - **Troy Goode**

68.0.9.2 Triagers

- `aravindvnair99` - **Aravind Nair**
- `carpasse` - **Carlos Serrano**
- `CBID2` - **Christine Belzie**
- `enyoghasim` - **David Enyoghasim**
- `UlisesGascon` - **Ulises Gascón** (he/him)

- `mertcanaltin` - **Mert Can Altin**
- `0ss` - **Salah**
- `import-brain` - **Eric Cheng** (he/him)
- `3imed-jaberi` - **Imed Jaberi**
- `dakshkhetan` - **Daksh Khetan** (he/him)
- `lucasraziel` - **Lucas Soares Do Rego**
- `IamLizu` - **S M Mahmudul Hasan** (he/him)
- `Sushmeet` - **Sushmeet Sunger**

Triagers emeriti members

68.0.9.2.1 Emeritus Triagers

- `AuggieH` - **Auggie Hudak**
- `G-Rath` - **Gareth Jones**
- `MohammadXroid` - **Mohammad Ayashi**
- `NawafSwe` - **Nawaf Alsharqi**
- `NotMoni` - **Moni**
- `VigneshMurugan` - **Vignesh Murugan**
- `davidmashe` - **David Ashe**
- `digitaIfabric` - **David**
- `e-l-i-s-e` - **Elise Bonner**
- `fed135` - **Frederic Charette**
- `firmanJS` - **Firman Abdul Hakim**
- `getspooky` - **Yasser Ameur**
- `ghinks` - **Glenn**
- `ghousemohamed` - **Ghouse Mohamed**
- `gireeshpunathil` - **Gireesh Punathil**
- `jake32321` - **Jake Reed**
- `jonchurch` - **Jon Church**
- `lekanikotun` - **Troy Goode**
- `marsonya` - **Lekan Ikotun**
- `mastermatt` - **Matt R. Wilson**
- `maxakuru` - **Max Edell**
- `mlrawlings` - **Michael Rawlings**
- `rodion-arr` - **Rodion Abdurakhimov**
- `sheplu` - **Jean Burellier**
- `tarunyadav1` - **Tarun yadav**
- `tunniclm` - **Mike Tunnicliffe**

68.0.10 License

[MIT](LICENSE)

Chapter 69

v1.3.1 / 2024-09-11

- deps: encodeurl2.0.0

69.1 v1.3.0 / 2024-09-03

- ignore status message for HTTP/2 (#53)

69.2 v1.2.1 / 2024-09-02

- Gracefully handle when handling an error and socket is null

69.3 1.2.0 / 2022-03-22

- Remove set content headers that break response
- deps: on-finished@2.4.1
- deps: statuses@2.0.1
  - Rename `425 Unordered Collection` to standard `425 Too Early`

69.4 1.1.2 / 2019-05-09

- Set stricter `Content-Security-Policy` header
- deps: parseurl1.3.3
- deps: statuses1.5.0

69.5 1.1.1 / 2018-03-06

- Fix 404 output for bad / missing pathnames
- deps: encodeurl1.0.2
  - Fix encoding `%` as last character
- deps: statuses1.4.0

69.6 1.1.0 / 2017-09-24

- Use `res.headersSent` when available

69.7 1.0.6 / 2017-09-22

- deps: debug@2.6.9

69.8 1.0.5 / 2017-09-15

- deps: parseurl1.3.2
  - perf: reduce overhead for full URLs
  - perf: unroll the "fast-path" `RegExp`

69.9 1.0.4 / 2017-08-03

- deps: debug@2.6.8

69.10 1.0.3 / 2017-05-16

- deps: debug@2.6.7
  - deps: ms@2.0.0

69.11 1.0.2 / 2017-04-22

- deps: debug@2.6.4
  - deps: ms@0.7.3

69.12 1.0.1 / 2017-03-21

- Fix missing `</html>` in HTML document
- deps: debug@2.6.3
  - Fix: `DEBUG_MAX_ARRAY_LENGTH`

69.13 1.0.0 / 2017-02-15

- Fix exception when `err` cannot be converted to a string
- Fully URL-encode the pathname in the 404 message
- Only include the pathname in the 404 message
- Send complete HTML document
- Set ‘Content-Security-Policy: default-src 'self'` header
- deps: debug@2.6.1
  - Allow colors in workers
  - Deprecated `DEBUG_FD` environment variable set to `3` or higher
  - Fix error when running under React Native
  - Use same color for same namespace
  - deps: ms@0.7.2

69.14 0.5.1 / 2016-11-12

- Fix exception when `err.headers` is not an object
- deps: statuses1.3.1
- perf: hoist regular expressions
- perf: remove duplicate validation path

69.15 0.5.0 / 2016-06-15

- Change invalid or non-numeric status code to 500
- Overwrite status message to match set status code
- Prefer `err.statusCode` if `err.status` is invalid
- Set response headers from `err.headers` object
- Use `statuses` instead of `http` module for status messages
  - Includes all defined status messages

69.16 0.4.1 / 2015-12-02

- deps: escape-html1.0.3
  - perf: enable strict mode
  - perf: optimize string replacement
  - perf: use faster string coercion

69.17 0.4.0 / 2015-06-14

- Fix a false-positive when unpiping in Node.js 0.8
- Support `statusCode` property on `Error` objects
- Use `unpipe` module for unpiping requests
- deps: escape-html@1.0.2
- deps: on-finished2.3.0
  - Add defined behavior for HTTP `CONNECT` requests
  - Add defined behavior for HTTP `Upgrade` requests
  - deps: ee-first@1.1.1
- perf: enable strict mode
- perf: remove argument reassignment

69.18 0.3.6 / 2015-05-11

- deps: debug2.2.0
  - deps: ms@0.7.1

69.19 0.3.5 / 2015-04-22

- deps: on-finished2.2.1
  - Fix `isFinished(req)` when data buffered

69.20 0.3.4 / 2015-03-15

- deps: debug2.1.3
  - Fix high intensity foreground color for bold
  - deps: ms@0.7.0

69.21 0.3.3 / 2015-01-01

- deps: debug2.1.1
- deps: on-finished2.2.0

69.22 0.3.2 / 2014-10-22

- deps: on-finished2.1.1
  - Fix handling of pipelined requests

69.23 0.3.1 / 2014-10-16

- deps: debug2.1.0
  - Implement `DEBUG_FD` env variable support

69.24 0.3.0 / 2014-09-17

- Terminate in progress response only on error
- Use `on-finished` to determine request status

69.25 0.2.0 / 2014-09-03

- Set `X-Content-Type-Options: nosniff` header
- deps: debug2.0.0

69.26 0.1.0 / 2014-07-16

- Respond after request fully read
  - prevents hung responses and socket hang ups
- deps: debug@1.0.4

69.27 0.0.3 / 2014-07-11

- deps: debug@1.0.3
  - Add support for multiple wildcards in namespaces

69.28 0.0.2 / 2014-06-19

- Handle invalid status codes

69.29 0.0.1 / 2014-06-05

- deps: debug@1.0.2

69.30 0.0.0 / 2014-06-05

- Extracted from connect/express

Chapter 70

finalhandler

Node.js function to invoke as the final step to respond to HTTP request.

70.1 Installation

This is a Node.js module available through the npm registry. Installation is done using the npm install command:

```
$ npm install finalhandler
```

70.2 API

```
var finalhandler = require('finalhandler')
```

70.2.1 finalhandler(req, res, [options])

Returns function to be invoked as the final step for the given `req` and `res`. This function is to be invoked as `fn(err)`. If `err` is falsy, the handler will write out a 404 response to the `res`. If it is truthy, an error response will be written out to the `res` or `res` will be terminated if a response has already started.
When an error is written, the following information is added to the response:

- The `res.statusCode` is set from `err.status` (or `err.statusCode`). If this value is outside the 4xx or 5xx range, it will be set to 500.
- The `res.statusMessage` is set according to the status code.
- The body will be the HTML of the status code message if `env` is "production', `otherwise will be`err.stack`.`
- `Any headers specified in an`err.headers` object.

The final handler will also unpipe anything from `req` when it is invoked.

70.2.1.1 options.env

By default, the environment is determined by `NODE_ENV` variable, but it can be overridden by this option.

70.2.1.2 options.onerror

Provide a function to be called with the `err` when it exists. Can be used for writing errors to a central location without excessive function generation. Called as `onerror(err, req, res)`.

70.3 Examples

70.3.1 always 404

```
var finalhandler = require('finalhandler')
var http = require('http')
var server = http.createServer(function (req, res) {
  var done = finalhandler(req, res)
  done()
})
server.listen(3000)
```

70.3.2 perform simple action

```
var finalhandler = require('finalhandler')
var fs = require('fs')
var http = require('http')
var server = http.createServer(function (req, res) {
  var done = finalhandler(req, res)
  fs.readFile('index.html', function (err, buf) {
    if (err) return done(err)
    res.setHeader('Content-Type', 'text/html')
    res.end(buf)
  })
})
server.listen(3000)
```

70.3.3 use with middleware-style functions

```
var finalhandler = require('finalhandler')
var http = require('http')
var serveStatic = require('serve-static')
var serve = serveStatic('public')
var server = http.createServer(function (req, res) {
  var done = finalhandler(req, res)
  serve(req, res, done)
})
server.listen(3000)
```

70.3.4 keep log of all errors

```
var finalhandler = require('finalhandler')
var fs = require('fs')
var http = require('http')
var server = http.createServer(function (req, res) {
  var done = finalhandler(req, res, { onerror:  logerror })
  fs.readFile('index.html', function (err, buf) {
    if (err) return done(err)
    res.setHeader('Content-Type', 'text/html')
    res.end(buf)
  })
})
server.listen(3000)
function logerror (err) {
  console.error(err.stack || err.toString())
}
```

70.4 License

[MIT](LICENSE)

Chapter 71

Security Policies and Procedures

71.1 Reporting a Bug

The `finalhandler` team and community take all security bugs seriously. Thank you for improving the security of Express. We appreciate your efforts and responsible disclosure and will make every effort to acknowledge your contributions.

Report security bugs by emailing the current owner(s) of `finalhandler`. This information can be found in the npm registry using the command `npm owner ls finalhandler`. If unsure or unable to get the information from the above, open an issue in the `project issue tracker` asking for the current contact information.

To ensure the timely response to your report, please ensure that the entirety of the report is contained within the email body and not solely behind a web link or an attachment.

At least one owner will acknowledge your email within 48 hours, and will send a more detailed response within 48 hours indicating the next steps in handling your report. After the initial reply to your report, the owners will endeavor to keep you informed of the progress towards a fix and full announcement, and may ask for additional information or guidance.

Chapter 72

0.2.0 / 2021-05-31

- Use `req.socket` over deprecated `req.connection`

72.1 0.1.2 / 2017-09-14

- perf: improve header parsing
- perf: reduce overhead when no `X-Forwarded-For` header

72.2 0.1.1 / 2017-09-10

- Fix trimming leading / trailing OWS
- perf: hoist regular expression

72.3 0.1.0 / 2014-09-21

- Initial release

Chapter 73

forwarded

Parse HTTP X-Forwarded-For header

73.1 Installation

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install forwarded
```

73.2 API

```
var forwarded = require('forwarded')
```

73.2.1 forwarded(req)

```
var addresses = forwarded(req)
```

Parse the `X-Forwarded-For` header from the request. Returns an array of the addresses, including the socket address for the `req`, in reverse order (i.e. index `0` is the socket address and the last index is the furthest address, typically the end-user).

73.3 Testing

```
$ npm test
```

73.4 License

[MIT](LICENSE)

Chapter 74

0.5.2 / 2017-09-13

- Fix regression matching multiple ETags in `If-None-Match`
- perf: improve `If-None-Match` token parsing

74.1 0.5.1 / 2017-09-11

- Fix handling of modified headers with invalid dates
- perf: improve ETag match loop

74.2 0.5.0 / 2017-02-21

- Fix incorrect result when `If-None-Match` has both `*` and ETags
- Fix weak `ETag` matching to match spec
- perf: delay reading header values until needed
- perf: skip checking modified time if ETag check failed
- perf: skip parsing `If-None-Match` when no `ETag` header
- perf: use `Date.parse` instead of `new Date`

74.3 0.4.0 / 2017-02-05

- Fix false detection of `no-cache` request directive
- perf: enable strict mode
- perf: hoist regular expressions
- perf: remove duplicate conditional
- perf: remove unnecessary boolean coercions

74.4 0.3.0 / 2015-05-12

- Add weak `ETag` matching support

74.5 0.2.4 / 2014-09-07

- Support Node.js 0.6

74.6 0.2.3 / 2014-09-07

- Move repository to jshttp

74.7 0.2.2 / 2014-02-19

- Revert "Fix for blank page on Safari reload"

74.8 0.2.1 / 2014-01-29

- Fix for blank page on Safari reload

74.9 0.2.0 / 2013-08-11

- Return stale for `Cache-Control: no-cache`

74.10 0.1.0 / 2012-06-15

- Add `If-None-Match: *` support

74.11 0.0.1 / 2012-06-10

- Initial release

Chapter 75

fresh

HTTP response freshness testing

75.1 Installation

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install fresh
```

75.2 API

```
var fresh = require('fresh')
```

75.2.1 fresh(reqHeaders, resHeaders)

Check freshness of the response using request and response headers.
When the response is still "fresh" in the client's cache `true` is returned, otherwise `false` is returned to indicate that the client cache is now stale and the full response should be sent.
When a client sends the `Cache-Control: no-cache` request header to indicate an end-to-end reload request, this module will return `false` to make handling these requests transparent.

75.3 Known Issues

This module is designed to only follow the HTTP specifications, not to work-around all kinda of client bugs (especially since this module typically does not recieve enough information to understand what the client actually is).
There is a known issue that in certain versions of Safari, Safari will incorrectly make a request that allows this module to validate freshness of the resource even when Safari does not have a representation of the resource in the cache.
The module `jumanji` can be used in an Express application to work-around this issue and also provides links to further reading on this Safari bug.

75.4 Example

75.4.1 API usage

```
var reqHeaders = { 'if-none-match': '"foo"' }
var resHeaders = { 'etag': '"bar"' }
fresh(reqHeaders, resHeaders)
// => false
var reqHeaders = { 'if-none-match': '"foo"' }
var resHeaders = { 'etag': '"foo"' }
fresh(reqHeaders, resHeaders)
// => true
```

75.4.2 Using with Node.js http server

```
var fresh = require('fresh')
var http = require('http')
var server = http.createServer(function (req, res) {
  // perform server logic
  // ...  including adding ETag / Last-Modified response headers
  if (isFresh(req, res)) {
    // client has a fresh copy of resource
    res.statusCode = 304
    res.end()
    return
  }
  // send the resource
  res.statusCode = 200
  res.end('hello, world!')
})
function isFresh (req, res) {
  return fresh(req.headers, {
    'etag':  res.getHeader('ETag'),
    'last-modified':  res.getHeader('Last-Modified')
  })
}
server.listen(3000)
```

75.5 License

[MIT](LICENSE)

Chapter 76

fs.realpath

A backwards-compatible fs.realpath for Node v6 and above

In Node v6, the JavaScript implementation of fs.realpath was replaced with a faster (but less resilient) native implementation. That raises new and platform-specific errors and cannot handle long or excessively symlink-looping paths.

This module handles those cases by detecting the new errors and falling back to the JavaScript implementation. On versions of Node prior to v6, it has no effect.

76.1 USAGE

```
var rp = require('fs.realpath')
// async version
rp.realpath(someLongAndLoopingPath, function (er, real) {
  // the ELOOP was handled, but it was a bit slower
})
// sync version
var real = rp.realpathSync(someLongAndLoopingPath)
// monkeypatch at your own risk!
// This replaces the fs.realpath/fs.realpathSync builtins
rp.monkeypatch()
// un-do the monkeypatching
rp.unmonkeypatch()
```

Chapter 77

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

77.1 <a href="https://github.com/ljharb/function-bind/compare/v1.1.1...↩ v1.1.2" >v1.1.2</a> - 2023-10-12

77.1.1 Merged

- Point to the correct file `#16`

77.1.2 Commits

- [Tests] migrate tests to Github Actions `4f8b57c`
- [Tests] remove `jscs` `90eb2ed`
- [meta] update `.gitignore` `53fcdc3`
- [Tests] up to `node` `v11.10`, `v10.15`, `v9.11`, `v8.15`, `v6.16`, `v4.9`; use `nvm install-latest-npm`; run audit script in tests `1fe8f6e`
- [meta] add `auto-changelog` `1921fcb`
- [Robustness] remove runtime dependency on all builtins except `.apply` `f743e61`
- Docs: enable badges; update wording `503cb12`
- [readme] update badges `290c5db`
- [Tests] switch to nyc for coverage `ea360ba`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `tape` `cae5e9e`
- [meta] add `funding` field; create FUNDING.yml `c9f4274`
- [Tests] fix eslint errors from #15 `f69aaa2`
- [actions] fix permissions `99a0cd9`
- [meta] use `npmignore` to autogenerate an npmignore file `f03b524`
- [Dev Deps] update `@ljharb/eslint-config`, `eslint`, `tape` `7af9300`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `covert`, `tape` `64a9127`
- [Tests] use `aud` instead of `npm audit` `e75069c`
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `tape` `d03555c`

- [meta] add `safe-publish-latest` 9c8f809
- [Dev Deps] update `@ljharb/eslint-config`, `tape` baf6893
- [meta] create SECURITY.md 4db1779
- [Tests] add `npm run audit` c8b38ec
- Revert "Point to the correct file" 05cdf0f

77.2 <a href="https://github.com/ljharb/function-bind/compare/v1.1.0...↩ v1.1.1" >v1.1.1</a> - 2017-08-28

77.2.1 Commits

- [Tests] up to `node` `v8`; newer npm breaks on older node; fix scripts 817f7d2
- [Dev Deps] update `eslint`, `jscs`, `tape`, `@ljharb/eslint-config` 854288b
- [Dev Deps] update `tape`, `jscs`, `eslint`, `@ljharb/eslint-config` 83e639f
- Only apps should have lockfiles 5ed97f5
- Use a SPDX-compliant “license” field. 5feefea

77.3 <a href="https://github.com/ljharb/function-bind/compare/v1.0.2...↩ v1.1.0" >v1.1.0</a> - 2016-02-14

77.3.1 Commits

- Update `eslint`, `tape`; use my personal shared `eslint` config 9c9062a
- Add `npm run eslint` dd96c56
- [New] return the native `bind` when available. 82186e0
- [Dev Deps] update `tape`, `jscs`, `eslint`, `@ljharb/eslint-config` a3dd767
- Update `eslint` 3dae2f7
- Update `tape`, `covert`, `jscs` a181eee
- [Tests] up to `node` `v5.6`, `v4.3` 964929a
- Test up to `io.js` `v2.1` 2be7310
- Update `tape`, `jscs`, `eslint`, `@ljharb/eslint-config` 45f3d68
- [Dev Deps] update `tape`, `jscs` 6e1340d
- [Tests] up to `io.js` `v3.3`, `node` `v4.1` d9bad2b
- Update `eslint` 935590c
- [Dev Deps] update `jscs`, `eslint`, `@ljharb/eslint-config` 8c9a1ef
- Test on `io.js` `v2.2` 9a3a38c
- Run `travis-ci` tests on `iojs` and `node` v0.12; speed up builds; allow 0.8 failures. 69afc26
- [Dev Deps] Update `tape`, `eslint` 36c1be0
- Update `tape`, `jscs` 98d8303
- Update `jscs` 9633a4e

- Update `tape`, `jscs` `c80ef0f`
- Test up to `io.js` `v3.0` `7e2c853`
- Test on `io.js` `v2.4` `5a199a2`
- Test on `io.js` `v2.3` `a511b88`
- Fixing a typo from 822b4e1938db02dc9584aa434fd3a45cb20caf43 `732d6b6`
- Update `jscs` `da52a48`
- Lock covert to v1.0.0. `d6150fd`

77.4 <a href="https://github.com/ljharb/function-bind/compare/v1.0.1...v1.0.2" >v1.0.2</a> - 2014-10-04

77.5 <a href="https://github.com/ljharb/function-bind/compare/v1.0.0...v1.0.1" >v1.0.1</a> - 2014-10-03

77.5.1 Merged

- make CI build faster `#3`

77.5.2 Commits

- Using my standard jscs.json `d8ee94c`
- Adding `npm run lint` `7571ab7`
- Using consistent indentation `e91a1b1`
- Updating jscs `7e17892`
- Using consistent quotes `c50b57f`
- Adding keywords `cb94631`
- Directly export a function expression instead of using a declaration, and relying on hoisting. `5a33c5f`
- Naming npm URL and badge in README; use SVG `2aef8fc`
- Naming deps URLs in README `04228d7`
- Naming travis-ci URLs in README; using SVG `62c810c`
- Make sure functions are invoked correctly (also passing coverage tests) `2b289b4`
- Removing the strict mode pragmas; they make tests fail. `1aa701d`
- Adding myself as a contributor `85fd57b`
- Adding strict mode pragmas `915b08e`
- Adding devDeps URLs to README `4ccc731`
- Fixing the description. `a7a472c`
- Using a function expression instead of a function declaration. `b5d3e4e`
- Updating tape `f086be6`
- Updating jscs `5f9bdb3`
- Updating jscs `9b409ba`

- Run coverage as part of tests. `8e1b6d4`
- Run linter as part of tests `c1ca83f`
- Updating covert `701e837`

77.6 <a href="https://github.com/ljharb/function-bind/compare/v0.2.0...↩ v1.0.0" >v1.0.0</a> - 2014-08-09

77.6.1 Commits

- Make sure old and unstable nodes don't fail Travis `27adca3`
- Fixing an issue when the bound function is called as a constructor in ES3. `e20122d`
- Adding `npm run coverage` `a2e29c4`
- Updating tape `b741168`
- Upgrading tape `63631a0`
- Updating tape `363cb46`

77.7 v0.2.0 - 2014-03-23

77.7.1 Commits

- Updating test coverage to match es5-shim. `aa94d44`
- initial `942ee07`
- Setting the bound function's length properly. `079f46a`
- Ensuring that some older browsers will throw when given a regex. `36ac55b`
- Removing npm scripts that don't have dependencies `9d2be60`
- Updating tape `297a4ac`
- Skipping length tests for now. `d9891ea`
- don't take my tea `dccd930`

Chapter 78

function-bind <sup><a href="https://npmjs.org/package/function-bind" ><img src="https://versionbadg.es/↩ Raynos/function-bind.svg" alt="Version Badge"/></a></sup>

[][license-url]

Implementation of function.prototype.bind
Old versions of phantomjs, Internet Explorer < 9, and node < 0.6 don't support `Function.prototype.bind`.

78.1 Example

```
Function.prototype.bind = require("function-bind")
```

78.2 Installation

```
npm install function-bind
```

78.3 Contributors

- Raynos

78.4 MIT Licenced

Chapter 79

generate-function

Module that helps you write generated functions in Node

```
npm install generate-function
```

79.1 Disclamer

Writing code that generates code is hard. You should only use this if you really, really, really need this for performance reasons (like schema validators / parsers etc).

79.2 Usage

```
const genfun = require('generate-function')
const { d } = genfun.formats
function addNumber (val) {
  const gen = genfun()
  gen(`
    function add (n) {')
      return n + ${d(val)}) // supports format strings to insert values
    }
  `)
  return gen.toFunction() // will compile the function
}
const add2 = addNumber(2)
console.log('1 + 2 =', add2(1))
console.log(add2.toString()) // prints the generated function
```

If you need to close over variables in your generated function pass them to `toFunction(scope)`

```
function multiply (a, b) {
  return a * b
}
function addAndMultiplyNumber (val) {
  const gen = genfun()

  gen(`
    function (n) {
      if (typeof n !== 'number') {
        throw new Error('argument should be a number')
      }
      const result = multiply(${d(val)}, n + ${d(val)})
      return result
    }
  `)
  // use gen.toString() if you want to see the generated source
  return gen.toFunction({multiply})
}
const addAndMultiply2 = addAndMultiplyNumber(2)
console.log(addAndMultiply2.toString())
console.log('(3 + 2) * 2 =', addAndMultiply2(3))
```

You can call `gen(src)` as many times as you want to append more source code to the function.

79.3 Variables

If you need a unique safe identifier for the scope of the generated function call 'str = gen.sym('friendlyName')`. These are safe to use for variable names etc.

79.4 Object properties

If you need to access an object property use the ‘str = gen.property('objectName’, 'propertyName')`.
This returns ‘'objectName.propertyName’`if`propertyName`is safe to use as a variable. Otherwise it returns`objectName[propertyNameAsString]`.
If you only pass ‘gen.property('propertyName’)`it will only return the`propertyName` part safely

79.5 License

MIT

Chapter 80

Changelog

All notable changes to this project will be documented in this file.
The format is based on Keep a Changelog and this project adheres to Semantic Versioning.

80.1 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.2.3...↩ v1.2.4" >v1.2.4</a> - 2024-02-05

80.1.1 Commits

- [Refactor] use all 7 <+ ES6 Errors from `es-errors` bcac811

80.2 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.2.2...↩ v1.2.3" >v1.2.3</a> - 2024-02-03

80.2.1 Commits

- [Refactor] use `es-errors`, so things that only need those do not need `get-intrinsic` f11db9c
- [Dev Deps] update `aud`, `es-abstract`, `mock-property`, `npmignore` b7ac7d1
- [meta] simplify `exports` faa0cc6
- [meta] add missing `engines.node` 774dd0b
- [Dev Deps] update `tape` 5828e8e
- [Robustness] use null objects for lookups eb9a11f
- [meta] add `sideEffects` flag 89bcc7a

80.3 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.2.1...↩ v1.2.2" >v1.2.2</a> - 2023-10-20

80.3.1 Commits

- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `call-bind`, `es-abstract`, `mock-property`, `object-inspect`, `tape` f51bcf2
- [Refactor] use `hasown` instead of `has` 18d14b7
- [Deps] update `function-bind` 6e109c8

80.4 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.2.0...↩ v1.2.1" >v1.2.1</a> - 2023-05-13

80.4.1 Commits

- [Fix] avoid a crash in envs without `__proto__` 7bad8d0
- [Dev Deps] update `es-abstract` c60e6b7

80.5 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.1.3...↩ v1.2.0" >v1.2.0</a> - 2023-01-19

80.5.1 Commits

- [actions] update checkout action ca6b12f
- [Dev Deps] update `@ljharb/eslint-config`, `es-abstract`, `object-inspect`, `tape` 41a3727
- [Fix] ensure `Error.prototype` is undeniable c511e97
- [Dev Deps] update `aud`, `es-abstract`, `tape` 1bef8a8
- [Dev Deps] update `aud`, `es-abstract` 0d41f16
- [New] add `BigInt64Array` and `BigUint64Array` a6cca25
- [Tests] use `gopd` ecf7722

80.6 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.1.2...↩ v1.1.3" >v1.1.3</a> - 2022-09-12

80.6.1 Commits

- [Dev Deps] update `es-abstract`, `es-value-fixtures`, `tape` 07ff291
- [Fix] properly check for % signs 50ac176

80.7 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.1.1...↩ v1.1.2" >v1.1.2</a> - 2022-06-08

80.7.1 Fixed

- [Fix] properly validate against extra % signs #16

80.7.2 Commits

- [actions] reuse common workflows 0972547
- [meta] use `npmignore` to autogenerate an npmignore file 5ba0b51
- [actions] use `node/install` instead of `node/run`; use `codecov` action c364492
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog`, `es-abstract`, `object-inspect`, `tape` dc04dad
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `es-abstract`, `object-inspect`, `safe-publish-latest`, `tape` 1c14059
- [Tests] use `mock-property` b396ef0

- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog`, `object-inspect`, `tape` c2c758d
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `es-abstract`, `es-value-fixtures`, `object-inspect`, `tape` 29e3c09
- [actions] update codecov uploader 8cbc141
- [Dev Deps] update `@ljharb/eslint-config`, `es-abstract`, `es-value-fixtures`, `object-inspect`, `tape` 10b6f5c
- [readme] add github actions/codecov badges 4e25400
- [Tests] use `for-each` instead of `foreach` c05b957
- [Dev Deps] update `es-abstract` 29b05ae
- [meta] use `prepublishOnly` script for npm 7+ 95c285d
- [Deps] update `has-symbols` 593cb4f
- [readme] fix repo URLs 1c8305b
- [Deps] update `has-symbols` c7138b6
- [Dev Deps] remove unused `has-bigints` bd63aff

80.8 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.1.0...↩ v1.1.1" >v1.1.1</a> - 2021-02-03

80.8.1 Fixed

- [meta] export `./package.json` #9

80.8.2 Commits

- [readme] flesh out the readme; use `evalmd` d12f12c
- [eslint] set up proper globals config 5a8c098
- [Dev Deps] update `eslint` 7b9a5c0

80.9 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.0.2...↩ v1.1.0" >v1.1.0</a> - 2021-01-25

80.9.1 Fixed

- [Refactor] delay `Function` eval until syntax-derived values are requested #3

80.9.2 Commits

- [Tests] migrate tests to Github Actions 2ab762b
- [meta] do not publish github action workflow files 5e7108e
- [Tests] add some coverage 01ac7a8
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `call-bind`, `es-abstract`, `tape`; add `call-bind` 911b672
- [Refactor] rearrange evalled constructors a bit 7e7e4bf
- [meta] add Automatic Rebase and Require Allow Edits workflows 0199968

80.10 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.0.1...↩ v1.0.2" >v1.0.2</a> - 2020-12-17

80.10.1 Commits

- [Fix] Throw for non-existent intrinsics `68f873b`
- [Fix] Throw for non-existent segments in the intrinsic path `8325dee`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `has-bigints`, `object-inspect` `0c227a7`
- [meta] do not lint coverage output `70d2419`

80.11 <a href="https://github.com/ljharb/get-intrinsic/compare/v1.0.0...↩ v1.0.1" >v1.0.1</a> - 2020-10-30

80.11.1 Commits

- [Tests] gather coverage data on every job `d1d280d`
- [Fix] add missing dependencies `5031771`
- [Tests] use `es-value-fixtures` `af48765`

80.12 v1.0.0 - 2020-10-29

80.12.1 Commits

- Implementation `bbce57c`
- Tests `17b4f0d`
- Initial commit `3153294`
- npm init `fb326c4`
- [meta] add Automatic Rebase and Require Allow Edits workflows `48862fb`
- [meta] add `auto-changelog` `5f28ad0`
- [meta] add "funding"; create `FUNDING.yml` `c2bbdde`
- [Tests] add `npm run lint` `0a84b98`
- Only apps should have lockfiles `9586c75`

Chapter 81

get-intrinsic <sup><a href="https://npmjs.org/package/get-intrinsic" ><img src="https://versionbadg.es/ljharb/get-intrinsic.svg" alt="Version Badge"/></a></sup>

[][license-url]

Get and robustly cache all JS language-level intrinsics at first require time.
See the syntax described in the JS spec for reference.

81.1 Example

```
var GetIntrinsic = require('get-intrinsic');
var assert = require('assert');
// static methods
assert.equal(GetIntrinsic('%Math.pow%'), Math.pow);
assert.equal(Math.pow(2, 3), 8);
assert.equal(GetIntrinsic('%Math.pow%')(2, 3), 8);
delete Math.pow;
assert.equal(GetIntrinsic('%Math.pow%')(2, 3), 8);
// instance methods
var arr = [1];
assert.equal(GetIntrinsic('%Array.prototype.push%'), Array.prototype.push);
assert.deepEqual(arr, [1]);
arr.push(2);
assert.deepEqual(arr, [1, 2]);
GetIntrinsic('%Array.prototype.push%').call(arr, 3);
assert.deepEqual(arr, [1, 2, 3]);
delete Array.prototype.push;
GetIntrinsic('%Array.prototype.push%').call(arr, 4);
assert.deepEqual(arr, [1, 2, 3, 4]);
// missing features
delete JSON.parse; // to simulate a real intrinsic that is missing in the environment
assert.throws(() => GetIntrinsic('%JSON.parse%'));
assert.equal(undefined, GetIntrinsic('%JSON.parse%', true));
```

81.2 Tests

Simply clone the repo, `npm install`, and run `npm test`

81.3 Security

Please email @ljharb or see https://tidelift.com/security if you have a potential security vulnerability to report.

Chapter 82

7.0

- Raise error if `options.cwd` is specified, and not a directory

82.1 6.0

- Remove comment and negation pattern support
- Ignore patterns are always in `dot:true` mode

82.2 5.0

- Deprecate comment and negation patterns
- Fix regression in `mark` and `nodir` options from making all cache keys absolute path.
- Abort if `fs.readdir` returns an error that's unexpected
- Don't emit `match` events for ignored items
- Treat ENOTSUP like ENOTDIR in readdir

82.3 4.5

- Add `options.follow` to always follow directory symlinks in globstar
- Add `options.realpath` to call `fs.realpath` on all results
- Always cache based on absolute path

82.4 4.4

- Add `options.ignore`
- Fix handling of broken symlinks

82.5 4.3

- Bump minimatch to 2.x
- Pass all tests on Windows

82.6 4.2

- Add `glob.hasMagic` function
- Add `options.nodir` flag

82.7 4.1

- Refactor sync and async implementations for performance
- Throw if callback provided to sync glob function
- Treat symbolic links in globstar results the same as Bash 4.3

82.8 4.0

- Use ^ for dependency versions (bumped major because this breaks older npm versions)
- Ensure callbacks are only ever called once
- switch to ISC license

82.9 3.x

- Rewrite in JavaScript
- Add support for setting root, cwd, and windows support
- Cache many fs calls
- Add globstar support
- emit match events

82.10 2.x

- Use `glob.h` and `fnmatch.h` from NetBSD

82.11 1.x

- `glob.h` static binding.

Chapter 83

Glob

Match files using the patterns the shell uses, like stars and stuff.

This is a glob implementation in JavaScript. It uses the `minimatch` library to do its matching.

83.1 Usage

Install with npm

```
npm i glob
var glob = require("glob")
// options is optional
glob("**/*.js", options, function (er, files) {
  // files is an array of filenames.
  // If the `nonull` option is set, and nothing
  // was found, then files is ["**/*.js"]
  // er is an error object or null.
})
```

83.2 Glob Primer

"Globs" are the patterns you type when you do stuff like `ls *.js` on the command line, or put `build/*` in a `.gitignore` file.
Before parsing the path part patterns, braced sections are expanded into a set. Braced sections start with `{` and end with `}`, with any number of comma-delimited sections within. Braced sections may contain slash characters, so `a{/b/c,bcd}` would expand into `a/b/c` and `abcd`.
The following characters have special magic meaning when used in a path portion:

- `*` Matches 0 or more characters in a single path portion
- `?` Matches 1 character
- `[...]` Matches a range of characters, similar to a RegExp range. If the first character of the range is `!` or `^` then it matches any character not in the range.
- `!(pattern|pattern|pattern)` Matches anything that does not match any of the patterns provided.
- `?(pattern|pattern|pattern)` Matches zero or one occurrence of the patterns provided.
- `+(pattern|pattern|pattern)` Matches one or more occurrences of the patterns provided.
- `*(a|b|c)` Matches zero or more occurrences of the patterns provided
- `@(pattern|pat*|pat?erN)` Matches exactly one of the patterns provided
- `**` If a "globstar" is alone in a path portion, then it matches zero or more directories and subdirectories searching for matches. It does not crawl symlinked directories.

83.2.1 Dots

If a file or directory path portion has a `.` as the first character, then it will not match any glob pattern unless that pattern's corresponding path part also has a `.` as its first character.
For example, the pattern `a/.*/c` would match the file at `a/.b/c`. However the pattern `a/*/c` would not, because `*` does not start with a dot character.
You can make glob treat dots as normal characters by setting `dot:true` in the options.

83.2.2 Basename Matching

If you set `matchBase:true` in the options, and the pattern has no slashes in it, then it will seek for any file anywhere in the tree with a matching basename. For example, `*.js` would match `test/simple/basic.js`.

83.2.3 Empty Sets

If no matching files are found, then an empty array is returned. This differs from the shell, where the pattern itself is returned. For example:

```
$ echo a*s*d*f
a*s*d*f
```

To get the bash-style behavior, set the `nonull:true` in the options.

83.2.4 See Also:

- `man sh`
- `man bash` (Search for "Pattern Matching")
- `man 3 fnmatch`
- `man 5 gitignore`
- minimatch documentation

83.3 glob.hasMagic(pattern, [options])

Returns `true` if there are any special characters in the pattern, and `false` otherwise.
Note that the options affect the results. If `noext:true` is set in the options object, then `+(a|b)` will not be considered a magic pattern. If the pattern has a brace expansion, like `a/{b/c,x/y}` then that is considered magical, unless `nobrace:true` is set in the options.

83.4 glob(pattern, [options], cb)

- `pattern {String}` Pattern to be matched
- `options {Object}`
- `cb {Function}`
  - `err {Error | null}`
  - `matches {Array<String>}` filenames found matching the pattern

Perform an asynchronous glob search.

83.5 glob.sync(pattern, [options])

- `pattern {String}` Pattern to be matched
- `options {Object}`
- return: `{Array<String>}` filenames found matching the pattern

Perform a synchronous glob search.

83.6 Class: glob.Glob

Create a Glob object by instantiating the `glob.Glob` class.

```
var Glob = require("glob").Glob
var mg = new Glob(pattern, options, cb)
```

It's an EventEmitter, and starts walking the filesystem to find matches immediately.

83.6.1 new glob.Glob(pattern, [options], [cb])

- `pattern` `{String}` pattern to search for
- `options` `{Object}`
- `cb` `{Function}` Called when an error occurs, or matches are found
  - `err` `{Error | null}`
  - `matches` `{Array<String>}` filenames found matching the pattern

Note that if the `sync` flag is set in the options, then matches will be immediately available on the `g.found` member.

83.6.2 Properties

- `minimatch` The minimatch object that the glob uses.
- `options` The options object passed in.
- `aborted` Boolean which is set to true when calling `abort()`. There is no way at this time to continue a glob search after aborting, but you can re-use the statCache to avoid having to duplicate syscalls.
- `cache` Convenience object. Each field has the following possible values:
  - `false` - Path does not exist
  - `true` - Path exists
  - "FILE'`- Path exists, and is not a directory *`'DIR'`- Path exists, and is a directory *`[file, entries, ...]`- Path exists, is a directory, and the array value is the results of`fs.readdir*statCache`Cache of`fs.stat` results, to prevent statting the same path multiple times.
- `symlinks` A record of which paths are symbolic links, which is relevant in resolving ** patterns.
- `realpathCache` An optional object which is passed to `fs.realpath` to minimize unnecessary syscalls. It is stored on the instantiated Glob object, and may be re-used.

83.6.3 Events

- `end` When the matching is finished, this is emitted with all the matches found. If the `nonull` option is set, and no match was found, then the `matches` list contains the original pattern. The matches are sorted, unless the `nosort` flag is set.
- `match` Every time a match is found, this is emitted with the specific thing that matched. It is not deduplicated or resolved to a realpath.
- `error` Emitted when an unexpected error is encountered, or whenever any fs error occurs if `options.↩ strict` is set.
- `abort` When `abort()` is called, this event is raised.

83.6.4 Methods

- `pause` Temporarily stop the search
- `resume` Resume the search
- `abort` Stop the search forever

83.6.5 Options

All the options that can be passed to Minimatch can also be passed to Glob to change pattern matching behavior. Also, some have been added, or have glob-specific ramifications.
All options are false by default, unless otherwise noted.
All options are added to the Glob object, as well.
If you are running many `glob` operations, you can pass a Glob object as the `options` argument to a subsequent operation to shortcut some `stat` and `readdir` calls. At the very least, you may pass in shared `symlinks`, `statCache`, `realpathCache`, and `cache` options, so that parallel glob operations will be sped up by sharing information about the filesystem.

- `cwd` The current working directory in which to search. Defaults to `process.cwd()`.
- `root` The place where patterns starting with / will be mounted onto. Defaults to `path.↵ resolve(options.cwd, "/")` (/ on Unix systems, and `C:\` or some such on Windows.)
- `dot` Include `.dot` files in normal matches and `globstar` matches. Note that an explicit dot in a portion of the pattern will always match dot files.
- `nomount` By default, a pattern starting with a forward-slash will be "mounted" onto the root setting, so that a valid filesystem path is returned. Set this flag to disable that behavior.
- `mark` Add a / character to directory matches. Note that this requires additional stat calls.
- `nosort` Don't sort the results.
- `stat` Set to true to stat *all* results. This reduces performance somewhat, and is completely unnecessary, unless `readdir` is presumed to be an untrustworthy indicator of file existence.
- `silent` When an unusual error is encountered when attempting to read a directory, a warning will be printed to stderr. Set the `silent` option to true to suppress these warnings.
- `strict` When an unusual error is encountered when attempting to read a directory, the process will just continue on in search of other matches. Set the `strict` option to raise an error in these cases.
- `cache` See `cache` property above. Pass in a previously generated cache object to save some fs calls.
- `statCache` A cache of results of filesystem information, to prevent unnecessary stat calls. While it should not normally be necessary to set this, you may pass the statCache from one glob() call to the options object of another, if you know that the filesystem will not change between calls. (See "Race Conditions" below.)
- `symlinks` A cache of known symbolic links. You may pass in a previously generated `symlinks` object to save `lstat` calls when resolving ∗∗ matches.
- `sync` DEPRECATED: use `glob.sync(pattern, opts)` instead.
- `nounique` In some cases, brace-expanded patterns can result in the same file showing up multiple times in the result set. By default, this implementation prevents duplicates in the result set. Set this flag to disable that behavior.
- `nonull` Set to never return an empty set, instead returning a set containing the pattern itself. This is the default in glob(3).
- `debug` Set to enable debug logging in minimatch and glob.
- `nobrace` Do not expand `{a,b}` and `{1..3}` brace sets.
- `noglobstar` Do not match ∗∗ against multiple filenames. (Ie, treat it as a normal ∗ instead.)
- `noext` Do not match `+(a|b)` "extglob" patterns.
- `nocase` Perform a case-insensitive match. Note: on case-insensitive filesystems, non-magic patterns will match by default, since `stat` and `readdir` will not raise errors.
- `matchBase` Perform a basename-only match if the pattern does not contain any slash characters. That is, `∗.js` would be treated as equivalent to `∗∗/∗.js`, matching all js files in all directories.

- `nodir` Do not match directories, only files. (Note: to match *only* directories, simply put a `/` at the end of the pattern.)
- `ignore` Add a pattern or an array of glob patterns to exclude matches. Note: `ignore` patterns are *always* in `dot:true` mode, regardless of any other settings.
- `follow` Follow symlinked directories when expanding `**` patterns. Note that this can result in a lot of duplicate references in the presence of cyclic links.
- `realpath` Set to true to call `fs.realpath` on all of the results. In the case of a symlink that cannot be resolved, the full absolute path to the matched entry is returned (though it will usually be a broken symlink)
- `absolute` Set to true to always receive absolute paths for matched files. Unlike `realpath`, this also affects the values returned in the `match` event.

83.7 Comparisons to other fnmatch/glob implementations

While strict compliance with the existing standards is a worthwhile goal, some discrepancies exist between node-glob and other implementations, and are intentional.
The double-star character `**` is supported by default, unless the `noglobstar` flag is set. This is supported in the manner of bsdglob and bash 4.3, where `**` only has special significance if it is the only thing in a path part. That is, `a/**/b` will match `a/x/y/b`, but `a/**b` will not.
Note that symlinked directories are not crawled as part of a `**`, though their contents may match against subsequent portions of the pattern. This prevents infinite loops and duplicates and the like.
If an escaped pattern has no matches, and the `nonull` flag is set, then glob returns the pattern as-provided, rather than interpreting the character escapes. For example, `glob.match([], "\\*a\\?")` will return `"\\*a\\?"` rather than `"*a?"`. This is akin to setting the `nullglob` option in bash, except that it does not resolve escaped pattern characters.
If brace expansion is not disabled, then it is performed before any other interpretation of the glob pattern. Thus, a pattern like `+(a|{b),c)}`, which would not be valid in bash or zsh, is expanded **first** into the set of `+(a|b)` and `+(a|c)`, and those patterns are checked for validity. Since those two are valid, matching proceeds.

83.7.1 Comments and Negation

Previously, this module let you mark a pattern as a "comment" if it started with a `#` character, or a "negated" pattern if it started with a `!` character.
These options were deprecated in version 5, and removed in version 6.
To specify things that should not match, use the `ignore` option.

83.8 Windows

Please only use forward-slashes in glob expressions.
Though windows uses either `/` or `\` as its path separator, only `/` characters are used by this glob implementation. You must use forward-slashes **only** in glob expressions. Back-slashes will always be interpreted as escape characters, not path separators.
Results from absolute patterns such as `/foo/*` are mounted onto the root setting using `path.join`. On windows, this will by default result in `/foo/*` matching `C:\foo\bar.txt`.

83.9 Race Conditions

Glob searching, by its very nature, is susceptible to race conditions, since it relies on directory walking and such.
As a result, it is possible that a file that exists when glob looks for it may have been deleted or modified by the time it returns the result.
As part of its internal implementation, this program caches all stat and readdir calls that it makes, in order to cut down on system overhead. However, this also makes it even more susceptible to races, especially if the cache or statCache objects are reused between glob calls.
Users are thus advised not to use a glob result as a guarantee of filesystem state in the face of rapid changes. For the vast majority of operations, this is never a problem.

83.10 Glob Logo

Glob's logo was created by Tanya Brassie. Logo files can be found here.
The logo is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

83.11 Contributing

Any change to behavior (including bugfixes) must come with a test.
Patches that fail tests or reduce performance will be rejected.

```
# to run tests
npm test
# to re-generate test fixtures
npm run test-regen
# to benchmark against bash/zsh
npm run bench
# to profile javascript
npm run prof
```

Chapter 84

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

84.1 <a href="https://github.com/ljharb/gopd/compare/v1.0.0...v1.0.1" >v1.0.1</a> - 2022-11-01

84.1.1 Commits

- [Fix] actually export gOPD instead of dP `4b624bf`

84.2 v1.0.0 - 2022-11-01

84.2.1 Commits

- Initial implementation, tests, readme `0911e01`
- Initial commit `b84e33f`
- [actions] add reusable workflows `12ae28a`
- npm init `280118b`
- [meta] add `auto-changelog` `bb78de5`
- [meta] create FUNDING.yml; add `funding` in package.json `11c22e6`
- [meta] use `npmignore` to autogenerate an npmignore file `4f4537a`
- Only apps should have lockfiles `c567022`

Chapter 85

gopd <sup><a href="https://npmjs.org/package/gopd" ><img src="https://versionbadg.es/ljharb/gopd.svg" alt="Version Badge"/></a></sup>

[][license-url]

`Object.getOwnPropertyDescriptor`, but accounts for IE's broken implementation.

85.1 Usage

```
var gOPD = require('gopd');
var assert = require('assert');
if (gOPD) {
    assert.equal(typeof gOPD, 'function', 'descriptors supported');
    // use gOPD like Object.getOwnPropertyDescriptor here
} else {
    assert.ok(!gOPD, 'descriptors not supported');
}
```

Chapter 86

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

86.1 <a href="https://github.com/inspect-js/has-property-descriptors/compare/v1.0.1...v1.0.2" >v1.0.2</a> - 2024-02-12

86.1.1 Commits

- [Refactor] use `es-define-property` `f93a8c8`
- [Dev Deps] update `aud`, `npmignore`, `tape` `42b0c9d`
- [Deps] update `get-intrinsic` `35e9b46`

86.2 <a href="https://github.com/inspect-js/has-property-descriptors/compare/v1.0.0...v1.0.1" >v1.0.1</a> - 2023-10-20

86.2.1 Commits

- [meta] use `npmignore` to autogenerate an npmignore file `5bbf4da`
- [actions] update rebase action to use reusable workflow `3a5585b`
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `tape` `e5c1212`
- [Dev Deps] update `aud`, `tape` `e942917`
- [Deps] update `get-intrinsic` `f4a44ec`
- [Deps] update `get-intrinsic` `eeb275b`

86.3 v1.0.0 - 2022-04-14

86.3.1 Commits

- Initial implementation, tests `303559f`
- Initial commit `3a7ca2d`
- read me `dd73dce`
- npm init `c1e6557`
- Only apps should have lockfiles `e72f7c6`

Chapter 87

has-property-descriptors <sup><a href="https↩ ://npmjs.org/package/has-property-descriptors" ><img src="https://versionbadg.es/inspect-js/has-property-descriptors.svg" alt="Version Badge"/></a></sup>

[][license-url]

Does the environment have full property descriptor support? Handles IE 8's broken defineProperty/gOPD.

87.1 Example

```
var hasPropertyDescriptors = require('has-property-descriptors');
var assert = require('assert');
assert.equal(hasPropertyDescriptors(), true); // will be `false` in IE 6-8, and ES5 engines
// Arrays can not have their length `[[Defined]]` in some engines
assert.equal(hasPropertyDescriptors.hasArrayLengthDefineBug(), false); // will be `true` in Firefox 4-22,
      and node v0.6
```

87.2 Tests

Simply clone the repo, `npm install`, and run `npm test`

Chapter 88

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

88.1 <a href="https://github.com/inspect-js/has-proto/compare/v1.0.↵ 2...v1.0.3" >v1.0.3</a> - 2024-02-19

88.1.1 Commits

- [types] add missing declaration file `26ecade`

88.2 <a href="https://github.com/inspect-js/has-proto/compare/v1.0.↵ 1...v1.0.2" >v1.0.2</a> - 2024-02-19

88.2.1 Commits

- add types `6435262`
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `npmignore`, `tape` `f16a5e4`
- [Refactor] tiny cleanup `d1f1a4b`
- [meta] add `sideEffects` flag `e7ab1a6`

88.3 <a href="https://github.com/inspect-js/has-proto/compare/v1.0.↵ 0...v1.0.1" >v1.0.1</a> - 2022-12-21

88.3.1 Commits

- [meta] correct URLs and description `ef34483`
- [patch] add an additional criteria `e81959e`
- [Dev Deps] update `aud` `2bec2c4`

88.4 v1.0.0 - 2022-12-12

88.4.1 Commits

- Initial implementation, tests, readme `6886fea`
- Initial commit `99129c8`
- npm init `2844ad8`

- Only apps should have lockfiles `c65bc5e`

Chapter 89

has-proto <sup><a href="https://npmjs.org/package/has-proto" ><img src="https://versionbadg.es/inspect-js/has-proto.svg" alt="Version Badge"/></a></sup>

[][license-url]

Does this environment have the ability to set the [[Prototype]] of an object on creation with `__proto__`?

89.1 Example

```
var hasProto = require('has-proto');
var assert = require('assert');
assert.equal(typeof hasProto(), 'boolean');
```

89.2 Tests

Simply clone the repo, `npm install`, and run `npm test`

Chapter 90

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

90.1 <a href="https://github.com/inspect-js/has-symbols/compare/v1.↩ 0.2...v1.0.3" >v1.0.3</a> - 2022-03-01

90.1.1 Commits

- [actions] use `node/install` instead of `node/run`; use `codecov` action `518b28f`
- [meta] add `bugs` and `homepage` fields; reorder package.json `c480b13`
- [actions] reuse common workflows `01d0ee0`
- [actions] update codecov uploader `6424ebe`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog`, `tape` `dfa7e7f`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `safe-publish-latest`, `tape` `0c8d436`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `tape` `9026554`
- [readme] add actions and codecov badges `eaa9682`
- [Dev Deps] update `eslint`, `tape` `bc7a3ba`
- [Dev Deps] update `eslint`, `auto-changelog` `0ace00a`
- [meta] use `prepublishOnly` script for npm 7+ `093f72b`
- [Tests] test on all 16 minors `9b80d3d`

90.2 <a href="https://github.com/inspect-js/has-symbols/compare/v1.↩ 0.1...v1.0.2" >v1.0.2</a> - 2021-02-27

90.2.1 Fixed

- [Fix] use a universal way to get the original Symbol `#11`

90.2.2 Commits

- [Tests] migrate tests to Github Actions 90ae798
- [meta] do not publish github action workflow files 29e60a1
- [Tests] run `nyc` on all tests 8476b91
- [readme] fix repo URLs, remove defunct badges 126288e
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog`, `core-js`, `get-own-property-symbols` d84bdfa
- [Tests] fix linting errors 0df3070
- [actions] add "Allow Edits" workflow 1e6bc29
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `tape` 36cea2a
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `tape` 1278338
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `tape` 1493254
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `core-js` b090bf2
- [actions] switch Automatic Rebase workflow to `pull_request_target` event 4addb7a
- [Dev Deps] update `auto-changelog`, `tape` 81d0baf
- [Dev Deps] update `auto-changelog`; add `aud` 1a4e561
- [readme] remove unused testling URLs 3000941
- [Tests] only audit prod deps 692e974
- [Dev Deps] update `@ljharb/eslint-config` 51c946c

90.3 <a href="https://github.com/inspect-js/has-symbols/compare/v1.↩ 0.0...v1.0.1" >v1.0.1</a> - 2019-11-16

90.3.1 Commits

- [Tests] use shared travis-ci configs ce396c9
- [Tests] up to `node` `v12.4`, `v11.15`, `v10.15`, `v9.11`, `v8.15`, `v7.10`, `v6.17`, `v4.9`; use `nvm install-latest-npm` 0690732
- [meta] add `auto-changelog` 2163d0b
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `core-js`, `safe-publish-latest`, `tape` 8e0951f
- [actions] add automatic rebasing / merge commit blocking b09cdb7
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `safe-publish-latest`, `core-js`, `get-own-property-symbols`, `tape` 1dd42cd
- [meta] create FUNDING.yml aa57a17
- Only apps should have lockfiles a2d8bea
- [Tests] use `npx aud` instead of `nsp` or `npm audit` with hoops 9e96cb7
- [meta] add `funding` field a0b32cf
- [Dev Deps] update `safe-publish-latest` cb9f0a5

90.4 v1.0.0 - 2016-09-19

90.4.1 Commits

- Tests. `ecb6eb9`
- package.json `88a337c`
- Initial commit `42e1e55`
- Initial implementation. `33f5cc6`
- read me `01f1170`

Chapter 91

has-symbols <sup><a href="https://npmjs.org/package/has-symbols" ><img src="https://versionbadg.es/inspect-js/has-symbols.svg" alt="Version Badge"/></a></sup>

[][license-url]

Determine if the JS environment has Symbol support. Supports spec, or shams.

91.1 Example

```
var hasSymbols = require('has-symbols');
hasSymbols() === true; // if the environment has native Symbol support.  Not polyfillable, not forgeable.
var hasSymbolsKinda = require('has-symbols/shams');
hasSymbolsKinda() === true; // if the environment has a Symbol sham that mostly follows the spec.
```

91.2 Supported Symbol shams

- get-own-property-symbols `npm` | `github`
- core-js `npm` | `github`

91.3 Tests

Simply clone the repo, `npm install`, and run `npm test`

Chapter 92

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

92.1 <a href="https://github.com/inspect-js/hasOwn/compare/v2.0.1...↩ v2.0.2" >v2.0.2</a> - 2024-03-10

92.1.1 Commits

- [types] use shared config `68e9d4d`
- [actions] remove redundant finisher; use reusable workflow `241a68e`
- [Tests] increase coverage `4125c0d`
- [Tests] skip `npm ls` in old node due to TS `01b9282`
- [types] improve predicate type `d340f85`
- [Dev Deps] update `tape` `70089fc`
- [Tests] use `@arethetypeswrong/cli` `50b272c`

92.2 <a href="https://github.com/inspect-js/hasOwn/compare/v2.0.0...↩ v2.0.1" >v2.0.1</a> - 2024-02-10

92.2.1 Commits

- [types] use a handwritten d.ts file; fix exported type `012b989`
- [Dev Deps] update `@types/function-bind`, `@types/mock-property`, `@types/tape`, `aud`, `mock-property`, `npmignore`, `tape`, `typescript` `977a56f`
- [meta] add `sideEffects` flag `3a60b7b`

92.3 <a href="https://github.com/inspect-js/hasOwn/compare/v1.0.1...↩ v2.0.0" >v2.0.0</a> - 2023-10-19

92.3.1 Commits

- revamped implementation, tests, readme `72bf8b3`
- [meta] revamp package.json `079775f`
- Only apps should have lockfiles `6640e23`

92.4 v1.0.1 - 2023-10-10

92.4.1 Commits

- Initial commit `8dbfde6`

Chapter 93

hasown <sup><a href="https://npmjs.org/package/hasown" ><img src="https://versionbadg.es/inspect-js/hasown.svg" alt="Version Badge"/></a></sup>

[][license-url]

A robust, ES3 compatible, "has own property" predicate.

93.1 Example

```
const assert = require('assert');
const hasOwn = require('hasown');
assert.equal(hasOwn({}, 'toString'), false);
assert.equal(hasOwn([], 'length'), true);
assert.equal(hasOwn({ a:  42 }, 'a'), true);
```

93.2 Tests

Simply clone the repo, `npm install`, and run `npm test`

Chapter 94

2.0.0 / 2021-12-17

- Drop support for Node.js 0.6
- Remove `I'mateapot` export; use `ImATeapot` instead
- Remove support for status being non-first argument
- Rename `UnorderedCollection` constructor to `TooEarly`
- deps: depd@2.0.0
  - Replace internal `eval` usage with `Function` constructor
  - Use instance methods on `process` to check for listeners
- deps: statuses@2.0.1
  - Fix messaging casing of `418 I'm a Teapot`
  - Remove code 306
  - Rename `425 Unordered Collection` to standard `425 Too Early`

94.1 2021-11-14 / 1.8.1

- deps: toidentifier@1.0.1

94.2 2020-06-29 / 1.8.0

- Add `isHttpError` export to determine if value is an HTTP error
- deps: setprototypeof@1.2.0

94.3 2019-06-24 / 1.7.3

- deps: inherits@2.0.4

94.4 2019-02-18 / 1.7.2

- deps: setprototypeof@1.1.1

94.5 2018-09-08 / 1.7.1

- Fix error creating objects in some environments

94.6 2018-07-30 / 1.7.0

- Set constructor name when possible
- Use `toidentifier` module to make class names
- deps: statuses@'>= 1.5.0 < 2'

94.7 2018-03-29 / 1.6.3

- deps: depd1.1.2
  - perf: remove argument reassignment
- deps: setprototypeof@1.1.0
- deps: statuses@'>= 1.4.0 < 2'

94.8 2017-08-04 / 1.6.2

- deps: depd@1.1.1
  - Remove unnecessary `Buffer` loading

94.9 2017-02-20 / 1.6.1

- deps: setprototypeof@1.0.3
  - Fix shim for old browsers

94.10 2017-02-14 / 1.6.0

- Accept custom 4xx and 5xx status codes in factory
- Add deprecation message to `"I'mateapot"` export
- Deprecate passing status code as anything except first argument in factory
- Deprecate using non-error status codes
- Make `message` property enumerable for `HttpError`s

94.11 2016-11-16 / 1.5.1

- deps: inherits@2.0.3
  - Fix issue loading in browser
- deps: setprototypeof@1.0.2
- deps: statuses@'>= 1.3.1 < 2'

94.12 2016-05-18 / 1.5.0

- Support new code `421 Misdirected Request`
- Use `setprototypeof` module to replace `__proto__` setting
- deps: statuses@'>= 1.3.0 < 2'
  - Add `421 Misdirected Request`
  - perf: enable strict mode
- perf: enable strict mode

94.13 2016-01-28 / 1.4.0

- Add `HttpError` export, for `err instanceof createError.HttpError`
- deps: inherits@2.0.1
- deps: statuses@'>= 1.2.1 < 2'
  - Fix message for status 451
  - Remove incorrect nginx status code

94.14 2015-02-02 / 1.3.1

- Fix regression where status can be overwritten in `createError` `props`

94.15 2015-02-01 / 1.3.0

- Construct errors using defined constructors from `createError`
- Fix error names that are not identifiers
  - `createError["I'mateapot"]` is now `createError.ImATeapot`
- Set a meaningful `name` property on constructed errors

94.16 2014-12-09 / 1.2.8

- Fix stack trace from exported function
- Remove `arguments.callee` usage

94.17 2014-10-14 / 1.2.7

- Remove duplicate line

94.18 2014-10-02 / 1.2.6

- Fix `expose` to be `true` for `ClientError` constructor

94.19 2014-09-28 / 1.2.5

- deps: statuses@1

94.20 2014-09-21 / 1.2.4

- Fix dependency version to work with old `npm`s

94.21 2014-09-21 / 1.2.3

- deps: statuses1.1.0

94.22 2014-09-21 / 1.2.2

- Fix publish error

94.23 2014-09-21 / 1.2.1

- Support Node.js 0.6
- Use `inherits` instead of `util`

94.24 2014-09-09 / 1.2.0

- Fix the way inheriting functions
- Support `expose` being provided in properties argument

94.25 2014-09-08 / 1.1.0

- Default status to 500
- Support provided `error` to extend

94.26 2014-09-08 / 1.0.1

- Fix accepting string message

94.27 2014-09-08 / 1.0.0

- Initial release

Chapter 95

http-errors

Create HTTP errors for Express, Koa, Connect, etc. with ease.

95.1 Install

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install http-errors
```

95.2 Example

```
var createError = require('http-errors')
var express = require('express')
var app = express()
app.use(function (req, res, next) {
  if (!req.user) return next(createError(401, 'Please login to view this page.'))
  next()
})
```

95.3 API

This is the current API, currently extracted from Koa and subject to change.

95.3.1 Error Properties

- `expose` - can be used to signal if `message` should be sent to the client, defaulting to `false` when `status` >= 500
- `headers` - can be an object of header names to values to be sent to the client, defaulting to `undefined`. When defined, the key names should all be lower-cased
- `message` - the traditional error message, which should be kept short and all single line
- `status` - the status code of the error, mirroring `statusCode` for general compatibility
- `statusCode` - the status code of the error, defaulting to `500`

95.3.2 createError([status], [message], [properties])

Create a new error object with the given message `msg`. The error object inherits from `createError.HttpError`.

```
var err = createError(404, 'This video does not exist!')
```

- `status: 500` - the status code as a number
- `message` - the message of the error, defaulting to node's text for that status code.
- `properties` - custom properties to attach to the object

95.3.3 createError([status], [error], [properties])

Extend the given `error` object with `createError.HttpError` properties. This will not alter the inheritance of the given `error` object, and the modified `error` object is the return value.

```
fs.readFile('foo.txt', function (err, buf) {
  if (err) {
    if (err.code === 'ENOENT') {
      var httpError = createError(404, err, { expose:  false })
    } else {
      var httpError = createError(500, err)
    }
  }
})
```

- `status` - the status code as a number
- `error` - the error object to extend
- `properties` - custom properties to attach to the object

95.3.4 createError.isHttpError(val)

Determine if the provided `val` is an `HttpError`. This will return `true` if the error inherits from the `Http↩ Error` constructor of this module or matches the "duck type" for an error this module creates. All outputs from the `createError` factory will return `true` for this function, including if an non-`HttpError` was passed into the factory.

95.3.5 new createError[code || name]([msg]))

Create a new error object with the given message `msg`. The error object inherits from `createError.Http↩ Error`.

```
var err = new createError.NotFound()
```

- `code` - the status code as a number
- `name` - the name of the error as a "bumpy case", i.e. `NotFound` or `InternalServerError`.

95.3.5.1 List of all constructors

Status Code	Constructor Name
400	BadRequest
401	Unauthorized
402	PaymentRequired
403	Forbidden
404	NotFound
405	MethodNotAllowed
406	NotAcceptable
407	ProxyAuthenticationRequired
408	RequestTimeout
409	Conflict
410	Gone
411	LengthRequired
412	PreconditionFailed
413	PayloadTooLarge
414	URITooLong
415	UnsupportedMediaType
416	RangeNotSatisfiable
417	ExpectationFailed
418	ImATeapot
421	MisdirectedRequest

Status Code	Constructor Name
422	UnprocessableEntity
423	Locked
424	FailedDependency
425	TooEarly
426	UpgradeRequired
428	PreconditionRequired
429	TooManyRequests
431	RequestHeaderFieldsTooLarge
451	UnavailableForLegalReasons
500	InternalServerError
501	NotImplemented
502	BadGateway
503	ServiceUnavailable
504	GatewayTimeout
505	HTTPVersionNotSupported
506	VariantAlsoNegotiates
507	InsufficientStorage
508	LoopDetected
509	BandwidthLimitExceeded
510	NotExtended
511	NetworkAuthenticationRequired

95.4 License

[MIT](LICENSE)

Chapter 96

0.4.24 / 2018-08-22

- Added MIK encoding (#196, by @Ivan-Kalatchev)

96.1 0.4.23 / 2018-05-07

- Fix deprecation warning in Node v10 due to the last usage of `new Buffer` (#185, by @felixbuenemann)
- Switched from NodeBuffer to Buffer in typings (#155 by @felixfbecker, #186 by @larssn)

96.2 0.4.22 / 2018-05-05

- Use older semver style for dependencies to be compatible with Node version 0.10 (#182, by @dougwilson)
- Fix tests to accomodate fixes in Node v10 (#182, by @dougwilson)

96.3 0.4.21 / 2018-04-06

- Fix encoding canonicalization (#156)
- Fix the paths in the "browser" field in package.json (#174 by @LMLB)
- Removed "contributors" section in package.json - see Git history instead.

96.4 0.4.20 / 2018-04-06

- Updated `new Buffer()` usages with recommended replacements as it's being deprecated in Node v10 (#176, #178 by @ChALkeR)

96.5 0.4.19 / 2017-09-09

- Fixed iso8859-1 codec regression in handling untranslatable characters (#162, caused by #147)
- Re-generated windows1255 codec, because it was updated in iconv project
- Fixed grammar in error message when iconv-lite is loaded with encoding other than utf8

96.6 0.4.18 / 2017-06-13

- Fixed CESU-8 regression in Node v8.

96.7 0.4.17 / 2017-04-22

- Updated typescript definition file to support Angular 2 AoT mode (#153 by @larssn)

96.8 0.4.16 / 2017-04-22

- Added support for React Native (#150)
- Changed iso8859-1 encoding to usine internal 'binary' encoding, as it's the same thing (#147 by @mscdex)
- Fixed typo in Readme (#138 by @jiangzhuo)
- Fixed build for Node v6.10+ by making correct version comparison
- Added a warning if iconv-lite is loaded not as utf-8 (see #142)

96.9 0.4.15 / 2016-11-21

- Fixed typescript type definition (#137)

96.10 0.4.14 / 2016-11-20

- Preparation for v1.0
- Added Node v6 and latest Node versions to Travis CI test rig
- Deprecated Node v0.8 support
- Typescript typings (@larssn)
- Fix encoding of Euro character in GB 18030 (inspired by @lygstate)
- Add ms prefix to dbcs windows encodings (@rokoroku)

96.11 0.4.13 / 2015-10-01

- Fix silly mistake in deprecation notice.

96.12 0.4.12 / 2015-09-26

- Node v4 support:
  - Added CESU-8 decoding (#106)
  - Added deprecation notice for `extendNodeEncodings`
  - Added Travis tests for Node v4 and io.js latest (#105 by @Mithgol)

96.13 0.4.11 / 2015-07-03

- Added CESU-8 encoding.

96.14 0.4.10 / 2015-05-26

- Changed UTF-16 endianness heuristic to take into account any ASCII chars, not just spaces. This should minimize the importance of "default" endianness.

96.15 0.4.9 / 2015-05-24

- Streamlined BOM handling: strip BOM by default, add BOM when encoding if addBOM: true. Added docs to Readme.
- UTF16 now uses UTF16-LE by default.
- Fixed minor issue with big5 encoding.
- Added io.js testing on Travis; updated node-iconv version to test against. Now we just skip testing SBCS encodings that node-iconv doesn't support.
- (internal refactoring) Updated codec interface to use classes.
- Use strict mode in all files.

96.16 0.4.8 / 2015-04-14

- added alias UNICODE-1-1-UTF-7 for UTF-7 encoding (#94)

96.17 0.4.7 / 2015-02-05

- stop official support of Node.js v0.8. Should still work, but no guarantees. reason: Packages needed for testing are hard to get on Travis CI.
- work in environment where Object.prototype is monkey patched with enumerable props (#89).

96.18 0.4.6 / 2015-01-12

- fix rare aliases of single-byte encodings (thanks @mscdex)
- double the timeout for dbcs tests to make them less flaky on travis

96.19 0.4.5 / 2014-11-20

- fix windows-31j and x-sjis encoding support (@nleush)
- minor fix: undefined variable reference when internal error happens

96.20 0.4.4 / 2014-07-16

- added encodings UTF-7 (RFC2152) and UTF-7-IMAP (RFC3501 Section 5.1.3)
- fixed streaming base64 encoding

96.21 0.4.3 / 2014-06-14

- added encodings UTF-16BE and UTF-16 with BOM

96.22 0.4.2 / 2014-06-12

- don't throw exception if `extendNodeEncodings()` is called more than once

96.23 0.4.1 / 2014-06-11

- codepage 808 added

96.24 0.4.0 / 2014-06-10

- code is rewritten from scratch
- all widespread encodings are supported
- streaming interface added
- browserify compatibility added
- (optional) extend core primitive encodings to make usage even simpler
- moved from vows to mocha as the testing framework

Chapter 97

Pure JS character encoding conversion <a href="https://travis-ci.org/ashtuchkin/iconv-lite" ><img src="https://travis-ci.org/ashtuchkin/iconv-lite.svg?branch=master" alt="Build Status"/></a>

- Doesn't need native code compilation. Works on Windows and in sandboxed environments like `Cloud9`.
- Used in popular projects like `Express.js (body_parser)`, `Grunt`, `Nodemailer`, `Yeoman` and others.
- Faster than `node-iconv` (see below for performance comparison).
- Intuitive encode/decode API
- Streaming support for Node v0.10+
- [Deprecated] Can extend Node.js primitives (buffers, streams) to support all iconv-lite encodings.
- In-browser usage via `Browserify` (~180k gzip compressed with Buffer shim included).
- Typescript `type definition file` included.
- React Native is supported (need to explicitly `npm install` two more modules: `buffer` and `stream`).
- License: MIT.

97.1 Usage

97.1.1 Basic API

```
var iconv = require('iconv-lite');
// Convert from an encoded buffer to js string.
str = iconv.decode(Buffer.from([0x68, 0x65, 0x6c, 0x6c, 0x6f]), 'win1251');
// Convert from js string to an encoded buffer.
buf = iconv.encode("Sample input string", 'win1251');
// Check if encoding is supported
iconv.encodingExists("us-ascii")
```

97.1.2 Streaming API (Node v0.10+)

```
// Decode stream (from binary stream to js strings)
http.createServer(function(req, res) {
    var converterStream = iconv.decodeStream('win1251');
    req.pipe(converterStream);
    converterStream.on('data', function(str) {
        console.log(str); // Do something with decoded strings, chunk-by-chunk.
    });
});
// Convert encoding streaming example
fs.createReadStream('file-in-win1251.txt')
    .pipe(iconv.decodeStream('win1251'))
    .pipe(iconv.encodeStream('ucs2'))
    .pipe(fs.createWriteStream('file-in-ucs2.txt'));
// Sugar:  all encode/decode streams have .collect(cb) method to accumulate data.
http.createServer(function(req, res) {
    req.pipe(iconv.decodeStream('win1251')).collect(function(err, body) {
        assert(typeof body == 'string');
        console.log(body); // full request body string
    });
});
```

97.1.3 [Deprecated] Extend Node.js own encodings

NOTE: This doesn't work on latest Node versions. See details.

```
// After this call all Node basic primitives will understand iconv-lite encodings.
iconv.extendNodeEncodings();
// Examples:
buf = new Buffer(str, 'win1251');
buf.write(str, 'gbk');
str = buf.toString('latin1');
assert(Buffer.isEncoding('iso-8859-15'));
Buffer.byteLength(str, 'us-ascii');
http.createServer(function(req, res) {
    req.setEncoding('big5');
    req.collect(function(err, body) {
        console.log(body);
    });
});
fs.createReadStream("file.txt", "shift_jis");
// External modules are also supported (if they use Node primitives, which they probably do).
request = require('request');
request({
    url:  "http://github.com/",
    encoding:  "cp932"
});
// To remove extensions
iconv.undoExtendNodeEncodings();
```

97.2 Supported encodings

- All node.js native encodings: utf8, ucs2 / utf16-le, ascii, binary, base64, hex.
- Additional unicode encodings: utf16, utf16-be, utf-7, utf-7-imap.
- All widespread singlebyte encodings: Windows 125x family, ISO-8859 family, IBM/DOS codepages, Macintosh family, KOI8 family, all others supported by iconv library. Aliases like 'latin1', 'us-ascii' also supported.
- All widespread multibyte encodings: CP932, CP936, CP949, CP950, GB2312, GBK, GB18030, Big5, Shift↩ _JIS, EUC-JP.

See all supported encodings on wiki.
Most singlebyte encodings are generated automatically from node-iconv. Thank you Ben Noordhuis and libiconv authors!
Multibyte encodings are generated from Unicode.org mappings and WHATWG Encoding Standard mappings. Thank you, respective authors!

97.3 Encoding/decoding speed

Comparison with node-iconv module (1000x256kb, on MacBook Pro, Core i5/2.6 GHz, Node v0.12.0). Note: your results may vary, so please always check on your hardware.

```
operation             iconv@2.1.4   iconv-lite@0.4.7
----------------------------------------------------------
encode('win1251')     ~96 Mb/s      ~320 Mb/s
decode('win1251')     ~95 Mb/s      ~246 Mb/s
```

97.4 BOM handling

- Decoding: BOM is stripped by default, unless overridden by passing `stripBOM: false` in options (f.↩ ex. `iconv.decode(buf, enc, {stripBOM: false})`). A callback might also be given as a `stripBOM` parameter - it'll be called if BOM character was actually found.
- If you want to detect UTF-8 BOM when decoding other encodings, use `node-autodetect-decoder-stream` module.
- Encoding: No BOM added, unless overridden by `addBOM: true` option.

97.5 UTF-16 Encodings

This library supports UTF-16LE, UTF-16BE and UTF-16 encodings. First two are straightforward, but UTF-16 is trying to be smart about endianness in the following ways:

- Decoding: uses BOM and 'spaces heuristic' to determine input endianness. Default is UTF-16LE, but can be overridden with ‘defaultEncoding: 'utf-16be’`option. Strips BOM unless`stripBOM: false`.`
- `Encoding: uses UTF-16LE and writes BOM by default. Use`addBOM: false` to override.

97.6 Other notes

When decoding, be sure to supply a Buffer to decode() method, otherwise `bad things usually happen`.
Untranslatable characters are set to � or ?. No transliteration is currently supported.
Node versions 0.10.31 and 0.11.13 are buggy, don't use them (see #65, #77).

97.7 Testing

```
$ git clone git@github.com:ashtuchkin/iconv-lite.git
$ cd iconv-lite
$ npm install
$ npm test

$ # To view performance:
$ node test/performance.js
$ # To view test coverage:
$ npm run coverage
$ open coverage/lcov-report/index.html
```

Chapter 98

inflight

Add callbacks to requests in flight to avoid async duplication

98.1 USAGE

```
var inflight = require('inflight')
// some request that does some stuff
function req(key, callback) {
  // key is any random string.    like a url or filename or whatever.
  //
  // will return either a falsey value, indicating that the
  // request for this key is already in flight, or a new callback
  // which when called will call all callbacks passed to inflightk
  // with the same key
  callback = inflight(key, callback)
  // If we got a falsey value back, then there's already a req going
  if (!callback) return
  // this is where you'd fetch the url or whatever
  // callback is also once()-ified, so it can safely be assigned
  // to multiple events etc.    First call wins.
  setTimeout(function() {
    callback(null, key)
  }, 100)
}
// only assigns a single setTimeout
// when it dings, all cbs get called
req('foo', cb1)
req('foo', cb2)
req('foo', cb3)
req('foo', cb4)
```

Chapter 99

README

Browser-friendly inheritance fully compatible with standard node.js inherits.
This package exports standard inherits from node.js util module in node environment, but also provides alternative browser-friendly implementation through browser field. Alternative implementation is a literal copy of standard one located in standalone module to avoid requiring of util. It also has a shim for old browsers with no Object.create support.
While keeping you sure you are using standard inherits implementation in node.js environment, it allows bundlers such as browserify to not include full util package to your client code if all you need is just inherits function. It worth, because browser shim for util package is large and inherits is often the single function you need from it.
It's recommended to use this package instead of ‘require('util’).inherits` for any code that has chances to be used not only in node.js but in browser too.

99.0.1 usage

```
var inherits = require('inherits');
// then use exactly as the standard one
```

99.0.2 note on version ∼1.0

Version ∼1.0 had completely different motivation and is not compatible neither with 2.0 nor with standard node.js inherits.
If you are using version ∼1.0 and planning to switch to ∼2.0, be careful:

- new version uses super_ instead of super for referencing superclass
- new version overwrites current prototype while old one preserves any existing fields on it

Chapter 100

ipaddr.js — an IPv6 and IPv4 address manipulation library <a href="https://travis-ci.org/whitequark/ipaddr.js" ><img src="https://travis-ci.org/whitequark/ipaddr.js.svg" alt="Build Status"/></a>

ipaddr.js is a small (1.9K minified and gzipped) library for manipulating IP addresses in JavaScript environments. It runs on both CommonJS runtimes (e.g. `nodejs`) and in a web browser.
ipaddr.js allows you to verify and parse string representation of an IP address, match it against a CIDR range or range list, determine if it falls into some reserved ranges (examples include loopback and private ranges), and convert between IPv4 and IPv4-mapped IPv6 addresses.

100.1 Installation

`npm install ipaddr.js`
or
`bower install ipaddr.js`

100.2 API

ipaddr.js defines one object in the global scope: `ipaddr`. In CommonJS, it is exported from the module:

```
var ipaddr = require('ipaddr.js');
```

The API consists of several global methods and two classes: ipaddr.IPv6 and ipaddr.IPv4.

100.2.1 Global methods

There are three global methods defined: `ipaddr.isValid`, `ipaddr.parse` and `ipaddr.process`. All of them receive a string as a single parameter.
The `ipaddr.isValid` method returns `true` if the address is a valid IPv4 or IPv6 address, and `false` otherwise. It does not throw any exceptions.
The `ipaddr.parse` method returns an object representing the IP address, or throws an `Error` if the passed string is not a valid representation of an IP address.
The `ipaddr.process` method works just like the `ipaddr.parse` one, but it automatically converts IPv4-mapped IPv6 addresses to their IPv4 counterparts before returning. It is useful when you have a Node.js instance listening on an IPv6 socket, and the `net.ivp6.bindv6only` sysctl parameter (or its equivalent on non-Linux

OS) is set to 0. In this case, you can accept IPv4 connections on your IPv6-only socket, but the remote address will be mangled. Use `ipaddr.process` method to automatically demangle it.

100.2.2 Object representation

Parsing methods return an object which descends from `ipaddr.IPv6` or `ipaddr.IPv4`. These objects share some properties, but most of them differ.

100.2.2.1 Shared properties

One can determine the type of address by calling `addr.kind()`. It will return either `"ipv6"` or `"ipv4"`.
An address can be converted back to its string representation with `addr.toString()`. Note that this method:

- does not return the original string used to create the object (in fact, there is no way of getting that string)
- returns a compact representation (when it is applicable)

A `match(range, bits)` method can be used to check if the address falls into a certain CIDR range. Note that an address can be (obviously) matched only against an address of the same type.
For example:

```
var addr = ipaddr.parse("2001:db8:1234::1");
var range = ipaddr.parse("2001:db8::");
addr.match(range, 32); // => true
```

Alternatively, `match` can also be called as `match([range, bits])`. In this way, it can be used together with the `parseCIDR(string)` method, which parses an IP address together with a CIDR range.
For example:

```
var addr = ipaddr.parse("2001:db8:1234::1");
addr.match(ipaddr.parseCIDR("2001:db8::/32")); // => true
```

A `range()` method returns one of predefined names for several special ranges defined by IP protocols. The exact names (and their respective CIDR ranges) can be looked up in the source: `IPv6 ranges` and `IPv4 ranges`. Some common ones include `"unicast"` (the default one) and `"reserved"`.
You can match against your own range list by using `ipaddr.subnetMatch(address, rangeList, defaultName)` method. It can work with a mix of IPv6 or IPv4 addresses, and accepts a name-to-subnet map as the range list. For example:

```
var rangeList = {
  documentationOnly:  [ ipaddr.parse('2001:db8::'), 32 ],
  tunnelProviders:  [
    [ ipaddr.parse('2001:470::'), 32 ], // he.net
    [ ipaddr.parse('2001:5c0::'), 32 ]  // freenet6
  ]
};
ipaddr.subnetMatch(ipaddr.parse('2001:470:8:66::1'), rangeList, 'unknown'); // => "tunnelProviders"
```

The addresses can be converted to their byte representation with `toByteArray()`. (Actually, JavaScript mostly does not know about byte buffers. They are emulated with arrays of numbers, each in range of 0..255.)

```
var bytes = ipaddr.parse('2a00:1450:8007::68').toByteArray(); // ipv6.google.com
bytes // => [42, 0x00, 0x14, 0x50, 0x80, 0x07, 0x00, <zeroes...>, 0x00, 0x68 ]
```

The `ipaddr.IPv4` and `ipaddr.IPv6` objects have some methods defined, too. All of them have the same interface for both protocols, and are similar to global methods.
`ipaddr.IPvX.isValid(string)` can be used to check if the string is a valid address for particular protocol, and `ipaddr.IPvX.parse(string)` is the error-throwing parser.
`ipaddr.IPvX.isValid(string)` uses the same format for parsing as the POSIX `inet_ntoa` function, which accepts unusual formats like `0xc0.168.1.1` or `0x10000000`. The function `ipaddr.IPv4.is↩ValidFourPartDecimal(string)` validates the IPv4 address and also ensures that it is written in four-part decimal format.

100.2.2.2 IPv6 properties

Sometimes you will want to convert IPv6 not to a compact string representation (with the `::` substitution); the `toNormalizedString()` method will return an address where all zeroes are explicit.
For example:

```
var addr = ipaddr.parse("2001:0db8::0001");
addr.toString(); // => "2001:db8::1"
addr.toNormalizedString(); // => "2001:db8:0:0:0:0:0:1"
```

The `isIPv4MappedAddress()` method will return `true` if this address is an IPv4-mapped one, and `to↩IPv4Address()` will return an IPv4 object address.
To access the underlying binary representation of the address, use `addr.parts`.

```
var addr = ipaddr.parse("2001:db8:10::1234:DEAD");
addr.parts // => [0x2001, 0xdb8, 0x10, 0, 0, 0, 0x1234, 0xdead]
```

A IPv6 zone index can be accessed via `addr.zoneId`:

```
var addr = ipaddr.parse("2001:db8::%eth0");
addr.zoneId // => 'eth0'
```

100.2.2.3 IPv4 properties

`toIPv4MappedAddress()` will return a corresponding IPv4-mapped IPv6 address.

To access the underlying representation of the address, use `addr.octets`.

```
var addr = ipaddr.parse("192.168.1.1");
addr.octets // => [192, 168, 1, 1]
```

`prefixLengthFromSubnetMask()` will return a CIDR prefix length for a valid IPv4 netmask or null if the netmask is not valid.

```
ipaddr.IPv4.parse('255.255.255.240').prefixLengthFromSubnetMask() == 28
ipaddr.IPv4.parse('255.192.164.0').prefixLengthFromSubnetMask()  == null
```

`subnetMaskFromPrefixLength()` will return an IPv4 netmask for a valid CIDR prefix length.

```
ipaddr.IPv4.subnetMaskFromPrefixLength(24) == "255.255.255.0"
ipaddr.IPv4.subnetMaskFromPrefixLength(29) == "255.255.255.248"
```

`broadcastAddressFromCIDR()` will return the broadcast address for a given IPv4 interface and netmask in CIDR notation.

```
ipaddr.IPv4.broadcastAddressFromCIDR("172.0.0.1/24") == "172.0.0.255"
```

`networkAddressFromCIDR()` will return the network address for a given IPv4 interface and netmask in CIDR notation.

```
ipaddr.IPv4.networkAddressFromCIDR("172.0.0.1/24") == "172.0.0.0"
```

100.2.2.4 Conversion

IPv4 and IPv6 can be converted bidirectionally to and from network byte order (MSB) byte arrays.

The `fromByteArray()` method will take an array and create an appropriate IPv4 or IPv6 object if the input satisfies the requirements. For IPv4 it has to be an array of four 8-bit values, while for IPv6 it has to be an array of sixteen 8-bit values.

For example:

```
var addr = ipaddr.fromByteArray([0x7f, 0, 0, 1]);
addr.toString(); // => "127.0.0.1"
```

or

```
var addr = ipaddr.fromByteArray([0x20, 1, 0xd, 0xb8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1])
addr.toString(); // => "2001:db8::1"
```

Both objects also offer a `toByteArray()` method, which returns an array in network byte order (MSB).

For example:

```
var addr = ipaddr.parse("127.0.0.1");
addr.toByteArray(); // => [0x7f, 0, 0, 1]
```

or

```
var addr = ipaddr.parse("2001:db8::1");
addr.toByteArray(); // => [0x20, 1, 0xd, 0xb8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
```

Chapter 101

is-property

Tests if a property of a JavaScript object can be accessed using the dot (.) notation or if it must be enclosed in brackets, (ie use x[" ... "])

101.1 Example

```
var isProperty = require("is-property")
console.log(isProperty("foo"))  //Prints true
console.log(isProperty("0"))    //Prints false
```

101.2 Install

```
npm install is-property
```

101.2.1 <tt>require("is-property")(str)</tt>

Checks if str is a property

- `str` is a string which we will test if it is a property or not

Returns true or false depending if str is a property

101.3 Credits

(c) 2013 Mikola Lysenko. MIT License

Chapter 102

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog`, and this project adheres to `Semantic Versioning`.

102.1 <a href="https://github.com/nodeca/js-yaml/compare/4.0.0...4.1.0" >4.1.0</a> - 2021-04-15

102.1.1 Added

- Types are now exported as `yaml.types.XXX`.
- Every type now has `options` property with original arguments kept as they were (see `yaml.types.↩ int.options` as an example).

102.1.2 Changed

- `Schema.extend()` now keeps old type order in case of conflicts (e.g. Schema.extend([a, b, c]).extend([b, a, d]) is now ordered as `abcd` instead of `cbad`).

102.2 <a href="https://github.com/nodeca/js-yaml/compare/3.14.0...4.0.0" >4.0.0</a> - 2021-01-03

102.2.1 Changed

- Check migration guide to see details for all breaking changes.
- Breaking: "unsafe" tags `!!js/function`, `!!js/regexp`, `!!js/undefined` are moved to `js-yaml-js-types` package.
- Breaking: removed `safe*` functions. Use `load`, `loadAll`, `dump` instead which are all now safe by default.
- `yaml.DEFAULT_SAFE_SCHEMA` and `yaml.DEFAULT_FULL_SCHEMA` are removed, use `yaml.↩ DEFAULT_SCHEMA` instead.
- `yaml.Schema.create(schema, tags)` is removed, use `schema.extend(tags)` instead.
- `!!binary` now always mapped to `Uint8Array` on load.
- Reduced nesting of `/lib` folder.
- Parse numbers according to YAML 1.2 instead of YAML 1.1 (`01234` is now decimal, `0o1234` is octal, `1:23` is parsed as string instead of base60).
- `dump()` no longer quotes `:`, `[`, `]`, `(`, `)` except when necessary, #470, #557.

- Line and column in exceptions are now formatted as `(X:Y)` instead of `at line X, column Y` (also present in compact format), #332.
- Code snippet created in exceptions now contains multiple lines with line numbers.
- `dump()` now serializes `undefined` as `null` in collections and removes keys with `undefined` in mappings, #571.
- `dump()` with `skipInvalid=true` now serializes invalid items in collections as null.
- Custom tags starting with `!` are now dumped as `!tag` instead of `!<!tag>`, #576.
- Custom tags starting with `tag:yaml.org,2002:` are now shorthanded using `!!`, #258.

102.2.2 Added

- Added `.mjs` (es modules) support.
- Added `quotingType` and `forceQuotes` options for dumper to configure string literal style, #290, #529.
- Added ‘styles: { '!!null': 'empty' }`option for dumper (serializes`{ foo: null }`as "</tt>foo: <tt>"), #570.`
- `Added`replacer`option (similar to option in JSON.stringify), #339.`
- `CustomTag` can now handle all tags or multiple tags with the same prefix, #385.

102.2.3 Fixed

- Astral characters are no longer encoded by `dump()`, #587.
- "duplicate mapping key" exception now points at the correct column, #452.
- Extra commas in flow collections (e.g. `[foo,,bar]`) now throw an exception instead of producing null, #321.
- __`proto`__ key no longer overrides object prototype, #164.
- Removed `bower.json`.
- Tags are now url-decoded in `load()` and url-encoded in `dump()` (previously usage of custom non-ascii tags may have led to invalid YAML that can't be parsed).
- Anchors now work correctly with empty nodes, #301.
- Fix incorrect parsing of invalid block mapping syntax, #418.
- Throw an error if block sequence/mapping indent contains a tab, #80.

102.3 [3.14.1] - 2020-12-07

102.3.1 Security

- Fix possible code execution in (already unsafe) `.load()` (in &anchor).

102.4 <a href="https://github.com/nodeca/js-yaml/compare/3.13.1...3.14.0" >3.14.0</a> - 2020-05-22

102.4.1 Changed

- Support `safe/loadAll(input, options)` variant of call.
- CI: drop outdated nodejs versions.
- Dev deps bump.

102.4.2 Fixed

- Quote = in plain scalars #519.
- Check the node type for `!<?>` tag in case user manually specifies it.
- Verify that there are no null-bytes in input.
- Fix wrong quote position when writing condensed flow, #526.

102.5 <a href="https://github.com/nodeca/js-yaml/compare/3.13.0...3.13.1" >3.13.1</a> - 2019-04-05

102.5.1 Security

- Fix possible code execution in (already unsafe) `.load()`, #480.

102.6 <a href="https://github.com/nodeca/js-yaml/compare/3.12.2...3.13.0" >3.13.0</a> - 2019-03-20

102.6.1 Security

- Security fix: `safeLoad()` can hang when arrays with nested refs used as key. Now throws exception for nested arrays. #475.

102.7 <a href="https://github.com/nodeca/js-yaml/compare/3.12.1...3.12.2" >3.12.2</a> - 2019-02-26

102.7.1 Fixed

- Fix `noArrayIndent` option for root level, #468.

102.8 <a href="https://github.com/nodeca/js-yaml/compare/3.12.0...3.12.1" >3.12.1</a> - 2019-01-05

102.8.1 Added

- Added `noArrayIndent` option, #432.

102.9 <a href="https://github.com/nodeca/js-yaml/compare/3.11.0...3.12.0" >3.12.0</a> - 2018-06-02

102.9.1 Changed

- Support arrow functions without a block statement, #421.

102.10 <a href="https://github.com/nodeca/js-yaml/compare/3.10.0...3.11.0" >3.11.0</a> - 2018-03-05

102.10.1 Added

- Add arrow functions suport for `!!js/function`.

102.10.2 Fixed

- Fix dump in bin/octal/hex formats for negative integers, #399.

102.11 <a href="https://github.com/nodeca/js-yaml/compare/3.9.1...3.10.0" >3.10.0</a> - 2017-09-10

102.11.1 Fixed

- Fix `condenseFlow` output (quote keys for sure, instead of spaces), #371, #370.
- Dump astrals as codepoints instead of surrogate pair, #368.

102.12 <a href="https://github.com/nodeca/js-yaml/compare/3.9.0...3.9.1" >3.9.1</a> - 2017-07-08

102.12.1 Fixed

- Ensure stack is present for custom errors in node 7.+, #351.

102.13 <a href="https://github.com/nodeca/js-yaml/compare/3.8.4...3.9.0" >3.9.0</a> - 2017-07-08

102.13.1 Added

- Add `condenseFlow` option (to create pretty URL query params), #346.

102.13.2 Fixed

- Support array return from safeLoadAll/loadAll, #350.

102.14 <a href="https://github.com/nodeca/js-yaml/compare/3.8.3...3.8.4" >3.8.4</a> - 2017-05-08

102.14.1 Fixed

- Dumper: prevent space after dash for arrays that wrap, #343.

102.15 <a href="https://github.com/nodeca/js-yaml/compare/3.8.2...3.8.3" >3.8.3</a> - 2017-04-05

102.15.1 Fixed

- Should not allow numbers to begin and end with underscore, #335.

102.16 <a href="https://github.com/nodeca/js-yaml/compare/3.8.1...3.8.2" >3.8.2</a> - 2017-03-02

102.16.1 Fixed

- Fix `!!float 123` (integers) parse, #333.
- Don't allow leading zeros in floats (except 0, 0.xxx).
- Allow positive exponent without sign in floats.

102.17 <a href="https://github.com/nodeca/js-yaml/compare/3.8.0...3.8.1" >3.8.1</a> - 2017-02-07

102.17.1 Changed

- Maintenance: update browserified build.

102.18 <a href="https://github.com/nodeca/js-yaml/compare/3.7.0...3.8.0" >3.8.0</a> - 2017-02-07

102.18.1 Fixed

- Fix reported position for `duplicated mapping key` errors. Now points to block start instead of block end. (#243, thanks to @shockey).

102.19 <a href="https://github.com/nodeca/js-yaml/compare/3.6.1...3.7.0" >3.7.0</a> - 2016-11-12

102.19.1 Added

- Support polymorphism for tags (#300, thanks to @monken).

102.19.2 Fixed

- Fix parsing of quotes followed by newlines (#304, thanks to @dplepage).

102.20 <a href="https://github.com/nodeca/js-yaml/compare/3.6.0...3.6.1" >3.6.1</a> - 2016-05-11

102.20.1 Fixed

- Fix output cut on a pipe, #286.

102.21 <a href="https://github.com/nodeca/js-yaml/compare/3.5.5...3.6.0" >3.6.0</a> - 2016-04-16

102.21.1 Fixed

- Dumper rewrite, fix multiple bugs with trailing `\n`. Big thanks to @aepsilon!
- Loader: fix leading/trailing newlines in block scalars, @aepsilon.

102.22 <a href="https://github.com/nodeca/js-yaml/compare/3.5.4...3.5.5" >3.5.5</a> - 2016-03-17

102.22.1 Fixed

- Date parse fix: don't allow dates with on digit in month and day, #268.

102.23 <a href="https://github.com/nodeca/js-yaml/compare/3.5.3...3.5.4" >3.5.4</a> - 2016-03-09

102.23.1 Added

- `noCompatMode` for dumper, to disable quoting YAML 1.1 values.

102.24 <a href="https://github.com/nodeca/js-yaml/compare/3.5.2...3.5.3" >3.5.3</a> - 2016-02-11

102.24.1 Changed

- Maintenance release.

102.25 <a href="https://github.com/nodeca/js-yaml/compare/3.5.1...3.5.2" >3.5.2</a> - 2016-01-11

102.25.1 Changed

- Maintenance: missed comma in bower config.

102.26 <a href="https://github.com/nodeca/js-yaml/compare/3.5.0...3.5.1" >3.5.1</a> - 2016-01-11

102.26.1 Changed

- Removed `inherit` dependency, #239.
- Better browserify workaround for esprima load.
- Demo rewrite.

102.27 <a href="https://github.com/nodeca/js-yaml/compare/3.4.6...3.5.0" >3.5.0</a> - 2016-01-10

102.27.1 Fixed

- Dumper. Fold strings only, #217.
- Dumper. `norefs` option, to clone linked objects, #229.
- Loader. Throw a warning for duplicate keys, #166.
- Improved browserify support (mark `esprima` & `Buffer` excluded).

102.28 <a href="https://github.com/nodeca/js-yaml/compare/3.4.5...3.4.6" >3.4.6</a> - 2015-11-26

102.28.1 Changed

- Use standalone `inherit` to keep browserified files clear.

102.29 <a href="https://github.com/nodeca/js-yaml/compare/3.4.4...3.4.5" >3.4.5</a> - 2015-11-23

102.29.1 Added

- Added `lineWidth` option to dumper.

102.30 <a href="https://github.com/nodeca/js-yaml/compare/3.4.3...3.4.4" >3.4.4</a> - 2015-11-21

102.30.1 Fixed

- Fixed floats dump (missed dot for scientific format), #220.
- Allow non-printable characters inside quoted scalars, #192.

102.31 <a href="https://github.com/nodeca/js-yaml/compare/3.4.2...3.4.3" >3.4.3</a> - 2015-10-10

102.31.1 Changed

- Maintenance release - deps bump (esprima, argparse).

102.32 <a href="https://github.com/nodeca/js-yaml/compare/3.4.1...3.4.2" >3.4.2</a> - 2015-09-09

102.32.1 Fixed

- Fixed serialization of duplicated entries in sequences, #205. Thanks to @vogelsgesang.

102.33 <a href="https://github.com/nodeca/js-yaml/compare/3.4.0...3.4.1" >3.4.1</a> - 2015-09-05

102.33.1 Fixed

- Fixed stacktrace handling in generated errors, for browsers (FF/IE).

102.34 <a href="https://github.com/nodeca/js-yaml/compare/3.3.1...3.4.0" >3.4.0</a> - 2015-08-23

102.34.1 Changed

- Don't throw on warnings anymore. Use `onWarning` option to catch.
- Throw error on unknown tags (was warning before).
- Reworked internals of error class.

102.34.2 Fixed

- Fixed multiline keys dump, #197. Thanks to @tcr.
- Fixed heading line breaks in some scalars (regression).

102.35 <a href="https://github.com/nodeca/js-yaml/compare/3.3.0...3.3.1" >3.3.1</a> - 2015-05-13

102.35.1 Added

- Added `.sortKeys` dumper option, thanks to @rjmunro.

102.35.2 Fixed

- Fixed astral characters support, #191.

102.36 <a href="https://github.com/nodeca/js-yaml/compare/3.2.7...3.3.0" >3.3.0</a> - 2015-04-26

102.36.1 Changed

- Significantly improved long strings formatting in dumper, thanks to @isaacs.
- Strip BOM if exists.

102.37 <a href="https://github.com/nodeca/js-yaml/compare/3.2.6...3.2.7" >3.2.7</a> - 2015-02-19

102.37.1 Changed

- Maintenance release.
- Updated dependencies.
- HISTORY.md -> CHANGELOG.md

102.38 <a href="https://github.com/nodeca/js-yaml/compare/3.2.5...3.2.6" >3.2.6</a> - 2015-02-07

102.38.1 Fixed

- Fixed encoding of UTF-16 surrogate pairs. (e.g. "\U0001F431" CAT FACE).
- Fixed demo dates dump (#113, thanks to @Hypercubed).

102.39 <a href="https://github.com/nodeca/js-yaml/compare/3.2.4...3.2.5" >3.2.5</a> - 2014-12-28

102.39.1 Fixed

- Fixed resolving of all built-in types on empty nodes.
- Fixed invalid warning on empty lines within quoted scalars and flow collections.
- Fixed bug: Tag on an empty node didn't resolve in some cases.

102.40 <a href="https://github.com/nodeca/js-yaml/compare/3.2.3...3.2.4" >3.2.4</a> - 2014-12-19

102.40.1 Fixed

- Fixed resolving of !!null tag on an empty node.

102.41 <a href="https://github.com/nodeca/js-yaml/compare/3.2.2...3.2.3" >3.2.3</a> - 2014-11-08

102.41.1 Fixed

- Implemented dumping of objects with circular and cross references.
- Partially fixed aliasing of constructed objects. (see issue #141 for details)

102.42 <a href="https://github.com/nodeca/js-yaml/compare/3.2.1...3.2.2" >3.2.2</a> - 2014-09-07

102.42.1 Fixed

- Fixed infinite loop on unindented block scalars.
- Rewritten base64 encode/decode in binary type, to keep code licence clear.

102.43 <a href="https://github.com/nodeca/js-yaml/compare/3.2.0...3.2.1" >3.2.1</a> - 2014-08-24

102.43.1 Fixed

- Nothig new. Just fix npm publish error.

102.44 <a href="https://github.com/nodeca/js-yaml/compare/3.1.0...3.2.0" >3.2.0</a> - 2014-08-24

102.44.1 Added

- Added input piping support to CLI.

102.44.2 Fixed

- Fixed typo, that could cause hand on initial indent (#139).

102.45 <a href="https://github.com/nodeca/js-yaml/compare/3.0.2...3.1.0" >3.1.0</a> - 2014-07-07

102.45.1 Changed

- 1.5x-2x speed boost.
- Removed deprecated ‘require('xxx.yml')` support.
- Significant code cleanup and refactoring.
- Internal API changed. If you used custom types - see updated examples. Others are not affected.

- Even if the input string has no trailing line break character, it will be parsed as if it has one.
- Added benchmark scripts.
- Moved bower files to /dist folder
- Bugfixes.

102.46 <a href="https://github.com/nodeca/js-yaml/compare/3.0.1...3.0.2" >3.0.2</a> - 2014-02-27

102.46.1 Fixed

- Fixed bug: "constructor" string parsed as `null`.

102.47 <a href="https://github.com/nodeca/js-yaml/compare/3.0.0...3.0.1" >3.0.1</a> - 2013-12-22

102.47.1 Fixed

- Fixed parsing of literal scalars. (issue #108)
- Prevented adding unnecessary spaces in object dumps. (issue #68)
- Fixed dumping of objects with very long (> 1024 in length) keys.

102.48 <a href="https://github.com/nodeca/js-yaml/compare/2.1.3...3.0.0" >3.0.0</a> - 2013-12-16

102.48.1 Changed

- Refactored code. Changed API for custom types.
- Removed output colors in CLI, dump json by default.
- Removed big dependencies from browser version (esprima, buffer). Load `esprima` manually, if `!!js/function` needed. `!!bin` now returns Array in browser
- AMD support.
- Don't quote dumped strings because of `-` & `?` (if not first char).
- **Deprecated** loading yaml files via `require()`, as not recommended behaviour for node.

102.49 <a href="https://github.com/nodeca/js-yaml/compare/2.1.2...2.1.3" >2.1.3</a> - 2013-10-16

102.49.1 Fixed

- Fix wrong loading of empty block scalars.

102.50 <a href="https://github.com/nodeca/js-yaml/compare/2.1.1...2.1.2" >2.1.2</a> - 2013-10-07

102.50.1 Fixed

- Fix unwanted line breaks in folded scalars.

102.51 <a href="https://github.com/nodeca/js-yaml/compare/2.1.0...2.1.1" >2.1.1</a> - 2013-10-02

102.51.1 Fixed

- Dumper now respects deprecated booleans syntax from YAML 1.0/1.1
- Fixed reader bug in JSON-like sequences/mappings.

102.52 <a href="https://github.com/nodeca/js-yaml/compare/2.0.5...2.1.0" >2.1.0</a> - 2013-06-05

102.52.1 Added

- Add standard YAML schemas: Failsafe (`FAILSAFE_SCHEMA`), JSON (`JSON_SCHEMA`) and Core (`CORE↩ _SCHEMA`).
- Add `skipInvalid` dumper option.

102.52.2 Changed

- Rename `DEFAULT_SCHEMA` to `DEFAULT_FULL_SCHEMA` and `SAFE_SCHEMA` to `DEFAULT_SAFE↩ _SCHEMA`.
- Use `safeLoad` for `require` extension.

102.52.3 Fixed

- Bug fix: export `NIL` constant from the public interface.

102.53 <a href="https://github.com/nodeca/js-yaml/compare/2.0.4...2.0.5" >2.0.5</a> - 2013-04-26

102.53.1 Security

- Close security issue in !!js/function constructor. Big thanks to @nealpoole for security audit.

102.54 <a href="https://github.com/nodeca/js-yaml/compare/2.0.3...2.0.4" >2.0.4</a> - 2013-04-08

102.54.1 Changed

- Updated .npmignore to reduce package size

102.55 <a href="https://github.com/nodeca/js-yaml/compare/2.0.2...2.0.3" >2.0.3</a> - 2013-02-26

102.55.1 Fixed

- Fixed dumping of empty arrays ans objects. ([] and {} instead of null)

102.56 <a href="https://github.com/nodeca/js-yaml/compare/2.0.1...2.0.2" >2.0.2</a> - 2013-02-15

102.56.1 Fixed

- Fixed input validation: tabs are printable characters.

102.57 <a href="https://github.com/nodeca/js-yaml/compare/2.0.0...2.0.1" >2.0.1</a> - 2013-02-09

102.57.1 Fixed

- Fixed error, when options not passed to function cass

102.58 <a href="https://github.com/nodeca/js-yaml/compare/1.0.3...2.0.0" >2.0.0</a> - 2013-02-09

102.58.1 Changed

- Full rewrite. New architecture. Fast one-stage parsing.
- Changed custom types API.
- Added YAML dumper.

102.59 <a href="https://github.com/nodeca/js-yaml/compare/1.0.2...1.0.3" >1.0.3</a> - 2012-11-05

102.59.1 Fixed

- Fixed utf-8 files loading.

102.60 <a href="https://github.com/nodeca/js-yaml/compare/1.0.1...1.0.2" >1.0.2</a> - 2012-08-02

102.60.1 Fixed

- Pull out hand-written shims. Use ES5-Shims for old browsers support. See #44.
- Fix timstamps incorectly parsed in local time when no time part specified.

102.61 <a href="https://github.com/nodeca/js-yaml/compare/1.0.0...1.0.1" >1.0.1</a> - 2012-07-07

102.61.1 Fixed

- Fixes ‘TypeError: 'undefined’ is not an object` under Safari. Thanks Phuong.
- Fix timestamps incorrectly parsed in local time. Thanks @caolan. Closes #46.

102.62 <a href="https://github.com/nodeca/js-yaml/compare/0.3.7...1.0.0" >1.0.0</a> - 2012-07-01

102.62.1 Changed

- `y`, `yes`, `n`, `no`, `on`, `off` are not converted to Booleans anymore. Fixes #42.
- `require(filename)` now returns a single document and throws an Error if file contains more than one document.
- CLI was merged back from js-yaml.bin

102.63 <a href="https://github.com/nodeca/js-yaml/compare/0.3.6...0.3.7" >0.3.7</a> - 2012-02-28

102.63.1 Fixed

- Fix export of `addConstructor()`. Closes #39.

102.64 <a href="https://github.com/nodeca/js-yaml/compare/0.3.5...0.3.6" >0.3.6</a> - 2012-02-22

102.64.1 Changed

- Removed AMD parts - too buggy to use. Need help to rewrite from scratch

102.64.2 Fixed

- Removed YUI compressor warning (renamed `double` variable). Closes #40.

102.65 <a href="https://github.com/nodeca/js-yaml/compare/0.3.4...0.3.5" >0.3.5</a> - 2012-01-10

102.65.1 Fixed

- Workagound for .npmignore fuckup under windows. Thanks to airportyh.

102.66 <a href="https://github.com/nodeca/js-yaml/compare/0.3.3...0.3.4" >0.3.4</a> - 2011-12-24

102.66.1 Fixed

- Fixes str[] for oldIEs support.
- Adds better has change support for browserified demo.
- improves compact output of Error. Closes #33.

102.67 <a href="https://github.com/nodeca/js-yaml/compare/0.3.2...0.3.3" >0.3.3</a> - 2011-12-20

102.67.1 Added

- adds `compact` stringification of Errors.

102.67.2 Changed

- jsyaml executable moved to separate module.

102.68 <a href="https://github.com/nodeca/js-yaml/compare/0.3.1...0.3.2" >0.3.2</a> - 2011-12-16

102.68.1 Added

- Added jsyaml executable.
- Added !!js/function support. Closes #12.

102.68.2 Fixed

- Fixes ug with block style scalars. Closes #26.
- All sources are passing JSLint now.
- Fixes bug in Safari. Closes #28.
- Fixes bug in Opers. Closes #29.
- Improves browser support. Closes #20.

102.69 <a href="https://github.com/nodeca/js-yaml/compare/0.3.0...0.3.1" >0.3.1</a> - 2011-11-18

102.69.1 Added

- Added AMD support for browserified version.
- Added permalinks for online demo YAML snippets. Now we have YPaste service, lol.
- Added !!js/regexp and !!js/undefined types. Partially solves #12.

102.69.2 Changed

- Wrapped browserified js-yaml into closure.

102.69.3 Fixed

- Fixed the resolvement of non-specific tags. Closes #17.
- Fixed !!set mapping.
- Fixed month parse in dates. Closes #19.

102.70 <a href="https://github.com/nodeca/js-yaml/compare/0.2.2...0.3.0" >0.3.0</a> - 2011-11-09

102.70.1 Added

- Added browserified version. Closes #13.
- Added live demo of browserified version.
- Ported some of the PyYAML tests. See #14.

102.70.2 Fixed

- Removed JS.Class dependency. Closes #3.
- Fixed timestamp bug when fraction was given.

102.71 <a href="https://github.com/nodeca/js-yaml/compare/0.2.1...0.2.2" >0.2.2</a> - 2011-11-06

102.71.1 Fixed

- Fixed crash on docs without —. Closes #8.
- Fixed multiline string parse
- Fixed tests/comments for using array as key

102.72 <a href="https://github.com/nodeca/js-yaml/compare/0.2.0...0.2.1" >0.2.1</a> - 2011-11-02

102.72.1 Fixed

- Fixed short file read (<4k). Closes #9.

102.73 <a href="https://github.com/nodeca/js-yaml/releases/tag/0.2.0" >0.2.0</a> - 2011-11-02

102.73.1 Changed

- First public release

Chapter 103

JS-YAML - YAML 1.2 parser / writer for JavaScript

Online Demo
This is an implementation of YAML, a human-friendly data serialization language. Started as PyYAML port, it was completely rewritten from scratch. Now it's very fast, and supports 1.2 spec.

103.1 Installation

103.1.1 YAML module for node.js

```
npm install js-yaml
```

103.1.2 CLI executable

If you want to inspect your YAML files from CLI, install js-yaml globally:

```
npm install -g js-yaml
```

103.1.2.1 Usage

```
usage:  js-yaml [-h] [-v] [-c] [-t] file
Positional arguments:
  file           File with YAML document(s)
Optional arguments:
  -h, --help     Show this help message and exit.
  -v, --version  Show program's version number and exit.
  -c, --compact  Display errors in compact mode
  -t, --trace    Show stack trace on error
```

103.2 API

Here we cover the most 'useful' methods. If you need advanced details (creating your own tags), see examples for more info.

```
const yaml = require('js-yaml');
const fs   = require('fs');
// Get document, or throw exception on error
try {
  const doc = yaml.load(fs.readFileSync('/home/ixti/example.yml', 'utf8'));
  console.log(doc);
} catch (e) {
  console.log(e);
}
```

103.2.1 load (string [, options])

Parses `string` as single YAML document. Returns either a plain object, a string, a number, `null` or `undefined`, or throws `YAMLException` on error. By default, does not support regexps, functions and undefined.
options:

- `filename` _(default: null)_ - string to be used as a file path in error/warning messages.

- `onWarning` _(default: null)_ - function to call on warning messages. Loader will call this function with an instance of `YAMLException` for each warning.
- `schema` _(default: `DEFAULT_SCHEMA`)_ - specifies a schema to use.
  - `FAILSAFE_SCHEMA` - only strings, arrays and plain objects: http://www.yaml.org/spec/1.2/spec.html#id2802346
  - `JSON_SCHEMA` - all JSON-supported types: http://www.yaml.org/spec/1.2/spec.html#id2803231
  - `CORE_SCHEMA` - same as `JSON_SCHEMA`: http://www.yaml.org/spec/1.2/spec.html#id2804923
  - `DEFAULT_SCHEMA` - all supported YAML types.
- `json` _(default: false)_ - compatibility with JSON.parse behaviour. If true, then duplicate keys in a mapping will override values rather than throwing an error.

NOTE: This function **does not** understand multi-document sources, it throws exception on those.
NOTE: JS-YAML **does not** support schema-specific tag resolution restrictions. So, the JSON schema is not as strictly defined in the YAML specification. It allows numbers in any notation, use `Null` and `NULL` as `null`, etc. The core schema also has no such restrictions. It allows binary notation for integers.

103.2.2 loadAll (string [, iterator] [, options])

Same as `load()`, but understands multi-document sources. Applies `iterator` to each document if specified, or returns array of documents.

```
const yaml = require('js-yaml');
yaml.loadAll(data, function (doc) {
  console.log(doc);
});
```

103.2.3 dump (object [, options])

Serializes `object` as a YAML document. Uses `DEFAULT_SCHEMA`, so it will throw an exception if you try to dump regexps or functions. However, you can disable exceptions by setting the `skipInvalid` option to `true`.
options:

- `indent` _(default: 2)_ - indentation width to use (in spaces).
- `noArrayIndent` _(default: false)_ - when true, will not add an indentation level to array elements
- `skipInvalid` _(default: false)_ - do not throw on invalid types (like function in the safe schema) and skip pairs and single values with such types.
- `flowLevel` _(default: -1)_ - specifies level of nesting, when to switch from block to flow style for collections. -1 means block style everwhere
- `styles` - "tag" => "style" map. Each tag may have own set of styles.
- `schema` _(default: `DEFAULT_SCHEMA`)_ specifies a schema to use.
- `sortKeys` _(default: `false`)_ - if `true`, sort keys when dumping YAML. If a function, use the function to sort the keys.
- `lineWidth` _(default: `80`)_ - set max line width. Set `-1` for unlimited width.
- `noRefs` _(default: `false`)_ - if `true`, don't convert duplicate objects into references
- `noCompatMode` _(default: `false`)_ - if `true` don't try to be compatible with older yaml versions. Currently: don't quote "yes", "no" and so on, as required for YAML 1.1

- condenseFlow _(default: false)_ - if true flow sequences will be condensed, omitting the space between a, b. Eg. '[a,b]', and omitting the space betweenkey: valueand quoting the key. Eg.'{"a":b}'Can be useful when using yaml for pretty URL query params as spaces are %-encoded. -quotingType_('or"<tt>, default:</tt>'<tt>)_ - strings will be quoted using this quoting style. If you specify single quotes, double quotes will still be used for non-printable characters. -</tt>forceQuotes<tt>_(default:</tt>false<tt>)_ - if</tt>true<tt>, all non-key strings will be quoted even if they normally don't need to. -</tt>replacer<tt>- callback</tt>function (key, value)<tt>called recursively on each key/value in source object (see</tt>replacer<tt>docs for</tt>JSON.stringify`). The following table show availlable styles (e.g. "canonical", "binary"...) available for each tag (.e.g. !!null, !!int ...). Yaml output is shown on the right side after <tt>=\></tt> (default setting) or <tt>-\></tt>: @code !!null "canonical" -> "~" "lowercase" => "null" "uppercase" -> "NULL" "camelcase" -> "Null" !!int "binary" -> "0b1", "0b101010", "0b1110001111010" "octal" -> "0o1", "0o52", "0o16172" "decimal" => "1", "42", "7290" "hexadecimal" -> "0x1", "0x2A", "0x1C7A" !!bool "lowercase" => "true", "false" "uppercase" -> "TRUE", "FALSE" "camelcase" -> "True", "False" !!float "lowercase" => ".nan", '.inf' "uppercase" -> ".NAN", '.INF' "camelcase" -> ".NaN", '.Inf' @endcode Example: @code dump(object, { 'styles': { '!!null': 'canonical' // dump null as ~ }, 'sortKeys': true // sort object keys }); @endcode @section autotoc_md1650 Supported YAML types The list of standard YAML tags and corresponding JavaScript types. See also <a href=" http://pyyaml.org/wiki/YAMLTagDiscussion" >YAML tag discussion</a> and <a href=" http://yaml.org/type/" >YAML types repository</a>. @code !!null " # null !!bool 'yes' # bool !!int '3...' # number !!float '3.14...' # number !!binary '...base64...' # buffer !!timestamp 'YYYY-...' # date !!omap [...] # array of key-value pairs !!pairs [...] # array or array pairs !!set { ... } # array of objects with given keys and null values !!str '...' # string !!seq [...] # array !!map { ... } # object @endcode <strong>JavaScript-specific tags</strong> See <a href=" https://github.com/nodeca/js-yaml-js-types" >js-yaml-js-types</a> for extra types. @section autotoc_md1651 Caveats Note, that you use arrays or objects as key in JS-YAML. JS does not allow objects or arrays as keys, and stringifies (by calling <tt>to↩
String()</tt> method) them at the moment of adding them. @code --- ? [foo, bar] : - baz ? { foo: bar } : - baz - baz @endcode @code { "foo,bar": ["baz"], "[object Object]": ["baz", "baz"] } @endcode Also, reading of properties on implicit block mapping keys is not supported yet. So, the following YAML document cannot be loaded. @code &anchor foo: foo: bar *anchor: duplicate key baz: bat *anchor: duplicate key @endcode @section autotoc_md1652 js-yaml for enterprise Available as part of the Tidelift Subscription The maintainers of js-yaml and thousands of other packages are working with Tidelift to deliver commercial support and maintenance for the open source dependencies you use to build your applications. Save time, reduce risk, and improve code health, while paying the maintainers of the exact dependencies you use. <a href=" https://tidelift.com/subscription/pkg/npm-js-yaml?utm↩
_source=npm-js-yaml&utm_medium=referral&utm_campaign=enterprise&utm_↩
term=repo" >Learn more.

Chapter 104

lodash.get v4.4.2

The `lodash` method `_.get` exported as a `Node.js` module.

104.1 Installation

Using npm:

```
$ {sudo -H} npm i -g npm
$ npm i --save lodash.get
```

In Node.js:

```
var get = require('lodash.get');
```

See the `documentation` or `package source` for more details.

Chapter 105

lodash.isequal v4.5.0

The `Lodash` method `_.isEqual` exported as a `Node.js` module.

105.1 Installation

Using npm:

```
$ {sudo -H} npm i -g npm
$ npm i --save lodash.isequal
```

In Node.js:

```
var isEqual = require('lodash.isequal');
```

See the `documentation` or `package source` for more details.

Chapter 106

lodash.mergewith v4.6.2

The Lodash method _.mergeWith exported as a Node.js module.

106.1 Installation

Using npm:

```
$ {sudo -H} npm i -g npm
$ npm i --save lodash.mergewith
```

In Node.js:

```
var mergeWith = require('lodash.mergewith');
```

See the documentation or package source for more details.

Chapter 107

long.js

A Long class for representing a 64 bit two's-complement integer value derived from the Closure Library for stand-alone use and extended with unsigned support.

107.1 Background

As of ECMA-262 5th Edition, "all the positive and negative integers whose magnitude is no greater than 2<sup>53</sup> are representable in the Number type", which is "representing the doubleprecision 64-bit format IEEE 754 values as specified in the IEEE Standard for Binary Floating-Point Arithmetic". The maximum safe integer in JavaScript is 2^{53}-1.
Example: 2^{64}-1 is 18446744073709551615 but in JavaScript it evaluates to 18446744073709552000.
Furthermore, bitwise operators in JavaScript "deal only with integers in the range 2<sup>31</sup> through 2<sup>31</sup>1, inclusive, or in the range 0 through 2<sup>32</sup>1, inclusive. These operators accept any value of the Number type but first convert each such value to one of 2<sup>32</sup> integer values."
In some use cases, however, it is required to be able to reliably work with and perform bitwise operations on the full 64 bits. This is where long.js comes into play.

107.2 Usage

The package exports an ECMAScript module with an UMD fallback.

```
$> npm install long
import Long from "long";
var value = new Long(0xFFFFFFFF, 0x7FFFFFFF);
console.log(value.toString());
...
```

Note that mixing ESM and CommonJS is not recommended as it yields different classes, albeit with the same functionality.

107.2.1 Usage with a CDN

- From GitHub via jsDelivr:
  https://cdn.jsdelivr.net/gh/dcodeIO/long.js@TAG/index.js (ESM)
- From npm via jsDelivr:
  https://cdn.jsdelivr.net/npm/long@VERSION/index.js (ESM)
  https://cdn.jsdelivr.net/npm/long@VERSION/umd/index.js (UMD)
- From npm via unpkg:
  https://unpkg.com/long@VERSION/index.js (ESM)
  https://unpkg.com/long@VERSION/umd/index.js (UMD)

Replace `TAG` respectively `VERSION` with a specific version or omit it (not recommended in production) to use main/latest.

107.3 API

107.3.1 Constructor

- new **Long**(low: `number`, high?: `number`, unsigned?: `boolean`)
  Constructs a 64 bit two's-complement integer, given its low and high 32 bit values as *signed* integers. See the from∗ functions below for more convenient ways of constructing Longs.

107.3.2 Fields

- Long#∗∗low∗∗: `number`
  The low 32 bits as a signed value.

- Long#∗∗high∗∗: `number`
  The high 32 bits as a signed value.

- Long#∗∗unsigned∗∗: `boolean`
  Whether unsigned or not.

107.3.3 Constants

- Long.∗∗ZERO∗∗: `Long`
  Signed zero.

- Long.∗∗ONE∗∗: `Long`
  Signed one.

- Long.∗∗NEG_ONE∗∗: `Long`
  Signed negative one.

- Long.∗∗UZERO∗∗: `Long`
  Unsigned zero.

- Long.∗∗UONE∗∗: `Long`
  Unsigned one.

- Long.∗∗MAX_VALUE∗∗: `Long`
  Maximum signed value.

- Long.∗∗MIN_VALUE∗∗: `Long`
  Minimum signed value.

- Long.∗∗MAX_UNSIGNED_VALUE∗∗: `Long`
  Maximum unsigned value.

107.3.4 Utility

- Long.∗∗isLong∗∗(obj: ∗): `boolean`
  Tests if the specified object is a Long.

- Long.∗∗fromBits∗∗(lowBits: `number`, highBits: `number`, unsigned?: `boolean`): `Long`
  Returns a Long representing the 64 bit integer that comes by concatenating the given low and high bits. Each is assumed to use 32 bits.

- Long.∗∗fromBytes∗∗(bytes: `number[]`, unsigned?: `boolean`, le?: `boolean`): `Long`
  Creates a Long from its byte representation.

- Long.∗∗fromBytesLE∗∗(bytes: `number[]`, unsigned?: `boolean`): `Long`
  Creates a Long from its little endian byte representation.

- Long.∗∗fromBytesBE∗∗(bytes: `number[]`, unsigned?: `boolean`): `Long`
  Creates a Long from its big endian byte representation.

- Long.∗∗fromInt∗∗(value: `number`, unsigned?: `boolean`): `Long`
  Returns a Long representing the given 32 bit integer value.
- Long.∗∗fromNumber∗∗(value: `number`, unsigned?: `boolean`): `Long`
  Returns a Long representing the given value, provided that it is a finite number. Otherwise, zero is returned.
- Long.∗∗fromString∗∗(str: `string`, unsigned?: `boolean`, radix?: `number`)
  Long.∗∗fromString∗∗(str: `string`, radix: `number`)
  Returns a Long representation of the given string, written using the specified radix.
- Long.∗∗fromValue∗∗(val: ∗, unsigned?: `boolean`): `Long`
  Converts the specified value to a Long using the appropriate from∗ function for its type.

107.3.5 Methods

- Long#∗∗add∗∗(addend: `Long | number | string`): `Long`
  Returns the sum of this and the specified Long.
- Long#∗∗and∗∗(other: `Long | number | string`): `Long`
  Returns the bitwise AND of this Long and the specified.
- Long#∗∗compare∗∗/∗∗comp∗∗(other: `Long | number | string`): `number`
  Compares this Long's value with the specified's. Returns `0` if they are the same, `1` if the this is greater and `-1` if the given one is greater.
- Long#∗∗divide∗∗/∗∗div∗∗(divisor: `Long | number | string`): `Long`
  Returns this Long divided by the specified.
- Long#∗∗equals∗∗/∗∗eq∗∗(other: `Long | number | string`): `boolean`
  Tests if this Long's value equals the specified's.
- Long#∗∗getHighBits∗∗(): `number`
  Gets the high 32 bits as a signed integer.
- Long#∗∗getHighBitsUnsigned∗∗(): `number`
  Gets the high 32 bits as an unsigned integer.
- Long#∗∗getLowBits∗∗(): `number`
  Gets the low 32 bits as a signed integer.
- Long#∗∗getLowBitsUnsigned∗∗(): `number`
  Gets the low 32 bits as an unsigned integer.
- Long#∗∗getNumBitsAbs∗∗(): `number`
  Gets the number of bits needed to represent the absolute value of this Long.
- Long#∗∗greaterThan∗∗/∗∗gt∗∗(other: `Long | number | string`): `boolean`
  Tests if this Long's value is greater than the specified's.
- Long#∗∗greaterThanOrEqual∗∗/∗∗gte∗∗/∗∗ge∗∗(other: `Long | number | string`): `boolean`
  Tests if this Long's value is greater than or equal the specified's.
- Long#∗∗isEven∗∗(): `boolean`
  Tests if this Long's value is even.
- Long#∗∗isNegative∗∗(): `boolean`
  Tests if this Long's value is negative.
- Long#∗∗isOdd∗∗(): `boolean`
  Tests if this Long's value is odd.
- Long#∗∗isPositive∗∗(): `boolean`
  Tests if this Long's value is positive or zero.

- Long#**isZero**/**eqz**(): `boolean`
  Tests if this Long's value equals zero.
- Long#**lessThan**/**lt**(other: `Long | number | string`): `boolean`
  Tests if this Long's value is less than the specified's.
- Long#**lessThanOrEqual**/**lte**/**le**(other: `Long | number | string`): `boolean`
  Tests if this Long's value is less than or equal the specified's.
- Long#**modulo**/**mod**/**rem**(divisor: `Long | number | string`): `Long`
  Returns this Long modulo the specified.
- Long#**multiply**/**mul**(multiplier: `Long | number | string`): `Long`
  Returns the product of this and the specified Long.
- Long#**negate**/**neg**(): `Long`
  Negates this Long's value.
- Long#**not**(): `Long`
  Returns the bitwise NOT of this Long.
- Long#**countLeadingZeros**/**clz**(): `number`
  Returns count leading zeros of this Long.
- Long#**countTrailingZeros**/**ctz**(): `number`
  Returns count trailing zeros of this Long.
- Long#**notEquals**/**neq**/**ne**(other: `Long | number | string`): `boolean`
  Tests if this Long's value differs from the specified's.
- Long#**or**(other: `Long | number | string`): `Long`
  Returns the bitwise OR of this Long and the specified.
- Long#**shiftLeft**/**shl**(numBits: `Long | number | string`): `Long`
  Returns this Long with bits shifted to the left by the given amount.
- Long#**shiftRight**/**shr**(numBits: `Long | number | string`): `Long`
  Returns this Long with bits arithmetically shifted to the right by the given amount.
- Long#**shiftRightUnsigned**/**shru**/**shr_u**(numBits: `Long | number | string`): `Long`
  Returns this Long with bits logically shifted to the right by the given amount.
- Long#**rotateLeft**/**rotl**(numBits: `Long | number | string`): `Long`
  Returns this Long with bits rotated to the left by the given amount.
- Long#**rotateRight**/**rotr**(numBits: `Long | number | string`): `Long`
  Returns this Long with bits rotated to the right by the given amount.
- Long#**subtract**/**sub**(subtrahend: `Long | number | string`): `Long`
  Returns the difference of this and the specified Long.
- Long#**toBytes**(le?: `boolean`): `number[]`
  Converts this Long to its byte representation.
- Long#**toBytesLE**(): `number[]`
  Converts this Long to its little endian byte representation.
- Long#**toBytesBE**(): `number[]`
  Converts this Long to its big endian byte representation.
- Long#**toInt**(): `number`
  Converts the Long to a 32 bit integer, assuming it is a 32 bit integer.
- Long#**toNumber**(): `number`
  Converts the Long to a the nearest floating-point representation of this value (double, 53 bit mantissa).

- Long#**toSigned**(): `Long`
  Converts this Long to signed.
- Long#**toString**(radix?: `number`): `string`
  Converts the Long to a string written in the specified radix.
- Long#**toUnsigned**(): `Long`
  Converts this Long to unsigned.
- Long#**xor**(other: `Long | number | string`): `Long`
  Returns the bitwise XOR of this Long and the given one.

107.4 WebAssembly support

`WebAssembly` supports 64-bit integer arithmetic out of the box, hence a `tiny WebAssembly module` is used to compute operations like multiplication, division and remainder more efficiently (slow operations like division are around twice as fast), falling back to floating point based computations in JavaScript where WebAssembly is not yet supported, e.g., in older versions of node.

107.5 Building

Building the UMD fallback:

```
$> npm run build
```

Running the `tests`:

```
$> npm test
```

Chapter 108

lru-cache

A cache object that deletes the least-recently-used items.
Specify a max number of the most recently used items that you want to keep, and this cache will keep that many of the most recently accessed items.
This is not primarily a TTL cache, and does not make strong TTL guarantees. There is no preemptive pruning of expired items by default, but you *may* set a TTL on the cache or on a single `set`. If you do so, it will treat expired items as missing, and delete them when fetched. If you are more interested in TTL caching than LRU caching, check out `@isaacs/ttlcache`.
As of version 7, this is one of the most performant LRU implementations available in JavaScript, and supports a wide diversity of use cases. However, note that using some of the features will necessarily impact performance, by causing the cache to have to do more work. See the "Performance" section below.

108.1 Installation

```
npm install lru-cache --save
```

108.2 Usage

```
// hybrid module, either works
import LRUCache from 'lru-cache'
// or:
const LRUCache = require('lru-cache')
// At least one of 'max', 'ttl', or 'maxSize' is required, to prevent
// unsafe unbounded storage.
//
// In most cases, it's best to specify a max for performance, so all
// the required memory allocation is done up-front.
//
// All the other options are optional, see the sections below for
// documentation on what each one does.   Most of them can be
// overridden for specific items in get()/set()
const options = {
  max:  500,
  // for use with tracking overall storage size
  maxSize:  5000,
  sizeCalculation:  (value, key) => {
    return 1
  },
  // for use when you need to clean up something when objects
  // are evicted from the cache
  dispose:  (value, key) => {
    freeFromMemoryOrWhatever(value)
  },
  // how long to live in ms
  ttl:  1000 * 60 * 5,
  // return stale items before removing from cache?
  allowStale:  false,
  updateAgeOnGet:  false,
  updateAgeOnHas:  false,
  // async method to use for cache.fetch(), for
  // stale-while-revalidate type of behavior
  fetchMethod:  async (key, staleValue, { options, signal }) => {},
}
const cache = new LRUCache(options)
cache.set('key', 'value')
cache.get('key') // "value"
```

```
// non-string keys ARE fully supported
// but note that it must be THE SAME object, not
// just a JSON-equivalent object.
var someObject = { a:  1 }
cache.set(someObject, 'a value')
// Object keys are not toString()-ed
cache.set('[object Object]', 'a different value')
assert.equal(cache.get(someObject), 'a value')
// A similar object with same keys/values won't work,
// because it's a different object identity
assert.equal(cache.get({ a:  1 }), undefined)
cache.clear() // empty the cache
```

If you put more stuff in it, then items will fall out.

108.3 Options

108.3.1 <tt>max</tt>

The maximum number of items that remain in the cache (assuming no TTL pruning or explicit deletions). Note that fewer items may be stored if size calculation is used, and `maxSize` is exceeded. This must be a positive finite intger.
At least one of `max`, `maxSize`, or `TTL` is required. This must be a positive integer if set.
It is strongly recommended to set a `max` to prevent unbounded growth of the cache. See "Storage Bounds Safety" below.

108.3.2 <tt>maxSize</tt>

Set to a positive integer to track the sizes of items added to the cache, and automatically evict items in order to stay below this size. Note that this may result in fewer than `max` items being stored.
Attempting to add an item to the cache whose calculated size is greater that this amount will be a no-op. The item will not be cached, and no other items will be evicted.
Optional, must be a positive integer if provided.
Sets `maxEntrySize` to the same value, unless a different value is provided for `maxEntrySize`.
At least one of `max`, `maxSize`, or `TTL` is required. This must be a positive integer if set.
Even if size tracking is enabled, **it is strongly recommended to set a `max` to prevent unbounded growth of the cache.** See "Storage Bounds Safety" below.

108.3.3 <tt>maxEntrySize</tt>

Set to a positive integer to track the sizes of items added to the cache, and prevent caching any item over a given size. Attempting to add an item whose calculated size is greater than this amount will be a no-op. The item will not be cached, and no other items will be evicted.
Optional, must be a positive integer if provided. Defaults to the value of `maxSize` if provided.

108.3.4 <tt>sizeCalculation</tt>

Function used to calculate the size of stored items. If you're storing strings or buffers, then you probably want to do something like `n => n.length`. The item is passed as the first argument, and the key is passed as the second argument.
This may be overridden by passing an options object to `cache.set()`.
Requires `maxSize` to be set.
If the `size` (or return value of `sizeCalculation`) for a given entry is greater than `maxEntrySize`, then the item will not be added to the cache.
Deprecated alias: `length`

108.3.5 <tt>fetchMethod</tt>

Function that is used to make background asynchronous fetches. Called with `fetchMethod(key, stale↩ Value, { signal, options, context })`. May return a Promise.
If `fetchMethod` is not provided, then `cache.fetch(key)` is equivalent to `Promise.resolve(cache.↩ get(key))`.

The `signal` object is an `AbortSignal` if that's available in the global object, otherwise it's a pretty close polyfill.
If at any time, `signal.aborted` is set to `true`, or if the `signal.onabort` method is called, or if it emits an "abort'`event which you can listen to with`addEventListener`, then that means that the fetch should be abandoned. This may be passed along to async functions aware of AbortController/AbortSignal behavior.
The `fetchMethod` should **only** return `undefined` or a Promise resolving to `undefined` if the AbortController signaled an `abort` event. In all other cases, it should return or resolve to a value suitable for adding to the cache.
The `options` object is a union of the options that may be provided to `set()` and `get()`. If they are modified, then that will result in modifying the settings to `cache.set()` when the value is resolved, and in the case of `noDeleteOnFetchRejection` and `allowStaleOnFetchRejection`, the handling of `fetchMethod` failures.
For example, a DNS cache may update the TTL based on the value returned from a remote DNS server by changing `options.ttl` in the `fetchMethod`.

108.3.6 <tt>fetchContext</tt>

Arbitrary data that can be passed to the `fetchMethod` as the `context` option.
Note that this will only be relevant when the `cache.fetch()` call needs to call `fetchMethod()`. Thus, any data which will meaningfully vary the fetch response needs to be present in the key. This is primarily intended for including `x-request-id` headers and the like for debugging purposes, which do not affect the `fetch↩ Method()` response.

108.3.7 <tt>noDeleteOnFetchRejection</tt>

If a `fetchMethod` throws an error or returns a rejected promise, then by default, any existing stale value will be removed from the cache.
If `noDeleteOnFetchRejection` is set to `true`, then this behavior is suppressed, and the stale value remains in the cache in the case of a rejected `fetchMethod`.
This is important in cases where a `fetchMethod` is *only* called as a background update while the stale value is returned, when `allowStale` is used.
This is implicitly in effect when `allowStaleOnFetchRejection` is set.
This may be set in calls to `fetch()`, or defaulted on the constructor, or overridden by modifying the options object in the `fetchMethod`.

108.3.8 <tt>allowStaleOnFetchRejection</tt>

Set to true to return a stale value from the cache when a `fetchMethod` throws an error or returns a rejected Promise.
If a `fetchMethod` fails, and there is no stale value available, the `fetch()` will resolve to `undefined`. Ie, all `fetchMethod` errors are suppressed.
Implies `noDeleteOnFetchRejection`.
This may be set in calls to `fetch()`, or defaulted on the constructor, or overridden by modifying the options object in the `fetchMethod`.

108.3.9 <tt>allowStaleOnFetchAbort</tt>

Set to true to return a stale value from the cache when the `AbortSignal` passed to the `fetchMethod` dispatches an "abort"` event, whether user-triggered, or due to internal cache behavior.
Unless `ignoreFetchAbort` is also set, the underlying `fetchMethod` will still be considered canceled, and its return value will be ignored and not cached.

108.3.10 <tt>ignoreFetchAbort</tt>

Set to true to ignore the `abort` event emitted by the `AbortSignal` object passed to `fetchMethod`, and still cache the resulting resolution value, as long as it is not `undefined`.
When used on its own, this means aborted `fetch()` calls are not immediately resolved or rejected when they are aborted, and instead take the full time to await.
When used with `allowStaleOnFetchAbort`, aborted `fetch()` calls will resolve immediately to their stale cached value or `undefined`, and will continue to process and eventually update the cache when they resolve,

as long as the resulting value is not `undefined`, thus supporting a "return stale on timeout while refreshing" mechanism by passing `AbortSignal.timeout(n)` as the signal.
For example:
```
const c = new LRUCache({
  ttl:  100,
  ignoreFetchAbort:  true,
  allowStaleOnFetchAbort:  true,
  fetchMethod:  async (key, oldValue, { signal }) => {
    // note:  do NOT pass the signal to fetch()!
    // let's say this fetch can take a long time.
    const res = await fetch(`https://slow-backend-server/${key}`)
    return await res.json()
  },
})
// this will return the stale value after 100ms, while still
// updating in the background for next time.
const val = await c.fetch('key', { signal:  AbortSignal.timeout(100) })
```
Note: regardless of this setting, an `abort` event *is still emitted on the `AbortSignal` object*, so may result in invalid results when passed to other underlying APIs that use AbortSignals.
This may be overridden on the `fetch()` call or in the `fetchMethod` itself.

108.3.11 <tt>dispose</tt>

Function that is called on items when they are dropped from the cache, as `this.dispose(value, key, reason)`.
This can be handy if you want to close file descriptors or do other cleanup tasks when items are no longer stored in the cache.
NOTE: It is called *before* the item has been fully removed from the cache, so if you want to put it right back in, you need to wait until the next tick. If you try to add it back in during the `dispose()` function call, it will break things in subtle and weird ways.
Unlike several other options, this may *not* be overridden by passing an option to `set()`, for performance reasons. If disposal functions may vary between cache entries, then the entire list must be scanned on every cache swap, even if no disposal function is in use.
The `reason` will be one of the following strings, corresponding to the reason for the item's deletion:

- `evict` Item was evicted to make space for a new addition
- `set` Item was overwritten by a new value
- `delete` Item was removed by explicit `cache.delete(key)` or by calling `cache.clear()`, which deletes everything.

The `dispose()` method is *not* called for canceled calls to `fetchMethod()`. If you wish to handle evictions, overwrites, and deletes of in-flight asynchronous fetches, you must use the `AbortSignal` provided.
Optional, must be a function.

108.3.12 <tt>disposeAfter</tt>

The same as `dispose`, but called *after* the entry is completely removed and the cache is once again in a clean state.
It is safe to add an item right back into the cache at this point. However, note that it is *very* easy to inadvertently create infinite recursion in this way.
The `disposeAfter()` method is *not* called for canceled calls to `fetchMethod()`. If you wish to handle evictions, overwrites, and deletes of in-flight asynchronous fetches, you must use the `AbortSignal` provided.

108.3.13 <tt>noDisposeOnSet</tt>

Set to `true` to suppress calling the `dispose()` function if the entry key is still accessible within the cache.
This may be overridden by passing an options object to `cache.set()`.
Boolean, default `false`. Only relevant if `dispose` or `disposeAfter` options are set.

108.3.14 <tt>ttl</tt>

Max time to live for items before they are considered stale. Note that stale items are NOT preemptively removed by default, and MAY live in the cache, contributing to its LRU max, long after they have expired.

Also, as this cache is optimized for LRU/MRU operations, some of the staleness/TTL checks will reduce performance.
This is not primarily a TTL cache, and does not make strong TTL guarantees. There is no pre-emptive pruning of expired items, but you *may* set a TTL on the cache, and it will treat expired items as missing when they are fetched, and delete them.
Optional, but must be a positive integer in ms if specified.
This may be overridden by passing an options object to `cache.set()`.
At least one of `max`, `maxSize`, or `TTL` is required. This must be a positive integer if set.
Even if ttl tracking is enabled, **it is strongly recommended to set a `max` to prevent unbounded growth of the cache.** See "Storage Bounds Safety" below.
If ttl tracking is enabled, and `max` and `maxSize` are not set, and `ttlAutopurge` is not set, then a warning will be emitted cautioning about the potential for unbounded memory consumption.
Deprecated alias: `maxAge`

108.3.15 <tt>noUpdateTTL</tt>

Boolean flag to tell the cache to not update the TTL when setting a new value for an existing key (ie, when updating a value rather than inserting a new value). Note that the TTL value is *always* set (if provided) when adding a new entry into the cache.
This may be passed as an option to `cache.set()`.
Boolean, default false.

108.3.16 <tt>ttlResolution</tt>

Minimum amount of time in ms in which to check for staleness. Defaults to `1`, which means that the current time is checked at most once per millisecond.
Set to `0` to check the current time every time staleness is tested.
Note that setting this to a higher value *will* improve performance somewhat while using ttl tracking, albeit at the expense of keeping stale items around a bit longer than intended.

108.3.17 <tt>ttlAutopurge</tt>

Preemptively remove stale items from the cache.
Note that this may *significantly* degrade performance, especially if the cache is storing a large number of items. It is almost always best to just leave the stale items in the cache, and let them fall out as new items are added.
Note that this means that `allowStale` is a bit pointless, as stale items will be deleted almost as soon as they expire.
Use with caution!
Boolean, default `false`

108.3.18 <tt>allowStale</tt>

By default, if you set `ttl`, it'll only delete stale items from the cache when you `get(key)`. That is, it's not preemptively pruning items.
If you set `allowStale:true`, it'll return the stale value as well as deleting it. If you don't set this, then it'll return `undefined` when you try to get a stale entry.
Note that when a stale entry is fetched, *even if it is returned due to `allowStale` being set*, it is removed from the cache immediately. You can immediately put it back in the cache if you wish, thus resetting the TTL.
This may be overridden by passing an options object to `cache.get()`. The `cache.has()` method will always return `false` for stale items.
Boolean, default false, only relevant if `ttl` is set.
Deprecated alias: `stale`

108.3.19 <tt>noDeleteOnStaleGet</tt>

When using time-expiring entries with `ttl`, by default stale items will be removed from the cache when the key is accessed with `cache.get()`.

Setting `noDeleteOnStaleGet` to `true` will cause stale items to remain in the cache, until they are explicitly deleted with `cache.delete(key)`, or retrieved with `noDeleteOnStaleGet` set to `false`.
This may be overridden by passing an options object to `cache.get()`.
Boolean, default false, only relevant if `ttl` is set.

108.3.20 <tt>updateAgeOnGet</tt>

When using time-expiring entries with `ttl`, setting this to `true` will make each item's age reset to 0 whenever it is retrieved from cache with `get()`, causing it to not expire. (It can still fall out of cache based on recency of use, of course.)
This may be overridden by passing an options object to `cache.get()`.
Boolean, default false, only relevant if `ttl` is set.

108.3.21 <tt>updateAgeOnHas</tt>

When using time-expiring entries with `ttl`, setting this to `true` will make each item's age reset to 0 whenever its presence in the cache is checked with `has()`, causing it to not expire. (It can still fall out of cache based on recency of use, of course.)
This may be overridden by passing an options object to `cache.has()`.
Boolean, default false, only relevant if `ttl` is set.

108.4 API

108.4.1 <tt>new LRUCache(options)</tt>

Create a new LRUCache. All options are documented above, and are on the cache as public members.

108.4.2 <tt>cache.max</tt>, <tt>cache.maxSize</tt>, <tt>cache.allowStale</tt>,

`cache.noDisposeOnSet`, `cache.sizeCalculation`, `cache.dispose`, `cache.maxSize`, `cache.ttl`, `cache.updateAgeOnGet`, `cache.updateAgeOnHas`
All option names are exposed as public members on the cache object.
These are intended for read access only. Changing them during program operation can cause undefined behavior.

108.4.3 <tt>cache.size</tt>

The total number of items held in the cache at the current moment.

108.4.4 <tt>cache.calculatedSize</tt>

The total size of items in cache when using size tracking.

108.4.5 <tt>set(key, value, [{ size, sizeCalculation, ttl, noDisposeOnSet, start, status }])</tt>

Add a value to the cache.
Optional options object may contain `ttl` and `sizeCalculation` as described above, which default to the settings on the cache object.
If `start` is provided, then that will set the effective start time for the TTL calculation. Note that this must be a previous value of `performance.now()` if supported, or a previous value of `Date.now()` if not.
Options object may also include `size`, which will prevent calling the `sizeCalculation` function and just use the specified number if it is a positive integer, and `noDisposeOnSet` which will prevent calling a `dispose` function in the case of overwrites.
If the `size` (or return value of `sizeCalculation`) for a given entry is greater than `maxEntrySize`, then the item will not be added to the cache.
Will update the recency of the entry.
Returns the cache object.
For the usage of the `status` option, see **Status Tracking** below.

108.4.6 <tt>get(key, { updateAgeOnGet, allowStale, status } = {}) => value</tt>

Return a value from the cache.
Will update the recency of the cache entry found.
If the key is not found, `get()` will return `undefined`. This can be confusing when setting values specifically to `undefined`, as in `cache.set(key, undefined)`. Use `cache.has()` to determine whether a key is present in the cache at all.
For the usage of the `status` option, see **Status Tracking** below.

108.4.7 <tt>async fetch(key, options = {}) => Promise</tt>

The following options are supported:

- `updateAgeOnGet`
- `allowStale`
- `size`
- `sizeCalculation`
- `ttl`
- `noDisposeOnSet`
- `forceRefresh`
- `status` - See **Status Tracking** below.
- `signal` - AbortSignal can be used to cancel the `fetch()`. Note that the `signal` option provided to the `fetchMethod` is a different object, because it must also respond to internal cache state changes, but aborting this signal will abort the one passed to `fetchMethod` as well.
- `fetchContext` - sets the `context` option passed to the underlying `fetchMethod`.

If the value is in the cache and not stale, then the returned Promise resolves to the value.
If not in the cache, or beyond its TTL staleness, then `fetchMethod(key, staleValue, { options, signal, context })` is called, and the value returned will be added to the cache once resolved.
If called with `allowStale`, and an asynchronous fetch is currently in progress to reload a stale value, then the former stale value will be returned.
If called with `forceRefresh`, then the cached item will be re-fetched, even if it is not stale. However, if `allowStale` is set, then the old value will still be returned. This is useful in cases where you want to force a reload of a cached value. If a background fetch is already in progress, then `forceRefresh` has no effect.
Multiple fetches for the same `key` will only call `fetchMethod` a single time, and all will be resolved when the value is resolved, even if different options are used.
If `fetchMethod` is not specified, then this is effectively an alias for `Promise.resolve(cache.get(key))`.
When the fetch method resolves to a value, if the fetch has not been aborted due to deletion, eviction, or being overwritten, then it is added to the cache using the options provided.
If the key is evicted or deleted before the `fetchMethod` resolves, then the AbortSignal passed to the `fetchMethod` will receive an `abort` event, and the promise returned by `fetch()` will reject with the reason for the abort.
If a `signal` is passed to the `fetch()` call, then aborting the signal will abort the fetch and cause the `fetch()` promise to reject with the reason provided.

108.4.8 <tt>peek(key, { allowStale } = {}) => value</tt>

Like `get()` but doesn't update recency or delete stale items.
Returns `undefined` if the item is stale, unless `allowStale` is set either on the cache or in the options object.

108.4.9 <tt>has(key, { updateAgeOnHas, status } = {}) => Boolean</tt>

Check if a key is in the cache, without updating the recency of use. Age is updated if `updateAgeOnHas` is set to `true` in either the options or the constructor.
Will return `false` if the item is stale, even though it is technically in the cache. The difference can be determined (if it matters) by using a `status` argument, and inspecting the `has` field.
For the usage of the `status` option, see **Status Tracking** below.

108.4.10 <tt>delete(key)</tt>

Deletes a key out of the cache.
Returns `true` if the key was deleted, `false` otherwise.

108.4.11 <tt>clear()</tt>

Clear the cache entirely, throwing away all values.
Deprecated alias: `reset()`

108.4.12 <tt>keys()</tt>

Return a generator yielding the keys in the cache, in order from most recently used to least recently used.

108.4.13 <tt>rkeys()</tt>

Return a generator yielding the keys in the cache, in order from least recently used to most recently used.

108.4.14 <tt>values()</tt>

Return a generator yielding the values in the cache, in order from most recently used to least recently used.

108.4.15 <tt>rvalues()</tt>

Return a generator yielding the values in the cache, in order from least recently used to most recently used.

108.4.16 <tt>entries()</tt>

Return a generator yielding `[key, value]` pairs, in order from most recently used to least recently used.

108.4.17 <tt>rentries()</tt>

Return a generator yielding `[key, value]` pairs, in order from least recently used to most recently used.

108.4.18 <tt>find(fn, [getOptions])</tt>

Find a value for which the supplied `fn` method returns a truthy value, similar to `Array.find()`.
`fn` is called as `fn(value, key, cache)`.
The optional `getOptions` are applied to the resulting `get()` of the item found.

108.4.19 <tt>dump()</tt>

Return an array of `[key, entry]` objects which can be passed to `cache.load()`
The `start` fields are calculated relative to a portable `Date.now()` timestamp, even if `performance.now()` is available.
Stale entries are always included in the `dump`, even if `allowStale` is false.
Note: this returns an actual array, not a generator, so it can be more easily passed around.

108.4.20 <tt>load(entries)</tt>

Reset the cache and load in the items in `entries` in the order listed. Note that the shape of the resulting cache may be different if the same options are not used in both caches.
The `start` fields are assumed to be calculated relative to a portable `Date.now()` timestamp, even if `performance.now()` is available.

108.4.21 <tt>purgeStale()</tt>

Delete any stale entries. Returns `true` if anything was removed, `false` otherwise.
Deprecated alias: `prune`

108.4.22 <tt>getRemainingTTL(key)</tt>

Return the number of ms left in the item's TTL. If item is not in cache, returns `0`. Returns `Infinity` if item is in cache without a defined TTL.

108.4.23 <tt>forEach(fn, [thisp])</tt>

Call the `fn` function with each set of `fn(value, key, cache)` in the LRU cache, from most recent to least recently used.
Does not affect recency of use.
If `thisp` is provided, function will be called in the `this`-context of the provided object.

108.4.24 <tt>rforEach(fn, [thisp])</tt>

Same as `cache.forEach(fn, thisp)`, but in order from least recently used to most recently used.

108.4.25 <tt>pop()</tt>

Evict the least recently used item, returning its value.
Returns `undefined` if cache is empty.

108.4.26 Internal Methods and Properties

In order to optimize performance as much as possible, "private" members and methods are exposed on the object as normal properties, rather than being accessed via Symbols, private members, or closure variables.
Do not use or rely on these. They will change or be removed without notice. They will cause undefined behavior if used inappropriately. There is no need or reason to ever call them directly.
This documentation is here so that it is especially clear that this not "undocumented" because someone forgot; it *is* documented, and the documentation is telling you not to do it.
Do not report bugs that stem from using these properties. They will be ignored.

- `initializeTTLTracking()` Set up the cache for tracking TTLs
- `updateItemAge(index)` Called when an item age is updated, by internal ID
- `setItemTTL(index)` Called when an item ttl is updated, by internal ID
- `isStale(index)` Called to check an item's staleness, by internal ID
- `initializeSizeTracking()` Set up the cache for tracking item size. Called automatically when a size is specified.
- `removeItemSize(index)` Updates the internal size calculation when an item is removed or modified, by internal ID
- `addItemSize(index)` Updates the internal size calculation when an item is added or modified, by internal ID
- `indexes()` An iterator over the non-stale internal IDs, from most recently to least recently used.

- `rindexes()` An iterator over the non-stale internal IDs, from least recently to most recently used.
- `newIndex()` Create a new internal ID, either reusing a deleted ID, evicting the least recently used ID, or walking to the end of the allotted space.
- `evict()` Evict the least recently used internal ID, returning its ID. Does not do any bounds checking.
- `connect(p, n)` Connect the `p` and `n` internal IDs in the linked list.
- `moveToTail(index)` Move the specified internal ID to the most recently used position.
- `keyMap` Map of keys to internal IDs
- `keyList` List of keys by internal ID
- `valList` List of values by internal ID
- `sizes` List of calculated sizes by internal ID
- `ttls` List of TTL values by internal ID
- `starts` List of start time values by internal ID
- `next` Array of "next" pointers by internal ID
- `prev` Array of "previous" pointers by internal ID
- `head` Internal ID of least recently used item
- `tail` Internal ID of most recently used item
- `free` Stack of deleted internal IDs

108.5 Status Tracking

Occasionally, it may be useful to track the internal behavior of the cache, particularly for logging, debugging, or for behavior within the `fetchMethod`. To do this, you can pass a `status` object to the `get()`, `set()`, `has()`, and `fetch()` methods.
The `status` option should be a plain JavaScript object.
The following fields will be set appropriately:

```
interface Status<V> {
  /**
   * The status of a set() operation.
   *
   * - add:  the item was not found in the cache, and was added
   * - update:  the item was in the cache, with the same value provided
   * - replace:  the item was in the cache, and replaced
   * - miss:  the item was not added to the cache for some reason
   */
  set?:  'add' | 'update' | 'replace' | 'miss'
  /**
   * the ttl stored for the item, or undefined if ttls are not used.
   */
  ttl?:  LRUMilliseconds
  /**
   * the start time for the item, or undefined if ttls are not used.
   */
  start?:  LRUMilliseconds
  /**
   * The timestamp used for TTL calculation
   */
  now?:  LRUMilliseconds
  /**
   * the remaining ttl for the item, or undefined if ttls are not used.
   */
  remainingTTL?:  LRUMilliseconds
  /**
   * The calculated size for the item, if sizes are used.
   */
  size?:  LRUSize
  /**
   * A flag indicating that the item was not stored, due to exceeding the
   * {@link maxEntrySize}
   */
  maxEntrySizeExceeded?:  true
  /**
   * The old value, specified in the case of `set:'update'` or
   * `set:'replace'`
   */
  oldValue?:  V
  /**
   * The results of a {@link has} operation
   *
   * - hit:  the item was found in the cache
   * - stale:  the item was found in the cache, but is stale
   * - miss:  the item was not found in the cache
   */
  has?:  'hit' | 'stale' | 'miss'
  /**
   * The status of a {@link fetch} operation.
   * Note that this can change as the underlying fetch() moves through
   * various states.
   *
   * - inflight:  there is another fetch() for this key which is in process
   * - get:  there is no fetchMethod, so {@link get} was called.
   * - miss:  the item is not in cache, and will be fetched.
   * - hit:  the item is in the cache, and was resolved immediately.
   * - stale:  the item is in the cache, but stale.
   * - refresh:  the item is in the cache, and not stale, but
   *   {@link forceRefresh} was specified.
   */
  fetch?:  'get' | 'inflight' | 'miss' | 'hit' | 'stale' | 'refresh'
  /**
   * The {@link fetchMethod} was called
   */
  fetchDispatched?:  true
  /**
   * The cached value was updated after a successful call to fetchMethod
   */
  fetchUpdated?:  true
  /**
   * The reason for a fetch() rejection.    Either the error raised by the
   * {@link fetchMethod}, or the reason for an AbortSignal.
   */
  fetchError?:  Error
  /**
   * The fetch received an abort signal
   */
  fetchAborted?:  true
  /**
   * The abort signal received was ignored, and the fetch was allowed to
   * continue.
   */
  fetchAbortIgnored?:  true
  /**
   * The fetchMethod promise resolved successfully
   */
  fetchResolved?:  true
  /**
   * The results of the fetchMethod promise were stored in the cache
   */
  fetchUpdated?:  true
  /**
   * The fetchMethod promise was rejected
   */
  fetchRejected?:  true
  /**
   * The status of a {@link get} operation.
   *
   * - fetching:  The item is currently being fetched.    If a previous value is
   *   present and allowed, that will be returned.
   * - stale:  The item is in the cache, and is stale.
   * - hit:  the item is in the cache
   * - miss:  the item is not in the cache
   */
  get?:  'stale' | 'hit' | 'miss'
  /**
   * A fetch or get operation returned a stale value.
   */
  returnedStale?:  true
}
```

108.6 Storage Bounds Safety

This implementation aims to be as flexible as possible, within the limits of safe memory consumption and optimal performance.

At initial object creation, storage is allocated for `max` items. If `max` is set to zero, then some performance is lost, and item count is unbounded. Either `maxSize` or `ttl` *must* be set if `max` is not specified.

If `maxSize` is set, then this creates a safe limit on the maximum storage consumed, but without the performance benefits of pre-allocation. When `maxSize` is set, every item *must* provide a size, either via the `size←Calculation` method provided to the constructor, or via a `size` or `sizeCalculation` option provided to `cache.set()`. The size of every item *must* be a positive integer.
If neither `max` nor `maxSize` are set, then `ttl` tracking must be enabled. Note that, even when tracking item `ttl`, items are *not* preemptively deleted when they become stale, unless `ttlAutopurge` is enabled. Instead, they are only purged the next time the key is requested. Thus, if `ttlAutopurge`, `max`, and `maxSize` are all not set, then the cache will potentially grow unbounded.
In this case, a warning is printed to standard error. Future versions may require the use of `ttlAutopurge` if `max` and `maxSize` are not specified.
If you truly wish to use a cache that is bound *only* by TTL expiration, consider using a `Map` object, and calling `setTimeout` to delete entries when they expire. It will perform much better than an LRU cache.
Here is an implementation you may use, under the same `license` as this package:
```
// a storage-unbounded ttl cache that is not an lru-cache
const cache = {
  data: new Map(),
  timers: new Map(),
  set: (k, v, ttl) => {
    if (cache.timers.has(k)) {
      clearTimeout(cache.timers.get(k))
    }
    cache.timers.set(
      k,
      setTimeout(() => cache.delete(k), ttl)
    )
    cache.data.set(k, v)
  },
  get: k => cache.data.get(k),
  has: k => cache.data.has(k),
  delete: k => {
    if (cache.timers.has(k)) {
      clearTimeout(cache.timers.get(k))
    }
    cache.timers.delete(k)
    return cache.data.delete(k)
  },
  clear: () => {
    cache.data.clear()
    for (const v of cache.timers.values()) {
      clearTimeout(v)
    }
    cache.timers.clear()
  },
}
```
If that isn't to your liking, check out `@isaacs/ttlcache`.

108.7 Performance

As of January 2022, version 7 of this library is one of the most performant LRU cache implementations in JavaScript.
Benchmarks can be extremely difficult to get right. In particular, the performance of set/get/delete operations on objects will vary *wildly* depending on the type of key used. V8 is highly optimized for objects with keys that are short strings, especially integer numeric strings. Thus any benchmark which tests *solely* using numbers as keys will tend to find that an object-based approach performs the best.
Note that coercing *anything* to strings to use as object keys is unsafe, unless you can be 100% certain that no other type of value will be used. For example:
```
const myCache = {}
const set = (k, v) => (myCache[k] = v)
const get = k => myCache[k]
set({}, 'please hang onto this for me')
set('[object Object]', 'oopsie')
```
Also beware of "Just So" stories regarding performance. Garbage collection of large (especially: deep) object graphs can be incredibly costly, with several "tipping points" where it increases exponentially. As a result, putting that off until later can make it much worse, and less predictable. If a library performs well, but only in a scenario where the object graph is kept shallow, then that won't help you if you are using large objects as keys.
In general, when attempting to use a library to improve performance (such as a cache like this one), it's best to choose an option that will perform well in the sorts of scenarios where you'll actually use it.
This library is optimized for repeated gets and minimizing eviction time, since that is the expected need of a LRU. Set operations are somewhat slower on average than a few other options, in part because of that optimization. It is assumed that you'll be caching some costly operation, ideally as rarely as possible, so optimizing set over get would be unwise.

If performance matters to you:

1. If it's at all possible to use small integer values as keys, and you can guarantee that no other types of values will be used as keys, then do that, and use a cache such as `lru-fast`, or `mnemonist's LRUCache` which uses an Object as its data store.

2. Failing that, if at all possible, use short non-numeric strings (ie, less than 256 characters) as your keys, and use `mnemonist's LRUCache`.

3. If the types of your keys will be long strings, strings that look like floats, `null`, objects, or some mix of types, or if you aren't sure, then this library will work well for you.

4. Do not use a `dispose` function, size tracking, or especially ttl behavior, unless absolutely needed. These features are convenient, and necessary in some use cases, and every attempt has been made to make the performance impact minimal, but it isn't nothing.

108.8 Breaking Changes in Version 7

This library changed to a different algorithm and internal data structure in version 7, yielding significantly better performance, albeit with some subtle changes as a result.
If you were relying on the internals of LRUCache in version 6 or before, it probably will not work in version 7 and above.
For more info, see the change log.

Chapter 109

README

lru.min

An extremely fast and efficient `LRU` **Cache** for **JavaScript** (**Browser** compatible) — **6.8KB**.

109.0.1 Why another LRU?

- **lru.min** is fully compatible with both **Node.js** _(8+)_, **Bun**, **Deno** and, browser environments. All of this, while maintaining the same high performance `_(and a little more)_` as the most popular **LRU** packages.

109.0.2 Install

```
# Node.js
npm i lru.min
# Bun
bun add lru.min
# Deno
deno add npm:lru.min
```

109.0.3 Usage

109.0.3.1 Quickstart

```
import { createLRU } from 'lru.min';
const max = 2;
const onEviction = (key, value) => {
  console.log(`Key "${key}" with value "${value}" has been evicted.`);
};
const LRU = createLRU({
  max,
  onEviction,
});
LRU.set('A', 'My Value');
LRU.set('B', 'Other Value');
LRU.set('C', 'Another Value');
// => Key "A" with value "My Value" has been evicted.
LRU.has('B');
LRU.get('B');
LRU.delete('B');
// => Key "B" with value "Other Value" has been evicted.
LRU.peek('C');
LRU.clear(); // LRU.evict(max)
// => Key "C" with value "Another Value" has been evicted.
LRU.set('D', "You're amazing ");
LRU.size; // 1
LRU.max; // 2
LRU.available; // 1
LRU.resize(10);
LRU.size; // 1
LRU.max; // 10
LRU.available; // 9
```

For *up-to-date* documentation, always follow the `README.md` in the **GitHub** repository.

109.0.3.2 Import

109.0.3.2.1 ES Modules

```
import { createLRU } from 'lru.min';
```

109.0.3.2.2 CommonJS

```
const { createLRU } = require('lru.min');
```

109.0.3.2.3 Browser

> Requires **ES6**.

```
<script src="https://cdn.jsdelivr.net/npm/lru.min@1.x.x/browser/lru.min.js"></script>
```

- You can use tools such as `Babel` to increase the compatibility rate.

109.0.3.3 Create a new LRU Cache

> Set maximum size when creating **LRU**.

```
const LRU = createLRU({ max: 150_000 });
```

Also, you can set a callback for every deletion/eviction:

```
const LRU = createLRU({
  max: 150_000,
  onEviction: (key, value) => {
    // do something
  },
});
```

109.0.3.4 Set a cache

Adds a key-value pair to the cache. Updates the value if the key already exists

```
LRU.set('key', 'value');
```

> `undefined` keys will simply be ignored.

109.0.3.5 Get a cache

Retrieves the value for a given key and moves the key to the most recent position.

```
LRU.get('key');
```

109.0.3.6 Peek a cache

Retrieves the value for a given key without changing its position.

```
LRU.peek('key');
```

109.0.3.7 Check if a key exists

```
LRU.has('key');
```

109.0.3.8 Delete a cache

```
LRU.delete('key');
```

109.0.3.9 Evict from the oldest cache

Evicts the specified number of the oldest items from the cache.

```
LRU.evict(1000);
```

> [!TIP]
>
> - Methods that perform eviction(s) when maximum size is reached: `set` and `resize`.
> - Methods that always perform eviction(s): `delete`, `clear`, and `evict` itself.

109.0.3.10 Resize the cache

Resizes the cache to a new maximum size, evicting items if necessary.

```
LRU.resize(50_000);
```

109.0.3.11 Clear the cache

Clears and disposes (if used) all key-value pairs from the cache.

```
LRU.clear();
```

109.0.3.12 Debugging

109.0.3.12.1 Get the max size of the cache `LRU.max;`

109.0.3.12.2 Get the current size of the cache `LRU.size;`

109.0.3.12.3 Get the available slots in the cache `LRU.available;`

109.0.3.13 Iterating the cache

109.0.3.13.1 Get all keys Iterates over all keys in the cache, from most recent to least recent.

```
const keys = [...LRU.keys()];
```

109.0.3.13.2 Get all values Iterates over all values in the cache, from most recent to least recent.

```
const values = [...LRU.values()];
```

109.0.3.13.3 Get all entries Iterates over `[key, value]` pairs in the cache, from most recent to least recent.

```
const entries = [...LRU.entries()];
```

109.0.3.13.4 Run a callback for each entry Iterates over each value-key pair in the cache, from most recent to least recent.

```
LRU.forEach((value, key) => {
  // do something
});
```

109.0.3.14 TypeScript

You can set types for both keys and values. For example:

```
import { createLRU } from 'lru.min';
type Key = number;
type Value = {
  name:  string;
};
const LRU = createLRU<Key, Value>({ max:  1000 });
LRU.set(1, { name:  'Peter' });
LRU.set(2, { name:  'Mary' });
```

Also:

```
import { createLRU, type CacheOptions } from 'lru.min';
type Key = number;
type Value = {
  name:  string;
};
const options:  CacheOptions<Key, Value> = {
  max:  10,
  onEviction(key, value) {
    console.log(key, value);
  },
};
// No need to repeat the type params
const LRU = createLRU(options);
LRU.set(1, { name:  'Peter' });
LRU.set(2, { name:  'Mary' });
```

109.0.3.15 Performance

The benchmark is performed by comparing `1,000,000` runs through a maximum cache limit of `100,000`, getting `333,333` caches and deleting `200,000` keys 10 consecutive times, clearing the cache every run.

- **`lru-cache`** `v11.0.0`
  - **`quick-lru`** `v7.0.0`

```
# Time:
  lru.min:        240.45ms
  lru-cache:    258.32ms
  quick-lru:    279.89ms
# CPU:
```

```
lru.min:        275558.30µs
lru-cache:     306858.30µs
quick-lru:     401318.80µs
```

- See detailed results and how the tests are run and compared in the **benchmark** directory.

109.0.4 Security Policy

Please check the **SECURITY.md**.

109.0.5 Contributing

See the **Contributing Guide** and please follow our **Code of Conduct**

109.0.6 Acknowledgements

lru.min is based and inspired on the architecture and code of both **lru-cache** and **quick-lru**, simplifying their core concepts for enhanced performance and compatibility.
For more comprehensive features such as **TTL** support, consider using and supporting them

- The architecture is mostly based on @isaacs — **lru-cache**.
- Most of the methods names and its functionalities were inspired by @sindresorhus — **quick-lru**.
-

109.0.6.0.1 What comes from <a href="https://github.com/isaacs/node-lru-cache" ><strong>lru-cache</strong></a>? Architecture's essence:

> *It's not the same code, but majority based on* this.

```
let free:  number[] = [];
const keyMap:  Map<Key, number> = new Map();
const keyList:  (Key | undefined)[] = new Array(max).fill(undefined);
const valList:  (Value | undefined)[] = new Array(max).fill(undefined);
const next:  number[] = new Array(max).fill(0);
const prev:  number[] = new Array(max).fill(0);
```

109.0.6.0.2 What comes from <a href="https://github.com/sindresorhus/quick-lru" ><strong>quick-lru</strong></a>? Name of methods and options _(including their final functionality ideas)_:

- `resize`
- `peek`
- `onEviction`
- `forEach`
- `entriesDescending` as `entries`

109.0.7 License

lru.min is under the **MIT License**.

Chapter 110

0.6.3 / 2021-05-23

- Fix HKSCS encoding to prefer Big5 codes if both Big5 and HKSCS codes are possible (#264)

110.1 0.6.2 / 2020-07-08

- Support Uint8Array-s decoding without conversion to Buffers, plus fix an edge case.

110.2 0.6.1 / 2020-06-28

- Support Uint8Array-s directly when decoding (#246, by @gyzerok)
- Unify package.json version ranges to be strictly semver-compatible (#241)
- Fix minor issue in UTF-32 decoder's endianness detection code.

110.3 0.6.0 / 2020-06-08

- Updated 'gb18030' encoding to :2005 edition (see https://github.com/whatwg/encoding/issues/22).
- Removed `iconv.extendNodeEncodings()` mechanism. It was deprecated 5 years ago and didn't work in recent Node versions.
- Reworked Streaming API behavior in browser environments to fix #204. Streaming API will be excluded by default in browser packs, saving ∼100Kb bundle size, unless enabled explicitly using ‘iconv.enable←StreamingAPI(require('stream'))`.
- Updates to development environment & tests:
  - Added ./test/webpack private package to test complex new use cases that need custom environment. It's tested as a separate job in Travis CI.
  - Updated generation code for the new EUC-KR index file format from Encoding Standard.
  - Removed Buffer() constructor in tests (#197 by @gabrielschulhof).

110.4 0.5.2 / 2020-06-08

- Added `iconv.getEncoder()` and `iconv.getDecoder()` methods to typescript definitions (#229).
- Fixed semver version to 6.1.2 to support Node 8.x (by @tanandara).
- Capped iconv version to 2.x as 3.x has dropped support for older Node versions.
- Switched from instanbul to c8 for code coverage.

110.5 0.5.1 / 2020-01-18

- Added cp720 encoding (#221, by @kr-deps)
- (minor) Changed Changelog.md formatting to use h2.

110.6 0.5.0 / 2019-06-26

- Added UTF-32 encoding, both little-endian and big-endian variants (UTF-32LE, UTF32-BE). If endianness is not provided for decoding, it's deduced automatically from the stream using a heuristic similar to what we use in UTF-16. (great work in #216 by @kshetline)
- Several minor updates to README (#217 by @oldj, plus some more)
- Added Node versions 10 and 12 to Travis test harness.

110.7 0.4.24 / 2018-08-22

- Added MIK encoding (#196, by @Ivan-Kalatchev)

110.8 0.4.23 / 2018-05-07

- Fix deprecation warning in Node v10 due to the last usage of `new Buffer` (#185, by @felixbuenemann)
- Switched from NodeBuffer to Buffer in typings (#155 by @felixfbecker, #186 by @larssn)

110.9 0.4.22 / 2018-05-05

- Use older semver style for dependencies to be compatible with Node version 0.10 (#182, by @dougwilson)
- Fix tests to accomodate fixes in Node v10 (#182, by @dougwilson)

110.10 0.4.21 / 2018-04-06

- Fix encoding canonicalization (#156)
- Fix the paths in the "browser" field in package.json (#174 by @LMLB)
- Removed "contributors" section in package.json - see Git history instead.

110.11 0.4.20 / 2018-04-06

- Updated `new Buffer()` usages with recommended replacements as it's being deprecated in Node v10 (#176, #178 by @ChALkeR)

110.12 0.4.19 / 2017-09-09

- Fixed iso8859-1 codec regression in handling untranslatable characters (#162, caused by #147)
- Re-generated windows1255 codec, because it was updated in iconv project
- Fixed grammar in error message when iconv-lite is loaded with encoding other than utf8

110.13 0.4.18 / 2017-06-13

- Fixed CESU-8 regression in Node v8.

110.14 0.4.17 / 2017-04-22

- Updated typescript definition file to support Angular 2 AoT mode (#153 by @larssn)

110.15 0.4.16 / 2017-04-22

- Added support for React Native (#150)
- Changed iso8859-1 encoding to usine internal 'binary' encoding, as it's the same thing (#147 by @mscdex)
- Fixed typo in Readme (#138 by @jiangzhuo)
- Fixed build for Node v6.10+ by making correct version comparison
- Added a warning if iconv-lite is loaded not as utf-8 (see #142)

110.16 0.4.15 / 2016-11-21

- Fixed typescript type definition (#137)

110.17 0.4.14 / 2016-11-20

- Preparation for v1.0
- Added Node v6 and latest Node versions to Travis CI test rig
- Deprecated Node v0.8 support
- Typescript typings (@larssn)
- Fix encoding of Euro character in GB 18030 (inspired by @lygstate)
- Add ms prefix to dbcs windows encodings (@rokoroku)

110.18 0.4.13 / 2015-10-01

- Fix silly mistake in deprecation notice.

110.19 0.4.12 / 2015-09-26

- Node v4 support:
  - Added CESU-8 decoding (#106)
  - Added deprecation notice for `extendNodeEncodings`
  - Added Travis tests for Node v4 and io.js latest (#105 by @Mithgol)

110.20 0.4.11 / 2015-07-03

- Added CESU-8 encoding.

110.21 0.4.10 / 2015-05-26

- Changed UTF-16 endianness heuristic to take into account any ASCII chars, not just spaces. This should minimize the importance of "default" endianness.

110.22 0.4.9 / 2015-05-24

- Streamlined BOM handling: strip BOM by default, add BOM when encoding if addBOM: true. Added docs to Readme.
- UTF16 now uses UTF16-LE by default.
- Fixed minor issue with big5 encoding.
- Added io.js testing on Travis; updated node-iconv version to test against. Now we just skip testing SBCS encodings that node-iconv doesn't support.
- (internal refactoring) Updated codec interface to use classes.
- Use strict mode in all files.

110.23 0.4.8 / 2015-04-14

- added alias UNICODE-1-1-UTF-7 for UTF-7 encoding (#94)

110.24 0.4.7 / 2015-02-05

- stop official support of Node.js v0.8. Should still work, but no guarantees. reason: Packages needed for testing are hard to get on Travis CI.
- work in environment where Object.prototype is monkey patched with enumerable props (#89).

110.25 0.4.6 / 2015-01-12

- fix rare aliases of single-byte encodings (thanks @mscdex)
- double the timeout for dbcs tests to make them less flaky on travis

110.26 0.4.5 / 2014-11-20

- fix windows-31j and x-sjis encoding support (@nleush)
- minor fix: undefined variable reference when internal error happens

110.27 0.4.4 / 2014-07-16

- added encodings UTF-7 (RFC2152) and UTF-7-IMAP (RFC3501 Section 5.1.3)
- fixed streaming base64 encoding

110.28 0.4.3 / 2014-06-14

- added encodings UTF-16BE and UTF-16 with BOM

110.29 0.4.2 / 2014-06-12

- don't throw exception if `extendNodeEncodings()` is called more than once

110.30 0.4.1 / 2014-06-11

- codepage 808 added

110.31 0.4.0 / 2014-06-10

- code is rewritten from scratch
- all widespread encodings are supported
- streaming interface added
- browserify compatibility added
- (optional) extend core primitive encodings to make usage even simpler
- moved from vows to mocha as the testing framework

Chapter 111

iconv-lite: Pure JS character encoding conversion

- No need for native code compilation. Quick to install, works on Windows and in sandboxed environments like Cloud9.
- Used in popular projects like Express.js (body_parser), Grunt, Nodemailer, Yeoman and others.
- Faster than node-iconv (see below for performance comparison).
- Intuitive encode/decode API, including Streaming support.
- In-browser usage via browserify or webpack (∼180kb gzip compressed with Buffer shim included).
- Typescript type definition file included.
- React Native is supported (need to install `stream` module to enable Streaming API).
- License: MIT.

111.1 Usage

111.1.1 Basic API

```
var iconv = require('iconv-lite');
// Convert from an encoded buffer to a js string.
str = iconv.decode(Buffer.from([0x68, 0x65, 0x6c, 0x6c, 0x6f]), 'win1251');
// Convert from a js string to an encoded buffer.
buf = iconv.encode("Sample input string", 'win1251');
// Check if encoding is supported
iconv.encodingExists("us-ascii")
```

111.1.2 Streaming API

```
// Decode stream (from binary data stream to js strings)
http.createServer(function(req, res) {
    var converterStream = iconv.decodeStream('win1251');
    req.pipe(converterStream);
    converterStream.on('data', function(str) {
        console.log(str); // Do something with decoded strings, chunk-by-chunk.
    });
});
// Convert encoding streaming example
fs.createReadStream('file-in-win1251.txt')
    .pipe(iconv.decodeStream('win1251'))
    .pipe(iconv.encodeStream('ucs2'))
    .pipe(fs.createWriteStream('file-in-ucs2.txt'));
// Sugar:  all encode/decode streams have .collect(cb) method to accumulate data.
http.createServer(function(req, res) {
    req.pipe(iconv.decodeStream('win1251')).collect(function(err, body) {
        assert(typeof body == 'string');
        console.log(body); // full request body string
    });
});
```

111.2 Supported encodings

- All node.js native encodings: utf8, ucs2 / utf16-le, ascii, binary, base64, hex.
- Additional unicode encodings: utf16, utf16-be, utf-7, utf-7-imap, utf32, utf32-le, and utf32-be.
- All widespread singlebyte encodings: Windows 125x family, ISO-8859 family, IBM/DOS codepages, Macintosh family, KOI8 family, all others supported by iconv library. Aliases like 'latin1', 'us-ascii' also supported.
- All widespread multibyte encodings: CP932, CP936, CP949, CP950, GB2312, GBK, GB18030, Big5, Shift↩ _JIS, EUC-JP.

See `all supported encodings on wiki`.
Most singlebyte encodings are generated automatically from `node-iconv`. Thank you Ben Noordhuis and libiconv authors!
Multibyte encodings are generated from `Unicode.org mappings` and `WHATWG Encoding Standard mappings`. Thank you, respective authors!

111.3 Encoding/decoding speed

Comparison with node-iconv module (1000x256kb, on MacBook Pro, Core i5/2.6 GHz, Node v0.12.0). Note: your results may vary, so please always check on your hardware.

```
operation             iconv@2.1.4   iconv-lite@0.4.7
----------------------------------------------------------
encode('win1251')     ~96 Mb/s      ~320 Mb/s
decode('win1251')     ~95 Mb/s      ~246 Mb/s
```

111.4 BOM handling

- Decoding: BOM is stripped by default, unless overridden by passing `stripBOM: false` in options (f.↩ ex. `iconv.decode(buf, enc, {stripBOM: false})`). A callback might also be given as a `stripBOM` parameter - it'll be called if BOM character was actually found.
- If you want to detect UTF-8 BOM when decoding other encodings, use `node-autodetect-decoder-stream` module.
- Encoding: No BOM added, unless overridden by `addBOM: true` option.

111.5 UTF-16 Encodings

This library supports UTF-16LE, UTF-16BE and UTF-16 encodings. First two are straightforward, but UTF-16 is trying to be smart about endianness in the following ways:

- Decoding: uses BOM and 'spaces heuristic' to determine input endianness. Default is UTF-16LE, but can be overridden with ‘defaultEncoding: 'utf-16be’`option. Strips BOM unless`stripBOM: false`.`
- `Encoding: uses UTF-16LE and writes BOM by default. Use`addBOM: false` to override.

111.6 UTF-32 Encodings

This library supports UTF-32LE, UTF-32BE and UTF-32 encodings. Like the UTF-16 encoding above, UTF-32 defaults to UTF-32LE, but uses BOM and 'spaces heuristics' to determine input endianness.

- The default of UTF-32LE can be overridden with the ‘defaultEncoding: 'utf-32be’`option. Strips BOM unless`stripBOM: false`.`
- `Encoding: uses UTF-32LE and writes BOM by default. Use`addBOM: false`to override.` (defaultEncoding: 'utf-32be'` can also be used here to change encoding.)

111.7 Other notes

When decoding, be sure to supply a Buffer to decode() method, otherwise `bad things usually happen`.
Untranslatable characters are set to � or ?. No transliteration is currently supported.
Node versions 0.10.31 and 0.11.13 are buggy, don't use them (see #65, #77).

111.8 Testing

```
$ git clone git@github.com:ashtuchkin/iconv-lite.git
$ cd iconv-lite
$ npm install
$ npm test

$ # To view performance:
$ node test/performance.js
$ # To view test coverage:
$ npm run coverage
$ open coverage/lcov-report/index.html
```

Chapter 112

README

112.1 MariaDB Node.js connector

Non-blocking MariaDB and MySQL client for Node.js.
MariaDB and MySQL client, 100% JavaScript, with TypeScript definition, with the Promise API.
version before 2.4 is compatible with Node.js 6+ version after 2.4 is compatible with Node.js 10+

112.1.1 Documentation

See `promise documentation` for detailed API.
`Callback documentation` describe the callback wrapper for compatibility with existing drivers.

112.1.2 Why a New Client?

While there are existing MySQL clients that work with MariaDB, (such as the `mysql` and `mysql2` clients), the MariaDB Node.js Connector offers new functionality, like Insert Streaming, Pipelining, `ed25519 plugin authentication` while making no compromises on performance.

112.1.2.1 Insert Streaming

Using a Readable stream in your application, you can stream `INSERT` statements to MariaDB through the Connector.

```
https.get('https://someContent', readableStream => {
    //readableStream implement Readable, driver will stream data to database
    connection.query("INSERT INTO myTable VALUE (?)", [readableStream]);
});
```

112.1.2.2 Pipelining

With Pipelining, the Connector sends commands without waiting for server results, preserving order. For instance, consider the use of executing two `INSERT` statements.
The Connector doesn't wait for query results before sending the next `INSERT` statement. Instead, it sends queries one after the other, avoiding much of the network latency.
For more information, see the Pipelining documentation.

112.1.2.3 Bulk insert

Some use cases require a large amount of data to be inserted into a database table. By using batch processing, these queries can be sent to the database in one call, thus improving performance.
For more information, see the Batch documentation.

112.1.3 Benchmarks

MariaDB provides benchmarks comparing the Connector with popular Node.js MySQL clients, including:

- `promise-mysql` version 4.0.4 + `mysql` version 2.17.1
- `mysql2` version 1.6.5

```
promise-mysql  :  646 ops/sec ±2.20%
mysql2         :  746 ops/sec ±2.35%
mariadb        :  961 ops/sec ±2.82%
```

query: **SELECT < all mysql fields >, 1 FROM mysql.user LIMIT 1**
For more information, see the Benchmarks page.

112.1.4 Quick Start

The MariaDB Connector is available through the Node.js repositories. You can install it using npm :

```
$ npm install mariadb
```

example:

```
const mariadb = require('mariadb');
const pool = mariadb.createPool({host:  process.env.DB_HOST, user:  process.env.DB_USER, connectionLimit:
      5});
async function asyncFunction() {
  let conn;
  try {
    conn = await pool.getConnection();
    const rows = await conn.query("SELECT 1 as val");
    // rows:  [ {val:  1}, meta:  ...  ]
    const res = await conn.query("INSERT INTO myTable value (?, ?)", [1, "mariadb"]);
    // res:  { affectedRows:  1, insertId:  1, warningStatus:  0 }
  } finally {
    if (conn) conn.release(); //release to pool
  }
}
```

112.1.5 Contributing

If you would like to contribute to the MariaDB Node.js Connector, please follow the instructions given in the Developers Guide.
To file an issue or follow the development, see `JIRA`.

Chapter 113

0.3.0 / 2014-09-07

- Support Node.js 0.6
- Throw error when parameter format invalid on parse

113.1 0.2.0 / 2014-06-18

- Add `typer.format()` to format media types

113.2 0.1.0 / 2014-06-17

- Accept `req` as argument to `parse`
- Accept `res` as argument to `parse`
- Parse media type with extra LWS between type and first parameter

113.3 0.0.0 / 2014-06-13

- Initial implementation

Chapter 114

media-typer

Simple RFC 6838 media type parser

114.1 Installation

```
$ npm install media-typer
```

114.2 API

```
var typer = require('media-typer')
```

114.2.1 typer.parse(string)

```
var obj = typer.parse('image/svg+xml; charset=utf-8')
```

Parse a media type string. This will return an object with the following properties (examples are shown for the string "image/svg+xml; charset=utf-8`):

- `type`: The type of the media type (always lower case). Example: "image`
- `subtype`: The subtype of the media type (always lower case). Example: "svg`
- `suffix`: The suffix of the media type (always lower case). Example: "xml`
- `parameters`: An object of the parameters in the media type (name of parameter always lower case). Example: '{charset: 'utf-8'}`

114.2.2 typer.parse(req)

```
var obj = typer.parse(req)
```

Parse the `content-type` header from the given `req`. Short-cut for 'typer.parse(req.headers['content-type'])`.

114.2.3 typer.parse(res)

```
var obj = typer.parse(res)
```

Parse the `content-type` header set on the given `res`. Short-cut for 'typer.parse(res.getHeader('content-type'))`.

114.2.4 typer.format(obj)

```
var obj = typer.format({type:  'image', subtype:  'svg', suffix:  'xml'})
```

Format an object into a media type string. This will return a string of the mime type for the given object. For the properties of the object, see the documentation for `typer.parse(string)`.

114.3 License

[MIT](LICENSE)

Chapter 115

1.0.1 / 2016-01-17

- perf: enable strict mode

115.1 1.0.0 / 2015-03-01

- Add option to only add new descriptors
- Add simple argument validation
- Add jsdoc to source file

115.2 0.0.2 / 2013-12-14

- Move repository to `component` organization

115.3 0.0.1 / 2013-10-29

- Initial release

Chapter 116

merge-descriptors

Merge objects using descriptors.

```
var thing = {
  get name() {
    return 'jon'
  }
}
var animal = {
}
merge(animal, thing)
animal.name === 'jon'
```

116.1 API

116.1.1 merge(destination, source)

Redefines `destination`'s descriptors with `source`'s. The return value is the `destination` object.

116.1.2 merge(destination, source, false)

Defines `source`'s descriptors on `destination` if `destination` does not have a descriptor by the same name. The return value is the `destination` object.

116.2 License

[MIT](LICENSE)

Chapter 117

1.1.2 / 2016-01-17

- perf: enable strict mode

117.1 1.1.1 / 2014-12-30

- Improve `browserify` support

117.2 1.1.0 / 2014-07-05

- Add `CONNECT` method

117.3 1.0.1 / 2014-06-02

- Fix module to work with harmony transform

117.4 1.0.0 / 2014-05-08

- Add `PURGE` method

117.5 0.1.0 / 2013-10-28

- Add `http.METHODS` support

Chapter 118

Methods

HTTP verbs that Node.js core's HTTP parser supports.
This module provides an export that is just like `http.METHODS` from Node.js core, with the following differences:

- All method names are lower-cased.
- Contains a fallback list of methods for Node.js versions that do not have a `http.METHODS` export (0.10 and lower).
- Provides the fallback list when using tools like `browserify` without pulling in the `http` shim module.

118.1 Install

```
$ npm install methods
```

118.2 API

```
var methods = require('methods')
```

118.2.1 methods

This is an array of lower-cased method names that Node.js supports. If Node.js provides the `http.METHODS` export, then this is the same array lower-cased, otherwise it is a snapshot of the verbs from Node.js 0.10.

118.3 License

[MIT](LICENSE)

Chapter 119

1.52.0 / 2022-02-21

- Add extensions from IANA for more `image/*` types
- Add extension `.asc` to `application/pgp-keys`
- Add extensions to various XML types
- Add new upstream MIME types

119.1 1.51.0 / 2021-11-08

- Add new upstream MIME types
- Mark `image/vnd.microsoft.icon` as compressible
- Mark `image/vnd.ms-dds` as compressible

119.2 1.50.0 / 2021-09-15

- Add deprecated iWorks mime types and extensions
- Add new upstream MIME types

119.3 1.49.0 / 2021-07-26

- Add extension `.trig` to `application/trig`
- Add new upstream MIME types

119.4 1.48.0 / 2021-05-30

- Add extension `.mvt` to `application/vnd.mapbox-vector-tile`
- Add new upstream MIME types
- Mark `text/yaml` as compressible

119.5 1.47.0 / 2021-04-01

- Add new upstream MIME types
- Remove ambigious extensions from IANA for `application/*+xml` types
- Update primary extension to `.es` for `application/ecmascript`

119.6 1.46.0 / 2021-02-13

- Add extension `.amr` to `audio/amr`
- Add extension `.m4s` to `video/iso.segment`
- Add extension `.opus` to `audio/ogg`
- Add new upstream MIME types

119.7 1.45.0 / 2020-09-22

- Add `application/ubjson` with extension `.ubj`
- Add `image/avif` with extension `.avif`
- Add `image/ktx2` with extension `.ktx2`
- Add extension `.dbf` to `application/vnd.dbf`
- Add extension `.rar` to `application/vnd.rar`
- Add extension `.td` to `application/urc-targetdesc+xml`
- Add new upstream MIME types
- Fix extension of `application/vnd.apple.keynote` to be `.key`

119.8 1.44.0 / 2020-04-22

- Add charsets from IANA
- Add extension `.cjs` to `application/node`
- Add new upstream MIME types

119.9 1.43.0 / 2020-01-05

- Add `application/x-keepass2` with extension `.kdbx`
- Add extension `.mxmf` to `audio/mobile-xmf`
- Add extensions from IANA for `application/*+xml` types
- Add new upstream MIME types

119.10 1.42.0 / 2019-09-25

- Add `image/vnd.ms-dds` with extension `.dds`
- Add new upstream MIME types
- Remove compressible from `multipart/mixed`

119.11 1.41.0 / 2019-08-30

- Add new upstream MIME types
- Add `application/toml` with extension `.toml`
- Mark `font/ttf` as compressible

119.12 1.40.0 / 2019-04-20

- Add extensions from IANA for `model/*` types
- Add `text/mdx` with extension `.mdx`

119.13 1.39.0 / 2019-04-04

- Add extensions `.siv` and `.sieve` to `application/sieve`
- Add new upstream MIME types

119.14 1.38.0 / 2019-02-04

- Add extension `.nq` to `application/n-quads`
- Add extension `.nt` to `application/n-triples`
- Add new upstream MIME types
- Mark `text/less` as compressible

119.15 1.37.0 / 2018-10-19

- Add extensions to HEIC image types
- Add new upstream MIME types

119.16 1.36.0 / 2018-08-20

- Add Apple file extensions from IANA
- Add extensions from IANA for `image/*` types
- Add new upstream MIME types

119.17 1.35.0 / 2018-07-15

- Add extension `.owl` to `application/rdf+xml`
- Add new upstream MIME types
  - Removes extension `.woff` from `application/font-woff`

119.18 1.34.0 / 2018-06-03

- Add extension `.csl` to `application/vnd.citationstyles.style+xml`
- Add extension `.es` to `application/ecmascript`
- Add new upstream MIME types
- Add `UTF-8` as default charset for `text/turtle`
- Mark all XML-derived types as compressible

119.19 1.33.0 / 2018-02-15

- Add extensions from IANA for `message/*` types
- Add new upstream MIME types
- Fix some incorrect OOXML types
- Remove `application/font-woff2`

119.20 1.32.0 / 2017-11-29

- Add new upstream MIME types
- Update `text/hjson` to registered `application/hjson`
- Add `text/shex` with extension `.shex`

119.21 1.31.0 / 2017-10-25

- Add `application/raml+yaml` with extension `.raml`
- Add `application/wasm` with extension `.wasm`
- Add new `font` type from IANA
- Add new upstream font extensions
- Add new upstream MIME types
- Add extensions for JPEG-2000 images

119.22 1.30.0 / 2017-08-27

- Add `application/vnd.ms-outlook`
- Add `application/x-arj`
- Add extension `.mjs` to `application/javascript`
- Add glTF types and extensions
- Add new upstream MIME types
- Add `text/x-org`
- Add VirtualBox MIME types
- Fix `source` records for `video/*` types that are IANA
- Update `font/opentype` to registered `font/otf`

119.23 1.29.0 / 2017-07-10

- Add `application/fido.trusted-apps+json`
- Add extension `.wadl` to `application/vnd.sun.wadl+xml`
- Add new upstream MIME types
- Add `UTF-8` as default charset for `text/css`

119.24 1.28.0 / 2017-05-14

- Add new upstream MIME types
- Add extension `.gz` to `application/gzip`
- Update extensions `.md` and `.markdown` to be `text/markdown`

119.25 1.27.0 / 2017-03-16

- Add new upstream MIME types
- Add `image/apng` with extension `.apng`

119.26 1.26.0 / 2017-01-14

- Add new upstream MIME types
- Add extension `.geojson` to `application/geo+json`

119.27 1.25.0 / 2016-11-11

- Add new upstream MIME types

119.28 1.24.0 / 2016-09-18

- Add `audio/mp3`
- Add new upstream MIME types

119.29 1.23.0 / 2016-05-01

- Add new upstream MIME types
- Add extension `.3gpp` to `audio/3gpp`

119.30 1.22.0 / 2016-02-15

- Add `text/slim`
- Add extension `.rng` to `application/xml`
- Add new upstream MIME types
- Fix extension of `application/dash+xml` to be `.mpd`
- Update primary extension to `.m4a` for `audio/mp4`

119.31 1.21.0 / 2016-01-06

- Add Google document types
- Add new upstream MIME types

119.32 1.20.0 / 2015-11-10

- Add `text/x-suse-ymp`
- Add new upstream MIME types

119.33 1.19.0 / 2015-09-17

- Add `application/vnd.apple.pkpass`
- Add new upstream MIME types

119.34 1.18.0 / 2015-09-03

- Add new upstream MIME types

119.35 1.17.0 / 2015-08-13

- Add `application/x-msdos-program`
- Add `audio/g711-0`
- Add `image/vnd.mozilla.apng`
- Add extension `.exe` to `application/x-msdos-program`

119.36 1.16.0 / 2015-07-29

- Add `application/vnd.uri-map`

119.37 1.15.0 / 2015-07-13

- Add `application/x-httpd-php`

119.38 1.14.0 / 2015-06-25

- Add `application/scim+json`
- Add `application/vnd.3gpp.ussd+xml`
- Add `application/vnd.biopax.rdf+xml`
- Add `text/x-processing`

119.39 1.13.0 / 2015-06-07

- Add nginx as a source
- Add `application/x-cocoa`
- Add `application/x-java-archive-diff`
- Add `application/x-makeself`
- Add `application/x-perl`
- Add `application/x-pilot`

- Add `application/x-redhat-package-manager`
- Add `application/x-sea`
- Add `audio/x-m4a`
- Add `audio/x-realaudio`
- Add `image/x-jng`
- Add `text/mathml`

119.40 1.12.0 / 2015-06-05

- Add `application/bdoc`
- Add `application/vnd.hyperdrive+json`
- Add `application/x-bdoc`
- Add extension `.rtf` to `text/rtf`

119.41 1.11.0 / 2015-05-31

- Add `audio/wav`
- Add `audio/wave`
- Add extension `.litcoffee` to `text/coffeescript`
- Add extension `.sfd-hdstx` to `application/vnd.hydrostatix.sof-data`
- Add extension `.n-gage` to `application/vnd.nokia.n-gage.symbian.install`

119.42 1.10.0 / 2015-05-19

- Add `application/vnd.balsamiq.bmpr`
- Add `application/vnd.microsoft.portable-executable`
- Add `application/x-ns-proxy-autoconfig`

119.43 1.9.1 / 2015-04-19

- Remove `.json` extension from `application/manifest+json`
  - This is causing bugs downstream

119.44 1.9.0 / 2015-04-19

- Add `application/manifest+json`
- Add `application/vnd.micro+json`
- Add `image/vnd.zbrush.pcx`
- Add `image/x-ms-bmp`

119.45 1.8.0 / 2015-03-13

- Add `application/vnd.citationstyles.style+xml`
- Add `application/vnd.fastcopy-disk-image`
- Add `application/vnd.gov.sk.xmldatacontainer+xml`
- Add extension `.jsonld` to `application/ld+json`

119.46 1.7.0 / 2015-02-08

- Add `application/vnd.gerber`
- Add `application/vnd.msa-disk-image`

119.47 1.6.1 / 2015-02-05

- Community extensions ownership transferred from `node-mime`

119.48 1.6.0 / 2015-01-29

- Add `application/jose`
- Add `application/jose+json`
- Add `application/json-seq`
- Add `application/jwk+json`
- Add `application/jwk-set+json`
- Add `application/jwt`
- Add `application/rdap+json`
- Add `application/vnd.gov.sk.e-form+xml`
- Add `application/vnd.ims.imsccv1p3`

119.49 1.5.0 / 2014-12-30

- Add `application/vnd.oracle.resource+json`
- Fix various invalid MIME type entries
  - `application/mbox+xml`
  - `application/oscp-response`
  - `application/vwg-multiplexed`
  - `audio/g721`

119.50 1.4.0 / 2014-12-21

- Add `application/vnd.ims.imsccv1p2`
- Fix various invalid MIME type entries
  - `application/vnd-acucobol`
  - `application/vnd-curl`
  - `application/vnd-dart`
  - `application/vnd-dxr`
  - `application/vnd-fdf`
  - `application/vnd-mif`
  - `application/vnd-sema`
  - `application/vnd-wap-wmlc`
  - `application/vnd.adobe.flash-movie`
  - `application/vnd.dece-zip`
  - `application/vnd.dvb_service`
  - `application/vnd.micrografx-igx`
  - `application/vnd.sealed-doc`
  - `application/vnd.sealed-eml`
  - `application/vnd.sealed-mht`
  - `application/vnd.sealed-ppt`
  - `application/vnd.sealed-tiff`
  - `application/vnd.sealed-xls`
  - `application/vnd.sealedmedia.softseal-html`
  - `application/vnd.sealedmedia.softseal-pdf`
  - `application/vnd.wap-slc`
  - `application/vnd.wap-wbxml`
  - `audio/vnd.sealedmedia.softseal-mpeg`
  - `image/vnd-djvu`
  - `image/vnd-svf`
  - `image/vnd-wap-wbmp`
  - `image/vnd.sealed-png`
  - `image/vnd.sealedmedia.softseal-gif`
  - `image/vnd.sealedmedia.softseal-jpg`
  - `model/vnd-dwf`
  - `model/vnd.parasolid.transmit-binary`
  - `model/vnd.parasolid.transmit-text`
  - `text/vnd-a`
  - `text/vnd-curl`
  - `text/vnd.wap-wml`
- Remove example template MIME types
  - `application/example`
  - `audio/example`
  - `image/example`
  - `message/example`
  - `model/example`
  - `multipart/example`
  - `text/example`
  - `video/example`

119.51 1.3.1 / 2014-12-16

- Fix missing extensions
  - `application/json5`
  - `text/hjson`

119.52 1.3.0 / 2014-12-07

- Add `application/a2l`
- Add `application/aml`
- Add `application/atfx`
- Add `application/atxml`
- Add `application/cdfx+xml`
- Add `application/dii`
- Add `application/json5`
- Add `application/lxf`
- Add `application/mf4`
- Add `application/vnd.apache.thrift.compact`
- Add `application/vnd.apache.thrift.json`
- Add `application/vnd.coffeescript`
- Add `application/vnd.enphase.envoy`
- Add `application/vnd.ims.imsccv1p1`
- Add `text/csv-schema`
- Add `text/hjson`
- Add `text/markdown`
- Add `text/yaml`

119.53 1.2.0 / 2014-11-09

- Add `application/cea`
- Add `application/dit`
- Add `application/vnd.gov.sk.e-form+zip`
- Add `application/vnd.tmd.mediaflex.api+xml`
- Type `application/epub+zip` is now IANA-registered

119.54 1.1.2 / 2014-10-23

- Rebuild database for `application/x-www-form-urlencoded` change

119.55 1.1.1 / 2014-10-20

- Mark `application/x-www-form-urlencoded` as compressible.

119.56 1.1.0 / 2014-09-28

- Add `application/font-woff2`

119.57 1.0.3 / 2014-09-25

- Fix engine requirement in package

119.58 1.0.2 / 2014-09-25

- Add `application/coap-group+json`
- Add `application/dcd`
- Add `application/vnd.apache.thrift.binary`
- Add `image/vnd.tencent.tap`
- Mark all JSON-derived types as compressible
- Update `text/vtt` data

119.59 1.0.1 / 2014-08-30

- Fix extension ordering

119.60 1.0.0 / 2014-08-30

- Add `application/atf`
- Add `application/merge-patch+json`
- Add `multipart/x-mixed-replace`
- Add ‘source: 'apache'’`metadata`
- `Add`source: 'iana'` metadata
- Remove badly-assumed charset data

Chapter 120

mime-db

This is a large database of mime types and information about them. It consists of a single, public JSON file and does not include any logic, allowing it to remain as un-opinionated as possible with an API. It aggregates data from the following sources:

- `http://www.iana.org/assignments/media-types/media-types.xhtml`
- `http://svn.apache.org/repos/asf/httpd/httpd/trunk/docs/conf/mime.types`
- `http://hg.nginx.org/nginx/raw-file/default/conf/mime.types`

120.1 Installation

```
npm install mime-db
```

120.1.1 Database Download

If you're crazy enough to use this in the browser, you can just grab the JSON file using `jsDelivr`. It is recommended to replace `master` with `a release tag` as the JSON format may change in the future.
```
https://cdn.jsdelivr.net/gh/jshttp/mime-db@master/db.json
```

120.2 Usage

```
var db = require('mime-db')
// grab data on .js files
var data = db['application/javascript']
```

120.3 Data Structure

The JSON file is a map lookup for lowercased mime types. Each mime type has the following properties:

- `.source` - where the mime type is defined. If not set, it's probably a custom media type.
  - `apache` - `Apache common media types`
  - `iana` - `IANA-defined media types`
  - `nginx` - `nginx media types`
- `.extensions[]` - known extensions associated with this mime type.
- `.compressible` - whether a file of this type can be gzipped.
- `.charset` - the default charset associated with this type, if any.

If unknown, every property could be `undefined`.

120.4 Contributing

To edit the database, only make PRs against `src/custom-types.json` or `src/custom-suffix.json`. The `src/custom-types.json` file is a JSON object with the MIME type as the keys and the values being an object with the following keys:

- `compressible` - leave out if you don't know, otherwise `true`/`false` to indicate whether the data represented by the type is typically compressible.
- `extensions` - include an array of file extensions that are associated with the type.
- `notes` - human-readable notes about the type, typically what the type is.
- `sources` - include an array of URLs of where the MIME type and the associated extensions are sourced from. This needs to be a `primary source`; links to type aggregating sites and Wikipedia are *not acceptable*.

To update the build, run `npm run build`.

120.4.1 Adding Custom Media Types

The best way to get new media types included in this library is to register them with the IANA. The community registration procedure is outlined in `RFC 6838 section 5`. Types registered with the IANA are automatically pulled into this library.
If that is not possible / feasible, they can be added directly here as a "custom" type. To do this, it is required to have a primary source that definitively lists the media type. If an extension is going to be listed as associateed with this media type, the source must definitively link the media type and extension as well.

Chapter 121

2.1.35 / 2022-03-12

- deps: `mime-db@1.52`.0
  - Add extensions from IANA for more `image/*` types
  - Add extension `.asc` to `application/pgp-keys`
  - Add extensions to various XML types
  - Add new upstream MIME types

121.1 2.1.34 / 2021-11-08

- deps: `mime-db@1.51`.0
  - Add new upstream MIME types

121.2 2.1.33 / 2021-10-01

- deps: `mime-db@1.50`.0
  - Add deprecated iWorks mime types and extensions
  - Add new upstream MIME types

121.3 2.1.32 / 2021-07-27

- deps: `mime-db@1.49`.0
  - Add extension `.trig` to `application/trig`
  - Add new upstream MIME types

121.4 2.1.31 / 2021-06-01

- deps: `mime-db@1.48`.0
  - Add extension `.mvt` to `application/vnd.mapbox-vector-tile`
  - Add new upstream MIME types

121.5 2.1.30 / 2021-04-02

- deps: `mime-db@1.47`.0
  - Add extension `.amr` to `audio/amr`
  - Remove ambigious extensions from IANA for `application/*+xml` types
  - Update primary extension to `.es` for `application/ecmascript`

121.6 2.1.29 / 2021-02-17

- deps: `mime-db@1.46`.0
  - Add extension `.amr` to `audio/amr`
  - Add extension `.m4s` to `video/iso.segment`
  - Add extension `.opus` to `audio/ogg`
  - Add new upstream MIME types

121.7 2.1.28 / 2021-01-01

- deps: `mime-db@1.45`.0
  - Add `application/ubjson` with extension `.ubj`
  - Add `image/avif` with extension `.avif`
  - Add `image/ktx2` with extension `.ktx2`
  - Add extension `.dbf` to `application/vnd.dbf`
  - Add extension `.rar` to `application/vnd.rar`
  - Add extension `.td` to `application/urc-targetdesc+xml`
  - Add new upstream MIME types
  - Fix extension of `application/vnd.apple.keynote` to be `.key`

121.8 2.1.27 / 2020-04-23

- deps: `mime-db@1.44`.0
  - Add charsets from IANA
  - Add extension `.cjs` to `application/node`
  - Add new upstream MIME types

121.9 2.1.26 / 2020-01-05

- deps: `mime-db@1.43`.0
  - Add `application/x-keepass2` with extension `.kdbx`
  - Add extension `.mxmf` to `audio/mobile-xmf`
  - Add extensions from IANA for `application/*+xml` types
  - Add new upstream MIME types

121.10 2.1.25 / 2019-11-12

- deps: `mime-db@1.42`.0
  - Add new upstream MIME types
  - Add `application/toml` with extension `.toml`
  - Add `image/vnd.ms-dds` with extension `.dds`

121.11 2.1.24 / 2019-04-20

- deps: `mime-db@1.40`.0
  - Add extensions from IANA for `model/*` types
  - Add `text/mdx` with extension `.mdx`

121.12 2.1.23 / 2019-04-17

- deps: mime-db1.39.0
  - Add extensions `.siv` and `.sieve` to `application/sieve`
  - Add new upstream MIME types

121.13 2.1.22 / 2019-02-14

- deps: mime-db1.38.0
  - Add extension `.nq` to `application/n-quads`
  - Add extension `.nt` to `application/n-triples`
  - Add new upstream MIME types

121.14 2.1.21 / 2018-10-19

- deps: mime-db1.37.0
  - Add extensions to HEIC image types
  - Add new upstream MIME types

121.15 2.1.20 / 2018-08-26

- deps: mime-db1.36.0
  - Add Apple file extensions from IANA
  - Add extensions from IANA for `image/*` types
  - Add new upstream MIME types

121.16 2.1.19 / 2018-07-17

- deps: mime-db1.35.0
  - Add extension `.csl` to `application/vnd.citationstyles.style+xml`
  - Add extension `.es` to `application/ecmascript`
  - Add extension `.owl` to `application/rdf+xml`
  - Add new upstream MIME types
  - Add UTF-8 as default charset for `text/turtle`

121.17 2.1.18 / 2018-02-16

- deps: mime-db1.33.0
  - Add `application/raml+yaml` with extension `.raml`
  - Add `application/wasm` with extension `.wasm`
  - Add `text/shex` with extension `.shex`
  - Add extensions for JPEG-2000 images
  - Add extensions from IANA for `message/*` types
  - Add new upstream MIME types
  - Update font MIME types
  - Update `text/hjson` to registered `application/hjson`

121.18 2.1.17 / 2017-09-01

- deps: mime-db1.30.0
  - Add `application/vnd.ms-outlook`
  - Add `application/x-arj`
  - Add extension `.mjs` to `application/javascript`
  - Add glTF types and extensions
  - Add new upstream MIME types
  - Add `text/x-org`
  - Add VirtualBox MIME types
  - Fix `source` records for `video/*` types that are IANA
  - Update `font/opentype` to registered `font/otf`

121.19 2.1.16 / 2017-07-24

- deps: mime-db1.29.0
  - Add `application/fido.trusted-apps+json`
  - Add extension `.wadl` to `application/vnd.sun.wadl+xml`
  - Add extension `.gz` to `application/gzip`
  - Add new upstream MIME types
  - Update extensions `.md` and `.markdown` to be `text/markdown`

121.20 2.1.15 / 2017-03-23

- deps: mime-db1.27.0
  - Add new mime types
  - Add `image/apng`

121.21 2.1.14 / 2017-01-14

- deps: mime-db1.26.0
  - Add new mime types

121.22 2.1.13 / 2016-11-18

- deps: mime-db1.25.0
  - Add new mime types

121.23 2.1.12 / 2016-09-18

- deps: mime-db1.24.0
  - Add new mime types
  - Add `audio/mp3`

121.24 2.1.11 / 2016-05-01

- deps: mime-db1.23.0
  - Add new mime types

121.25 2.1.10 / 2016-02-15

- deps: mime-db1.22.0
  - Add new mime types
  - Fix extension of `application/dash+xml`
  - Update primary extension for `audio/mp4`

121.26 2.1.9 / 2016-01-06

- deps: mime-db1.21.0
  - Add new mime types

121.27 2.1.8 / 2015-11-30

- deps: mime-db1.20.0
  - Add new mime types

121.28 2.1.7 / 2015-09-20

- deps: mime-db1.19.0
  - Add new mime types

121.29 2.1.6 / 2015-09-03

- deps: mime-db1.18.0
  - Add new mime types

121.30 2.1.5 / 2015-08-20

- deps: mime-db1.17.0
  - Add new mime types

121.31 2.1.4 / 2015-07-30

- deps: mime-db1.16.0
  - Add new mime types

121.32 2.1.3 / 2015-07-13

- deps: mime-db1.15.0
  - Add new mime types

121.33 2.1.2 / 2015-06-25

- deps: mime-db1.14.0
  - Add new mime types

121.34 2.1.1 / 2015-06-08

- perf: fix deopt during mapping

121.35 2.1.0 / 2015-06-07

- Fix incorrectly treating extension-less file name as extension
  - i.e. "path/to/json'`will no longer return`application/json
- `Fix`.charset(type)`to accept parameters`
- `Fix`.charset(type)` to match case-insensitive
- Improve generation of extension to MIME mapping
- Refactor internals for readability and no argument reassignment
- Prefer `application/*` MIME types from the same source
- Prefer any type over `application/octet-stream`
- deps: mime-db1.13.0
  - Add nginx as a source
  - Add new mime types

121.36 2.0.14 / 2015-06-06

- deps: mime-db1.12.0
  - Add new mime types

121.37 2.0.13 / 2015-05-31

- deps: mime-db1.11.0
  - Add new mime types

121.38 2.0.12 / 2015-05-19

- deps: mime-db1.10.0
  - Add new mime types

121.39 2.0.11 / 2015-05-05

- deps: mime-db1.9.1
  - Add new mime types

121.40 2.0.10 / 2015-03-13

- deps: mime-db1.8.0
  - Add new mime types

121.41 2.0.9 / 2015-02-09

- deps: mime-db1.7.0
  - Add new mime types
  - Community extensions ownership transferred from `node-mime`

121.42 2.0.8 / 2015-01-29

- deps: mime-db1.6.0
  - Add new mime types

121.43 2.0.7 / 2014-12-30

- deps: mime-db1.5.0
  - Add new mime types
  - Fix various invalid MIME type entries

121.44 2.0.6 / 2014-12-30

- deps: mime-db1.4.0
  - Add new mime types
  - Fix various invalid MIME type entries
  - Remove example template MIME types

121.45 2.0.5 / 2014-12-29

- deps: mime-db1.3.1
  - Fix missing extensions

121.46 2.0.4 / 2014-12-10

- deps: mime-db1.3.0
  - Add new mime types

121.47 2.0.3 / 2014-11-09

- deps: mime-db1.2.0
  - Add new mime types

121.48 2.0.2 / 2014-09-28

- deps: mime-db1.1.0
  - Add new mime types
  - Update charsets

121.49 2.0.1 / 2014-09-07

- Support Node.js 0.6

121.50 2.0.0 / 2014-09-02

- Use `mime-db`
- Remove `.define()`

121.51 1.0.2 / 2014-08-04

- Set charset=utf-8 for `text/javascript`

121.52 1.0.1 / 2014-06-24

- Add `text/jsx` type

121.53 1.0.0 / 2014-05-12

- Return `false` for unknown types
- Set charset=utf-8 for `application/json`

121.54 0.1.0 / 2014-05-02

- Initial release

Chapter 122

mime-types

The ultimate javascript content-type utility.
Similar to `the mime@1.x module`, except:

- **No fallbacks.** Instead of naively returning the first available type, `mime-types` simply returns `false`, so do ‘var type = mime.lookup('unrecognized') || 'application/octet-stream'.
- `No`new Mime()`business, so you could do`var lookup = require('mime-types').lookup.
- `No`.define()`functionality`
- `Bug fixes for`.lookup(path)`

Otherwise, the API is compatible with `mime` 1.x.

122.1 Install

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install mime-types
```

122.2 Adding Types

All mime types are based on `mime-db`, so open a PR there if you'd like to add mime types.

122.3 API

```
var mime = require('mime-types')
```

All functions return `false` if input is invalid or not found.

122.3.1 mime.lookup(path)

Lookup the content-type associated with a file.

```
mime.lookup('json') // 'application/json'
mime.lookup('.md') // 'text/markdown'
mime.lookup('file.html') // 'text/html'
mime.lookup('folder/file.js') // 'application/javascript'
mime.lookup('folder/.htaccess') // false
mime.lookup('cats') // false
```

122.3.2 mime.contentType(type)

Create a full content-type header given a content-type or extension. When given an extension, `mime.lookup` is used to get the matching content-type, otherwise the given content-type is used. Then if the content-type does not already have a `charset` parameter, `mime.charset` is used to get the default charset and add to the returned content-type.

```
mime.contentType('markdown') // 'text/x-markdown; charset=utf-8'
mime.contentType('file.json') // 'application/json; charset=utf-8'
mime.contentType('text/html') // 'text/html; charset=utf-8'
mime.contentType('text/html; charset=iso-8859-1') // 'text/html; charset=iso-8859-1'
// from a full path
mime.contentType(path.extname('/path/to/file.json')) // 'application/json; charset=utf-8'
```

122.3.3 mime.extension(type)

Get the default extension for a content-type.

```
mime.extension('application/octet-stream') // 'bin'
```

122.3.4 mime.charset(type)

Lookup the implied default charset of a content-type.

```
mime.charset('text/markdown') // 'UTF-8'
```

122.3.5 var type = mime.types[extension]

A map of content-types by extension.

122.3.6 [extensions...] = mime.extensions[type]

A map of extensions by content-type.

122.4 License

[MIT](LICENSE)

Chapter 123

Changelog

123.1 v1.6.0 (24/11/2017)

No changelog for this release.

123.2 v2.0.4 (24/11/2017)

- [**closed**] Switch to mime-score module for resolving extension contention issues. #182
- [**closed**] Update mime-db to 1.31.0 in v1.x branch #181

123.3 v1.5.0 (22/11/2017)

- [**closed**] need ES5 version ready in npm package #179
- [**closed**] mime-db no trace of iWork - pages / numbers / etc. #178
- [**closed**] How it works in brownser ? #176
- [**closed**] Missing `./Mime` #175
- [**closed**] Vulnerable Regular Expression #167

123.4 v2.0.3 (25/09/2017)

No changelog for this release.

123.5 v1.4.1 (25/09/2017)

- [**closed**] Issue when bundling with webpack #172

123.6 v2.0.2 (15/09/2017)

- [**V2**] fs.readFileSync is not a function #165
- [**closed**] The extension for video/quicktime should map to .mov, not .qt #164
- [**V2**] [v2 Feedback request] Mime class API #163
- [**V2**] [v2 Feedback request] Resolving conflicts over extensions #162
- [**V2**] Allow callers to load module with official, full, or no defined types. #161
- [**V2**] Use "facets" to resolve extension conflicts #160

- [**V2**] Remove fs and path dependencies #152
- [**V2**] Default content-type should not be application/octet-stream #139
- [**V2**] reset mime-types #124
- [**V2**] Extensionless paths should return null or false #113

123.7 v2.0.1 (14/09/2017)

- [**closed**] Changelog for v2.0 does not mention breaking changes #171
- [**closed**] MIME breaking with 'class' declaration as it is without 'use strict mode' #170

123.8 v2.0.0 (12/09/2017)

- [**closed**] woff and woff2 #168

123.9 v1.4.0 (28/08/2017)

- [**closed**] support for ac3 voc files #159
- [**closed**] Help understanding change from application/xml to text/xml #158
- [**closed**] no longer able to override mimetype #157
- [**closed**] application/vnd.adobe.photoshop #147
- [**closed**] Directories should appear as something other than application/octet-stream #135
- [**closed**] requested features #131
- [**closed**] Make types.json loading optional? #129
- [**closed**] Cannot find module './types.json' #120
- [**V2**] .wav files show up as "audio/x-wav" instead of "audio/x-wave" #118
- [**closed**] Don't be a pain in the ass for node community #108
- [**closed**] don't make default_type global #78
- [**closed**] mime.extension() fails if the content-type is parameterized #74

123.10 v1.3.6 (11/05/2017)

- [**closed**] .md should be text/markdown as of March 2016 #154
- [**closed**] Error while installing mime #153
- [**closed**] application/manifest+json #149
- [**closed**] Dynamic adaptive streaming over HTTP (DASH) file extension typo #141
- [**closed**] charsets image/png undefined #140
- [**closed**] Mime-db dependency out of date #130
- [**closed**] how to support plist #126
- [**closed**] how does .types file format look like? #123
- [**closed**] Feature: support for expanding MIME patterns #121
- [**closed**] DEBUG_MIME doesn't work #117

123.11 v1.3.4 (06/02/2015)

No changelog for this release.

123.12 v1.3.3 (06/02/2015)

No changelog for this release.

123.13 v1.3.1 (05/02/2015)

- [**closed**] Consider adding support for Handlebars .hbs file ending #111
- [**closed**] Consider adding support for hjson. #110
- [**closed**] Add mime type for Opus audio files #94
- [**closed**] Consider making the `Requesting New Types` information more visible #77

123.14 v1.3.0 (05/02/2015)

- [**closed**] Add common name? #114
- [**closed**] application/x-yaml #104
- [**closed**] Add mime type for WOFF file format 2.0 #102
- [**closed**] application/x-msi for .msi #99
- [**closed**] Add mimetype for gettext translation files #98
- [**closed**] collaborators #88
- [**closed**] getting errot in installation of mime module...any1 can help? #87
- [**closed**] should application/json's charset be utf8? #86
- [**closed**] Add "license" and "licenses" to package.json #81
- [**closed**] lookup with extension-less file on Windows returns wrong type #68

123.15 v1.2.11 (15/08/2013)

- [**closed**] Update mime.types #65
- [**closed**] Publish a new version #63
- [**closed**] README should state upfront that "application/octet-stream" is default for unknown extension #55
- [**closed**] Suggested improvement to the charset API #52

123.16 v1.2.10 (25/07/2013)

- [**closed**] Mime type for woff files should be application/font-woff and not application/x-font-woff #62
- [**closed**] node.types in conflict with mime.types #51

123.17 v1.2.9 (17/01/2013)

- [**closed**] Please update "mime" NPM #49
- [**closed**] Please add semicolon #46
- [**closed**] parse full mime types #43

123.18 v1.2.8 (10/01/2013)

- [**closed**] /js directory mime is application/javascript. Is it correct? #47
- [**closed**] Add mime types for lua code. #45

123.19 v1.2.7 (19/10/2012)

- [**closed**] cannot install 1.2.7 via npm #41
- [**closed**] Transfer ownership to @broofa #36
- [**closed**] it's wrong to set charset to UTF-8 for text #30
- [**closed**] Allow multiple instances of MIME types container #27

123.20 v1.2.5 (16/02/2012)

- [**closed**] When looking up a types, check hasOwnProperty #23
- [**closed**] Bump version to 1.2.2 #18
- [**closed**] No license #16
- [**closed**] Some types missing that are used by html5/css3 #13
- [**closed**] npm install fails for 1.2.1 #12
- [**closed**] image/pjpeg + image/x-png #10
- [**closed**] symlink #8
- [**closed**] gzip #2
- [**closed**] ALL CAPS filenames return incorrect mime type #1

Chapter 124

mime

Comprehensive MIME type mapping API based on mime-db module.

124.1 Install

Install with npm:

```
npm install mime
```

124.2 Contributing / Testing

```
npm run test
```

124.3 Command Line

```
mime [path_string]
```

E.g.

```
> mime scripts/jquery.js
application/javascript
```

124.4 API - Queries

124.4.1 mime.lookup(path)

Get the mime type associated with a file, if no mime type is found `application/octet-stream` is returned. Performs a case-insensitive lookup using the extension in `path` (the substring after the last '/' or '.'). E.g.

```
var mime = require('mime');
mime.lookup('/path/to/file.txt');         // => 'text/plain'
mime.lookup('file.txt');                  // => 'text/plain'
mime.lookup('.TXT');                      // => 'text/plain'
mime.lookup('htm');                       // => 'text/html'
```

124.4.2 mime.default_type

Sets the mime type returned when `mime.lookup` fails to find the extension searched for. (Default is `application/octet-stream`.)

124.4.3 mime.extension(type)

Get the default extension for `type`

```
mime.extension('text/html');                 // => 'html'
mime.extension('application/octet-stream');  // => 'bin'
```

124.4.4 mime.charsets.lookup()

Map mime-type to charset

```
mime.charsets.lookup('text/plain');        // => 'UTF-8'
```

(The logic for charset lookups is pretty rudimentary. Feel free to suggest improvements.)

124.5 API - Defining Custom Types

Custom type mappings can be added on a per-project basis via the following APIs.

124.5.1 mime.define()

Add custom mime/extension mappings

```
mime.define({
    'text/x-some-format':  ['x-sf', 'x-sft', 'x-sfml'],
    'application/x-my-type':  ['x-mt', 'x-mtt'],
    // etc ...
});
mime.lookup('x-sft');                 // => 'text/x-some-format'
```

The first entry in the extensions array is returned by `mime.extension()`. E.g.

```
mime.extension('text/x-some-format'); // => 'x-sf'
```

124.5.2 mime.load(filepath)

Load mappings from an Apache ".types" format file

```
mime.load('./my_project.types');
```

The .types file format is simple - See the `types` dir for examples.

Chapter 125

minimatch

A minimal matching utility.

This is the matching library used internally by npm.
It works by converting glob expressions into JavaScript `RegExp` objects.

125.1 Usage

```
var minimatch = require("minimatch")
minimatch("bar.foo", "*.foo") // true!
minimatch("bar.foo", "*.bar") // false!
minimatch("bar.foo", "*.+(bar|foo)", { debug:  true }) // true, and noisy!
```

125.2 Features

Supports these glob features:

- Brace Expansion
- Extended glob matching
- "Globstar" `**` matching

See:

- `man sh`
- `man bash`
- `man 3 fnmatch`
- `man 5 gitignore`

125.3 Minimatch Class

Create a minimatch object by instantiating the `minimatch.Minimatch` class.

```
var Minimatch = require("minimatch").Minimatch
var mm = new Minimatch(pattern, options)
```

125.3.1 Properties

- `pattern` The original pattern the minimatch object represents.
- `options` The options supplied to the constructor.
- `set` A 2-dimensional array of regexp or string expressions. Each row in the array corresponds to a brace-expanded pattern. Each item in the row corresponds to a single path-part. For example, the pattern `{a,b/c}/d` would expand to a set of patterns like:

```
[ [ a, d ]
, [ b, c, d ] ]
```

If a portion of the pattern doesn't have any "magic" in it (that is, it's something like `"foo"` rather than `fo*o?`), then it will be left as a string rather than converted to a regular expression.

- `regexp` Created by the `makeRe` method. A single regular expression expressing the entire pattern. This is useful in cases where you wish to use the pattern somewhat like `fnmatch(3)` with `FNM_PATH` enabled.
- `negate` True if the pattern is negated.
- `comment` True if the pattern is a comment.
- `empty` True if the pattern is `""`.

125.3.2 Methods

- `makeRe` Generate the `regexp` member if necessary, and return it. Will return `false` if the pattern is invalid.
- `match(fname)` Return true if the filename matches the pattern, or false otherwise.
- `matchOne(fileArray, patternArray, partial)` Take a `/`-split filename, and match it against a single row in the `regExpSet`. This method is mainly for internal use, but is exposed so that it can be used by a glob-walker that needs to avoid excessive filesystem calls.

All other methods are internal, and will be called as necessary.

125.3.3 minimatch(path, pattern, options)

Main export. Tests a path against the pattern using the options.

```
var isJS = minimatch(file, "*.js", { matchBase: true })
```

125.3.4 minimatch.filter(pattern, options)

Returns a function that tests its supplied argument, suitable for use with `Array.filter`. Example:

```
var javascripts = fileList.filter(minimatch.filter("*.js", {matchBase: true}))
```

125.3.5 minimatch.match(list, pattern, options)

Match against the list of files, in the style of fnmatch or glob. If nothing is matched, and options.nonull is set, then return a list containing the pattern itself.

```
var javascripts = minimatch.match(fileList, "*.js", {matchBase: true}))
```

125.3.6 minimatch.makeRe(pattern, options)

Make a regular expression object from the pattern.

125.4 Options

All options are `false` by default.

125.4.1 debug

Dump a ton of stuff to stderr.

125.4.2 nobrace

Do not expand `{a,b}` and `{1..3}` brace sets.

125.4.3 noglobstar

Disable ** matching against multiple folder names.

125.4.4 dot

Allow patterns to match filenames starting with a period, even if the pattern does not explicitly have a period in that spot.
Note that by default, `a/**/b` will **not** match `a/.d/b`, unless `dot` is set.

125.4.5 noext

Disable "extglob" style patterns like `+(a|b)`.

125.4.6 nocase

Perform a case-insensitive match.

125.4.7 nonull

When a match is not found by `minimatch.match`, return a list containing the pattern itself if this option is set. When not set, an empty list is returned if there are no matches.

125.4.8 matchBase

If set, then patterns without slashes will be matched against the basename of the path if it contains slashes. For example, `a?b` would match the path `/xyz/123/acb`, but not `/xyz/acb/123`.

125.4.9 nocomment

Suppress the behavior of treating `#` at the start of a pattern as a comment.

125.4.10 nonegate

Suppress the behavior of treating a leading `!` character as negation.

125.4.11 flipNegate

Returns from negate expressions the same as if they were not negated. (Ie, true on a hit, false on a miss.)

125.4.12 partial

Compare a partial path to a pattern. As long as the parts of the path that are present are not contradicted by the pattern, it will be treated as a match. This is useful in applications where you're walking through a folder structure, and don't yet have the full path, but want to ensure that you do not walk down paths that can never be a match.
For example,

```
minimatch('/a/b', '/a/*/c/d', { partial: true })  // true, might be /a/b/c/d
minimatch('/a/b', '/**/d', { partial: true })     // true, might be /a/b/.../d
minimatch('/x/y/z', '/a/**/z', { partial: true }) // false, because x !== a
```

125.4.13 allowWindowsEscape

Windows path separator `\` is by default converted to `/`, which prohibits the usage of `\` as a escape character. This flag skips that behavior and allows using the escape character.

125.5 Comparisons to other fnmatch/glob implementations

While strict compliance with the existing standards is a worthwhile goal, some discrepancies exist between minimatch and other implementations, and are intentional.
If the pattern starts with a ! character, then it is negated. Set the `nonegate` flag to suppress this behavior, and treat leading ! characters normally. This is perhaps relevant if you wish to start the pattern with a negative extglob pattern like `!(a|B)`. Multiple ! characters at the start of a pattern will negate the pattern multiple times.
If a pattern starts with `#`, then it is treated as a comment, and will not match anything. Use `\#` to match a literal `#` at the start of a line, or set the `nocomment` flag to suppress this behavior.
The double-star character `**` is supported by default, unless the `noglobstar` flag is set. This is supported in the manner of bsdglob and bash 4.1, where `**` only has special significance if it is the only thing in a path part. That is, `a/**/b` will match `a/x/y/b`, but `a/**b` will not.
If an escaped pattern has no matches, and the `nonull` flag is set, then minimatch.match returns the pattern as-provided, rather than interpreting the character escapes. For example, `minimatch.match([], "\\*a\\?")` will return `"\\*a\\?"` rather than `"*a?"`. This is akin to setting the `nullglob` option in bash, except that it does not resolve escaped pattern characters.
If brace expansion is not disabled, then it is performed before any other interpretation of the glob pattern. Thus, a pattern like `+(a|{b),c)}`, which would not be valid in bash or zsh, is expanded **first** into the set of `+(a|b)` and `+(a|c)`, and those patterns are checked for validity. Since those two are valid, matching proceeds.

Chapter 126

<tt>0.5.45</tt> <em>2024-02-04</em>

- Updated data to IANA TZDB `2024a`.

126.0.1 <tt>0.5.44</tt> <em>2023-12-29</em>

- Updated data to IANA TZDB `2023d`.
- Fixed `.valueOf()` to return `NaN` for invalid zoned objects (matching default `moment`) `#1082`.
- Performance improvements:
  - Use binary search when looking up zone information `#720`.
  - Avoid redundant checks in `tz.guess()`.
  - Avoid redundant `getZone()` calls in `.tz()`.

126.0.2 <tt>0.5.43</tt> <em>2023-03-31</em>

- Updated data to IANA TZDB `2023c`

126.0.3 <tt>0.5.42</tt> <em>2023-03-24</em>

- Updated data to IANA TZDB `2023b`

126.0.4 <tt>0.5.41</tt> <em>2023-02-25</em>

- Updated `moment` npm dependency to `2.29.4` to remove automated warnings about insecure dependencies. Moment Timezone still works with core Moment `2.9.0` and higher.
- Updated all dev dependencies including UglifyJS, which produces the minified builds.
- Added deprecation warning to the pre-built `moment-timezone-with-data-2012-2022` bundles `#1035`. Use the rolling `moment-timezone-with-data-10-year-range` files instead.

126.0.5 <tt>0.5.40</tt> <em>2022-12-11</em>

- Updated data to IANA TZDB `2022g`

126.0.6 <tt>0.5.39</tt> <em>2022-11-13</em>

- Updated data to IANA TZDB `2022f`

126.0.7 <tt>0.5.38</tt> <em>2022-10-15</em>

- Updated data to IANA TZDB `2022e`
- Added `moment.tz.dataVersion` property to TypeScript definitions `#930`
- Removed temporary `.tar.gz` files from npm releases `#1000`

126.0.8 <tt>0.5.37</tt> <em>2022-08-25</em>

- Re-publish npm package, because of extra folder present in 0.5.36, check https://github.↩ com/moment/moment-timezone/issues/999

126.0.9 <tt>0.5.36</tt> <em>2022-08-25</em>

- IANA TZDB 2022c
- improvements/fixes to data pipeline

126.0.10 <tt>0.5.35</tt> <em>2022-08-23</em>

- Fix command injection in data pipeline https://github.com/moment/moment-timezone/security/adviso GHSA-56x4-j7p9-fcf9
- Fix cleartext transmission of sensitive information https://github.com/moment/moment-timezone/security GHSA-v78c-4p63-2j6c

Thanks to the OpenSSF Alpha-Omega project for reporting these!

126.0.11 <tt>0.5.34</tt> <em>2021-11-10</em>

- Updated data to IANA TZDB `2021e`

126.0.12 <tt>0.5.33</tt> <em>2021-02-06</em>

- Updated data to IANA TZDB `2021a`

126.0.13 <tt>0.5.32</tt> <em>2020-11-14</em>

- Updated data to IANA TZDB `2020d`

126.0.14 <tt>0.5.31</tt> <em>2020-05-16</em>

- Fixed Travis builds for Node.js 4 and 6

126.0.15 <tt>0.5.30</tt> <em>2020-05-16</em>

- Updated data to IANA TZDB `2020a`
- Fixed typescript definitions

NOTE: You might need to un-install @types/moment-timezone. Check https://github.com/moment/moment-timezone for more info

126.0.16 <tt>0.5.29</tt> <em>2020-05-16</em>

- Merged fix of es6 module loading issue https://github.com/moment/moment-timezone/commit/1fd42349
- Merged PR with typescript declarations https://github.com/moment/moment-timezone/commit/ed529ea6
- Merged fixes to changelog https://github.com/moment/moment-timezone/commit/adb7d7b43c7328d8

126.0.17 <tt>0.5.28</tt> <em>2020-02-21</em>

Merged pull request #410 from @adgrace:

- Added a method `moment.tz.zonesForCountry(country_code)` which returns all timezones for the country
- Added a method `moment.tz(timezone_id).countries()` to get countries for some time zone
- Added a method `moment.tz.countries()` to get all country codes
- And as you know `moment.tz.zones()` already exists

126.0.18 <tt>0.5.27</tt> <em>2019-10-14</em>

- Updated data to IANA TZDB `2019c`

126.0.19 <tt>0.5.26</tt> <em>2019-06-06</em>

- Updated data to IANA TZDB `2019b`
- Fix: stabilize Array.sort #762

126.0.20 <tt>0.5.25</tt> <em>2019-04-17</em>

- Fix `moment.tz.dataVersion` to return `2019a` #742
- Update path in bower.json

126.0.21 <tt>0.5.24</tt> <em>2019-04-17</em>

- Updated data to IANA TZDB `2019a` #737
- Start shipping both a 1970-1930 file and a rolling 10-year file #614 #697
- Fixed bug where `_z` time zone name was not cleared with `.local()` or `.utcOffset(offset)` #738

126.0.22 <tt>0.5.23</tt> <em>2018-10-28</em>

- Fix minor issue with tz guessing in Russia #691

126.0.23 <tt>0.5.22</tt> <em>2018-10-28</em>

- Updated data to IANA TZDB `2018g` #689
- Fix issue with missing LMT entries for some zones, and fix data builds on Linux and Windows #308

126.0.24 <tt>0.5.21</tt> <em>2018-06-23</em>

- Bugfix: revert breaking change introduced in 0.5.18

126.0.25 <tt>0.5.20</tt> <em>2018-06-18</em>

- Bugfix: accidentally commented code

126.0.26 <tt>0.5.19</tt> <em>2018-06-18</em>

- Revert: moved moment to peerDependencies

126.0.27 <tt>0.5.18</tt> <em>2018-06-18</em>

- Return error when timezone name is not a string.
- Moved moment to peerDependencies #628
- Prefer nodejs to amd declaration #573

126.0.28 <tt>0.5.17</tt> <em>2018-05-12</em>

- Updated data to IANA TZDB `2018d`. #616

126.0.29 <tt>0.5.16</tt> <em>2018-04-18</em>

- Fixed Etc/UTC timezone recognition, updated tests. #599
- Updated minified files to contain IANA TZDB `2018d` data

126.0.30 <tt>0.5.15</tt> <em>2018-04-17</em>

- Updated data to IANA TZDB `2018d`. #596

126.0.31 <tt>0.5.14</tt> <em>2017-10-30</em>

- Ensure Intl response is valid when guessing time zone. #553
- Updated data to IANA TZDB `2017c`. #552
- Convert to tz keeping wall time #505
- Make all time zones available for guessing. #483
- zone.offset has been deprecated in favor of zone.utcOffset #398
- Check for timestamp formats when parsing #348

126.0.32 <tt>0.5.13</tt> <em>2017-04-04</em>

- Bumped version to address Bower cache issues with last release. #474
- (No actual changes otherwise)

126.0.33 <tt>0.5.12</tt> <em>2017-04-02</em>

- Updated data to IANA TZDB `2017b`. #422
- Build the truncated data file as 2012-2022 (+/- 5 years).

126.0.34 <tt>0.5.11</tt> <em>2016-12-23</em>

- Remove log statement when data is loaded twice. #352

126.0.35 <tt>0.5.10</tt> <em>2016-11-27</em>

- Updated data to IANA TZDB `2016j`. #422

126.0.36 <tt>0.5.9</tt> <em>2016-11-03</em>

- Fixed the output of `moment.tz.version`. #413

126.0.37 <tt>0.5.8</tt> <em>2016-11-03</em>

- Updated data to IANA TZDB `2016i`. #411

126.0.38 <tt>0.5.7</tt> <em>2016-10-21</em>

- Updated data to IANA TZDB `2016h`. #403

126.0.39 <tt>0.5.6</tt> <em>2016-10-08</em>

- Updated data to IANA TZDB `2016g`. #394

126.0.40 <tt>0.5.5</tt> <em>2016-07-24</em>

- Updated data to IANA TZDB `2016f`. #360

126.0.41 <tt>0.5.4</tt> <em>2016-05-03</em>

- Updated data to IANA TZDB `2016d`. #336
- Ignore the results from `Intl.DateTimeFormat().resolvedOptions().timeZone` if it is undefined. #322

126.0.42 <tt>0.5.3</tt> <em>2016-03-24</em>

- Updated data to IANA TZDB `2016c`. #321

126.0.43 <tt>0.5.2</tt> <em>2016-03-15</em>

- Updated data to IANA TZDB `2016b`. #315

126.0.44 <tt>0.5.1</tt> <em>2016-03-01</em>

- Updated data to IANA TZDB `2016a`. #299
- Fixed bug when `Date#toTimeString` did not return a known format. #302 #303
- Added lookup on `Intl.DateTimeFormat().resolvedOptions().timeZone` to `moment.↩ tz.guess()`. #304 #291

126.0.45 <tt>0.5.0</tt> <em>2015-12-28</em>

- Added support for guessing the user's timezone via `moment.tz.guess()`. #285
- Fixed UMD export issue when there was an html element with `id=exports`. #275
- Removed jspm specific dependencies from `package.json`. #284

126.0.46 <tt>0.4.1</tt> <em>2015-10-07</em>

- Updated data to IANA TZDB `2015e`. #253
- Updated data to IANA TZDB `2015f`. #253
- Updated data to IANA TZDB `2015g`. #255
- Added jspm dependencies for moment. #234
- Included builds directory in npm. #237
- Removed version field from bower.json. #230

126.0.47 <tt>0.4.0</tt> <em>2015-05-30</em>

- Updated data to IANA TZDB `2015b`. #201
- Updated data to IANA TZDB `2015c`. #214
- Updated data to IANA TZDB `2015d`. #214
- Updated zone getter to allow lazy unpacking to improve initial page load times. #216
- Added a `package.json` `jspm:main` entry point. #194
- Added `composer.json`. #222
- Added an error message when trying to load moment-timezone twice. #212

126.0.48 <tt>0.3.1</tt> <em>2015-03-16</em>

- Updated data to IANA TZDB `2015a`. #183

126.0.49 <tt>0.3.0</tt> <em>2015-01-13</em>

- *Breaking:* Added country data to the `meta/*.json` files. Restructured the data to support multiple countries per zone. #162
- Added the ability to set a default timezone for all new moments. #152
- Fixed a bug when passing a moment with an offset to `moment.tz`. #169
- Fixed a deprecation in moment core, changing `moment#zone` to `moment#utcOffset`. #168

126.0.50 <tt>0.2.5</tt> <em>2014-11-12</em>

- Updated data to IANA TZDB `2014j`. #151

126.0.51 <tt>0.2.4</tt> <em>2014-10-20</em>

- Updated data to IANA TZDB `2014i`. #142

126.0.52 <tt>0.2.3</tt> <em>2014-10-20</em>

- Updated data to IANA TZDB `2014h`. #141

126.0.53 <tt>0.2.2</tt> <em>2014-09-04</em>

- Updated data to IANA TZDB `2014g`. #126
- Added a warning when using `moment-timezone` with `moment<2.6.0`.

126.0.54 <tt>0.2.1</tt> <em>2014-08-02</em>

- Fixed support for `moment@2.8.1+`.

126.0.55 <tt>0.2.0</tt> <em>2014-07-21</em>

- Added the ability to configure whether ambiguous or invalid input is rolled forward or backward. #101
- Added `moment>=2.6.0` as a dependency in `bower.json`. #107
- Fixed getting the name of a zone that was added as a linked zone. #104
- Added an error message when a zone was not loaded. #106

126.0.56 <tt>0.1.0</tt> <em>2014-06-23</em>

- *Breaking:* Changed data format from Zones+Rules to just Zones. #82
- *Breaking:* Removed `moment.tz.{addRule,addZone,zoneExists,zones}` as they are no longer relevant with the new data format.
- Made library 20x faster. JSPerf results
- Completely rewrote internals to support new data format.
- Updated the data collection process to get data directly from http://www.iana.org/time-zones.
- Updated data to IANA TZDB `2014e`.
- Updated `bower.json` to use a browser specific `main:` entry point.
- Added built files with included data.
- Added support for accurately parsing input around DST changes. #93
- Added comprehensive documentation at momentjs.com/timezone/docs/.
- Added `moment.tz.link` for linking two identical zones.
- Added `moment.tz.zone` for getting a loaded zone.
- Added `moment.tz.load` for loading a bundled version of data from the IANA TZDB.
- Added `moment.tz.names` for getting the names of all the loaded timezones.
- Added `moment.tz.unpack` and `moment.tz.unpackBase60` for unpacking data.
- Added `moment-timezone-utils.js` for working with the packed and unpacked data.
- Fixed major memory leak. #79
- Fixed global export to allow use in web workers. #78
- Fixed global export in browser environments that define `window.module`. #76

126.0.57 <tt>0.0.6</tt> <em>2014-04-20</em>

- Fixed issue with preventing loading moment-timezone more than once. #75

126.0.58 <tt>0.0.5</tt> <em>2014-04-17</em>

- Improved performance with memoization. #39
- Published only necessary files to npm. #46
- Added better handling of timezones around DST. #53 #61 #70
- Added Browserify support. #41
- Added `moment.tz.zoneExists` #73
- Fixed cloning moments with a timezone. #71
- Prevent loading moment-timezone more than once. #74

126.0.59 <tt>0.0.3</tt> <em>2013-10-10</em>

- Added Bower support.
- Added support for newer versions of moment.
- Added support for constructing a moment with a string and zone.
- Added more links and timezone names in moment-timezone.json

126.0.60 <tt>0.0.1</tt> <em>2013-07-17</em>

- Initial version.

Chapter 127

<a href="http://momentjs.com/timezone/" >Moment Timezone</a>

[][license-url]
IANA Time zone support for Moment.js

127.1 Project Status

Moment-Timezone is an add-on for Moment.js. Both are considered legacy projects, now in maintenance mode. In most cases, you should choose a different library.
For more details and recommendations, please see `Project Status` in the Moment docs.
Thank you.

127.2 Resources

- `Documentation`
- /home/visi02/Universidad/Curso 24-25/Proyecto PBIO/Codigos_Generales_PBIO_Sprint0/Backend/nodejs/node↩ _modules/moment-timezone/changelog.md "Changelog"
- `Stack Overflow`

127.3 Examples

```
var june = moment("2014-06-01T12:00:00Z");
june.tz('America/Los_Angeles').format('ha z'); // 5am PDT
june.tz('America/New_York').format('ha z');    // 8am EDT
june.tz('Asia/Tokyo').format('ha z');          // 9pm JST
june.tz('Australia/Sydney').format('ha z');    // 10pm EST
var dec = moment("2014-12-01T12:00:00Z");
dec.tz('America/Los_Angeles').format('ha z');  // 4am PST
dec.tz('America/New_York').format('ha z');     // 7am EST
dec.tz('Asia/Tokyo').format('ha z');           // 9pm JST
dec.tz('Australia/Sydney').format('ha z');     // 11pm EST
```

127.4 License

Moment-timezone is freely distributable under the terms of the [MIT license][license-url].

Chapter 128

Changelog

128.0.1 2.30.1

- Release Dec 27, 2023
- Revert https://github.com/moment/moment/pull/5827, because it's breaking a lot of TS code.

128.0.2 2.30.0 <a href="https://gist.github.com/ichernev/e277bcd1f0eeabb834f60a777237925a" >Full changelog</a>

- Release Dec 26, 2023

128.0.3 2.29.4

- Release Jul 6, 2022
  - #6015 [bugfix] Fix ReDoS in preprocessRFC2822 regex

128.0.4 2.29.3 <a href="https://gist.github.com/ichernev/edebd440f49adcaec72e5e77b791d8be" >Full changelog</a>

- Release Apr 17, 2022
  - #5995 [bugfix] Remove const usage
  - #5990 misc: fix advisory link

128.0.5 2.29.2 <a href="https://gist.github.com/ichernev/1904b564f6679d9aac1ae08ce13bc45c" >See full changelog</a>

- Release Apr 3 2022

Address https://github.com/moment/moment/security/advisories/GHSA-8hfj-j24r-96c4

128.0.6 2.29.1 <a href="https://gist.github.com/marwahaha/cc478ba01a1292ab4bd4e861d164d99b" >See full changelog</a>

- Release Oct 6, 2020

Updated deprecation message, bugfix in hi locale

128.0.7 2.29.0 <a href="https://gist.github.↩ com/marwahaha/b0111718641a6461800066549957ec14" >See full changelog</a>

- Release Sept 22, 2020

New locales (es-mx, bn-bd). Minor bugfixes and locale improvements. More tests. Moment is in maintenance mode. Read more at this link: https://momentjs.com/docs/#/-project-status/

128.0.8 2.28.0 <a href="https://gist.github.com/marwahaha/028fd6c2b2470b2804857cfd63c0e94f" >See full changelog</a>

- Release Sept 13, 2020

Fix bug where .format() modifies original instance, and locale updates

128.0.9 2.27.0 <a href="https://gist.github.com/marwahaha/5100c9c2f42019067b1f6cefc333daa7" >See full changelog</a>

- Release June 18, 2020

Added Turkmen locale, other locale improvements, slight TypeScript fixes

128.0.10 2.26.0 <a href="https://gist.github.↩ com/marwahaha/0725c40740560854a849b096ea7b7590" >See full changelog</a>

- Release May 19, 2020

TypeScript fixes and many locale improvements

128.0.11 2.25.3

- Release May 4, 2020

Remove package.json module property. It looks like webpack behaves differently for modules loaded via module vs jsnext:main.

128.0.12 2.25.2

- Release May 4, 2020

This release includes ES Module bundled moment, separate from it's source code under dist/ folder. This might alleviate issues with finding the `./locale subfolder for loading locales. This might also mean now webpack will bundle all locales automatically, unless told otherwise.

128.0.13 2.25.1

- Release May 1, 2020

This is a quick patch release to address some of the issues raised after releasing 2.25.0.

- 2e268635 [misc] Revert #5269 due to webpack warning
- 226799e1 [locale] fil: Fix metadata comment
- a83a521 [bugfix] Fix typeoff usages
- e324334 [pkg] Add ts3.1-typings in npm package
- 28cc23e [misc] Remove deleted generated locale en-SG

128.0.14 2.25.0 <a href="https://gist.github.com/ichernev/6148e64df2427e455b10ce6a18de1a65" >See full changelog</a>

- Release May 1, 2020
- #4611 022dc038 [feature] Support for strict string parsing, fixes #2469
- #4599 4b615b9d [feature] Add support for eras in en and jp
- #4296 757d4ff8 [feature] Accept custom relative thresholds in duration.humanize
- 18 bigfixes
- 36 locale fixes
- 5 new locales (oc-lnc, zh-mo, en-in, gom-deva, fil)

128.0.15 2.24.0 <a href="https://gist.github.↩com/marwahaha/12366fe45bee328f33acf125d4cd540e" >See full changelog</a>

- Release Jan 21, 2019
- #4338 [bugfix] Fix startOf/endOf DST issues while boosting performance
- #4553 [feature] Add localeSort param to Locale weekday methods
- #4887 [bugfix] Make Duration::as work with quarters
- 3 new locales (it-ch, ga, en-SG)
- Lots of locale improvements

128.0.16 2.23.0 <a href="https://gist.github.↩com/marwahaha/eadb7ac11b761290399a576f8b2419a5" >See full changelog</a>

- Release Dec 12, 2018
- #4863 [new locale] added Kurdish language (ku)
- #4417 [bugfix] isBetween should return false for invalid dates
- #4700 [bugfix] Fix #4698: Use ISO WeekYear for HTML5_FMT.WEEK
- #4563 [feature] Fix #4518: Add support to add/subtract ISO weeks
- other locale changes, build process changes, typos

128.0.17 2.22.2 <a href="https://gist.github.↩com/marwahaha/4d992c13c2dbc0f59d4d8acae1dc6d3a" >See full changelog</a>

- Release May 31, 2018
- #4564 [bugfix] Avoid using trim()
- #4453 [bugfix] Treat periods as periods, not regex-anything period, for weekday parsing in strict mode.
- Minor locale improvements (pa-in, be, az)

128.0.18 2.22.1 <a href="https://gist.github.↩ com/marwahaha/ff2cd13d0eda08afb7a237b10aae558c" >See full changelog</a>

- Release Apr 14, 2018
- #4495 [bugfix] Added HTML5_FMT to moment.d.ts
- Minor locale improvements
- QUnit upgrade and coveralls reporting

128.0.19 2.22.0 <a href="https://gist.github.↩ com/marwahaha/ae895025dac3f0641fa9ec2e36d282bb" >See full changelog</a>

- Release Mar 30, 2018
- #4423 [new locale] Added Mongolian locale mn
- Various locale improvements
- Minor misc changes

128.0.20 2.21.0 <a href="https://gist.github.↩ com/marwahaha/80d19ef882b71df1948df7865efdd40e" >See full changelog</a>

- Release Mar 2, 2018
- #4391 [bugfix] Fix #4390: use offset properly in toISOString
- #4310 [bugfix] Fix #3883 lazy load parentLocale in defineLocale, fallback to global if missing
- #4085 [misc] Print console warning when setting non-existent locales
- #4371 [misc] fix deprecated rollup options
- New locales: ug-cn, en-il, tg
- Various locale improvements

128.0.21 2.20.1 <a href="https://gist.github.↩ com/marwahaha/d72c1cb22076373be889b16272cbd187" >See changelog</a>

- Release Dec 18, 2017
- #4359 [locale] Fix Arabic locale for months (again)
- #4357 [misc] Add optional parameter keepOffset to toISOString

128.0.22 2.20.0 <a href="https://gist.github.↩ com/marwahaha/e0d4135fbf8bb75fa85c4aa2bddc5031" >See full changelog</a>

- Release Dec 16, 2017
- #4312 [bugfix] Fix #4251: Avoid RFC2822 in utc() test
- #4240 [bugfix] Fix incorrect strict parsing with full-width parentheses
- #4341 [feature] Prevent toISOString converting to UTC (issue #1751)

- #4154 [feature] add format constants to support output to HTML5 input type formats (see #3928)
- #4143 [new locale] mt: Maltese language
- #4183 [locale] Relative seconds i18n
- Various other locale improvements

128.0.23 2.19.4 <a href="https://gist.github.↩ com/marwahaha/d3b7b0ddf4bdae512244f16e8cc59efb" >See changelog</a>

- Release Dec 10, 2017
- #4332 [bugfix] Fix weekday verification for UTC and offset days (fixes #4227)
- #4336 [bugfix] Fix #4334: Remove unused function call argument
- #4246 [misc] Add 'ss' relative time key to typescript definition

128.0.24 2.19.3 <a href="https://gist.github.↩ com/marwahaha/3654006bc0c2e522451c08d12c0bfabf" >See changelog</a>

- Release Nov 29, 2017
- #4326 [bugfix] Fix for ReDOS vulnerability (see #4163)
- #4289 [misc] Fix spelling and formatting for U.S. for es-us

128.0.25 2.19.2 <a href="https://gist.github.com/ichernev/76b1a3f33d3a8ff9665ce434a45221d0" >See changelog (it's the same >:D)</a>

- Release Nov 11, 2017
- #4255 [bugfix] Fix year setter for random days in a leap year, fixes #4238
- #4242 [bugfix] updateLocale now tries to load parent, fixes #3626

128.0.26 2.19.1

- Release Oct 11, 2017

Make react native and webpack both work

- #4225 #4226 #4232

128.0.27 2.19.0 <a href="https://gist.github.com/ichernev/5f3f4eb02761b4f765a0cccf02cec603" >See full changelog</a>

- Release Oct 10, 2017

128.1 Fix React Native 0.49+ crash

- #4213 [critical] Rename dynamic require to avoid React Native crash
- #4214 [fixup] Move require rename inside try/catch, fixes #4213

128.2 Features

- #3735 [feature] Ignore NaN values in setters
- #4106 [fixup] Drop isNumeric utility fn, fixes #3735
- #4080 [feature] Implement a clone method for durations, fixes #4078
- #4215 [misc] TS: Add duration.clone(), for #4080

128.3 Packaging

- #4003 [pkg] bower: Remove tests from package
- #3904 [pkg] jsnext:main -> module in package.json
- #4060 [pkg] Account for new rollup interface

Bugfixes, new locales, locale fixes etc...

128.3.1 2.18.1

- Release Mar 22, 2017
- #3853 [misc] Fix invalid whitespace character causing inability to parse moment.js

128.3.2 2.18.0 <a href="https://gist.github.com/ichernev/78920c5a1e419fb28c6e4546d1b7235c" >See full changelog</a>

- Release Mar 18, 2017

128.4 Features

- #3708 [feature] RFC2822 parsing
- #3611 [feature] Durations gain validity
- #3738 [feature] Enable relative time for multiple seconds, request #2558
- #3766 [feature] Add support for k and kk format parsing

128.5 Bugfixes

- #3643 [bugfix] Fixes #3520, parseZone incorrectly handled minutes under 16
- #3710 [bugfix] Fixes #3632, toISOString returns null for invalid date
- #3787 [bugfix] Fixes #3717, ensure day-of-year is non-zero
- #3780 [bugfix] Fixes #3765: Ensure year 0 is formatted with YYYY
- #3806 [bugfix] Fixes #3805, fix locale month getters for standalone/format cases

7 new locales, many locale improvements and some misc changes

128.5.1 2.17.1 <a href="https://gist.github.com/ichernev/f38280b2b29c4932914a6d3a4e50bfb2" >Also available here</a>

- Release Dec 03, 2016
- #3638 [misc] TS: Make typescript definitions work with 1.x
- #3628 [misc] Adds "sign CLA" link to `CONTRIBUTING.md`
- #3640 [misc] Fix locale issues

128.5.2 2.17.0 <a href="https://gist.github.com/ichernev/ed58f76fb95205eeac653d719972b90c" >Also available here</a>

- Release Nov 22, 2016
- #3435 [new locale] yo: Yoruba (Nigeria) locale
- #3595 [bugfix] Fix accidental reference to global "value" variable
- #3506 [bugfix] Fix invalid moments returning valid dates to method calls
- #3563 [locale] ca: Change future relative time
- #3504 [tests] Fixes #3463, parseZone not handling Z correctly (tests only)
- #3591 [misc] typescript: update typescript to 2.0.8, add strictNullChecks=true
- #3597 [misc] Fixed capitalization in nuget spec

128.5.3 2.16.0 <a href="https://gist.github.com/ichernev/17bffc1005a032cb1a8ac4c1558b4994" >See full changelog</a>

- Release Nov 9, 2016

128.6 Features

- #3530 [feature] Check whether input is date before checking if format is array
- #3515 [feature] Fix #2300: Default to current week.

128.7 Bugfixes

- #3546 [bugfix] Implement lazy-loading of child locales with missing prents
- #3523 [bugfix] parseZone should handle UTC
- #3502 [bugfix] Fix #3500: ISO 8601 parsing should match the full string, not the beginning of the string.
- #3581 [bugfix] Fix parseZone, redo #3504, fix #3463

128.8 New Locales

- #3416 [new locale] nl-be: Dutch (Belgium) locale
- #3393 [new locale] ar-dz: Arabic (Algeria) locale
- #3342 [new locale] tet: Tetun Dili (East Timor) locale

And more locale, build and typescript improvements

128.8.1 2.15.2

- Release Oct 23, 2016
- #3525 Speedup month standalone/format regexes **(IMPORTANT)**
- #3466 Fix typo of Javanese

128.8.2 2.15.1

- Release Sept 20, 2016
- #3438 Fix locale autoload, revert #3344

128.8.3 2.15.0 <a href="https://gist.github.com/ichernev/10e1c5bf647545c72ca30e9628a09ed3" >See full changelog</a>

- Release Sept 12, 2016

128.9 New Locales

- #3255 [new locale] mi: Maori language
- #3267 [new locale] ar-ly: Arabic (Libya) locale
- #3333 [new locale] zh-hk: Chinese (Hong Kong) locale

128.10 Bugfixes

- #3276 [bugfix] duration: parser: Support ms durations in .NET syntax
- #3312 [bugfix] locales: Enable locale-data getters without moment (fixes #3284)
- #3381 [bugfix] parsing: Fix parseZone without timezone in string, fixes #3083
- #3383 [bugfix] toJSON: Fix isValid so that toJSON works after a moment is frozen
- #3427 [bugfix] ie8: Fix IE8 (regression in 2.14.x)

128.11 Packaging

- #3299 [pkg] npm: Do not include .npmignore in npm package
- #3273 [pkg] jspm: Include moment.d.ts file in package
- #3344 [pkg] exports: use module.require for nodejs

Also some locale and typescript improvements

128.11.1 2.14.1

- Release July 20, 2016
- #3280 Fix typescript definitions

128.11.2 2.14.0 <a href="https://gist.github.com/ichernev/812e79ac36a7829a22598fe964bfc18a" >See full changelog</a>

- Release July 20, 2016

128.12 New Features

- #3233 Introduce month.isFormat for format/standalone discovery
- #2848 Allow user to get/set the rounding method used when calculating relative time
- #3112 optimize configFromStringAndFormat
- #3147 Call calendar format function with moment context
- #3160 deprecate isDSTShifted
- #3175 make moment calendar extensible with ad-hoc options
- #3191 toDate returns a copy of the internal date object
- #3192 Adding support for rollup import.
- #3238 Handle empty object and empty array for creation as now
- #3082 Use relative AMD moment dependency

128.13 Bugfixes

- #3241 Escape all 24 mixed pieces, not only first 12 in computeMonthsParse
- #3008 Object setter orders sets based on size of unit
- #3177 Bug Fix #2704 - isoWeekday(String) inconsistent with isoWeekday(Number)
- #3230 fix passing date with format string to ignore format string
- #3232 Fix negative 0 in certain diff cases
- #3235 Use proper locale inheritance for the base locale, fixes #3137

Plus es-do locale and locale bugfixes

128.13.1 2.13.0 <a href="https://gist.github.com/ichernev/0132fcf5b61f7fc140b0bb0090480d49" >See full changelog</a>

- Release April 18, 2016

128.14 Enhancements:

- #2982 Add 'date' as alias to 'day' for startOf() and endOf().
- #2955 Add parsing negative components in durations when ISO 8601
- #2991 isBetween support for both open and closed intervals
- #3105 Add localeSorted argument to weekday listers
- #3102 Add k and kk formatting tokens

128.15 Bugfixes

- #3109 Fix #1756 Resolved thread-safe issue on server side.
- #3078 Fix parsing for months/weekdays with weird characters
- #3098 Use Z suffix when in UTC mode (#3020)
- #2995 Fix floating point rounding errors in durations
- #3059 fix bug where diff returns -0 in month-related diffs
- #3045 Fix mistaking any input for 'a' token
- #2877 Use explicit .valueOf() calls instead of coercion
- #3036 Year setter should keep time when DST changes

Plus 3 new locales and locale fixes.

128.15.1 2.12.0 <a href="https://gist.github.com/ichernev/6e5bfdf8d6522fc4ac73" >See full changelog</a>

- Release March 7, 2016

128.16 Enhancements:

- #2932 List loaded locales
- #2818 Parse ISO-8061 duration containing both day and week values
- #2774 Implement locale inheritance and locale updating

128.17 Bugfixes:

- #2970 change add subtract to handle decimal values by rounding
- #2887 Fix toJSON casting of invalid moment
- #2897 parse string arguments for month() correctly, closes #2884
- #2946 Fix usage suggestions for min and max

128.18 New locales:

- #2917 Locale Punjabi(Gurmukhi) India format conversion

And more

128.18.1 2.11.2 (Fix ReDoS attack vector)

- Release February 7, 2016
- #2939 use full-string match to speed up aspnet regex match

128.18.2 2.11.1 <a href="https://gist.github.com/ichernev/8ec3ee25b749b4cff3c2" >See full changelog</a>

- Release January 9, 2016

128.19 Bugfixes:

- #2881 Revert "Merge pull request #2746 from mbad0la:develop" Sep->Sept
- #2868 Add format and parse token Y, so it actually works
- #2865 Use typeof checks for undefined for global variables
- #2858 Fix Date mocking regression introduced in 2.11.0
- #2864 Include changelog in npm release
- #2830 dep: add grunt-cli
- #2869 Fix months parsing for some locales

128.19.1 2.11.0 <a href="https://gist.github.com/ichernev/6594bc29719dde6b2f66" >See full changelog</a>

- Release January 4, 2016
- #2624 Proper handling of invalid moments
- #2634 Fix strict month parsing issue in cs,ru,sk
- #2735 Reset the locale back to 'en' after defining all locales in min/locales.js
- #2702 Week rework
- #2746 Changed September Abbreviation to "Sept" in locale-specific english files and default locale file
- #2646 Fix #2645 - invalid dates pre-1970
- #2641 Implement basic format and comma as ms separator in ISO 8601
- #2665 Implement stricter weekday parsing
- #2700 Add [Hh]mm and [Hh]mmss formatting tokens, so you can parse 123 with hmm for example
- #2565 #2835 Expose arguments used for moment creation with creationData (fix #2443)
- #2648 fix issue #2640: support instanceof operator
- #2709 Add isSameOrAfter and isSameOrBefore comparison methods
- #2721 Fix moment creation from object with strings values
- #2740 Enable 'd hh:mm:ss.sss' format for durations
- #2766 #2833 Alternate Clock Source Support

128.19.2 2.10.6

- Release July 28, 2015

#2515 Fix regression introduced in `2.10.5` related to `moment.ISO_8601` parsing.

128.19.3 2.10.5 <a href="https://gist.github.com/ichernev/6ec13ac7efc396da44b2" >See full changelog</a>

- Release July 26, 2015

Important changes:

- #2357 Improve unit bubbling for ISO dates this fixes day to year conversions to work around end-of-year (∼365 days). As a side effect 365 days is 11 months and 30 days, and 366 days is one year.
- #2438 Fix inconsistent moment.min and moment.max results Return invalid result if any of the inputs is invalid
- #2494 Fix two digit year parsing with YYYY format This brings the benefits of YY to YYYY
- #2368 perf: use faster form of copying dates, across the board improvement

128.19.4 2.10.3 <a href="https://gist.github.com/ichernev/f264b9bed5b00f8b1b7f" >See full changelog</a>

- Release May 13, 2015
- add `moment.fn.to` and `moment.fn.toNow` (similar to `from` and `fromNow`)
- new locales (Sinhalese (si), Montenegrin (me), Javanese (ja))
- performance improvements

128.19.5 2.10.2

- Release April 9, 2015
- fixed moment-with-locales in browser env caused by esperanto change

128.19.6 2.10.1

- regression: Add moment.duration.fn back

128.19.7 2.10.0

Ported code to es6 modules.

128.19.8 2.9.0 <a href="https://gist.github.com/ichernev/0c9a9b49951111a27ce7" >See full changelog</a>

- Release January 8, 2015

languages:

- 2104 Frisian (fy) language file with unit test
- 2097 add ar-tn locale

deprecations:

- 2074 Implement `moment.fn.utcOffset`, deprecate `moment.fn.zone`

features:

- 2088 add moment.fn.isBetween
- 2054 Call updateOffset when creating moment (needed for default timezone in moment-timezone)
- 1893 Add moment.isDate method

- 1825 Implement toJSON function on Duration
- 1809 Allowing moment.set() to accept a hash of units
- 2128 Add firstDayOfWeek, firstDayOfYear locale getters
- 2131 Add quarter diff support

Some bugfixes and language improvements – full changelog

128.19.9 2.8.4 <a href="https://gist.github.com/ichernev/a4fcb0a46d74e4b9b996" >See full changelog</a>

- Release November 19, 2014

Features:

- #2000 Add LTS localised format that includes seconds
- #1960 added formatToken 'x' for unix offset in milliseconds #1938
- #1965 Support 24:00:00.000 to mean next day, at midnight.
- #2002 Accept 'date' key when creating moment with object
- #2009 Use native toISOString when we can

Some bugfixes and language improvements – full changelog

128.19.10 2.8.3

- Release September 5, 2014

Bugfixes:

- #1801 proper pluralization for Arabic
- #1833 improve spm integration
- #1871 fix zone bug caused by Firefox 24
- #1882 Use hh:mm in Czech
- #1883 Fix 2.8.0 regression in duration as conversions
- #1890 Faster travis builds
- #1892 Faster isBefore/After/Same
- #1848 Fix flaky month diffs
- #1895 Fix 2.8.0 regression in moment.utc with format array
- #1896 Support setting invalid instance locale (noop)
- #1897 Support moment([str]) in addition to moment([int])

128.19.11 2.8.2

- Release August 22, 2014

Minor bugfixes:

- #1874 use `Object.prototype.hasOwnProperty` instead of `obj.hasOwnProperty` (ie8 bug)
- #1873 add `duration#toString()`
- #1859 better month/weekday names in norwegian
- #1812 meridiem parsing for greek
- #1804 spanish del -> de
- #1800 korean LT improvement

128.19.12 2.8.1

- Release August 1, 2014
- bugfix #1813: fix moment().lang([key]) incompatibility

128.19.13 2.8.0 <a href="https://gist.github.com/ichernev/ac3899324a5fa6c8c9b4" >See changelog</a>

- Release July 31, 2014
- incompatible changes
  - #1761: moments created without a language are no longer following the global language, in case it changes. Only newly created moments take the global language by default. In case you're affected by this, wait, comment on #1797 and wait for a proper reimplementation
  - #1642: 45 days is no longer "a month" according to humanize, cutoffs for month, and year have changed. Hopefully your code does not depend on a particular answer from humanize (which it shouldn't anyway)
  - #1784: if you use the human readable English datetime format in a weird way (like storing them in a database) that would break when the format changes you're at risk.
- deprecations (old behavior will be dropped in 3.0)
  - #1761 `lang` is renamed to `locale`, `langData` -> `localeData`. Also there is now `define←Locale` that should be used when creating new locales
  - #1763 `add(unit, value)` and `subtract(unit, value)` are now deprecated. Use `add(value, unit)` and `subtract(value, unit)` instead.
  - #1759 rename `duration.toIsoString` to `duration.toISOString`. The js standard library and moment's `toISOString` follow that convention.
- new locales
  - #1789 Tibetan (bo)
  - #1786 Africaans (af)
  - #1778 Burmese (my)
  - #1727 Belarusian (be)
- bugfixes, locale bugfixes, performance improvements, features

128.19.14 2.7.0 <a href="https://gist.github.com/ichernev/b0a3d456d5a84c9901d7" >See changelog</a>

- Release June 12, 2014
- new languages
  - #1678 Bengali (bn)
  - #1628 Azerbaijani (az)
  - #1633 Arabic, Saudi Arabia (ar-sa)
  - #1648 Austrian German (de-at)
- features
  - #1663 configurable relative time thresholds
  - #1554 support anchor time in moment.calendar
  - #1693 support moment.ISO_8601 as parsing format
  - #1637 add moment.min and moment.max and deprecate min/max instance methods
  - #1704 support string value in add/subtract
  - #1647 add spm support (package manager)
- bugfixes

128.19.15 2.6.0 <a href="https://gist.github.com/ichernev/10544682" >See changelog</a>

- Release April 12 , 2014
- languages
  - #1529 Serbian-Cyrillic (sr-cyr)
  - #1544, #1546 Khmer Cambodia (km)
- features
  - #1419, #1468, #1467, #1546 better handling of timezone-d moments around DST
  - #1462 add weeksInYear and isoWeeksInYear
  - #1475 support ordinal parsing
  - #1499 composer support
  - #1577, #1604 put Date parsing in moment.createFromInputFallback so it can be properly deprecated and controlled in the future
  - #1545 extract two-digit year parsing in moment.parseTwoDigitYear, so it can be overwritten
  - #1590 (see #1574) set AMD global before module definition to better support non AMD module dependencies used in AMD environment
  - #1589 remove global in Node.JS environment (was not working before, nobody complained, was scheduled for removal anyway)
  - #1586 support quarter setting and parsing
- 18 bugs fixed

128.19.16 2.5.1

- Release January 22, 2014
- languages
  - #1392 Armenian (hy-am)
- bugfixes
  - #1429 fixes #1423 weird chrome-32 bug with js object creation
  - #1421 remove html entities from Welsh
  - #1418 fixes #1401 improved non-padded tokens in strict matching
  - #1417 fixes #1404 handle buggy moment object created by property cloning
  - #1398 fixes #1397 fix Arabic-like week number parsing
  - #1396 add leftZeroFill(4) to GGGG and gggg formats
  - #1373 use lowercase for months and days in Catalan
- testing
  - #1374 run tests on multiple browser/os combos via SauceLabs and Travis

128.19.17 2.5.0 <a href="https://gist.github.com/ichernev/8104451" >See changelog</a>

- Release Dec 24, 2013
- New languages
  - Luxemburish (lb) 1247
  - Serbian (rs) 1319

  - Tamil (ta) 1324
  - Macedonian (mk) 1337
- Features
  - 1311 Add quarter getter and format token `Q`
  - 1303 strict parsing now respects number of digits per token (fix 1196)
  - 0d30bb7 add jspm support
  - 1347 improve zone parsing
  - 1362 support merideam parsing in Korean
- 22 bugfixes

128.19.18 2.4.0

- Release Oct 27, 2013
- **Deprecate** globally exported moment, will be removed in next major
- New languages
  - Farose (fo) #1206
  - Tagalog/Filipino (tl-ph) #1197
  - Welsh (cy) #1215
- Bugfixes
  - properly handle Z at the end of iso RegExp #1187
  - chinese meridian time improvements #1076
  - fix language tests #1177
  - remove some failing tests (that should have never existed :)) #1185 #1183
  - handle russian noun cases in weird cases #1195

128.19.19 2.3.1

- Release Oct 9, 2013

Removed a trailing comma [1169] and fixed a bug with `months`, `weekdays` getters #1171.

128.19.20 2.3.0 <a href="https://gist.github.com/ichernev/6864354" >See changelog</a>

- Release Oct 7, 2013

Changed isValid, added strict parsing. Week tokens parsing.

128.19.21 2.2.1

- Release Sep 12, 2013

Fixed bug in string prototype test. Updated authors and contributors.

128.19.22 2.2.0 <a href="https://gist.github.com/ichernev/00f837a9baf46a3565e4" >See changelog</a>

- Release Sep 11, 2013

Added bower support.
Language files now use UMD.
Creating moment defaults to current date/month/year.
Added a bundle of moment and all language files.

128.19.23 2.1.0 <a href="https://gist.github.com/timrwood/b8c2d90d528eddb53ab5" >See changelog</a>

- Release Jul 8, 2013

Added better week support.
Added ability to set offset with `moment#zone`.
Added ability to set month or weekday from a string.
Added `moment#min` and `moment#max`

128.19.24 2.0.0 <a href="https://gist.github.com/timrwood/e72f2eef320ed9e37c51" >See changelog</a>

- Release Feb 9, 2013

Added short form localized tokens.
Added ability to define language a string should be parsed in.
Added support for reversed add/subtract arguments.
Added support for ‘endOf('week')`and`startOf('week')`.
Fixed the logic for ‘moment::diff(Moment, 'months')`and`moment::diff(Moment, 'years')`
`moment#diff` now floors instead of rounds.
Normalized `moment#toString`.
Added `isSame`, `isAfter`, and `isBefore` methods.
Added better week support.
Added `moment#toJSON`
Bugfix: Fixed parsing of first century dates
Bugfix: Parsing 10Sep2001 should work as expected
Bugfix: Fixed weirdness with `moment.utc()` parsing.
Changed language ordinal method to return the number + ordinal instead of just the ordinal.
Changed two digit year parsing cutoff to match strptime.
Removed `moment#sod` and `moment#eod` in favor of `moment#startOf` and `moment#endOf`.
Removed `moment.humanizeDuration()` in favor of `moment.duration().humanize()`.
Removed the lang data objects from the top level namespace.
Duplicate `Date` passed to `moment()` instead of referencing it.

128.19.25 1.7.2 <a href="https://github.com/timrwood/moment/issues/456" >See discussion</a>

- Release Oct 2, 2012

Bugfixes

128.19.26 1.7.1 <a href="https://github.com/timrwood/moment/issues/384" >See discussion</a>

- Release Oct 1, 2012

Bugfixes

128.19.27 1.7.0 <a href="https://github.com/timrwood/moment/issues/288" >See discussion</a>

- Release Jul 26, 2012

Added `moment.fn.endOf()` and `moment.fn.startOf()`.
Added validation via `moment.fn.isValid()`.
Made formatting method 3x faster. http://jsperf.com/momentjs-cached-format-functions
Add support for month/weekday callbacks in `moment.fn.format()`
Added instance specific languages.

Added two letter weekday abbreviations with the formatting token `dd`.
Various language updates.
Various bugfixes.

128.19.28 1.6.0 <a href="https://github.com/timrwood/moment/pull/268" >See discussion</a>

- Release Apr 26, 2012

Added Durations.
Revamped parser to support parsing non-separated strings (YYYYMMDD vs YYYY-MM-DD).
Added support for millisecond parsing and formatting tokens (S SS SSS)
Added a getter for `moment.lang()`
Various bugfixes.
There are a few things deprecated in the 1.6.0 release.

1. The format tokens `z` and `zz` (timezone abbreviations like EST CST MST etc) will no longer be supported. Due to inconsistent browser support, we are unable to consistently produce this value. See `this issue` for more background.
2. The method `moment.fn.native` is deprecated in favor of `moment.fn.toDate`. There continue to be issues with Google Closure Compiler throwing errors when using `native`, even in valid instances.
3. The way to customize am/pm strings is being changed. This would only affect you if you created a custom language file. For more information, see `this issue`.

128.19.29 1.5.0 <a href="https://github.↩ com/timrwood/moment/issues?milestone=10&page=1&state=closed" >See milestone</a>

- Release Mar 20, 2012

Added UTC mode.
Added automatic ISO8601 parsing.
Various bugfixes.

128.19.30 1.4.0 <a href="https://github.↩ com/timrwood/moment/issues?milestone=8&state=closed" >See milestone</a>

- Release Feb 4, 2012

Added `moment.fn.toDate` as a replacement for `moment.fn.native`.
Added `moment.fn.sod` and `moment.fn.eod` to get the start and end of day.
Various bugfixes.

128.19.31 1.3.0 <a href="https://github.↩ com/timrwood/moment/issues?milestone=7&state=closed" >See milestone</a>

- Release Jan 5, 2012

Added support for parsing month names in the current language.
Added escape blocks for parsing tokens.
Added `moment.fn.calendar` to format strings like 'Today 2:30 PM', 'Tomorrow 1:25 AM', and 'Last Sunday 4:30 AM'.
Added `moment.fn.day` as a setter.
Various bugfixes

128.19.32 1.2.0 <a href="https://github.↩ com/timrwood/moment/issues?milestone=4&state=closed" >See milestone</a>

- Release Dec 7, 2011

Added timezones to parser and formatter.
Added `moment.fn.isDST`.
Added `moment.fn.zone` to get the timezone offset in minutes.

128.19.33 1.1.2 <a href="https://github.↩ com/timrwood/moment/issues?milestone=6&state=closed" >See milestone</a>

- Release Nov 18, 2011

Various bugfixes

128.19.34 1.1.1 <a href="https://github.↩ com/timrwood/moment/issues?milestone=5&state=closed" >See milestone</a>

- Release Nov 12, 2011

Added time specific diffs (months, days, hours, etc)

128.19.35 1.1.0

- Release Oct 28, 2011

Added `moment.fn.format` localized masks. 'L LL LLL LLLL' `issue 29`
Fixed `issue 31`.

128.19.36 1.0.1

- Release Oct 18, 2011

Added `moment.version` to get the current version.
Removed `window !== undefined` when checking if module exists to support browserify. `issue 25`

128.19.37 1.0.0

- Release

Added convenience methods for getting and setting date parts.
Added better support for `moment.add()`.
Added better lang support in NodeJS.
Renamed library from underscore.date to Moment.js

128.19.38 0.6.1

- Release Oct 12, 2011

Added Portuguese, Italian, and French language support

128.19.39 0.6.0

- Release Sep 21, 2011

Added _date.lang() support. Added support for passing multiple formats to try to parse a date. _date("07-10-1986", ["MM-DD-YYYY", "YYYY-MM-DD"]); Made parse from string and single format 25% faster.

128.19.40 0.5.2

- Release Jul 11, 2011

Bugfix for `issue 8` and `issue 9`.

128.19.41 0.5.1

- Release Jun 17, 2011

Bugfix for `issue 5`.

128.19.42 0.5.0

- Release Jun 13, 2011

Dropped the redundant `_date.date()` in favor of `_date()`. Removed `_date.now()`, as it is a duplicate of `_date()` with no parameters. Removed `_date.isLeapYear(yearNumber)`. Use `_date([year↩ Number]).isLeapYear()` instead. Exposed customization options through the `_date.relativeTime`, `_date.weekdays`, `_date.weekdaysShort`, `_date.months`, `_date.monthsShort`, and `_date.↩ ordinal` variables instead of the `_date.customize()` function.

128.19.43 0.4.1

- Release May 9, 2011

Added date input formats for input strings.

128.19.44 0.4.0

- Release May 9, 2011

Added underscore.date to npm. Removed dependencies on underscore.

128.19.45 0.3.2

- Release Apr 9, 2011

Added ‘‘z’`and`'zz'`to_`.date().format()`. Cleaned up some redundant code to trim off some bytes.

128.19.46 0.3.1

- Release Mar 25, 2011

Cleaned up the namespace. Moved all date manipulation and display functions to the _.date() object.

128.19.47 0.3.0

- Release Mar 25, 2011

Switched to the Underscore methodology of not mucking with the native objects' prototypes. Made chaining possible.

128.19.48 0.2.1

- Release

Changed date names to be a more pseudo standardized 'dddd, MMMM Do YYYY, h:mm:ss a'. Added `Date.↩ prototype` functions `add`, `subtract`, `isdst`, and `isleapyear`.

128.19.49 0.2.0

- Release

Changed function names to be more concise. Changed date format from php date format to custom format.

128.19.50 0.1.0

- Release

Initial release

Chapter 129

<a href="http://momentjs.com/" >Moment.js</a>

[][license-url]
A JavaScript date library for parsing, validating, manipulating, and formatting dates.

129.1 Project Status

Moment.js is a legacy project, now in maintenance mode. In most cases, you should choose a different library.
For more details and recommendations, please see `Project Status` in the docs.
Thank you.

129.2 Resources

- `Documentation`
- /home/visi02/Universidad/Curso 24-25/Proyecto PBIO/Codigos_Generales_PBIO_Sprint0/Backend/nodejs/node↩
_modules/moment/CHANGELOG.md "Changelog"
- `Stack Overflow`

129.3 License

Moment.js is freely distributable under the terms of the [MIT license][license-url].

Chapter 130

license

Chapter 131

ms

Use this package to easily convert various time formats to milliseconds.

131.1 Examples

```
ms('2 days')  // 172800000
ms('1d')      // 86400000
ms('10h')     // 36000000
ms('2.5 hrs') // 9000000
ms('2h')      // 7200000
ms('1m')      // 60000
ms('5s')      // 5000
ms('1y')      // 31557600000
ms('100')     // 100
```

131.1.1 Convert from milliseconds

```
ms(60000)             // "1m"
ms(2 * 60000)         // "2m"
ms(ms('10 hours'))    // "10h"
```

131.1.2 Time format written-out

```
ms(60000, { long: true })             // "1 minute"
ms(2 * 60000, { long: true })         // "2 minutes"
ms(ms('10 hours'), { long: true })    // "10 hours"
```

131.2 Features

- Works both in `node` and in the browser.
- If a number is supplied to `ms`, a string with a unit is returned.
- If a string that contains the number is supplied, it returns it as a number (e.g.: it returns `100` for “100”`).
- If you pass a string with a number and a valid unit, the number of equivalent ms is returned.

131.3 Caught a bug?

1. `Fork` this repository to your own GitHub account and then `clone` it to your local device
2. Link the package to the global module directory: `npm link`
3. Within the module you want to test your local development instance of ms, just link it to the dependencies: `npm link ms`. Instead of the default one from npm, node will now use your clone of ms!

As always, you can run the tests using: `npm test`

Chapter 132

caching_sha2_password

https://dev.mysql.com/doc/refman/8.0/en/caching-sha2-pluggable-authentication.↩
html

```
const mysql = require('mysql');
mysql.createConnection({
  authPlugins: {
    caching_sha2_password: mysql.authPlugins.caching_sha2_password({
      onServerPublikKey: function(key) {
        console.log(key);
      },
      serverPublicKey: 'xxxyyy',
      overrideIsSecure: true //
    })
  }
});
```

Chapter 133

0.6.3 / 2021-05-23

- Fix HKSCS encoding to prefer Big5 codes if both Big5 and HKSCS codes are possible (#264)

133.1 0.6.2 / 2020-07-08

- Support Uint8Array-s decoding without conversion to Buffers, plus fix an edge case.

133.2 0.6.1 / 2020-06-28

- Support Uint8Array-s directly when decoding (#246, by @gyzerok)
- Unify package.json version ranges to be strictly semver-compatible (#241)
- Fix minor issue in UTF-32 decoder's endianness detection code.

133.3 0.6.0 / 2020-06-08

- Updated 'gb18030' encoding to :2005 edition (see https://github.com/whatwg/encoding/issues/22).
- Removed `iconv.extendNodeEncodings()` mechanism. It was deprecated 5 years ago and didn't work in recent Node versions.
- Reworked Streaming API behavior in browser environments to fix #204. Streaming API will be excluded by default in browser packs, saving ∼100Kb bundle size, unless enabled explicitly using ‘iconv.enable←↩ StreamingAPI(require('stream'))`.
- Updates to development environment & tests:
  - Added ./test/webpack private package to test complex new use cases that need custom environment. It's tested as a separate job in Travis CI.
  - Updated generation code for the new EUC-KR index file format from Encoding Standard.
  - Removed Buffer() constructor in tests (#197 by @gabrielschulhof).

133.4 0.5.2 / 2020-06-08

- Added `iconv.getEncoder()` and `iconv.getDecoder()` methods to typescript definitions (#229).
- Fixed semver version to 6.1.2 to support Node 8.x (by @tanandara).
- Capped iconv version to 2.x as 3.x has dropped support for older Node versions.
- Switched from instanbul to c8 for code coverage.

133.5 0.5.1 / 2020-01-18

- Added cp720 encoding (#221, by @kr-deps)
- (minor) Changed Changelog.md formatting to use h2.

133.6 0.5.0 / 2019-06-26

- Added UTF-32 encoding, both little-endian and big-endian variants (UTF-32LE, UTF32-BE). If endianness is not provided for decoding, it's deduced automatically from the stream using a heuristic similar to what we use in UTF-16. (great work in #216 by @kshetline)
- Several minor updates to README (#217 by @oldj, plus some more)
- Added Node versions 10 and 12 to Travis test harness.

133.7 0.4.24 / 2018-08-22

- Added MIK encoding (#196, by @Ivan-Kalatchev)

133.8 0.4.23 / 2018-05-07

- Fix deprecation warning in Node v10 due to the last usage of `new Buffer` (#185, by @felixbuenemann)
- Switched from NodeBuffer to Buffer in typings (#155 by @felixfbecker, #186 by @larssn)

133.9 0.4.22 / 2018-05-05

- Use older semver style for dependencies to be compatible with Node version 0.10 (#182, by @dougwilson)
- Fix tests to accomodate fixes in Node v10 (#182, by @dougwilson)

133.10 0.4.21 / 2018-04-06

- Fix encoding canonicalization (#156)
- Fix the paths in the "browser" field in package.json (#174 by @LMLB)
- Removed "contributors" section in package.json - see Git history instead.

133.11 0.4.20 / 2018-04-06

- Updated `new Buffer()` usages with recommended replacements as it's being deprecated in Node v10 (#176, #178 by @ChALkeR)

133.12 0.4.19 / 2017-09-09

- Fixed iso8859-1 codec regression in handling untranslatable characters (#162, caused by #147)
- Re-generated windows1255 codec, because it was updated in iconv project
- Fixed grammar in error message when iconv-lite is loaded with encoding other than utf8

133.13 0.4.18 / 2017-06-13

- Fixed CESU-8 regression in Node v8.

133.14 0.4.17 / 2017-04-22

- Updated typescript definition file to support Angular 2 AoT mode (#153 by @larssn)

133.15 0.4.16 / 2017-04-22

- Added support for React Native (#150)
- Changed iso8859-1 encoding to usine internal 'binary' encoding, as it's the same thing (#147 by @mscdex)
- Fixed typo in Readme (#138 by @jiangzhuo)
- Fixed build for Node v6.10+ by making correct version comparison
- Added a warning if iconv-lite is loaded not as utf-8 (see #142)

133.16 0.4.15 / 2016-11-21

- Fixed typescript type definition (#137)

133.17 0.4.14 / 2016-11-20

- Preparation for v1.0
- Added Node v6 and latest Node versions to Travis CI test rig
- Deprecated Node v0.8 support
- Typescript typings (@larssn)
- Fix encoding of Euro character in GB 18030 (inspired by @lygstate)
- Add ms prefix to dbcs windows encodings (@rokoroku)

133.18 0.4.13 / 2015-10-01

- Fix silly mistake in deprecation notice.

133.19 0.4.12 / 2015-09-26

- Node v4 support:
  - Added CESU-8 decoding (#106)
  - Added deprecation notice for `extendNodeEncodings`
  - Added Travis tests for Node v4 and io.js latest (#105 by @Mithgol)

133.20 0.4.11 / 2015-07-03

- Added CESU-8 encoding.

133.21 0.4.10 / 2015-05-26

- Changed UTF-16 endianness heuristic to take into account any ASCII chars, not just spaces. This should minimize the importance of "default" endianness.

133.22 0.4.9 / 2015-05-24

- Streamlined BOM handling: strip BOM by default, add BOM when encoding if addBOM: true. Added docs to Readme.
- UTF16 now uses UTF16-LE by default.
- Fixed minor issue with big5 encoding.
- Added io.js testing on Travis; updated node-iconv version to test against. Now we just skip testing SBCS encodings that node-iconv doesn't support.
- (internal refactoring) Updated codec interface to use classes.
- Use strict mode in all files.

133.23 0.4.8 / 2015-04-14

- added alias UNICODE-1-1-UTF-7 for UTF-7 encoding (#94)

133.24 0.4.7 / 2015-02-05

- stop official support of Node.js v0.8. Should still work, but no guarantees. reason: Packages needed for testing are hard to get on Travis CI.
- work in environment where Object.prototype is monkey patched with enumerable props (#89).

133.25 0.4.6 / 2015-01-12

- fix rare aliases of single-byte encodings (thanks @mscdex)
- double the timeout for dbcs tests to make them less flaky on travis

133.26 0.4.5 / 2014-11-20

- fix windows-31j and x-sjis encoding support (@nleush)
- minor fix: undefined variable reference when internal error happens

133.27 0.4.4 / 2014-07-16

- added encodings UTF-7 (RFC2152) and UTF-7-IMAP (RFC3501 Section 5.1.3)
- fixed streaming base64 encoding

133.28 0.4.3 / 2014-06-14

- added encodings UTF-16BE and UTF-16 with BOM

133.29 0.4.2 / 2014-06-12

- don't throw exception if `extendNodeEncodings()` is called more than once

133.30 0.4.1 / 2014-06-11

- codepage 808 added

133.31 0.4.0 / 2014-06-10

- code is rewritten from scratch
- all widespread encodings are supported
- streaming interface added
- browserify compatibility added
- (optional) extend core primitive encodings to make usage even simpler
- moved from vows to mocha as the testing framework

Chapter 134

iconv-lite: Pure JS character encoding conversion

- No need for native code compilation. Quick to install, works on Windows and in sandboxed environments like Cloud9.
- Used in popular projects like Express.js (body_parser), Grunt, Nodemailer, Yeoman and others.
- Faster than node-iconv (see below for performance comparison).
- Intuitive encode/decode API, including Streaming support.
- In-browser usage via browserify or webpack (~180kb gzip compressed with Buffer shim included).
- Typescript type definition file included.
- React Native is supported (need to install `stream` module to enable Streaming API).
- License: MIT.

134.1 Usage

134.1.1 Basic API

```
var iconv = require('iconv-lite');
// Convert from an encoded buffer to a js string.
str = iconv.decode(Buffer.from([0x68, 0x65, 0x6c, 0x6c, 0x6f]), 'win1251');
// Convert from a js string to an encoded buffer.
buf = iconv.encode("Sample input string", 'win1251');
// Check if encoding is supported
iconv.encodingExists("us-ascii")
```

134.1.2 Streaming API

```
// Decode stream (from binary data stream to js strings)
http.createServer(function(req, res) {
    var converterStream = iconv.decodeStream('win1251');
    req.pipe(converterStream);
    converterStream.on('data', function(str) {
        console.log(str); // Do something with decoded strings, chunk-by-chunk.
    });
});
// Convert encoding streaming example
fs.createReadStream('file-in-win1251.txt')
    .pipe(iconv.decodeStream('win1251'))
    .pipe(iconv.encodeStream('ucs2'))
    .pipe(fs.createWriteStream('file-in-ucs2.txt'));
// Sugar:  all encode/decode streams have .collect(cb) method to accumulate data.
http.createServer(function(req, res) {
    req.pipe(iconv.decodeStream('win1251')).collect(function(err, body) {
        assert(typeof body == 'string');
        console.log(body); // full request body string
    });
});
```

134.2 Supported encodings

- All node.js native encodings: utf8, ucs2 / utf16-le, ascii, binary, base64, hex.
- Additional unicode encodings: utf16, utf16-be, utf-7, utf-7-imap, utf32, utf32-le, and utf32-be.
- All widespread singlebyte encodings: Windows 125x family, ISO-8859 family, IBM/DOS codepages, Macintosh family, KOI8 family, all others supported by iconv library. Aliases like 'latin1', 'us-ascii' also supported.
- All widespread multibyte encodings: CP932, CP936, CP949, CP950, GB2312, GBK, GB18030, Big5, Shift↩ _JIS, EUC-JP.

See `all supported encodings on wiki`.
Most singlebyte encodings are generated automatically from `node-iconv`. Thank you Ben Noordhuis and libiconv authors!
Multibyte encodings are generated from `Unicode.org mappings` and `WHATWG Encoding Standard mappings`. Thank you, respective authors!

134.3 Encoding/decoding speed

Comparison with node-iconv module (1000x256kb, on MacBook Pro, Core i5/2.6 GHz, Node v0.12.0). Note: your results may vary, so please always check on your hardware.

```
operation            iconv@2.1.4   iconv-lite@0.4.7
----------------------------------------------------------
encode('win1251')    ~96 Mb/s      ~320 Mb/s
decode('win1251')    ~95 Mb/s      ~246 Mb/s
```

134.4 BOM handling

- Decoding: BOM is stripped by default, unless overridden by passing `stripBOM: false` in options (f.↩ ex. `iconv.decode(buf, enc, {stripBOM: false})`). A callback might also be given as a `stripBOM` parameter - it'll be called if BOM character was actually found.
- If you want to detect UTF-8 BOM when decoding other encodings, use `node-autodetect-decoder-stream` module.
- Encoding: No BOM added, unless overridden by `addBOM: true` option.

134.5 UTF-16 Encodings

This library supports UTF-16LE, UTF-16BE and UTF-16 encodings. First two are straightforward, but UTF-16 is trying to be smart about endianness in the following ways:

- Decoding: uses BOM and 'spaces heuristic' to determine input endianness. Default is UTF-16LE, but can be overridden with ‘defaultEncoding: 'utf-16be'`option. Strips BOM unless`stripBOM: false`.`
- `Encoding: uses UTF-16LE and writes BOM by default. Use`addBOM: false` to override.

134.6 UTF-32 Encodings

This library supports UTF-32LE, UTF-32BE and UTF-32 encodings. Like the UTF-16 encoding above, UTF-32 defaults to UTF-32LE, but uses BOM and 'spaces heuristics' to determine input endianness.

- The default of UTF-32LE can be overridden with the ‘defaultEncoding: 'utf-32be'`option. Strips BOM unless`stripBOM: false`.`
- `Encoding: uses UTF-32LE and writes BOM by default. Use`addBOM: false`to override.` (defaultEncoding: 'utf-32be'` can also be used here to change encoding.)

134.7 Other notes

When decoding, be sure to supply a Buffer to decode() method, otherwise `bad things usually happen`.
Untranslatable characters are set to � or ?. No transliteration is currently supported.
Node versions 0.10.31 and 0.11.13 are buggy, don't use them (see #65, #77).

134.8 Testing

```
$ git clone git@github.com:ashtuchkin/iconv-lite.git
$ cd iconv-lite
$ npm install
$ npm test

$ # To view performance:
$ node test/performance.js
$ # To view test coverage:
$ npm run coverage
$ open coverage/lcov-report/index.html
```

Chapter 135

README

135.1 MySQL2

English | | Português (BR)

> MySQL client for Node.js with focus on performance. Supports prepared statements, non-utf8 encodings, binary log protocol, compression, ssl much more.

Table of Contents

135.1.1 History and Why MySQL2

MySQL2 project is a continuation of MySQL-Native. Protocol parser code was rewritten from scratch and api changed to match popular Node MySQL. MySQL2 team is working together with Node MySQL team to factor out shared code and move it under mysqljs organization.
MySQL2 is mostly API compatible with Node MySQL and supports majority of features. MySQL2 also offers these additional features:

- Faster / Better Performance
- Prepared Statements
- MySQL Binary Log Protocol
- MySQL Server
- Extended support for Encoding and Collation
- Promise Wrapper
- Compression
- SSL and Authentication Switch
- Custom Streams
- Pooling

135.1.2 Installation

MySQL2 is free from native bindings and can be installed on Linux, Mac OS or Windows without any issues.

```
npm install --save mysql2
```

If you are using TypeScript, you will need to install `@types/node`.

```
npm install --save-dev @types/node
```

For TypeScript documentation and examples, see `here`.

135.1.3 Documentation

- `Quickstart`
  - `First Query`, `Using Prepared Statements`, `Using Connection Pools` and more.
- `Documentation`
- `Examples`
- `FAQ`

135.1.4 Acknowledgements

- Internal protocol is written by `@sidorares` `MySQL-Native`.
- Constants, SQL parameters interpolation, Pooling, `ConnectionConfig` class taken from `Node My↩ SQL`.
- SSL upgrade code based on `@TooTallNate` `code`.
- Secure connection / compressed connection api flags compatible to `MariaSQL` client.
- `Contributors`.

135.1.5 Contributing

Want to improve something in **MySQL2**? Please check `Contributing.md` for detailed instruction on how to get started.
To contribute in **MySQL2 Documentation**, please visit the `Website Contributing Guidelines` for detailed instruction on how to get started.

Chapter 136

README

136.1 named-placeholders

compiles "select foo where foo.id = :bar and foo.baz < :baz" into "select foo where foo.id = ? and foo.baz < ?" + ["bar", "baz"]

136.1.1 usage

```
npm install named-placeholders
```

see this mysql2 discussion

```
var mysql = require('mysql');
var toUnnamed = require('named-placeholders')();
var q = toUnnamed('select 1+:test', { test:  123});
mysql.createConnection().query(q[0], q[1]);
```

136.1.2 credits

parser is based on @mscdex code of his excellent node-mariasql library

Chapter 137

0.6.3 / 2022-01-22

- Revert "Lazy-load modules from main entry point"

137.1 0.6.2 / 2019-04-29

- Fix sorting charset, encoding, and language with extra parameters

137.2 0.6.1 / 2016-05-02

- perf: improve `Accept` parsing speed
- perf: improve `Accept-Charset` parsing speed
- perf: improve `Accept-Encoding` parsing speed
- perf: improve `Accept-Language` parsing speed

137.3 0.6.0 / 2015-09-29

- Fix including type extensions in parameters in `Accept` parsing
- Fix parsing `Accept` parameters with quoted equals
- Fix parsing `Accept` parameters with quoted semicolons
- Lazy-load modules from main entry point
- perf: delay type concatenation until needed
- perf: enable strict mode
- perf: hoist regular expressions
- perf: remove closures getting spec properties
- perf: remove a closure from media type parsing
- perf: remove property delete from media type parsing

137.4 0.5.3 / 2015-05-10

- Fix media type parameter matching to be case-insensitive

137.5 0.5.2 / 2015-05-06

- Fix comparing media types with quoted values
- Fix splitting media types with quoted commas

137.6 0.5.1 / 2015-02-14

- Fix preference sorting to be stable for long acceptable lists

137.7 0.5.0 / 2014-12-18

- Fix list return order when large accepted list
- Fix missing identity encoding when q=0 exists
- Remove dynamic building of Negotiator class

137.8 0.4.9 / 2014-10-14

- Fix error when media type has invalid parameter

137.9 0.4.8 / 2014-09-28

- Fix all negotiations to be case-insensitive
- Stable sort preferences of same quality according to client order
- Support Node.js 0.6

137.10 0.4.7 / 2014-06-24

- Handle invalid provided languages
- Handle invalid provided media types

137.11 0.4.6 / 2014-06-11

- Order by specificity when quality is the same

137.12 0.4.5 / 2014-05-29

- Fix regression in empty header handling

137.13 0.4.4 / 2014-05-29

- Fix behaviors when headers are not present

137.14 0.4.3 / 2014-04-16

- Handle slashes on media params correctly

137.15 0.4.2 / 2014-02-28

- Fix media type sorting
- Handle media types params strictly

137.16 0.4.1 / 2014-01-16

- Use most specific matches

137.17 0.4.0 / 2014-01-09

- Remove preferred prefix from methods

Chapter 138

negotiator

An HTTP content negotiator for Node.js

138.1 Installation

```
$ npm install negotiator
```

138.2 API

```
var Negotiator = require('negotiator')
```

138.2.1 Accept Negotiation

```
availableMediaTypes = ['text/html', 'text/plain', 'application/json']
// The negotiator constructor receives a request object
negotiator = new Negotiator(request)
// Let's say Accept header is 'text/html, application/*;q=0.2, image/jpeg;q=0.8'
negotiator.mediaTypes()
// -> ['text/html', 'image/jpeg', 'application/*']
negotiator.mediaTypes(availableMediaTypes)
// -> ['text/html', 'application/json']
negotiator.mediaType(availableMediaTypes)
// -> 'text/html'
```

You can check a working example at `examples/accept.js`.

138.2.1.1 Methods

138.2.1.1.1 mediaType() Returns the most preferred media type from the client.

138.2.1.1.2 mediaType(availableMediaType) Returns the most preferred media type from a list of available media types.

138.2.1.1.3 mediaTypes() Returns an array of preferred media types ordered by the client preference.

138.2.1.1.4 mediaTypes(availableMediaTypes) Returns an array of preferred media types ordered by priority from a list of available media types.

138.2.2 Accept-Language Negotiation

```
negotiator = new Negotiator(request)
availableLanguages = ['en', 'es', 'fr']
// Let's say Accept-Language header is 'en;q=0.8, es, pt'
negotiator.languages()
// -> ['es', 'pt', 'en']
negotiator.languages(availableLanguages)
// -> ['es', 'en']
language = negotiator.language(availableLanguages)
// -> 'es'
```

You can check a working example at `examples/language.js`.

138.2.2.1 Methods

138.2.2.1.1 language() Returns the most preferred language from the client.

138.2.2.1.2 language(availableLanguages) Returns the most preferred language from a list of available languages.

138.2.2.1.3 languages() Returns an array of preferred languages ordered by the client preference.

138.2.2.1.4 languages(availableLanguages) Returns an array of preferred languages ordered by priority from a list of available languages.

138.2.3 Accept-Charset Negotiation

```
availableCharsets = ['utf-8', 'iso-8859-1', 'iso-8859-5']
negotiator = new Negotiator(request)
// Let's say Accept-Charset header is 'utf-8, iso-8859-1;q=0.8, utf-7;q=0.2'
negotiator.charsets()
// -> ['utf-8', 'iso-8859-1', 'utf-7']
negotiator.charsets(availableCharsets)
// -> ['utf-8', 'iso-8859-1']
negotiator.charset(availableCharsets)
// -> 'utf-8'
```

You can check a working example at `examples/charset.js`.

138.2.3.1 Methods

138.2.3.1.1 charset() Returns the most preferred charset from the client.

138.2.3.1.2 charset(availableCharsets) Returns the most preferred charset from a list of available charsets.

138.2.3.1.3 charsets() Returns an array of preferred charsets ordered by the client preference.

138.2.3.1.4 charsets(availableCharsets) Returns an array of preferred charsets ordered by priority from a list of available charsets.

138.2.4 Accept-Encoding Negotiation

```
availableEncodings = ['identity', 'gzip']
negotiator = new Negotiator(request)
// Let's say Accept-Encoding header is 'gzip, compress;q=0.2, identity;q=0.5'
negotiator.encodings()
// -> ['gzip', 'identity', 'compress']
negotiator.encodings(availableEncodings)
// -> ['gzip', 'identity']
negotiator.encoding(availableEncodings)
// -> 'gzip'
```

You can check a working example at `examples/encoding.js`.

138.2.4.1 Methods

138.2.4.1.1 encoding() Returns the most preferred encoding from the client.

138.2.4.1.2 encoding(availableEncodings) Returns the most preferred encoding from a list of available encodings.

138.2.4.1.3 encodings() Returns an array of preferred encodings ordered by the client preference.

138.2.4.1.4 encodings(availableEncodings) Returns an array of preferred encodings ordered by priority from a list of available encodings.

138.3 See Also

The `accepts` module builds on this module and provides an alternative interface, mime type validation, and more.

138.4 License

[MIT](LICENSE)

Chapter 139

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

139.1 <a href="https://github.com/inspect-js/object-inspect/compare/v1.13.1...v1.13.2" >v1.13.2</a> - 2024-06-21

139.1.1 Commits

- [readme] update badges `8a51e6b`
- [Dev Deps] update `@ljharb/eslint-config`, `tape` `ef05f58`
- [Dev Deps] update `error-cause`, `has-tostringtag`, `tape` `c0c6c26`
- [Fix] Don't throw when `global` is not defined `d4d0965`
- [meta] add missing `engines.node` `17a352a`
- [Dev Deps] update `globalthis` `9c08884`
- [Dev Deps] update `error-cause` `6af352d`
- [Dev Deps] update `npmignore` `94e617d`
- [Dev Deps] update `mock-property` `2ac24d7`
- [Dev Deps] update `tape` `46125e5`

139.2 <a href="https://github.com/inspect-js/object-inspect/compare/v1.13.0...v1.13.1" >v1.13.1</a> - 2023-10-19

139.2.1 Commits

- [Fix] in IE 8, global can !== window despite them being prototypes of each other `30d0859`

139.3 <a href="https://github.com/inspect-js/object-inspect/compare/v1.12.3...v1.13.0" >v1.13.0</a> - 2023-10-14

139.3.1 Commits

- [New] add special handling for the global object `431bab2`
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `tape` `fd4f619`
- [Dev Deps] update `mock-property`, `tape` `b453f6c`

- [Dev Deps] update `error-cause` e8ffc57
- [Dev Deps] update `tape` 054b8b9
- [Dev Deps] temporarily remove `aud` due to breaking change in transitive deps 2476845
- [Dev Deps] pin `glob`, since v10.3.8+ requires a broken `jackspeak` 383fa5e
- [Dev Deps] pin `jackspeak` since 2.1.2+ depends on npm aliases, which kill the install process in npm < 6 68c244c

139.4 <a href="https://github.com/inspect-js/object-inspect/compare/v1.12.2...v1.12.3" >v1.12.3</a> - 2023-01-12

139.4.1 Commits

- [Fix] in eg FF 24, collections lack forEach 75fc226
- [actions] update rebase action to use reusable workflow 250a277
- [Dev Deps] update `aud`, `es-value-fixtures`, `tape` 66a19b3
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `error-cause` c43d332
- [Tests] add `@pkgjs/support` to `postlint` e2618d2

139.5 <a href="https://github.com/inspect-js/object-inspect/compare/v1.12.1...v1.12.2" >v1.12.2</a> - 2022-05-26

139.5.1 Commits

- [Fix] use `util.inspect` for a custom inspection symbol method e243bf2
- [meta] add support info ca20ba3
- [Fix] ignore `cause` in node v16.9 and v16.10 where it has a bug 86aa553

139.6 <a href="https://github.com/inspect-js/object-inspect/compare/v1.12.0...v1.12.1" >v1.12.1</a> - 2022-05-21

139.6.1 Commits

- [Tests] use `mock-property` 4ec8893
- [meta] use `npmignore` to autogenerate an npmignore file 07f868c
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog`, `tape` b05244b
- [Dev Deps] update `@ljharb/eslint-config`, `error-cause`, `es-value-fixtures`, `functions-have-names` `tape` d037398
- [Fix] properly handle callable regexes in older engines 848fe48

139.7 <a href="https://github.com/inspect-js/object-inspect/compare/v1.11.1...v1.12.0" >v1.12.0</a> - 2021-12-18

139.7.1 Commits

- [New] add `numericSeparator` boolean option 2d2d537
- [Robustness] cache more prototype methods 191533d
- [New] ensure an Error's `cause` is displayed 53bc2ce
- [Dev Deps] update `eslint`, `@ljharb/eslint-config` bc164b6
- [Robustness] cache `RegExp.prototype.test` a314ab8
- [meta] fix auto-changelog settings 5ed0983

139.8 <a href="https://github.com/inspect-js/object-inspect/compare/v1.11.0...v1.11.1" >v1.11.1</a> - 2021-12-05

139.8.1 Commits

- [meta] add `auto-changelog` 7dbdd22
- [actions] reuse common workflows c8823bc
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `safe-publish-latest`, `tape` 7532b12
- [Refactor] use `has-tostringtag` to behave correctly in the presence of symbol shams 94abb5d
- [actions] update codecov uploader 5ed5102
- [Dev Deps] update `eslint`, `tape` 37b2ad2
- [meta] add `sideEffects` flag d341f90

139.9 <a href="https://github.com/inspect-js/object-inspect/compare/v1.10.3...v1.11.0" >v1.11.0</a> - 2021-07-12

139.9.1 Commits

- [New] `customInspect`: add `symbol` option, to mimic modern util.inspect behavior e973a6e
- [Dev Deps] update `eslint` 05f1cb3

139.10 <a href="https://github.com/inspect-js/object-inspect/compare/v1.10.2...v1.10.3" >v1.10.3</a> - 2021-05-07

139.10.1 Commits

- [Fix] handle core-js Symbol shams 4acfc2c
- [readme] update badges 95c323a
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud` cb38f48

139.11 <a href="https://github.com/inspect-js/object-inspect/compare/v1.10.1...v1.10.2" >v1.10.2</a> - 2021-04-17

139.11.1 Commits

- [Fix] use a robust check for a boxed Symbol `87f12d6`

139.12 <a href="https://github.com/inspect-js/object-inspect/compare/v1.10.0...v1.10.1" >v1.10.1</a> - 2021-04-17

139.12.1 Commits

- [Fix] use a robust check for a boxed bigint `d5ca829`

139.13 <a href="https://github.com/inspect-js/object-inspect/compare/v1.9.0...v1.10.0" >v1.10.0</a> - 2021-04-17

139.13.1 Commits

- [Tests] increase coverage `d8abb8a`
- [actions] use `node/install` instead of `node/run`; use `codecov` action `4bfec2e`
- [New] respect `Symbol.toStringTag` on objects `799b58f`
- [Fix] do not allow Symbol.toStringTag to masquerade as builtins `d6c5b37`
- [New] add `WeakRef` support `b6d898e`
- [meta] do not publish github action workflow files `918cdfc`
- [meta] create `FUNDING.yml` `0bb5fc5`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `tape` `22c8dc0`
- [meta] use `prepublishOnly` script for npm 7+ `e52ee09`
- [Dev Deps] update `eslint` `7c4e6fd`

139.14 <a href="https://github.com/inspect-js/object-inspect/compare/v1.8.0...v1.9.0" >v1.9.0</a> - 2020-11-30

139.14.1 Commits

- [Tests] migrate tests to Github Actions `d262251`
- [New] add enumerable own Symbols to plain object output `ee60c03`
- [Tests] add passing tests `01ac3e4`
- [actions] add "Require Allow Edits" action `c2d7746`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `core-js` `70058de`
- [Fix] hex characters in strings should be uppercased, to match node `assert` `6ab8faa`
- [Tests] run `nyc` on all tests `4c47372`
- [Tests] node 0.8 has an unpredictable property order; fix `groups` test by removing property `f192069`
- [New] add enumerable properties to Function inspect result, per node's `assert` `fd38e1b`

- [Tests] fix tests for node < 10, due to regex match `groups` 2ac6462
- [Dev Deps] update `eslint`, `@ljharb/eslint-config` 44b59e2
- [Robustness] cache `Symbol.prototype.toString` f3c2074
- [Dev Deps] update `eslint` 9411294
- [meta] `require-allow-edits` no longer requires an explicit github token 36c0220
- [actions] update rebase checkout action to v2 55a39a6
- [actions] switch Automatic Rebase workflow to `pull_request_target` event f59fd3c
- [Dev Deps] update `eslint` a492bec

139.15 <a href="https://github.com/inspect-js/object-inspect/compare/v1.7.0...v1.8.0" >v1.8.0</a> - 2020-06-18

139.15.1 Fixed

- [New] add `indent` option #27

139.15.2 Commits

- [Tests] add codecov 4324cbb
- [New] add `maxStringLength` option b3995cb
- [New] add `customInspect` option, to disable custom inspect methods 28b9179
- [Tests] add Date and RegExp tests 3b28eca
- [actions] add automatic rebasing / merge commit blocking 0d9c6c0
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `core-js`, `tape`; add `aud` 7c204f2
- [readme] fix repo URLs, remove testling 34ca9a0
- [Fix] when truncating a deep array, note it as `[Array]` instead of just `[Object]` f74c82d
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `tape` 1a8a5ea
- [Fix] do not be fooled by a function's own `toString` method 7cb5c65
- [patch] indicate explicitly that anon functions are anonymous, to match node 81ebdd4
- [Dev Deps] loosen the `core-js` dep e7472e8
- [Dev Deps] update `tape` 699827e
- [meta] add `safe-publish-latest` c5d2868
- [Dev Deps] update `@ljharb/eslint-config` 9199501

139.16 <a href="https://github.com/inspect-js/object-inspect/compare/v1.6.0...v1.7.0" >v1.7.0</a> - 2019-11-10

139.16.1 Commits

- [Tests] use shared travis-ci configs `19899ed`
- [Tests] add linting `a00f057`
- [Tests] lint last file `2698047`
- [Tests] up to `node` `v12.7`, `v11.15`, `v10.16`, `v8.16`, `v6.17` `589e87a`
- [New] add support for `WeakMap` and `WeakSet` `3ddb3e4`
- [meta] clean up license so github can detect it properly `27527bb`
- [Tests] cover `util.inspect.custom` `36d47b9`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `core-js`, `tape` `b614eaa`
- [Tests] fix coverage thresholds `7b7b176`
- [Tests] bigint tests now can run on unflagged node `063af31`
- [Refactor] add early bailout to `isMap` and `isSet` checks `fc51047`
- [meta] add `funding` field `7f9953a`
- [Tests] Fix invalid strict-mode syntax with hexadecimal `a8b5425`
- [Dev Deps] update `@ljharb/eslint-config` `98df157`
- add copyright to LICENSE `bb69fd0`
- [Tests] use `npx aud` in `posttest` `4838353`
- [Tests] move `0.6` to allowed failures, because it won't build on travis `1bff32a`

139.17 <a href="https://github.com/inspect-js/object-inspect/compare/v1.5.0...v1.6.0" >v1.6.0</a> - 2018-05-02

139.17.1 Commits

- [New] add support for boxed BigInt primitives `356c66a`
- [Tests] up to `node` `v10.0`, `v9.11`, `v8.11`, `v6.14`, `v4.9` `c77b65b`
- [New] Add support for upcoming `BigInt` `1ac548e`
- [Tests] run bigint tests in CI with –harmony-bigint flag `d31b738`
- [Dev Deps] update `core-js`, `tape` `ff9eff6`
- [Docs] fix example to use `safer-buffer` `48cae12`

139.18 <a href="https://github.com/inspect-js/object-inspect/compare/v1.4.1...v1.5.0" >v1.5.0</a> - 2017-12-25

139.18.1 Commits

- [New] add `quoteStyle` option `f5a72d2`
- [Tests] add more test coverage `30ebe4e`
- [Tests] require 0.6 to pass `99a008c`

139.19 <a href="https://github.com/inspect-js/object-inspect/compare/v1.4.0...v1.4.1" >v1.4.1</a> - 2017-12-19

139.19.1 Commits

- [Tests] up to `node` `v9.3`, `v8.9`, `v6.12` `6674476`
- [Fix] `inspect(Object(-0))` should be “Object(-0)”, not “Object(0)” `d0a031f`

139.20 <a href="https://github.com/inspect-js/object-inspect/compare/v1.3.0...v1.4.0" >v1.4.0</a> - 2017-10-24

139.20.1 Commits

- [Tests] add `npm run coverage` `3b48fb2`
- [Tests] remove commented-out osx builds `71e24db`
- [New] add support for `util.inspect.custom`, in node only. `20cca77`
- [Tests] up to `node` `v8.6`; use `nvm install-latest-npm` to ensure new npm doesn’t break old node `252952d`
- [Tests] up to `node` `v8.8` `4aa868d`
- [Dev Deps] update `core-js`, `tape` `59483d1`

139.21 <a href="https://github.com/inspect-js/object-inspect/compare/v1.2.2...v1.3.0" >v1.3.0</a> - 2017-07-31

139.21.1 Fixed

- [Fix] Map/Set: work around core-js bug < v2.5.0 `#9`

139.21.2 Commits

- [New] add support for arrays with additional object keys `0d19937`
- [Tests] up to `node` `v8.2`, `v7.10`, `v6.11`; fix new npm breaking on older nodes `e24784a`
- Only apps should have lockfiles `c6faebc`
- [Dev Deps] update `tape` `7345a0a`

139.22 <a href="https://github.com/inspect-js/object-inspect/compare/v1.2.1...v1.2.2" >v1.2.2</a> - 2017-03-24

139.22.1 Commits

- [Tests] up to `node` `v7.7`, `v6.10`, `v4.8`; improve test matrix `a2ddc15`
- [Tests] up to `node` `v7.0`, `v6.9`, `v5.12`, `v4.6`, `io.js` `v3.3`; improve test matrix `a48949f`
- [Performance] check for primitive types as early as possible. `3b8092a`
- [Refactor] remove unneeded `else`s. `7255034`
- [Refactor] avoid recreating `lowbyte` function every time. `81edd34`
- [Fix] differentiate -0 from 0 `521d345`

- [Refactor] move object key gathering into separate function `aca6265`
- [Refactor] consolidate wrapping logic for boxed primitives into a function. `4e440cd`
- [Robustness] use `typeof` instead of comparing to literal `undefined` `5ca6f60`
- [Refactor] consolidate Map/Set notations. `4e576e5`
- [Tests] ensure that this function remains anonymous, despite ES6 name inference. `7540ae5`
- [Refactor] explicitly coerce Error objects to strings. `7f4ca84`
- [Refactor] split up `var` declarations for debuggability `6f2c11e`
- [Robustness] cache `Object.prototype.toString` `df44a20`
- [Dev Deps] update `tape` `3ec714e`
- [Dev Deps] update `tape` `beb72d9`

139.23 <a href="https://github.com/inspect-js/object-inspect/compare/v1.2.0...v1.2.1" >v1.2.1</a> - 2016-04-09

139.23.1 Fixed

- [Fix] fix Boolean `false` object inspection. `#7`

139.24 <a href="https://github.com/inspect-js/object-inspect/compare/v1.1.0...v1.2.0" >v1.2.0</a> - 2016-04-09

139.24.1 Fixed

- [New] add support for inspecting String/Number/Boolean objects. `#6`

139.24.2 Commits

- [Dev Deps] update `tape` `742caa2`

139.25 <a href="https://github.com/inspect-js/object-inspect/compare/1.0.2...v1.1.0" >v1.1.0</a> - 2015-12-14

139.25.1 Merged

- [New] add ES6 Map/Set support. `#4`

139.25.2 Fixed

- [New] add ES6 Map/Set support. `#3`

139.25.3 Commits

- Update `travis.yml` to test on bunches of `iojs` and `node` versions. `4c1fd65`
- [Dev Deps] update `tape` `88a907e`

139.26 <a href="https://github.com/inspect-js/object-inspect/compare/1.0.1...1.0.2" >1.0.2</a> - 2015-08-07

139.26.1 Commits

- [Fix] Cache `Object.prototype.hasOwnProperty` in case it's deleted later. `1d0075d`
- [Dev Deps] Update `tape` `ca8d5d7`
- gitignore node_modules since this is a reusable modules. `ed41407`

139.27 <a href="https://github.com/inspect-js/object-inspect/compare/1.0.0...1.0.1" >1.0.1</a> - 2015-07-19

139.27.1 Commits

- Make `inspect` work with symbol primitives and objects, including in node 0.11 and 0.12. `ddf1b94`
- bump tape `103d674`
- use newer travis config `d497276`

139.28 <a href="https://github.com/inspect-js/object-inspect/compare/0.4.0...1.0.0" >1.0.0</a> - 2014-08-05

139.28.1 Commits

- error inspect works properly `260a22d`
- seen coverage `57269e8`
- htmlelement instance coverage `397ffe1`
- more element coverage `6905cc2`
- failing test for type errors `385b615`
- fn name coverage `edc906d`
- server-side element test `362d1d3`
- custom inspect fn `e89b0f6`
- fixed browser test `b530882`
- depth test, matches node `1cfd9e0`
- exercise hasOwnProperty path `8d753fb`
- more cases covered for errors `c5c46a5`
- \W obj key test case `b0eceee`
- coverage for explicit depth param `e12b91c`

139.29 <a href="https://github.com/inspect-js/object-inspect/compare/0.3.1...0.4.0" >0.4.0</a> - 2014-03-21

139.29.1 Commits

- passing lowbyte interpolation test `b847511`
- lowbyte test `4a2b0e1`

139.30 <a href="https://github.com/inspect-js/object-inspect/compare/0.3.0...0.3.1" >0.3.1</a> - 2014-03-04

139.30.1 Commits

- sort keys a07b19c

139.31 <a href="https://github.com/inspect-js/object-inspect/compare/0.2.0...0.3.0" >0.3.0</a> - 2014-03-04

139.31.1 Commits

- [] and {} instead of [ ] and { } 654c44b

139.32 <a href="https://github.com/inspect-js/object-inspect/compare/0.1.3...0.2.0" >0.2.0</a> - 2014-03-04

139.32.1 Commits

- failing holes test 99cdfad
- regex already work e324033
- failing undef/null test 1f88a00
- holes in the all example 7d345f3
- check for .inspect(), fixes Buffer use-case c3f7546
- fixes for holes ce25f73
- weird null behavior 405c1ea
- tape is actually a devDependency, upgrade 703b0ce
- put date in the example a342219
- passing the null test 4ab737e

139.33 <a href="https://github.com/inspect-js/object-inspect/compare/0.1.1...0.1.3" >0.1.3</a> - 2013-07-26

139.33.1 Commits

- special isElement() check 882768a
- oh right old IEs don't have indexOf either 36d1275

139.34 <a href="https://github.com/inspect-js/object-inspect/compare/0.1.0...0.1.1" >0.1.1</a> - 2013-07-26

139.34.1 Commits

- tests! 4422fd9
- fix for ie<9, doesn't have hasOwnProperty 6b7d611
- fix for all IEs: no f.name 4e0c2f6
- badges 5ed0d88

139.35 <a href="https://github.com/inspect-js/object-inspect/compare/0.0.0...0.1.0" >0.1.0</a> - 2013-07-26

139.35.1 Commits

- [Function] for functions `ad5c485`

139.36 0.0.0 - 2013-07-26

139.36.1 Commits

- working browser example `34be6b6`
- package.json etc `cad51f2`
- docs complete `b80cce2`
- circular example `4b4a7b9`
- string rep `7afb479`

Chapter 140

object-inspect <sup><a href="https://npmjs.org/package/object-inspect" ><img src="https://versionbadg.es/inspect-js/object-inspect.svg" alt="Version Badge"/></a></sup>

string representations of objects in node and the browser
[][license-url]

140.1 example

140.1.1 circular

```
var inspect = require('object-inspect');
var obj = { a: 1, b: [3,4] };
obj.c = obj;
console.log(inspect(obj));
```

140.1.2 dom element

```
var inspect = require('object-inspect');
var d = document.createElement('div');
d.setAttribute('id', 'beep');
d.innerHTML = '<b>wooo</b><i>iiiii</i>';
console.log(inspect([ d, { a: 3, b : 4, c: [5,6,[7,[8,[9]]]] } ]));
```

output:

```
[ <div id="beep">...</div>, { a: 3, b: 4, c: [ 5, 6, [ 7, [ 8, [ ... ] ] ] ] } ]
```

140.2 methods

```
var inspect = require('object-inspect')
```

140.2.1 var s = inspect(obj, opts={})

Return a string `s` with the string representation of `obj` up to a depth of `opts.depth`.
Additional options:

- `quoteStyle`: must be "single" or "double", if present. Default "single'`for strings,`'double'`for HTML elements. -`maxStringLength`: must be`0`, a positive integer,`Infinity`, or`null`, if present. Default`Infinity`. -`customInspect`: When`true`, a custom inspect method function will be invoked (either undere the`util.inspect.custom`symbol, or the`inspect`property).

When the string'symbol', only the symbol method will be invoked. Defaulttrue. -indent: must be "\t",null, or a positive integer. Defaultnull. -numeric↩ Separator: must be a boolean, if present. Defaultfalse. Iftrue, all numbers will be printed with numeric separators (eg,1234.5678will be printed as'1↩ _234.567_8'`)

140.3 install

With npm do:

```
npm install object-inspect
```

140.4 license

MIT

Chapter 141

2.4.1 / 2022-02-22

- Fix error on early async hooks implementations

141.1 2.4.0 / 2022-02-21

- Prevent loss of async hooks context

141.2 2.3.0 / 2015-05-26

- Add defined behavior for HTTP `CONNECT` requests
- Add defined behavior for HTTP `Upgrade` requests
- deps: ee-first@1.1.1

141.3 2.2.1 / 2015-04-22

- Fix `isFinished(req)` when data buffered

141.4 2.2.0 / 2014-12-22

- Add message object to callback arguments

141.5 2.1.1 / 2014-10-22

- Fix handling of pipelined requests

141.6 2.1.0 / 2014-08-16

- Check if `socket` is detached
- Return `undefined` for `isFinished` if state unknown

141.7 2.0.0 / 2014-08-16

- Add `isFinished` function
- Move to `jshttp` organization
- Remove support for plain socket argument

- Rename to `on-finished`
- Support both `req` and `res` as arguments
- deps: ee-first@1.0.5

141.8 1.2.2 / 2014-06-10

- Reduce listeners added to emitters
  - avoids "event emitter leak" warnings when used multiple times on same request

141.9 1.2.1 / 2014-06-08

- Fix returned value when already finished

141.10 1.2.0 / 2014-06-05

- Call callback when called on already-finished socket

141.11 1.1.4 / 2014-05-27

- Support node.js 0.8

141.12 1.1.3 / 2014-04-30

- Make sure errors passed as instanceof `Error`

141.13 1.1.2 / 2014-04-18

- Default the `socket` to passed-in object

141.14 1.1.1 / 2014-01-16

- Rename module to `finished`

141.15 1.1.0 / 2013-12-25

- Call callback when called on already-errored socket

141.16 1.0.1 / 2013-12-20

- Actually pass the error to the callback

141.17 1.0.0 / 2013-12-20

- Initial release

Chapter 142

on-finished

Execute a callback when a HTTP request closes, finishes, or errors.

142.1 Install

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install on-finished
```

142.2 API

```
var onFinished = require('on-finished')
```

142.2.1 onFinished(res, listener)

Attach a listener to listen for the response to finish. The listener will be invoked only once when the response finished. If the response finished to an error, the first argument will contain the error. If the response has already finished, the listener will be invoked.

Listening to the end of a response would be used to close things associated with the response, like open files.

Listener is invoked as `listener(err, res)`.

```
onFinished(res, function (err, res) {
  // clean up open fds, etc.
  // err contains the error if request error'd
})
```

142.2.2 onFinished(req, listener)

Attach a listener to listen for the request to finish. The listener will be invoked only once when the request finished. If the request finished to an error, the first argument will contain the error. If the request has already finished, the listener will be invoked.

Listening to the end of a request would be used to know when to continue after reading the data.

Listener is invoked as `listener(err, req)`.

```
var data = "
req.setEncoding('utf8')
req.on('data', function (str) {
  data += str
})
onFinished(req, function (err, req) {
  // data is read unless there is err
})
```

142.2.3 onFinished.isFinished(res)

Determine if `res` is already finished. This would be useful to check and not even start certain operations if the response has already finished.

142.2.4 onFinished.isFinished(req)

Determine if `req` is already finished. This would be useful to check and not even start certain operations if the request has already finished.

142.3 Special Node.js requests

142.3.1 HTTP CONNECT method

The meaning of the `CONNECT` method from RFC 7231, section 4.3.6:

> The CONNECT method requests that the recipient establish a tunnel to the destination origin server identified by the request-target and, if successful, thereafter restrict its behavior to blind forwarding of packets, in both directions, until the tunnel is closed. Tunnels are commonly used to create an end-to-end virtual connection, through one or more proxies, which can then be secured using TLS (Transport Layer Security, [RFC5246]).

In Node.js, these request objects come from the ''connect'' event on the HTTP server.
When this module is used on a HTTP `CONNECT` request, the request is considered "finished" immediately, **due to limitations in the Node.js interface**. This means if the `CONNECT` request contains a request entity, the request will be considered "finished" even before it has been read.
There is no such thing as a response object to a `CONNECT` request in Node.js, so there is no support for one.

142.3.2 HTTP Upgrade request

The meaning of the `Upgrade` header from RFC 7230, section 6.1:

> The "Upgrade" header field is intended to provide a simple mechanism for transitioning from HTTP/1.1 to some other protocol on the same connection.

In Node.js, these request objects come from the ''upgrade'' event on the HTTP server.
When this module is used on a HTTP request with an `Upgrade` header, the request is considered "finished" immediately, **due to limitations in the Node.js interface**. This means if the `Upgrade` request contains a request entity, the request will be considered "finished" even before it has been read.
There is no such thing as a response object to a `Upgrade` request in Node.js, so there is no support for one.

142.4 Example

The following code ensures that file descriptors are always closed once the response finishes.
```
var destroy = require('destroy')
var fs = require('fs')
var http = require('http')
var onFinished = require('on-finished')
http.createServer(function onRequest (req, res) {
  var stream = fs.createReadStream('package.json')
  stream.pipe(res)
  onFinished(res, function () {
    destroy(stream)
  })
})
```

142.5 License

[MIT](LICENSE)

Chapter 143

once

Only call a function once.

143.1 usage

```
var once = require('once')
function load (file, cb) {
  cb = once(cb)
  loader.load('file')
  loader.once('load', cb)
  loader.once('error', cb)
}
```

Or add to the Function.prototype in a responsible way:

```
// only has to be done once
require('once').proto()
function load (file, cb) {
  cb = cb.once()
  loader.load('file')
  loader.once('load', cb)
  loader.once('error', cb)
}
```

Ironically, the prototype feature makes this module twice as complicated as necessary.

To check whether you function has been called, use `fn.called`. Once the function is called for the first time the return value of the original function is saved in `fn.value` and subsequent calls will continue to return this value.

```
var once = require('once')
function load (cb) {
  cb = once(cb)
  var stream = createStream()
  stream.once('data', cb)
  stream.once('end', function () {
    if (!cb.called) cb(new Error('not found'))
  })
}
```

143.2 <tt>once.strict(func)</tt>

Throw an error if the function is called twice.

Some functions are expected to be called only once. Using `once` for them would potentially hide logical errors.

In the example below, the `greet` function has to call the callback only once:

```
function greet (name, cb) {
  // return is missing from the if statement
  // when no name is passed, the callback is called twice
  if (!name) cb('Hello anonymous')
  cb('Hello ' + name)
}
function log (msg) {
  console.log(msg)
}
// this will print 'Hello anonymous' but the logical error will be missed
greet(null, once(msg))
// once.strict will print 'Hello anonymous' and throw an error when the callback will be called the second
      time
greet(null, once.strict(msg))
```

Chapter 144

openapi-types Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog`, and this project adheres to `Semantic Versioning`.

144.1 1.4.0 - 2021-01-05

144.1.1 Added

- Added an index signature to `OperationObject` to allow access to extensions.

144.1.2 Fixed

- Added `undefined` to the index signature for `PathsObject` to prevent unsafe null access when `strictNullChecks` is enabled.

144.2 1.3.5 - 2019-05-13

144.2.1 Fixed

- Amended missing usage of PathsObject in OpenAPIV3.Document interface.

144.3 1.3.4 - 2019-01-31

144.3.1 Fixed

- OpenAPIV3: relax security requirement object types (#327)

144.4 1.3.3 - 2019-01-22

144.4.1 Fixed

- Allowing to set a property of BaseSchemaObject as a reference to another SchemaObject (#312)

144.5 1.3.2 - 2018-10-17

144.5.1 Added

- Added `Operation` to `OpenAPI` namespace.

144.6 1.3.1 - 2018-10-03

144.6.1 Fixed

- Updating .npmignore to publish `dist`

144.7 1.3.0 - 2018-10-03

144.7.1 Added

- `OpenAPI.Parameter` - Represents a parameter across all OpenAPI versions that have the notion of a parameter.

144.8 1.2.0 - 2018-09-29

144.8.1 Added

- `OpenAPI.Parameters` - Represents parameters across all OpenAPI versions that have the notion of parameters.
- exporting `OpenAPIV2.Parameters` and `OpenAPIV2.Parameter`.

Chapter 145

openapi-types <a href="https://npmjs.org/package/openapi-types" ><img src="http://img.shields.io/npm/v/openapi-types.svg" alt="NPM version"/></a> <a href="https://npmjs.org/package/openapi-types" ><img src="http://img.shields.io/npm/dm/openapi-types.svg" alt="Downloads"/></a> <a href="https://coveralls.↩ io/github/kogosoftwarellc/open-api?branch=main" ><img src="https://coveralls.↩ io/repos/github/kogosoftwarellc/open-api/badge.↩ svg?branch=main" alt="Coveralls Status"/></a>

Types for OpenAPI documents.

145.1 Usage

```
import { OpenAPIV2, OpenAPIV3, OpenAPIV3_1 } from "openapi-types";
function processV2(doc:  OpenAPIV2.Document) {}
function processV3(doc:  OpenAPIV3.Document) {}
function processV3_1(doc:  OpenAPIV3_1.Document) {}
```

145.2 LICENSE

Chapter 146

1.3.3 / 2019-04-15

- Fix Node.js 0.8 return value inconsistencies

146.1 1.3.2 / 2017-09-09

- perf: reduce overhead for full URLs
- perf: unroll the "fast-path" `RegExp`

146.2 1.3.1 / 2016-01-17

- perf: enable strict mode

146.3 1.3.0 / 2014-08-09

- Add `parseurl.original` for parsing `req.originalUrl` with fallback
- Return `undefined` if `req.url` is `undefined`

146.4 1.2.0 / 2014-07-21

- Cache URLs based on original value
- Remove no-longer-needed URL mis-parse work-around
- Simplify the "fast-path" `RegExp`

146.5 1.1.3 / 2014-07-08

- Fix typo

146.6 1.1.2 / 2014-07-08

- Seriously fix Node.js 0.8 compatibility

146.7 1.1.1 / 2014-07-08

- Fix Node.js 0.8 compatibility

146.8 1.1.0 / 2014-07-08

- Incorporate URL href-only parse fast-path

146.9 1.0.1 / 2014-03-08

- Add missing `require`

146.10 1.0.0 / 2014-03-08

- Genesis from `connect`

Chapter 147

parseurl

Parse a URL with memoization.

147.1 Install

This is a Node.js module available through the npm registry. Installation is done using the npm install command:

```
$ npm install parseurl
```

147.2 API

```
var parseurl = require('parseurl')
```

147.2.1 parseurl(req)

Parse the URL of the given request object (looks at the `req.url` property) and return the result. The result is the same as `url.parse` in Node.js core. Calling this function multiple times on the same `req` where `req.url` does not change will return a cached parsed object, rather than parsing again.

147.2.2 parseurl.original(req)

Parse the original URL of the given request object and return the result. This works by trying to parse `req.originalUrl` if it is a string, otherwise parses `req.url`. The result is the same as `url.parse` in Node.js core. Calling this function multiple times on the same `req` where `req.originalUrl` does not change will return a cached parsed object, rather than parsing again.

147.3 Benchmark

```
$ npm run-script bench
> parseurl@1.3.3 bench nodejs-parseurl
> node benchmark/index.js
  http_parser@2.8.0
  node@10.6.0
  v8@6.7.288.46-node.13
  uv@1.21.0
  zlib@1.2.11
  ares@1.14.0
  modules@64
  nghttp2@1.32.0
  napi@3
  openssl@1.1.0h
  icu@61.1
  unicode@10.0
  cldr@33.0
  tz@2018c
> node benchmark/fullurl.js
  Parsing URL "http://localhost:8888/foo/bar?user=tj&pet=fluffy"
  4 tests completed.
  fasturl            x 2,207,842 ops/sec ±3.76% (184 runs sampled)
  nativeurl - legacy x   507,180 ops/sec ±0.82% (191 runs sampled)
  nativeurl - whatwg x   290,044 ops/sec ±1.96% (189 runs sampled)
  parseurl           x   488,907 ops/sec ±2.13% (192 runs sampled)
> node benchmark/pathquery.js
  Parsing URL "/foo/bar?user=tj&pet=fluffy"
  4 tests completed.
  fasturl            x 3,812,564 ops/sec ±3.15% (188 runs sampled)
  nativeurl - legacy x 2,651,631 ops/sec ±1.68% (189 runs sampled)
  nativeurl - whatwg x   161,837 ops/sec ±2.26% (189 runs sampled)
  parseurl           x 4,166,338 ops/sec ±2.23% (184 runs sampled)
> node benchmark/samerequest.js
  Parsing URL "/foo/bar?user=tj&pet=fluffy" on same request object
  4 tests completed.
  fasturl            x  3,821,651 ops/sec ±2.42% (185 runs sampled)
  nativeurl - legacy x  2,651,162 ops/sec ±1.90% (187 runs sampled)
  nativeurl - whatwg x    175,166 ops/sec ±1.44% (188 runs sampled)
  parseurl           x 14,912,606 ops/sec ±3.59% (183 runs sampled)
> node benchmark/simplepath.js
  Parsing URL "/foo/bar"
  4 tests completed.
  fasturl            x 12,421,765 ops/sec ±2.04% (191 runs sampled)
  nativeurl - legacy x  7,546,036 ops/sec ±1.41% (188 runs sampled)
  nativeurl - whatwg x    198,843 ops/sec ±1.83% (189 runs sampled)
  parseurl           x 24,244,006 ops/sec ±0.51% (194 runs sampled)
> node benchmark/slash.js
  Parsing URL "/"
  4 tests completed.
  fasturl            x 17,159,456 ops/sec ±3.25% (188 runs sampled)
  nativeurl - legacy x 11,635,097 ops/sec ±3.79% (184 runs sampled)
  nativeurl - whatwg x    240,693 ops/sec ±0.83% (189 runs sampled)
  parseurl           x 42,279,067 ops/sec ±0.55% (190 runs sampled)
```

147.4 License

[MIT](LICENSE)

Chapter 148

path-is-absolute <a href="https://travis-ci.↩ org/sindresorhus/path-is-absolute" ><img src="https://travis-ci.org/sindresorhus/path-is-absolute.svg?branch=master" alt="Build Status"/></a>

Node.js 0.12 `path.isAbsolute()` `ponyfill`

148.1 Install

```
$ npm install --save path-is-absolute
```

148.2 Usage

```
const pathIsAbsolute = require('path-is-absolute');
// Running on Linux
pathIsAbsolute('/home/foo');
//=> true
pathIsAbsolute('C:/Users/foo');
//=> false
// Running on Windows
pathIsAbsolute('C:/Users/foo');
//=> true
pathIsAbsolute('/home/foo');
//=> false
// Running on any OS
pathIsAbsolute.posix('/home/foo');
//=> true
pathIsAbsolute.posix('C:/Users/foo');
//=> false
pathIsAbsolute.win32('C:/Users/foo');
//=> true
pathIsAbsolute.win32('/home/foo');
//=> false
```

148.3 API

See the `path.isAbsolute()` `docs`.

148.3.1 pathIsAbsolute(path)

148.3.2 pathIsAbsolute.posix(path)

POSIX specific version.

148.3.3 pathIsAbsolute.win32(path)

Windows specific version.

148.4 License

MIT © `Sindre Sorhus`

Chapter 149

Path-to-RegExp

Turn an Express-style path string such as `/user/:name` into a regular expression.
Note: This is a legacy branch. You should upgrade to `1.x`.

149.1 Usage

```
var pathToRegexp = require('path-to-regexp');
```

149.1.1 pathToRegexp(path, keys, options)

- **path** A string in the express format, an array of such strings, or a regular expression
- **keys** An array to be populated with the keys present in the url. Once the function completes, this will be an array of strings.
- **options**
  - **options.sensitive** Defaults to false, set this to true to make routes case sensitive
  - **options.strict** Defaults to false, set this to true to make the trailing slash matter.
  - **options.end** Defaults to true, set this to false to only match the prefix of the URL.

```
var keys = [];
var exp = pathToRegexp('/foo/:bar', keys);
//keys = ['bar']
//exp = /^\/foo\/(?:([^\/]+?))\/?$/i
```

149.2 Live Demo

You can see a live demo of this library in use at `express-route-tester`.

149.3 License

MIT

Chapter 150

Please upgrade Node <a href="https://www.↩ npmjs.org/package/please-upgrade-node" ><img src="http://img.shields.io/npm/dm/please-upgrade-node.svg?style=flat" alt=""/></a> <a href="https://travis-ci.org/typicode/please-upgrade-node" ><img src="https://travis-ci.org/typicode/please-upgrade-node.svg?branch=master" alt="Build Status"/></a> <a href="https://www.npmjs.↩ com/package/please-upgrade-node" ><img src="https://img.shields.io/npm/v/please-upgrade-node.svg" alt="npm"/></a>

:information_desk_person: show a message to your users to upgrade Node instead of a stacktrace

It's common for new Node users to miss or not understand engines warning when installing a CLI. This package displays a beginner-friendly message if their Node version is below the one expected.

```
$ node -v
0.12
$ modern-cli
modern-cli requires at least version 6 of Node, please upgrade
```

150.1 Support

If you like this project, you can support me on `GitHub Sponsors`

150.2 Usage

```
npm install please-upgrade-node
```

Add `please-upgrade-node` at the top of your CLI

```
#!/usr/bin/env node
const pkg = require('./package.json')
require('please-upgrade-node')(pkg) // <- Must run BEFORE requiring any other modules
```

Set in your `package.json` the required Node version

```
{
  "engines": {
    "node": ">=6"
  }
}
```

Important: >= is the only operator supported by `please-upgrade-node` (e.g. `>=6`, `>=6.0`, `>=6.0.0`).

150.3 Options

You can set custom `exitCode` and `message` function if needed

```
pleaseUpgradeNode(pkg, {
  exitCode: 0, // Default: 1
  message: function(requiredVersion) {
    return 'Oops this program require Node ' + requiredVersion
  }
})
```

Important: to keep `message` function compatible with older versions of Node, avoid using ES6 features like => or string interpolation.

150.4 See also

- `pkg-ok` - :ok_hand: Prevents publishing a module with bad paths
- `husky` - :dog: Git hooks made easy
- `update-notifier` - Update notifications for your CLI app

Thanks to `zeit/serve` for the error message inspiration.

150.5 License

MIT - `Typicode :cactus:` - `Patreon`

Chapter 151

2.0.7 / 2021-05-31

- deps: forwarded@0.2.0
  - Use `req.socket` over deprecated `req.connection`

151.1 2.0.6 / 2020-02-24

- deps: ipaddr.js@1.9.1

151.2 2.0.5 / 2019-04-16

- deps: ipaddr.js@1.9.0

151.3 2.0.4 / 2018-07-26

- deps: ipaddr.js@1.8.0

151.4 2.0.3 / 2018-02-19

- deps: ipaddr.js@1.6.0

151.5 2.0.2 / 2017-09-24

- deps: forwarded0.1.2
  - perf: improve header parsing
  - perf: reduce overhead when no `X-Forwarded-For` header

151.6 2.0.1 / 2017-09-10

- deps: forwarded0.1.1
  - Fix trimming leading / trailing OWS
  - perf: hoist regular expression
- deps: ipaddr.js@1.5.2

151.7 2.0.0 / 2017-08-08

- Drop support for Node.js below 0.10

151.8 1.1.5 / 2017-07-25

- Fix array argument being altered
- deps: ipaddr.js@1.4.0

151.9 1.1.4 / 2017-03-24

- deps: ipaddr.js@1.3.0

151.10 1.1.3 / 2017-01-14

- deps: ipaddr.js@1.2.0

151.11 1.1.2 / 2016-05-29

- deps: ipaddr.js@1.1.1
  - Fix IPv6-mapped IPv4 validation edge cases

151.12 1.1.1 / 2016-05-03

- Fix regression matching mixed versions against multiple subnets

151.13 1.1.0 / 2016-05-01

- Fix accepting various invalid netmasks
  - IPv4 netmasks must be contingous
  - IPv6 addresses cannot be used as a netmask
- deps: ipaddr.js@1.1.0

151.14 1.0.10 / 2015-12-09

- deps: ipaddr.js@1.0.5
  - Fix regression in `isValid` with non-string arguments

151.15 1.0.9 / 2015-12-01

- deps: ipaddr.js@1.0.4
  - Fix accepting some invalid IPv6 addresses
  - Reject CIDRs with negative or overlong masks
- perf: enable strict mode

151.16 1.0.8 / 2015-05-10

- deps: ipaddr.js@1.0.1

151.17 1.0.7 / 2015-03-16

- deps: ipaddr.js@0.1.9
  - Fix OOM on certain inputs to `isValid`

151.18 1.0.6 / 2015-02-01

- deps: ipaddr.js@0.1.8

151.19 1.0.5 / 2015-01-08

- deps: ipaddr.js@0.1.6

151.20 1.0.4 / 2014-11-23

- deps: ipaddr.js@0.1.5
  - Fix edge cases with `isValid`

151.21 1.0.3 / 2014-09-21

- Use `forwarded` npm module

151.22 1.0.2 / 2014-09-18

- Fix a global leak when multiple subnets are trusted
- Support Node.js 0.6
- deps: ipaddr.js@0.1.3

151.23 1.0.1 / 2014-06-03

- Fix links in npm package

151.24 1.0.0 / 2014-05-08

- Add `trust` argument to determine proxy trust on
  - Accepts custom function
  - Accepts IPv4/IPv6 address(es)
  - Accepts subnets
  - Accepts pre-defined names
- Add optional `trust` argument to `proxyaddr.all` to stop at first untrusted
- Add `proxyaddr.compile` to pre-compile `trust` function to make subsequent calls faster

151.25 0.0.1 / 2014-05-04

- Fix bad npm publish

151.26 0.0.0 / 2014-05-04

- Initial release

Chapter 152

proxy-addr

Determine address of proxied request

152.1 Install

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install proxy-addr
```

152.2 API

```
var proxyaddr = require('proxy-addr')
```

152.2.1 proxyaddr(req, trust)

Return the address of the request, using the given `trust` parameter.
The `trust` argument is a function that returns `true` if you trust the address, `false` if you don't. The closest untrusted address is returned.

```
proxyaddr(req, function (addr) { return addr === '127.0.0.1' })
proxyaddr(req, function (addr, i) { return i < 1 })
```

The `trust` argument may also be a single IP address string or an array of trusted addresses, as plain IP addresses, CIDR-formatted strings, or IP/netmask strings.

```
proxyaddr(req, '127.0.0.1')
proxyaddr(req, ['127.0.0.0/8', '10.0.0.0/8'])
proxyaddr(req, ['127.0.0.0/255.0.0.0', '192.168.0.0/255.255.0.0'])
```

This module also supports IPv6. Your IPv6 addresses will be normalized automatically (i.e. `fe80::00ed:1` equals `fe80:0:0:0:0:0:ed:1`).

```
proxyaddr(req, '::1')
proxyaddr(req, ['::1/128', 'fe80::/10'])
```

This module will automatically work with IPv4-mapped IPv6 addresses as well to support node.js in IPv6-only mode. This means that you do not have to specify both `::ffff:a00:1` and `10.0.0.1`.
As a convenience, this module also takes certain pre-defined names in addition to IP addresses, which expand into IP addresses:

```
proxyaddr(req, 'loopback')
proxyaddr(req, ['loopback', 'fc00:ac:1ab5:fff::1/64'])
```

- `loopback`: IPv4 and IPv6 loopback addresses (like `::1` and `127.0.0.1`).
- `linklocal`: IPv4 and IPv6 link-local addresses (like `fe80::1:1:1:1` and `169.254.0.1`).
- `uniquelocal`: IPv4 private addresses and IPv6 unique-local addresses (like `fc00:ac:1ab5:fff::1` and `192.168.0.1`).

When `trust` is specified as a function, it will be called for each address to determine if it is a trusted address. The function is given two arguments: `addr` and `i`, where `addr` is a string of the address to check and `i` is a number that represents the distance from the socket address.

152.2.2 proxyaddr.all(req, [trust])

Return all the addresses of the request, optionally stopping at the first untrusted. This array is ordered from closest to furthest (i.e. `arr[0] === req.connection.remoteAddress`).

```
proxyaddr.all(req)
```

The optional `trust` argument takes the same arguments as `trust` does in `proxyaddr(req, trust)`.

```
proxyaddr.all(req, 'loopback')
```

152.2.3 proxyaddr.compile(val)

Compiles argument `val` into a `trust` function. This function takes the same arguments as `trust` does in `proxyaddr(req, trust)` and returns a function suitable for `proxyaddr(req, trust)`.

```
var trust = proxyaddr.compile('loopback')
var addr = proxyaddr(req, trust)
```

This function is meant to be optimized for use against every request. It is recommend to compile a trust function up-front for the trusted configuration and pass that to `proxyaddr(req, trust)` for each request.

152.3 Testing

```
$ npm test
```

152.4 Benchmarks

```
$ npm run-script bench
```

152.5 License

[MIT](LICENSE)

Chapter 153

<strong>6.13.0</strong>

- [New] `parse`: add `strictDepth` option (#511)
- [Tests] use `npm audit` instead of `aud`

153.1 <strong>6.12.3</strong>

- [Fix] `parse`: properly account for `strictNullHandling` when `allowEmptyArrays`
- [meta] fix changelog indentation

153.2 <strong>6.12.2</strong>

- [Fix] `parse`: parse encoded square brackets (#506)
- [readme] add CII best practices badge

153.3 <strong>6.12.1</strong>

- [Fix] `parse`: Disable `decodeDotInKeys` by default to restore previous behavior (#501)
- [Performance] `utils`: Optimize performance under large data volumes, reduce memory usage, and speed up processing (#502)
- [Refactor] `utils`: use `+=`
- [Tests] increase coverage

153.4 <strong>6.12.0</strong>

- [New] `parse/stringify`: add `decodeDotInKeys`/`encodeDotKeys` options (#488)
- [New] `parse`: add `duplicates` option
- [New] `parse/stringify`: add `allowEmptyArrays` option to allow [] in object values (#487)
- [Refactor] `parse/stringify`: move allowDots config logic to its own variable
- [Refactor] `stringify`: move option-handling code into `normalizeStringifyOptions`
- [readme] update readme, add logos (#484)
- [readme] `stringify`: clarify default `arrayFormat` behavior
- [readme] fix line wrapping
- [readme] remove dead badges

- [Deps] update `side-channel`
- [meta] make the dist build 50% smaller
- [meta] add `sideEffects` flag
- [meta] run build in prepack, not prepublish
- [Tests] `parse`: remove useless tests; add coverage
- [Tests] `stringify`: increase coverage
- [Tests] use `mock-property`
- [Tests] `stringify`: improve coverage
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `has-override-mistake`, `has-property-descriptors`, `mock-property`, `npmignore`, `object-inspect`, `tape`
- [Dev Deps] pin `glob`, since v10.3.8+ requires a broken `jackspeak`
- [Dev Deps] pin `jackspeak` since 2.1.2+ depends on npm aliases, which kill the install process in npm < 6

153.5 <strong>6.11.2</strong>

- [Fix] `parse`: Fix parsing when the global Object prototype is frozen (#473)
- [Tests] add passing test cases with empty keys (#473)

153.6 <strong>6.11.1</strong>

- [Fix] `stringify`: encode comma values more consistently (#463)
- [readme] add usage of `filter` option for injecting custom serialization, i.e. of custom types (#447)
- [meta] remove extraneous code backticks (#457)
- [meta] fix changelog markdown
- [actions] update checkout action
- [actions] restrict action permissions
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `object-inspect`, `tape`

153.7 <strong>6.11.0</strong>

- [New] [Fix] `stringify`: revert 0e903c0; add `commaRoundTrip` option (#442)
- [readme] fix version badge

153.8 <strong>6.10.5</strong>

- [Fix] `stringify`: with `arrayFormat: comma`, properly include an explicit `[]` on a single-item array (#434)

153.9 <strong>6.10.4</strong>

- [Fix] `stringify`: with `arrayFormat: comma`, include an explicit `[]` on a single-item array (#441)
- [meta] use `npmignore` to autogenerate an npmignore file
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `has-symbol`, `object-inspect`, `tape`

153.10 <strong>6.10.3</strong>

- [Fix] `parse`: ignore `__proto__` keys (#428)
- [Robustness] `stringify`: avoid relying on a global `undefined` (#427)
- [actions] reuse common workflows
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `object-inspect`, `tape`

153.11 <strong>6.10.2</strong>

- [Fix] `stringify`: actually fix cyclic references (#426)
- [Fix] `stringify`: avoid encoding arrayformat comma when `encodeValuesOnly = true` (#424)
- [readme] remove travis badge; add github actions/codecov badges; update URLs
- [Docs] add note and links for coercing primitive values (#408)
- [actions] update codecov uploader
- [actions] update workflows
- [Tests] clean up stringify tests slightly
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `object-inspect`, `safe-publish-latest`, `tape`

153.12 <strong>6.10.1</strong>

- [Fix] `stringify`: avoid exception on repeated object values (#402)

153.13 <strong>6.10.0</strong>

- [New] `stringify`: throw on cycles, instead of an infinite loop (#395, #394, #393)
- [New] `parse`: add `allowSparse` option for collapsing arrays with missing indices (#312)
- [meta] fix README.md (#399)
- [meta] only run `npm run dist` in publish, not install
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `has-symbols`, `tape`
- [Tests] fix tests on node v0.6
- [Tests] use `ljharb/actions/node/install` instead of `ljharb/actions/node/run`
- [Tests] Revert "[meta] ignore eclint transitive audit warning"

153.14 <strong>6.9.7</strong>

- [Fix] `parse`: ignore `__proto__` keys (#428)
- [Fix] `stringify`: avoid encoding arrayformat comma when `encodeValuesOnly = true` (#424)
- [Robustness] `stringify`: avoid relying on a global `undefined` (#427)
- [readme] remove travis badge; add github actions/codecov badges; update URLs
- [Docs] add note and links for coercing primitive values (#408)

- [Tests] clean up stringify tests slightly
- [meta] fix README.md (#399)
- Revert "[meta] ignore eclint transitive audit warning"
- [actions] backport actions from main
- [Dev Deps] backport updates from main

153.15 <strong>6.9.6</strong>

- [Fix] restore `dist` dir; mistakenly removed in d4f6c32

153.16 <strong>6.9.5</strong>

- [Fix] `stringify`: do not encode parens for RFC1738
- [Fix] `stringify`: fix arrayFormat comma with empty array/objects (#350)
- [Refactor] `format`: remove `util.assign` call
- [meta] add "Allow Edits" workflow; update rebase workflow
- [actions] switch Automatic Rebase workflow to `pull_request_target` event
- [Tests] `stringify`: add tests for #378
- [Tests] migrate tests to Github Actions
- [Tests] run `nyc` on all tests; use `tape` runner
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `browserify`, `mkdirp`, `object-inspect`, `tape`; add `aud`

153.17 <strong>6.9.4</strong>

- [Fix] `stringify`: when `arrayFormat` is `comma`, respect `serializeDate` (#364)
- [Refactor] `stringify`: reduce branching (part of #350)
- [Refactor] move `maybeMap` to `utils`
- [Dev Deps] update `browserify`, `tape`

153.18 <strong>6.9.3</strong>

- [Fix] proper comma parsing of URL-encoded commas (#361)
- [Fix] parses comma delimited array while having percent-encoded comma treated as normal text (#336)

153.19 <strong>6.9.2</strong>

- [Fix] `parse`: Fix parsing array from object with `comma` true (#359)
- [Fix] `parse`: throw a TypeError instead of an Error for bad charset (#349)
- [meta] ignore eclint transitive audit warning
- [meta] fix indentation in package.json
- [meta] add tidelift marketing copy

- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `object-inspect`, `has-symbols`, `tape`, `mkdirp`, `iconv-lite`
- [actions] add automatic rebasing / merge commit blocking

153.20 <strong>6.9.1</strong>

- [Fix] `parse`: with comma true, handle field that holds an array of arrays (#335)
- [Fix] `parse`: with comma true, do not split non-string values (#334)
- [meta] add `funding` field
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`
- [Tests] use shared travis-ci config

153.21 <strong>6.9.0</strong>

- [New] `parse/stringify`: Pass extra key/value argument to `decoder` (#333)
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `evalmd`
- [Tests] `parse`: add passing `arrayFormat` tests
- [Tests] add `posttest` using `npx aud` to run `npm audit` without a lockfile
- [Tests] up to `node` `v12.10`, `v11.15`, `v10.16`, `v8.16`
- [Tests] `Buffer.from` in node v5.0-v5.9 and v4.0-v4.4 requires a TypedArray

153.22 <strong>6.8.3</strong>

- [Fix] `parse`: ignore __`proto`__ keys (#428)
- [Robustness] `stringify`: avoid relying on a global `undefined` (#427)
- [Fix] `stringify`: avoid encoding arrayformat comma when `encodeValuesOnly = true` (#424)
- [readme] remove travis badge; add github actions/codecov badges; update URLs
- [Tests] clean up stringify tests slightly
- [Docs] add note and links for coercing primitive values (#408)
- [meta] fix README.md (#399)
- [actions] backport actions from main
- [Dev Deps] backport updates from main
- [Refactor] `stringify`: reduce branching
- [meta] do not publish workflow files

153.23 <strong>6.8.2</strong>

- [Fix] proper comma parsing of URL-encoded commas (#361)
- [Fix] parses comma delimited array while having percent-encoded comma treated as normal text (#336)

153.24 <strong>6.8.1</strong>

- [Fix] `parse`: Fix parsing array from object with `comma` true (#359)
- [Fix] `parse`: throw a TypeError instead of an Error for bad charset (#349)
- [Fix] `parse`: with comma true, handle field that holds an array of arrays (#335)
- [fix] `parse`: with comma true, do not split non-string values (#334)
- [meta] add tidelift marketing copy
- [meta] add `funding` field
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `tape`, `safe-publish-latest`, `evalmd`, `has-symbols`, `iconv-lite`, `mkdirp`, `object-inspect`
- [Tests] `parse`: add passing `arrayFormat` tests
- [Tests] use shared travis-ci configs
- [Tests] `Buffer.from` in node v5.0-v5.9 and v4.0-v4.4 requires a TypedArray
- [actions] add automatic rebasing / merge commit blocking

153.25 <strong>6.8.0</strong>

- [New] add `depth=false` to preserve the original key; [Fix] `depth=0` should preserve the original key (#326)
- [New] [Fix] stringify symbols and bigints
- [Fix] ensure node 0.12 can stringify Symbols
- [Fix] fix for an impossible situation: when the formatter is called with a non-string value
- [Refactor] `formats`: tiny bit of cleanup.
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `browserify`, `safe-publish-latest`, `iconv-lite`, `tape`
- [Tests] add tests for `depth=0` and `depth=false` behavior, both current and intuitive/intended (#326)
- [Tests] use `eclint` instead of `editorconfig-tools`
- [docs] readme: add security note
- [meta] add github sponsorship
- [meta] add FUNDING.yml
- [meta] Clean up license text so it's properly detected as BSD-3-Clause

153.26 <strong>6.7.3</strong>

- [Fix] `parse`: ignore `__proto__` keys (#428)
- [Fix] `stringify`: avoid encoding arrayformat comma when `encodeValuesOnly = true` (#424)
- [Robustness] `stringify`: avoid relying on a global `undefined` (#427)
- [readme] remove travis badge; add github actions/codecov badges; update URLs
- [Docs] add note and links for coercing primitive values (#408)
- [meta] fix README.md (#399)

- [meta] do not publish workflow files
- [actions] backport actions from main
- [Dev Deps] backport updates from main
- [Tests] use `nyc` for coverage
- [Tests] clean up stringify tests slightly

153.27 <strong>6.7.2</strong>

- [Fix] proper comma parsing of URL-encoded commas (#361)
- [Fix] parses comma delimited array while having percent-encoded comma treated as normal text (#336)

153.28 <strong>6.7.1</strong>

- [Fix] `parse`: Fix parsing array from object with `comma` true (#359)
- [Fix] `parse`: with comma true, handle field that holds an array of arrays (#335)
- [fix] `parse`: with comma true, do not split non-string values (#334)
- [Fix] `parse`: throw a TypeError instead of an Error for bad charset (#349)
- [Fix] fix for an impossible situation: when the formatter is called with a non-string value
- [Refactor] `formats`: tiny bit of cleanup.
- readme: add security note
- [meta] add tidelift marketing copy
- [meta] add `funding` field
- [meta] add FUNDING.yml
- [meta] Clean up license text so it's properly detected as BSD-3-Clause
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `tape`, `safe-publish-latest`, `evalmd`, `iconv-lite`, `mkdirp`, `object-inspect`, `browserify`
- [Tests] `parse`: add passing `arrayFormat` tests
- [Tests] use shared travis-ci configs
- [Tests] `Buffer.from` in node v5.0-v5.9 and v4.0-v4.4 requires a TypedArray
- [Tests] add tests for `depth=0` and `depth=false` behavior, both current and intuitive/intended
- [Tests] use `eclint` instead of `editorconfig-tools`
- [actions] add automatic rebasing / merge commit blocking

153.29 <strong>6.7.0</strong>

- [New] `stringify/parse`: add `comma` as an `arrayFormat` option (#276, #219)
- [Fix] correctly parse nested arrays (#212)
- [Fix] `utils.merge`: avoid a crash with a null target and a truthy non-array source, also with an array source
- [Robustness] `stringify`: cache `Object.prototype.hasOwnProperty`
- [Refactor] `utils`: `isBuffer`: small tweak; add tests
- [Refactor] use cached `Array.isArray`
- [Refactor] `parse/stringify`: make a function to normalize the options
- [Refactor] `utils`: reduce observable [[Get]]s
- [Refactor] `stringify/utils`: cache `Array.isArray`
- [Tests] always use `String(x)` over `x.toString()`
- [Tests] fix Buffer tests to work in node < 4.5 and node < 5.10
- [Tests] temporarily allow coverage to fail

153.30 <strong>6.6.1</strong>

- [Fix] `parse`: ignore `__proto__` keys (#428)
- [Fix] fix for an impossible situation: when the formatter is called with a non-string value
- [Fix] `utils.merge`: avoid a crash with a null target and an array source
- [Fix] `utils.merge`: avoid a crash with a null target and a truthy non-array source
- [Fix] correctly parse nested arrays
- [Robustness] `stringify`: avoid relying on a global `undefined` (#427)
- [Robustness] `stringify`: cache `Object.prototype.hasOwnProperty`
- [Refactor] `formats`: tiny bit of cleanup.
- [Refactor] `utils`: `isBuffer`: small tweak; add tests
- [Refactor]: `stringify/utils`: cache `Array.isArray`
- [Refactor] `utils`: reduce observable [[Get]]s
- [Refactor] use cached `Array.isArray`
- [Refactor] `parse/stringify`: make a function to normalize the options
- [readme] remove travis badge; add github actions/codecov badges; update URLs
- [Docs] Clarify the need for "arrayLimit" option
- [meta] fix README.md (#399)
- [meta] do not publish workflow files
- [meta] Clean up license text so it's properly detected as BSD-3-Clause
- [meta] add FUNDING.yml
- [meta] Fixes typo in CHANGELOG.md
- [actions] backport actions from main
- [Tests] fix Buffer tests to work in node < 4.5 and node < 5.10
- [Tests] always use `String(x)` over `x.toString()`
- [Dev Deps] backport from main

153.31 <strong>6.6.0</strong>

- [New] Add support for iso-8859-1, utf8 "sentinel" and numeric entities (#268)
- [New] move two-value combine to a `utils` function (#189)
- [Fix] `stringify`: fix a crash with `strictNullHandling` and a custom `filter/serializeDate` (#279)
- [Fix] when `parseArrays` is false, properly handle keys ending in `[]` (#260)
- [Fix] `stringify`: do not crash in an obscure combo of `interpretNumericEntities`, a bad custom `decoder`, & `iso-8859-1`
- [Fix] `utils`: `merge`: fix crash when `source` is a truthy primitive & no options are provided
- [refactor] `stringify`: Avoid arr = arr.concat(...), push to the existing instance (#269)
- [Refactor] `parse`: only need to reassign the var once
- [Refactor] `parse/stringify`: clean up `charset` options checking; fix defaults
- [Refactor] add missing defaults
- [Refactor] `parse`: one less `concat` call
- [Refactor] `utils`: `compactQueue`: make it explicitly side-effecting
- [Dev Deps] update `browserify`, `eslint`, `@ljharb/eslint-config`, `iconv-lite`, `safe-publish-latest`, `tape`
- [Tests] up to `node` `v10.10`, `v9.11`, `v8.12`, `v6.14`, `v4.9`; pin included builds to LTS

153.32 <strong>6.5.3</strong>

- [Fix] `parse`: ignore `__proto__` keys (#428)
- [Fix] `utils.merge`: avoid a crash with a null target and a truthy non-array source
- [Fix] correctly parse nested arrays
- [Fix] `stringify`: fix a crash with `strictNullHandling` and a custom `filter/serializeDate` (#279)
- [Fix] `utils`: `merge`: fix crash when `source` is a truthy primitive & no options are provided
- [Fix] when `parseArrays` is false, properly handle keys ending in `[]`
- [Fix] fix for an impossible situation: when the formatter is called with a non-string value
- [Fix] `utils.merge`: avoid a crash with a null target and an array source
- [Refactor] `utils`: reduce observable [[Get]]s
- [Refactor] use cached `Array.isArray`
- [Refactor] `stringify`: Avoid arr = arr.concat(...), push to the existing instance (#269)
- [Refactor] `parse`: only need to reassign the var once
- [Robustness] `stringify`: avoid relying on a global `undefined` (#427)
- [readme] remove travis badge; add github actions/codecov badges; update URLs
- [Docs] Clean up license text so it's properly detected as BSD-3-Clause
- [Docs] Clarify the need for "arrayLimit" option

- [meta] fix README.md (#399)
- [meta] add FUNDING.yml
- [actions] backport actions from main
- [Tests] always use `String(x)` over `x.toString()`
- [Tests] remove nonexistent tape option
- [Dev Deps] backport from main

153.33 <strong>6.5.2</strong>

- [Fix] use `safer-buffer` instead of `Buffer` constructor
- [Refactor] utils: `module.exports` one thing, instead of mutating `exports` (#230)
- [Dev Deps] update `browserify`, `eslint`, `iconv-lite`, `safer-buffer`, `tape`, `browserify`

153.34 <strong>6.5.1</strong>

- [Fix] Fix parsing & compacting very deep objects (#224)
- [Refactor] name utils functions
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `tape`
- [Tests] up to `node` `v8.4`; use `nvm install-latest-npm` so newer npm doesn't break older node
- [Tests] Use precise dist for Node.js 0.6 runtime (#225)
- [Tests] make 0.6 required, now that it's passing
- [Tests] on `node` `v8.2`; fix npm on node 0.6

153.35 <strong>6.5.0</strong>

- [New] add `utils.assign`
- [New] pass default encoder/decoder to custom encoder/decoder functions (#206)
- [New] `parse`/`stringify`: add `ignoreQueryPrefix`/`addQueryPrefix` options, respectively (#213)
- [Fix] Handle stringifying empty objects with addQueryPrefix (#217)
- [Fix] do not mutate `options` argument (#207)
- [Refactor] `parse`: cache index to reuse in else statement (#182)
- [Docs] add various badges to readme (#208)
- [Dev Deps] update `eslint`, `browserify`, `iconv-lite`, `tape`
- [Tests] up to `node` `v8.1`, `v7.10`, `v6.11`; npm v4.6 breaks on node < v1; npm v5+ breaks on node < v4
- [Tests] add `editorconfig-tools`

153.36 <strong>6.4.1</strong>

- [Fix] `parse`: ignore `__proto__` keys (#428)
- [Fix] fix for an impossible situation: when the formatter is called with a non-string value
- [Fix] use `safer-buffer` instead of `Buffer` constructor
- [Fix] `utils.merge`: avoid a crash with a null target and an array source
- [Fix] `utils.merge`: avoid a crash with a null target and a truthy non-array source
- [Fix] `stringify`: fix a crash with `strictNullHandling` and a custom `filter/serializeDate` (#279)
- [Fix] `utils`: `merge`: fix crash when `source` is a truthy primitive & no options are provided
- [Fix] when `parseArrays` is false, properly handle keys ending in `[]`
- [Robustness] `stringify`: avoid relying on a global `undefined` (#427)
- [Refactor] use cached `Array.isArray`
- [Refactor] `stringify`: Avoid arr = arr.concat(...), push to the existing instance (#269)
- [readme] remove travis badge; add github actions/codecov badges; update URLs
- [Docs] Clarify the need for "arrayLimit" option
- [meta] fix README.md (#399)
- [meta] Clean up license text so it's properly detected as BSD-3-Clause
- [meta] add FUNDING.yml
- [actions] backport actions from main
- [Tests] remove nonexistent tape option
- [Dev Deps] backport from main

153.37 <strong>6.4.0</strong>

- [New] `qs.stringify`: add `encodeValuesOnly` option
- [Fix] follow `allowPrototypes` option during merge (#201, #201)
- [Fix] support keys starting with brackets (#202, #200)
- [Fix] chmod a-x
- [Dev Deps] update `eslint`
- [Tests] up to `node v7.7`, `v6.10`,`v4.8`; disable osx builds since they block linux builds
- [eslint] reduce warnings

153.38 <strong>6.3.3</strong>

- [Fix] `parse`: ignore `__proto__` keys (#428)
- [Fix] fix for an impossible situation: when the formatter is called with a non-string value
- [Fix] `utils.merge`: avoid a crash with a null target and an array source
- [Fix] `utils.merge`: avoid a crash with a null target and a truthy non-array source
- [Fix] `stringify`: fix a crash with `strictNullHandling` and a custom `filter/serializeDate` (#279)
- [Fix] `utils`: `merge`: fix crash when `source` is a truthy primitive & no options are provided
- [Fix] when `parseArrays` is false, properly handle keys ending in `[]`
- [Robustness] `stringify`: avoid relying on a global `undefined` (#427)
- [Refactor] use cached `Array.isArray`
- [Refactor] `stringify`: Avoid arr = arr.concat(...), push to the existing instance (#269)
- [Docs] Clarify the need for "arrayLimit" option
- [meta] fix README.md (#399)
- [meta] Clean up license text so it's properly detected as BSD-3-Clause
- [meta] add FUNDING.yml
- [actions] backport actions from main
- [Tests] use `safer-buffer` instead of `Buffer` constructor
- [Tests] remove nonexistent tape option
- [Dev Deps] backport from main

153.39 <strong>6.3.2</strong>

- [Fix] follow `allowPrototypes` option during merge (#201, #200)
- [Dev Deps] update `eslint`
- [Fix] chmod a-x
- [Fix] support keys starting with brackets (#202, #200)
- [Tests] up to `node` `v7.7`, `v6.10`,`v4.8`; disable osx builds since they block linux builds

153.40 <strong>6.3.1</strong>

- [Fix] ensure that `allowPrototypes: false` does not ever shadow Object.prototype properties (thanks, @snyk!)
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `browserify`, `iconv-lite`, `qs-iconv`, `tape`
- [Tests] on all node minors; improve test matrix
- [Docs] document stringify option `allowDots` (#195)
- [Docs] add empty object and array values example (#195)
- [Docs] Fix minor inconsistency/typo (#192)

- [Docs] document stringify option `sort` (#191)
- [Refactor] `stringify`: throw faster with an invalid encoder
- [Refactor] remove unnecessary escapes (#184)
- Remove contributing.md, since `qs` is no longer part of `hapi` (#183)

153.41 <strong>6.3.0</strong>

- [New] Add support for RFC 1738 (#174, #173)
- [New] `stringify`: Add `serializeDate` option to customize Date serialization (#159)
- [Fix] ensure `utils.merge` handles merging two arrays
- [Refactor] only constructors should be capitalized
- [Refactor] capitalized var names are for constructors only
- [Refactor] avoid using a sparse array
- [Robustness] `formats`: cache `String#replace`
- [Dev Deps] update `browserify`, `eslint`, `@ljharb/eslint-config`; add `safe-publish-latest`
- [Tests] up to `node` `v6.8`, `v4.6`; improve test matrix
- [Tests] flesh out arrayLimit/arrayFormat tests (#107)
- [Tests] skip Object.create tests when null objects are not available
- [Tests] Turn on eslint for test files (#175)

153.42 <strong>6.2.4</strong>

- [Fix] `parse`: ignore __`proto`__ keys (#428)
- [Fix] `utils.merge`: avoid a crash with a null target and an array source
- [Fix] `utils.merge`: avoid a crash with a null target and a truthy non-array source
- [Fix] `utils`: `merge`: fix crash when `source` is a truthy primitive & no options are provided
- [Fix] when `parseArrays` is false, properly handle keys ending in `[]`
- [Robustness] `stringify`: avoid relying on a global `undefined` (#427)
- [Refactor] use cached `Array.isArray`
- [Docs] Clarify the need for "arrayLimit" option
- [meta] fix README.md (#399)
- [meta] Clean up license text so it’s properly detected as BSD-3-Clause
- [meta] add FUNDING.yml
- [actions] backport actions from main
- [Tests] use `safer-buffer` instead of `Buffer` constructor
- [Tests] remove nonexistent tape option
- [Dev Deps] backport from main

153.43 <strong>6.2.3</strong>

- [Fix] follow `allowPrototypes` option during merge (#201, #200)
- [Fix] chmod a-x
- [Fix] support keys starting with brackets (#202, #200)
- [Tests] up to `node` `v7.7`, `v6.10`,`v4.8`; disable osx builds since they block linux builds

153.44 <strong>6.2.2</strong>

- [Fix] ensure that `allowPrototypes: false` does not ever shadow Object.prototype properties

153.45 <strong>6.2.1</strong>

- [Fix] ensure `key[]=x&key[]&key[]=y` results in 3, not 2, values
- [Refactor] Be explicit and use `Object.prototype.hasOwnProperty.call`
- [Tests] remove `parallelshell` since it does not reliably report failures
- [Tests] up to `node` `v6.3`, `v5.12`
- [Dev Deps] update `tape`, `eslint`, `@ljharb/eslint-config`, `qs-iconv`

153.46 <a href="https://github.↩ com/ljharb/qs/issues?milestone=36&state=closed" ><strong>6.2.0</strong></a>

- [New] pass Buffers to the encoder/decoder directly (#161)
- [New] add "encoder" and "decoder" options, for custom param encoding/decoding (#160)
- [Fix] fix compacting of nested sparse arrays (#150)

153.47 <strong>6.1.2</strong>

- [Fix] follow `allowPrototypes` option during merge (#201, #200)
- [Fix] chmod a-x
- [Fix] support keys starting with brackets (#202, #200)
- [Tests] up to `node` `v7.7`, `v6.10`,`v4.8`; disable osx builds since they block linux builds

153.48 <strong>6.1.1</strong>

- [Fix] ensure that `allowPrototypes: false` does not ever shadow Object.prototype properties

153.49 <a href="https://github.↩ com/ljharb/qs/issues?milestone=35&state=closed" ><strong>6.1.0</strong></a>

- [New] allowDots option for `stringify` (#151)
- [Fix] "sort" option should work at a depth of 3 or more (#151)
- [Fix] Restore `dist` directory; will be removed in v7 (#148)

153.50 <strong>6.0.4</strong>

- [Fix] follow `allowPrototypes` option during merge (#201, #200)
- [Fix] chmod a-x
- [Fix] support keys starting with brackets (#202, #200)
- [Tests] up to `node` `v7.7`, `v6.10`,`v4.8`; disable osx builds since they block linux builds

153.51 <strong>6.0.3</strong>

- [Fix] ensure that `allowPrototypes: false` does not ever shadow Object.prototype properties
- [Fix] Restore `dist` directory; will be removed in v7 (#148)

153.52 <a href="https://github.↵com/ljharb/qs/issues?milestone=33&state=closed"><strong>6.0.2</strong></a>

- Revert ES6 requirement and restore support for node down to v0.8.

153.53 <a href="https://github.↵com/ljharb/qs/issues?milestone=32&state=closed"><strong>6.0.1</strong></a>

- `**#127**` Fix engines definition in package.json

153.54 <a href="https://github.↵com/ljharb/qs/issues?milestone=31&state=closed"><strong>6.0.0</strong></a>

- `**#124**` Use ES6 and drop support for node < v4

153.55 <strong>5.2.1</strong>

- [Fix] ensure `key[]=x&key[]&key[]=y` results in 3, not 2, values

153.56 <a href="https://github.↵com/ljharb/qs/issues?milestone=30&state=closed"><strong>5.2.0</strong></a>

- `**#64**` Add option to sort object keys in the query string

153.57 <a href="https://github.↵com/ljharb/qs/issues?milestone=29&state=closed"><strong>5.1.0</strong></a>

- `**#117**` make URI encoding stringified results optional
- `**#106**` Add flag `skipNulls` to optionally skip null values in stringify

153.58 <a href="https://github.↩ com/ljharb/qs/issues?milestone=28&state=closed" ><strong>5.0.0</strong></a>

- **#114** default allowDots to false
- **#100** include dist to npm

153.59 <a href="https://github.↩ com/ljharb/qs/issues?milestone=26&state=closed" ><strong>4.0.0</strong></a>

- **#98** make returning plain objects and allowing prototype overwriting properties optional

153.60 <a href="https://github.↩ com/ljharb/qs/issues?milestone=24&state=closed" ><strong>3.1.0</strong></a>

- **#89** Add option to disable "Transform dot notation to bracket notation"

153.61 <a href="https://github.↩ com/ljharb/qs/issues?milestone=23&state=closed" ><strong>3.0.0</strong></a>

- **#80** qs.parse silently drops properties
- **#77** Perf boost
- **#60** Add explicit option to disable array parsing
- **#74** Bad parse when turning array into object
- **#81** Add a `filter` option
- **#68** Fixed issue with recursion and passing strings into objects.
- **#66** Add mixed array and object dot notation support Closes: #47
- **#76** RFC 3986
- **#85** No equal sign
- **#84** update license attribute

153.62 <a href="https://github.↩ com/ljharb/qs/issues?milestone=20&state=closed" ><strong>2.4.1</strong></a>

- **#73** Property 'hasOwnProperty' of object #<Object> is not a function

153.63 <a href="https://github.↩ com/ljharb/qs/issues?milestone=19&state=closed" ><strong>2.4.0</strong></a>

- **#70** Add arrayFormat option

153.64 <a href="https://github.↩ com/ljharb/qs/issues?milestone=18&state=closed" ><strong>2.3.3</strong></a>

- **#59** make sure array indexes are >= 0, closes #57
- **#58** make qs usable for browser loader

153.65 <a href="https://github.↩ com/ljharb/qs/issues?milestone=17&state=closed" ><strong>2.3.2</strong></a>

- **#55** allow merging a string into an object

153.66 <a href="https://github.↩ com/ljharb/qs/issues?milestone=16&state=closed" ><strong>2.3.1</strong></a>

- **#52** Return "undefined" and "false" instead of throwing "TypeError".

153.67 <a href="https://github.↩ com/ljharb/qs/issues?milestone=15&state=closed" ><strong>2.3.0</strong></a>

- **#50** add option to omit array indices, closes #46

153.68 <a href="https://github.↩ com/ljharb/qs/issues?milestone=14&state=closed" ><strong>2.2.5</strong></a>

- **#39** Is there an alternative to Buffer.isBuffer?
- **#49** refactor utils.merge, fixes #45
- **#41** avoid browserifying Buffer, for #39

153.69 <a href="https://github.↩ com/ljharb/qs/issues?milestone=13&state=closed" ><strong>2.2.4</strong></a>

- **#38** how to handle object keys beginning with a number

153.70 <a href="https://github.↩ com/ljharb/qs/issues?milestone=12&state=closed" ><strong>2.2.3</strong></a>

- **#37** parser discards first empty value in array
- **#36** Update to lab 4.x

153.71 <a href="https://github.↩ com/ljharb/qs/issues?milestone=11&state=closed" ><strong>2.2.2</strong></a>

- **#33** Error when plain object in a value
- **#34** use Object.prototype.hasOwnProperty.call instead of obj.hasOwnProperty
- **#24** Changelog? Semver?

153.72 <a href="https://github.↩ com/ljharb/qs/issues?milestone=10&state=closed" ><strong>2.2.1</strong></a>

- **#32** account for circular references properly, closes #31
- **#31** qs.parse stackoverflow on circular objects

153.73 <a href="https://github.↩ com/ljharb/qs/issues?milestone=9&state=closed" ><strong>2.2.0</strong></a>

- **#26** Don't use Buffer global if it's not present
- **#30** Bug when merging non-object values into arrays
- **#29** Don't call Utils.clone at the top of Utils.merge
- **#23** Ability to not limit parameters?

153.74 <a href="https://github.↩ com/ljharb/qs/issues?milestone=8&state=closed" ><strong>2.1.0</strong></a>

- **#22** Enable using a RegExp as delimiter

153.75 <a href="https://github.↩ com/ljharb/qs/issues?milestone=7&state=closed" ><strong>2.0.0</strong></a>

- **#18** Why is there arrayLimit?
- **#20** Configurable parametersLimit
- **#21** make all limits optional, for #18, for #20

153.76 <a href="https://github.↩ com/ljharb/qs/issues?milestone=6&state=closed" ><strong>1.2.2</strong></a>

- `**#19**` Don't overwrite null values

153.77 <a href="https://github.↩ com/ljharb/qs/issues?milestone=5&state=closed" ><strong>1.2.1</strong></a>

- `**#16**` ignore non-string delimiters
- `**#15**` Close code block

153.78 <a href="https://github.↩ com/ljharb/qs/issues?milestone=4&state=closed" ><strong>1.2.0</strong></a>

- `**#12**` Add optional delim argument
- `**#13**` fix #11: flattened keys in array are now correctly parsed

153.79 <a href="https://github.↩ com/ljharb/qs/issues?milestone=3&state=closed" ><strong>1.1.0</strong></a>

- `**#7**` Empty values of a POST array disappear after being submitted
- `**#9**` Should not omit equals signs (=) when value is null
- `**#6**` Minor grammar fix in README

153.80 <a href="https://github.↩ com/ljharb/qs/issues?milestone=2&state=closed" ><strong>1.0.2</strong></a>

- `**#5**` array holes incorrectly copied into object on large index

Chapter 154

LICENSE

Chapter 155

README

155.1 qs <sup><a href="https://npmjs.org/package/qs" ><img src="https://versionbadg.es/ljharb/qs.svg" alt="Version Badge"/></a></sup>

[][license-url]

A querystring parsing and stringifying library with some added security.
Lead Maintainer: Jordan Harband
The **qs** module was originally created and maintained by TJ Holowaychuk.

155.1.1 Usage

```
var qs = require('qs');
var assert = require('assert');
var obj = qs.parse('a=c');
assert.deepEqual(obj, { a: 'c' });
var str = qs.stringify(obj);
assert.equal(str, 'a=c');
```

155.1.1.1 Parsing Objects

[](#preventEval)
`qs.parse(string, [options]);`
qs allows you to create nested objects within your query strings, by surrounding the name of sub-keys with square brackets `[]`. For example, the string "foo[bar]=baz` converts to:

```
assert.deepEqual(qs.parse('foo[bar]=baz'), {
    foo: {
        bar: 'baz'
    }
});
```

When using the `plainObjects` option the parsed value is returned as a null object, created via `Object.create(null)` and as such you should be aware that prototype methods will not exist on it and a user may set those names to whatever value they like:

```
var nullObject = qs.parse('a[hasOwnProperty]=b', { plainObjects: true });
assert.deepEqual(nullObject, { a: { hasOwnProperty: 'b' } });
```

By default parameters that would overwrite properties on the object prototype are ignored, if you wish to keep the data from those fields either use `plainObjects` as mentioned above, or set `allowPrototypes` to `true` which will allow user input to overwrite those properties. *WARNING* It is generally a bad idea to enable this option as it can cause problems when attempting to use the properties that have been overwritten. Always be careful with this option.

```
var protoObject = qs.parse('a[hasOwnProperty]=b', { allowPrototypes: true });
assert.deepEqual(protoObject, { a: { hasOwnProperty: 'b' } });
```

URI encoded strings work too:

```
assert.deepEqual(qs.parse('a%5Bb%5D=c'), {
    a: { b: 'c' }
});
```

You can also nest your objects, like "foo[bar][baz]=foobarbaz`:

```
assert.deepEqual(qs.parse('foo[bar][baz]=foobarbaz'), {
    foo: {
        bar: {
            baz: 'foobarbaz'
        }
    }
});
```

By default, when nesting objects **qs** will only parse up to 5 children deep. This means if you attempt to parse a string like "a[b][c][d][e][f][g][h][i]=j"` your resulting object will be:

```
var expected = {
    a: {
        b: {
            c: {
                d: {
                    e: {
                        f: {
                            '[g][h][i]': 'j'
                        }
                    }
                }
            }
        }
    }
};
var string = 'a[b][c][d][e][f][g][h][i]=j';
assert.deepEqual(qs.parse(string), expected);
```

This depth can be overridden by passing a `depth` option to `qs.parse(string, [options])`:

```
var deep = qs.parse('a[b][c][d][e][f][g][h][i]=j', { depth: 1 });
assert.deepEqual(deep, { a: { b: { '[c][d][e][f][g][h][i]': 'j' } } });
```

You can configure **qs** to throw an error when parsing nested input beyond this depth using the `strictDepth` option (defaulted to false):

```
try {
    qs.parse('a[b][c][d][e][f][g][h][i]=j', { depth: 1, strictDepth: true });
} catch (err) {
    assert(err instanceof RangeError);
    assert.strictEqual(err.message, 'Input depth exceeded depth option of 1 and strictDepth is true');
}
```

The depth limit helps mitigate abuse when **qs** is used to parse user input, and it is recommended to keep it a reasonably small number. The strictDepth option adds a layer of protection by throwing an error when the limit is exceeded, allowing you to catch and handle such cases.

For similar reasons, by default **qs** will only parse up to 1000 parameters. This can be overridden by passing a `parameterLimit` option:

```
var limited = qs.parse('a=b&c=d', { parameterLimit: 1 });
assert.deepEqual(limited, { a: 'b' });
```

To bypass the leading question mark, use `ignoreQueryPrefix`:

```
var prefixed = qs.parse('?a=b&c=d', { ignoreQueryPrefix: true });
assert.deepEqual(prefixed, { a: 'b', c: 'd' });
```

An optional delimiter can also be passed:

```
var delimited = qs.parse('a=b;c=d', { delimiter: ';' });
assert.deepEqual(delimited, { a: 'b', c: 'd' });
```

Delimiters can be a regular expression too:

```
var regexed = qs.parse('a=b;c=d,e=f', { delimiter: /[;,]/ });
assert.deepEqual(regexed, { a: 'b', c: 'd', e: 'f' });
```

Option `allowDots` can be used to enable dot notation:

```
var withDots = qs.parse('a.b=c', { allowDots: true });
assert.deepEqual(withDots, { a: { b: 'c' } });
```

Option `decodeDotInKeys` can be used to decode dots in keys Note: it implies `allowDots`, so `parse` will error if you set `decodeDotInKeys` to `true`, and `allowDots` to `false`.

```
var withDots = qs.parse('name%252Eobj.first=John&name%252Eobj.last=Doe', { decodeDotInKeys: true });
assert.deepEqual(withDots, { 'name.obj': { first: 'John', last: 'Doe' }});
```

Option `allowEmptyArrays` can be used to allowing empty array values in object

```
var withEmptyArrays = qs.parse('foo[]&bar=baz', { allowEmptyArrays: true });
assert.deepEqual(withEmptyArrays, { foo: [], bar: 'baz' });
```

Option `duplicates` can be used to change the behavior when duplicate keys are encountered

```
assert.deepEqual(qs.parse('foo=bar&foo=baz'), { foo: ['bar', 'baz'] });
assert.deepEqual(qs.parse('foo=bar&foo=baz', { duplicates: 'combine' }), { foo: ['bar', 'baz'] });
assert.deepEqual(qs.parse('foo=bar&foo=baz', { duplicates: 'first' }), { foo: 'bar' });
assert.deepEqual(qs.parse('foo=bar&foo=baz', { duplicates: 'last' }), { foo: 'baz' });
```

If you have to deal with legacy browsers or services, there's also support for decoding percent-encoded octets as iso-8859-1:

```
var oldCharset = qs.parse('a=%A7', { charset: 'iso-8859-1' });
assert.deepEqual(oldCharset, { a: '§' });
```

Some services add an initial `utf8=` value to forms so that old Internet Explorer versions are more likely to submit the form as utf-8. Additionally, the server can check the value against wrong encodings of the checkmark character and detect that a query string or `application/x-www-form-urlencoded` body was *not* sent as utf-8, eg. if the form had an `accept-charset` parameter or the containing page had a different character set.

qs supports this mechanism via the `charsetSentinel` option. If specified, the `utf8` parameter will be omitted from the returned object. It will be used to switch to `iso-8859-1`/`utf-8` mode depending on how the checkmark is encoded.

Important: When you specify both the `charset` option and the `charsetSentinel` option, the `charset` will be overridden when the request contains a `utf8` parameter from which the actual charset can be deduced. In that sense the `charset` will behave as the default charset rather than the authoritative charset.

```
var detectedAsUtf8 = qs.parse('utf8=%E2%9C%93&a=%C3%B8', {
    charset: 'iso-8859-1',
    charsetSentinel: true
});
assert.deepEqual(detectedAsUtf8, { a: 'ø' });
// Browsers encode the checkmark as &#10003; when submitting as iso-8859-1:
var detectedAsIso8859_1 = qs.parse('utf8=%26%2310003%3B&a=%F8', {
    charset: 'utf-8',
    charsetSentinel: true
});
assert.deepEqual(detectedAsIso8859_1, { a: 'ø' });
```

If you want to decode the `&#...;` syntax to the actual character, you can specify the `interpretNumeric←Entities` option as well:

```
var detectedAsIso8859_1 = qs.parse('a=%26%239786%3B', {
    charset: 'iso-8859-1',
    interpretNumericEntities: true
});
assert.deepEqual(detectedAsIso8859_1, { a: '' });
```

It also works when the charset has been detected in `charsetSentinel` mode.

155.1.1.2 Parsing Arrays

qs can also parse arrays using a similar `[]` notation:

```
var withArray = qs.parse('a[]=b&a[]=c');
assert.deepEqual(withArray, { a: ['b', 'c'] });
```

You may specify an index as well:

```
var withIndexes = qs.parse('a[1]=c&a[0]=b');
assert.deepEqual(withIndexes, { a: ['b', 'c'] });
```

Note that the only difference between an index in an array and a key in an object is that the value between the brackets must be a number to create an array. When creating arrays with specific indices, **qs** will compact a sparse array to only the existing values preserving their order:

```
var noSparse = qs.parse('a[1]=b&a[15]=c');
assert.deepEqual(noSparse, { a: ['b', 'c'] });
```

You may also use `allowSparse` option to parse sparse arrays:

```
var sparseArray = qs.parse('a[1]=2&a[3]=5', { allowSparse: true });
assert.deepEqual(sparseArray, { a: [, '2', , '5'] });
```

Note that an empty string is also a value, and will be preserved:

```
var withEmptyString = qs.parse('a[]=&a[]=b');
assert.deepEqual(withEmptyString, { a: ['', 'b'] });
var withIndexedEmptyString = qs.parse('a[0]=b&a[1]=&a[2]=c');
assert.deepEqual(withIndexedEmptyString, { a: ['b', '', 'c'] });
```

qs will also limit specifying indices in an array to a maximum index of `20`. Any array members with an index of greater than `20` will instead be converted to an object with the index as the key. This is needed to handle cases when someone sent, for example, `a[999999999]` and it will take significant time to iterate over this huge array.

```
var withMaxIndex = qs.parse('a[100]=b');
assert.deepEqual(withMaxIndex, { a: { '100': 'b' } });
```

This limit can be overridden by passing an `arrayLimit` option:

```
var withArrayLimit = qs.parse('a[1]=b', { arrayLimit: 0 });
assert.deepEqual(withArrayLimit, { a: { '1': 'b' } });
```

To disable array parsing entirely, set `parseArrays` to `false`.

```
var noParsingArrays = qs.parse('a[]=b', { parseArrays: false });
assert.deepEqual(noParsingArrays, { a: { '0': 'b' } });
```

If you mix notations, **qs** will merge the two items into an object:

```
var mixedNotation = qs.parse('a[0]=b&a[b]=c');
assert.deepEqual(mixedNotation, { a: { '0': 'b', b: 'c' } });
```

You can also create arrays of objects:

```
var arraysOfObjects = qs.parse('a[][b]=c');
assert.deepEqual(arraysOfObjects, { a: [{ b: 'c' }] });
```

Some people use comma to join array, **qs** can parse it:

```
var arraysOfObjects = qs.parse('a=b,c', { comma: true })
assert.deepEqual(arraysOfObjects, { a: ['b', 'c'] })
```

(*this cannot convert nested objects, such as `a={b:1},{c:d}`*)

155.1.1.3 Parsing primitive/scalar values (numbers, booleans, null, etc)

By default, all values are parsed as strings. This behavior will not change and is explained in `issue #91`.

```
var primitiveValues = qs.parse('a=15&b=true&c=null');
assert.deepEqual(primitiveValues, { a: '15', b: 'true', c: 'null' });
```

If you wish to auto-convert values which look like numbers, booleans, and other values into their primitive counterparts, you can use the `query-types Express JS middleware` which will auto-convert all request query parameters.

155.1.1.4 Stringifying

[](#preventEval)
```
qs.stringify(object, [options]);
```
When stringifying, **qs** by default URI encodes output. Objects are stringified as you would expect:
```
assert.equal(qs.stringify({ a: 'b' }), 'a=b');
assert.equal(qs.stringify({ a: { b: 'c' } }), 'a%5Bb%5D=c');
```
This encoding can be disabled by setting the `encode` option to `false`:
```
var unencoded = qs.stringify({ a: { b: 'c' } }, { encode: false });
assert.equal(unencoded, 'a[b]=c');
```
Encoding can be disabled for keys by setting the `encodeValuesOnly` option to `true`:
```
var encodedValues = qs.stringify(
    { a: 'b', c: ['d', 'e=f'], f: [['g'], ['h']] },
    { encodeValuesOnly: true }
);
assert.equal(encodedValues,'a=b&c[0]=d&c[1]=e%3Df&f[0][0]=g&f[1][0]=h');
```
This encoding can also be replaced by a custom encoding method set as `encoder` option:
```
var encoded = qs.stringify({ a: { b: 'c' } }, { encoder: function (str) {
    // Passed in values `a`, `b`, `c`
    return // Return encoded string
}})
```
(Note: the `encoder` option does not apply if `encode` is `false`)

Analogue to the `encoder` there is a `decoder` option for `parse` to override decoding of properties and values:
```
var decoded = qs.parse('x=z', { decoder: function (str) {
    // Passed in values `x`, `z`
    return // Return decoded string
}})
```
You can encode keys and values using different logic by using the type argument provided to the encoder:
```
var encoded = qs.stringify({ a: { b: 'c' } }, { encoder: function (str, defaultEncoder, charset, type) {
    if (type === 'key') {
        return // Encoded key
    } else if (type === 'value') {
        return // Encoded value
    }
}})
```
The type argument is also provided to the decoder:
```
var decoded = qs.parse('x=z', { decoder: function (str, defaultDecoder, charset, type) {
    if (type === 'key') {
        return // Decoded key
    } else if (type === 'value') {
        return // Decoded value
    }
}})
```
Examples beyond this point will be shown as though the output is not URI encoded for clarity. Please note that the return values in these cases *will* be URI encoded during real usage.

When arrays are stringified, they follow the `arrayFormat` option, which defaults to `indices`:
```
qs.stringify({ a: ['b', 'c', 'd'] });
// 'a[0]=b&a[1]=c&a[2]=d'
```
You may override this by setting the `indices` option to `false`, or to be more explicit, the `arrayFormat` option to `repeat`:
```
qs.stringify({ a: ['b', 'c', 'd'] }, { indices: false });
// 'a=b&a=c&a=d'
```
You may use the `arrayFormat` option to specify the format of the output array:
```
qs.stringify({ a: ['b', 'c'] }, { arrayFormat: 'indices' })
// 'a[0]=b&a[1]=c'
qs.stringify({ a: ['b', 'c'] }, { arrayFormat: 'brackets' })
// 'a[]=b&a[]=c'
qs.stringify({ a: ['b', 'c'] }, { arrayFormat: 'repeat' })
// 'a=b&a=c'
qs.stringify({ a: ['b', 'c'] }, { arrayFormat: 'comma' })
// 'a=b,c'
```
Note: when using `arrayFormat` set to “comma’`, you can also pass the`commaRoundTrip`option set to`true`or`false`, to append`[]` on single-item arrays, so that they can round trip through a parse.

When objects are stringified, by default they use bracket notation:
```
qs.stringify({ a: { b: { c: 'd', e: 'f' } } });
// 'a[b][c]=d&a[b][e]=f'
```
You may override this to use dot notation by setting the `allowDots` option to `true`:
```
qs.stringify({ a: { b: { c: 'd', e: 'f' } } }, { allowDots: true });
// 'a.b.c=d&a.b.e=f'
```
You may encode the dot notation in the keys of object with option `encodeDotInKeys` by setting it to `true`:

Note: it implies `allowDots`, so `stringify` will error if you set `decodeDotInKeys` to `true`, and `allowDots` to `false`. Caveat: when `encodeValuesOnly` is `true` as well as `encodeDotInKeys`, only dots in keys and nothing else will be encoded.

```
qs.stringify({ "name.obj": { "first": "John", "last": "Doe" } }, { allowDots: true, encodeDotInKeys:
     true })
// 'name%252Eobj.first=John&name%252Eobj.last=Doe'
```

You may allow empty array values by setting the `allowEmptyArrays` option to `true`:

```
qs.stringify({ foo: [], bar: 'baz' }, { allowEmptyArrays: true });
// 'foo[]&bar=baz'
```

Empty strings and null values will omit the value, but the equals sign (=) remains in place:

```
assert.equal(qs.stringify({ a: '' }), 'a=');
```

Key with no values (such as an empty object or array) will return nothing:

```
assert.equal(qs.stringify({ a: [] }), '');
assert.equal(qs.stringify({ a: {} }), '');
assert.equal(qs.stringify({ a: [{}] }), '');
assert.equal(qs.stringify({ a: { b: []} }), '');
assert.equal(qs.stringify({ a: { b: {}} }), '');
```

Properties that are set to `undefined` will be omitted entirely:

```
assert.equal(qs.stringify({ a: null, b: undefined }), 'a=');
```

The query string may optionally be prepended with a question mark:

```
assert.equal(qs.stringify({ a: 'b', c: 'd' }, { addQueryPrefix: true }), '?a=b&c=d');
```

The delimiter may be overridden with stringify as well:

```
assert.equal(qs.stringify({ a: 'b', c: 'd' }, { delimiter: ';' }), 'a=b;c=d');
```

If you only want to override the serialization of `Date` objects, you can provide a `serializeDate` option:

```
var date = new Date(7);
assert.equal(qs.stringify({ a: date }), 'a=1970-01-01T00:00:00.007Z'.replace(/:/g, '%3A'));
assert.equal(
    qs.stringify({ a: date }, { serializeDate: function (d) { return d.getTime(); } }),
    'a=7'
);
```

You may use the `sort` option to affect the order of parameter keys:

```
function alphabeticalSort(a, b) {
    return a.localeCompare(b);
}
assert.equal(qs.stringify({ a: 'c', z: 'y', b : 'f' }, { sort: alphabeticalSort }), 'a=c&b=f&z=y');
```

Finally, you can use the `filter` option to restrict which keys will be included in the stringified output. If you pass a function, it will be called for each key to obtain the replacement value. Otherwise, if you pass an array, it will be used to select properties and array indices for stringification:

```
function filterFunc(prefix, value) {
    if (prefix == 'b') {
        // Return an `undefined` value to omit a property.
        return;
    }
    if (prefix == 'e[f]') {
        return value.getTime();
    }
    if (prefix == 'e[g][0]') {
        return value * 2;
    }
    return value;
}
qs.stringify({ a: 'b', c: 'd', e: { f: new Date(123), g: [2] } }, { filter: filterFunc });
// 'a=b&c=d&e[f]=123&e[g][0]=4'
qs.stringify({ a: 'b', c: 'd', e: 'f' }, { filter: ['a', 'e'] });
// 'a=b&e=f'
qs.stringify({ a: ['b', 'c', 'd'], e: 'f' }, { filter: ['a', 0, 2] });
// 'a[0]=b&a[2]=d'
```

You could also use `filter` to inject custom serialization for user defined types. Consider you're working with some api that expects query strings of the format for ranges:

```
https://domain.com/endpoint?range=30...70
```

For which you model as:

```
class Range {
    constructor(from, to) {
        this.from = from;
        this.to = to;
    }
}
```

You could *inject* a custom serializer to handle values of this type:

```
qs.stringify(
    {
        range: new Range(30, 70),
    },
    {
        filter: (prefix, value) => {
            if (value instanceof Range) {
                return `${value.from}...${value.to}`;
            }
            // serialize the usual way
            return value;
        },
    }
);
// range=30...70
```

155.1.1.5 Handling of <tt>null</tt> values

By default, `null` values are treated like empty strings:

```
var withNull = qs.stringify({ a:  null, b:  ” });
assert.equal(withNull, ’a=&b=’);
```

Parsing does not distinguish between parameters with and without equal signs. Both are converted to empty strings.

```
var equalsInsensitive = qs.parse(’a&b=’);
assert.deepEqual(equalsInsensitive, { a:  ”, b:  ” });
```

To distinguish between `null` values and empty strings use the `strictNullHandling` flag. In the result string the `null` values have no = sign:

```
var strictNull = qs.stringify({ a:  null, b:  ” }, { strictNullHandling:  true });
assert.equal(strictNull, ’a&b=’);
```

To parse values without = back to `null` use the `strictNullHandling` flag:

```
var parsedStrictNull = qs.parse(’a&b=’, { strictNullHandling:  true });
assert.deepEqual(parsedStrictNull, { a:  null, b:  ” });
```

To completely skip rendering keys with `null` values, use the `skipNulls` flag:

```
var nullsSkipped = qs.stringify({ a:  ’b’, c:  null}, { skipNulls:  true });
assert.equal(nullsSkipped, ’a=b’);
```

If you're communicating with legacy systems, you can switch to `iso-8859-1` using the `charset` option:

```
var iso = qs.stringify({ æ:  ’æ’ }, { charset:  ’iso-8859-1’ });
assert.equal(iso, ’%E6=%E6’);
```

Characters that don't exist in `iso-8859-1` will be converted to numeric entities, similar to what browsers do:

```
var numeric = qs.stringify({ a:  ’’  }, { charset:  ’iso-8859-1’ });
assert.equal(numeric, ’a=%26%239786%3B’);
```

You can use the `charsetSentinel` option to announce the character by including an `utf8=` parameter with the proper encoding if the checkmark, similar to what Ruby on Rails and others do when submitting forms.

```
var sentinel = qs.stringify({ a:  ’’  }, { charsetSentinel:  true });
assert.equal(sentinel, ’utf8=%E2%9C%93&a=%E2%98%BA’);
var isoSentinel = qs.stringify({ a:  ’æ’ }, { charsetSentinel:  true, charset:  ’iso-8859-1’ });
assert.equal(isoSentinel, ’utf8=%26%2310003%3B&a=%E6’);
```

155.1.1.6 Dealing with special character sets

By default the encoding and decoding of characters is done in `utf-8`, and `iso-8859-1` support is also built in via the `charset` parameter.

If you wish to encode querystrings to a different character set (i.e. `Shift JIS`) you can use the `qs-iconv` library:

```
var encoder = require(’qs-iconv/encoder’)(’shift_jis’);
var shiftJISEncoded = qs.stringify({ a:  ’’  }, { encoder:  encoder });
assert.equal(shiftJISEncoded, ’a=%82%B1%82%F1%82%C9%82%BF%82%CD%81I’);
```

This also works for decoding of query strings:

```
var decoder = require(’qs-iconv/decoder’)(’shift_jis’);
var obj = qs.parse(’a=%82%B1%82%F1%82%C9%82%BF%82%CD%81I’, { decoder:  decoder });
assert.deepEqual(obj, { a:  ’’  });
```

155.1.1.7 RFC 3986 and RFC 1738 space encoding

RFC3986 used as default option and encodes ' ' to *%20* which is backward compatible. In the same time, output can be stringified as per RFC1738 with ' ' equal to '+'.

```
assert.equal(qs.stringify({ a:  ’b c’ }), ’a=b%20c’);
assert.equal(qs.stringify({ a:  ’b c’ }, { format :  ’RFC3986’ }), ’a=b%20c’);
assert.equal(qs.stringify({ a:  ’b c’ }, { format :  ’RFC1738’ }), ’a=b+c’);
```

155.1.2 Security

Please email `@ljharb` or see `https://tidelift.com/security` if you have a potential security vulnerability to report.

155.1.3 qs for enterprise

Available as part of the Tidelift Subscription

The maintainers of qs and thousands of other packages are working with Tidelift to deliver commercial support and maintenance for the open source dependencies you use to build your applications. Save time, reduce risk, and improve code health, while paying the maintainers of the exact dependencies you use. `Learn more.`

155.1.4 Acknowledgements

qs logo by `NUMI`:
`<img src="https://raw.githubusercontent.com/numi-hq/open-design/main/assets/numi-lockup`
`png" alt="NUMI Logo" style="width: 200px;"/>`

Chapter 156

1.2.1 / 2019-05-10

- Improve error when `str` is not a string

156.1 1.2.0 / 2016-06-01

- Add `combine` option to combine overlapping ranges

156.2 1.1.0 / 2016-05-13

- Fix incorrectly returning -1 when there is at least one valid range
- perf: remove internal function

156.3 1.0.3 / 2015-10-29

- perf: enable strict mode

156.4 1.0.2 / 2014-09-08

- Support Node.js 0.6

156.5 1.0.1 / 2014-09-07

- Move repository to jshttp

156.6 1.0.0 / 2013-12-11

- Add repository to package.json
- Add MIT license

156.7 0.0.4 / 2012-06-17

- Change ret -1 for unsatisfiable and -2 when invalid

156.8 0.0.3 / 2012-06-17

- Fix last-byte-pos default to len - 1

156.9 0.0.2 / 2012-06-14

- Add `.type`

156.10 0.0.1 / 2012-06-11

- Initial release

Chapter 157

range-parser

Range header field parser.

157.1 Installation

This is a Node.js module available through the npm registry. Installation is done using the npm install command:

```
$ npm install range-parser
```

157.2 API

```
var parseRange = require('range-parser')
```

157.2.1 parseRange(size, header, options)

Parse the given `header` string where `size` is the maximum size of the resource. An array of ranges will be returned or negative numbers indicating an error parsing.

- `-2` signals a malformed header string
- `-1` signals an unsatisfiable range

```
// parse header from request
var range = parseRange(size, req.headers.range)
// the type of the range
if (range.type === 'bytes') {
  // the ranges
  range.forEach(function (r) {
    // do something with r.start and r.end
  })
}
```

157.2.1.1 Options

These properties are accepted in the options object.

157.2.1.1.1 combine

Specifies if overlapping & adjacent ranges should be combined, defaults to `false`. When `true`, ranges will be combined and returned as if they were specified that way in the header.

```
parseRange(100, 'bytes=50-55,0-10,5-10,56-60', { combine:  true })
// => [
//      { start:  0,  end:  10 },
//      { start:  50, end:  60 }
//    ]
```

157.3 License

[MIT](LICENSE)

Chapter 158

2.5.2 / 2023-02-21

- Fix error message for non-stream argument

158.1 2.5.1 / 2022-02-28

- Fix error on early async hooks implementations

158.2 2.5.0 / 2022-02-21

- Prevent loss of async hooks context
- Prevent hanging when stream is not readable
- deps: http-errors@2.0.0
  - deps: depd@2.0.0
  - deps: statuses@2.0.1

158.3 2.4.3 / 2022-02-14

- deps: bytes@3.1.2

158.4 2.4.2 / 2021-11-16

- deps: bytes@3.1.1
- deps: http-errors@1.8.1
  - deps: setprototypeof@1.2.0
  - deps: toidentifier@1.0.1

158.5 2.4.1 / 2019-06-25

- deps: http-errors@1.7.3
  - deps: inherits@2.0.4

158.6 2.4.0 / 2019-04-17

- deps: bytes@3.1.0
  - Add petabyte (`pb`) support

- deps: http-errors@1.7.2
  - Set constructor name when possible
  - deps: setprototypeof@1.1.1
  - deps: statuses@'>= 1.5.0 < 2'
- deps: `iconv-lite@0.4.24`
  - Added encoding MIK

158.7 2.3.3 / 2018-05-08

- deps: http-errors@1.6.3
  - deps: depd1.1.2
  - deps: setprototypeof@1.1.0
  - deps: statuses@'>= 1.3.1 < 2'
- deps: `iconv-lite@0.4.23`
  - Fix loading encoding with year appended
  - Fix deprecation warnings on Node.js 10+

158.8 2.3.2 / 2017-09-09

- deps: `iconv-lite@0.4.19`
  - Fix ISO-8859-1 regression
  - Update Windows-1255

158.9 2.3.1 / 2017-09-07

- deps: bytes@3.0.0
- deps: http-errors@1.6.2
  - deps: depd@1.1.1
- perf: skip buffer decoding on overage chunk

158.10 2.3.0 / 2017-08-04

- Add TypeScript definitions
- Use `http-errors` for standard emitted errors
- deps: bytes@2.5.0
- deps: `iconv-lite@0.4.18`
  - Add support for React Native
  - Add a warning if not loaded as utf-8
  - Fix CESU-8 decoding in Node.js 8
  - Improve speed of ISO-8859-1 encoding

158.11 2.2.0 / 2017-01-02

- deps: `iconv-lite@0.4.15`
  - Added encoding MS-31J
  - Added encoding MS-932
  - Added encoding MS-936
  - Added encoding MS-949
  - Added encoding MS-950
  - Fix GBK/GB18030 handling of Euro character

158.12 2.1.7 / 2016-06-19

- deps: bytes@2.4.0
- perf: remove double-cleanup on happy path

158.13 2.1.6 / 2016-03-07

- deps: bytes@2.3.0
  - Drop partial bytes on all parsed units
  - Fix parsing byte string that looks like hex

158.14 2.1.5 / 2015-11-30

- deps: bytes@2.2.0
- deps: `iconv-lite@0.4.13`

158.15 2.1.4 / 2015-09-27

- Fix masking critical errors from `iconv-lite`
- deps: `iconv-lite@0.4.12`
  - Fix CESU-8 decoding in Node.js 4.x

158.16 2.1.3 / 2015-09-12

- Fix sync callback when attaching data listener causes sync read
  - Node.js 0.10 compatibility issue

158.17 2.1.2 / 2015-07-05

- Fix error stack traces to skip `makeError`
- deps: `iconv-lite@0.4.11`
  - Add encoding CESU-8

158.18 2.1.1 / 2015-06-14

- Use `unpipe` module for unpiping requests

158.19 2.1.0 / 2015-05-28

- deps: `iconv-lite@0.4.10`
  - Improved UTF-16 endianness detection
  - Leading BOM is now removed when decoding
  - The encoding UTF-16 without BOM now defaults to UTF-16LE when detection fails

158.20 2.0.2 / 2015-05-21

- deps: bytes@2.1.0
  - Slight optimizations

158.21 2.0.1 / 2015-05-10

- Fix a false-positive when unpiping in Node.js 0.8

158.22 2.0.0 / 2015-05-08

- Return a promise without callback instead of thunk
- deps: bytes@2.0.1
  - units no longer case sensitive when parsing

158.23 1.3.4 / 2015-04-15

- Fix hanging callback if request aborts during read
- deps: iconv-lite@0.4.8
  - Add encoding alias UNICODE-1-1-UTF-7

158.24 1.3.3 / 2015-02-08

- deps: iconv-lite@0.4.7
  - Gracefully support enumerables on `Object.prototype`

158.25 1.3.2 / 2015-01-20

- deps: iconv-lite@0.4.6
  - Fix rare aliases of single-byte encodings

158.26 1.3.1 / 2014-11-21

- deps: iconv-lite@0.4.5
  - Fix Windows-31J and X-SJIS encoding support

158.27 1.3.0 / 2014-07-20

- Fully unpipe the stream on error
  - Fixes `Cannot switch to old mode now` error on Node.js 0.10+

158.28 1.2.3 / 2014-07-20

- deps: iconv-lite@0.4.4
  - Added encoding UTF-7

158.29 1.2.2 / 2014-06-19

- Send invalid encoding error to callback

158.30 1.2.1 / 2014-06-15

- deps: iconv-lite@0.4.3
  - Added encodings UTF-16BE and UTF-16 with BOM

158.31 1.2.0 / 2014-06-13

- Passing string as `options` interpreted as encoding
- Support all encodings from `iconv-lite`

158.32 1.1.7 / 2014-06-12

- use `string_decoder` module from npm

158.33 1.1.6 / 2014-05-27

- check encoding for old streams1
- support node.js < 0.10.6

158.34 1.1.5 / 2014-05-14

- bump bytes

158.35 1.1.4 / 2014-04-19

- allow true as an option
- bump bytes

158.36 1.1.3 / 2014-03-02

- fix case when length=null

158.37 1.1.2 / 2013-12-01

- be less strict on state.encoding check

158.38 1.1.1 / 2013-11-27

- add engines

158.39 1.1.0 / 2013-11-27

- add err.statusCode and err.type
- allow for encoding option to be true
- pause the stream instead of dumping on error
- throw if the stream's encoding is set

158.40 1.0.1 / 2013-11-19

- dont support streams1, throw if dev set encoding

158.41 1.0.0 / 2013-11-17

- rename `expected` option to `length`

158.42 0.2.0 / 2013-11-15

- republish

158.43 0.1.1 / 2013-11-15

- use bytes

158.44 0.1.0 / 2013-11-11

- generator support

158.45 0.0.3 / 2013-10-10

- update repo

158.46 0.0.2 / 2013-09-14

- dump stream on bad headers
- listen to events after defining received and buffers

158.47 0.0.1 / 2013-09-14

- Initial release

Chapter 159

raw-body

Gets the entire buffer of a stream either as a `Buffer` or a string. Validates the stream's length against an expected length and maximum limit. Ideal for parsing request bodies.

159.1 Install

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install raw-body
```

159.1.1 TypeScript

This module includes a `TypeScript` declaration file to enable auto complete in compatible editors and type information for TypeScript projects. This module depends on the Node.js types, so install `@types/node`:

```
$ npm install @types/node
```

159.2 API

```
var getRawBody = require('raw-body')
```

159.2.1 getRawBody(stream, [options], [callback])

Returns a promise if no callback specified and global `Promise` exists.
Options:

- `length` - The length of the stream. If the contents of the stream do not add up to this length, an `400` error code is returned.
- `limit` - The byte limit of the body. This is the number of bytes or any string format supported by `bytes`, for example `1000`, "500kb'`or`'3mb"`. If the body ends up being larger than this limit, a `413` error code is returned.
- `encoding` - The encoding to use to decode the body into a string. By default, a `Buffer` instance will be returned when no encoding is specified. Most likely, you want `utf-8`, so setting `encoding` to `true` will decode as `utf-8`. You can use any type of encoding supported by `iconv-lite`.

You can also pass a string in place of options to just specify the encoding.
If an error occurs, the stream will be paused, everything unpiped, and you are responsible for correctly disposing the stream. For HTTP requests, you may need to finish consuming the stream if you want to keep the socket open for future requests. For streams that use file descriptors, you should `stream.destroy()` or `stream.close()` to prevent leaks.

159.3 Errors

This module creates errors depending on the error condition during reading. The error may be an error from the underlying Node.js implementation, but is otherwise an error created by this module, which has the following attributes:

- `limit` - the limit in bytes
- `length` and `expected` - the expected length of the stream
- `received` - the received bytes
- `encoding` - the invalid encoding
- `status` and `statusCode` - the corresponding status code for the error
- `type` - the error type

159.3.1 Types

The errors from this module have a `type` property which allows for the programmatic determination of the type of error returned.

159.3.1.1 encoding.unsupported

This error will occur when the `encoding` option is specified, but the value does not map to an encoding supported by the `iconv-lite` module.

159.3.1.2 entity.too.large

This error will occur when the `limit` option is specified, but the stream has an entity that is larger.

159.3.1.3 request.aborted

This error will occur when the request stream is aborted by the client before reading the body has finished.

159.3.1.4 request.size.invalid

This error will occur when the `length` option is specified, but the stream has emitted more bytes.

159.3.1.5 stream.encoding.set

This error will occur when the given stream has an encoding set on it, making it a decoded stream. The stream should not have an encoding set and is expected to emit `Buffer` objects.

159.3.1.6 stream.not.readable

This error will occur when the given stream is not readable.

159.4 Examples

159.4.1 Simple Express example

```
var contentType = require('content-type')
var express = require('express')
var getRawBody = require('raw-body')
var app = express()
app.use(function (req, res, next) {
  getRawBody(req, {
    length: req.headers['content-length'],
    limit: '1mb',
    encoding: contentType.parse(req).parameters.charset
  }, function (err, string) {
    if (err) return next(err)
    req.text = string
    next()
  })
})
// now access req.text
```

159.4.2 Simple Koa example

```
var contentType = require('content-type')
var getRawBody = require('raw-body')
var koa = require('koa')
var app = koa()
app.use(function * (next) {
  this.text = yield getRawBody(this.req, {
    length:  this.req.headers['content-length'],
    limit:  '1mb',
    encoding:  contentType.parse(this.req).parameters.charset
  })
  yield next
})
// now access this.text
```

159.4.3 Using as a promise

To use this library as a promise, simply omit the `callback` and a promise is returned, provided that a global `Promise` is defined.

```
var getRawBody = require('raw-body')
var http = require('http')
var server = http.createServer(function (req, res) {
  getRawBody(req)
    .then(function (buf) {
      res.statusCode = 200
      res.end(buf.length + ' bytes submitted')
    })
    .catch(function (err) {
      res.statusCode = 500
      res.end(err.message)
    })
})
server.listen(3000)
```

159.4.4 Using with TypeScript

```
import * as getRawBody from 'raw-body';
import * as http from 'http';
const server = http.createServer((req, res) => {
  getRawBody(req)
  .then((buf) => {
    res.statusCode = 200;
    res.end(buf.length + ' bytes submitted');
  })
  .catch((err) => {
    res.statusCode = err.statusCode;
    res.end(err.message);
  });
});
server.listen(3000);
```

159.5 License

[MIT](LICENSE)

Chapter 160

Security Policies and Procedures

160.1 Reporting a Bug

The `raw-body` team and community take all security bugs seriously. Thank you for improving the security of Express. We appreciate your efforts and responsible disclosure and will make every effort to acknowledge your contributions.
Report security bugs by emailing the current owners of `raw-body`. This information can be found in the npm registry using the command `npm owner ls raw-body`. If unsure or unable to get the information from the above, open an issue in the `project issue tracker` asking for the current contact information.
To ensure the timely response to your report, please ensure that the entirety of the report is contained within the email body and not solely behind a web link or an attachment.
At least one owner will acknowledge your email within 48 hours, and will send a more detailed response within 48 hours indicating the next steps in handling your report. After the initial reply to your report, the owners will endeavor to keep you informed of the progress towards a fix and full announcement, and may ask for additional information or guidance.

Chapter 161

safe-buffer <a href="https://travis-ci.org/feross/safe-buffer" ><img src="https://img.shields.↩ io/travis/feross/safe-buffer/master.svg" alt="travis"/></a> <a href="https://npmjs.org/package/safe-buffer" ><img src="https://img.shields.io/npm/v/safe-buffer.svg" alt="npm"/></a> <a href="https://npmjs.org/package/safe-buffer" ><img src="https://img.shields.io/npm/dm/safe-buffer.svg" alt="downloads"/></a> <a href="https://standardjs.com" ><img src="https://img.shields.io/badge/code_style-standard-brightgreen.svg" alt="javascript style guide"/></a>

161.0.0.1 Safer Node.js Buffer API

∗∗Use the new Node.js Buffer APIs (`Buffer.from`, `Buffer.alloc`, `Buffer.allocUnsafe`, `Buffer.↩ allocUnsafeSlow`) in all versions of Node.js.∗∗

safe-buffer <a href="https://travis-ci.org/feross/safe-buffer" ><img src="https://img.shields.io/travis/feross/safe-buffer/master.svg" alt="travis"/></a> <a href="https://npmjs.org/package/safe-buffer" ><img src="https://img.shields.io/npm/v/safe-buffer.svg" alt="npm"/></a> <a href="https://npmjs.org/package/safe-buffer" ><img src="https://img.shields.io/npm/dm/safe-buffer.svg" alt="downloads"/></a> <a href="https://standardjs.com" ><img src="https://img.shields.io/badge/code_style-standard-brightgreen.svg" alt="javascript style guide"/></a>

Uses the built-in implementation when available.

161.1 install

```
npm install safe-buffer
```

161.2 usage

The goal of this package is to provide a safe replacement for the node.js `Buffer`.
It's a drop-in replacement for `Buffer`. You can use it by adding one `require` line to the top of your node.js modules:

```
var Buffer = require('safe-buffer').Buffer
// Existing buffer code will continue to work without issues:
new Buffer('hey', 'utf8')
new Buffer([1, 2, 3], 'utf8')
new Buffer(obj)
new Buffer(16) // create an uninitialized buffer (potentially unsafe)
// But you can use these new explicit APIs to make clear what you want:
Buffer.from('hey', 'utf8') // convert from many types to a Buffer
Buffer.alloc(16) // create a zero-filled buffer (safe)
Buffer.allocUnsafe(16) // create an uninitialized buffer (potentially unsafe)
```

161.3 api

161.3.1 Class Method: Buffer.from(array)

- `array` {Array}

Allocates a new `Buffer` using an `array` of octets.

```
const buf = Buffer.from([0x62,0x75,0x66,0x66,0x65,0x72]);
  // creates a new Buffer containing ASCII bytes
  // ['b','u','f','f','e','r']
```

A `TypeError` will be thrown if `array` is not an `Array`.

161.3.2 Class Method: Buffer.from(arrayBuffer[, byteOffset[, length]])

- `arrayBuffer` {ArrayBuffer} The `.buffer` property of a `TypedArray` or a `new ArrayBuffer()`
- `byteOffset` {Number} Default: `0`
- `length` {Number} Default: `arrayBuffer.length - byteOffset`

When passed a reference to the `.buffer` property of a `TypedArray` instance, the newly created `Buffer` will share the same allocated memory as the TypedArray.

```
const arr = new Uint16Array(2);
arr[0] = 5000;
arr[1] = 4000;
const buf = Buffer.from(arr.buffer); // shares the memory with arr;
console.log(buf);
  // Prints:  <Buffer 88 13 a0 0f>
// changing the TypedArray changes the Buffer also
arr[1] = 6000;
console.log(buf);
  // Prints:  <Buffer 88 13 70 17>
```

The optional `byteOffset` and `length` arguments specify a memory range within the `arrayBuffer` that will be shared by the `Buffer`.

```
const ab = new ArrayBuffer(10);
const buf = Buffer.from(ab, 0, 2);
console.log(buf.length);
  // Prints:  2
```

A `TypeError` will be thrown if `arrayBuffer` is not an `ArrayBuffer`.

161.3.3 Class Method: Buffer.from(buffer)

- `buffer` {Buffer}

Copies the passed `buffer` data onto a new `Buffer` instance.
```
const buf1 = Buffer.from('buffer');
const buf2 = Buffer.from(buf1);
buf1[0] = 0x61;
console.log(buf1.toString());
  // 'auffer'
console.log(buf2.toString());
  // 'buffer' (copy is not changed)
```
A `TypeError` will be thrown if `buffer` is not a `Buffer`.

161.3.4 Class Method: Buffer.from(str[, encoding])

- `str` {String} String to encode.
- `encoding` {String} Encoding to use, Default: "utf8\`

Creates a new `Buffer` containing the given JavaScript string `str`. If provided, the `encoding` parameter identifies the character encoding. If not provided, `encoding` defaults to "utf8\`.
```
const buf1 = Buffer.from('this is a tést');
console.log(buf1.toString());
  // prints: this is a tést
console.log(buf1.toString('ascii'));
  // prints: this is a tC)st
const buf2 = Buffer.from('7468697320697320612074c3a97374', 'hex');
console.log(buf2.toString());
  // prints: this is a tést
```
A `TypeError` will be thrown if `str` is not a string.

161.3.5 Class Method: Buffer.alloc(size[, fill[, encoding]])

- `size` {Number}
- `fill` {Value} Default: `undefined`
- `encoding` {String} Default: `utf8`

Allocates a new `Buffer` of `size` bytes. If `fill` is `undefined`, the `Buffer` will be *zero-filled*.
```
const buf = Buffer.alloc(5);
console.log(buf);
  // <Buffer 00 00 00 00 00>
```
The `size` must be less than or equal to the value of 'require('buffer').kMaxLength `(on 64-bit architectures,`k↩
MaxLength`is` (2^31)-1`). Otherwise, a [`RangeError`][] is thrown. A zero-length Buffer will be created if a`size\` less than or equal to 0 is specified.
If `fill` is specified, the allocated `Buffer` will be initialized by calling `buf.fill(fill)`. See [`buf.fill()`][] for more information.
```
const buf = Buffer.alloc(5, 'a');
console.log(buf);
  // <Buffer 61 61 61 61 61>
```
If both `fill` and `encoding` are specified, the allocated `Buffer` will be initialized by calling `buf.fill(fill, encoding)`. For example:
```
const buf = Buffer.alloc(11, 'aGVsbG8gd29ybGQ=', 'base64');
console.log(buf);
  // <Buffer 68 65 6c 6c 6f 20 77 6f 72 6c 64>
```
Calling `Buffer.alloc(size)` can be significantly slower than the alternative `Buffer.alloc↩ Unsafe(size)` but ensures that the newly created `Buffer` instance contents will *never contain sensitive data*.
A `TypeError` will be thrown if `size` is not a number.

161.3.6 Class Method: Buffer.allocUnsafe(size)

- `size` {Number}

Allocates a new *non-zero-filled* `Buffer` of `size` bytes. The `size` must be less than or equal to the value of 'require('buffer').kMaxLength `(on 64-bit architectures,`kMaxLength`is`(2^31)-1`). Otherwise, a [`RangeError`][] is thrown. A zero-length Buffer will be created if a`size\` less than or equal to 0 is specified.
The underlying memory for `Buffer` instances created in this way is *not initialized*. The contents of the newly created `Buffer` are unknown and *may contain sensitive data*. Use [`buf.fill(0)`][] to initialize such `Buffer` instances to zeroes.

```
const buf = Buffer.allocUnsafe(5);
console.log(buf);
  // <Buffer 78 e0 82 02 01>
  // (octets will be different, every time)
buf.fill(0);
console.log(buf);
// <Buffer 00 00 00 00 00>
```

A `TypeError` will be thrown if `size` is not a number.

Note that the `Buffer` module pre-allocates an internal `Buffer` instance of size `Buffer.poolSize` that is used as a pool for the fast allocation of new `Buffer` instances created using `Buffer.allocUnsafe(size)` (and the deprecated `new Buffer(size)` constructor) only when `size` is less than or equal to `Buffer.poolSize >> 1` (floor of `Buffer.poolSize` divided by two). The default value of `Buffer.poolSize` is `8192` but can be modified.

Use of this pre-allocated internal memory pool is a key difference between calling `Buffer.alloc(size, fill)` vs. `Buffer.allocUnsafe(size).fill(fill)`. Specifically, `Buffer.alloc(size, fill)` will *never* use the internal Buffer pool, while `Buffer.allocUnsafe(size).fill(fill)` *will* use the internal Buffer pool if `size` is less than or equal to half `Buffer.poolSize`. The difference is subtle but can be important when an application requires the additional performance that `Buffer.allocUnsafe(size)` provides.

161.3.7 Class Method: Buffer.allocUnsafeSlow(size)

- `size` {Number}

Allocates a new *non-zero-filled* and non-pooled `Buffer` of `size` bytes. The `size` must be less than or equal to the value of ‘require('buffer').kMaxLength`(on 64-bit architectures,`kMaxLength`is` (2^31)-1`). Otherwise, a `[RangeError`][] is thrown. A zero-length Buffer will be created if a`size` less than or equal to 0 is specified.

The underlying memory for `Buffer` instances created in this way is *not initialized*. The contents of the newly created `Buffer` are unknown and *may contain sensitive data*. Use [`buf.fill(0)`][] to initialize such `Buffer` instances to zeroes.

When using `Buffer.allocUnsafe()` to allocate new `Buffer` instances, allocations under 4KB are, by default, sliced from a single pre-allocated `Buffer`. This allows applications to avoid the garbage collection overhead of creating many individually allocated Buffers. This approach improves both performance and memory usage by eliminating the need to track and cleanup as many `Persistent` objects.

However, in the case where a developer may need to retain a small chunk of memory from a pool for an indeterminate amount of time, it may be appropriate to create an un-pooled Buffer instance using `Buffer.allocUnsafeSlow()` then copy out the relevant bits.

```
// need to keep around a few small chunks of memory
const store = [];
socket.on('readable', () => {
  const data = socket.read();
  // allocate for retained data
  const sb = Buffer.allocUnsafeSlow(10);
  // copy the data into the new allocation
  data.copy(sb, 0, 0, 10);
  store.push(sb);
});
```

Use of `Buffer.allocUnsafeSlow()` should be used only as a last resort *after* a developer has observed undue memory retention in their applications.

A `TypeError` will be thrown if `size` is not a number.

161.3.8 All the Rest

The rest of the `Buffer` API is exactly the same as in node.js. See the docs.

161.4 Related links

- Node.js issue: Buffer(number) is unsafe
- Node.js Enhancement Proposal: Buffer.from/Buffer.alloc/Buffer.zalloc/Buffer() soft-deprecate

161.5 Why is <tt>Buffer</tt> unsafe?

Today, the node.js `Buffer` constructor is overloaded to handle many different argument types like `String`, `Array`, `Object`, `TypedArrayView` (`Uint8Array`, etc.), `ArrayBuffer`, and also `Number`.
The API is optimized for convenience: you can throw any type at it, and it will try to do what you want.
Because the Buffer constructor is so powerful, you often see code like this:

```
// Convert UTF-8 strings to hex
function toHex (str) {
  return new Buffer(str).toString('hex')
}
```

But what happens if `toHex` is called with a `Number` argument?

161.5.1 Remote Memory Disclosure

If an attacker can make your program call the `Buffer` constructor with a `Number` argument, then they can make it allocate uninitialized memory from the node.js process. This could potentially disclose TLS private keys, user data, or database passwords.
When the `Buffer` constructor is passed a `Number` argument, it returns an **UNINITIALIZED** block of memory of the specified `size`. When you create a `Buffer` like this, you **MUST** overwrite the contents before returning it to the user.
From the `node.js docs`:

> `new Buffer(size)`
>
> - `size` Number
>
> The underlying memory for `Buffer` instances created in this way is not initialized. **The contents of a newly created `Buffer` are unknown and could contain sensitive data.** Use `buf.fill(0)` to initialize a Buffer to zeroes.

(Emphasis our own.)
Whenever the programmer intended to create an uninitialized `Buffer` you often see code like this:

```
var buf = new Buffer(16)
// Immediately overwrite the uninitialized buffer with data from another buffer
for (var i = 0; i < buf.length; i++) {
  buf[i] = otherBuf[i]
}
```

161.5.2 Would this ever be a problem in real code?

Yes. It's surprisingly common to forget to check the type of your variables in a dynamically-typed language like JavaScript.
Usually the consequences of assuming the wrong type is that your program crashes with an uncaught exception. But the failure mode for forgetting to check the type of arguments to the `Buffer` constructor is more catastrophic.
Here's an example of a vulnerable service that takes a JSON payload and converts it to hex:

```
// Take a JSON payload {str: "some string"} and convert it to hex
var server = http.createServer(function (req, res) {
  var data = ''
  req.setEncoding('utf8')
  req.on('data', function (chunk) {
    data += chunk
  })
  req.on('end', function () {
    var body = JSON.parse(data)
    res.end(new Buffer(body.str).toString('hex'))
  })
})
server.listen(8080)
```

In this example, an http client just has to send:

```
{
  "str": 1000
}
```

and it will get back 1,000 bytes of uninitialized memory from the server.
This is a very serious bug. It's similar in severity to the `the Heartbleed bug` that allowed disclosure of OpenSSL process memory by remote attackers.

161.5.3 Which real-world packages were vulnerable?

161.5.3.1 <a href="https://www.npmjs.com/package/bittorrent-dht" ><tt>bittorrent-dht</tt></a>

Mathias Buus and I (Feross Aboukhadijeh) found this issue in one of our own packages, bittorrent-dht. The bug would allow anyone on the internet to send a series of messages to a user of bittorrent-dht and get them to reveal 20 bytes at a time of uninitialized memory from the node.js process.
Here's the commit that fixed it. We released a new fixed version, created a Node Security Project disclosure, and deprecated all vulnerable versions on npm so users will get a warning to upgrade to a newer version.

161.5.3.2 <a href="https://www.npmjs.com/package/ws" ><tt>ws</tt></a>

That got us wondering if there were other vulnerable packages. Sure enough, within a short period of time, we found the same issue in ws, the most popular WebSocket implementation in node.js.
If certain APIs were called with `Number` parameters instead of `String` or `Buffer` as expected, then uninitialized server memory would be disclosed to the remote peer.
These were the vulnerable methods:

```
socket.send(number)
socket.ping(number)
socket.pong(number)
```

Here's a vulnerable socket server with some echo functionality:

```
server.on('connection', function (socket) {
  socket.on('message', function (message) {
    message = JSON.parse(message)
    if (message.type === 'echo') {
      socket.send(message.data) // send back the user's message
    }
  })
})
```

`socket.send(number)` called on the server, will disclose server memory.
Here's the release where the issue was fixed, with a more detailed explanation. Props to Arnout Kazemier for the quick fix. Here's the Node Security Project disclosure.

161.5.4 What's the solution?

It's important that node.js offers a fast way to get memory otherwise performance-critical applications would needlessly get a lot slower.
But we need a better way to *signal our intent* as programmers. **When we want uninitialized memory, we should request it explicitly.**
Sensitive functionality should not be packed into a developer-friendly API that loosely accepts many different types. This type of API encourages the lazy practice of passing variables in without checking the type very carefully.

161.5.4.1 A new API: <tt>Buffer.allocUnsafe(number)</tt>

The functionality of creating buffers with uninitialized memory should be part of another API. We propose `Buffer.allocUnsafe(number)`. This way, it's not part of an API that frequently gets user input of all sorts of different types passed into it.

```
var buf = Buffer.allocUnsafe(16) // careful, uninitialized memory!
// Immediately overwrite the uninitialized buffer with data from another buffer
for (var i = 0; i < buf.length; i++) {
  buf[i] = otherBuf[i]
}
```

161.5.5 How do we fix node.js core?

We sent a PR to node.js core (merged as `semver-major`) which defends against one case:

```
var str = 16
new Buffer(str, 'utf8')
```

In this situation, it's implied that the programmer intended the first argument to be a string, since they passed an encoding as a second argument. Today, node.js will allocate uninitialized memory in the case of `new Buffer(number, encoding)`, which is probably not what the programmer intended.
But this is only a partial solution, since if the programmer does `new Buffer(variable)` (without an `encoding` parameter) there's no way to know what they intended. If `variable` is sometimes a number, then uninitialized memory will sometimes be returned.

161.5.6 What's the real long-term fix?

We could deprecate and remove `new Buffer(number)` and use `Buffer.allocUnsafe(number)` when we need uninitialized memory. But that would break 1000s of packages.
~~We believe the best solution is to:~~
~~1. Change `new Buffer(number)` to return safe, zeroed-out memory~~
~~2. Create a new API for creating uninitialized Buffers. We propose: `Buffer.allocUnsafe(number)`~~

161.5.6.1 Update

We now support adding three new APIs:

- `Buffer.from(value)` - convert from any type to a buffer
- `Buffer.alloc(size)` - create a zero-filled buffer
- `Buffer.allocUnsafe(size)` - create an uninitialized buffer with given size

This solves the core problem that affected `ws` and `bittorrent-dht` which is `Buffer(variable)` getting tricked into taking a number argument.
This way, existing code continues working and the impact on the npm ecosystem will be minimal. Over time, npm maintainers can migrate performance-critical code to use `Buffer.allocUnsafe(number)` instead of `new Buffer(number)`.

161.5.7 Conclusion

We think there's a serious design issue with the `Buffer` API as it exists today. It promotes insecure software by putting high-risk functionality into a convenient API with friendly "developer ergonomics".
This wasn't merely a theoretical exercise because we found the issue in some of the most popular npm packages.
Fortunately, there's an easy fix that can be applied today. Use `safe-buffer` in place of `buffer`.

```
var Buffer = require('safe-buffer').Buffer
```

Eventually, we hope that node.js core can switch to this new, safer behavior. We believe the impact on the ecosystem would be minimal since it's not a breaking change. Well-maintained, popular packages would be updated to use `Buffer.alloc` quickly, while older, insecure packages would magically become safe from this attack vector.

161.6 links

- `Node.js PR: buffer: throw if both length and enc are passed`
- `Node Security Project disclosure for ws`
- `Node Security Project disclosure forbittorrent-dht`

161.7 credit

The original issues in `bittorrent-dht` (`disclosure`) and `ws` (`disclosure`) were discovered by `Mathias Buus` and `Feross Aboukhadijeh`.
Thanks to `Adam Baldwin` for helping disclose these issues and for his work running the `Node Security Project`.
Thanks to `John Hiesey` for proofreading this README and auditing the code.

161.8 license

MIT. Copyright (C) `Feross Aboukhadijeh`

Chapter 162

Porting to the Buffer.from/Buffer.alloc API

162.1 Overview

- Variant 1: Drop support for Node.js 4.4.x and 5.0.0 — 5.9.x. (*recommended*)
- Variant 2: Use a polyfill
- Variant 3: manual detection, with safeguards

162.1.1 Finding problematic bits of code using grep

Just run ‘grep -nrE ’[^a-zA-Z](Slow)?Buffer\s∗(' –exclude-dir node_modules`.
It will find all the potentially unsafe places in your own code (with some considerably unlikely exceptions).

162.1.2 Finding problematic bits of code using Node.js 8

If you’re using Node.js 8.0.0 (which is recommended), Node.js exposes multiple options that help with finding the relevant pieces of code:

- `--trace-warnings` will make Node.js show a stack trace for this warning and other warnings that are printed by Node.js.
- `--trace-deprecation` does the same thing, but only for deprecation warnings.
- `--pending-deprecation` will show more types of deprecation warnings. In particular, it will show the `Buffer()` deprecation warning, even on Node.js 8.

You can set these flags using an environment variable:
```
$ export NODE_OPTIONS='--trace-warnings --pending-deprecation'
$ cat example.js
'use strict';
const foo = new Buffer('foo');
$ node example.js
(node:7147) [DEP0005] DeprecationWarning:  The Buffer() and new Buffer() constructors are not recommended
      for use due to security and usability concerns.  Please use the new Buffer.alloc(),
      Buffer.allocUnsafe(), or Buffer.from() construction methods instead.
    at showFlaggedDeprecation (buffer.js:127:13)
    at new Buffer (buffer.js:148:3)
    at Object.<anonymous> (/path/to/example.js:2:13)
    [...  more stack trace lines ...]
```

162.1.3 Finding problematic bits of code using linters

Eslint rules `no-buffer-constructor` or `node/no-deprecated-api` also find calls to deprecated `Buffer()` API. Those rules are included in some pre-sets.
There is a drawback, though, that it doesn't always `work correctly` when `Buffer` is overriden e.g. with a polyfill, so recommended is a combination of this and some other method described above.

162.2 Variant 1: Drop support for Node.js 4.4.x and 5.0.0 — 5.9.x.

This is the recommended solution nowadays that would imply only minimal overhead.
The Node.js 5.x release line has been unsupported since July 2016, and the Node.js 4.x release line reaches its End of Life in April 2018 (→ `Schedule`). This means that these versions of Node.js will *not* receive any updates, even in case of security issues, so using these release lines should be avoided, if at all possible.
What you would do in this case is to convert all `new Buffer()` or `Buffer()` calls to use `Buffer.alloc()` or `Buffer.from()`, in the following way:

- For `new Buffer(number)`, replace it with `Buffer.alloc(number)`.
- For `new Buffer(string)` (or `new Buffer(string, encoding)`), replace it with `Buffer.from(string)` (or `Buffer.from(string, encoding)`).
- For all other combinations of arguments (these are much rarer), also replace `new Buffer(...arguments)` with `Buffer.from(...arguments)`.

Note that `Buffer.alloc()` is also *faster* on the current Node.js versions than `new Buffer(size).fill(0)`, which is what you would otherwise need to ensure zero-filling.
Enabling eslint rule `no-buffer-constructor` or `node/no-deprecated-api` is recommended to avoid accidential unsafe Buffer API usage.
There is also a `JSCodeshift codemod` for automatically migrating Buffer constructors to `Buffer.alloc()` or `Buffer.from()`. Note that it currently only works with cases where the arguments are literals or where the constructor is invoked with two arguments.
If you currently support those older Node.js versions and dropping them would be a semver-major change for you, or if you support older branches of your packages, consider using Variant 2 or Variant 3 on older branches, so people using those older branches will also receive the fix. That way, you will eradicate potential issues caused by unguarded Buffer API usage and your users will not observe a runtime deprecation warning when running your code on Node.js 10.

162.3 Variant 2: Use a polyfill

Utilize `safer-buffer` as a polyfill to support older Node.js versions.
You would take exacly the same steps as in Variant 1, but with a polyfill ‘const Buffer = require('safer-buffer').Buffer`in all files where you use the new`Buffer` api.
Make sure that you do not use old `new Buffer` API — in any files where the line above is added, using old `new Buffer()` API will *throw*. It will be easy to notice that in CI, though.
Alternatively, you could use `buffer-from` and/or `buffer-alloc` `ponyfills` — those are great, the only downsides being 4 deps in the tree and slightly more code changes to migrate off them (as you would be using e.g. `Buffer.from` under a different name). If you need only `Buffer.from` polyfilled — `buffer-from` alone which comes with no extra dependencies.
Alternatively, you could use `safe-buffer` — it also provides a polyfill, but takes a different approach which has `it's drawbacks`. It will allow you to also use the older `new Buffer()` API in your code, though — but that's arguably a benefit, as it is problematic, can cause issues in your code, and will start emitting runtime deprecation warnings starting with Node.js 10.
Note that in either case, it is important that you also remove all calls to the old Buffer API manually — just throwing in `safe-buffer` doesn't fix the problem by itself, it just provides a polyfill for the new API. I have seen people doing that mistake.
Enabling eslint rule `no-buffer-constructor` or `node/no-deprecated-api` is recommended.
Don't forget to drop the polyfill usage once you drop support for Node.js $<$ 4.5.0.

162.4 Variant 3 — manual detection, with safeguards

This is useful if you create Buffer instances in only a few places (e.g. one), or you have your own wrapper around them.

162.4.1 Buffer(0)

This special case for creating empty buffers can be safely replaced with `Buffer.concat([])`, which returns the same result all the way down to Node.js 0.8.x.

162.4.2 Buffer(notNumber)

Before:

```
var buf = new Buffer(notNumber, encoding);
```

After:

```
var buf;
if (Buffer.from && Buffer.from !== Uint8Array.from) {
  buf = Buffer.from(notNumber, encoding);
} else {
  if (typeof notNumber === 'number')
    throw new Error('The "size" argument must be of type number.');
  buf = new Buffer(notNumber, encoding);
}
```

`encoding` is optional.
Note that the `typeof notNumber` before `new Buffer` is required (for cases when `notNumber` argument is not hard-coded) and *is not caused by the deprecation of Buffer constructor* — it's exactly *why* the Buffer constructor is deprecated. Ecosystem packages lacking this type-check caused numereous security issues — situations when unsanitized user input could end up in the `Buffer(arg)` create problems ranging from DoS to leaking sensitive information to the attacker from the process memory.
When `notNumber` argument is hardcoded (e.g. literal `"abc"` or `[0,1,2]`), the `typeof` check can be omitted.
Also note that using TypeScript does not fix this problem for you — when libs written in `TypeScript` are used from JS, or when user input ends up there — it behaves exactly as pure JS, as all type checks are translation-time only and are not present in the actual JS code which TS compiles to.

162.4.3 Buffer(number)

For Node.js 0.10.x (and below) support:

```
var buf;
if (Buffer.alloc) {
  buf = Buffer.alloc(number);
} else {
  buf = new Buffer(number);
  buf.fill(0);
}
```

Otherwise (Node.js 0.12.x):

```
const buf = Buffer.alloc ?  Buffer.alloc(number) :  new Buffer(number).fill(0);
```

162.5 Regarding Buffer.allocUnsafe

Be extra cautious when using `Buffer.allocUnsafe`:

- Don't use it if you don't have a good reason to
  - e.g. you probably won't ever see a performance difference for small buffers, in fact, those might be even faster with `Buffer.alloc()`,
  - if your code is not in the hot code path — you also probably won't notice a difference,
  - keep in mind that zero-filling minimizes the potential risks.
- If you use it, make sure that you never return the buffer in a partially-filled state,
  - if you are writing to it sequentially — always truncate it to the actuall written length

Errors in handling buffers allocated with `Buffer.allocUnsafe` could result in various issues, ranged from undefined behaviour of your code to sensitive data (user input, passwords, certs) leaking to the remote attacker.
Note that the same applies to `new Buffer` usage without zero-filling, depending on the Node.js version (and lacking type checks also adds DoS to the list of potential problems).

162.6 FAQ

162.6.1 What is wrong with the <tt>Buffer</tt> constructor?

The `Buffer` constructor could be used to create a buffer in many different ways:

- `new Buffer(42)` creates a `Buffer` of 42 bytes. Before Node.js 8, this buffer contained *arbitrary memory* for performance reasons, which could include anything ranging from program source code to passwords and encryption keys.
- 'new Buffer('abc')`creates a`Buffer`that contains the UTF-8-encoded version of the string`'abc'`. A second argument could specify another encoding: For example,` new Buffer(string, 'base64')` could be used to convert a Base64 string into the original sequence of bytes that it represents.
- There are several other combinations of arguments.

This meant that, in code like `var buffer = new Buffer(foo);`, *it is not possible to tell what exactly the contents of the generated buffer are* without knowing the type of `foo`.
Sometimes, the value of `foo` comes from an external source. For example, this function could be exposed as a service on a web server, converting a UTF-8 string into its Base64 form:
```
function stringToBase64(req, res) {
  // The request body should have the format of `{ string:  'foobar' }`
  const rawBytes = new Buffer(req.body.string)
  const encoded = rawBytes.toString('base64')
  res.end({ encoded:  encoded })
}
```
Note that this code does *not* validate the type of `req.body.string`:

- `req.body.string` is expected to be a string. If this is the case, all goes well.
- `req.body.string` is controlled by the client that sends the request.
- If `req.body.string` is the *number* `50`, the `rawBytes` would be 50 bytes:
  - Before Node.js 8, the content would be uninitialized
  - After Node.js 8, the content would be `50` bytes with the value `0`

Because of the missing type check, an attacker could intentionally send a number as part of the request. Using this, they can either:

- Read uninitialized memory. This **will** leak passwords, encryption keys and other kinds of sensitive information. (Information leak)
- Force the program to allocate a large amount of memory. For example, when specifying `500000000` as the input value, each request will allocate 500MB of memory. This can be used to either exhaust the memory available of a program completely and make it crash, or slow it down significantly. (Denial of Service)

Both of these scenarios are considered serious security issues in a real-world web server context.
when using `Buffer.from(req.body.string)` instead, passing a number will always throw an exception instead, giving a controlled behaviour that can always be handled by the program.

162.6.2 The <tt>Buffer()</tt> constructor has been deprecated for a while. Is this really an issue?

Surveys of code in the `npm` ecosystem have shown that the `Buffer()` constructor is still widely used. This includes new code, and overall usage of such code has actually been *increasing*.

Chapter 163

safer-buffer <a href="https://travis-ci.org/ChALkeR/safer-buffer" ><img src="https://travis-ci.org/ChALkeR/safer-buffer.svg?branch=master" alt="travis"/></a> <a href="https://npmjs.org/package/safer-buffer" ><img src="https://img.shields.io/npm/v/safer-buffer.svg" alt="npm"/></a> <a href="https://standardjs.com" ><img src="https://img.shields.io/badge/code_style-standard-brightgreen.svg" alt="javascript style guide"/></a> <a href="https://github.com/nodejs/security-wg/blob/master/processes/responsible_disclosure↩_template.md" ><img src="https://img.shields.io/badge/Security-↩Responsible%20Disclosure-green.svg" alt="Security Responsible Disclosure"/></a>

Modern Buffer API polyfill without footguns, working on Node.js from 0.8 to current.

163.1 How to use?

First, port all `Buffer()` and `new Buffer()` calls to `Buffer.alloc()` and `Buffer.from()` API.
Then, to achieve compatibility with outdated Node.js versions (`<4.5.0` and 5.x `<5.9.0`), use ‘const Buffer = require('safer-buffer').Buffer`in all files where you make calls to the new Buffer API.` Use`var`instead of`const` if you need that for your Node.js version range support.
Also, see the `porting Buffer` guide.

163.2 Do I need it?

Hopefully, not — dropping support for outdated Node.js versions should be fine nowdays, and that is the recommended path forward. You *do* need to port to the `Buffer.alloc()` and `Buffer.from()` though.
See the `porting guide` for a better description.

163.3 Why not <a href="https://npmjs.com/safe-buffer" >safe-buffer</a>?

In short: while `safe-buffer` serves as a polyfill for the new API, it allows old API usage and itself contains footguns.
`safe-buffer` could be used safely to get the new API while still keeping support for older Node.js versions (like this module), but while analyzing ecosystem usage of the old Buffer API I found out that `safe-buffer` is itself causing problems in some cases.
For example, consider the following snippet:

```
$ cat example.unsafe.js
console.log(Buffer(20))
$ ./node-v6.13.0-linux-x64/bin/node example.unsafe.js
<Buffer 0a 00 00 00 00 00 00 00 28 13 de 02 00 00 00 00 05 00 00 00>
$ standard example.unsafe.js
standard: Use JavaScript Standard Style (https://standardjs.com)
  /home/chalker/repo/safer-buffer/example.unsafe.js:2:13:  'Buffer()' was deprecated since v6.  Use
      'Buffer.alloc()' or 'Buffer.from()' (use 'https://www.npmjs.com/package/safe-buffer' for '<4.5.0')
      instead.
```

This is allocates and writes to console an uninitialized chunk of memory. `standard` linter (among others) catch that and warn people to avoid using unsafe API.
Let's now throw in `safe-buffer`!

```
$ cat example.safe-buffer.js
const Buffer = require('safe-buffer').Buffer
console.log(Buffer(20))
$ standard example.safe-buffer.js
$ ./node-v6.13.0-linux-x64/bin/node example.safe-buffer.js
<Buffer 08 00 00 00 00 00 00 00 28 58 01 82 fe 7f 00 00 00 00 00 00>
```

See the problem? Adding in `safe-buffer` *magically removes the lint warning*, but the behavior remains idential to what we had before, and when launched on Node.js 6.x LTS — this dumps out chunks of uninitialized memory. *And this code will still emit runtime warnings on Node.js 10.x and above.*
That was done by design. I first considered changing `safe-buffer`, prohibiting old API usage or emitting warnings on it, but that significantly diverges from `safe-buffer` design. After some discussion, it was decided to move my approach into a separate package, and *this is that separate package*.
This footgun is not imaginary — I observed top-downloaded packages doing that kind of thing, «fixing» the lint warning by blindly including `safe-buffer` without any actual changes.
Also in some cases, even if the API *was* migrated to use of safe Buffer API — a random pull request can bring unsafe Buffer API usage back to the codebase by adding new calls — and that could go unnoticed even if you have a linter prohibiting that (becase of the reason stated above), and even pass CI. *I also observed that being done in popular packages.*
Some examples:

- `webdriverio` (a module with 548 759 downloads/month),
- `websocket-stream` (218 288 d/m, fix in `maxogden/websocket-stream#142`),
- `node-serialport` (113 138 d/m, fix in `node-serialport/node-serialport#1510`),
- `karma` (3 973 193 d/m, fix in `karma-runner/karma#2947`),
- `spdy-transport` (5 970 727 d/m, fix in `spdy-http2/spdy-transport#53`).

- And there are a lot more over the ecosystem.

I filed a PR at mysticatea/eslint-plugin-node#110 to partially fix that (for cases when that lint rule is used), but it is a semver-major change for linter rules and presets, so it would take significant time for that to reach actual setups. *It also hasn't been released yet (2018-03-20).*
Also, `safer-buffer` discourages the usage of `.allocUnsafe()`, which is often done by a mistake. It still supports it with an explicit concern barier, by placing it under 'require('safer-buffer/dangereous')`.

163.4 But isn't throwing bad?

Not really. It's an error that could be noticed and fixed early, instead of causing havoc later like unguarded `new Buffer()` calls that end up receiving user input can do.
This package affects only the files where 'var Buffer = require('safer-buffer').Buffer` was done, so it is really simple to keep track of things and make sure that you don't mix old API usage with that. Also, CI should hint anything that you might have missed.
New commits, if tested, won't land new usage of unsafe Buffer API this way. *Node.js 10.x also deals with that by printing a runtime depecation warning.*

163.4.1 Would it affect third-party modules?

No, unless you explicitly do an awful thing like monkey-patching or overriding the built-in `Buffer`. Don't do that.

163.4.2 But I don't want throwing...

That is also fine!
Also, it could be better in some cases when you don't comprehensive enough test coverage.
In that case — just don't override `Buffer` and use 'var SaferBuffer = require('safer-buffer').Buffer` instead.
That way, everything using `Buffer` natively would still work, but there would be two drawbacks:

- `Buffer.from`/`Buffer.alloc` won't be polyfilled — use `SaferBuffer.from` and `SaferBuffer.alloc` instead.
- You are still open to accidentally using the insecure deprecated API — use a linter to catch that.

Note that using a linter to catch accidential `Buffer` constructor usage in this case is strongly recommended. `Buffer` is not overriden in this usecase, so linters won't get confused.

163.5 «Without footguns»?

Well, it is still possible to do *some* things with `Buffer` API, e.g. accessing `.buffer` property on older versions and duping things from there. You shouldn't do that in your code, probabably.
The intention is to remove the most significant footguns that affect lots of packages in the ecosystem, and to do it in the proper way.
Also, this package doesn't protect against security issues affecting some Node.js versions, so for usage in your own production code, it is still recommended to update to a Node.js version supported by upstream.

safer-buffer <a href="https://travis-ci.org/ChALkeR/safer-buffer" ><img src="https://travis-ci.org/ChALkeR/safer-buffer.svg?branch=master" alt="travis"/></a> <a href="https://npmjs.org/package/safer-buffer" ><img src="https://img.shields.io/npm/v/safer-buffer.svg" alt="npm"/></a> <a href="https://standardjs.com" ><img src="https://img.shields.io/badge/code_style-standard-brightgreen.svg" alt="javascript style guide"/></a> <a href="https://github.com/nodejs/security-wg/blob/master/processes/responsible_disclosure_template.md" ><img src="https://img.shields.io/badge/Security-Responsible%20Disclosure-green.svg" alt="Security Responsible Disclosure"/></a>

Chapter 164

semver-compare

compare two semver version strings, returning -1, 0, or 1
The return value can be fed straight into `[].sort`.

164.1 example

```
var cmp = require('semver-compare');
var versions = [
    '1.2.3',
    '4.11.6',
    '4.2.0',
    '1.5.19',
    '1.5.5',
    '4.1.3',
    '2.3.1',
    '10.5.5',
    '11.3.0'
];
console.log(versions.sort(cmp).join('\n'));
```

prints:

```
1.2.3
1.5.5
1.5.19
2.3.1
4.1.3
4.2.0
4.11.6
10.5.5
11.3.0
```

whereas the default lexicographic sort (`versions.sort()`) would be:

```
1.2.3
1.5.19
1.5.5
10.5.5
11.3.0
2.3.1
4.1.3
4.11.6
4.2.0
```

164.2 methods

```
var cmp = require('semver-compare')
```

164.2.1 cmp(a, b)

If the semver string `a` is greater than `b`, return `1`. If the semver string `b` is greater than `a`, return `-1`. If `a` equals `b`, return 0;

164.3 install

With `npm` do:

```
npm install semver-compare
```

164.4 license

MIT

Chapter 165

0.19.0 / 2024-09-10

- Remove link renderization in html while redirecting

165.1 0.18.0 / 2022-03-23

- Fix emitted 416 error missing headers property
- Limit the headers removed for 304 response
- deps: depd@2.0.0
  - Replace internal `eval` usage with `Function` constructor
  - Use instance methods on `process` to check for listeners
- deps: destroy@1.2.0
- deps: http-errors@2.0.0
  - deps: depd@2.0.0
  - deps: statuses@2.0.1
- deps: on-finished@2.4.1
- deps: statuses@2.0.1

165.2 0.17.2 / 2021-12-11

- pref: ignore empty http tokens
- deps: http-errors@1.8.1
  - deps: inherits@2.0.4
  - deps: toidentifier@1.0.1
  - deps: setprototypeof@1.2.0
- deps: ms@2.1.3

165.3 0.17.1 / 2019-05-10

- Set stricter CSP header in redirect & error responses
- deps: range-parser1.2.1

165.4 0.17.0 / 2019-05-03

- deps: http-errors1.7.2
  - Set constructor name when possible
  - Use `toidentifier` module to make class names
  - deps: depd1.1.2
  - deps: setprototypeof@1.1.1
  - deps: statuses@'>= 1.5.0 < 2'
- deps: mime@1.6.0
  - Add extensions for JPEG-2000 images
  - Add new `font/*` types from IANA
  - Add WASM mapping
  - Update `.bdoc` to `application/bdoc`
  - Update `.bmp` to `image/bmp`
  - Update `.m4a` to `audio/mp4`
  - Update `.rtf` to `application/rtf`
  - Update `.wav` to `audio/wav`
  - Update `.xml` to `application/xml`
  - Update generic extensions to `application/octet-stream`: `.deb`, `.dll`, `.dmg`, `.exe`, `.iso`, `.msi`
  - Use mime-score module to resolve extension conflicts
- deps: ms@2.1.1
  - Add `week/w` support
  - Fix negative number handling
- deps: statuses1.5.0
- perf: remove redundant `path.normalize` call

165.5 0.16.2 / 2018-02-07

- Fix incorrect end tag in default error & redirects
- deps: depd1.1.2
  - perf: remove argument reassignment
- deps: encodeurl1.0.2
  - Fix encoding `%` as last character
- deps: statuses1.4.0

165.6 0.16.1 / 2017-09-29

- Fix regression in edge-case behavior for empty `path`

165.7 0.16.0 / 2017-09-27

- Add `immutable` option
- Fix missing `</html>` in default error & redirects
- Use instance methods on steam to check for listeners
- deps: mime@1.4.1
  - Add 70 new types for file extensions
  - Set charset as "UTF-8" for .js and .json
- perf: improve path validation speed

165.8 0.15.6 / 2017-09-22

- deps: debug@2.6.9
- perf: improve `If-Match` token parsing

165.9 0.15.5 / 2017-09-20

- deps: etag1.8.1
  - perf: replace regular expression with substring
- deps: fresh@0.5.2
  - Fix handling of modified headers with invalid dates
  - perf: improve ETag match loop
  - perf: improve `If-None-Match` token parsing

165.10 0.15.4 / 2017-08-05

- deps: debug@2.6.8
- deps: depd1.1.1
  - Remove unnecessary `Buffer` loading
- deps: http-errors1.6.2
  - deps: depd@1.1.1

165.11 0.15.3 / 2017-05-16

- deps: debug@2.6.7
  - deps: ms@2.0.0
- deps: ms@2.0.0

165.12 0.15.2 / 2017-04-26

- deps: debug@2.6.4
  - Fix `DEBUG_MAX_ARRAY_LENGTH`
  - deps: ms@0.7.3
- deps: ms@1.0.0

165.13 0.15.1 / 2017-03-04

- Fix issue when `Date.parse` does not return `NaN` on invalid date
- Fix strict violation in broken environments

165.14 0.15.0 / 2017-02-25

- Support `If-Match` and `If-Unmodified-Since` headers
- Add `res` and `path` arguments to `directory` event
- Remove usage of `res._headers` private field
  - Improves compatibility with Node.js 8 nightly
- Send complete HTML document in redirect & error responses
- Set default CSP header in redirect & error responses
- Use `res.getHeaderNames()` when available
- Use `res.headersSent` when available
- deps: debug@2.6.1
  - Allow colors in workers
  - Deprecated `DEBUG_FD` environment variable set to `3` or higher
  - Fix error when running under React Native
  - Use same color for same namespace
  - deps: ms@0.7.2
- deps: etag1.8.0
- deps: fresh@0.5.0
  - Fix false detection of `no-cache` request directive
  - Fix incorrect result when `If-None-Match` has both `*` and ETags
  - Fix weak `ETag` matching to match spec
  - perf: delay reading header values until needed
  - perf: enable strict mode
  - perf: hoist regular expressions
  - perf: remove duplicate conditional
  - perf: remove unnecessary boolean coercions
  - perf: skip checking modified time if ETag check failed
  - perf: skip parsing `If-None-Match` when no `ETag` header
  - perf: use `Date.parse` instead of `new Date`
- deps: http-errors1.6.1
  - Make `message` property enumerable for `HttpError`s
  - deps: setprototypeof@1.0.3

165.15 0.14.2 / 2017-01-23

- deps: http-errors1.5.1
  - deps: inherits@2.0.3
  - deps: setprototypeof@1.0.2
  - deps: statuses@'>= 1.3.1 < 2'
- deps: ms@0.7.2
- deps: statuses1.3.1

165.16 0.14.1 / 2016-06-09

- Fix redirect error when `path` contains raw non-URL characters
- Fix redirect when `path` starts with multiple forward slashes

165.17 0.14.0 / 2016-06-06

- Add `acceptRanges` option
- Add `cacheControl` option
- Attempt to combine multiple ranges into single range
- Correctly inherit from `Stream` class
- Fix `Content-Range` header in 416 responses when using `start/end` options
- Fix `Content-Range` header missing from default 416 responses
- Ignore non-byte `Range` headers
- deps: http-errors1.5.0
  - Add `HttpError` export, for `err instanceof createError.HttpError`
  - Support new code `421 Misdirected Request`
  - Use `setprototypeof` module to replace __`proto`__ setting
  - deps: inherits@2.0.1
  - deps: statuses@'>= 1.3.0 < 2'
  - perf: enable strict mode
- deps: range-parser1.2.0
  - Fix incorrectly returning -1 when there is at least one valid range
  - perf: remove internal function
- deps: statuses1.3.0
  - Add `421 Misdirected Request`
  - perf: enable strict mode
- perf: remove argument reassignment

165.18 0.13.2 / 2016-03-05

- Fix invalid `Content-Type` header when `send.mime.default_type` unset

165.19 0.13.1 / 2016-01-16

- deps: depd1.1.0
  - Support web browser loading
  - perf: enable strict mode
- deps: destroy1.0.4
  - perf: enable strict mode
- deps: escape-html1.0.3
  - perf: enable strict mode
  - perf: optimize string replacement
  - perf: use faster string coercion
- deps: range-parser1.0.3
  - perf: enable strict mode

165.20 0.13.0 / 2015-06-16

- Allow Node.js HTTP server to set `Date` response header
- Fix incorrectly removing `Content-Location` on 304 response
- Improve the default redirect response headers
- Send appropriate headers on default error response
- Use `http-errors` for standard emitted errors
- Use `statuses` instead of `http` module for status messages
- deps: escape-html@1.0.2
- deps: etag1.7.0
  - Improve stat performance by removing hashing
- deps: fresh@0.3.0
  - Add weak `ETag` matching support
- deps: on-finished2.3.0
  - Add defined behavior for HTTP `CONNECT` requests
  - Add defined behavior for HTTP `Upgrade` requests
  - deps: ee-first@1.1.1
- perf: enable strict mode
- perf: remove unnecessary array allocations

165.21 0.12.3 / 2015-05-13

- deps: debug2.2.0
  - deps: ms@0.7.1
- deps: depd1.0.1
- deps: etag1.6.0
  - Improve support for JXcore
  - Support "fake" stats objects in environments without `fs`
- deps: ms@0.7.1
  - Prevent extraordinarily long inputs
- deps: on-finished2.2.1

165.22 0.12.2 / 2015-03-13

- Throw errors early for invalid `extensions` or `index` options
- deps: debug2.1.3
  - Fix high intensity foreground color for bold
  - deps: ms@0.7.0

165.23 0.12.1 / 2015-02-17

- Fix regression sending zero-length files

165.24 0.12.0 / 2015-02-16

- Always read the stat size from the file
- Fix mutating passed-in `options`
- deps: mime@1.3.4

165.25 0.11.1 / 2015-01-20

- Fix `root` path disclosure

165.26 0.11.0 / 2015-01-05

- deps: debug2.1.1
- deps: etag1.5.1
  - deps: crc@3.2.1
- deps: ms@0.7.0
  - Add `milliseconds`
  - Add `msecs`
  - Add `secs`
  - Add `mins`
  - Add `hrs`
  - Add `yrs`
- deps: on-finished2.2.0

165.27 0.10.1 / 2014-10-22

- deps: on-finished2.1.1
  - Fix handling of pipelined requests

165.28 0.10.0 / 2014-10-15

- deps: debug2.1.0
  - Implement `DEBUG_FD` env variable support
- deps: depd1.0.0
- deps: etag1.5.0
  - Improve string performance
  - Slightly improve speed for weak ETags over 1KB

165.29 0.9.3 / 2014-09-24

- deps: etag1.4.0
  - Support "fake" stats objects

165.30 0.9.2 / 2014-09-15

- deps: depd@0.4.5
- deps: etag1.3.1
- deps: range-parser1.0.2

165.31 0.9.1 / 2014-09-07

- deps: fresh@0.2.4

165.32 0.9.0 / 2014-09-07

- Add `lastModified` option
- Use `etag` to generate `ETag` header
- deps: debug2.0.0

165.33 0.8.5 / 2014-09-04

- Fix malicious path detection for empty string path

165.34 0.8.4 / 2014-09-04

- Fix a path traversal issue when using `root`

165.35 0.8.3 / 2014-08-16

- deps: destroy@1.0.3
  - renamed from dethroy
- deps: on-finished@2.1.0

165.36 0.8.2 / 2014-08-14

- Work around `fd` leak in Node.js 0.10 for `fs.ReadStream`
- deps: dethroy@1.0.2

165.37 0.8.1 / 2014-08-05

- Fix `extensions` behavior when file already has extension

165.38 0.8.0 / 2014-08-05

- Add `extensions` option

165.39 0.7.4 / 2014-08-04

- Fix serving index files without root dir

165.40 0.7.3 / 2014-07-29

- Fix incorrect 403 on Windows and Node.js 0.11

165.41 0.7.2 / 2014-07-27

- deps: depd@0.4.4
  - Work-around v8 generating empty stack traces

165.42 0.7.1 / 2014-07-26

- deps: depd@0.4.3
  - Fix exception when global `Error.stackTraceLimit` is too low

165.43 0.7.0 / 2014-07-20

- Deprecate `hidden` option; use `dotfiles` option
- Add `dotfiles` option
- deps: debug@1.0.4
- deps: depd@0.4.2
  - Add `TRACE_DEPRECATION` environment variable
  - Remove non-standard grey color from color output
  - Support `--no-deprecation` argument
  - Support `--trace-deprecation` argument

165.44 0.6.0 / 2014-07-11

- Deprecate `from` option; use `root` option
- Deprecate `send.etag()` – use `etag` in `options`
- Deprecate `send.hidden()` – use `hidden` in `options`
- Deprecate `send.index()` – use `index` in `options`
- Deprecate `send.maxage()` – use `maxAge` in `options`
- Deprecate `send.root()` – use `root` in `options`
- Cap `maxAge` value to 1 year
- deps: debug@1.0.3
  - Add support for multiple wildcards in namespaces

165.45 0.5.0 / 2014-06-28

- Accept string for `maxAge` (converted by `ms`)
- Add `headers` event
- Include link in default redirect response
- Use `EventEmitter.listenerCount` to count listeners

165.46 0.4.3 / 2014-06-11

- Do not throw un-catchable error on file open race condition
- Use `escape-html` for HTML escaping
- deps: debug@1.0.2
  - fix some debugging output colors on node.js 0.8
- deps: finished@1.2.2
- deps: fresh@0.2.2

165.47 0.4.2 / 2014-06-09

- fix "event emitter leak" warnings
- deps: debug@1.0.1
- deps: finished@1.2.1

165.48 0.4.1 / 2014-06-02

- Send `max-age` in `Cache-Control` in correct format

165.49 0.4.0 / 2014-05-27

- Calculate ETag with md5 for reduced collisions
- Fix wrong behavior when index file matches directory
- Ignore stream errors after request ends
  - Goodbye `EBADF, read`
- Skip directories in index file search
- deps: debug@0.8.1

165.50 0.3.0 / 2014-04-24

- Fix sending files with dots without root set
- Coerce option types
- Accept API options in options object
- Set etags to "weak"
- Include file path in etag
- Make "Can't set headers after they are sent." catchable
- Send full entity-body for multi range requests
- Default directory access to 403 when index disabled
- Support multiple index paths
- Support "If-Range" header
- Control whether to generate etags
- deps: `mime@1.2.11`

165.51 0.2.0 / 2014-01-29

- update range-parser and fresh

165.52 0.1.4 / 2013-08-11

- update fresh

165.53 0.1.3 / 2013-07-08

- Revert "Fix fd leak"

165.54 0.1.2 / 2013-07-03

- Fix fd leak

165.55 0.1.0 / 2012-08-25

- add options parameter to send() that is passed to fs.createReadStream() [kanongil]

165.56 0.0.4 / 2012-08-16

- allow custom "Accept-Ranges" definition

165.57 0.0.3 / 2012-07-16

- fix normalization of the root directory. Closes #3

165.58 0.0.2 / 2012-07-09

- add passing of req explicitly for now (YUCK)

165.59 0.0.1 / 2010-01-03

- Initial release

Chapter 166

1.0.2 / 2018-01-21

- Fix encoding `%` as last character

166.1 1.0.1 / 2016-06-09

- Fix encoding unpaired surrogates at start/end of string

166.2 1.0.0 / 2016-06-08

- Initial release

Chapter 167

encodeurl

Encode a URL to a percent-encoded form, excluding already-encoded sequences

167.1 Installation

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install encodeurl
```

167.2 API

```
var encodeUrl = require('encodeurl')
```

167.2.1 encodeUrl(url)

Encode a URL to a percent-encoded form, excluding already-encoded sequences.
This function will take an already-encoded URL and encode all the non-URL code points (as UTF-8 byte sequences).
This function will not encode the "%" character unless it is not part of a valid sequence (`%20` will be left as-is, but `foo` will be encoded as `%25foo`).
This encode is meant to be "safe" and does not throw errors. It will try as hard as it can to properly encode the given URL, including replacing any raw, unpaired surrogate pairs with the Unicode replacement character prior to encoding.
This function is *similar* to the intrinsic function `encodeURI`, except it will not encode the `%` character if that is part of a valid sequence, will not encode `[` and `]` (for IPv6 hostnames) and will replace raw, unpaired surrogate pairs with the Unicode replacement character (instead of throwing).

167.3 Examples

167.3.1 Encode a URL containing user-controled data

```
var encodeUrl = require('encodeurl')
var escapeHtml = require('escape-html')
http.createServer(function onRequest (req, res) {
  // get encoded form of inbound url
  var url = encodeUrl(req.url)
  // create html message
  var body = '<p>Location ' + escapeHtml(url) + ' not found</p>'
  // send a 404
  res.statusCode = 404
  res.setHeader('Content-Type', 'text/html; charset=UTF-8')
  res.setHeader('Content-Length', String(Buffer.byteLength(body, 'utf-8')))
  res.end(body, 'utf-8')
})
```

167.3.2 Encode a URL for use in a header field

```
var encodeUrl = require('encodeurl')
var escapeHtml = require('escape-html')
var url = require('url')
http.createServer(function onRequest (req, res) {
  // parse inbound url
  var href = url.parse(req)
  // set new host for redirect
  href.host = 'localhost'
  href.protocol = 'https:'
  href.slashes = true
  // create location header
  var location = encodeUrl(url.format(href))
  // create html message
  var body = '<p>Redirecting to new site:  '  + escapeHtml(location) + '</p>'
  // send a 301
  res.statusCode = 301
  res.setHeader('Content-Type', 'text/html; charset=UTF-8')
  res.setHeader('Content-Length', String(Buffer.byteLength(body, 'utf-8')))
  res.setHeader('Location', location)
  res.end(body, 'utf-8')
})
```

167.4 Testing

```
$ npm test
$ npm run lint
```

167.5 References

- RFC 3986: Uniform Resource Identifier (URI): Generic Syntax
- WHATWG URL Living Standard

167.6 License

[MIT](LICENSE)

Chapter 168

license

Chapter 169

ms

Use this package to easily convert various time formats to milliseconds.

169.1 Examples

```
ms('2 days')  // 172800000
ms('1d')      // 86400000
ms('10h')     // 36000000
ms('2.5 hrs') // 9000000
ms('2h')      // 7200000
ms('1m')      // 60000
ms('5s')      // 5000
ms('1y')      // 31557600000
ms('100')     // 100
ms('-3 days') // -259200000
ms('-1h')     // -3600000
ms('-200')    // -200
```

169.1.1 Convert from Milliseconds

```
ms(60000)             // "1m"
ms(2 * 60000)         // "2m"
ms(-3 * 60000)        // "-3m"
ms(ms('10 hours'))    // "10h"
```

169.1.2 Time Format Written-Out

```
ms(60000, { long: true })             // "1 minute"
ms(2 * 60000, { long: true })         // "2 minutes"
ms(-3 * 60000, { long: true })        // "-3 minutes"
ms(ms('10 hours'), { long: true })    // "10 hours"
```

169.2 Features

- Works both in `Node.js` and in the browser
- If a number is supplied to `ms`, a string with a unit is returned
- If a string that contains the number is supplied, it returns it as a number (e.g.: it returns `100` for "100")
- If you pass a string with a number and a valid unit, the number of equivalent milliseconds is returned

169.3 Related Packages

- `ms.macro` - Run `ms` as a macro at build-time.

169.4 Caught a Bug?

1. `Fork` this repository to your own GitHub account and then `clone` it to your local device

2. Link the package to the global module directory: `npm link`
3. Within the module you want to test your local development instance of ms, just link it to the dependencies: `npm link ms`. Instead of the default one from npm, Node.js will now use your clone of ms!

As always, you can run the tests using: `npm test`

Chapter 170

send

Send is a library for streaming files from the file system as a http response supporting partial responses (Ranges), conditional-GET negotiation (If-Match, If-Unmodified-Since, If-None-Match, If-Modified-Since), high test coverage, and granular events which may be leveraged to take appropriate actions in your application or framework.
Looking to serve up entire folders mapped to URLs? Try `serve-static`.

170.1 Installation

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install send
```

170.2 API

```
var send = require('send')
```

170.2.1 send(req, path, [options])

Create a new `SendStream` for the given path to send to a `res`. The `req` is the Node.js HTTP request and the `path` is a urlencoded path to send (urlencoded, not the actual file-system path).

170.2.1.1 Options

170.2.1.1.1 acceptRanges Enable or disable accepting ranged requests, defaults to true. Disabling this will not send `Accept-Ranges` and ignore the contents of the `Range` request header.

170.2.1.1.2 cacheControl Enable or disable setting `Cache-Control` response header, defaults to true. Disabling this will ignore the `immutable` and `maxAge` options.

170.2.1.1.3 dotfiles Set how "dotfiles" are treated when encountered. A dotfile is a file or directory that begins with a dot ("."). Note this check is done on the path itself without checking if the path actually exists on the disk. If `root` is specified, only the dotfiles above the root are checked (i.e. the root itself can be within a dotfile when when set to "deny").

- "allow'`No special treatment for dotfiles. -`'deny'`Send a 403 for any request for a dotfile. -`'ignore"` Pretend like the dotfile does not exist and 404.

The default value is *similar* to "ignore"`, with the exception that this default will not ignore the files within a directory that begins with a dot, for backward-compatibility.

170.2.1.1.4 end Byte offset at which the stream ends, defaults to the length of the file minus 1. The end is inclusive in the stream, meaning `end: 3` will include the 4th byte in the stream.

170.2.1.1.5 etag Enable or disable etag generation, defaults to true.

170.2.1.1.6 extensions If a given file doesn't exist, try appending one of the given extensions, in the given order. By default, this is disabled (set to `false`). An example value that will serve extension-less HTML files: '['html', 'htm']`. This is skipped if the requested file already has an extension.

170.2.1.1.7 immutable Enable or disable the `immutable` directive in the `Cache-Control` response header, defaults to `false`. If set to `true`, the `maxAge` option should also be specified to enable caching. The `immutable` directive will prevent supported clients from making conditional requests during the life of the `maxAge` option to check if the file has changed.

170.2.1.1.8 index By default send supports "index.html" files, to disable this set `false` or to supply a new index pass a string or an array in preferred order.

170.2.1.1.9 lastModified Enable or disable `Last-Modified` header, defaults to true. Uses the file system's last modified value.

170.2.1.1.10 maxAge Provide a max-age in milliseconds for http caching, defaults to 0. This can also be a string accepted by the `ms` module.

170.2.1.1.11 root Serve files relative to `path`.

170.2.1.1.12 start Byte offset at which the stream starts, defaults to 0. The start is inclusive, meaning `start: 2` will include the 3rd byte in the stream.

170.2.1.2 Events

The `SendStream` is an event emitter and will emit the following events:

- `error` an error occurred `(err)`
- `directory` a directory was requested `(res, path)`
- `file` a file was requested `(path, stat)`
- `headers` the headers are about to be set on a file `(res, path, stat)`
- `stream` file streaming has started `(stream)`
- `end` streaming has completed

170.2.1.3 .pipe

The `pipe` method is used to pipe the response into the Node.js HTTP response object, typically `send(req, path, options).pipe(res)`.

170.2.2 .mime

The `mime` export is the global instance of of the `mime npm module`.
This is used to configure the MIME types that are associated with file extensions as well as other options for how to resolve the MIME type of a file (like the default type to use for an unknown file extension).

170.3 Error-handling

By default when no `error` listeners are present an automatic response will be made, otherwise you have full control over the response, aka you may show a 5xx page etc.

170.4 Caching

It does *not* perform internal caching, you should use a reverse proxy cache such as Varnish for this, or those fancy things called CDNs. If your application is small enough that it would benefit from single-node memory caching, it's small enough that it does not need caching at all ;).

170.5 Debugging

To enable `debug()` instrumentation output export **DEBUG**:

```
$ DEBUG=send node app
```

170.6 Running tests

```
$ npm install
$ npm test
```

170.7 Examples

170.7.1 Serve a specific file

This simple example will send a specific file to all requests.

```
var http = require('http')
var send = require('send')
var server = http.createServer(function onRequest (req, res) {
  send(req, '/path/to/index.html')
    .pipe(res)
})
server.listen(3000)
```

170.7.2 Serve all files from a directory

This simple example will just serve up all the files in a given directory as the top-level. For example, a request `GET /foo.txt` will send back `/www/public/foo.txt`.

```
var http = require('http')
var parseUrl = require('parseurl')
var send = require('send')
var server = http.createServer(function onRequest (req, res) {
  send(req, parseUrl(req).pathname, { root: '/www/public' })
    .pipe(res)
})
server.listen(3000)
```

170.7.3 Custom file types

```
var http = require('http')
var parseUrl = require('parseurl')
var send = require('send')
// Default unknown types to text/plain
send.mime.default_type = 'text/plain'
// Add a custom type
send.mime.define({
  'application/x-my-type': ['x-mt', 'x-mtt']
})
var server = http.createServer(function onRequest (req, res) {
  send(req, parseUrl(req).pathname, { root: '/www/public' })
    .pipe(res)
})
server.listen(3000)
```

170.7.4 Custom directory index view

This is a example of serving up a structure of directories with a custom function to render a listing of a directory.

```
var http = require('http')
var fs = require('fs')
var parseUrl = require('parseurl')
var send = require('send')
// Transfer arbitrary files from within /www/example.com/public/*
// with a custom handler for directory listing
var server = http.createServer(function onRequest (req, res) {
  send(req, parseUrl(req).pathname, { index: false, root: '/www/public' })
```

```
    .once('directory', directory)
    .pipe(res)
})
server.listen(3000)
// Custom directory handler
function directory (res, path) {
  var stream = this
  // redirect to trailing slash for consistent url
  if (!stream.hasTrailingSlash()) {
    return stream.redirect(path)
  }
  // get directory list
  fs.readdir(path, function onReaddir (err, list) {
    if (err) return stream.error(err)
    // render an index for the directory
    res.setHeader('Content-Type', 'text/plain; charset=UTF-8')
    res.end(list.join('\n') + '\n')
  })
}
```

170.7.5 Serving from a root directory with custom error-handling

```
var http = require('http')
var parseUrl = require('parseurl')
var send = require('send')
var server = http.createServer(function onRequest (req, res) {
  // your custom error-handling logic:
  function error (err) {
    res.statusCode = err.status || 500
    res.end(err.message)
  }
  // your custom headers
  function headers (res, path, stat) {
    // serve all files for download
    res.setHeader('Content-Disposition', 'attachment')
  }
  // your custom directory handling logic:
  function redirect () {
    res.statusCode = 301
    res.setHeader('Location', req.url + '/')
    res.end('Redirecting to ' + req.url + '/')
  }
  // transfer arbitrary files from within
  // /www/example.com/public/*
  send(req, parseUrl(req).pathname, { root:  '/www/public' })
    .on('error', error)
    .on('directory', redirect)
    .on('headers', headers)
    .pipe(res)
})
server.listen(3000)
```

170.8 License

[MIT](LICENSE)

Chapter 171

Security Policies and Procedures

171.1 Reporting a Bug

The `send` team and community take all security bugs seriously. Thank you for improving the security of Express. We appreciate your efforts and responsible disclosure and will make every effort to acknowledge your contributions.
Report security bugs by emailing the current owner(s) of `send`. This information can be found in the npm registry using the command `npm owner ls send`. If unsure or unable to get the information from the above, open an issue in the `project issue tracker` asking for the current contact information.
To ensure the timely response to your report, please ensure that the entirety of the report is contained within the email body and not solely behind a web link or an attachment.
At least one owner will acknowledge your email within 48 hours, and will send a more detailed response within 48 hours indicating the next steps in handling your report. After the initial reply to your report, the owners will endeavor to keep you informed of the progress towards a fix and full announcement, and may ask for additional information or guidance.

Chapter 172

seq-queue - queue to keep request process in sequence

Seq-queue is simple tool to keep requests to be executed in order.
As we known, Node.js codes run in asynchronous mode and the callbacks are unordered. But sometimes we may need the requests to be processed in order. For example, in a game, a player would do some operations such as turn right and go ahead. And in the server side, we would like to process these requests one by one, not do them all at the same time.
Seq-queue takes the responsibility to make the asynchronous, unordered processing flow into serial and ordered. It's simple but not a repeated wheel.
Seq-queue is a FIFO task queue and we can push tasks as we wish, anytime(before the queue closed), anywhere(if we hold the queue instance). A task is known as a function and we can do anything in the function and just need to call `task.done()` to tell the queue current task has finished. It promises that a task in queue would not be executed util all tasks before it finished.
Seq-queue add timeout for each task execution. If a task throws an uncaught exception in its call back or a developer forgets to call `task.done()` callback, queue would be blocked and would not execute the left tasks. To avoid these situations, seq-queue set a timeout for each task. If a task timeout, queue would drop the task and notify develop by a 'timeout' event and then invoke the next task. Any `task.done()` invoked in a timeout task would be ignored.

- Tags: node.js

172.1 Installation

```
npm install seq-queue
```

172.2 Usage

```
var seqqueue = require('seq-queue');
var queue = seqqueue.createQueue(1000);
queue.push(
  function(task) {
    setTimeout(function() {
      console.log('hello ');
      task.done();
    }, 500);
  },
  function() {
    console.log('task timeout');
  },
  1000
);
queue.push(
  function(task) {
    setTimeout(function() {
      console.log('world~');
      task.done();
    }, 500);
  }
);
```

172.3 API

172.3.1 seqqueue.createQueue(timeout)

Create a new queue instance. A global timeout value in ms for the new instance can be set by `timeout` parameter or use the default timeout (3s) by no parameter.

172.3.2 queue.push(fn, ontimeout, timeout)

Add a task into the queue instance.

172.3.2.1 Arguments

- fn(task) - The function that describes the content of task and would be invoke by queue. `fn` takes a arguemnt task and we *must* call task.done() to tell queue current task has finished.
- ontimeout() - Callback for task timeout.
- timeout - Timeout in ms for `fn`. If specified, it would overwrite the global timeout that set by `createQueue` for `fn`.

172.3.3 queue.close(force)

Close the queue. A closed queue would stop receiving new task immediately. And the left tasks would be treated in different ways decided by `force`.

172.3.3.1 Arguments

- force - If true, queue would stop working immediately and ignore any tasks left in queue. Otherwise queue would execute the tasks in queue and then stop.

172.4 Event

Seq-queue instances extend the EventEmitter and would emit events in their life cycles.

172.4.1 'timeout'(totask)

If current task not invoke task.done() within the timeout ms, a timeout event would be emit. totask.fn and totask.↩
timeout is the `fn` and `timeout` arguments that passed by `queue.push(2)`.

172.4.2 'error'(err, task)

If the task function (not callbacks) throws an uncaught error, queue would emit an error event and passes the err and task informations by event callback arguments.

172.4.3 'closed'

Emit when the close(false) is invoked.

172.4.4 'drained'

Emit when close(true) is invoked or all tasks left have finished in closed status.

Chapter 173

1.16.2 / 2024-09-11

- deps: encodeurl2.0.0

173.1 1.16.1 / 2024-09-11

- deps: `send@0.19`.0

173.2 1.16.0 / 2024-09-10

- Remove link renderization in html while redirecting

173.3 1.15.0 / 2022-03-24

- deps: `send@0.18`.0
  - Fix emitted 416 error missing headers property
  - Limit the headers removed for 304 response
  - deps: depd@2.0.0
  - deps: destroy@1.2.0
  - deps: http-errors@2.0.0
  - deps: on-finished@2.4.1
  - deps: statuses@2.0.1

173.4 1.14.2 / 2021-12-15

- deps: `send@0.17`.2
  - deps: http-errors@1.8.1
  - deps: ms@2.1.3
  - pref: ignore empty http tokens

173.5 1.14.1 / 2019-05-10

- Set stricter CSP header in redirect response
- deps: `send@0.17`.1
  - deps: range-parser1.2.1

173.6 1.14.0 / 2019-05-07

- deps: parseurl1.3.3
- deps: `send@0.17`.0
  - deps: http-errors1.7.2
  - deps: mime@1.6.0
  - deps: ms@2.1.1
  - deps: statuses1.5.0
  - perf: remove redundant `path.normalize` call

173.7 1.13.2 / 2018-02-07

- Fix incorrect end tag in redirects
- deps: encodeurl1.0.2
  - Fix encoding `%` as last character
- deps: `send@0.16`.2
  - deps: depd1.1.2
  - deps: encodeurl1.0.2
  - deps: statuses1.4.0

173.8 1.13.1 / 2017-09-29

- Fix regression when `root` is incorrectly set to a file
- deps: `send@0.16`.1

173.9 1.13.0 / 2017-09-27

- deps: `send@0.16`.0
  - Add 70 new types for file extensions
  - Add `immutable` option
  - Fix missing `</html>` in default error & redirects
  - Set charset as "UTF-8" for .js and .json
  - Use instance methods on steam to check for listeners
  - deps: mime@1.4.1
  - perf: improve path validation speed

173.10 1.12.6 / 2017-09-22

- deps: `send@0.15`.6
  - deps: debug@2.6.9
  - perf: improve `If-Match` token parsing
- perf: improve slash collapsing

173.11 1.12.5 / 2017-09-21

- deps: parseurl1.3.2
  - perf: reduce overhead for full URLs
  - perf: unroll the "fast-path" `RegExp`
- deps: `send@0.15`.5
  - Fix handling of modified headers with invalid dates
  - deps: etag1.8.1
  - deps: fresh@0.5.2

173.12 1.12.4 / 2017-08-05

- deps: `send@0.15`.4
  - deps: debug@2.6.8
  - deps: depd1.1.1
  - deps: http-errors1.6.2

173.13 1.12.3 / 2017-05-16

- deps: `send@0.15`.3
  - deps: debug@2.6.7

173.14 1.12.2 / 2017-04-26

- deps: `send@0.15`.2
  - deps: debug@2.6.4

173.15 1.12.1 / 2017-03-04

- deps: `send@0.15`.1
  - Fix issue when `Date.parse` does not return `NaN` on invalid date
  - Fix strict violation in broken environments

173.16 1.12.0 / 2017-02-25

- Send complete HTML document in redirect response
- Set default CSP header in redirect response
- deps: `send@0.15`.0
  - Fix false detection of `no-cache` request directive
  - Fix incorrect result when `If-None-Match` has both ∗ and ETags
  - Fix weak `ETag` matching to match spec
  - Remove usage of `res._headers` private field
  - Support `If-Match` and `If-Unmodified-Since` headers
  - Use `res.getHeaderNames()` when available

  - Use `res.headersSent` when available
  - deps: debug@2.6.1
  - deps: etag1.8.0
  - deps: fresh@0.5.0
  - deps: http-errors1.6.1

173.17 1.11.2 / 2017-01-23

- deps: `send@0.14`.2
  - deps: http-errors1.5.1
  - deps: ms@0.7.2
  - deps: statuses1.3.1

173.18 1.11.1 / 2016-06-10

- Fix redirect error when `req.url` contains raw non-URL characters
- deps: `send@0.14`.1

173.19 1.11.0 / 2016-06-07

- Use status code 301 for redirects
- deps: `send@0.14`.0
  - Add `acceptRanges` option
  - Add `cacheControl` option
  - Attempt to combine multiple ranges into single range
  - Correctly inherit from `Stream` class
  - Fix `Content-Range` header in 416 responses when using `start/end` options
  - Fix `Content-Range` header missing from default 416 responses
  - Ignore non-byte `Range` headers
  - deps: http-errors1.5.0
  - deps: range-parser1.2.0
  - deps: statuses1.3.0
  - perf: remove argument reassignment

173.20 1.10.3 / 2016-05-30

- deps: `send@0.13`.2
  - Fix invalid `Content-Type` header when `send.mime.default_type` unset

173.21 1.10.2 / 2016-01-19

- deps: parseurl1.3.1
  - perf: enable strict mode

173.22 1.10.1 / 2016-01-16

- deps: escape-html1.0.3
  - perf: enable strict mode
  - perf: optimize string replacement
  - perf: use faster string coercion
- deps: `send@0.13`.1
  - deps: depd1.1.0
  - deps: destroy1.0.4
  - deps: escape-html1.0.3
  - deps: range-parser1.0.3

173.23 1.10.0 / 2015-06-17

- Add `fallthrough` option
  - Allows declaring this middleware is the final destination
  - Provides better integration with Express patterns
- Fix reading options from options prototype
- Improve the default redirect response headers
- deps: escape-html@1.0.2
- deps: `send@0.13`.0
  - Allow Node.js HTTP server to set `Date` response header
  - Fix incorrectly removing `Content-Location` on 304 response
  - Improve the default redirect response headers
  - Send appropriate headers on default error response
  - Use `http-errors` for standard emitted errors
  - Use `statuses` instead of `http` module for status messages
  - deps: escape-html@1.0.2
  - deps: etag1.7.0
  - deps: fresh@0.3.0
  - deps: on-finished2.3.0
  - perf: enable strict mode
  - perf: remove unnecessary array allocations
- perf: enable strict mode
- perf: remove argument reassignment

173.24 1.9.3 / 2015-05-14

- deps: `send@0.12`.3
  - deps: debug2.2.0
  - deps: depd1.0.1
  - deps: etag1.6.0
  - deps: ms@0.7.1
  - deps: on-finished2.2.1

173.25 1.9.2 / 2015-03-14

- deps: `send@0.12`.2
  - Throw errors early for invalid `extensions` or `index` options
  - deps: debug2.1.3

173.26 1.9.1 / 2015-02-17

- deps: `send@0.12`.1
  - Fix regression sending zero-length files

173.27 1.9.0 / 2015-02-16

- deps: `send@0.12`.0
  - Always read the stat size from the file
  - Fix mutating passed-in `options`
  - deps: mime@1.3.4

173.28 1.8.1 / 2015-01-20

- Fix redirect loop in Node.js 0.11.14
- deps: `send@0.11`.1
  - Fix root path disclosure

173.29 1.8.0 / 2015-01-05

- deps: `send@0.11`.0
  - deps: debug2.1.1
  - deps: etag1.5.1
  - deps: ms@0.7.0
  - deps: on-finished2.2.0

173.30 1.7.2 / 2015-01-02

- Fix potential open redirect when mounted at root

173.31 1.7.1 / 2014-10-22

- deps: `send@0.10`.1
  - deps: on-finished2.1.1

173.32 1.7.0 / 2014-10-15

- deps: `send@0.10`.0
  - deps: debug2.1.0
  - deps: depd1.0.0
  - deps: etag1.5.0

173.33 1.6.5 / 2015-02-04

- Fix potential open redirect when mounted at root
  - Back-ported from v1.7.2

173.34 1.6.4 / 2014-10-08

- Fix redirect loop when index file serving disabled

173.35 1.6.3 / 2014-09-24

- deps: send@0.9.3
  - deps: etag1.4.0

173.36 1.6.2 / 2014-09-15

- deps: send@0.9.2
  - deps: depd@0.4.5
  - deps: etag1.3.1
  - deps: range-parser1.0.2

173.37 1.6.1 / 2014-09-07

- deps: send@0.9.1
  - deps: fresh@0.2.4

173.38 1.6.0 / 2014-09-07

- deps: send@0.9.0
  - Add `lastModified` option
  - Use `etag` to generate `ETag` header
  - deps: debug2.0.0

173.39 1.5.4 / 2014-09-04

- deps: send@0.8.5
  - Fix a path traversal issue when using `root`
  - Fix malicious path detection for empty string path

173.40 1.5.3 / 2014-08-17

- deps: send@0.8.3

173.41 1.5.2 / 2014-08-14

- deps: send@0.8.2
  - Work around `fd` leak in Node.js 0.10 for `fs.ReadStream`

173.42 1.5.1 / 2014-08-09

- Fix parsing of weird `req.originalUrl` values
- deps: parseurl1.3.0
- deps: utils-merge@1.0.0

173.43 1.5.0 / 2014-08-05

- deps: send@0.8.1
  - Add `extensions` option

173.44 1.4.4 / 2014-08-04

- deps: send@0.7.4
  - Fix serving index files without root dir

173.45 1.4.3 / 2014-07-29

- deps: send@0.7.3
  - Fix incorrect 403 on Windows and Node.js 0.11

173.46 1.4.2 / 2014-07-27

- deps: send@0.7.2
  - deps: depd@0.4.4

173.47 1.4.1 / 2014-07-26

- deps: send@0.7.1
  - deps: depd@0.4.3

173.48 1.4.0 / 2014-07-21

- deps: parseurl1.2.0
  - Cache URLs based on original value
  - Remove no-longer-needed URL mis-parse work-around
  - Simplify the "fast-path" `RegExp`
- deps: send@0.7.0
  - Add `dotfiles` option
  - deps: debug@1.0.4
  - deps: depd@0.4.2

173.49 1.3.2 / 2014-07-11

- deps: send@0.6.0
  - Cap `maxAge` value to 1 year
  - deps: debug@1.0.3

173.50 1.3.1 / 2014-07-09

- deps: parseurl1.1.3
  - faster parsing of href-only URLs

173.51 1.3.0 / 2014-06-28

- Add `setHeaders` option
- Include HTML link in redirect response
- deps: send@0.5.0
  - Accept string for `maxAge` (converted by `ms`)

173.52 1.2.3 / 2014-06-11

- deps: send@0.4.3
  - Do not throw un-catchable error on file open race condition
  - Use `escape-html` for HTML escaping
  - deps: debug@1.0.2
  - deps: finished@1.2.2
  - deps: fresh@0.2.2

173.53 1.2.2 / 2014-06-09

- deps: send@0.4.2
  - fix "event emitter leak" warnings
  - deps: debug@1.0.1
  - deps: finished@1.2.1

173.54 1.2.1 / 2014-06-02

- use `escape-html` for escaping
- deps: send@0.4.1
  - Send `max-age` in `Cache-Control` in correct format

173.55 1.2.0 / 2014-05-29

- deps: send@0.4.0
  - Calculate ETag with md5 for reduced collisions
  - Fix wrong behavior when index file matches directory
  - Ignore stream errors after request ends
  - Skip directories in index file search
  - deps: debug@0.8.1

173.56 1.1.0 / 2014-04-24

- Accept options directly to `send` module
- deps: send@0.3.0

173.57 1.0.4 / 2014-04-07

- Resolve relative paths at middleware setup
- Use parseurl to parse the URL from request

173.58 1.0.3 / 2014-03-20

- Do not rely on connect-like environments

173.59 1.0.2 / 2014-03-06

- deps: send@0.2.0

173.60 1.0.1 / 2014-03-05

- Add mime export for back-compat

173.61 1.0.0 / 2014-03-05

- Genesis from `connect`

Chapter 174

serve-static

174.1 Install

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install serve-static
```

174.2 API

```
var serveStatic = require('serve-static')
```

174.2.1 serveStatic(root, options)

Create a new middleware function to serve files from within a given root directory. The file to serve will be determined by combining `req.url` with the provided root directory. When a file is not found, instead of sending a 404 response, this module will instead call `next()` to move on to the next middleware, allowing for stacking and fall-backs.

174.2.1.1 Options

174.2.1.1.1 acceptRanges Enable or disable accepting ranged requests, defaults to true. Disabling this will not send `Accept-Ranges` and ignore the contents of the `Range` request header.

174.2.1.1.2 cacheControl Enable or disable setting `Cache-Control` response header, defaults to true. Disabling this will ignore the `immutable` and `maxAge` options.

174.2.1.1.3 dotfiles Set how "dotfiles" are treated when encountered. A dotfile is a file or directory that begins with a dot ("."). Note this check is done on the path itself without checking if the path actually exists on the disk. If `root` is specified, only the dotfiles above the root are checked (i.e. the root itself can be within a dotfile when set to "deny").

- ''allow'`No special treatment for dotfiles. -`'deny'`Deny a request for a dotfile and 403/`next()`. -`'ignore'`Pretend like the dotfile does not exist and 404/`next()`.

The default value is similar to ''ignore'`, with the exception that this default will not ignore the files within a directory that begins with a dot.

174.2.1.1.4 etag Enable or disable etag generation, defaults to true.

174.2.1.1.5 extensions Set file extension fallbacks. When set, if a file is not found, the given extensions will be added to the file name and search for. The first that exists will be served. Example: '['html', 'htm']`.
The default value is `false`.

174.2.1.1.6 fallthrough Set the middleware to have client errors fall-through as just unhandled requests, otherwise forward a client error. The difference is that client errors like a bad request or a request to a non-existent file will cause this middleware to simply `next()` to your next middleware when this value is `true`. When this value is `false`, these errors (even 404s), will invoke `next(err)`.
Typically `true` is desired such that multiple physical directories can be mapped to the same web address or for routes to fill in non-existent files.
The value `false` can be used if this middleware is mounted at a path that is designed to be strictly a single file system directory, which allows for short-circuiting 404s for less overhead. This middleware will also reply to all methods.
The default value is `true`.

174.2.1.1.7 immutable Enable or disable the `immutable` directive in the `Cache-Control` response header, defaults to `false`. If set to `true`, the `maxAge` option should also be specified to enable caching. The `immutable` directive will prevent supported clients from making conditional requests during the life of the `maxAge` option to check if the file has changed.

174.2.1.1.8 index By default this module will send "index.html" files in response to a request on a directory. To disable this set `false` or to supply a new index pass a string or an array in preferred order.

174.2.1.1.9 lastModified Enable or disable `Last-Modified` header, defaults to true. Uses the file system's last modified value.

174.2.1.1.10 maxAge Provide a max-age in milliseconds for http caching, defaults to 0. This can also be a string accepted by the `ms` module.

174.2.1.1.11 redirect Redirect to trailing "/" when the pathname is a dir. Defaults to `true`.

174.2.1.1.12 setHeaders Function to set custom headers on response. Alterations to the headers need to occur synchronously. The function is called as `fn(res, path, stat)`, where the arguments are:

- `res` the response object
- `path` the file path that is being sent
- `stat` the stat object of the file that is being sent

174.3 Examples

174.3.1 Serve files with vanilla node.js http server

```
var finalhandler = require('finalhandler')
var http = require('http')
var serveStatic = require('serve-static')
// Serve up public/ftp folder
var serve = serveStatic('public/ftp', { index:  ['index.html', 'index.htm'] })
// Create server
var server = http.createServer(function onRequest (req, res) {
  serve(req, res, finalhandler(req, res))
})
// Listen
server.listen(3000)
```

174.3.2 Serve all files as downloads

```
var contentDisposition = require('content-disposition')
var finalhandler = require('finalhandler')
var http = require('http')
var serveStatic = require('serve-static')
// Serve up public/ftp folder
var serve = serveStatic('public/ftp', {
  index:  false,
  setHeaders:  setHeaders
})
// Set header to force download
```

```
function setHeaders (res, path) {
  res.setHeader('Content-Disposition', contentDisposition(path))
}
// Create server
var server = http.createServer(function onRequest (req, res) {
  serve(req, res, finalhandler(req, res))
})
// Listen
server.listen(3000)
```

174.3.3 Serving using express

174.3.3.1 Simple

This is a simple example of using Express.

```
var express = require('express')
var serveStatic = require('serve-static')
var app = express()
app.use(serveStatic('public/ftp', { index:  ['default.html', 'default.htm'] }))
app.listen(3000)
```

174.3.3.2 Multiple roots

This example shows a simple way to search through multiple directories. Files are searched for in `public-optimized/` first, then `public/` second as a fallback.

```
var express = require('express')
var path = require('path')
var serveStatic = require('serve-static')
var app = express()
app.use(serveStatic(path.join(__dirname, 'public-optimized')))
app.use(serveStatic(path.join(__dirname, 'public')))
app.listen(3000)
```

174.3.3.3 Different settings for paths

This example shows how to set a different max age depending on the served file type. In this example, HTML files are not cached, while everything else is for 1 day.

```
var express = require('express')
var path = require('path')
var serveStatic = require('serve-static')
var app = express()
app.use(serveStatic(path.join(__dirname, 'public'), {
  maxAge:  '1d',
  setHeaders:  setCustomCacheControl
}))
app.listen(3000)
function setCustomCacheControl (res, path) {
  if (serveStatic.mime.lookup(path) === 'text/html') {
    // Custom Cache-Control for HTML files
    res.setHeader('Cache-Control', 'public, max-age=0')
  }
}
```

174.4 License

[MIT](LICENSE)

Chapter 175

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

175.1 <a href="https://github.com/ljharb/set-function-length/compare/v1.2.1...v1.2.2" >v1.2.2</a> - 2024-03-09

175.1.1 Commits

- [types] use shared config `027032f`
- [actions] remove redundant finisher; use reusable workflow `1fd4fb1`
- [types] use a handwritten d.ts file instead of emit `01b9761`
- [Deps] update `define-data-property`, `get-intrinsic`, `has-property-descriptors` `bee8eaf`
- [Dev Deps] update `call-bind`, `tape` `5dae579`
- [Tests] use `@arethetypeswrong/cli` `7e22425`

175.2 <a href="https://github.com/ljharb/set-function-length/compare/v1.2.0...v1.2.1" >v1.2.1</a> - 2024-02-06

175.2.1 Commits

- [Dev Deps] update `call-bind`, `tape`, `typescript` `d9a4601`
- [Deps] update `define-data-property`, `get-intrinsic` `38d39ae`
- [Refactor] use `es-errors`, so things that only need those do not need `get-intrinsic` `b4bfe5a`

175.3 <a href="https://github.com/ljharb/set-function-length/compare/v1.1.1...v1.2.0" >v1.2.0</a> - 2024-01-14

175.3.1 Commits

- [New] add types `f6d9088`
- [Fix] ensure `env` properties are always booleans `0c42f84`
- [Dev Deps] update `aud`, `call-bind`, `npmignore`, `tape` `2b75f75`
- [Deps] update `get-intrinsic`, `has-property-descriptors` `19bf0fc`
- [meta] add `sideEffects` flag `8bb9b78`

175.4 <a href="https://github.com/ljharb/set-function-length/compare/v1.1.0...v1.1.1" >v1.1.1</a> - 2023-10-19

175.4.1 Fixed

- [Fix] move `define-data-property` to runtime deps #2

175.4.2 Commits

- [Dev Deps] update `object-inspect`; add missing `call-bind` 5aecf79

175.5 <a href="https://github.com/ljharb/set-function-length/compare/v1.0.1...v1.1.0" >v1.1.0</a> - 2023-10-13

175.5.1 Commits

- [New] add `env` entry point 475c87a
- [Tests] add coverage with `nyc` 14f0bf8
- [eslint] fix linting failure fb516f9
- [Deps] update `define-data-property` d727e7c

175.6 <a href="https://github.com/ljharb/set-function-length/compare/v1.0.0...v1.0.1" >v1.0.1</a> - 2023-10-12

175.6.1 Commits

- [Refactor] use `get-intrinsic`, since it‘s in the dep graph anyways 278a954
- [meta] add `exports` 72acfe5

175.7 v1.0.0 - 2023-10-12

175.7.1 Commits

- Initial implementation, tests, readme fce14e1
- Initial commit ca7ba85
- npm init 6a7e493
- Only apps should have lockfiles d2bf6c4

Chapter 176

set-function-length <sup><a href="https://npmjs.org/package/set-function-length" ><img src="https://versionbadg.es/ljharb/set-function-length.svg" alt="Version Badge"/></a></sup>

[][license-url]

Set a function's length.
Arguments:

- `fn`: the function
- `length`: the new length. Must be an integer between 0 and 2**32.
- `loose`: Optional. If true, and the length fails to be set, do not throw. Default false.

Returns `fn`.

176.1 Usage

```
var setFunctionLength = require('set-function-length');
var assert = require('assert');
function zero() {}
function one(_) {}
function two(_, __) {}
assert.equal(zero.length, 0);
assert.equal(one.length, 1);
assert.equal(two.length, 2);
assert.equal(setFunctionLength(zero, 10), zero);
assert.equal(setFunctionLength(one, 11), one);
assert.equal(setFunctionLength(two, 12), two);
assert.equal(zero.length, 10);
assert.equal(one.length, 11);
assert.equal(two.length, 12);
```

Chapter 177

Polyfill for <tt>Object.setPrototypeOf</tt>

A simple cross platform implementation to set the prototype of an instianted object. Supports all modern browsers and at least back to IE8.

177.1 Usage:

```
$ npm install --save setprototypeof
var setPrototypeOf = require('setprototypeof')
var obj = {}
setPrototypeOf(obj, {
  foo:  function () {
    return 'bar'
  }
})
obj.foo() // bar
```

TypeScript is also supported:

```
import setPrototypeOf from 'setprototypeof'
```

Chapter 178

Changelog

All notable changes to this project will be documented in this file.
The format is based on `Keep a Changelog` and this project adheres to `Semantic Versioning`.

178.1 <a href="https://github.com/ljharb/side-channel/compare/v1.0.↩ 5...v1.0.6" >v1.0.6</a> - 2024-02-29

178.1.1 Commits

- add types `9beef66`
- [meta] simplify `exports` `4334cf9`
- [Deps] update `call-bind` `d6043c4`
- [Dev Deps] update `tape` `6aca376`

178.2 <a href="https://github.com/ljharb/side-channel/compare/v1.0.↩ 4...v1.0.5" >v1.0.5</a> - 2024-02-06

178.2.1 Commits

- [actions] reuse common workflows `3d2e1ff`
- [meta] use `npmignore` to autogenerate an npmignore file `04296ea`
- [meta] add `.editorconfig`; add `eclint` `130f0a6`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog`, `safe-publish-latest`, `tape` `d480c2f`
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog`, `tape` `ecbe70e`
- [actions] update rebase action `75240b9`
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `npmignore`, `tape` `ae8d281`
- [Dev Deps] update `@ljharb/eslint-config`, `aud`, `tape` `7125b88`
- [Deps] update `call-bind`, `get-intrinsic`, `object-inspect` `82577c9`
- [Deps] update `call-bind`, `get-intrinsic`, `object-inspect` `550aadf`
- [Tests] increase coverage `5130877`
- [Deps] update `get-intrinsic`, `object-inspect` `ba0194c`

- [meta] add missing `engines.node` 985fd24
- [Refactor] use `es-errors`, so things that only need those do not need `get-intrinsic` 40227a8
- [Deps] update `get-intrinsic` a989b40
- [Deps] update `object-inspect` aec42d2

178.3 <a href="https://github.com/ljharb/side-channel/compare/v1.0.↩ 3...v1.0.4" >v1.0.4</a> - 2020-12-29

178.3.1 Commits

- [Tests] migrate tests to Github Actions 10909cb
- [Refactor] Use a linked list rather than an array, and move accessed nodes to the beginning 195613f
- [meta] do not publish github action workflow files 290ec29
- [Tests] run `nyc` on all tests; use `tape` runner ea6d030
- [actions] add "Allow Edits" workflow d464d8f
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog` 02daca8
- [Refactor] use `call-bind` and `get-intrinsic` instead of `es-abstract` e09d481
- [Deps] update `object.assign` ee83aa8
- [actions] update rebase action to use checkout v2 7726b0b

178.4 <a href="https://github.com/ljharb/side-channel/compare/v1.0.↩ 2...v1.0.3" >v1.0.3</a> - 2020-08-23

178.4.1 Commits

- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `aud`, `auto-changelog`, `tape` 1f10561
- [Deps] update `es-abstract`, `object-inspect` bc20159
- [Dev Deps] update `@ljharb/eslint-config`, `tape` b9b2b22
- [Dev Deps] update `eslint`, `@ljharb/eslint-config`, `tape` 7055ab4
- [Dev Deps] update `auto-changelog`; add `aud` d278c37
- [actions] switch Automatic Rebase workflow to `pull_request_target` event 3bcf982
- [Tests] only audit prod deps 18d01c4
- [Deps] update `es-abstract` 6ab096d
- [Dev Deps] update `tape` 9dc174c
- [Deps] update `es-abstract` 431d0f0
- [Deps] update `es-abstract` 49869fd
- [meta] Add package.json to package's exports 77d9cdc

178.5 <a href="https://github.com/ljharb/side-channel/compare/v1.0.↩ 1...v1.0.2" >v1.0.2</a> - 2019-12-20

178.5.1 Commits

- [Dev Deps] update `@ljharb/eslint-config`, `tape` `4a526df`
- [Deps] update `es-abstract` `d4f6e62`

178.6 <a href="https://github.com/ljharb/side-channel/compare/v1.0.↩ 0...v1.0.1" >v1.0.1</a> - 2019-12-01

178.6.1 Commits

- [Fix] add missing "exports" `d212907`

178.7 v1.0.0 - 2019-12-01

178.7.1 Commits

- Initial implementation `dbebd3a`
- Initial tests `73bdefe`
- Initial commit `43c03e1`
- npm init `5c090a7`
- [meta] add `auto-changelog` `a5c4e56`
- [actions] add automatic rebasing / merge commit blocking `bab1683`
- [meta] add `funding` field; create FUNDING.yml `63d7aea`
- [Tests] add `npm run lint` `46a5a81`
- Only apps should have lockfiles `8b16b03`
- [meta] add `safe-publish-latest` `2f098ef`

Chapter 179

side-channel

Store information about any JS value in a side channel. Uses WeakMap if available.

Chapter 180

2.3.3 / 2022-03-06

- Fix escaping `Date` objects from foreign isolates

180.1 2.3.2 / 2020-04-15

- perf: remove outdated array pattern

180.2 2.3.1 / 2018-02-24

- Fix incorrectly replacing non-placeholders in SQL

180.3 2.3.0 / 2017-10-01

- Add `.toSqlString()` escape overriding
- Add `raw` method to wrap raw strings for escape overriding
- Small performance improvement on `escapeId`

180.4 2.2.0 / 2016-11-01

- Escape invalid `Date` objects as `NULL`

180.5 2.1.0 / 2016-09-26

- Accept numbers and other value types in `escapeId`
- Run `buffer.toString()` through escaping

180.6 2.0.1 / 2016-06-06

- Fix npm package to include missing `lib/` directory

180.7 2.0.0 / 2016-06-06

- Bring repository up-to-date with `mysql` module changes
- Support Node.js 0.6.x

180.8 1.0.0 / 2014-11-09

- Support Node.js 0.8.x

180.9 0.0.1 / 2014-02-25

- Initial release

Chapter 181

sqlstring

Simple SQL escape and format for MySQL

181.1 Install

```
$ npm install sqlstring
```

181.2 Usage

```
var SqlString = require('sqlstring');
```

181.2.1 Escaping query values

Caution These methods of escaping values only works when the `NO_BACKSLASH_ESCAPES` SQL mode is disabled (which is the default state for MySQL servers).
In order to avoid SQL Injection attacks, you should always escape any user provided data before using it inside a SQL query. You can do so using the `SqlString.escape()` method:

```
var userId = 'some user provided value';
var sql    = 'SELECT * FROM users WHERE id = ' + SqlString.escape(userId);
console.log(sql); // SELECT * FROM users WHERE id = 'some user provided value'
```

Alternatively, you can use `?` characters as placeholders for values you would like to have escaped like this:

```
var userId = 1;
var sql    = SqlString.format('SELECT * FROM users WHERE id = ?', [userId]);
console.log(sql); // SELECT * FROM users WHERE id = 1
```

Multiple placeholders are mapped to values in the same order as passed. For example, in the following query `foo` equals `a`, `bar` equals `b`, `baz` equals `c`, and `id` will be `userId`:

```
var userId = 1;
var sql    = SqlString.format('UPDATE users SET foo = ?, bar = ?, baz = ?  WHERE id = ?',
  ['a', 'b', 'c', userId]);
console.log(sql); // UPDATE users SET foo = 'a', bar = 'b', baz = 'c' WHERE id = 1
```

This looks similar to prepared statements in MySQL, however it really just uses the same `SqlString.escape()` method internally.
Caution This also differs from prepared statements in that all `?` are replaced, even those contained in comments and strings.
Different value types are escaped differently, here is how:

- Numbers are left untouched
- Booleans are converted to `true` / `false`
- Date objects are converted to ''YYYY-mm-dd HH:ii:ss'`strings`
- `Buffers are converted to hex strings, e.g.`X'0fa5'`
- Strings are safely escaped
- Arrays are turned into list, e.g. '['a', 'b']`turns into`'a', 'b'

- `Nested arrays are turned into grouped lists (for bulk inserts), e.g.`[['a', 'b'], ['c', 'd']]`turns into`('a', 'b'), ('c', 'd')
- `Objects that have a`toSqlString`method will have`.toSqlString()` called and the returned value is used as the raw SQL.
- Objects are turned into ‘key = 'val'` pairs for each enumerable property on the object. If the property's value is a function, it is skipped; if the property's value is an object, toString() is called on it and the returned value is used.
- `undefined` / `null` are converted to `NULL`
- `NaN` / `Infinity` are left as-is. MySQL does not support these, and trying to insert them as values will trigger MySQL errors until they implement support.

You may have noticed that this escaping allows you to do neat things like this:

```
var post  = {id: 1, title: 'Hello MySQL'};
var sql = SqlString.format('INSERT INTO posts SET ?', post);
console.log(sql); // INSERT INTO posts SET `id` = 1, `title` = 'Hello MySQL'
```

And the `toSqlString` method allows you to form complex queries with functions:

```
var CURRENT_TIMESTAMP = { toSqlString: function() { return 'CURRENT_TIMESTAMP()'; } };
var sql = SqlString.format('UPDATE posts SET modified = ? WHERE id = ?', [CURRENT_TIMESTAMP, 42]);
console.log(sql); // UPDATE posts SET modified = CURRENT_TIMESTAMP() WHERE id = 42
```

To generate objects with a `toSqlString` method, the `SqlString.raw()` method can be used. This creates an object that will be left un-touched when using in a `?` placeholder, useful for using functions as dynamic values:
Caution The string provided to `SqlString.raw()` will skip all escaping functions when used, so be careful when passing in unvalidated input.

```
var CURRENT_TIMESTAMP = SqlString.raw('CURRENT_TIMESTAMP()');
var sql = SqlString.format('UPDATE posts SET modified = ? WHERE id = ?', [CURRENT_TIMESTAMP, 42]);
console.log(sql); // UPDATE posts SET modified = CURRENT_TIMESTAMP() WHERE id = 42
```

If you feel the need to escape queries by yourself, you can also use the escaping function directly:

```
var sql = 'SELECT * FROM posts WHERE title=' + SqlString.escape('Hello MySQL');
console.log(sql); // SELECT * FROM posts WHERE title='Hello MySQL'
```

181.2.2 Escaping query identifiers

If you can't trust an SQL identifier (database / table / column name) because it is provided by a user, you should escape it with `SqlString.escapeId(identifier)` like this:

```
var sorter = 'date';
var sql    = 'SELECT * FROM posts ORDER BY ' + SqlString.escapeId(sorter);
console.log(sql); // SELECT * FROM posts ORDER BY `date`
```

It also supports adding qualified identifiers. It will escape both parts.

```
var sorter = 'date';
var sql    = 'SELECT * FROM posts ORDER BY ' + SqlString.escapeId('posts.' + sorter);
console.log(sql); // SELECT * FROM posts ORDER BY `posts`.`date`
```

If you do not want to treat `.` as qualified identifiers, you can set the second argument to `true` in order to keep the string as a literal identifier:

```
var sorter = 'date.2';
var sql    = 'SELECT * FROM posts ORDER BY ' + SqlString.escapeId(sorter, true);
console.log(sql); // SELECT * FROM posts ORDER BY `date.2`
```

Alternatively, you can use `??` characters as placeholders for identifiers you would like to have escaped like this:

```
var userId = 1;
var columns = ['username', 'email'];
var sql     = SqlString.format('SELECT ?? FROM ?? WHERE id = ?', [columns, 'users', userId]);
console.log(sql); // SELECT `username`, `email` FROM `users` WHERE id = 1
```

Please note that this last character sequence is experimental and syntax might change
When you pass an Object to `.escape()` or `.format()`, `.escapeId()` is used to avoid SQL injection in object keys.

181.2.3 Formatting queries

You can use `SqlString.format` to prepare a query with multiple insertion points, utilizing the proper escaping for ids and values. A simple example of this follows:

```
var userId  = 1;
var inserts = ['users', 'id', userId];
var sql     = SqlString.format('SELECT * FROM ?? WHERE ?? = ?', inserts);
console.log(sql); // SELECT * FROM `users` WHERE `id` = 1
```

Following this you then have a valid, escaped query that you can then send to the database safely. This is useful if you are looking to prepare the query before actually sending it to the database. You also have the option (but are

not required) to pass in `stringifyObject` and `timeZone`, allowing you provide a custom means of turning objects into strings, as well as a location-specific/timezone-aware `Date`.
This can be further combined with the `SqlString.raw()` helper to generate SQL that includes MySQL functions as dynamic vales:

```
var userId = 1;
var data   = { email:  'foobar@example.com', modified:  SqlString.raw('NOW()') };
var sql    = SqlString.format('UPDATE ??  SET ? WHERE `id` = ?', ['users', data, userId]);
console.log(sql); // UPDATE `users` SET `email` = 'foobar@example.com', `modified` = NOW() WHERE `id` = 1
```

181.3 License

[MIT](LICENSE)

Chapter 182

2.0.1 / 2021-01-03

- Fix returning values from `Object.prototype`

182.1 2.0.0 / 2020-04-19

- Drop support for Node.js 0.6
- Fix messaging casing of `418 I'm a Teapot`
- Remove code 306
- Remove `status[code]` exports; use `status.message[code]`
- Remove `status[msg]` exports; use `status.code[msg]`
- Rename `425 Unordered Collection` to standard `425 Too Early`
- Rename `STATUS_CODES` export to `message`
- Return status message for `statuses(code)` when given code

182.2 1.5.0 / 2018-03-27

- Add `103 Early Hints`

182.3 1.4.0 / 2017-10-20

- Add `STATUS_CODES` export

182.4 1.3.1 / 2016-11-11

- Fix return type in JSDoc

182.5 1.3.0 / 2016-05-17

- Add `421 Misdirected Request`
- perf: enable strict mode

182.6 1.2.1 / 2015-02-01

- Fix message for status 451
  - `451 Unavailable For Legal Reasons`

182.7 1.2.0 / 2014-09-28

- Add `208 Already Repored`
- Add `226 IM Used`
- Add `306 (Unused)`
- Add `415 Unable For Legal Reasons`
- Add `508 Loop Detected`

182.8 1.1.1 / 2014-09-24

- Add missing 308 to `codes.json`

182.9 1.1.0 / 2014-09-21

- Add `codes.json` for universal support

182.10 1.0.4 / 2014-08-20

- Package cleanup

182.11 1.0.3 / 2014-06-08

- Add 308 to `.redirect` category

182.12 1.0.2 / 2014-03-13

- Add `.retry` category

182.13 1.0.1 / 2014-03-12

- Initial release

Chapter 183

statuses

HTTP status utility for node.
This module provides a list of status codes and messages sourced from a few different projects:

- The `IANA Status Code Registry`
- The `Node.js project`
- The `NGINX project`
- The `Apache HTTP Server project`

183.1 Installation

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install statuses
```

183.2 API

```
var status = require('statuses')
```

183.2.1 status(code)

Returns the status message string for a known HTTP status code. The code may be a number or a string. An error is thrown for an unknown status code.

```
status(403) // => 'Forbidden'
status('403') // => 'Forbidden'
status(306) // throws
```

183.2.2 status(msg)

Returns the numeric status code for a known HTTP status message. The message is case-insensitive. An error is thrown for an unknown status message.

```
status('forbidden') // => 403
status('Forbidden') // => 403
status('foo') // throws
```

183.2.3 status.codes

Returns an array of all the status codes as `Integer`s.

183.2.4 status.code[msg]

Returns the numeric status code for a known status message (in lower-case), otherwise `undefined`.

```
status['not found'] // => 404
```

183.2.5 status.empty[code]

Returns `true` if a status code expects an empty body.

```
status.empty[200] // => undefined
status.empty[204] // => true
status.empty[304] // => true
```

183.2.6 status.message[code]

Returns the string message for a known numeric status code, otherwise `undefined`. This object is the same format as the `Node.js http module http.STATUS_CODES`.

```
status.message[404] // => 'Not Found'
```

183.2.7 status.redirect[code]

Returns `true` if a status code is a valid redirect status.

```
status.redirect[200] // => undefined
status.redirect[301] // => true
```

183.2.8 status.retry[code]

Returns `true` if you should retry the rest.

```
status.retry[501] // => undefined
status.retry[503] // => true
```

183.3 License

[MIT](LICENSE)

Chapter 184

Change Log

184.0.1 v6.0.9 (2021/03/07 10:34 +00:00)

- `#256` fix(anchors): applied a fix for anchors living in seperate files with. . . (#256) (@goldsziggy)

184.0.2 v7.0.0-rc.4 (2021/03/01 16:00 +00:00)

- `15bce04` correction: v6.0.8 (@kalinchernev)

184.0.3 v6.0.8 (2021/03/01 15:20 +00:00)

- `fc3d62c` v6.0.7 (@kalinchernev)
- `#253` fix(specification): apply fix for multiple anchors (#253) (@goldsziggy)

184.0.4 v7.0.0-rc.3 (2021/02/26 14:33 +00:00)

- `8873370` Update docs (@kalinchernev)
- `#250` Update FIRST-STEPS.md (#250) (@azizkale)
- `da152ea` Update docs (@kalinchernev)

184.0.5 v6.0.6 (2021/02/16 09:01 +00:00)

- `32540d1` docs version bump (@kalinchernev)
- `12a4e6d` Update landing page readme (@kalinchernev)
- `c65d659` Update issue templates (@kalinchernev)

184.0.6 v6.0.5 (2021/02/15 15:52 +00:00)

- `05bc5a9` bump (@kalinchernev)
- `a56522f` changelog (@kalinchernev)
- `#247` feat: support cjs and update docs (#247) (@kalinchernev)

184.0.7 v6.0.3 (2021/02/15 08:40 +00:00)

- `#245` fix yaml formatting (#245) (@kalinchernev)

184.0.8 v6.0.2 (2021/01/29 11:02 +00:00)

- `87e51e9` chore: prepare v6.0.2 (@kalinchernev)
- `#241` Issue with error message when using deprecated "apis" key (#241) (@CleyFaye)

184.0.9 v6.0.1 (2021/01/07 06:23 +00:00)

- `92d51ba` update version (@kalinchernev)
- `#238` fix: yaml export format (#238) (@kalinchernev)

184.0.10 v6.0.0 (2020/12/23 09:38 +00:00)

- `7aaa3e0` Update TYPESCRIPT.md (@kalinchernev)
- `d2965db` Release 6.0 (@kalinchernev)
- `0d0be79` Update documentation (@kalinchernev)
- `6db97e2` update changelog (@kalinchernev)
- `a72a91f` prepare rc5 (@kalinchernev)

184.0.11 v6.0.0-rc.5 (2020/12/16 08:56 +00:00)

- `#235` feat: support x-webhooks (#235) (@kalinchernev)
- `348687e` Simplify docs (@kalinchernev)
- `d2b194c` Update changelog (@kalinchernev)

184.0.12 v6.0.0-rc.4 (2020/12/12 17:15 +00:00)

- `#234` Remove 'api' attr from swagger definition (#234) (@mits87)

184.0.13 v6.0.0-rc.3 (2020/11/28 15:37 +00:00)

- `7ec0825` chore: prepare v6.0.0-rc.3 (@kalinchernev)
- `d99fbd5` chore: prepare v6.0.0-rc.3 (@kalinchernev)
- `9686d62` docs: update examples (@kalinchernev)
- `c4cea4c` docs: update changelog upcoming (@kalinchernev)
- `74395f2` feat: support custom encoding in api files (@kalinchernev)
- `270c0af` documentation fixes (@kalinchernev)
- `d217725` remove github-changes because of vulnerabilities (@kalinchernev)
- `5908281` add changelog (@kalinchernev)

184.0.14 v6.0.0-rc.2 (2020/11/28 09:34 +00:00)

- `3379925` sync published version (@kalinchernev)
- `#229` Add empty array fallback to regex match (#229) (@evans)

184.0.15 v6.0.0-rc.1 (2020/11/27 06:38 +00:00)

- `2a08842` bump (@kalinchernev)
- `#227` feat: handle yaml references between separate documents (#227) (@kalinchernev)

184.0.16 v5.0.1 (2020/11/08 08:22 +00:00)

- `#224` Update README.md (#224) (@zeevo)

184.0.17 v5.0.0 (2020/10/28 15:36 +00:00)

- #220 chore: refactor helpers (#220) (@kalinchernev)

184.0.18 v4.3.1 (2020/10/22 06:10 +00:00)

- b95f784 bump version (@kalinchernev)
- bf44557 add back coverage ignore-s (@kalinchernev)
- c3cdcc5 refactor cli a bit (@kalinchernev)
- 2b7d6a8 add specs for public functions (@kalinchernev)
- 105f461 coverage for yaml malformatting (@kalinchernev)
- 1f2d9d6 increase coverage for initial error handling (@kalinchernev)
- 9058113 remove deprecated coverage ignore lines (@kalinchernev)
- 0d66d5c remove @requires (@kalinchernev)
- 068581a remove unnecessary @function annotations (@kalinchernev)
- a19d7e0 remove @module definition (@kalinchernev)
- fe97fee remove vscode settings (@kalinchernev)

184.0.19 v4.3.0 (2020/10/13 13:47 +00:00)

- #52 feat: add coffeescript support (#52) (@Aslan11)

184.0.20 v4.2.3 (2020/10/08 14:58 +00:00)

- #214 chore: use jest over mocha (#214) (@kalinchernev)

184.0.21 v4.2.2 (2020/10/08 14:49 +00:00)

- #217 make a test with patch version (#217) (@kalinchernev)

184.0.22 v4.2.1 (2020/10/08 14:44 +00:00)

- e3a670b update github workflow (@kalinchernev)
- 3017902 update docs (@kalinchernev)
- 8e9da9b remove circle ci (@kalinchernev)
- 2644db6 update readme (@kalinchernev)
- #213 Add github actions (#213) (@kalinchernev)
- 98d7cf4 Update version (@kalinchernev)

184.0.23 v4.2.0 (2020/09/25 10:47 +00:00)

- #200 add openapi jsdoc annotation with test (#200) (@Uzlopak)
- #208 Better describing errors (#208) (@allisonverdam)

184.0.24 v4.1.0 (2020/09/25 10:24 +00:00)

- f508180 Update dependencies (@kalinchernev)

184.0.25 v4.0.0 (2020/03/22 13:18 +00:00)

- 5ab0e06 Upgrade to Node Dubnium (@kalinchernev)

184.0.26 v3.7.0 (2020/03/22 13:09 +00:00)

- a9c0a24 Downgrade deps to node carbon (@kalinchernev)
- d105727 Update mocka opts (@kalinchernev)
- #191 Bump acorn from 7.1.0 to 7.1.1 (#191) (@dependabot[bot])

184.0.27 v3.6.0 (2020/03/22 12:32 +00:00)

- 598b7e7 Upgrade deps + bump to 3.6.0 (@kalinchernev)

184.0.28 v3.5.0 (2019/12/04 17:38 +00:00)

- 9b5b392 Release 3.5.0 (@kalinchernev)
- #183 feat: update commander to latest version (#183) (@jamesburns-rts)
- 591b0aa Upgrades (@kalinchernev)
- #180 Bump eslint-utils from 1.3.1 to 1.4.3 (#180) (@dependabot[bot])
- #181 Bump lodash from 4.17.11 to 4.17.15 (#181) (@dependabot[bot])

184.0.29 v3.4.0 (2019/08/08 08:59 +00:00)

- 80c350f Release 3.4.0 (@kalinchernev)
- #170 refactor(parser): expose specification builder methods (#170) (@gautier-lefebvre)

184.0.30 v3.3.0 (2019/07/08 09:05 +00:00)

- ce9ad85 Release 3.3.0 (@kalinchernev)
- #166 chore(deps): upgrade all dependencies (#166) (@kalinchernev)
- #165 Update swagger-parser to fix remote execution bug (#165) (@posquit0)
- #164 Update outdated dev dependencies (#164) (@posquit0)
- #162 fix: JSDoc Official website link (#162) (@katalonne)

184.0.31 v3.2.9 (2019/04/16 06:42 +00:00)

- ece24be Release 3.2.9 (@kalinchernev)
- #157 Updating js-yaml dependency to 3.13.1 to fix remote execution vulnerability (#157) (@lerignoux)

184.0.32 v3.2.8 (2019/03/22 06:40 +00:00)

- 098078b Release 3.2.8 (@kalinchernev)
- #156 Bump js-yaml version to fix https://www.npmjs.com/advisories/788 (#156) (@iansltx)
- #154 fix: CLI usage example (#154) (@GabrielDelepine)

184.0.33 v3.2.7 (2019/02/12 14:18 +00:00)

- 9769dfd Release 3.2.7 (@kalinchernev)
- #151 Removes apis from input Def before generation. (#151) (@spencermcw)

184.0.34 v3.2.6 (2018/11/30 16:09 +00:00)

- 3563890 Release 3.2.6 (@kalinchernev)
- #147 Adding specification configuration documentation to GETTING-STARTED.md (#147) (Mason Everett)

184.0.35 v3.2.5 (2018/11/26 18:44 +00:00)

- d0555ae Release 3.2.5 (@kalinchernev)
- #145 fix: remove es2017 specific language feature (#145) (-morn)

184.0.36 v3.2.4 (2018/11/23 10:31 +00:00)

- 714d42b Release 3.2.4 (@kalinchernev)
- #143 Return false for non-empty objects (#143) (-morn)
- #140 Update CLI usage example (#140) (@ndelvalle)
- #135 Quick spelling change (#135) (@antonjb)

184.0.37 v3.2.3 (2018/09/19 06:50 +00:00)

- 36984bf Correction (@kalinchernev)
- 2d2a6c7 Release 3.2.2 (@kalinchernev)
- e2e12fa Simplify (@kalinchernev)
- 36e2a48 Add documentation section (@kalinchernev)

184.0.38 v3.2.0 (2018/09/17 20:28 +00:00)

- d4fc538 Release 3.2.0 (@kalinchernev)
- #96 error reporting help with new "verbose" flag (#96) (@liquidg3)

184.0.39 v3.1.1 (2018/09/16 15:47 +00:00)

- fc5443e Release 3.1.1 (@kalinchernev)
- 9a3aaef correction (@kalinchernev)
- 2c116a1 Add documentation (@kalinchernev)
- aa7833d Add documentation (@kalinchernev)
- a9d9bc6 Add notes on relative paths (@kalinchernev)

184.0.40 v3.1.0 (2018/09/16 09:49 +00:00)

- 6177568 Release 3.1.0 (@kalinchernev)
- #105 Feature/yaml input definition (#105) (@ehmicky)
- 4627728 Update badge (@kalinchernev)
- #131 refactor(helpers): remove unused functions (#131) (@kalinchernev)
- #108 Allow YAML anchors (#108) (@ehmicky)

184.0.41 3.0.3 (2018/09/04 13:22 +00:00)

- 7b1d059 Release 3.0.3 (@kalinchernev)
- #130 chore(deps): upgrades (#130) (@kalinchernev)
- #127 fix npm start, add env PORT as param (#127) (@Laboratory)

184.0.42 3.0.2 (2018/08/01 11:34 +00:00)

- d676ba4 Release 3.0.2 (same as 3.0.0, but npm cache issues) (@kalinchernev)
- #122 OpenAPI support (#122) (@kalinchernev)
- #121 Upgrades (#121) (@kalinchernev)

184.0.43 v1.10.3 (2018/07/22 12:59 +00:00)

- 0da844f Release 1.10.3 (@kalinchernev)
- #115 Fix Issue #78 - Prevent paths overriding each other (#115) (@willvincent)
- #109 Do not serialize YAML anchors (#109) (@ehmicky)

184.0.44 1.10.2 (2018/07/18 18:39 +00:00)

- #120 Release 1.10.2 (#120) (@kalinchernev)
- #119 Upgrade CircleCI to 2.0 (#119) (@kalinchernev)
- #113 Update GETTING-STARTED.md (#113) (@hg-pyun)
- #110 Upgrade chokidar 1.7.0 -> 2.0.3 (#110) (@ehmicky)
- #118 Remove deprecated shield (#118) (@kalinchernev)
- #97 Fix misleading description (#97) (@danielkhan)
- #87 Fix apis name in CLI docs and error message (#87) (@sapegin)
- #88 Add missed space (#88) (@sapegin)

184.0.45 v1.9.7 (2017/07/24 17:11 +00:00)

- #86 Bump to v1.9.7 (#86) (@drGrove)
- #82 chore(deps): update dependencies and bump release (#82) (@kalinchernev)
- #79 Remove unused swagger keys from swagger output object (#79) (@dolphub)
- #77 chore(deps): update packages (#77) (@kalinchernev)
- #72 Adding support for APIs in definition file. (#72) (Jesse O'Brien)

184.0.46 v1.9.3 (2017/04/29 15:39 +00:00)

- #69 docs(readme): improve documentation (#69) (@kalinchernev)
- cd60d8a Removing Donation Link (@chdanielmueller)

184.0.47 v1.9.2 (2017/02/22 13:05 +00:00)

- 6243281 Release v.1.9.2 (@kalinchernev)
- #59 Merge pull request #59 from Surnet/fix/dependencies (@Surnet)
- b8fdd61 Moving chokidar dependency. (@kalinchernev)

184.0.48 v1.9.1 (2017/01/22 11:26 +00:00)

- 50c3a2c Release v1.9.1 (@kalinchernev)
- #55 Merge pull request #55 from Surnet/chore/npmignore (@Surnet)
- 3e9a2b2 Ignore c9 hidden folder on npm publish (@kalinchernev)
- d7deb0b Default output file is actually swagger.json (@kalinchernev)

184.0.49 v1.9.0 (2017/01/17 09:25 +00:00)

- ee44bb2 Release v1.9.0 (@kalinchernev)
- #49 Merge pull request #49 from Surnet/feature/watch-task (@Surnet)
- 6993f69 Update tests (@kalinchernev)
- 7117fd8 Update to latest master (@kalinchernev)
- #53 Merge pull request #53 from mandrean/feat/yaml-output (@mandrean)
- f249ee4 Remove extra space (@kalinchernev)
- 87527c2 Turn off coverage requirements for example code (@kalinchernev)
- 5d4e599 Avoid tags duplication (@kalinchernev)
- 8986e85 Add CLI support for YAML output (@mandrean)
- 103aab6 Update documentation of CLI tool (@kalinchernev)
- ccfa0f9 Watch only files with API documentation (@kalinchernev)
- 5713535 Improve cli watch task (@kalinchernev)
- fd16962 Include watch task in cli tool. (@kalinchernev)

184.0.50 v1.8.4 (2016/12/29 12:04 +00:00)

- 20b9516 Release v1.8.4 (@kalinchernev)
- #45 Merge pull request #45 from toefraz/master (@toefraz)
- 5ff7718 Adding tests to verify
- e0bcb38 Ignore deprecation check (@toefraz)

184.0.51 v1.8.3 (2016/12/04 13:23 +00:00)

- d777cde Release v1.8.3 (@kalinchernev)
- #43 Merge pull request #43 from Surnet/chore/dependencies (@Surnet)
- ce3859f chore(dependencies): update modules (@kalinchernev)

184.0.52 v1.8.2 (2016/11/18 19:08 +00:00)

- 44e4168 Release v1.8.2 (@kalinchernev)
- #42 Merge pull request #42 from cikasfm/master (@cikasfm)
- a85c4b8 Updating sample to comply with Swagger v2 (@cikasfm)

184.0.53 v1.8.1 (2016/10/27 11:55 +00:00)

- dd1aa0a Release v1.8.1 (@kalinchernev)
- eaa1e80 Release v1.7.0 (@kalinchernev)
- #40 Merge pull request #40 from Surnet/feature/swagger-spec-revision (@Surnet)
- 50c99c0 add iterator to seek for problematic tags (@kalinchernev)
- 1d27a43 include seekWrong method (@kalinchernev)
- b0e73f1 Preparing for general reporting of deprecated properties. (@kalinchernev)
- 4f3641d Including checks for securityDefinitions and responses (@kalinchernev)
- 87f6cf5 Adding test for accepting parameter in singular and plural (@kalinchernev)
- 70f924d Updating example files (@kalinchernev)
- 6360c1d Reducing function cyclomatic complexity (@kalinchernev)
- ae4e4b8 Separating the switch. (@kalinchernev)
- f204dde Correcting definition. (@kalinchernev)
- 5c55c1e End of day commit. (@kalinchernev)
- ff58ba0 addDataToSwaggerObject requires parameters. (@kalinchernev)
- 0579d90 Separating swagger related functions in a module. (@kalinchernev)

184.0.54 v1.7.0 (2016/09/17 17:33 +00:00)

- 973e6b0 Bump to v1.7.0 (@drGrove)
- #36 Tags property parsing refactoring (#36) (@kalinchernev)

184.0.55 v1.6.0 (2016/09/02 05:30 +00:00)

- c809ffc Bump to v1.6.0 (@drGrove)
- #29 Don't override user provided swagger definition options (#29) (@brantw)
- #30 support multiple tags as an array and in definition (#30) (@efmr)
- #31 CLI should use package version (#31) (@drGrove)

184.0.56 v1.5.0 (2016/08/30 07:13 +00:00)

- 3891ee2 Bump to v1.5.0 (@drGrove)

184.0.57 v1.4.1 (2016/08/30 07:10 +00:00)

- 0752561 Bump to v1.4.1 (@drGrove)
- #27 CLI (#27) (@kalinchernev)

184.0.58 v1.4.0 (2016/08/25 08:52 +00:00)

- 72ae1bb Bump to v1.4.0 (@drGrove)

184.0.59 v1.3.1 (2016/07/03 18:12 +00:00)

- `3cccf49` Version bump (@chdanielmueller)
- `#23` Merge pull request #23 from simast/master (@simast)
- `f01b48b` Added swagger tags support.
- `e6d3eec` Update README.md (@chdanielmueller)

184.0.60 v1.3.0 (2016/04/07 14:10 +00:00)

- `#20` Merge pull request #20 from jonboiser/master (@jonboiser)
- `0af0c5d` Update to version 1.3.0
- `257bdd3` Remove Login2 definition.
- `1cb5fa0` Create test case for using globs options.apis.
- `2753f8e` Factor out glob converter function.
- `e73de29` Raise jshint maxstatements to 20.
- `2e2ef7c` options.api accepts glob strings

184.0.61 v1.2.1 (2016/03/12 14:09 +00:00)

- `e42fb0c` Update version number, change node test version (@chdanielmueller)
- `#18` Merge pull request #18 from mprokopowicz/master (@mprokopowicz)
- `78e7b77` use path.extname in favor of path.parse to stay compatible with node 0.10.x (@mprokopowicz)

184.0.62 v1.2.0 (2016/02/26 18:52 +00:00)

- `0b7fe37` Updating version number and dependencies (@chdanielmueller)
- `#15` Merge pull request #15 from Cloudoki/master (@Cloudoki)
- `f31413a` remove console.log and add forgotten file (@efmr)
- `3e60da2` accept yaml as api docs (@efmr)
- `4d7da54` pass lint and test (@efmr)
- `20b95f2` add parameters, responses, securityDefinitions (@efmr)
- `#13` Merge pull request #13 from ami44/master (@ami44)
- `863a3bb` Add example to load external definitions

184.0.63 v1.1.2 (2015/11/25 10:06 +00:00)

- `fb86b3d` Update version number to 1.1.2 (@chdanielmueller)
- `#11` Merge pull request #11 from trendfischer/master (@trendfischer)
- `b1695c0` Removed required path module (@trendfischer)
- `ec1de9c` Removed filename restriction for "\∗.js" (@trendfischer)

184.0.64 v1.1.1 (2015/09/27 17:07 +00:00)

- `afe96f0` Updating dependencies (@chdanielmueller)
- `#10` Merge pull request #10 from drGrove/docs/definition (@drGrove)
- `c77cd0e` Adds docs for definition creation (@drGrove)

184.0.65 v1.1.0 (2015/08/30 15:20 +00:00)

- `511778e` Updating version number and fixing spelling error (@chdanielmueller)
- `#8` Merge pull request #8 from drGrove/feature/definitions (@drGrove)
- `f264e0e` Adds editorconfig (@drGrove)
- `6a512b1` Adds support for external definitions. Updates maxstatements to 15 (@drGrove)

184.0.66 v1.0.1 (2015/08/23 08:25 +00:00)

- `359f239` Update version number (@chdanielmueller)
- `#6` Merge pull request #6 from drGrove/master (@drGrove)
- `0f7a70c` Adds parser for separate definitions (@drGrove)
- `7b88757` Removes console.logs (@drGrove)
- `07e40ad` Adds object merging to allow for swagger docs of same endpoint with different methods [GET,POST] to be associated with the endpoint (@drGrove)
- `#5` Merge pull request #5 from chdanielmueller/master (@chdanielmueller)
- `ccd119e` More documentation (@chdanielmueller)
- `#4` Merge pull request #4 from chdanielmueller/master (@chdanielmueller)
- `b0e7534` Adding npm and Gratipay badges (@chdanielmueller)

184.0.67 v1.0.0 (2015/06/09 16:33 +00:00)

- `#3` Merge pull request #3 from chdanielmueller/master (@chdanielmueller)
- `62bd690` Update to version 1.0.0 (@chdanielmueller)
- `3077045` README.md (@chdanielmueller)
- `1ddc012` Remove swagger-tools (@chdanielmueller)
- `f63a68b` Exclude swagger serving from library (@chdanielmueller)
- `d6fa533` Clean swagger spec in example (@chdanielmueller)
- `#2` Merge pull request #2 from chdanielmueller/master (@chdanielmueller)
- `4c0a6e0` Fixing circle.yml (@chdanielmueller)
- `96e12c7` Adding Badges to README.md (@chdanielmueller)
- `3fa16f2` Adding Codacy Coverage Token (@chdanielmueller)
- `64b3121` Typo (@chdanielmueller)
- `87e1730` Allow more statements (@chdanielmueller)
- `e81460e` Increasing coverage (@chdanielmueller)
- `f3449bb` Refactoring (@chdanielmueller)
- `617a73f` Adding tests (@chdanielmueller)
- `2d2ecfe` Prepare for CI (@chdanielmueller)
- `fea32b5` Cleanup README.md (@chdanielmueller)
- `535b66d` Adding references to external dependencies (@chdanielmueller)

- c99b300 Remove jsdoc from example (@chdanielmueller)
- 365f12b Fix jscs and jshint errors (@chdanielmueller)
- d7d6964 Adding coverage test (@chdanielmueller)
- e3f6c3f Tweaks to the package.json (@chdanielmueller)
- 5a68333 Remove example files (@chdanielmueller)
- 0df567c Rename project, change URLs, Fix LICENSE (@chdanielmueller)
- 1562666 Adding tests (@chdanielmueller)
- e2768ac Add jsdoc config (@chdanielmueller)
- 79a0758 Implement jscs and jshint (@chdanielmueller)
- #1 Merge pull request #1 from chdanielmueller/refactor (@chdanielmueller)
- 5313d40 Working login example (@chdanielmueller)
- a6b81b6 Adding swaggerDefinition (@chdanielmueller)
- c80bb01 Switch to simple PetStore example (@chdanielmueller)
- e807d99 Adding swagger-example in json and yml (@chdanielmueller)
- 824d1c1 Beautify Code (@chdanielmueller)
- afed184 Remove express as dependency (@chdanielmueller)
- 0051b73 Change Formatting (@chdanielmueller)
- 5cd7b15 Remove unneeded swaggerUI (@chdanielmueller)
- dffb84a Modifiable URLs for documentation and swaggerUI (@chdanielmueller)
- ebf2d72 Adding swagger-tools (@chdanielmueller)
- 5b4c1eb It works!!! Now I have to refactor and do some tweaks. (@devlouisc)
- d850f3f Fix listing issues (@devlouisc)
- 7a82b46 Remove example directory from ESLint ignore (@devlouisc)
- fff57be Clean up comments (@devlouisc)
- e65e356 Add beginnings of new example app (@devlouisc)
- f05a51f Fix linting (@devlouisc)
- a98b8ab Reorganize files (@devlouisc)
- 51ac313 Update metadata with new project name and trim dependencies (@devlouisc)
- 8c86b95 Remove support for YAML and CoffeeScript (@devlouisc)
- e0df0ed Remove old implementation (@devlouisc)
- 00f3be7 Fix listing problems (@devlouisc)
- 0ba898b Finish implementation of logic (@devlouisc)
- cc1fedd Add example Swagger 2.0 JSON object (@devlouisc)
- 62eae1d Add documentation and beginnings of parsing logic (@devlouisc)
- 465d359 Fine tune linting (@devlouisc)
- bd2b1b4 Add and configure ESLint (@devlouisc)

- e88fc7a Add Swagger UI (@devlouisc)
- 95bd90f Add options validation (@devlouisc)
- 33a1f98 Update license (@devlouisc)
- 1160f1d Update npm dependency references (@devlouisc)
- 3f54b0d Remove ignored files that do not pertain to my system (@devlouisc)
- 8031793 Remove global flag from npm install (@devlouisc)
- a75ff89 Fix formatting of comments (@devlouisc)
- ca5e5b2 Format and alphabetize text (@devlouisc)
- #25 Merge pull request #25 from miltonguty/patch-1 (@miltonguty)
- 8e6f862 enable "Access-Control-Allow-Origin" in request (@miltonguty)
- #24 Merge pull request #24 from relvao/master (@relvao)
- e61fcbd fix Express 4 warning (@relvao)
- #21 Merge pull request #21 from ElectricHummingbird/master (@ElectricHummingbird)
- 4ef14e4 allow fullSwaggerJSONPath to be set in the cfg.
- #18 Merge pull request #18 from tlvince/refactor/jshint (@tlvince)
- #17 Merge pull request #17 from tlvince/feature/info (@tlvince)
- 388396c Pass info object (@tlvince)
- 7aa9e99 Fix JSHint warning (@tlvince)
- #16 Merge pull request #16 from gierschv/feat-expose (@gierschv)
- f136d74 Expose descriptor & resources (@gierschv)
- #15 Merge pull request #15 from relvao/master (@relvao)
- 6190d11 Update index.js (@relvao)
- #12 Merge pull request #12 from sposmen/master (@sposmen)
- #13 Merge pull request #13 from johnywith1n/master (@johnywith1n)
- 0839ab0 remove doc about basePath being optional.
- bc2fba2 set version of express to 3.5.1 since v4.0.0 doesnt work with the example app.
- d8c80f3 require base path. match swagger json endpoints based on the full path which may include any parts from the basePath after the host.
- 32b28b2 Middleware function support (@sposmen)
- #10 Merge pull request #10 from sposmen/master (@sposmen)
- 7c15437 Merge branch 'master' of github.com:sposmen/swagger-express (@sposmen)
- 36e27fd README documentantion complement (@sposmen)
- 5e022e6 README documentantion complement (@sposmen)
- 1aebc40 New Jade standard (avoiding errors) (@sposmen)
- 5768a3d Specific origin of examples (@sposmen)
- e528cd2 Coffee support added (@sposmen)

- e38c3b9 update version (@fliptoo)
- 14ccf0b fixed swagger option (@fliptoo)
- #6 Merge pull request #6 from slajax/master (@slajax)
- 9623f70 added basePath to readme (@kc-dot-io)
- 1b4f3d6 basePath should still be able to be set so it can be passed to swagger.js (@kc-dot-io)
- e97684d Fixes #4 (@fliptoo)
- 9e89f97 Update Example (@fliptoo)
- 4581d7a Fixes #4 (@fliptoo)
- d481656 Update Version (@fliptoo)
- 96262f9 Update Example (@fliptoo)
- 6e29773 Set version on js-yaml to suppress deprecation warning output. (@fliptoo)
- #3 Merge pull request #3 from calmdev/master (@calmdev)
- 39ca987 Updates the README file. (@calmdev)
- cf31676 Base path isn't needed anymore, because we aren't modifying the UI's source. (@calmdev)
- da8bbc1 This doesn't work with latest swagger UI. Also, people should be encouraged to update and build the swagger UI from source instead of relying on swagger-express to modify the default discovery URL. (@calmdev)
- 3b86bff Serves the swagger static assets from specified swaggerUI path. (@calmdev)
- c182a47 Adds option to specify swagger's web interface url. (@calmdev)
- ce80517 Adds option to specify swagger's JSON url. (@calmdev)
- 196d119 Update README.md (@fliptoo)
- 37a06c1 update version (@fliptoo)
- 69dc86d commit package.json (@fliptoo)
- #1 Merge pull request #1 from stelcheck/develop (@stelcheck)
- 579f1bb ∗ Bugfix: was using path instead of resourcePath (@stelcheck)
- 98fe645 ∗ Code linting (@stelcheck)
- 53212be Update README.md (@fliptoo)
- 7a6f4b2 Update README.md (@fliptoo)
- d379519 fix require path (@fliptoo)
- 251fbaa fix resources (@fliptoo)
- fed18da edit npmignore (@fliptoo)
- 086f3fc change to example (@fliptoo)
- 4f27c86 Update README.md (@fliptoo)
- 6a2e19c Create README.md (@fliptoo)
- 7b1d808 initial commit (@fliptoo)
- 6f64e49 Initial commit (@fliptoo)

Chapter 185

swagger-jsdoc

This library reads your `JSDoc`-annotated source code and generates an `OpenAPI (Swagger) specification`.

185.1 Getting started

Imagine having API files like these:

```
/**
 * @openapi
 * /:
 *   get:
 *     description: Welcome to swagger-jsdoc!
 *     responses:
 *       200:
 *         description: Returns a mysterious string.
 */
app.get('/', (req, res) => {
  res.send('Hello World!');
});
```

The library will take the contents of `@openapi` (or `@swagger`) with the following configuration:

```
const swaggerJsdoc = require('swagger-jsdoc');
const options = {
  definition: {
    openapi: '3.0.0',
    info: {
      title: 'Hello World',
      version: '1.0.0',
    },
  },
  apis: ['./src/routes*.js'], // files containing annotations as above
};
const openapiSpecification = swaggerJsdoc(options);
```

The resulting `openapiSpecification` will be a `swagger tools`-compatible (and validated) specification.

185.2 System requirements

- Node.js 12.x or higher

You are viewing `swagger-jsdoc` v6 which is published in CommonJS module system.

185.3 Installation

```
npm install swagger-jsdoc --save
```

Or

```
yarn add swagger-jsdoc
```

185.4 Supported specifications

- OpenAPI 3.x

- Swagger 2
- AsyncAPI 2.0

185.5 Validation of swagger docs

By default `swagger-jsdoc` tries to parse all docs to it's best capabilities. If you'd like to you can instruct an Error to be thrown instead if validation failed by setting the options flag `failOnErrors` to `true`. This is for instance useful if you want to verify that your swagger docs validate using a unit test.

```
const swaggerJsdoc = require('swagger-jsdoc');
const options = {
  failOnErrors: true, // Whether or not to throw when parsing errors. Defaults to false.
  definition: {
    openapi: '3.0.0',
    info: {
      title: 'Hello World',
      version: '1.0.0',
    },
  },
  apis: ['./src/routes*.js'],
};
const openapiSpecification = swaggerJsdoc(options);
```

185.6 Documentation

Click on the version you are using from navigation bar for further details.

Chapter 186

swagger-jsdoc

This library reads your JSDoc-annotated source code and generates an OpenAPI (Swagger) specification.

186.1 Getting started

Imagine having API files like these:

```
/**
 * @openapi
 * /:
 *   get:
 *     description: Welcome to swagger-jsdoc!
 *     responses:
 *       200:
 *         description: Returns a mysterious string.
 */
app.get('/', (req, res) => {
  res.send('Hello World!');
});
```

The library will take the contents of `@openapi` (or `@swagger`) with the following configuration:

```
const swaggerJsdoc = require('swagger-jsdoc');
const options = {
  definition: {
    openapi: '3.0.0',
    info: {
      title: 'Hello World',
      version: '1.0.0',
    },
  },
  apis: ['./src/routes*.js'], // files containing annotations as above
};
const openapiSpecification = swaggerJsdoc(options);
```

The resulting `openapiSpecification` will be a swagger tools-compatible (and validated) specification.

186.2 System requirements

- Node.js 12.x or higher

You are viewing `swagger-jsdoc` v6 which is published in CommonJS module system.

186.3 Installation

```
npm install swagger-jsdoc --save
```

Or

```
yarn add swagger-jsdoc
```

186.4 Supported specifications

- OpenAPI 3.x

- Swagger 2
- AsyncAPI 2.0

186.5 Validation of swagger docs

By default `swagger-jsdoc` tries to parse all docs to it's best capabilities. If you'd like to you can instruct an Error to be thrown instead if validation failed by setting the options flag `failOnErrors` to `true`. This is for instance useful if you want to verify that your swagger docs validate using a unit test.

```
const swaggerJsdoc = require('swagger-jsdoc');
const options = {
  failOnErrors: true, // Whether or not to throw when parsing errors. Defaults to false.
  definition: {
    openapi: '3.0.0',
    info: {
      title: 'Hello World',
      version: '1.0.0',
    },
  },
  apis: ['./src/routes*.js'],
};
const openapiSpecification = swaggerJsdoc(options);
```

186.6 Documentation

Click on the version you are using for further details:

- `7.x`
- `6.x`
- `5.x`

Chapter 187

Change Log

All notable changes will be documented in this file. Swagger Parser adheres to `Semantic Versioning`.

187.1 <a href="https://github.com/APIDevTools/swagger-parser/tree/v10.0.0" >v10.0.0</a> (2020-07-10)

187.1.0.1 Breaking Changes

- Removed the `YAML` export. We recommend using `@stoplight/yaml` instead

187.1.0.2 Other Changes

- Added a new `continueOnError option` that allows you to get all errors rather than just the first one

`Full Changelog`

187.2 <a href="https://github.com/APIDevTools/swagger-parser/tree/v9.0.0" >v9.0.0</a> (2020-03-14)

- Moved Swagger Parser to the `@APIDevTools scope` on NPM
- The "swagger-parser" NPM package is now just a wrapper around the scoped "@apidevtools/swagger-parser" package

`Full Changelog`

187.3 <a href="https://github.com/APIDevTools/swagger-parser/tree/v8.0.0" >v8.0.0</a> (2019-06-22)

187.3.0.1 Potentially Breaking Changes

- `The validate() function` now uses `the official JSON Schemas` for Swagger 2.0 and OpenAPI 3.0 schema validation. We tested this change on `over 1,500 real-world APIs` and there were **no breaking changes**, but we're bumped the major version number just to be safe.

`Full Changelog`

187.4 <a href="https://github.com/APIDevTools/swagger-parser/tree/v7.0.0" >v7.0.0</a> (2019-06-12)

187.4.0.1 Breaking Changes

- Dropped support for Node 6
- Updated all code to ES6+ syntax (async/await, template literals, arrow functions, etc.)
- No longer including a pre-built bundle in the package. such as `Webpack`, `Rollup`, `Parcel`, or `Browserify` to include Swagger Parser in your app

187.4.0.2 Other Changes

- Added `TypeScript definitions`

`Full Changelog`

187.5 <a href="https://github.com/APIDevTools/swagger-parser/tree/v6.0.0" >v6.0.0</a> (2018-10-05)

- Dropped support for `Bower`, since it has been deprecated
- Removed the `debug` dependency

`Full Changelog`

187.6 <a href="https://github.com/APIDevTools/swagger-parser/tree/v5.0.0" >v5.0.0</a> (2018-05-25)

- After `months` and `months` of delays, initial support for OpenAPI 3.0 is finally here! A big "Thank You!" to `Leo Long` for doing the work and submitting `PR #88`.

`Full Changelog`

187.7 <a href="https://github.com/APIDevTools/swagger-parser/tree/v4.1.0" >v4.1.0</a> (2018-04-11)

- `@marcelstoer` submitted `PR #83` and `PR #84`, both of which improve the `validate() method`. It will now detect when a JSON Schema in your API definition has `required` properties that don't exist.

`Full Changelog`

187.8 <a href="https://github.com/APIDevTools/swagger-parser/tree/v4.0.0" >v4.0.0</a> (2017-10-19)

187.8.0.1 Breaking Changes

- Update the `Swagger 2.0 JSON schema`, so it's possible that an API that previously passed validation may no longer pass due to changes in the Swagger schema
- To reduce the size of this library, it no longer includes polyfills for `Promises` and `TypedArrays`, which are natively supported in the latest versions of Node and web browsers. If you need to support older browsers (such as IE9), then just use `this Promise polyfill` and `this TypedArray polyfill`.

187.8.0.2 Minor Changes

- `PR #74` - Fixes `an edge-case bug` with the `validate()` method and `x-` vendor extensions

`Full Changelog`

187.9 <a href="https://github.com/APIDevTools/swagger-parser/tree/v4.0.0-beta.2" >v4.0.0-beta.2</a> (2016-04-25)

187.9.0.1 Just one small fix

Fixed `issue #13`. You can now pass a URL *and* an object to any method.
`SwaggerParser.validate("http://example.com/my-schema.json", mySchemaObject, {})`

> **NOTE:** As shown in the example above, you *must* also pass an options object (even an empty object will work), otherwise, the method signature looks like you're just passing a URL and options.

`Full Changelog`

187.10 <a href="https://github.com/APIDevTools/swagger-parser/tree/v4.0.0-beta.1" >v4.0.0-beta.1</a> (2016-04-10)

187.10.0.1 Plug-ins !!!

That's the major new feature in this version. Originally requested in `PR #8`, and refined a few times over the past few months, the plug-in API is now finalized and ready to use. You can now define your own `resolvers` and `parsers`.

187.10.0.2 Breaking Changes

The available `options have changed`, mostly due to the new plug-in API. There's not a one-to-one mapping of old options to new options, so you'll have to read the docs and determine which options you need to set. If any. The out-of-the-box configuration works for most people.
All of the `caching options have been removed`. Instead, files are now cached by default, and the entire cache is reset for each new parse operation. Caching options may come back in a future release, if there is enough demand for it. If you used the old caching options, please open an issue and explain your use-case and requirements. I need a better understanding of what caching functionality is actually needed by users.

187.10.0.3 Bug Fixes

Lots of little bug fixes, and a couple major bug fixes:

- `completely rewrote the bundling logic` to fix `issue #16`
- Added support for `root-level $refs`

`Full Changelog`

187.11 <a href="https://github.com/APIDevTools/swagger-parser/tree/v3.3.0" >v3.3.0</a> (2015-10-02)

Updated to the latest version of the Official Swagger 2.0 Schema. The schema hadn't been updated for six months, so Swagger Parser was missing several recent changes.
Full Changelog

187.12 <a href="https://github.com/APIDevTools/swagger-parser/tree/v3.2.0" >v3.2.0</a> (2015-10-01)

Swagger Parser now uses call-me-maybe to support promises or callbacks.
Full Changelog

187.13 <a href="https://github.com/APIDevTools/swagger-parser/tree/v3.1.0" >v3.1.0</a> (2015-09-28)

Fixed several bugs with circular references, particularly with the validate() method.
Added a new $refs.circular option to determine how circular references are handled. Options are fully-dereferencing them (default), throwing an error, or ignoring them.
Full Changelog

187.14 <a href="https://github.com/APIDevTools/swagger-parser/tree/v3.0.0" >v3.0.0</a> (2015-09-25)

This is a **complete rewrite** of Swagger Parser. Major changes include:
Swagger 2.0 Compliant
Previous versions of Swagger Parser were based on early drafts of Swagger 2.0, and were not compliant with the final version of the spec. Swagger Parser v3.0 is now compliant with the final spec as well as related specs, such as JSON Reference and JSON Pointer
All-New API
The old API only had a single method: parse(). But depending on which options you passed it, the method did *much* more than its name implied. The new API has separate methods to make things a bit more intuitive. The most commonly used will be validate(), bundle(), and dereference(). The parse() and resolve() methods are also available, but these are mostly just for internal use by the other methods.
Asynchronous API
The old API was "asynchronous", but it relied on global state, so it did not support multiple simultaneous operations. The new API is truly asynchronous and supports both ES6 Promises and Node-style callbacks.
New JSON Schema Validator
Swagger Parser has switched from tv4 to Z-Schema, which is faster and performs more accurate validation. This means that some APIs that previously passed validation will now fail. But that's a *good* thing!
Full Changelog

Chapter 188

Swagger 2.0 and OpenAPI 3.0 parser/validator

[](LICENSE)

188.1 Features

- Parses Swagger specs in **JSON** or **YAML** format
- Validates against the `Swagger 2.0 schema` or `OpenAPI 3.0 Schema`
- `Resolves` all `$ref` pointers, including external files and URLs
- Can `bundle` all your Swagger files into a single file that only has *internal* `$ref` pointers
- Can `dereference` all `$ref` pointers, giving you a normal JavaScript object that's easy to work with
- **`Tested`** in Node.js and all modern web browsers on Mac, Windows, and Linux
- Tested on **`over 1,500 real-world APIs`** from Google, Microsoft, Facebook, Spotify, etc.
- Supports `circular references`, nested references, back-references, and cross-references
- Maintains object reference equality — `$ref` pointers to the same value always resolve to the same object instance

188.2 Related Projects

- `Swagger CLI`
- `Swagger Express Middleware`

188.3 Example

```
SwaggerParser.validate(myAPI, (err, api) => {
  if (err) {
    console.error(err);
  }
  else {
    console.log("API name:  %s, Version:  %s", api.info.title, api.info.version);
  }
});
```

Or use `async/await` or `Promise` syntax instead. The following example is the same as above:

```
try {
  let api = await SwaggerParser.validate(myAPI);
  console.log("API name:  %s, Version:  %s", api.info.title, api.info.version);
}
catch(err) {
  console.error(err);
}
```

For more detailed examples, please see the `API Documentation`

188.4 Installation

Install using npm:

```
npm install @apidevtools/swagger-parser
```

188.5 Usage

When using Swagger Parser in Node.js apps, you'll probably want to use **CommonJS** syntax:

```
const SwaggerParser = require("@apidevtools/swagger-parser");
```

When using a transpiler such as Babel or TypeScript, or a bundler such as Webpack or Rollup, you can use **ECMAScript modules** syntax instead:

```
import SwaggerParser from "@apidevtools/swagger-parser";
```

188.6 Browser support

Swagger Parser supports recent versions of every major web browser. Older browsers may require Babel and/or polyfills.

To use Swagger Parser in a browser, you'll need to use a bundling tool such as Webpack, Rollup, Parcel, or Browserify. Some bundlers may require a bit of configuration, such as setting `browser: true` in rollup-plugin-resolve.

188.7 API Documentation

Full API documentation is available right here

188.8 Contributing

I welcome any contributions, enhancements, and bug-fixes. Open an issue on GitHub and submit a pull request.

188.8.0.1 Building/Testing

To build/test the project locally on your computer:

1. **Clone this repo**
   `git clone https://github.com/APIDevTools/swagger-parser.git`
2. **Install dependencies**
   `npm install`
3. **Run the build script**
   `npm run build`
4. **Run the tests**
   `npm test`

188.9 License

Swagger Parser is 100% free and open-source, under the [MIT license](LICENSE). Use it however you want.

This package is Treeware. If you use it in production, then we ask that you **buy the world a tree** to thank us for our work. By contributing to the Treeware forest you'll be creating employment for local families and restoring wildlife habitats.

188.10 Big Thanks To

Thanks to these awesome companies for their support of Open Source developers

Chapter 189

Swagger UI Dist

189.1 API

This module, `swagger-ui-dist`, exposes Swagger-UI's entire dist folder as a dependency-free npm module.
Use `swagger-ui` instead, if you'd like to have npm install dependencies for you.
`SwaggerUIBundle` and `SwaggerUIStandalonePreset` can be imported:

```
import { SwaggerUIBundle, SwaggerUIStandalonePreset } from "swagger-ui-dist"
```

To get an absolute path to this directory for static file serving, use the exported `getAbsoluteFSPath` method:

```
const swaggerUiAssetPath = require("swagger-ui-dist").getAbsoluteFSPath()
// then instantiate server that serves files from the swaggerUiAssetPath
```

For anything else, check the `Swagger-UI` repository.

Chapter 190

Swagger UI Express

Statements	Branches	Functions	Lines

This module allows you to serve auto-generated `swagger-ui` generated API docs from express, based on a `swagger.json` file. The result is living documentation for your API hosted from your API server via a route.
Swagger version is pulled from npm module swagger-ui-dist. Please use a lock file or specify the version of swagger-ui-dist you want to ensure it is consistent across environments.
You may be also interested in:

- `swagger-jsdoc`: Allows you to markup routes with jsdoc comments. It then produces a full swagger yml config dynamically, which you can pass to this module to produce documentation. See below under the usage section for more info.
- `swagger tools`: Various tools, including swagger editor, swagger code gen etc.

190.1 Usage

Install using npm:

```
$ npm install swagger-ui-express
```

Express setup `app.js`

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument));
```

or if you are using Express router

```
const router = require('express').Router();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
router.use('/api-docs', swaggerUi.serve);
router.get('/api-docs', swaggerUi.setup(swaggerDocument));
```

Open `http://`<app_host>:<app_port>/api-docs in your browser to view the documentation.
If you want to set up routing based on the swagger document checkout `swagger-express-router`

190.1.1 <a href="https://www.npmjs.com/package/swagger-jsdoc" >swagger-jsdoc</a>

If you are using swagger-jsdoc simply pass the swaggerSpec into the setup function:

```
// Initialize swagger-jsdoc -> returns validated swagger spec in json format
const swaggerSpec = swaggerJSDoc(options);
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerSpec));
```

190.1.2 Swagger Explorer

By default the Swagger Explorer bar is hidden, to display it pass true as the 'explorer' property of the options to the setup function:

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {
  explorer: true
};
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument, options));
```

190.1.3 Custom swagger options

To pass custom options e.g. validatorUrl, to the SwaggerUi client pass an object as the 'swaggerOptions' property of the options to the setup function:

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {
  swaggerOptions: {
    validatorUrl: null
  }
};
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument, options));
```

For all the available options, refer to `Swagger UI Configuration`

190.1.4 Custom CSS styles

To customize the style of the swagger page, you can pass custom CSS as the 'customCss' property of the options to the setup function.
E.g. to hide the swagger header:

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {
  customCss: '.swagger-ui .topbar { display: none }'
};
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument, options));
```

190.1.5 Custom CSS styles from Url

You can also pass the url to a custom css file, the value must be the public url of the file and can be relative or absolute to the swagger path.

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {
  customCssUrl: '/custom.css'
};
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument, options));
```

You can also pass an array of css urls to load multiple css files.

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {
  customCssUrl: [
    '/custom.css',
    'https://example.com/other-custom.css'
  ]
};
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument, options));
```

190.1.6 Custom JS

If you would like to have full control over your HTML you can provide your own javascript file, value accepts absolute or relative path. Value must be the public url of the js file.

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {
  customJs: '/custom.js'
};
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument, options));
```

You can also pass an array of js urls to load multiple js files.

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {
  customJs: [
    '/custom.js',
    'https://example.com/other-custom.js'
  ]
};
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument, options));
```

It is also possible to add inline javascript, either as string or array of string.

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {
  customJsStr: 'console.log("Hello World")'
};
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument, options));
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {
  customJsStr: [
    'console.log("Hello World")',
    `
    var x = 1
    console.log(x)
    `
  ]
};
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument, options));
```

190.1.7 Load swagger from url

To load your swagger from a url instead of injecting the document, pass `null` as the first parameter, and pass the relative or absolute URL as the 'url' property to 'swaggerOptions' in the setup function.

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
var options = {
  swaggerOptions: {
    url: 'http://petstore.swagger.io/v2/swagger.json'
  }
}
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(null, options));
```

To load multiple swagger documents from urls as a dropdown in the explorer bar, pass an array of object with `name` and `url` to 'urls' property to 'swaggerOptions' in the setup function.

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
var options = {
  explorer: true,
  swaggerOptions: {
    urls: [
      {
        url: 'http://petstore.swagger.io/v2/swagger.json',
        name: 'Spec1'
      },
      {
        url: 'http://petstore.swagger.io/v2/swagger.json',
        name: 'Spec2'
      }
    ]
  }
}
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(null, options));
```

Make sure 'explorer' option is set to 'true' in your setup options for the dropdown to be visible.

190.1.8 Load swagger from yaml file

To load your swagger specification yaml file you need to use a module able to convert yaml to json; for instance `yaml`.

```
npm install yaml
```

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const fs = require("fs")
const YAML = require('yaml')
const file  = fs.readFileSync('./swagger.yaml', 'utf8')
const swaggerDocument = YAML.parse(file)
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument));
```

190.1.9 Modify swagger file on the fly before load

To dynamically set the host, or any other content, in the swagger file based on the incoming request object you may pass the json via the req object; to achieve this just do not pass the the swagger json to the setup function and it will look for `swaggerDoc` in the `req` object.

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {}
app.use('/api-docs', function(req, res, next){
    swaggerDocument.host = req.get('host');
    req.swaggerDoc = swaggerDocument;
    next();
}, swaggerUi.serveFiles(swaggerDocument, options), swaggerUi.setup());
```

190.1.10 Two swagger documents

To run 2 swagger ui instances with different swagger documents, use the serveFiles function instead of the serve function. The serveFiles function has the same signature as the setup function.

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocumentOne = require('./swagger-one.json');
const swaggerDocumentTwo = require('./swagger-two.json');
var options = {}
app.use('/api-docs-one', swaggerUi.serveFiles(swaggerDocumentOne, options),
      swaggerUi.setup(swaggerDocumentOne));
app.use('/api-docs-two', swaggerUi.serveFiles(swaggerDocumentTwo, options),
      swaggerUi.setup(swaggerDocumentTwo));
app.use('/api-docs-dynamic', function(req, res, next){
  req.swaggerDoc = swaggerDocument;
  next();
}, swaggerUi.serveFiles(), swaggerUi.setup());
```

190.1.11 Link to Swagger document

To render a link to the swagger document for downloading within the swagger ui - then serve the swagger doc as an endpoint and use the url option to point to it:

```
const express = require('express');
const app = express();
const swaggerUi = require('swagger-ui-express');
const swaggerDocument = require('./swagger.json');
var options = {
    swaggerOptions:  {
        url:  "/api-docs/swagger.json",
    },
}
app.get("/api-docs/swagger.json", (req, res) => res.json(swaggerDocument));
app.use('/api-docs', swaggerUi.serveFiles(null, options), swaggerUi.setup(null, options));
```

190.2 Requirements

- Node v0.10.32 or above
- Express 4 or above

190.3 Testing

- Install phantom
- `npm install`
- `npm test`

Chapter 191

1.0.1 / 2021-11-14

- pref: enable strict mode

191.1 1.0.0 / 2018-07-09

- Initial release

Chapter 192

toidentifier

Convert a string of words to a JavaScript identifier

192.1 Install

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install toidentifier
```

192.2 Example

```
var toIdentifier = require('toidentifier')
console.log(toIdentifier('Bad Request'))
// => "BadRequest"
```

192.3 API

This CommonJS module exports a single default function: `toIdentifier`.

192.3.1 toIdentifier(string)

Given a string as the argument, it will be transformed according to the following rules and the new string will be returned:

1. Split into words separated by space characters (`0x20`).
2. Upper case the first character of each word.
3. Join the words together with no separator.
4. Remove all non-word (`[0-9a-z_]`) characters.

192.4 License

[MIT](LICENSE)

192.5 autotoc_md3338

Chapter 193

1.6.18 / 2019-04-26

- Fix regression passing request object to `typeis.is`

193.1 1.6.17 / 2019-04-25

- deps: mime-types2.1.24
  - Add Apple file extensions from IANA
  - Add extension `.csl` to `application/vnd.citationstyles.style+xml`
  - Add extension `.es` to `application/ecmascript`
  - Add extension `.nq` to `application/n-quads`
  - Add extension `.nt` to `application/n-triples`
  - Add extension `.owl` to `application/rdf+xml`
  - Add extensions `.siv` and `.sieve` to `application/sieve`
  - Add extensions from IANA for `image/*` types
  - Add extensions from IANA for `model/*` types
  - Add extensions to HEIC image types
  - Add new mime types
  - Add `text/mdx` with extension `.mdx`
- perf: prevent internal `throw` on invalid type

193.2 1.6.16 / 2018-02-16

- deps: mime-types2.1.18
  - Add `application/raml+yaml` with extension `.raml`
  - Add `application/wasm` with extension `.wasm`
  - Add `text/shex` with extension `.shex`
  - Add extensions for JPEG-2000 images
  - Add extensions from IANA for `message/*` types
  - Add extension `.mjs` to `application/javascript`
  - Add extension `.wadl` to `application/vnd.sun.wadl+xml`
  - Add extension `.gz` to `application/gzip`
  - Add glTF types and extensions
  - Add new mime types
  - Update extensions `.md` and `.markdown` to be `text/markdown`
  - Update font MIME types
  - Update `text/hjson` to registered `application/hjson`

193.3 1.6.15 / 2017-03-31

- deps: mime-types2.1.15
  - Add new mime types

193.4 1.6.14 / 2016-11-18

- deps: mime-types2.1.13
  - Add new mime types

193.5 1.6.13 / 2016-05-18

- deps: mime-types2.1.11
  - Add new mime types

193.6 1.6.12 / 2016-02-28

- deps: mime-types2.1.10
  - Add new mime types
  - Fix extension of `application/dash+xml`
  - Update primary extension for `audio/mp4`

193.7 1.6.11 / 2016-01-29

- deps: mime-types2.1.9
  - Add new mime types

193.8 1.6.10 / 2015-12-01

- deps: mime-types2.1.8
  - Add new mime types

193.9 1.6.9 / 2015-09-27

- deps: mime-types2.1.7
  - Add new mime types

193.10 1.6.8 / 2015-09-04

- deps: mime-types2.1.6
  - Add new mime types

193.11 1.6.7 / 2015-08-20

- Fix type error when given invalid type to match against
- deps: mime-types2.1.5
  - Add new mime types

193.12 1.6.6 / 2015-07-31

- deps: mime-types2.1.4
  - Add new mime types

193.13 1.6.5 / 2015-07-16

- deps: mime-types2.1.3
  - Add new mime types

193.14 1.6.4 / 2015-07-01

- deps: mime-types2.1.2
  - Add new mime types
- perf: enable strict mode
- perf: remove argument reassignment

193.15 1.6.3 / 2015-06-08

- deps: mime-types2.1.1
  - Add new mime types
- perf: reduce try block size
- perf: remove bitwise operations

193.16 1.6.2 / 2015-05-10

- deps: mime-types2.0.11
  - Add new mime types

193.17 1.6.1 / 2015-03-13

- deps: mime-types2.0.10
  - Add new mime types

193.18 1.6.0 / 2015-02-12

- fix false-positives in `hasBody` `Transfer-Encoding` check
- support wildcard for both type and subtype (`*/*`)

193.19 1.5.7 / 2015-02-09

- fix argument reassignment
- deps: mime-types2.0.9
  - Add new mime types

193.20 1.5.6 / 2015-01-29

- deps: mime-types2.0.8
  - Add new mime types

193.21 1.5.5 / 2014-12-30

- deps: mime-types2.0.7
  - Add new mime types
  - Fix missing extensions
  - Fix various invalid MIME type entries
  - Remove example template MIME types
  - deps: mime-db1.5.0

193.22 1.5.4 / 2014-12-10

- deps: mime-types2.0.4
  - Add new mime types
  - deps: mime-db1.3.0

193.23 1.5.3 / 2014-11-09

- deps: mime-types2.0.3
  - Add new mime types
  - deps: mime-db1.2.0

193.24 1.5.2 / 2014-09-28

- deps: mime-types2.0.2
  - Add new mime types
  - deps: mime-db1.1.0

193.25 1.5.1 / 2014-09-07

- Support Node.js 0.6
- deps: media-typer@0.3.0
- deps: mime-types2.0.1
  - Support Node.js 0.6

193.26 1.5.0 / 2014-09-05

- fix `hasbody` to be true for `content-length: 0`

193.27 1.4.0 / 2014-09-02

- update mime-types

193.28 1.3.2 / 2014-06-24

- use ∼ range on mime-types

193.29 1.3.1 / 2014-06-19

- fix global variable leak

193.30 1.3.0 / 2014-06-19

- improve type parsing
  - invalid media type never matches
  - media type not case-sensitive
  - extra LWS does not affect results

193.31 1.2.2 / 2014-06-19

- fix behavior on unknown type argument

193.32 1.2.1 / 2014-06-03

- switch dependency from `mime` to `mime-types@1.0.0`

193.33 1.2.0 / 2014-05-11

- support suffix matching:
  - `+json` matches `application/vnd+json`
  - `*/vnd+json` matches `application/vnd+json`
  - `application/*+json` matches `application/vnd+json`

193.34 1.1.0 / 2014-04-12

- add non-array values support
- expose internal utilities:
  - `.is()`
  - `.hasBody()`
  - `.normalize()`
  - `.match()`

193.35 1.0.1 / 2014-03-30

- add `multipart` as a shorthand

Chapter 194

type-is

Infer the content-type of a request.

194.0.1 Install

This is a Node.js module available through the npm registry. Installation is done using the npm install command:

```
$ npm install type-is
```

194.1 API

```
var http = require('http')
var typeis = require('type-is')
http.createServer(function (req, res) {
  var istext = typeis(req, ['text/*'])
  res.end('you ' + (istext ? 'sent' : 'did not send') + ' me text')
})
```

194.1.1 typeis(request, types)

Checks if the `request` is one of the `types`. If the request has no body, even if there is a `Content-Type` header, then `null` is returned. If the `Content-Type` header is invalid or does not matches any of the `types`, then `false` is returned. Otherwise, a string of the type that matched is returned.
The `request` argument is expected to be a Node.js HTTP request. The `types` argument is an array of type strings.
Each type in the `types` array can be one of the following:

- A file extension name such as `json`. This name will be returned if matched.
- A mime type such as `application/json`.
- A mime type with a wildcard such as `*/*` or `*/json` or `application/*`. The full mime type will be returned if matched.
- A suffix such as `+json`. This can be combined with a wildcard such as `*/vnd+json` or `application/*+json`. The full mime type will be returned if matched.

Some examples to illustrate the inputs and returned value:

```
// req.headers.content-type = 'application/json'
typeis(req, ['json']) // => 'json'
typeis(req, ['html', 'json']) // => 'json'
typeis(req, ['application/*']) // => 'application/json'
typeis(req, ['application/json']) // => 'application/json'
typeis(req, ['html']) // => false
```

194.1.2 typeis.hasBody(request)

Returns a Boolean if the given `request` has a body, regardless of the `Content-Type` header.

Having a body has no relation to how large the body is (it may be 0 bytes). This is similar to how file existence works. If a body does exist, then this indicates that there is data to read from the Node.js request stream.

```
if (typeis.hasBody(req)) {
  // read the body, since there is one
  req.on('data', function (chunk) {
    // ...
  })
}
```

194.1.3 typeis.is(mediaType, types)

Checks if the `mediaType` is one of the `types`. If the `mediaType` is invalid or does not matches any of the `types`, then `false` is returned. Otherwise, a string of the type that matched is returned.
The `mediaType` argument is expected to be a `media type` string. The `types` argument is an array of type strings.
Each type in the `types` array can be one of the following:

- A file extension name such as `json`. This name will be returned if matched.
- A mime type such as `application/json`.
- A mime type with a wildcard such as `*/*` or `*/json` or `application/*`. The full mime type will be returned if matched.
- A suffix such as `+json`. This can be combined with a wildcard such as `*/vnd+json` or `application/*+json`. The full mime type will be returned if matched.

Some examples to illustrate the inputs and returned value:

```
var mediaType = 'application/json'
typeis.is(mediaType, ['json']) // => 'json'
typeis.is(mediaType, ['html', 'json']) // => 'json'
typeis.is(mediaType, ['application/*']) // => 'application/json'
typeis.is(mediaType, ['application/json']) // => 'application/json'
typeis.is(mediaType, ['html']) // => false
```

194.2 Examples

194.2.1 Example body parser

```
var express = require('express')
var typeis = require('type-is')
var app = express()
app.use(function bodyParser (req, res, next) {
  if (!typeis.hasBody(req)) {
    return next()
  }
  switch (typeis(req, ['urlencoded', 'json', 'multipart'])) {
    case 'urlencoded':
      // parse urlencoded body
      throw new Error('implement urlencoded body parsing')
    case 'json':
      // parse json body
      throw new Error('implement json body parsing')
    case 'multipart':
      // parse multipart body
      throw new Error('implement multipart body parsing')
    default:
      // 415 error code
      res.statusCode = 415
      res.end()
      break
  }
})
```

194.3 License

[MIT](LICENSE)

Chapter 195

1.0.0 / 2015-06-14

- Initial release

Chapter 196

unpipe

Unpipe a stream from all destinations.

196.1 Installation

```
$ npm install unpipe
```

196.2 API

```
var unpipe = require('unpipe')
```

196.2.1 unpipe(stream)

Unpipes all destinations from a given stream. With stream 2+, this is equivalent to `stream.unpipe()`. When used with streams 1 style streams (typically Node.js 0.8 and below), this module attempts to undo the actions done in `stream.pipe(dest)`.

196.3 License

[MIT](LICENSE)

Chapter 197

utils-merge

Merges the properties from a source object into a destination object.

197.1 Install

```
$ npm install utils-merge
```

197.2 Usage

```
var a = { foo: 'bar' }
  , b = { bar: 'baz' };
merge(a, b);
// => { foo: 'bar', bar: 'baz' }
```

197.3 License

The MIT License
Copyright (c) 2013-2017 Jared Hanson < http://jaredhanson.net/>

Chapter 198

validator.js

![Backers on Open Collective](https://opencollective.com/validatorjs/backers/badge.svg) ![Sponsors on Open Collective](https://opencollective.com/validatorjs/sponsors/badge.svg)
A library of string validators and sanitizers.

198.1 Strings only

This library validates and sanitizes strings only.
If you're not sure if your input is a string, coerce it using ‘input + ”`. Passing anything other than a string will result in an error.

198.2 Installation and Usage

198.2.1 Server-side usage

Install the library with `npm install validator`

198.2.1.1 No ES6

```
var validator = require('validator');
validator.isEmail('foo@bar.com'); //=> true
```

198.2.1.2 ES6

```
import validator from 'validator';
```

Or, import only a subset of the library:

```
import isEmail from 'validator/lib/isEmail';
```

198.2.1.3 Tree-shakeable ES imports

```
import isEmail from 'validator/es/lib/isEmail';
```

198.2.2 Client-side usage

The library can be loaded either as a standalone script, or through an `AMD`-compatible loader

```
<script type="text/javascript" src="validator.min.js"></script>
<script type="text/javascript">
  validator.isEmail('foo@bar.com'); //=> true
</script>
```

The library can also be installed through `bower`

```
$ bower install validator-js
```

CDN

```
<script src="https://unpkg.com/validator@latest/validator.min.js"></script>
```

198.3 Validators

Here is a list of the validators currently available.

Validator	Description
contains(str, seed [, options])	check if the string contains the seed. `options` is an object that defaults to `{ ignoreCase: false, minOccurrences: 1 }`. Options: `ignoreCase`: Ignore case when doing comparison, default false. `minOccurences`: Minimum number of occurrences for the seed in the string. Defaults to 1.
equals(str, comparison)	check if the string matches the comparison.
isAbaRouting(str)	check if the string is an ABA routing number for US bank account / cheque.
isAfter(str [, options])	check if the string is a date that is after the specified date. `options` is an object that defaults to `{ comparison↩ Date: Date().toString() }`. **Options:** `comparisonDate`: Date to compare to. Defaults to `Date().toString()` (now).
isAlpha(str [, locale, options])	check if the string contains only letters (a-zA-Z). `locale` is one of ‘['ar', 'ar-AE', 'ar-BH', 'ar-DZ', 'ar-EG', 'ar-IQ', 'ar-JO', 'ar-KW', 'ar-LB', 'ar-LY', 'ar-MA', 'ar-QA', 'ar-QM', 'ar-SA', 'ar-SD', 'ar-SY', 'ar-TN', 'ar-YE', 'bg-BG', 'bn', 'cs-CZ', 'da-DK', 'de-DE', 'el-GR', 'en-AU', 'en-GB', 'en-HK', 'en-IN', 'en-NZ', 'en-US', 'en-ZA', 'en-ZM', 'eo', 'es-ES', 'fa-IR', 'fi-FI', 'fr-CA', 'fr-FR', 'he', 'hi-IN', 'hu-HU', 'it-IT', 'kk-KZ', 'ko-KR', 'ja-JP', 'ku-IQ', 'nb-NO', 'nl-NL', 'nn-NO', 'pl-PL', 'pt-BR', 'pt-PT', 'ru-RU', 'si-LK', 'sl-SI', 'sk-SK', 'sr-RS', 'sr-RS@latin', 'sv-SE', 'th-TH', 'tr-TR', 'uk-UA']`and defaults to`en-US`. Locale list is`validator.is↩ AlphaLocales`.options`is an optional object that can be supplied with the following key(s)↩ :`ignore`which can either be a String or Reg↩ Exp of characters to be ignored e.g. " -" will ignore spaces and -'s. \ilinebr </td> </tr> <tr class="markdownTableRow↩ Even"> <td class="markdownTableBodyNone"> **isAlphanumeric(str [, locale, options])** \ilinebr </td> <td class="markdown↩ TableBodyNone"> check if the string contains only letters and numbers (a-z↩ A-Z0-9).<br/><br/>`locale`is one of`['ar', 'ar-AE', 'ar-BH', 'ar-DZ', 'ar-EG', 'ar-IQ', 'ar-JO', 'ar-KW', 'ar-LB', 'ar-↩ LY', 'ar-MA', 'ar-QA', 'ar-QM', 'ar-SA', 'ar-SD', 'ar-SY', 'ar-TN', 'ar-YE', 'bn', 'bg-BG', 'cs-CZ', 'da-DK', 'de-DE', 'el-GR', 'en-AU', 'en-GB', 'en-HK', 'en-IN', 'en-NZ', 'en-US', 'en-ZA', 'en-ZM', 'eo', 'es-ES', 'fa-IR', 'fi-FI', 'fr-CA', 'fr-FR', 'he', 'hi-IN', 'hu-HU', 'it-IT', 'kk-KZ', 'ko-KR', 'ja-JP','ku-IQ', 'nb-NO', 'nl-NL', 'nn-NO', 'pl-PL', 'pt-BR', 'pt-PT', 'ru-RU', 'si-LK', 'sl-SI', 'sk-SK', 'sr-RS', 'sr-↩ RS@latin', 'sv-SE', 'th-TH', 'tr-TR', 'uk-UA']`) and defaults to`en-US`. Locale list is`validator.isAlphanumeric↩ Locales`.options`is an optional object that can be supplied with the following key(s):`ignore` which can either be a String or RegExp of characters to be ignored e.g. " -" will ignore spaces and -'s.
isAscii(str)	check if the string contains ASCII chars only.

Validator	Description
isBase32(str [, options])	check if the string is base32 encoded. `options` is optional and defaults to `{ crockford: false }`. When `crockford` is true it tests the given base32 encoded string using `Crockford's base32 alternative`.
isBase58(str)	check if the string is base58 encoded.
isBase64(str [, options])	check if the string is base64 encoded. `options` is optional and defaults to `{ urlSafe: false }` when `urlSafe` is true it tests the given base64 encoded string is `url safe`.
isBefore(str [, date])	check if the string is a date that is before the specified date.
isBIC(str)	check if the string is a BIC (Bank Identification Code) or SWIFT code.
isBoolean(str [, options])	check if the string is a boolean. `options` is an object which defaults to `{ loose: false }`. If `loose` is set to false, the validator will strictly match ['true', 'false', '0', '1']. If `loose` is set to true, the validator will also match 'yes', 'no', and will match a valid boolean string of any case. (e.↩ g.: ['true', 'True', 'TRUE']).
isBtcAddress(str)	check if the string is a valid BTC address.
isByteLength(str [, options])	check if the string's length (in UTF-8 bytes) falls in a range. `options` is an object which defaults to `{ min: 0, max: undefined }`.
isCreditCard(str [, options])	check if the string is a credit card number. `options` is an optional object that can be supplied with the following key(s): `provider` is an optional key whose value should be a string, and defines the company issuing the credit card. Valid values include '['amex', 'dinersclub', 'discover', 'jcb', 'mastercard', 'unionpay', 'visa']`or blank will check for any provider. \ilinebr </td> </tr> <tr class="markdownTableRow↩ Odd"> <td class="markdownTableBodyNone"> **isCurrency(str [, options])** \ilinebr </td> <td class="markdownTableBodyNone"> check if the string is a valid currency amount.<br/><br/>`options`is an object which defaults to`{ symbol: '$', require_symbol: false, allow_↩ space_after_symbol: false, symbol_after_digits: false, allow↩ _negatives: true, parens_for_negatives: false, negative_↩ sign_before_digits: false, negative_sign_after_digits: false, allow_negative_sign_placeholder: false, thousands_separator: ',', decimal_separator: '.', allow_decimal: true, require_decimal: false, digits_after_decimal: [2], allow_space_after_digits: false `}.<br/>**Note:** The array`digits_after_decimal` is filled with the exact number of digits allowed not a range, for example a range 1 to 3 will be given as [1, 2, 3].
isDataURI(str)	check if the string is a `data uri format`.

Validator	Description
isDate(str [, options])	check if the string is a valid date. e.g. [2002-07-15, new Date()]. options is an object which can contain the keys format, strictMode and/or delimiters. format is a string and defaults to YYYY/MM/DD. strictMode is a boolean and defaults to false. If strictMode is set to true, the validator will reject strings different from format. delimiters is an array of allowed date delimiters and defaults to '['/', '-']. \ilinebr </td> </tr> <tr class="markdownTableRowEven"> <td class="markdownTableBodyNone"> **is↩ Decimal(str [, options])** \ilinebr </td> <td class="markdownTableBodyNone"> check if the string represents a decimal number, such as 0.1, .3, 1.1, 1.00003, 4.0, etc.<br/><br/>optionsis an object which defaults to{force_decimal: false, decimal_digits: '1,', locale: 'en-US'}.<br/><br/>localedetermines the decimal separator and is one of['ar', 'ar-AE', 'ar-↩ BH', 'ar-DZ', 'ar-EG', 'ar-IQ', 'ar-JO', 'ar-KW', 'ar-LB', 'ar-LY', 'ar-MA', 'ar-QA', 'ar-QM', 'ar-SA', 'ar-SD', 'ar-SY', 'ar-TN', 'ar-YE', 'bg-BG', 'cs-CZ', 'da-DK', 'de-DE', 'el-GR', 'en-AU', 'en-GB', 'en-↩ HK', 'en-IN', 'en-NZ', 'en-US', 'en-ZA', 'en-ZM', 'es-ES', 'fa', 'fa-AF', 'fa-IR', 'fr-FR', 'fr-CA', 'hu-HU', 'id-ID', 'it-IT', 'ku-IQ', 'nb-NO', 'nl-↩ NL', 'nn-NO', 'pl-PL', 'pl-Pl', 'pt-BR', 'pt-PT', 'ru-RU', 'sl-SI', 'sr-RS', 'sr-RS@latin', 'sv-SE', 'tr-TR', 'uk-UA', 'vi-VN'].<br/>**↩ Note:**decimal_digits' is given as a range like '1,3', a specific value like '3' or min like '1,'.
isDivisibleBy(str, number)	check if the string is a number that is divisible by another.
isEAN(str)	check if the string is an EAN (European Article Number).

Validator	Description
isEmail(str [, options])	check if the string is an email. options is an object which defaults to '{ allow_display_name: false, require_display_name: false, allow_utf8_local_part: true, require_tld: true, allow_ip_domain: false, allow_underscores: false, domain_specific_validation: false, blacklisted_chars: '', host_blacklist: [] }. If allow_display_name is set to true, the validator will also match Display Name <email-address>. If require_display_name is set to true, the validator will reject strings without the format Display Name <email-address>. If allow_utf8_local_part' is set to false, the validator will not allow any non-English UTF8 character in email address' local part. If require_tld is set to false, email addresses without a TLD in their domain will also be matched. If ignore_max_length is set to true, the validator will not check for the standard max length of an email. If allow_ip_domain is set to true, the validator will allow IP addresses in the host part. If domain↩_specific_validation is true, some additional validation will be enabled, e.g. disallowing certain syntactically valid email addresses that are rejected by Gmail. If blacklisted_↩chars receives a string, then the validator will reject emails that include any of the characters in the string, in the name part. If host_blacklist is set to an array of strings and the part of the email after the @ symbol matches one of the strings defined in it, the validation fails. If host_whitelist is set to an array of strings and the part of the email after the @ symbol matches none of the strings defined in it, the validation fails.
isEmpty(str [, options])	check if the string has a length of zero. options is an object which defaults to { ignore_↩whitespace: false }.
isEthereumAddress(str)	check if the string is an Ethereum address. Does not validate address checksums.

Validator	Description
isFloat(str [, options])	check if the string is a float. `options` is an object which can contain the keys `min`, `max`, `gt`, and/or `lt` to validate the float is within boundaries (e.g. `{ min: 7.22, max: 9.55 }`) it also has `locale` as an option. `min` and `max` are equivalent to 'greater or equal' and 'less or equal', respectively while `gt` and `lt` are their strict counterparts. `locale` determines the decimal separator and is one of '['ar', 'ar-AE', 'ar-BH', 'ar-DZ', 'ar-EG', 'ar-IQ', 'ar-JO', 'ar-KW', 'ar-LB', 'ar-LY', 'ar-MA', 'ar-QA', 'ar-QM', 'ar-SA', 'ar-SD', 'ar-SY', 'ar-TN', 'ar-YE', 'bg-BG', 'cs-CZ', 'da-DK', 'de-DE', 'en-AU', 'en-GB', 'en-HK', 'en-IN', 'en-NZ', 'en-US', 'en-ZA', 'en-ZM', 'es-ES', 'fr-CA', 'fr-FR', 'hu-HU', 'it-IT', 'nb-NO', 'nl-NL', 'nn-NO', 'pl-PL', 'pt-BR', 'pt-PT', 'ru-RU', 'sl-SI', 'sr-RS', 'sr-RS@latin', 'sv-SE', 'tr-TR', 'uk-UA']`. Locale list is`validator.isFloatLocales`. \ilinebr </td> </tr> <tr class="markdownTableRowOdd"> <td class="markdownTableBodyNone"> **isFQDN(str [, options])** \ilinebr </td> <td class="markdownTableBodyNone"> check if the string is a fully qualified domain name (e.g. domain.com).<br/><br/>`options`is an object which defaults to`{ require_tld: true, allow_underscores: false, allow_trailing_dot: false, allow_numeric_tld: false, allow_wildcard: false, ignore_max_length: false }`. If`allow_wildcard`is set to true, the validator will allow domain starting with`*.`(e.g.`*.example.com`or`*.shop.example.com`). \ilinebr </td> </tr> <tr class="markdownTableRowEven"> <td class="markdownTableBodyNone"> **isFreightContainerID(str)** \ilinebr </td> <td class="markdownTableBodyNone"> alias for`isISO6346`, check if the string is a valid ISO 6346 shipping container identification.
isFullWidth(str)	check if the string contains any full-width chars.
isHalfWidth(str)	check if the string contains any half-width chars.
isHash(str, algorithm)	check if the string is a hash of type algorithm. Algorithm is one of '['crc32', 'crc32b', 'md4', 'md5', 'ripemd128', 'ripemd160', 'sha1', 'sha256', 'sha384', 'sha512', 'tiger128', 'tiger160', 'tiger192']`.
isHexadecimal(str)	check if the string is a hexadecimal number.
isHexColor(str)	check if the string is a hexadecimal color.
isHSL(str)	check if the string is an HSL (hue, saturation, lightness, optional alpha) color based on CSS Colors Level 4 specification. Comma-separated format supported. Space-separated format supported with the exception of a few edge cases (ex: `hsl(200grad+.1%62%/1)`).

Validator	Description
isIBAN(str, [, options])	check if the string is an IBAN (International Bank Account Number). options is an object which accepts two attributes: whitelist: where you can restrict IBAN codes you want to receive data from and blacklist: where you can remove some of the countries from the current list. For both you can use an array with the following values '['AD','AE','AL','AT','AZ','BA','BE','BG','BH','BR','BY','CH','CR','CY','CZ','DE','DK','DO',' \ilinebr </td> </tr> <tr class="markdownTableRowEven"> <td class="markdownTableBodyNone"> **isIdentityCard(str [, locale])** \ilinebr </td> <td class="markdownTableBodyNone"> check if the string is a valid identity card code.<br/><br/>localeis one of['LK', 'PL', 'ES', 'FI', 'IN', 'IT', 'IR', 'MZ', 'NO', 'TH', 'zh-TW', 'he-IL', 'ar-LY', 'ar-TN', 'zh-CN', 'zh-HK']OR'any''. If 'any' is used, function will check if any of the locales match. Defaults to 'any'.
isIMEI(str [, options]))	check if the string is a valid IMEI number. IMEI should be of format ############### or ##-######-######-#. options is an object which can contain the keys allow_hyphens. Defaults to first format. If allow_hyphens is set to true, the validator will validate the second format.
isIn(str, values)	check if the string is in an array of allowed values.
isInt(str [, options])	check if the string is an integer. options is an object which can contain the keys min and/or max to check the integer is within boundaries (e.g. { min: 10, max: 99 }). options can also contain the key allow_leading_zeroes, which when set to false will disallow integer values with leading zeroes (e.g. { allow_leading_zeroes: false }). Finally, options can contain the keys gt and/or lt which will enforce integers being greater than or less than, respectively, the value provided (e.g. {gt: 1, lt: 4} for a number between 1 and 4).
isIP(str [, version])	check if the string is an IP (version 4 or 6).
isIPRange(str [, version])	check if the string is an IP Range (version 4 or 6).
isISBN(str [, options])	check if the string is an ISBN. options is an object that has no default. **Options:** version: ISBN version to compare to. Accepted values are '10' and '13'. If none provided, both will be tested.
isISIN(str)	check if the string is an ISIN (stock/security identifier).
isISO6346(str)	check if the string is a valid ISO 6346 shipping container identification.
isISO6391(str)	check if the string is a valid ISO 639-1 language code.

Validator	Description
isISO8601(str [, options])	check if the string is a valid ISO 8601 date. options is an object which defaults to { strict: false, strictSeparator: false }. If strict is true, date strings with invalid dates like 2009-02-29 will be invalid. If strictSeparator is true, date strings with date and time separated by anything other than a T will be invalid.
isISO31661Alpha2(str)	check if the string is a valid ISO 3166-1 alpha-2 officially assigned country code.
isISO31661Alpha3(str)	check if the string is a valid ISO 3166-1 alpha-3 officially assigned country code.
isISO4217(str)	check if the string is a valid ISO 4217 officially assigned currency code.
isISRC(str)	check if the string is an ISRC.
isISSN(str [, options])	check if the string is an ISSN. options is an object which defaults to { case_↩ sensitive: false, require_hyphen: false }. If case_sensitive is true, ISSNs with a lowercase "x'as the check digit are rejected. \ilinebr </td> </tr> <tr class="markdownTable↩ RowEven"> <td class="markdownTableBody↩ None"> **isJSON(str [, options])** \ilinebr </td> <td class="markdownTableBodyNone"> check if the string is valid JSON (note↩ : uses JSON.parse).<br/><br/>optionsis an object which defaults to{ allow_primitives: false }. Ifallow_primitives' is true, the primitives 'true', 'false' and 'null' are accepted as valid JSON values.
isJWT(str)	check if the string is valid JWT token.
isLatLong(str [, options])	check if the string is a valid latitude-longitude coordinate in the format lat,long or lat, long. options is an object that defaults to { checkDMS: false }. Pass checkDMS as true to validate DMS(degrees, minutes, and seconds) latitude-longitude format.
isLength(str [, options])	check if the string's length falls in a range. options is an object which defaults to { min: 0, max: undefined }. Note: this function takes into account surrogate pairs.
isLicensePlate(str, locale)	check if the string matches the format of a country's license plate. locale is one of '['cs-CZ', 'de-DE', 'de-LI', 'en-IN', 'en-PK', 'es-AR', 'hu-HU', 'pt-BR', 'pt-PT', 'sq-AL', 'sv-SE']or'any''.
isLocale(str)	check if the string is a locale.
isLowercase(str)	check if the string is lowercase.
isLuhnNumber(str)	check if the string passes the Luhn algorithm check.

Validator	Description
isMACAddress(str [, options])	check if the string is a MAC address. options is an object which defaults to { no_separators: false }. If no_separators is true, the validator will allow MAC addresses without separators. Also, it allows the use of hyphens, spaces or dots e.g. '01 02 03 04 05 ab', '01-02-03-04-05-ab' or '0102.0304.05ab'. The options also allow a eui property to specify if it needs to be validated against EUI-48 or EUI-64. The accepted values of eui are: 48, 64.
isMagnetURI(str)	check if the string is a Magnet URI format.
isMailtoURI(str, [, options])	check if the string is a Mailto URI format. options is an object of validating emails inside the URI (check isEmails options for details).
isMD5(str)	check if the string is a MD5 hash. Please note that you can also use the ‘isHash(str, 'md5’)` function. Keep in mind that MD5 has some collision weaknesses compared to other algorithms (e.g., SHA).
isMimeType(str)	check if the string matches to a valid MIME type format.
isMobilePhone(str [, locale [, options]])	check if the string is a mobile phone number, locale is either an array of locales (e.g. ‘['sk-SK’, 'sr-RS']) OR one of['am-Am', 'ar-AE', 'ar-BH', 'ar-DZ', 'ar-EG', 'ar-EH', 'ar-IQ', 'ar-JO', 'ar-KW', 'ar-PS', 'ar-SA', 'ar-SD', 'ar-SY', 'ar-TN', 'ar-YE', 'az-AZ', 'az-LB', 'az-LY', 'be-BY', 'bg-BG', 'bn-BD', 'bs-BA', 'ca-AD', 'cs-CZ', 'da-DK', 'de-AT', 'de-CH', 'de-DE', 'de-LU', 'dv-MV', 'dz-BT', 'el-CY', 'el-GR', 'en-AG', 'en-AI', 'en-AU', 'en-BM', 'en-BS', 'en-BW', 'en-CA', 'en-GB', 'en-GG', 'en-GH', 'en-GY', 'en-HK', 'en-IE', 'en-IN', 'en-JM', 'en-KE', 'en-KI', 'en-KN', 'en-LS', 'en-MO', 'en-MT', 'en-MU', 'en-MW', 'en-NG', 'en-NZ', 'en-PG', 'en-PH', 'en-PK', 'en-RW', 'en-SG', 'en-SL', 'en-SS', 'en-TZ', 'en-UG', 'en-US', 'en-ZA', 'en-ZM', 'en-ZW', 'es-AR', 'es-BO', 'es-CL', 'es-CO', 'es-CR', 'es-CU', 'es-DO', 'es-EC', 'es-ES', 'es-HN', 'es-MX', 'es-NI', 'es-PA', 'es-PE', 'es-PY', 'es-SV', 'es-UY', 'es-VE', 'et-EE', 'fa-AF', 'fa-IR', 'fi-FI', 'fj-FJ', 'fo-FO', 'fr-BE', 'fr-BF', 'fr-BJ', 'fr-CD', 'fr-CF', 'fr-FR', 'fr-GF', 'fr-GP', 'fr-MQ', 'fr-PF', 'fr-RE', 'fr-WF', 'ga-IE', 'he-IL', 'hu-HU', 'id-ID', 'ir-IR', 'it-IT', 'it-SM', 'ja-JP', 'ka-GE', 'kk-KZ', 'kl-GL', 'ko-KR', 'ky-KG', 'lt-LT', 'mg-MG', 'mn-MN', 'ms-MY', 'my-MM', 'mz-MZ', 'nb-NO', 'ne-NP', 'nl-AW', 'nl-BE', 'nl-NL', 'nn-NO', 'pl-PL', 'pt-AO', 'pt-BR', 'pt-PT', 'ro-Md', 'ro-RO', 'ru-RU', 'si-LK', 'sk-SK', 'sl-SI', 'so-SO', 'sq-AL', 'sr-RS', 'sv-SE', 'tg-TJ', 'th-TH', 'tk-TM', 'tr-TR', 'uk-UA', 'uz-UZ', 'vi-VN', 'zh-CN', 'zh-HK', 'zh-MO', 'zh-TW']OR defaults to'any''. If 'any’ or a falsey value is used, function will check if any of the locales match). options is an optional object that can be supplied with the following keys: strictMode, if this is set to true, the mobile phone number must be supplied with the country code and therefore must start with +. Locale list is validator.isMobilePhoneLocales.
isMongoId(str)	check if the string is a valid hex-encoded representation of a MongoDB ObjectId.
isMultibyte(str)	check if the string contains one or more multibyte chars.

Validator	Description
isNumeric(str [, options])	check if the string contains only numbers. `options` is an object which defaults to `{ no_symbols: false }` it also has `locale` as an option. If `no_symbols` is true, the validator will reject numeric strings that feature a symbol (e.g. +, −, or .). `locale` determines the decimal separator and is one of '['ar', 'ar-AE', 'ar-BH', 'ar-DZ', 'ar-EG', 'ar-IQ', 'ar-JO', 'ar-KW', 'ar-LB', 'ar-LY', 'ar-MA', 'ar-QA', 'ar-QM', 'ar-SA', 'ar-SD', 'ar-SY', 'ar-TN', 'ar-YE', 'bg-BG', 'cs-CZ', 'da-DK', 'de-DE', 'en-AU', 'en-↩GB', 'en-HK', 'en-IN', 'en-NZ', 'en-US', 'en-ZA', 'en-ZM', 'es-ES', 'fr-FR', 'fr-CA', 'hu-HU', 'it-IT', 'nb-NO', 'nl-NL', 'nn-NO', 'pl-PL', 'pt-BR', 'pt-PT', 'ru-RU', 'sl-SI', 'sr-RS', 'sr-RS@latin', 'sv-SE', 'tr-TR', 'uk-UA']`.
isOctal(str)	check if the string is a valid octal number.
isPassportNumber(str, countryCode)	check if the string is a valid passport number. `countryCode` is one of '['AM', 'AR', 'AT', 'AU', 'AZ', 'BE', 'BG', 'BY', 'BR', 'CA', 'CH', 'CN', 'CY', 'CZ', 'DE', 'DK', 'DZ', 'EE', 'ES', 'FI', 'FR', 'GB', 'GR', 'HR', 'HU', 'IE', 'IN', 'IR', 'ID', 'IS', 'IT', 'JM', 'JP', 'KR', 'KZ', 'LI', 'LT', 'LU', 'LV', 'LY', 'MT', 'MX', 'MY', 'MZ', 'NL', 'NZ', 'PH', 'PK', 'PL', 'PT', 'RO', 'RU', 'SE', 'SL', 'SK', 'TH', 'TR', 'UA', 'US', 'ZA']`.
isPort(str)	check if the string is a valid port number.
isPostalCode(str, locale)	check if the string is a postal code. `locale` is one of '['AD', 'AT', 'AU', 'AZ', 'BA', 'BE', 'BG', 'BR', 'BY', 'CA', 'CH', 'CN', 'CZ', 'DE', 'DK', 'DO', 'DZ', 'EE', 'ES', 'FI', 'FR', 'GB', 'GR', 'HR', 'HT', 'HU', 'ID', 'IE', 'IL', 'IN', 'IR', 'IS', 'IT', 'JP', 'KE', 'KR', 'LI', 'LK', 'LT', 'LU', 'LV', 'MG', 'MT', 'MX', 'MY', 'NL', 'NO', 'NP', 'NZ', 'PL', 'PR', 'PT', 'RO', 'RU', 'SA', 'SE', 'SG', 'SI', 'SK', 'TH', 'TN', 'TW', 'UA', 'US', 'ZA', 'ZM']`OR`'any''. If 'any' is used, function will check if any of the locales match. Locale list is `validator.isPostalCodeLocales`.
isRFC3339(str)	check if the string is a valid `RFC 3339` date.
isRgbColor(str [, includePercentValues])	check if the string is a rgb or rgba color. `includePercentValues` defaults to `true`. If you don't want to allow to set `rgb` or `rgba` values with percents, like `rgb(5%,5%,5%)`, or `rgba(90%,90%,90%,.3)`, then set it to false.
isSemVer(str)	check if the string is a Semantic Versioning Specification (Sem↩Ver).
isSurrogatePair(str)	check if the string contains any surrogate pairs chars.
isUppercase(str)	check if the string is uppercase.
isSlug(str)	check if the string is of type slug.

Validator	Description
isStrongPassword(str [, options])	check if the string can be considered a strong password or not. Allows for custom requirements or scoring rules. If `return↩ Score` is true, then the function returns an integer score for the password rather than a boolean. Default options: `{ minLength: 8, minLowercase: 1, min↩ Uppercase: 1, minNumbers: 1, minSymbols: 1, returnScore: false, pointsPerUnique↩ : 1, pointsPerRepeat: 0.5, pointsFor↩ ContainingLower: 10, pointsForContaining↩ Upper: 10, pointsForContainingNumber: 10, pointsForContainingSymbol: 10 }`

Validator	Description
isTime(str [, options])	check if the string is a valid time e.g. [23:01:59, new Date().toLocaleTimeString()]. options is an object which can contain the keys hour↩ Format or mode. hourFormat is a key and defaults to ''hour24'.<br/><br/>modeis a key and defaults to'default'. <br/><br/>hourFomatcan contain the values'hour12'or'hour24','hour24'will validate hours in 24 format and'hour12'will validate hours in 12 format. <br/><br/>modecan contain the values'default'or'with↩ Seconds','default'will validateHH:MMformat,'with↩ Seconds'will validate theHH:MM:SSformat. \ilinebr </td> </tr> <tr class="markdown↩ TableRowOdd"> <td class="markdown↩ TableBodyNone"> **isTaxID(str, locale)** \ilinebr </td> <td class="markdownTable↩ BodyNone"> check if the string is a valid Tax Identification Number. Default locale isen-US.<br/><br/>More info about exact TIN support can be found insrc/lib/isTaxID.↩ js.<br/><br/>Supported locales:['bg-BG', 'cs-CZ', 'de-AT', 'de-DE', 'dk-DK', 'el-CY', 'el-GR', 'en-CA', 'en-GB', 'en-IE', 'en-US', 'es-AR', 'es-ES', 'et-EE', 'fi-FI', 'fr-BE', 'fr-↩ CA', 'fr-FR', 'fr-LU', 'hr-HR', 'hu-HU', 'it-IT', 'lb-LU', 'lt-LT', 'lv-LV', 'mt-MT', 'nl-BE', 'nl-NL', 'pl-PL', 'pt-BR', 'pt-PT', 'ro-↩ RO', 'sk-SK', 'sl-SI', 'sv-SE', 'uk-UA']. \ilinebr </td> </tr> <tr class="markdownTableRowEven"> <td class="markdownTableBodyNone"> **is↩ URL(str [, options])** \ilinebr </td> <td class="markdownTableBodyNone"> check if the string is a URL.<br/><br/>optionsis an object which defaults to{ protocols: ['http','https','ftp'], require_tld: true, require_protocol: false, require_host↩ : true, require_port: false, require_valid_protocol: true, allow_underscores: false, host_whitelist: false, host_↩ blacklist: false, allow_trailing_dot: false, allow_protocol_↩ relative_urls: false, allow_fragments: true, allow_query_↩ components: true, disallow_auth: false, validate_length↩ : true }.<br/><br/>require_protocol- if set to true isURL will return false if protocol is not present in the URL.<br/>require_↩ valid_protocol- isURL will check if the URL's protocol is present in the protocols option.<br/>protocols- valid protocols can be modified with this option.<br/>require_↩ host- if set to false isURL will not check if host is present in the URL.<br/>require↩ _port- if set to true isURL will check if port is present in the URL.<br/>allow_↩ protocol_relative_urls- if set to true protocol relative URLs will be allowed.<br/>allow_↩ fragments- if set to false isURL will return false if fragments are present.<br/>allow↩ _query_components- if set to false isURL will return false if query components are present.<br/>validate_length` - if set to false isURL will
	skip string length validation (2083 characters is IE max URL length).

Validator	Description
isUUID(str [, version])	check if the string is a UUID (version 1, 2, 3, 4 or 5).
isVariableWidth(str)	check if the string contains a mixture of full and half-width chars.
isVAT(str, countryCode)	check if the string is a `valid VAT number` if validation is available for the given country code matching `ISO 3166-1 alpha-2`. `countryCode` is one of '['AL', 'AR', 'AT', 'AU', 'BE', 'BG', 'BO', 'BR', 'BY', 'CA', 'CH', 'CL', 'CO', 'CR', 'CY', 'CZ', 'DE', 'DK', 'DO', 'EC', 'EE', 'EL', 'ES', 'FI', 'FR', 'GB', 'GT', 'HN', 'HR', 'HU', 'ID', 'IE', 'IL', 'IN', 'IS', 'IT', 'KZ', 'LT', 'LU', 'LV', 'MK', 'MT', 'MX', 'NG', 'NI', 'NL', 'NO', 'NZ', 'PA', 'PE', 'PH', 'PL', 'PT', 'PY', 'RO', 'RS', 'RU', 'SA', 'SE', 'SI', 'SK', 'SM', 'SV', 'TR', 'UA', 'UY', 'UZ', 'VE']`. \ilinebr </td> </tr> <tr class="markdownTableRowEven"> <td class="markdownTableBodyNone"> **is↩ Whitelisted(str, chars)** \ilinebr </td> <td class="markdownTableBodyNone"> check if the string consists only of characters that appear in the whitelist`chars`. \ilinebr </td> </tr> <tr class="markdownTableRow↩ Odd"> <td class="markdownTableBodyNone"> **matches(str, pattern [, modifiers])** \ilinebr </td> <td class="markdownTable↩ BodyNone"> check if the string matches the pattern.<br/><br/>Either`matches('foo', /foo/i)`or`matches('foo', 'foo', 'i')`.

198.4 Sanitizers

Here is a list of the sanitizers currently available.

Sanitizer	Description
blacklist(input, chars)	remove characters that appear in the blacklist. The characters are used in a RegExp and so you will need to escape some chars, e.g. 'blacklist(input, '\[\]')`. \ilinebr </td> </tr> <tr class="markdown↩ TableRowEven"> <td class="markdownTable↩ BodyNone"> **escape(input)** \ilinebr </td> <td class="markdownTableBodyNone"> replace`<, >, &, ', "` and `/` with HTML entities.
ltrim(input [, chars])	trim characters from the left-side of the input.

Sanitizer	Description
∗∗normalizeEmail(email [, options])∗∗	canonicalize an email address. (This doesn't validate that the input is an email, if you want to validate the email use isEmail beforehand). \`options\` is an object with the following keys and default values: • ∗all_lowercase: true∗ - Transforms the local part (before the @ symbol) of all email addresses to lowercase. Please note that this may violate RFC 5321, which gives providers the possibility to treat the local part of email addresses in a case sensitive way (although in practice most - yet not all - providers don't). The domain part of the email address is always lowercased, as it is case insensitive per RFC 1035. • ∗gmail_lowercase: true∗ - Gmail addresses are known to be case-insensitive, so this switch allows lowercasing them even when ∗all_lowercase∗ is set to false. Please note that when ∗all_lowercase∗ is true, Gmail addresses are lowercased regardless of the value of this setting. • ∗gmail_remove_dots: true∗: Removes dots from the local part of the email address, as Gmail ignores them (e.g. "john.doe" and "johndoe" are considered equal). • ∗gmail_remove_subaddress: true∗: Normalizes addresses by removing "sub-addresses", which is the part following a "+" sign (e.g. "foo+bar@gmail.com" becomes "foo@gmail.com"). • ∗gmail_convert_googlemaildotcom: true∗: Converts addresses with domain @googlemail.com to @gmail.com, as they're equivalent. • ∗outlookdotcom_lowercase: true∗ - Outlook.com addresses (including Windows Live and Hotmail) are known to be case-insensitive, so this switch allows lowercasing them even when ∗all_lowercase∗ is set to false. Please note that when ∗all← _lowercase∗ is true, Outlook.com addresses are lowercased regardless of the value of this setting. • ∗outlookdotcom_remove_subaddress: true∗: Normalizes addresses by removing "sub-addresses", which is the part following a "+" sign (e.g. "foo+bar@outlook.com" becomes "foo@outlook.com"). • ∗yahoo_lowercase: true∗ - Yahoo Mail addresses are known to be case-insensitive, so this switch allows lowercasing them even when ∗all_lowercase∗ is set to false. Please note that when ∗all_lowercase∗ is true, Yahoo Mail addresses are lowercased regardless of the value of this setting. • ∗yahoo_remove_subaddress: true∗: Normalizes addresses by removing "sub-addresses", which is the part following a "-" sign (e.g. "foo-bar@yahoo.com" becomes "foo@yahoo.com"). • ∗icloud_lowercase: true∗ - iCloud addresses (including MobileMe) are known to be case-insensitive, so this switch allows lowercasing them even when ∗all_lowercase∗ is set to false. Please note that when ∗all_lowercase∗ is true, i← Cloud addresses are lowercased regardless of the value of this setting. • ∗icloud_remove_subaddress: true∗: Normalizes addresses by removing "sub-addresses", which is the part following a "+" sign (e.g. "foo+bar@icloud.com" becomes "foo@icloud.com").

Sanitizer	Description
∗∗rtrim(input [, chars])∗∗	trim characters from the right-side of the input.
∗∗stripLow(input [, keep_new_lines])∗∗	remove characters with a numerical value < 32 and 127, mostly control characters. If `keep_new_lines` is `true`, newline characters are preserved (`<br>` and `\r`, hex `0xA` and `0xD`). Unicode-safe in JavaScript.
∗∗toBoolean(input [, strict])∗∗	convert the input string to a boolean. Everything except for `'0'`, `'false'` and `''` returns `true`. In strict mode only `'1'` and `'true'` return `true`.
∗∗toDate(input)∗∗	convert the input string to a date, or `null` if the input is not a date.
∗∗toFloat(input)∗∗	convert the input string to a float, or `NaN` if the input is not a float.
∗∗toInt(input [, radix])∗∗	convert the input string to an integer, or `NaN` if the input is not an integer.
∗∗trim(input [, chars])∗∗	trim characters (whitespace by default) from both sides of the input.
∗∗unescape(input)∗∗	replace HTML encoded entities with `<`, `>`, `&`, `'`, `"and/. \ilinebr </td> </tr> <tr class="markdown TableRowOdd"> <td class="markdownTableBody None"> **whitelist(input, chars)** \ilinebr </td> <td class="markdownTableBodyNone"> remove characters that do not appear in the whitelist. The characters are used in a RegExp and so you will need to escape some chars, e.g.whitelist(input, '\[\]')`.

198.4.1 XSS Sanitization

XSS sanitization was removed from the library in 2d5d6999.
For an alternative, have a look at Yahoo's xss-filters library or at DOMPurify.

198.5 Maintainers

- chriso - **Chris O'Hara** (author)
- profnandaa - **Anthony Nandaa**
- ezkemboi - **Ezrqn Kemboi**
- tux-tn - **Sarhan Aissi**

198.6 Reading

Remember, validating can be troublesome sometimes. See A list of articles about programming assumptions commonly made that aren't true.

198.7 License (MIT)

Chapter 199

1.1.2 / 2017-09-23

- perf: improve header token parsing speed

199.1 1.1.1 / 2017-03-20

- perf: hoist regular expression

199.2 1.1.0 / 2015-09-29

- Only accept valid field names in the `field` argument
  - Ensures the resulting string is a valid HTTP header value

199.3 1.0.1 / 2015-07-08

- Fix setting empty header from empty `field`
- perf: enable strict mode
- perf: remove argument reassignments

199.4 1.0.0 / 2014-08-10

- Accept valid `Vary` header string as `field`
- Add `vary.append` for low-level string manipulation
- Move to `jshttp` orgainzation

199.5 0.1.0 / 2014-06-05

- Support array of fields to set

199.6 0.0.0 / 2014-06-04

- Initial release

Chapter 200

vary

Manipulate the HTTP Vary header

200.1 Installation

This is a `Node.js` module available through the `npm registry`. Installation is done using the `npm install command`:

```
$ npm install vary
```

200.2 API

```
var vary = require('vary')
```

200.2.1 vary(res, field)

Adds the given header `field` to the `Vary` response header of `res`. This can be a string of a single field, a string of a valid `Vary` header, or an array of multiple fields.

This will append the header if not already listed, otherwise leaves it listed in the current location.

```
// Append "Origin" to the Vary header of the response
vary(res, 'Origin')
```

200.2.2 vary.append(header, field)

Adds the given header `field` to the `Vary` response header string `header`. This can be a string of a single field, a string of a valid `Vary` header, or an array of multiple fields.

This will append the header if not already listed, otherwise leaves it listed in the current location. The new header string is returned.

```
// Get header string appending "Origin" to "Accept, User-Agent"
vary.append('Accept, User-Agent', 'Origin')
```

200.3 Examples

200.3.1 Updating the Vary header when content is based on it

```
var http = require('http')
var vary = require('vary')
http.createServer(function onRequest (req, res) {
  // about to user-agent sniff
  vary(res, 'User-Agent')
  var ua = req.headers['user-agent'] || ''
  var isMobile = /mobi|android|touch|mini/i.test(ua)
  // serve site, depending on isMobile
  res.setHeader('Content-Type', 'text/html')
  res.end('You are (probably) ' + (isMobile ? '' : 'not ') + 'a mobile user')
})
```

200.4 Testing

```
$ npm test
```

200.5 License

[MIT](LICENSE)

Chapter 201

wrappy

Callback wrapping utility

201.1 USAGE

```
var wrappy = require("wrappy")
// var wrapper = wrappy(wrapperFunction)
// make sure a cb is called only once
// See also:  http://npm.im/once for this specific use case
var once = wrappy(function (cb) {
  var called = false
  return function () {
    if (called) return
    called = true
    return cb.apply(this, arguments)
  }
})
function printBoo () {
  console.log('boo')
}
// has some rando property
printBoo.iAmBooPrinter = true
var onlyPrintOnce = once(printBoo)
onlyPrintOnce() // prints 'boo'
onlyPrintOnce() // does nothing
// random property is retained!
assert.equal(onlyPrintOnce.iAmBooPrinter, true)
```

Chapter 202

YAML <a href="https://www.npmjs.com/package/yaml"><img align="right" src="https://badge.fury.io/js/yaml.svg" title="npm package" /></a>

`yaml` is a JavaScript parser and stringifier for `YAML`, a human friendly data serialization standard. It supports both parsing and stringifying data using all versions of YAML, along with all common data schemas. As a particularly distinguishing feature, `yaml` fully supports reading and writing comments and blank lines in YAML documents.
The library is released under the ISC open source license, and the code is `available on GitHub`. It has no external dependencies and runs on Node.js 6 and later, and in browsers from IE 11 upwards.
For the purposes of versioning, any changes that break any of the endpoints or APIs documented here will be considered semver-major breaking changes. Undocumented library internals may change between minor versions, and previous APIs may be deprecated (but not removed).
For more information, see the project's documentation site: **`eemeli.org/yaml`**
To install:

```
npm install yaml@next
```

Note: These docs are for `yaml@2`. For v1, see the `v1.10.0 tag` for the source and `eemeli.org/yaml/v1` for the documentation.

202.1 API Overview

The API provided by `yaml` has three layers, depending on how deep you need to go: `Parse & Stringify`, `Documents`, and the `CST Parser`. The first has the simplest API and "just works", the second gets you all the bells and whistles supported by the library along with a decent `AST`, and the third is the closest to YAML source, making it fast, raw, and crude.

```
import YAML from 'yaml'
// or
const YAML = require('yaml')
```

202.1.1 Parse & Stringify

- `YAML.parse(str, reviver?, options?): value`
- `YAML.stringify(value, replacer?, options?): string`

202.1.2 YAML Documents

- `YAML.defaultOptions`
- `YAML.Document`
  - `constructor(value, replacer?, options?)`
  - `defaults`

  - #createNode(value, options?): Node
  - #anchors
  - #contents
  - #errors
- YAML.parseAllDocuments(str, options?): YAML.Document[]
- YAML.parseDocument(str, options?): YAML.Document

```
import { Pair, YAMLMap, YAMLSeq } from 'yaml/types'
```

- new Pair(key, value)
- new YAMLMap()
- new YAMLSeq()

202.1.3 CST Parser

```
import parseCST from 'yaml/parse-cst'
```

- parseCST(str): CSTDocument[]
- YAML.parseCST(str): CSTDocument[]

202.2 YAML.parse

```
# file.yml
YAML:
  - A human-readable data serialization language
  - https://en.wikipedia.org/wiki/YAML
yaml:
  - A complete JavaScript implementation
  - https://www.npmjs.com/package/yaml
import fs from 'fs'
import YAML from 'yaml'
YAML.parse('3.14159')
// 3.14159
YAML.parse('[ true, false, maybe, null ]\n')
// [ true, false, 'maybe', null ]
const file = fs.readFileSync('./file.yml', 'utf8')
YAML.parse(file)
// { YAML:
//   [ 'A human-readable data serialization language',
//     'https://en.wikipedia.org/wiki/YAML' ],
//   yaml:
//   [ 'A complete JavaScript implementation',
//     'https://www.npmjs.com/package/yaml' ] }
```

202.3 YAML.stringify

```
import YAML from 'yaml'
YAML.stringify(3.14159)
// '3.14159\n'
YAML.stringify([true, false, 'maybe', null])
// `- true
// - false
// - maybe
// - null
// `
YAML.stringify({ number: 3, plain: 'string', block: 'two\nlines\n' })
// `number: 3
// plain: string
// block: |
//   two
//   lines
// `
```

Browser testing provided by:

Chapter 203

Commander.js

The complete solution for `node.js` command-line interfaces.
Read this in other languages: English |

        * Parsing Configuration
        * Legacy options as properties
        * TypeScript
        * createCommand()
        * Node options such as `--harmony`
        * Debugging stand-alone executable subcommands
        * Display error
        * Override exit and output handling
        * Additional documentation
    - Support
        * Commander for enterprise

For information about terms used in this document see: terminology

203.1 Installation

```
npm install commander
```

203.2 Quick Start

You write code to describe your command line interface. Commander looks after parsing the arguments into options and command-arguments, displays usage errors for problems, and implements a help system.
Commander is strict and displays an error for unrecognised options. The two most used option types are a boolean option, and an option which takes its value from the following argument.
Example file: split.js

```
const { program } = require('commander');
program
  .option('--first')
  .option('-s, --separator <char>');
program.parse();
const options = program.opts();
const limit = options.first ? 1 : undefined;
console.log(program.args[0].split(options.separator, limit));
$ node split.js -s / --fits a/b/c
error: unknown option '--fits'
(Did you mean --first?)
$ node split.js -s / --first a/b/c
[ 'a' ]
```

Here is a more complete program using a subcommand and with descriptions for the help. In a multi-command program, you have an action handler for each command (or stand-alone executables for the commands).
Example file: string-util.js

```
const { Command } = require('commander');
const program = new Command();
program
  .name('string-util')
  .description('CLI to some JavaScript string utilities')
  .version('0.8.0');
program.command('split')
  .description('Split a string into substrings and display as an array')
  .argument('<string>', 'string to split')
  .option('--first', 'display just the first substring')
  .option('-s, --separator <char>', 'separator character', ',')
  .action((str, options) => {
    const limit = options.first ? 1 : undefined;
    console.log(str.split(options.separator, limit));
  });
program.parse();
$ node string-util.js help split
Usage: string-util split [options] <string>
Split a string into substrings and display as an array.
Arguments:
  string                  string to split
Options:
  --first                 display just the first substring
  -s, --separator <char>  separator character (default: ",")
  -h, --help              display help for command
$ node string-util.js split --separator=/ a/b/c
[ 'a', 'b', 'c' ]
```

More samples can be found in the examples directory.

203.3 Declaring <em>program</em> variable

Commander exports a global object which is convenient for quick programs. This is used in the examples in this README for brevity.

```
// CommonJS (.cjs)
const { program } = require('commander');
```

For larger programs which may use commander in multiple ways, including unit testing, it is better to create a local Command object to use.

```
// CommonJS (.cjs)
const { Command } = require('commander');
const program = new Command();
// ECMAScript (.mjs)
import { Command } from 'commander';
const program = new Command();
// TypeScript (.ts)
import { Command } from 'commander';
const program = new Command();
```

203.4 Options

Options are defined with the `.option()` method, also serving as documentation for the options. Each option can have a short flag (single character) and a long name, separated by a comma or space or vertical bar ('|').

The parsed options can be accessed by calling `.opts()` on a `Command` object, and are passed to the action handler.

Multi-word options such as "--template-engine" are camel-cased, becoming `program.opts().templateEngine` etc.

An option and its option-argument can be separated by a space, or combined into the same argument. The option-argument can follow the short option directly or follow an = for a long option.

```
serve -p 80
serve -p80
serve --port 80
serve --port=80
```

You can use `--` to indicate the end of the options, and any remaining arguments will be used without being interpreted.

By default options on the command line are not positional, and can be specified before or after other arguments.

There are additional related routines for when `.opts()` is not enough:

- `.optsWithGlobals()` returns merged local and global option values
- `.getOptionValue()` and `.setOptionValue()` work with a single option value
- `.getOptionValueSource()` and `.setOptionValueWithSource()` include where the option value came from

203.4.1 Common option types, boolean and value

The two most used option types are a boolean option, and an option which takes its value from the following argument (declared with angle brackets like `--expect <value>`). Both are `undefined` unless specified on command line.

Example file: `options-common.js`

```
program
  .option('-d, --debug', 'output extra debugging')
  .option('-s, --small', 'small pizza size')
  .option('-p, --pizza-type <type>', 'flavour of pizza');
program.parse(process.argv);
const options = program.opts();
if (options.debug) console.log(options);
console.log('pizza details:');
if (options.small) console.log('- small pizza size');
if (options.pizzaType) console.log(`- ${options.pizzaType}`);
$ pizza-options -p
error: option '-p, --pizza-type <type>' argument missing
$ pizza-options -d -s -p vegetarian
{ debug: true, small: true, pizzaType: 'vegetarian' }
pizza details:
- small pizza size
- vegetarian
$ pizza-options --pizza-type=cheese
pizza details:
- cheese
```

Multiple boolean short options may be combined together following the dash, and may be followed by a single short option taking a value. For example `-d -s -p cheese` may be written as `-ds -p cheese` or even `-dsp cheese`.
Options with an expected option-argument are greedy and will consume the following argument whatever the value. So `--id -xyz` reads `-xyz` as the option-argument.
`program.parse(arguments)` processes the arguments, leaving any args not consumed by the program options in the `program.args` array. The parameter is optional and defaults to `process.argv`.

203.4.2 Default option value

You can specify a default value for an option.
Example file: `options-defaults.js`

```
program
  .option('-c, --cheese <type>', 'add the specified type of cheese', 'blue');
program.parse();
console.log(`cheese: ${program.opts().cheese}`);
$ pizza-options
cheese: blue
$ pizza-options --cheese stilton
cheese: stilton
```

203.4.3 Other option types, negatable boolean and boolean|value

You can define a boolean option long name with a leading `no-` to set the option value to false when used. Defined alone this also makes the option true by default.
If you define `--foo` first, adding `--no-foo` does not change the default value from what it would otherwise be.
Example file: `options-negatable.js`

```
program
  .option('--no-sauce', 'Remove sauce')
  .option('--cheese <flavour>', 'cheese flavour', 'mozzarella')
  .option('--no-cheese', 'plain with no cheese')
  .parse();
const options = program.opts();
const sauceStr = options.sauce ? 'sauce' : 'no sauce';
const cheeseStr = (options.cheese === false) ? 'no cheese' : `${options.cheese} cheese`;
console.log(`You ordered a pizza with ${sauceStr} and ${cheeseStr}`);
$ pizza-options
You ordered a pizza with sauce and mozzarella cheese
$ pizza-options --sauce
error: unknown option '--sauce'
$ pizza-options --cheese=blue
You ordered a pizza with sauce and blue cheese
$ pizza-options --no-sauce --no-cheese
You ordered a pizza with no sauce and no cheese
```

You can specify an option which may be used as a boolean option but may optionally take an option-argument (declared with square brackets like `--optional [value]`).
Example file: `options-boolean-or-value.js`

```
program
  .option('-c, --cheese [type]', 'Add cheese with optional type');
program.parse(process.argv);
const options = program.opts();
if (options.cheese === undefined) console.log('no cheese');
else if (options.cheese === true) console.log('add cheese');
else console.log(`add cheese type ${options.cheese}`);
$ pizza-options
no cheese
$ pizza-options --cheese
add cheese
$ pizza-options --cheese mozzarella
add cheese type mozzarella
```

Options with an optional option-argument are not greedy and will ignore arguments starting with a dash. So `id` behaves as a boolean option for `--id -5`, but you can use a combined form if needed like `--id=-5`.
For information about possible ambiguous cases, see options taking varying arguments.

203.4.4 Required option

You may specify a required (mandatory) option using `.requiredOption()`. The option must have a value after parsing, usually specified on the command line, or perhaps from a default value (say from environment). The method is otherwise the same as `.option()` in format, taking flags and description, and optional default value or custom processing.
Example file: `options-required.js`

```
program
  .requiredOption('-c, --cheese <type>', 'pizza must have cheese');
program.parse();
$ pizza
error:  required option '-c, --cheese <type>' not specified
```

203.4.5 Variadic option

You may make an option variadic by appending `...` to the value placeholder when declaring the option. On the command line you can then specify multiple option-arguments, and the parsed option value will be an array. The extra arguments are read until the first argument starting with a dash. The special argument `--` stops option processing entirely. If a value is specified in the same argument as the option then no further values are read.
Example file: options-variadic.js

```
program
  .option('-n, --number <numbers...>', 'specify numbers')
  .option('-l, --letter [letters...]', 'specify letters');
program.parse();
console.log('Options:  ', program.opts());
console.log('Remaining arguments:  ', program.args);
$ collect -n 1 2 3 --letter a b c
Options:    { number:  [ '1', '2', '3' ], letter:  [ 'a', 'b', 'c' ] }
Remaining arguments:    []
$ collect --letter=A -n80 operand
Options:    { number:  [ '80' ], letter:  [ 'A' ] }
Remaining arguments:    [ 'operand' ]
$ collect --letter -n 1 -n 2 3 -- operand
Options:    { number:  [ '1', '2', '3' ], letter:  true }
Remaining arguments:    [ 'operand' ]
```

For information about possible ambiguous cases, see options taking varying arguments.

203.4.6 Version option

The optional `version` method adds handling for displaying the command version. The default option flags are `-V` and `--version`, and when present the command prints the version number and exits.

```
program.version('0.0.1');
$ ./examples/pizza -V
0.0.1
```

You may change the flags and description by passing additional parameters to the `version` method, using the same syntax for flags as the `option` method.

```
program.version('0.0.1', '-v, --vers', 'output the current version');
```

203.4.7 More configuration

You can add most options using the `.option()` method, but there are some additional features available by constructing an `Option` explicitly for less common cases.
Example files: options-extra.js, options-env.js, options-conflicts.js, options-implies.↩
js

```
program
  .addOption(new Option('-s, --secret').hideHelp())
  .addOption(new Option('-t, --timeout <delay>', 'timeout in seconds').default(60, 'one minute'))
  .addOption(new Option('-d, --drink <size>', 'drink size').choices(['small', 'medium', 'large']))
  .addOption(new Option('-p, --port <number>', 'port number').env('PORT'))
  .addOption(new Option('--donate [amount]', 'optional donation in
      dollars').preset('20').argParser(parseFloat))
  .addOption(new Option('--disable-server', 'disables the server').conflicts('port'))
  .addOption(new Option('--free-drink', 'small drink included free ').implies({ drink:  'small' }));
$ extra --help
Usage:  help [options]
Options:
  -t, --timeout <delay>  timeout in seconds (default:  one minute)
  -d, --drink <size>     drink cup size (choices:  "small", "medium", "large")
  -p, --port <number>    port number (env:  PORT)
  --donate [amount]      optional donation in dollars (preset:  "20")
  --disable-server       disables the server
  --free-drink           small drink included free
  -h, --help             display help for command
$ extra --drink huge
error:  option '-d, --drink <size>' argument 'huge' is invalid.  Allowed choices are small, medium, large.
$ PORT=80 extra --donate --free-drink
Options:    { timeout:  60, donate:  20, port:  '80', freeDrink:  true, drink:  'small' }
$ extra --disable-server --port 8000
error:  option '--disable-server' cannot be used with option '-p, --port <number>'
```

Specify a required (mandatory) option using the `Option` method `.makeOptionMandatory()`. This matches the `Command` method .requiredOption().

203.4.8 Custom option processing

You may specify a function to do custom processing of option-arguments. The callback function receives two parameters, the user specified option-argument and the previous value for the option. It returns the new value for the option.
This allows you to coerce the option-argument to the desired type, or accumulate values, or do entirely custom processing.
You can optionally specify the default/starting value for the option after the function parameter.
Example file: `options-custom-processing.js`

```
function myParseInt(value, dummyPrevious) {
  // parseInt takes a string and a radix
  const parsedValue = parseInt(value, 10);
  if (isNaN(parsedValue)) {
    throw new commander.InvalidArgumentError('Not a number.');
  }
  return parsedValue;
}
function increaseVerbosity(dummyValue, previous) {
  return previous + 1;
}
function collect(value, previous) {
  return previous.concat([value]);
}
function commaSeparatedList(value, dummyPrevious) {
  return value.split(',');
}
program
  .option('-f, --float <number>', 'float argument', parseFloat)
  .option('-i, --integer <number>', 'integer argument', myParseInt)
  .option('-v, --verbose', 'verbosity that can be increased', increaseVerbosity, 0)
  .option('-c, --collect <value>', 'repeatable value', collect, [])
  .option('-l, --list <items>', 'comma separated list', commaSeparatedList)
;
program.parse();
const options = program.opts();
if (options.float !== undefined) console.log(`float:  ${options.float}`);
if (options.integer !== undefined) console.log(`integer:  ${options.integer}`);
if (options.verbose > 0) console.log(`verbosity:  ${options.verbose}`);
if (options.collect.length > 0) console.log(options.collect);
if (options.list !== undefined) console.log(options.list);
$ custom -f 1e2
float:  100
$ custom --integer 2
integer:  2
$ custom -v -v -v
verbose:  3
$ custom -c a -c b -c c
[ 'a', 'b', 'c' ]
$ custom --list x,y,z
[ 'x', 'y', 'z' ]
```

203.5 Commands

You can specify (sub)commands using `.command()` or `.addCommand()`. There are two ways these can be implemented: using an action handler attached to the command, or as a stand-alone executable file (described in more detail later). The subcommands may be nested (`example`).
In the first parameter to `.command()` you specify the command name. You may append the command-arguments after the command name, or specify them separately using `.argument()`. The arguments may be `<required>` or `[optional]`, and the last argument may also be `variadic...`.
You can use `.addCommand()` to add an already configured subcommand to the program.
For example:

```
// Command implemented using action handler (description is supplied separately to `.command`)
// Returns new command for configuring.
program
  .command('clone <source> [destination]')
  .description('clone a repository into a newly created directory')
  .action((source, destination) => {
    console.log('clone command called');
  });
// Command implemented using stand-alone executable file, indicated by adding description as second
      parameter to `.command`.
// Returns `this` for adding more commands.
program
  .command('start <service>', 'start named service')
  .command('stop [service]', 'stop named service, or all if no name supplied');
// Command prepared separately.
// Returns `this` for adding more commands.
```

```
program
  .addCommand(build.makeBuildCommand());
```

Configuration options can be passed with the call to `.command()` and `.addCommand()`. Specifying `hidden: true` will remove the command from the generated help output. Specifying `isDefault: true` will run the subcommand if no other subcommand is specified (`example`).
You can add alternative names for a command with `.alias()`. (`example`)
For safety, `.addCommand()` does not automatically copy the inherited settings from the parent command. There is a helper routine `.copyInheritedSettings()` for copying the settings when they are wanted.

203.5.1 Command-arguments

For subcommands, you can specify the argument syntax in the call to `.command()` (as shown above). This is the only method usable for subcommands implemented using a stand-alone executable, but for other subcommands you can instead use the following method.
To configure a command, you can use `.argument()` to specify each expected command-argument. You supply the argument name and an optional description. The argument may be `<required>` or `[optional]`. You can specify a default value for an optional command-argument.
Example file: `argument.js`

```
program
  .version('0.1.0')
  .argument('<username>', 'user to login')
  .argument('[password]', 'password for user, if required', 'no password given')
  .action((username, password) => {
    console.log('username:', username);
    console.log('password:', password);
  });
```

The last argument of a command can be variadic, and only the last argument. To make an argument variadic you append `...` to the argument name. A variadic argument is passed to the action handler as an array. For example:

```
program
  .version('0.1.0')
  .command('rmdir')
  .argument('<dirs...>')
  .action(function (dirs) {
    dirs.forEach((dir) => {
      console.log('rmdir %s', dir);
    });
  });
```

There is a convenience method to add multiple arguments at once, but without descriptions:

```
program
  .arguments('<username> <password>');
```

203.5.1.1 More configuration

There are some additional features available by constructing an `Argument` explicitly for less common cases.
Example file: `arguments-extra.js`

```
program
  .addArgument(new commander.Argument('<drink-size>', 'drink cup size').choices(['small', 'medium',
      'large']))
  .addArgument(new commander.Argument('[timeout]', 'timeout in seconds').default(60, 'one minute'))
```

203.5.1.2 Custom argument processing

You may specify a function to do custom processing of command-arguments (like for option-arguments). The callback function receives two parameters, the user specified command-argument and the previous value for the argument. It returns the new value for the argument.
The processed argument values are passed to the action handler, and saved as `.processedArgs`.
You can optionally specify the default/starting value for the argument after the function parameter.
Example file: `arguments-custom-processing.js`

```
program
  .command('add')
  .argument('<first>', 'integer argument', myParseInt)
  .argument('[second]', 'integer argument', myParseInt, 1000)
  .action((first, second) => {
    console.log(`${first} + ${second} = ${first + second}`);
  })
;
```

203.5.2 Action handler

The action handler gets passed a parameter for each command-argument you declared, and two additional parameters which are the parsed options and the command object itself.
Example file: thank.js
```
program
  .argument('<name>')
  .option('-t, --title <honorific>', 'title to use before name')
  .option('-d, --debug', 'display some debugging')
  .action((name, options, command) => {
    if (options.debug) {
      console.error('Called %s with options %o', command.name(), options);
    }
    const title = options.title ? `${options.title} ` : '';
    console.log(`Thank-you ${title}${name}`);
  });
```
If you prefer, you can work with the command directly and skip declaring the parameters for the action handler. The `this` keyword is set to the running command and can be used from a function expression (but not from an arrow function).
Example file: action-this.js
```
program
  .command('serve')
  .argument('<script>')
  .option('-p, --port <number>', 'port number', 80)
  .action(function() {
    console.error('Run script %s on port %s', this.args[0], this.opts().port);
  });
```
You may supply an `async` action handler, in which case you call `.parseAsync` rather than `.parse`.
```
async function run() { /* code goes here */ }
async function main() {
  program
    .command('run')
    .action(run);
  await program.parseAsync(process.argv);
}
```
A command's options and arguments on the command line are validated when the command is used. Any unknown options or missing arguments will be reported as an error. You can suppress the unknown option checks with `.allowUnknownOption()`. By default it is not an error to pass more arguments than declared, but you can make this an error with `.allowExcessArguments(false)`.

203.5.3 Stand-alone executable (sub)commands

When `.command()` is invoked with a description argument, this tells Commander that you're going to use stand-alone executables for subcommands. Commander will search the files in the directory of the entry script for a file with the name combination `command-subcommand`, like `pm-install` or `pm-search` in the example below. The search includes trying common file extensions, like `.js`. You may specify a custom name (and path) with the `executableFile` configuration option. You may specify a custom search directory for subcommands with `.executableDir()`.
You handle the options for an executable (sub)command in the executable, and don't declare them at the top-level.
Example file: pm
```
program
  .name('pm')
  .version('0.1.0')
  .command('install [name]', 'install one or more packages')
  .command('search [query]', 'search with optional query')
  .command('update', 'update installed packages', { executableFile: 'myUpdateSubCommand' })
  .command('list', 'list packages installed', { isDefault: true });
program.parse(process.argv);
```
If the program is designed to be installed globally, make sure the executables have proper modes, like `755`.

203.5.4 Life cycle hooks

You can add callback hooks to a command for life cycle events.
Example file: hook.js
```
program
  .option('-t, --trace', 'display trace statements for commands')
  .hook('preAction', (thisCommand, actionCommand) => {
    if (thisCommand.opts().trace) {
      console.log(`About to call action handler for subcommand: ${actionCommand.name()}`);
      console.log('arguments: %O', actionCommand.args);
      console.log('options: %o', actionCommand.opts());
    }
  });
```

The callback hook can be `async`, in which case you call `.parseAsync` rather than `.parse`. You can add multiple hooks per event.
The supported events are:

event name	when hook called	callback parameters
`preAction`, `postAction`	before/after action handler for this command and its nested subcommands	`(thisCommand, action←Command)`
`preSubcommand`	before parsing direct subcommand	`(thisCommand, subcommand)`

203.6 Automated help

The help information is auto-generated based on the information commander already knows about your program. The default help option is `-h,--help`.
Example file: pizza
```
$ node ./examples/pizza --help
Usage:  pizza [options]
An application for pizza ordering
Options:
  -p, --peppers        Add peppers
  -c, --cheese <type>  Add the specified type of cheese (default:  "marble")
  -C, --no-cheese      You do not want any cheese
  -h, --help           display help for command
```
A `help` command is added by default if your command has subcommands. It can be used alone, or with a subcommand name to show further help for the subcommand. These are effectively the same if the `shell` program has implicit help:
```
shell help
shell --help
shell help spawn
shell spawn --help
```

203.6.1 Custom help

You can add extra text to be displayed along with the built-in help.
Example file: custom-help
```
program
  .option('-f, --foo', 'enable some foo');
program.addHelpText('after', `
Example call:
  $ custom-help --help`);
```
Yields the following help output:
```
Usage:  custom-help [options]
Options:
  -f, --foo   enable some foo
  -h, --help  display help for command
Example call:
  $ custom-help --help
```
The positions in order displayed are:

- `beforeAll`: add to the program for a global banner or header
- `before`: display extra information before built-in help
- `after`: display extra information after built-in help
- `afterAll`: add to the program for a global footer (epilog)

The positions "beforeAll" and "afterAll" apply to the command and all its subcommands.
The second parameter can be a string, or a function returning a string. The function is passed a context object for your convenience. The properties are:

- error: a boolean for whether the help is being displayed due to a usage error
- command: the Command which is displaying the help

203.6.2 Display help after errors

The default behaviour for usage errors is to just display a short error message. You can change the behaviour to show the full help or a custom help message after an error.

```
program.showHelpAfterError();
// or
program.showHelpAfterError('(add --help for additional information)');
$ pizza --unknown
error:  unknown option '--unknown'
(add --help for additional information)
```

The default behaviour is to suggest correct spelling after an error for an unknown command or option. You can disable this.

```
program.showSuggestionAfterError(false);
$ pizza --hepl
error:  unknown option '--hepl'
(Did you mean --help?)
```

203.6.3 Display help from code

`.help()`: display help information and exit immediately. You can optionally pass `{ error: true }` to display on stderr and exit with an error status.
`.outputHelp()`: output help information without exiting. You can optionally pass `{ error: true }` to display on stderr.
`.helpInformation()`: get the built-in command help information as a string for processing or displaying yourself.

203.6.4 .name

The command name appears in the help, and is also used for locating stand-alone executable subcommands.
You may specify the program name using `.name()` or in the Command constructor. For the program, Commander will fallback to using the script name from the full arguments passed into `.parse()`. However, the script name varies depending on how your program is launched so you may wish to specify it explicitly.

```
program.name('pizza');
const pm = new Command('pm');
```

Subcommands get a name when specified using `.command()`. If you create the subcommand yourself to use with `.addCommand()`, then set the name using `.name()` or in the Command constructor.

203.6.5 .usage

This allows you to customise the usage description in the first line of the help. Given:

```
program
  .name("my-command")
  .usage("[global options] command")
```

The help will start with:

```
Usage:  my-command [global options] command
```

203.6.6 .description and .summary

The description appears in the help for the command. You can optionally supply a shorter summary to use when listed as a subcommand of the program.

```
program
  .command("duplicate")
  .summary("make a copy")
  .description(`Make a copy of the current project.
This may require additional disk space.
  `);
```

203.6.7 .helpOption(flags, description)

By default every command has a help option. You may change the default help flags and description. Pass false to disable the built-in help option.

```
program
  .helpOption('-e, --HELP', 'read more information');
```

203.6.8 .addHelpCommand()

A help command is added by default if your command has subcommands. You can explicitly turn on or off the implicit help command with `.addHelpCommand()` and `.addHelpCommand(false)`.
You can both turn on and customise the help command by supplying the name and description:

```
program.addHelpCommand('assist [command]', 'show assistance');
```

203.6.9 More configuration

The built-in help is formatted using the Help class. You can configure the Help behaviour by modifying data properties and methods using `.configureHelp()`, or by subclassing using `.createHelp()` if you prefer.
The data properties are:

- `helpWidth`: specify the wrap width, useful for unit tests
- `sortSubcommands`: sort the subcommands alphabetically
- `sortOptions`: sort the options alphabetically
- `showGlobalOptions`: show a section with the global options from the parent command(s)

There are methods getting the visible lists of arguments, options, and subcommands. There are methods for formatting the items in the lists, with each item having a *term* and *description*. Take a look at `.formatHelp()` to see how they are used.
Example file: `configure-help.js`

```
program.configureHelp({
  sortSubcommands: true,
  subcommandTerm: (cmd) => cmd.name() // Just show the name, instead of short usage.
});
```

203.7 Custom event listeners

You can execute custom actions by listening to command and option events.

```
program.on('option:verbose', function () {
  process.env.VERBOSE = this.opts().verbose;
});
```

203.8 Bits and pieces

203.8.1 .parse() and .parseAsync()

The first argument to `.parse` is the array of strings to parse. You may omit the parameter to implicitly use `process.argv`.
If the arguments follow different conventions than node you can pass a `from` option in the second parameter:

- 'node': default, `argv[0]` is the application and `argv[1]` is the script being run, with user parameters after that
- 'electron': `argv[1]` varies depending on whether the electron application is packaged
- 'user': all of the arguments from the user

For example:

```
program.parse(process.argv); // Explicit, node conventions
program.parse(); // Implicit, and auto-detect electron
program.parse(['-f', 'filename'], { from: 'user' });
```

203.8.2 Parsing Configuration

If the default parsing does not suit your needs, there are some behaviours to support other usage patterns.
By default program options are recognised before and after subcommands. To only look for program options before subcommands, use `.enablePositionalOptions()`. This lets you use an option for a different purpose in subcommands.
Example file: `positional-options.js`

With positional options, the `-b` is a program option in the first line and a subcommand option in the second line:

```
program -b subcommand
program subcommand -b
```

By default options are recognised before and after command-arguments. To only process options that come before the command-arguments, use `.passThroughOptions()`. This lets you pass the arguments and following options through to another program without needing to use `--` to end the option processing. To use pass through options in a subcommand, the program needs to enable positional options.
Example file: `pass-through-options.js`
With pass through options, the `--port=80` is a program option in the first line and passed through as a command-argument in the second line:

```
program --port=80 arg
program arg --port=80
```

By default the option processing shows an error for an unknown option. To have an unknown option treated as an ordinary command-argument and continue looking for options, use `.allowUnknownOption()`. This lets you mix known and unknown options.
By default the argument processing does not display an error for more command-arguments than expected. To display an error for excess arguments, use<tt>.allowExcessArguments(false).

203.8.3 Legacy options as properties

Before Commander 7, the option values were stored as properties on the command. This was convenient to code but the downside was possible clashes with existing properties of `Command`. You can revert to the old behaviour to run unmodified legacy code by using `.storeOptionsAsProperties()`.

```
program
  .storeOptionsAsProperties()
  .option('-d, --debug')
  .action((commandAndOptions) => {
    if (commandAndOptions.debug) {
      console.error(`Called ${commandAndOptions.name()}`);
    }
  });
```

203.8.4 TypeScript

extra-typings: There is an optional project to infer extra type information from the option and argument definitions. This adds strong typing to the options returned by `.opts()` and the parameters to `.action()`. See `commander-js/extra-typings` for more.

```
import { Command } from '@commander-js/extra-typings';
```

ts-node: If you use `ts-node` and stand-alone executable subcommands written as `.ts` files, you need to call your program through node to get the subcommands called correctly. e.g.

```
node -r ts-node/register pm.ts
```

203.8.5 createCommand()

This factory function creates a new command. It is exported and may be used instead of using `new`, like:

```
const { createCommand } = require('commander');
const program = createCommand();
```

`createCommand` is also a method of the Command object, and creates a new command rather than a subcommand. This gets used internally when creating subcommands using `.command()`, and you may override it to customise the new subcommand (example file `custom-command-class.js`).

203.8.6 Node options such as <tt>--harmony</tt>

You can enable `--harmony` option in two ways:

- Use `#! /usr/bin/env node --harmony` in the subcommands scripts. (Note Windows does not support this pattern.)
- Use the `--harmony` option when call the command, like `node --harmony examples/pm publish`. The `--harmony` option will be preserved when spawning subcommand process.

203.8.7 Debugging stand-alone executable subcommands

An executable subcommand is launched as a separate child process.

If you are using the node inspector for debugging executable subcommands using `node --inspect` et al, the inspector port is incremented by 1 for the spawned subcommand.
If you are using VSCode to debug executable subcommands you need to set the `"autoAttachChildProcesses": true` flag in your launch.json configuration.

203.8.8 Display error

This routine is available to invoke the Commander error handling for your own error conditions. (See also the next section about exit handling.)
As well as the error message, you can optionally specify the `exitCode` (used with `process.exit`) and `code` (used with `CommanderError`).

```
program.error('Password must be longer than four characters');
program.error('Custom processing has failed', { exitCode: 2, code: 'my.custom.error' });
```

203.8.9 Override exit and output handling

By default Commander calls `process.exit` when it detects errors, or after displaying the help or version. You can override this behaviour and optionally supply a callback. The default override throws a `CommanderError`.
The override callback is passed a `CommanderError` with properties `exitCode` number, `code` string, and `message`. The default override behaviour is to throw the error, except for async handling of executable subcommand completion which carries on. The normal display of error messages or version or help is not affected by the override which is called after the display.

```
program.exitOverride();
try {
  program.parse(process.argv);
} catch (err) {
  // custom processing...
}
```

By default Commander is configured for a command-line application and writes to stdout and stderr. You can modify this behaviour for custom applications. In addition, you can modify the display of error messages.
Example file: `configure-output.js`

```
function errorColor(str) {
  // Add ANSI escape codes to display text in red.
  return `\x1b[31m${str}\x1b[0m`;
}
program
  .configureOutput({
    // Visibly override write routines as example!
    writeOut: (str) => process.stdout.write(`[OUT] ${str}`),
    writeErr: (str) => process.stdout.write(`[ERR] ${str}`),
    // Highlight errors in color.
    outputError: (str, write) => write(errorColor(str))
  });
```

203.8.10 Additional documentation

There is more information available about:

- deprecated features still supported for backwards compatibility
- options taking varying arguments

203.9 Support

The current version of Commander is fully supported on Long Term Support versions of Node.js, and requires at least v12.20.0. (For older versions of Node.js, use an older version of Commander.)
The main forum for free and community support is the project `Issues` on GitHub.

203.9.1 Commander for enterprise

Available as part of the Tidelift Subscription
The maintainers of Commander and thousands of other packages are working with Tidelift to deliver commercial support and maintenance for the open source dependencies you use to build your applications. Save time, reduce risk, and improve code health, while paying the maintainers of the exact dependencies you use. `Learn more.`

Chapter 204

z-schema validator

- version 3.0 runs also in the browsers now, run tests yourself here

204.1 Topics

- Usage
- Features
- Options
- Benchmarks
- Contributors

204.2 Usage

Validator will try to perform sync validation when possible for speed, but supports async callbacks when they are necessary.

204.2.1 Development:

These repository has several submodules and should be cloned as follows: >git clone **–recursive** https://github.com/zaggino/z-schema.git

204.2.2 CLI:

```
npm install --global z-schema
z-schema --help
z-schema mySchema.json
z-schema mySchema.json myJson.json
z-schema --strictMode mySchema.json myJson.json
```

204.2.3 NodeJS:

```
var ZSchema = require("z-schema");
var options = ...  // see below for possible option values
var validator = new ZSchema(options);
```

204.2.4 Sync mode:

```
var valid = validator.validate(json, schema);
// this will return a native error object with name and message
var error = validator.getLastError();
// this will return an array of validation errors encountered
var errors = validator.getLastErrors();
...
```

204.2.5 Async mode:

```
validator.validate(json, schema, function (err, valid) {
    ...
});
```

204.2.6 Browser:

```
<script type="text/javascript" src="../dist/ZSchema-browser-min.js"></script>
<script type="text/javascript">
    var validator = new ZSchema();
    var valid = validator.validate("string", { "type":  "string" });
    console.log(valid);
</script>
```

204.2.7 Remote references and schemas:

In case you have some remote references in your schemas, you have to download those schemas before using validator. Otherwise you'll get `UNRESOLVABLE_REFERENCE` error when trying to compile a schema.

```
var validator = new ZSchema();
var json = {};
var schema = { "$ref":  "http://json-schema.org/draft-04/schema#" };
var valid = validator.validate(json, schema);
var errors = validator.getLastErrors();
// valid === false
// errors.length === 1
// errors[0].code === "UNRESOLVABLE_REFERENCE"
var requiredUrl = "http://json-schema.org/draft-04/schema";
request(requiredUrl, function (error, response, body) {
    validator.setRemoteReference(requiredUrl, JSON.parse(body));
    var valid = validator.validate(json, schema);
    var errors = validator.getLastErrors();
    // valid === true
    // errors === undefined
}
```

If you're able to load schemas synchronously, you can use `ZSchema.setSchemaReader` feature:

```
ZSchema.setSchemaReader(function (uri) {
    var someFilename = path.resolve(__dirname, "..", "schemas", uri + ".json");
    return JSON.parse(fs.readFileSync(someFilename, "utf8"));
});
```

204.3 Features

- Validate against subschema
- Compile arrays of schemas and use references between them
- Register a custom format
- Automatic downloading of remote schemas
- Prefill default values to object using format
- Define a custom timeout for all async operations
- Disallow validation of empty arrays as arrays
- Disallow validation of empty strings as strings
- Disallow schemas that don't have a type specified
- Disallow schemas that contain unrecognized keywords and are not validated by parent schemas
- Assume additionalItems/additionalProperties are defined in schemas as false
- Force additionalItems/additionalProperties to be defined in schemas
- Force items to be defined in array type schemas
- Force minItems to be defined in array type schemas
- Force maxItems to be defined in array type schemas

- Force minLength to be defined in string type schemas
- Force maxLength to be defined in string type schemas
- Force properties or patternProperties to be defined in object type schemas
- Ignore remote references to schemas that are not cached or resolvable
- Ignore case mismatch when validating enum values
- Only allow strictly absolute URIs to be used in schemas
- Turn on z-schema strict mode
- Set validator to collect as many errors as possible
- Report paths in errors as arrays so they can be processed easier

204.3.1 Validate against subschema

In case you don't want to split your schema into multiple schemas using reference for any reason, you can use option schemaPath when validating:

```
var valid = validator.validate(cars, schema, { schemaPath:  "definitions.car.definitions.cars" });
```

See more details in the `test`.

204.3.2 Compile arrays of schemas and use references between them

You can use validator to compile an array of schemas that have references between them and then validate against one of those schemas:

```
var schemas = [
    {
        id:  "personDetails",
        type:  "object",
        properties:  {
            firstName:  { type:  "string" },
            lastName:  { type:  "string" }
        },
        required:  ["firstName", "lastName"]
    },
    {
        id:  "addressDetails",
        type:  "object",
        properties:  {
            street:  { type:  "string" },
            city:  { type:  "string" }
        },
        required:  ["street", "city"]
    },
    {
        id:  "personWithAddress",
        allOf:  [
            { $ref:  "personDetails" },
            { $ref:  "addressDetails" }
        ]
    }
];
var data = {
    firstName:  "Martin",
    lastName:  "Zagora",
    street:  "George St",
    city:  "Sydney"
};
var validator = new ZSchema();
// compile & validate schemas first, z-schema will automatically handle array
var allSchemasValid = validator.validateSchema(schemas);
// allSchemasValid === true
// now validate our data against the last schema
var valid = validator.validate(data, schemas[2]);
// valid === true
```

204.3.3 Register a custom format

You can register any format of your own. Your sync validator function should always respond with a boolean:

```
ZSchema.registerFormat("xstring", function (str) {
    return str === "xxx";
});
```

Async format validators are also supported, they should accept two arguments, value and a callback to which they need to respond:
```
ZSchema.registerFormat("xstring", function (str, callback) {
    setTimeout(function () {
        callback(str === "xxx");
    }, 1);
});
```

204.3.4 Helper method to check the formats that have been registered

```
var registeredFormats = ZSchema.getRegisteredFormats();
//registeredFormats will now contain an array of all formats that have been registered with z-schema
```

204.3.5 Automatic downloading of remote schemas

Automatic downloading of remote schemas was removed from version `3.x` but is still possible with a bit of extra code, see `this test` for more information on this.

204.3.6 Prefill default values to object using format

Using format, you can pre-fill values of your choosing into the objects like this:
```
ZSchema.registerFormat("fillHello", function (obj) {
    obj.hello = "world";
    return true;
});
var data = {};
var schema = {
    "type":  "object",
    "format":  "fillHello"
};
validator.validate(data, schema);
// data.hello === "world"
```

204.4 Options

204.4.1 asyncTimeout

Defines a time limit, which should be used when waiting for async tasks like async format validators to perform their validation, before the validation fails with an `ASYNC_TIMEOUT` error.
```
var validator = new ZSchema({
    asyncTimeout:  2000
});
```

204.4.2 noEmptyArrays

When true, validator will assume that minimum count of items in any `array` is 1, except when `minItems: 0` is explicitly defined.
```
var validator = new ZSchema({
    noEmptyArrays:  true
});
```

204.4.3 noEmptyStrings

When true, validator will assume that minimum length of any string to pass type `string` validation is 1, except when `minLength: 0` is explicitly defined.
```
var validator = new ZSchema({
    noEmptyStrings:  true
});
```

204.4.4 noTypeless

When true, validator will fail validation for schemas that don't specify a `type` of object that they expect.
```
var validator = new ZSchema({
    noTypeless:  true
});
```

204.4.5 noExtraKeywords

When true, validator will fail for schemas that use keywords not defined in JSON Schema specification and doesn't provide a parent schema in `$schema` property to validate the schema.

```
var validator = new ZSchema({
    noExtraKeywords:  true
});
```

204.4.6 assumeAdditional

When true, validator assumes that additionalItems/additionalProperties are defined as false so you don't have to manually fix all your schemas.

```
var validator = new ZSchema({
    assumeAdditional:  true
});
```

When an array, validator assumes that additionalItems/additionalProperties are defined as false, but allows some properties to pass.

```
var validator = new ZSchema({
    assumeAdditional:  ["$ref"]
});
```

204.4.7 forceAdditional

When true, validator doesn't validate schemas where additionalItems/additionalProperties should be defined to either true or false.

```
var validator = new ZSchema({
    forceAdditional:  true
});
```

204.4.8 forceItems

When true, validator doesn't validate schemas where `items` are not defined for `array` type schemas. This is to avoid passing anything through an array definition.

```
var validator = new ZSchema({
    forceItems:  true
});
```

204.4.9 forceMinItems

When true, validator doesn't validate schemas where `minItems` is not defined for `array` type schemas. This is to avoid passing zero-length arrays which application doesn't expect to handle.

```
var validator = new ZSchema({
    forceMinItems:  true
});
```

204.4.10 forceMaxItems

When true, validator doesn't validate schemas where `maxItems` is not defined for `array` type schemas. This is to avoid passing arrays with unlimited count of elements which application doesn't expect to handle.

```
var validator = new ZSchema({
    forceMaxItems:  true
});
```

204.4.11 forceMinLength

When true, validator doesn't validate schemas where `minLength` is not defined for `string` type schemas. This is to avoid passing zero-length strings which application doesn't expect to handle.

```
var validator = new ZSchema({
    forceMinLength:  true
});
```

204.4.12 forceMaxLength

When true, validator doesn't validate schemas where `maxLength` is not defined for `string` type schemas. This is to avoid passing extremly large strings which application doesn't expect to handle.

```
var validator = new ZSchema({
    forceMaxLength:  true
});
```

204.4.13 forceProperties

When true, validator doesn't validate schemas where `properties` or `patternProperties` is not defined for `object` type schemas. This is to avoid having objects with unexpected properties in application.

```
var validator = new ZSchema({
    forceProperties: true
});
```

204.4.14 ignoreUnresolvableReferences

When true, validator doesn't end with error when a remote reference is unreachable. **This setting is not recommended in production outside of testing.**

```
var validator = new ZSchema({
    ignoreUnresolvableReferences: true
});
```

204.4.15 enumCaseInsensitiveComparison

When true, validator will return a `ENUM_CASE_MISMATCH` when the enum values mismatch only in case.

```
var validator = new ZSchema({
    enumCaseInsensitiveComparison: true
});
```

204.4.16 strictUris

When true, all strings of format `uri` must be an absolute URIs and not only URI references. See more details in `this issue`.

```
var validator = new ZSchema({
    strictUris: true
});
```

204.4.17 strictMode

Strict mode of z-schema is currently equal to the following:

```
if (this.options.strictMode === true) {
    this.options.forceAdditional  = true;
    this.options.forceItems       = true;
    this.options.forceMaxLength   = true;
    this.options.forceProperties  = true;
    this.options.noExtraKeywords  = true;
    this.options.noTypeless       = true;
    this.options.noEmptyStrings   = true;
    this.options.noEmptyArrays    = true;
}
var validator = new ZSchema({
    strictMode: true
});
```

204.4.18 breakOnFirstError

default: `false`
When true, will stop validation after the first error is found:

```
var validator = new ZSchema({
    breakOnFirstError: true
});
```

204.4.19 reportPathAsArray

Report error paths as an array of path segments instead of a string:

```
var validator = new ZSchema({
    reportPathAsArray: true
});
```

204.4.20 ignoreUnknownFormats

By default, z-schema reports all unknown formats, formats not defined by JSON Schema and not registered using `ZSchema.registerFormat`, as an error. But the `JSON Schema specification` says that validator implementations *"they SHOULD offer an option to disable validation"* for `format`. That being said, setting this option to `true` will disable treating unknown formats as errlrs

```
var validator = new ZSchema({
    ignoreUnknownFormats:  true
});
```

204.4.21 includeErrors

By default, z-schema reports all errors. If interested only in a subset of the errors, passing the option `include←Errors` to `validate` will perform validations only for those errors.

```
var validator = new ZSchema();
// will only execute validation for "INVALID_TYPE" error.
validator.validate(json, schema, {includeErrors:  ["INVALID_TYPE"]});
```

204.4.22 customValidator

Warning: Use only if know what you are doing. Always consider using custom format before using this option.
Register function to be called as part of validation process on every subshema encounter during validation.
Let's make a real-life example with this feature. Imagine you have number of transactions:

```
{
    "fromId":  1034834329,
    "toId":  1034834543,
    "amount":  200
}
```

So you write the schema:

```
{
    "type":  "object",
    "properties":  {
        "fromId":  {
            "type":  "integer"
        },
        "toId":  {
            "type":  "integer"
        },
        "amount":  {
            "type":  "number"
        }
    }
}
```

But how to check that `fromId` and `toId` are never equal. In JSON Schema Draft4 there is no possibility to do this. Actually, it's easy to just write validation code for such simple payloads. But what if you have to do the same check for many objects in different places of JSON payload. One solution is to add custom keyword `uniqueProperties` with array of property names as a value. So in our schema we would need to add:

```
"uniqueProperties":  [
    "fromId",
    "toId"
]
```

To teach `z-schema` about this new keyword we need to write handler for it:

```
function customValidatorFn(report, schema, json) {
    // check if our custom property is present
    if (Array.isArray(schema.uniqueProperties)) {
        var seenValues = [];
        schema.uniqueProperties.forEach(function (prop) {
            var value = json[prop];
            if (typeof value !== 'undefined') {
                if (seenValues.indexOf(value) !== -1) {
                    // report error back to z-schema core
                    report.addCustomError("NON_UNIQUE_PROPERTY_VALUE",
                        "Property \"{0}\" has non-unique value:  {1}",
                        [prop, value], null, schema.description);
                }
                seenValues.push(value)
            }
        });
    }
}
var validator = new ZSchema({
    // register our custom validator inside z-schema
    customValidator:  customValidatorFn
});
```

Let's test it:

```
var data = {
    fromId:  1034834346,
    toId:  1034834346,
    amount:  50
};
validator.validate(data, schema);
console.log(validator.getLastErrors())
//[ { code:  'NON_UNIQUE_PROPERTY_VALUE',
//    params:  [ 'toId', 1034834346 ],
//    message:  'Property "toId" has non-unique value:  1034834346',
//    path:  '#/',
//    schemaId:  undefined } ]
```

Note: before creating your own keywords you should consider all compatibility issues.

204.5 Benchmarks

So how does it compare to version 2.x and others?

NOTE: these tests are purely orientational, they don't consider extra features any of the validator may support and implement

`rawgithub.com/zaggino/z-schema/master/benchmark/results.html`

204.6 Contributors

Thanks for contributing to:

- `Jeremy Whitlock`
- `Oleksiy Krivoshey`

and to everyone submitting `issues` on GitHub

Chapter 205

Sprint0

205.1 ¿Cómo funciona esta primera versión?

He seguido las instrucciones de Jordi Bataller, creando una carpeta propia para MariaDB y Node.js, con sus respectivos Dockerfiles. Después, he creado un archivo `docker-compose.yml` que relaciona ambos contenedores y automatiza todo aquello que deberíamos hacer a mano.

205.1.1 MariaDB

El Dockerfile de MariaDB funciona de la siguiente manera:

- **Imagen Base**: Se utiliza la última imagen de MariaDB de la web oficial de Docker Hub (`mariadb`). Para crear esta imagen se ejecuta el comando `docker pull mariadb`.
- **Contraseña**: Se crea una variable que establece una contraseña (en esta primera versión es `1234`).
- **Inicialización**: Se ejecuta el comando para crear la base de datos (ubicada en el directorio `/sql` dentro de la misma carpeta de MariaDB) y se define la ruta del entrypoint que inicializará la base de datos en el contenedor de MariaDB.

205.1.1.1 SQL

El archivo `ejemploBBDD.sql` contiene la creación de la base de datos y la tabla que se utilizará en el proyecto. Aquí se guardará la información que se reciba de los sensores de gas. En este instante, contiene los siguientes campos:

- `id`: Es la Primary Key y es un valor que se autoincrementa.
- `hora`: Está definida por el tipo de variable `TIME` y debe ser `NOT NULL`, debido a que no puede dar un valor 0. Tendrá la siguiente estructura: "00:00", siendo el primer par de ceros las horas (de 1 a 24) y el segundo par los minutos.
- `lugar`: Está definido por el tipo de variable `VARCHAR` que puede contener hasta 255 caracteres, tampoco puede ser `NULL`.
- `id_sensor`: Es un valor entero `INT` y será el encargado de identificar a cada sensor individualmente.
- `valorGas`: Es la medición que dará el sensor de gas y está definida por un número `DECIMAL(5,2)`, que tampoco puede ser `NULL`.
- `valorTemperatura`: Es la medición que dará el sensor de temperatura. Sigue estando definida por un número `DECIMAL(5,2)` y su implementación se realizará en un futuro.

Un ejemplo de inserción de datos podría ser el siguiente:
```
INSERT INTO mediciones (id, hora, lugar, idSensor, valorGas, valorTemperatura)
VALUES (333, '12:00', 'Cartagena', 130, 55, 35);
```
Mediante la consola de MariaDB desde dentro de su contenedor, se ha comprobado de manera exitosa que los datos se insertan correctamente y que la tabla se actualiza recargando la página en tu propia máquina en local: `http://localhost:8080`

205.1.2 Node.js

Dentro de la carpeta de Node.js podrás encontrar todos los módulos que se crean al iniciar por primera vez npm con Node.js (el comando es: `npm init -y`).
El Dockerfile de Node.js funciona de la siguiente manera:

- **Imagen Base**: Se utiliza la última imagen de Node.js de la web oficial de Docker Hub (`node`). Para crear esta imagen se ejecuta el comando `docker pull node`.
- **Directorio de Trabajo**: Se crea el directorio donde se ejecutará nuestra lógica de negocio ubicada en el archivo `main.js`.
- **Copiar Archivos de Dependencias**: Se copian los archivos generados por el comando `npm init -y` (`package.json` y `package-lock.json`) desde nuestro host al sistema de archivos del contenedor Docker.
- **Instalación de Dependencias**: Para comunicarse con la base de datos de MariaDB, es necesario que se ejecute la instalación con el comando `npm install mariadb`.
- **Copiar Código**: Con `COPY . .` copiamos el resto del código de nuestra máquina host al contenedor Docker en el directorio correspondiente.
- **Exponer Puerto**: Abrimos el puerto 8080 de nuestra máquina local con el comando `EXPOSE 8080`. De esta manera podremos acceder al JSON de la base de datos con la siguiente URL: `http↩://localhost:8080`.
- **Ejecutar Aplicación**: Por último, ejecutamos en la consola nuestro `main.js` con el comando `CMD ["node", "main.js"]`.

205.1.2.1 ServidorREST

En el archivo `servidorREST.js` se ha creado un pequeño servidor REST donde se ha configurado la base de datos de MariaDB, pasándole los parámetros para poder acceder a la misma. También, se ha implementado una API con Swagger, que es una herramienta gráfica que ayuda a probar la API creada. Esta API se encarga de hacer consultas a la base de datos de manera de prueba. Puede realizar tanto GET como POST, apuntando a la tabla de mediciones (que contiene los campos: `id`, `hora`, `lugar`, `idSensor`, `valorGas`, `valorTemperatura`).
El funcionamiento de la API se puede ver de manera muy visual yendo a la siguiente dirección URL: `http↩://localhost:3000/api-docs/`.

205.1.3 docker-compose.yml

Este es un archivo que se encarga de unir tanto el Dockerfile de MariaDB como el Dockerfile de Node.js. Con este archivo, se ejecutan automáticamente ambos sin la necesidad de teclear nada por la terminal.

- **MariaDB**: Se abre un puerto en 3306 y un puerto de Docker (también el 3306). Después, crea una red interna que es esencial para que ambos contenedores funcionen correctamente sincronizados. En mi caso se llama "redsprint0". He implementado también una verificación de salud del contenedor de MariaDB. El comando `healthcheck` se encarga de pasar unos tests internos que aseguran el correcto funcionamiento del contenedor.
- **Node.js**: En el módulo de Node.js se abre también un puerto (en este caso el 8080 para que no se solape con el 3306 de MariaDB) y se crea también la misma red que hará de puente entre los dos contenedores.

Chapter 206

Indice jerárquico

206.1 Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

Chapter 207

Índice de clases

207.1 Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

Chapter 208

Indice de archivos

208.1 Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

Chapter 209

Documentación de las clases

209.1 Referencia de la Clase ServicioEnEmisora::Caracteristica

Métodos públicos

- **Caracteristica** (const char ∗nombreCaracteristica_)
- **Caracteristica** (const char ∗nombreCaracteristica_, uint8_t props, SecureMode_t permisoRead, Secure←Mode_t permisoWrite, uint8_t tam)
- void **asignarPropiedadesPermisosYTamanyoDatos** (uint8_t props, SecureMode_t permisoRead, SecureMode_t permisoWrite, uint8_t tam)
- uint16_t **escribirDatos** (const char ∗str)
- uint16_t **notificarDatos** (const char ∗str)
- void **instalarCallbackCaracteristicaEscrita** (CallbackCaracteristicaEscrita cb)
- void **activar** ()
- **Caracteristica** (const char ∗nombreCaracteristica_)
- **Caracteristica** (const char ∗nombreCaracteristica_, uint8_t props, SecureMode_t permisoRead, Secure←Mode_t permisoWrite, uint8_t tam)
- void **asignarPropiedadesPermisosYTamanyoDatos** (uint8_t props, SecureMode_t permisoRead, SecureMode_t permisoWrite, uint8_t tam)
- uint16_t **escribirDatos** (const char ∗str)
- uint16_t **notificarDatos** (const char ∗str)
- void **instalarCallbackCaracteristicaEscrita** (CallbackCaracteristicaEscrita cb)
- void **activar** ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Arduino_Fijo/PBIO_Sprint0_fijos/ServicioEnEmisora.h
- Arduino_Reales/PBIO_Sprint0/ServicioEnEmisora.h

209.2 Referencia de la Clase EmisoraBLE

Tipos públicos

- using **CallbackConexionEstablecida** = void(uint16_t connHandle)
- using **CallbackConexionTerminada** = void(uint16_t connHandle, uint8_t reason)
- using **CallbackConexionEstablecida** = void(uint16_t connHandle)
- using **CallbackConexionTerminada** = void(uint16_t connHandle, uint8_t reason)

Métodos públicos

- **EmisoraBLE** (const char ∗nombreEmisora_, const uint16_t fabricanteID_, const int8_t txPower_)
- void **encenderEmisora** ()
- void **encenderEmisora** (CallbackConexionEstablecida cbce, CallbackConexionTerminada cbct)

- void **detenerAnuncio** ()
- bool **estaAnunciando** ()
- void **emitirAnuncioIBeacon** (uint8_t ∗beaconUUID, int16_t major, int16_t minor, uint8_t rssi)
- void **emitirAnuncioIBeaconLibre** (const char ∗carga, const uint8_t tamanyoCarga)
- bool **anyadirServicio** (ServicioEnEmisora &servicio)
- bool **anyadirServicioConSusCaracteristicas** (ServicioEnEmisora &servicio)
- template<typename ... T>
  bool **anyadirServicioConSusCaracteristicas** (ServicioEnEmisora &servicio, ServicioEnEmisora::Caracteristica &caracteristica, T &... restoCaracteristicas)
- template<typename ... T>
  bool **anyadirServicioConSusCaracteristicasYActivar** (ServicioEnEmisora &servicio, T &... resto←
  Caracteristicas)
- void **instalarCallbackConexionEstablecida** (CallbackConexionEstablecida cb)
- void **instalarCallbackConexionTerminada** (CallbackConexionTerminada cb)
- BLEConnection ∗ **getConexion** (uint16_t connHandle)
- **EmisoraBLE** (const char ∗nombreEmisora_, const uint16_t fabricanteID_, const int8_t txPower_)
- void **encenderEmisora** ()
- void **encenderEmisora** (CallbackConexionEstablecida cbce, CallbackConexionTerminada cbct)
- void **detenerAnuncio** ()
- bool **estaAnunciando** ()
- void **emitirAnuncioIBeacon** (uint8_t ∗beaconUUID, int16_t major, int16_t minor, uint8_t rssi)
- void **emitirAnuncioIBeaconLibre** (const char ∗carga, const uint8_t tamanyoCarga)
- bool **anyadirServicio** (ServicioEnEmisora &servicio)
- bool **anyadirServicioConSusCaracteristicas** (ServicioEnEmisora &servicio)
- template<typename ... T>
  bool **anyadirServicioConSusCaracteristicas** (ServicioEnEmisora &servicio, ServicioEnEmisora::Caracteristica &caracteristica, T &... restoCaracteristicas)
- template<typename ... T>
  bool **anyadirServicioConSusCaracteristicasYActivar** (ServicioEnEmisora &servicio, T &... resto←
  Caracteristicas)
- void **instalarCallbackConexionEstablecida** (CallbackConexionEstablecida cb)
- void **instalarCallbackConexionTerminada** (CallbackConexionTerminada cb)
- BLEConnection ∗ **getConexion** (uint16_t connHandle)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Arduino_Fijo/PBIO_Sprint0_fijos/EmisoraBLE.h
- Arduino_Reales/PBIO_Sprint0/EmisoraBLE.h

209.3 Referencia de la Clase com.example.biometria3a.ExampleInstrumentedTest

Métodos públicos

- void **useAppContext** ()

209.3.1 Descripción detallada

Instrumented test, which will execute on an Android device.

Ver también

`Testing documentation`

La documentación para esta clase fue generada a partir del siguiente fichero:

- Android/Biometria3a/app/src/androidTest/java/com/example/biometria3a/ExampleInstrumentedTest.java

209.4 Referencia de la Clase com.example.biometria3a.ExampleUnitTest

Métodos públicos

- void **addition_isCorrect** ()

209.4.1 Descripción detallada

Example local unit test, which will execute on the development machine (host).

Ver también

`Testing documentation`

La documentación para esta clase fue generada a partir del siguiente fichero:

- Android/Biometria3a/app/src/test/java/com/example/biometria3a/ExampleUnitTest.java

209.5 Referencia de la Clase LED

Métodos públicos

- **LED** (int numero)
- void **encender** ()
- void **apagar** ()
- void **alternar** ()
- void **brillar** (long tiempo)
- **LED** (int numero)
- void **encender** ()
- void **apagar** ()
- void **alternar** ()
- void **brillar** (long tiempo)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Arduino_Fijo/PBIO_Sprint0_fijos/LED.h
- Arduino_Reales/PBIO_Sprint0/LED.h

209.6 Referencia de la Clase com.example.biometria3a.Logica

Métodos públicos

- void **guardarMedicion** (Medidas medicion)

La documentación para esta clase fue generada a partir del siguiente fichero:

- Android/Biometria3a/app/src/main/java/com/example/biometria3a/Logica.java

209.7 Referencia de la Clase com.example.biometria3a.MainActivity

Diagrama de herencias de com.example.biometria3a.MainActivity

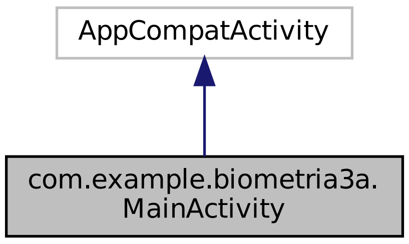
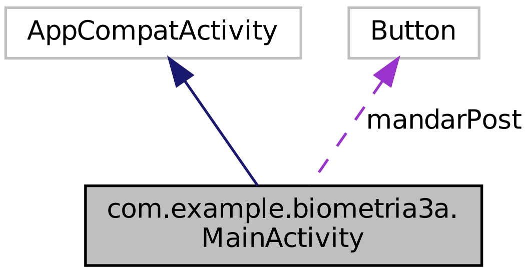


Diagrama de colaboración para com.example.biometria3a.MainActivity:



Métodos públicos

- void **botonBuscarDispositivosBTLEPulsado** (View v)
- void **botonBuscarNuestroDispositivoBTLEPulsado** (View v)
- void **botonDetenerBusquedaDispositivosBTLEPulsado** (View v)
- void **onRequestPermissionsResult** (int requestCode, String[ ] permissions, int[ ] grantResults)
- void **boton_enviar_pulsado_client** (View quien)

Atributos públicos

- Button **mandarPost**

Métodos protegidos

- void **onCreate** (Bundle savedInstanceState)

La documentación para esta clase fue generada a partir del siguiente fichero:

- Android/Biometria3a/app/src/main/java/com/example/biometria3a/MainActivity.java

209.8 Referencia de la Clase com.example.biometria3a.Medidas

Métodos públicos

- int **getMedicion** ()
- void **setMedicion** (int medicion)
- **Medidas** (int medicion, int tipoSensor, double latitud, double longitud)
- int **getTipoSensor** ()
- void **setTipoSensor** (int tipoSensor)
- double **getLatitud** ()
- void **setLatitud** (double latitud)
- double **getLongitud** ()
- void **setLongitud** (double longitud)

La documentación para esta clase fue generada a partir del siguiente fichero:

- Android/Biometria3a/app/src/main/java/com/example/biometria3a/Medidas.java

209.9 Referencia de la Clase Medidor

Métodos públicos

- void **iniciarMedidor** ()
- float **medirCO2** ()
- void **iniciarMedidor** ()
- float **medirCO2** ()
- float **medirTemperatura** ()
- float **getVgas** ()
- float **getVref** ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Arduino_Fijo/PBIO_Sprint0_fijos/Medidor_fijo.h
- Arduino_Reales/PBIO_Sprint0/Medidor.h

209.10 Referencia de la Clase com.example.biometria3a.PeticionarioREST

Diagrama de herencias de com.example.biometria3a.PeticionarioREST

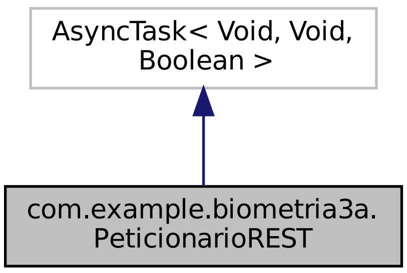
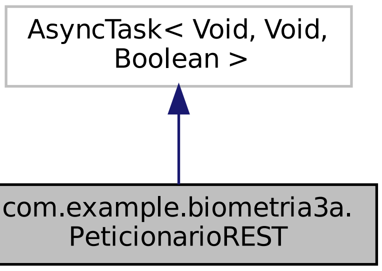

AsyncTask< Void, Void, Boolean >
com.example.biometria3a. PeticionarioREST

Diagrama de colaboración para com.example.biometria3a.PeticionarioREST:


AsyncTask< Void, Void, Boolean >
com.example.biometria3a.PeticionarioREST

Clases

- interface RespuestaREST

Métodos públicos

- void **hacerPeticionREST** (String metodo, String urlDestino, String cuerpo, RespuestaREST laRespuesta)

Métodos protegidos

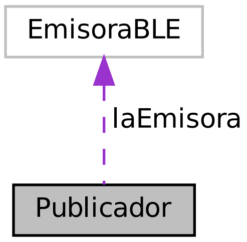
- Boolean **doInBackground** (Void... params)
- void **onPostExecute** (Boolean comoFue)

La documentación para esta clase fue generada a partir del siguiente fichero:

- Android/Biometria3a/app/src/main/java/com/example/biometria3a/PeticionarioREST.java

209.11 Referencia de la Clase Publicador

Diagrama de colaboración para Publicador:



Tipos públicos

- enum **MedicionesID** { **CO2** = 11 , **CO2** = 11 , **TEMPERATURA** = 12 , **RUIDO** = 13 }
- enum **MedicionesID** { **CO2** = 11 , **CO2** = 11 , **TEMPERATURA** = 12 , **RUIDO** = 13 }

Métodos públicos

- void **encenderEmisora** ()
- void **publicarCO2** (int16_t valorCO2, uint8_t contador, long tiempoEspera)
- void **encenderEmisora** ()
- void **publicarCO2** (int16_t valorCO2, uint8_t contador, long tiempoEspera)
- void **publicarTemperatura** (int16_t valorTemperatura, uint8_t contador, long tiempoEspera)
- void **publicarCoXTemp** (int16_t valorCox, int16_t valorTemperatura, long tiempoEspera)

Atributos públicos

- EmisoraBLE laEmisora
- const int **RSSI** = -53

209.11.1 Documentación de los datos miembro

209.11.1.1 laEmisora

```
EmisoraBLE Publicador::laEmisora
```

Valor inicial:

```
{
    "Manolito",
      0x004c,
      4
      }
```

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Arduino_Fijo/PBIO_Sprint0_fijos/Publicador.h
- Arduino_Reales/PBIO_Sprint0/Publicador.h

209.12 Referencia de la Clase PuertoSerie

Métodos públicos

- **PuertoSerie** (long baudios)
- void **esperarDisponible** ()
- template<typename T >
  void **escribir** (T mensaje)
- **PuertoSerie** (long baudios)
- void **esperarDisponible** ()
- template<typename T >
  void **escribir** (T mensaje)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Arduino_Fijo/PBIO_Sprint0_fijos/PuertoSerie.h
- Arduino_Reales/PBIO_Sprint0/PuertoSerie.h

209.13 Referencia de la Interfaz com.example.biometria3a.PeticionarioREST.RespuestaREST

Métodos públicos

- void **callback** (int codigo, String cuerpo)

La documentación para este interfaz fue generada a partir del siguiente fichero:

- Android/Biometria3a/app/src/main/java/com/example/biometria3a/PeticionarioREST.java

209.14 Referencia de la Clase ServicioEnEmisora

Clases

- class Caracteristica

Tipos públicos

- using **CallbackCaracteristicaEscrita** = void(uint16_t conn_handle, BLECharacteristic ∗chr, uint8_t ∗data, uint16_t len)
- using **CallbackCaracteristicaEscrita** = void(uint16_t conn_handle, BLECharacteristic ∗chr, uint8_t ∗data, uint16_t len)

Métodos públicos

- **ServicioEnEmisora** (const char ∗nombreServicio_)
- void **escribeUUID** ()
- void **anyadirCaracteristica** (Caracteristica &car)
- void **activarServicio** ()
- **operator BLEService &** ()
- **ServicioEnEmisora** (const char ∗nombreServicio_)
- void **escribeUUID** ()
- void **anyadirCaracteristica** (Caracteristica &car)
- void **activarServicio** ()
- **operator BLEService &** ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Arduino_Fijo/PBIO_Sprint0_fijos/ServicioEnEmisora.h
- Arduino_Reales/PBIO_Sprint0/ServicioEnEmisora.h

209.15 Referencia de la Clase com.example.biometria3a.TramaIBeacon

Métodos públicos

- byte[ ] **getPrefijo** ()
- byte[ ] **getUUID** ()
- byte[ ] **getMajor** ()
- byte[ ] **getMinor** ()
- byte **getTxPower** ()
- byte[ ] **getLosBytes** ()
- byte[ ] **getAdvFlags** ()
- byte[ ] **getAdvHeader** ()
- byte[ ] **getCompanyID** ()
- byte **getiBeaconType** ()
- byte **getiBeaconLength** ()

- **TramaIBeacon** (byte[ ] bytes)

La documentación para esta clase fue generada a partir del siguiente fichero:

- Android/Biometria3a/app/src/main/java/com/example/biometria3a/TramaIBeacon.java

209.16 Referencia de la Clase com.example.biometria3a.Utilidades

Métodos públicos estáticos

- static byte[ ] **stringToBytes** (String texto)
- static UUID **stringToUUID** (String uuid)
- static String **uuidToString** (UUID uuid)
- static String **uuidToHexString** (UUID uuid)
- static String **bytesToString** (byte[ ] bytes)
- static byte[ ] **dosLongToBytes** (long masSignificativos, long menosSignificativos)
- static int **bytesToInt** (byte[ ] bytes)
- static long **bytesToLong** (byte[ ] bytes)
- static int **bytesToIntOK** (byte[ ] bytes)
- static String **bytesToHexString** (byte[ ] bytes)

La documentación para esta clase fue generada a partir del siguiente fichero:

- Android/Biometria3a/app/src/main/java/com/example/biometria3a/Utilidades.java

Chapter 210

Documentación de archivos

210.1 EmisoraBLE.h

```
// -*- mode: c++ -*-

// ----------------------------------------------------------
// Jordi Bataller i Mascarell
// 2019-07-07
// ----------------------------------------------------------
#ifndef EMISORA_H_INCLUIDO
#define EMISORA_H_INCLUIDO

// Buena introducción: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap
// https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/

// fuente: https://www.instructables.com/id/Beaconeddystone-and-Adafruit-NRF52-Advertise-Your-/
// https://github.com/nkolban/ESP32_BLE_Arduino/blob/master/src/BLEBeacon.h

// https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/
// https://learn.adafruit.com/bluefruit-nrf52-feather-learning-guide/bleadvertising

// ----------------------------------------------------------
// ----------------------------------------------------------
#include "ServicioEnEmisora.h"

// ----------------------------------------------------------
// ----------------------------------------------------------
class EmisoraBLE {
private:

  const char * nombreEmisora;
  const uint16_t fabricanteID;
  const int8_t txPower;

public:

  // .........................................................
  // .........................................................
  using CallbackConexionEstablecida = void ( uint16_t connHandle );
  using CallbackConexionTerminada = void ( uint16_t connHandle, uint8_t reason);

  // .........................................................
  // .........................................................
  EmisoraBLE( const char * nombreEmisora_, const uint16_t fabricanteID_,
              const int8_t txPower_ )
    :
    nombreEmisora( nombreEmisora_ ) ,
    fabricanteID( fabricanteID_ ) ,
    txPower( txPower_ )
  {
    // no encender ahora la emisora, tal vez sea por el println()
    // que hace que todo falle si lo llamo en el contructor
    // ( = antes que configuremos Serial )
    // No parece que sea por el println,
    // por tanto NO_encenderEmisora();
  } // ()

  // .........................................................
  // .........................................................
  /* creo que no me sirve esta versión porque parece
que no se instalen los callbacks si la emisora no está encendida,
pero no la puedo encender en el constructor
EmisoraBLE( const char * nombreEmisora_, const uint16_t fabricanteID_,
const int8_t txPower_,
CallbackConexionEstablecida cbce,
CallbackConexionTerminada cbct
)
:
EmisoraBLE ( nombreEmisora_, fabricanteID_, txPower_ )
{
instalarCallbackConexionEstablecida( cbce );
instalarCallbackConexionTerminada( cbct );
} // ()
*/

  // ........................................................
  // ........................................................
  void encenderEmisora() {
    // Serial.println ( "Bluefruit.begin() " );
     Bluefruit.begin();

     // por si acaso:
     (*this).detenerAnuncio();
  } // ()

  // ........................................................
  // ........................................................
  void encenderEmisora( CallbackConexionEstablecida cbce,
                        CallbackConexionTerminada cbct ) {

    encenderEmisora();

    instalarCallbackConexionEstablecida( cbce );
    instalarCallbackConexionTerminada( cbct );

  } // ()

  // ........................................................
  // ........................................................
  void detenerAnuncio() {

    if ( (*this).estaAnunciando() ) {
       // Serial.println ( "Bluefruit.Advertising.stop() " );
       Bluefruit.Advertising.stop();
    }

   }  // ()

   // ........................................................
   // estaAnunciando() -> Boleano
   // ........................................................
   bool estaAnunciando() {
     return Bluefruit.Advertising.isRunning();
   } // ()

   // ........................................................
   // ........................................................
   void emitirAnuncioIBeacon( uint8_t * beaconUUID, int16_t major, int16_t minor, uint8_t rssi ) {

     //
     //
     //
     (*this).detenerAnuncio();

     //
     // creo el beacon
     //
     BLEBeacon elBeacon( beaconUUID, major, minor, rssi );
     elBeacon.setManufacturer( (*this).fabricanteID );

     //
     // parece que esto debe ponerse todo aquí
     //

     Bluefruit.setTxPower( (*this).txPower );
     Bluefruit.setName( (*this).nombreEmisora );
     Bluefruit.ScanResponse.addName(); // para que envíe el nombre de emisora (?!)

     //
     // pongo el beacon
     //
     Bluefruit.Advertising.setBeacon( elBeacon );

     //
     // ?  qué valorers poner aquí
     //
     Bluefruit.Advertising.restartOnDisconnect(true); // no hace falta, pero lo pongo
     Bluefruit.Advertising.setInterval(100, 100);    // in unit of 0.625 ms

     //
     // empieza el anuncio, 0 = tiempo indefinido (ya lo pararán)
     //
     Bluefruit.Advertising.start( 0 );

   } // ()

   // ........................................................
   //
   // Ejemplo de Beacon (31 bytes)
   //
   // https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/
   //
   // The iBeacon Prefix contains the hex data :  0x0201061AFF004C0215.  This breaks down as follows:
   //
   // 0x020106 defines the advertising packet as BLE General Discoverable and BR/EDR high-speed incompatible.
   // Effectively it says this is only broadcasting, not connecting.
   //
   // 0x1AFF says the following data is 26 bytes long and is Manufacturer Specific Data.
   //
   // 0x004C is Apple’s Bluetooth Sig ID and is the part of this spec that makes it Apple-dependent.
   //
   // 0x02 is a secondary ID that denotes a proximity beacon, which is used by all iBeacons.
   //
   // 0x15 defines the remaining length to be 21 bytes (16+2+2+1).
   //
   // Por ejemmplo:
   //
   // 1.  prefijo:  9bytes
   //       0x02, 0x01, 0x06,       // advFlags 3bytes
   //       0x1a, 0xff,             // advHeader 2 (0x1a = 26 = 25(lenght de 0x4c a 0xca)+1)   0xFF -> BLE_GAP_AD_TYPE_MANUFACTURER_SPECIFIC_DATA
   //       0x4c, 0x00,             // companyID 2bytes
   //       0x02,                   // ibeacon type 1 byte
   //       0x15,                   // ibeacon length 1 byte (dec=21 lo que va a continuación:  desde  la ’f’ hasta 0x01)
   //
   // 2.  uuid:  16bytes
   // ’f’, ’i’, ’s’, ’t’, ’r’, ’o’, ’f’, ’i’, ’s’, ’t’, ’r’, ’o’, 0xa7, 0x10, 0x96, 0xe0
   //
   // 2 major:  2bytes
   // 0x04, 0xd2,
   //
   // minor:  2bytes
   // 0x10, 0xe1,
   //
   // 0xca, // tx power :  1bytes
   //
   // 0x01, // este es el byte 31 = BLE_GAP_ADV_SET_DATA_SIZE_MAX, parece que sobra
   //
   // ........................................................
   // Para enviar como carga libre los últimos 21 bytes de un iBeacon (lo que normalmente sería uuid-16 major-2 minor-2 txPower-1)
   // ........................................................
   /*
 void emitirAnuncioIBeaconLibre( const char * carga ) {

 const uint8_t tamanyoCarga = strlen( carga );
 */
   void emitirAnuncioIBeaconLibre( const char * carga, const uint8_t tamanyoCarga ) {

     (*this).detenerAnuncio();

     Bluefruit.Advertising.clearData();
     Bluefruit.ScanResponse.clearData(); // hace falta?

     // Bluefruit.setTxPower( (*this).txPower ); creo que no lo pongo porque es uno de los bytes de la parte de carga que utilizo
     Bluefruit.setName( (*this).nombreEmisora );
     Bluefruit.ScanResponse.addName();

     Bluefruit.Advertising.addFlags(BLE_GAP_ADV_FLAGS_LE_ONLY_GENERAL_DISC_MODE);

     // con este parece que no va  !
     // Bluefruit.Advertising.addFlags(BLE_GAP_ADV_FLAG_LE_GENERAL_DISC_MODE);

     //
     // hasta ahora habrá, supongo, ya puestos los 5 primeros bytes.  Efectivamente.
     // Falta poner 4 bytes fijos (company ID, beacon type, longitud) y 21 de carga
     //
     uint8_t restoPrefijoYCarga[4+21] = {
       0x4c, 0x00, // companyID 2
       0x02, // ibeacon type 1byte
       21, // ibeacon length 1byte (dec=21)  longitud del resto // 0x15 // ibeacon length 1byte (dec=21) longitud del resto
       ’-’, ’-’, ’-’, ’-’,
       ’-’, ’-’, ’-’, ’-’,
       ’-’, ’-’, ’-’, ’-’,
       ’-’, ’-’, ’-’, ’-’,
       '-', '-', '-', '-',
       '-'
     };

     //
     // addData() hay que usarlo sólo una vez.  Por eso copio la carga
     // en el anterior array, donde he dejado 21 sitios libres
     //
     memcpy( &restoPrefijoYCarga[4], &carga[0], ( tamanyoCarga > 21 ?  21 :  tamanyoCarga ) );

     //
     // copio la carga para emitir
     //
     Bluefruit.Advertising.addData( BLE_GAP_AD_TYPE_MANUFACTURER_SPECIFIC_DATA,
                                    &restoPrefijoYCarga[0],
                                    4+21 );

     //
     // ?  qué valores poner aquí ?
     //
     Bluefruit.Advertising.restartOnDisconnect(true);
     Bluefruit.Advertising.setInterval(100, 100);    // in unit of 0.625 ms

     Bluefruit.Advertising.setFastTimeout( 1 );      // number of seconds in fast mode
     //
     // empieza el anuncio, 0 = tiempo indefinido (ya lo pararán)
     //
     Bluefruit.Advertising.start( 0 );

     Globales::elPuerto.escribir( "emitiriBeacon libre  Bluefruit.Advertising.start( 0 );  \n");
   } // ()

   // ........................................................
   // ........................................................
   bool anyadirServicio( ServicioEnEmisora & servicio ) {

     Globales::elPuerto.escribir( " Bluefruit.Advertising.addService( servicio ); \n");

     bool r = Bluefruit.Advertising.addService( servicio );

     if ( !  r ) {
       Serial.println( " SERVICION NO AÑADIDO \n");
     }


     return r;
      // nota:  uso conversión de tipo de servicio (ServicioEnEmisora) a BLEService
      // para addService()
   } // ()


   // ........................................................
   // ........................................................
   bool anyadirServicioConSusCaracteristicas( ServicioEnEmisora & servicio ) {
     return (*this).anyadirServicio( servicio );
   } //

   // ........................................................
   template <typename ...  T>
   bool anyadirServicioConSusCaracteristicas( ServicioEnEmisora & servicio,
                                              ServicioEnEmisora::Caracteristica & caracteristica,
                                              T& ...  restoCaracteristicas) {

     servicio.anyadirCaracteristica( caracteristica );

     return anyadirServicioConSusCaracteristicas( servicio, restoCaracteristicas...  );

   } // ()

   // ........................................................
   template <typename ...  T>
   bool anyadirServicioConSusCaracteristicasYActivar( ServicioEnEmisora & servicio,
                                                      // ServicioEnEmisora::Caracteristica &
   caracteristica,
                                                      T& ...  restoCaracteristicas) {

     bool r = anyadirServicioConSusCaracteristicas( servicio, restoCaracteristicas...  );

     servicio.activarServicio();

     return r;

   } // ()

   // ........................................................
   // ........................................................
   void instalarCallbackConexionEstablecida( CallbackConexionEstablecida cb ) {
     Bluefruit.Periph.setConnectCallback( cb );
   } // ()

   // .........................................................
   // .........................................................
   void instalarCallbackConexionTerminada( CallbackConexionTerminada cb ) {
     Bluefruit.Periph.setDisconnectCallback( cb );
   } // ()

   // .........................................................
   // .........................................................
   BLEConnection * getConexion( uint16_t connHandle ) {
     return Bluefruit.Connection( connHandle );
   } // ()

 }; // class

 #endif

 // ----------------------------------------------------------
 // ----------------------------------------------------------
 // ----------------------------------------------------------
 // ----------------------------------------------------------

```

210.2 EmisoraBLE.h

```
// -*- mode:  c++ -*-

// ----------------------------------------------------------
// Jordi Bataller i Mascarell
// 2019-07-07
// ----------------------------------------------------------
#ifndef EMISORA_H_INCLUIDO
#define EMISORA_H_INCLUIDO

// Buena introducción:  https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap
// https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/

// fuente:  https://www.instructables.com/id/Beaconeddystone-and-Adafruit-NRF52-Advertise-Your-/
// https://github.com/nkolban/ESP32_BLE_Arduino/blob/master/src/BLEBeacon.h

// https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/
// https://learn.adafruit.com/bluefruit-nrf52-feather-learning-guide/bleadvertising

// ----------------------------------------------------------
// ----------------------------------------------------------
#include "ServicioEnEmisora.h"

// ----------------------------------------------------------
// ----------------------------------------------------------
class EmisoraBLE {
private:

  const char * nombreEmisora;
  const uint16_t fabricanteID;
  const int8_t txPower;

public:

  // .........................................................
  // .........................................................
  using CallbackConexionEstablecida = void ( uint16_t connHandle );
  using CallbackConexionTerminada = void ( uint16_t connHandle, uint8_t reason);

  // .........................................................
  // .........................................................
  EmisoraBLE( const char * nombreEmisora_, const uint16_t fabricanteID_,
              const int8_t txPower_ )
    :
    nombreEmisora( nombreEmisora_ ) ,
    fabricanteID( fabricanteID_ ) ,
    txPower( txPower_ )
  {
    // no encender ahora la emisora, tal vez sea por el println()
    // que hace que todo falle si lo llamo en el contructor
    // ( = antes que configuremos Serial )
    // No parece que sea por el println,
    // por tanto NO_encenderEmisora();
  } // ()

  // .........................................................
  // .........................................................
  /* creo que no me sirve esta versión porque parece
que no se instalen los callbacks si la emisora no está encendida,
pero no la puedo encender en el constructor
EmisoraBLE( const char * nombreEmisora_, const uint16_t fabricanteID_,
const int8_t txPower_,
CallbackConexionEstablecida cbce,
CallbackConexionTerminada cbct
)
:
EmisoraBLE ( nombreEmisora_, fabricanteID_, txPower_ )
{
instalarCallbackConexionEstablecida( cbce );
instalarCallbackConexionTerminada( cbct );
} // ()
*/

  // .........................................................
  // .........................................................
  void encenderEmisora() {
    // Serial.println ( "Bluefruit.begin() " );
     Bluefruit.begin();

     // por si acaso:
     (*this).detenerAnuncio();
  } // ()

  // .........................................................
  // .........................................................
  void encenderEmisora( CallbackConexionEstablecida cbce,
                        CallbackConexionTerminada cbct ) {

    encenderEmisora();

    instalarCallbackConexionEstablecida( cbce );
    instalarCallbackConexionTerminada( cbct );

  } // ()

  // .........................................................
  // .........................................................
  void detenerAnuncio() {

    if ( (*this).estaAnunciando() ) {
      // Serial.println ( "Bluefruit.Advertising.stop() " );
      Bluefruit.Advertising.stop();
    }

  }  // ()

  // .........................................................
  // estaAnunciando() -> Boleano
  // .........................................................
  bool estaAnunciando() {
    return Bluefruit.Advertising.isRunning();
  } // ()

  // .........................................................
  // .........................................................
  void emitirAnuncioIBeacon( uint8_t * beaconUUID, int16_t major, int16_t minor, uint8_t rssi ) {

    //
    //
    //
    (*this).detenerAnuncio();

    //
    // creo el beacon
    //
    BLEBeacon elBeacon( beaconUUID, major, minor, rssi );
    elBeacon.setManufacturer( (*this).fabricanteID );

    //
    // parece que esto debe ponerse todo aquí
    //

    Bluefruit.setTxPower( (*this).txPower );
    Bluefruit.setName( (*this).nombreEmisora );
    Bluefruit.ScanResponse.addName(); // para que envíe el nombre de emisora (?!)

    //
    // pongo el beacon
    //
    Bluefruit.Advertising.setBeacon( elBeacon );

    //
    // ?  qué valorers poner aquí
    //
    Bluefruit.Advertising.restartOnDisconnect(true); // no hace falta, pero lo pongo
    Bluefruit.Advertising.setInterval(100, 100);    // in unit of 0.625 ms

    //
    // empieza el anuncio, 0 = tiempo indefinido (ya lo pararán)
    //
    Bluefruit.Advertising.start( 0 );

  } // ()

  // ........................................................
  //
  // Ejemplo de Beacon (31 bytes)
  //
  // https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/
  //
  // The iBeacon Prefix contains the hex data :  0x0201061AFF004C0215.  This breaks down as follows:
  //
  // 0x020106 defines the advertising packet as BLE General Discoverable and BR/EDR high-speed
  incompatible.
  // Effectively it says this is only broadcasting, not connecting.
  //
  // 0x1AFF says the following data is 26 bytes long and is Manufacturer Specific Data.
  //
  // 0x004C is Apple’s Bluetooth Sig ID and is the part of this spec that makes it Apple-dependent.
  //
  // 0x02 is a secondary ID that denotes a proximity beacon, which is used by all iBeacons.
  //
  // 0x15 defines the remaining length to be 21 bytes (16+2+2+1).
  //
  // Por ejemmplo:
  //
  // 1.  prefijo:  9bytes
  //       0x02, 0x01, 0x06,       // advFlags 3bytes
  //       0x1a, 0xff,             // advHeader 2 (0x1a = 26 = 25(lenght de 0x4c a 0xca)+1)   0xFF ->
  BLE_GAP_AD_TYPE_MANUFACTURER_SPECIFIC_DATA
  //       0x4c, 0x00,             // companyID 2bytes
  //       0x02,                   // ibeacon type 1 byte
  //       0x15,                   // ibeacon length 1 byte (dec=21 lo que va a continuación:  desde  la
  ’f’ hasta 0x01)
  //
  // 2.  uuid:  16bytes
  // ’f’, ’i’, ’s’, ’t’, ’r’, ’o’, ’f’, ’i’, ’s’, ’t’, ’r’, ’o’, 0xa7, 0x10, 0x96, 0xe0
  //
  // 2 major:  2bytes
  // 0x04, 0xd2,
  //
  // minor:  2bytes
  // 0x10, 0xe1,
  //
  // 0xca, // tx power :  1bytes
  //
  // 0x01, // este es el byte 31 = BLE_GAP_ADV_SET_DATA_SIZE_MAX, parece que sobra
  //
  // ........................................................
  // Para enviar como carga libre los últimos 21 bytes de un iBeacon (lo que normalmente sería uuid-16
  major-2 minor-2 txPower-1)
  // ........................................................
  /*
void emitirAnuncioIBeaconLibre( const char * carga ) {

const uint8_t tamanyoCarga = strlen( carga );
*/
  void emitirAnuncioIBeaconLibre( const char * carga, const uint8_t tamanyoCarga ) {

    (*this).detenerAnuncio();

    Bluefruit.Advertising.clearData();
    Bluefruit.ScanResponse.clearData(); // hace falta?

    // Bluefruit.setTxPower( (*this).txPower ); creo que no lo pongo porque es uno de los bytes de la
  parte de carga que utilizo
    Bluefruit.setName( (*this).nombreEmisora );
    Bluefruit.ScanResponse.addName();

    Bluefruit.Advertising.addFlags(BLE_GAP_ADV_FLAGS_LE_ONLY_GENERAL_DISC_MODE);

    // con este parece que no va  !
    // Bluefruit.Advertising.addFlags(BLE_GAP_ADV_FLAG_LE_GENERAL_DISC_MODE);

    //
    // hasta ahora habrá, supongo, ya puestos los 5 primeros bytes.  Efectivamente.
    // Falta poner 4 bytes fijos (company ID, beacon type, longitud) y 21 de carga
    //
    uint8_t restoPrefijoYCarga[4+21] = {
      0x4c, 0x00, // companyID 2
      0x02, // ibeacon type 1byte
      21, // ibeacon length 1byte (dec=21)  longitud del resto // 0x15 // ibeacon length 1byte (dec=21)
  longitud del resto
```

```
    '-', '-', '-', '-',
    '-', '-', '-', '-',
    '-', '-', '-', '-',
    '-', '-', '-', '-',
    '-', '-', '-', '-',
    '-'
  };

  //
  // addData() hay que usarlo sólo una vez.  Por eso copio la carga
  // en el anterior array, donde he dejado 21 sitios libres
  //
  memcpy( &restoPrefijoYCarga[4], &carga[0], ( tamanyoCarga > 21 ?  21 :  tamanyoCarga ) );

  //
  // copio la carga para emitir
  //
  Bluefruit.Advertising.addData( BLE_GAP_AD_TYPE_MANUFACTURER_SPECIFIC_DATA,
                                 &restoPrefijoYCarga[0],
                                 4+21 );

  //
  // ?  qué valores poner aquí ?
  //
  Bluefruit.Advertising.restartOnDisconnect(true);
  Bluefruit.Advertising.setInterval(100, 100);    // in unit of 0.625 ms

  Bluefruit.Advertising.setFastTimeout( 1 );      // number of seconds in fast mode
  //
  // empieza el anuncio, 0 = tiempo indefinido (ya lo pararán)
  //
  Bluefruit.Advertising.start( 0 );

  Globales::elPuerto.escribir( "emitiriBeacon libre  Bluefruit.Advertising.start( 0 );  \n");
} // ()

// .........................................................
// .........................................................
bool anyadirServicio( ServicioEnEmisora & servicio ) {

  Globales::elPuerto.escribir( " Bluefruit.Advertising.addService( servicio ); \n");

  bool r = Bluefruit.Advertising.addService( servicio );

  if ( !  r ) {
    Serial.println( " SERVICION NO AÑADIDO \n");
  }


  return r;
   // nota:  uso conversión de tipo de servicio (ServicioEnEmisora) a BLEService
   // para addService()
} // ()


// .........................................................
// .........................................................
bool anyadirServicioConSusCaracteristicas( ServicioEnEmisora & servicio ) {
  return (*this).anyadirServicio( servicio );
} //

// .........................................................
template <typename ...  T>
bool anyadirServicioConSusCaracteristicas( ServicioEnEmisora & servicio,
                                           ServicioEnEmisora::Caracteristica & caracteristica,
                                           T& ...  restoCaracteristicas) {

  servicio.anyadirCaracteristica( caracteristica );

  return anyadirServicioConSusCaracteristicas( servicio, restoCaracteristicas...  );

} // ()

// .........................................................
template <typename ...  T>
bool anyadirServicioConSusCaracteristicasYActivar( ServicioEnEmisora & servicio,
                                                   // ServicioEnEmisora::Caracteristica & caracteristica,
                                                   T& ...  restoCaracteristicas) {

  bool r = anyadirServicioConSusCaracteristicas( servicio, restoCaracteristicas...  );

  servicio.activarServicio();

  return r;

} // ()
```

```

  // .........................................................
  // .........................................................
  void instalarCallbackConexionEstablecida( CallbackConexionEstablecida cb ) {
    Bluefruit.Periph.setConnectCallback( cb );
  } // ()

  // .........................................................
  // .........................................................
  void instalarCallbackConexionTerminada( CallbackConexionTerminada cb ) {
    Bluefruit.Periph.setDisconnectCallback( cb );
  } // ()

  // .........................................................
  // .........................................................
  BLEConnection * getConexion( uint16_t connHandle ) {
    return Bluefruit.Connection( connHandle );
  } // ()

}; // class

#endif

// ----------------------------------------------------------
// ----------------------------------------------------------
// ----------------------------------------------------------
// ----------------------------------------------------------

```

210.3 LED.h

```
// -*- mode:  c++ -*-

#ifndef LED_H_INCLUIDO
#define LED_H_INCLUIDO

// ----------------------------------------------------------
// Jordi Bataller i Mascarell
// 2019-07-07
// ----------------------------------------------------------

// ----------------------------------------------------------
// ----------------------------------------------------------
void esperar (long tiempo) {
  delay (tiempo);
}

// ----------------------------------------------------------
// ----------------------------------------------------------
class LED {
private:
  int numeroLED;
  bool encendido;
public:

  // .........................................................
  // .........................................................
  LED (int numero)
    :  numeroLED (numero), encendido(false)
  {
    pinMode(numeroLED, OUTPUT);
    apagar ();
  }

  // .........................................................
  // .........................................................
  void encender () {
    digitalWrite(numeroLED, HIGH);
    encendido = true;
  }

  // .........................................................
  // .........................................................
  void apagar () {
      digitalWrite(numeroLED, LOW);
      encendido = false;
  }

  // .........................................................
  // .........................................................
  void alternar () {
    if (encendido) {
      apagar();
    } else {
      encender ();
    }
  } // ()

  // .........................................................
  // .........................................................
  void brillar (long tiempo) {
    encender ();
    esperar(tiempo);
    apagar ();
  }
}; // class

// -----------------------------------------------------------
// -----------------------------------------------------------
// -----------------------------------------------------------
// -----------------------------------------------------------
#endif
```

210.4 LED.h

```
// -*- mode:  c++ -*-

#ifndef LED_H_INCLUIDO
#define LED_H_INCLUIDO

// -----------------------------------------------------------
// Jordi Bataller i Mascarell
// 2019-07-07
// -----------------------------------------------------------

// -----------------------------------------------------------
// -----------------------------------------------------------
void esperar (long tiempo) {
  delay (tiempo);
}

// -----------------------------------------------------------
// -----------------------------------------------------------
class LED {
private:
  int numeroLED;
  bool encendido;
public:

  // .........................................................
  // .........................................................
  LED (int numero)
    :  numeroLED (numero), encendido(false)
  {
    pinMode(numeroLED, OUTPUT);
    apagar ();
  }

  // .........................................................
  // .........................................................
  void encender () {
    digitalWrite(numeroLED, HIGH);
    encendido = true;
  }

  // .........................................................
  // .........................................................
  void apagar () {
      digitalWrite(numeroLED, LOW);
      encendido = false;
  }

  // .........................................................
  // .........................................................
  void alternar () {
    if (encendido) {
      apagar();
    } else {
      encender ();
    }
  } // ()

  // .........................................................
  // .........................................................
  void brillar (long tiempo) {
    encender ();
    esperar(tiempo);
    apagar ();
  }
}; // class
```

```

// ----------------------------------------------------------
// ----------------------------------------------------------
// ----------------------------------------------------------
// ----------------------------------------------------------
#endif
```

210.5 Medidor_fijo.h

```
// -*- mode:  c++ -*-

#ifndef MEDIDOR_H_INCLUIDO
#define MEDIDOR_H_INCLUIDO
#define pin_gas 28
#define pin_vref 5
#define TIAGain 499
float sensitivityCode = -41.26;


class Medidor {
  private:

  public:
  // .....................................................
  // constructor
  // .....................................................
  Medidor(  ) {
  } // ()

  // .....................................................
  // .....................................................
  void iniciarMedidor() {
    // las cosas que no se puedan hacer en el constructor, if any
  } // ()

  float medirCO2() {
    return 2222;
  } // ()

  // .....................................................
  // .....................................................



}; // class

// ------------------------------------------------------
// ------------------------------------------------------
// ------------------------------------------------------
// ------------------------------------------------------
#endif
```

210.6 Publicador.h

```
// -*- mode:  c++ -*-

// --------------------------------------------------------------
// Jordi Bataller i Mascarell
// --------------------------------------------------------------

#ifndef PUBLICADOR_H_INCLUIDO
#define PUBLICADOR_H_INCLUIDO

// --------------------------------------------------------------
// --------------------------------------------------------------
class Publicador {

  // ............................................................
  // ............................................................
private:

  /*uint8_t beaconUUID[16] = { //UUID del Beacon // --------------------------------------------- CAMBIAR
'E', 'P', 'S', 'G', '-', 'G', 'T', 'I',
'-', 'P', 'R', 'O', 'Y', '-', '3', 'A'
};*/

uint8_t beaconUUID[16] = { //UUID del Beacon // --------------------------------------------- CAMBIAR
  'E', 'S', 'T', 'O', '-', 'E', 'S', '-',
  'U', 'N', '-', 'T', 'E', 'X', 'T', 'O'
  };


  // ...........................................................
  // ...........................................................
public:
  EmisoraBLE laEmisora {
    "Manolito", //  nombre emisora   //Nombre que aparece como dispositivo en nRFConnect ----------- CAMBIAR
      0x004c, // fabricanteID (Apple)
      4 // txPower
      };

  const int RSSI = -53;

  // ...........................................................
  // ...........................................................
public:

  // ...........................................................
  // ...........................................................
  enum MedicionesID  {  // --------------------------------------------- CAMBIAR
    CO2 = 11,
    };

  // ...........................................................
  // ...........................................................
  Publicador( ) {
    // ATENCION: no hacerlo aquí.  (*this).laEmisora.encenderEmisora();
    // Pondremos un método para llamarlo desde el setup() más tarde
  } // ()

  // ...........................................................
  // ...........................................................
  void encenderEmisora() {
    (*this).laEmisora.encenderEmisora();
  } // ()

  // ...........................................................
  // ...........................................................
  void publicarCO2( int16_t valorCO2, uint8_t contador, long tiempoEspera ) {

    //
    // 1.  empezamos anuncio
    //
    uint16_t major = (MedicionesID::CO2 « 8) + contador;
    (*this).laEmisora.emitirAnuncioIBeacon( (*this).beaconUUID,
                                            major,
                                            valorCO2, // minor
                                            (*this).RSSI // rssi
                                    );

    /*
Globales::elPuerto.escribir( "   publicarCO2():  valor=" );
Globales::elPuerto.escribir( valorCO2 );
Globales::elPuerto.escribir( "   contador=" );
Globales::elPuerto.escribir( contador );
Globales::elPuerto.escribir( "   todo="  );
Globales::elPuerto.escribir( major );
Globales::elPuerto.escribir( "\n" );
*/

    //
    // 2.  esperamos el tiempo que nos digan
    //
    esperar( tiempoEspera );

    //
    // 3.  paramos anuncio
    //
    (*this).laEmisora.detenerAnuncio();
  } // ()


}; // class

// --------------------------------------------------------------
// --------------------------------------------------------------
// --------------------------------------------------------------
// --------------------------------------------------------------
#endif
```

210.7 Publicador.h

```
// -*- mode:  c++ -*-

// --------------------------------------------------------------
// Jordi Bataller i Mascarell
// ------------------------------------------------------------

#ifndef PUBLICADOR_H_INCLUIDO
#define PUBLICADOR_H_INCLUIDO

 // ------------------------------------------------------------
 // ------------------------------------------------------------
 class Publicador {

   // ............................................................
   // ............................................................
 private:

   /*uint8_t beaconUUID[16] = { //UUID del Beacon // -------------------------------------------- CAMBIAR
 'E', 'P', 'S', 'G', '-', 'G', 'T', 'I',
 '-', 'P', 'R', 'O', 'Y', '-', '3', 'A'
 };*/

 uint8_t beaconUUID[16] = { //UUID del Beacon // -------------------------------------------- CAMBIAR
   'E', 'S', 'T', 'O', '-', 'E', 'S', '-',
   'U', 'N', '-', 'T', 'E', 'X', 'T', 'O'
   };


   // ............................................................
   // ............................................................
 public:
   EmisoraBLE laEmisora {
     "Manolito", //  nombre emisora   //Nombre que aparece como dispositivo en nRFConnect -----------
    CAMBIAR
       0x004c, // fabricanteID (Apple)
       4 // txPower
       };

   const int RSSI = -53;

   // ............................................................
   // ............................................................
 public:

   // ............................................................
   // ............................................................
   enum MedicionesID  {  // -------------------------------------------- CAMBIAR
     CO2 = 11,
     TEMPERATURA = 12,
     RUIDO = 13
   };

   // ............................................................
   // ............................................................
   Publicador( ) {
     // ATENCION: no hacerlo aquí.  (*this).laEmisora.encenderEmisora();
     // Pondremos un método para llamarlo desde el setup() más tarde
   } // ()

   // ............................................................
   // ............................................................
   void encenderEmisora() {
     (*this).laEmisora.encenderEmisora();
   } // ()

   // ............................................................
   // ............................................................
   void publicarCO2( int16_t valorCO2, uint8_t contador, long tiempoEspera ) {

     //
     // 1.  empezamos anuncio
     //
     uint16_t major = (MedicionesID::CO2 « 8) + contador;
     (*this).laEmisora.emitirAnuncioIBeacon( (*this).beaconUUID,
                                             major,
                                             valorCO2, // minor
                                             (*this).RSSI // rssi
                                     );

     /*
 Globales::elPuerto.escribir( "   publicarCO2():  valor=" );
 Globales::elPuerto.escribir( valorCO2 );
 Globales::elPuerto.escribir( "   contador=" );
 Globales::elPuerto.escribir( contador );
 Globales::elPuerto.escribir( "   todo="  );
 Globales::elPuerto.escribir( major );
 Globales::elPuerto.escribir( "\n" );
 */

     //
     // 2.  esperamos el tiempo que nos digan
     //
     esperar( tiempoEspera );

     //
     // 3.  paramos anuncio
     //
     (*this).laEmisora.detenerAnuncio();
   } // ()

   // ............................................................
   // ............................................................
   void publicarTemperatura( int16_t valorTemperatura,
                             uint8_t contador, long tiempoEspera ) {

     uint16_t major = (MedicionesID::TEMPERATURA « 8) + contador;
     (*this).laEmisora.emitirAnuncioIBeacon( (*this).beaconUUID,
                                             major,
                                             valorTemperatura, // minor
                                             (*this).RSSI // rssi
                                     );
     esperar( tiempoEspera );

     (*this).laEmisora.detenerAnuncio();
   } // ()

 //-----------------------------------------------------------------------------------------------
 // Función para publicar el valor de concentración de COx y temperatura como un anuncio Beacon.
   void publicarCoXTemp(int16_t valorCox, int16_t valorTemperatura,long tiempoEspera)
   {
   (*this).laEmisora.emitirAnuncioIBeacon( (*this).beaconUUID,
                       valorCox,
                       valorTemperatura, // minor
                       (*this).RSSI // rssi
                   );
   esperar( tiempoEspera );

   (*this).laEmisora.detenerAnuncio();
   }

 }; // class

 // --------------------------------------------------------------
 // --------------------------------------------------------------
 // --------------------------------------------------------------
 // --------------------------------------------------------------
 #endif
```

210.8 PuertoSerie.h

```

// -*- mode:  c++ -*-

// ----------------------------------------------------------
// Jordi Bataller i Mascarell
// 2019-07-07
// ----------------------------------------------------------

#ifndef PUERTO_SERIE_H_INCLUIDO
#define PUERTO_SERIE_H_INCLUIDO

// ----------------------------------------------------------
// ----------------------------------------------------------
class PuertoSerie  {

public:
  // .........................................................
  // .........................................................
  PuertoSerie (long baudios) {
    Serial.begin( baudios );
    // mejor no poner esto aquí:  while ( !Serial ) delay(10);
  } // ()

  // .........................................................
  // .........................................................
  void esperarDisponible() {

    delay(10);

  } // ()

  // .........................................................
  // .........................................................
  template<typename T>
  void escribir (T mensaje) {
    Serial.print( mensaje );
  } // ()

}; // class PuertoSerie

// ----------------------------------------------------------
// ----------------------------------------------------------
// ----------------------------------------------------------
// ----------------------------------------------------------
#endif
```

210.9 PuertoSerie.h

```

// -*- mode:  c++ -*-

// ----------------------------------------------------------
// Jordi Bataller i Mascarell
// 2019-07-07
// ----------------------------------------------------------

#ifndef PUERTO_SERIE_H_INCLUIDO
#define PUERTO_SERIE_H_INCLUIDO

// ----------------------------------------------------------
// ----------------------------------------------------------
class PuertoSerie  {

public:
  // .......................................................
  // .......................................................
  PuertoSerie (long baudios) {
    Serial.begin( baudios );
    // mejor no poner esto aquí:  while ( !Serial ) delay(10);
  } // ()

  // .......................................................
  // .......................................................
  void esperarDisponible() {

    delay(10);

  } // ()

  // .......................................................
  // .......................................................
  template<typename T>
  void escribir (T mensaje) {
    Serial.print( mensaje );
  } // ()

}; // class PuertoSerie

// ----------------------------------------------------------
// ----------------------------------------------------------
// ----------------------------------------------------------
// ----------------------------------------------------------
#endif
```

210.10 ServicioEnEmisora.h

```
// -*- mode:  c++ -*-

// ----------------------------------------------------------
// Jordi Bataller i Mascarell
// 2019-07-17
// ----------------------------------------------------------
#ifndef SERVICIO_EMISORA_H_INCLUIDO
#define SERVICIO_EMISORA_H_INCLUIDO

// ----------------------------------------------------
// ----------------------------------------------------
#include <vector>

// ----------------------------------------------------
// alReves() utilidad
// pone al revés el contenido de una array en el mismo array
// ----------------------------------------------------
template< typename T >
T *  alReves( T * p, int n ) {
  T aux;

  for( int i=0; i < n/2; i++ ) {
    aux = p[i];
    p[i] = p[n-i-1];
    p[n-i-1] = aux;
  }
  return p;
} // ()

// ----------------------------------------------------
// ----------------------------------------------------
uint8_t * stringAUint8AlReves( const char * pString, uint8_t * pUint, int tamMax ) {

    int longitudString =  strlen( pString );
    int longitudCopiar = ( longitudString > tamMax ?  tamMax :  longitudString );
    // copio nombreServicio -> uuidServicio pero al revés
    for( int i=0; i<=longitudCopiar-1; i++ ) {
      pUint[ tamMax-i-1 ] = pString[ i ];
    } // for

    return pUint;
} // ()

// ----------------------------------------------------------
// ----------------------------------------------------------
class ServicioEnEmisora {

public:


  // --------------------------------------------------------
  // --------------------------------------------------------

  // .......................................................
  // .......................................................
  using CallbackCaracteristicaEscrita = void ( uint16_t conn_handle,
                                               BLECharacteristic * chr,
                                               uint8_t * data, uint16_t len);
  // .......................................................
  // .......................................................
  class Caracteristica {
  private:
    uint8_t uuidCaracteristica[16] = { // el uuid se copia aquí (al revés) a partir de un string-c
      // least signficant byte, el primero
      '0', '1', '2', '3',
      '4', '5', '6', '7',
      '8', '9', 'A', 'B',
      'C', 'D', 'E', 'F'
    };

    //
    //
    //
    BLECharacteristic laCaracteristica;

  public:

    // .......................................................
    // .......................................................
    Caracteristica( const char * nombreCaracteristica_ )
      :
      laCaracteristica( stringAUint8AlReves( nombreCaracteristica_, &uuidCaracteristica[0], 16 ) )
    {

    } // ()

    // .......................................................
    // .......................................................
    Caracteristica( const char * nombreCaracteristica_ ,
                    uint8_t props,
                    SecureMode_t permisoRead,
                    SecureMode_t permisoWrite,
                    uint8_t tam )
      :
      Caracteristica( nombreCaracteristica_ ) // llamada al otro constructor
    {
      (*this).asignarPropiedadesPermisosYTamanyoDatos( props, permisoRead, permisoWrite, tam );
    } // ()

  private:
    // .......................................................
    // CHR_PROPS_WRITE , CHR_PROPS_READ ,  CHR_PROPS_NOTIFY
    // .......................................................
    void asignarPropiedades ( uint8_t props ) {
      // no puedo escribir AUN si el constructor llama a esto:  Serial.println( " laCaracteristica.setProperties( props ); ");
      (*this).laCaracteristica.setProperties( props );
    } // ()

    // .......................................................
    // SecureMode_t::SECMODE_OPEN  , SecureMode_t::SECMODE_NO_ACCESS
    // .......................................................
    void asignarPermisos( SecureMode_t permisoRead, SecureMode_t permisoWrite ) {
      // no puedo escribir AUN si el constructor llama a esto:  Serial.println( "laCaracteristica.setPermission( permisoRead, permisoWrite ); " );
      (*this).laCaracteristica.setPermission( permisoRead, permisoWrite );
    } // ()

    // .......................................................
    // .......................................................
    void asignarTamanyoDatos( uint8_t tam ) {
      // no puedo escribir AUN si el constructor llama a esto:  Serial.print( " (*this).laCaracteristica.setFixedLen( tam = " );
      // no puedo escribir AUN si el constructor llama a esto:  Serial.println( tam );
      // (*this).laCaracteristica.setFixedLen( tam );
      (*this).laCaracteristica.setMaxLen( tam );
    } // ()

  public:
    // .......................................................
    // .......................................................
    void asignarPropiedadesPermisosYTamanyoDatos( uint8_t props,
                                                 SecureMode_t permisoRead,
                                                 SecureMode_t permisoWrite,
                                                 uint8_t tam ) {
      asignarPropiedades( props );
      asignarPermisos( permisoRead, permisoWrite );
      asignarTamanyoDatos( tam );
    } // ()


    // .......................................................
    // .......................................................
    uint16_t escribirDatos( const char * str ) {
      // Serial.print( " return (*this).laCaracteristica.write( str  = " );
      // Serial.println( str );

      uint16_t r = (*this).laCaracteristica.write( str );

      // Serial.print( "»>Escritos " ); Serial.print( r ); Serial.println( " bytes con write() " );

      return r;
    } // ()

    // .......................................................
    // .......................................................
    uint16_t notificarDatos( const char * str ) {

      uint16_t r = laCaracteristica.notify( &str[0] );

      return r;
    } //  ()

    // .......................................................
    // .......................................................
    void instalarCallbackCaracteristicaEscrita( CallbackCaracteristicaEscrita cb ) {
      (*this).laCaracteristica.setWriteCallback( cb );
    } // ()

    // .......................................................
    // .......................................................
    void activar() {
      err_t error = (*this).laCaracteristica.begin();
      Globales::elPuerto.escribir(  " (*this).laCaracteristica.begin(); error = " );
      Globales::elPuerto.escribir(  error );
    } // ()

  }; // class Caracteristica

  // ------------------------------------------------------
  // ------------------------------------------------------
private:

  uint8_t uuidServicio[16] = { // el uuid se copia aquí (al revés) a partir de un string-c
    // least signficant byte, el primero
    '0', '1', '2', '3',
    '4', '5', '6', '7',
    '8', '9', 'A', 'B',
    'C', 'D', 'E', 'F'
  };

  //
  //
  //
  BLEService elServicio;

  //
  //
  //
  std::vector< Caracteristica * > lasCaracteristicas;

public:

  // .........................................................
  // .........................................................
  ServicioEnEmisora( const char * nombreServicio_ )
    :
    elServicio( stringAUint8AlReves( nombreServicio_, &uuidServicio[0], 16 ) )
  {

  } // ()

  // .........................................................
  // .........................................................
  void escribeUUID() {
    Serial.println ( "**********" );
    for (int i=0; i<= 15; i++) {
      Serial.print( (char) uuidServicio[i] );
    }
    Serial.println ( "\n**********" );
  } // ()

  // .........................................................
  // .........................................................
  void anyadirCaracteristica( Caracteristica & car ) {
    (*this).lasCaracteristicas.push_back( & car );
  } // ()

  // .........................................................
  // .........................................................
  void activarServicio( ) {
    // entiendo que al llegar aquí ya ha sido configurado
    // todo:  características y servicio

    err_t error = (*this).elServicio.begin();
    Serial.print( " (*this).elServicio.begin(); error = " );
    Serial.println( error );

    for( auto pCar :  (*this).lasCaracteristicas ) {
      (*pCar).activar();
    } // for

  } // ()

  // .........................................................
  // .........................................................
  operator BLEService&() {
    // "conversión de tipo":  si pongo esta clase en un sitio donde necesitan un BLEService
    return elServicio;
  } // ()

}; // class

#endif

// ----------------------------------------------------------
// ----------------------------------------------------------
// ----------------------------------------------------------
// ----------------------------------------------------------

```

210.11 ServicioEnEmisora.h

```
// -*- mode:  c++ -*-

// ----------------------------------------------------------
// Jordi Bataller i Mascarell
// 2019-07-17
// ----------------------------------------------------------
#ifndef SERVICIO_EMISORA_H_INCLUIDO
#define SERVICIO_EMISORA_H_INCLUIDO

// ----------------------------------------------------
// ----------------------------------------------------
#include <vector>

// ----------------------------------------------------
// alReves() utilidad
// pone al revés el contenido de una array en el mismo array
// ----------------------------------------------------
template< typename T >
```

```
T *  alReves( T * p, int n ) {
  T aux;

  for( int i=0; i < n/2; i++ ) {
    aux = p[i];
    p[i] = p[n-i-1];
    p[n-i-1] = aux;
  }
  return p;
} // ()

// -----------------------------------------------------
// -----------------------------------------------------
uint8_t * stringAUint8AlReves( const char * pString, uint8_t * pUint, int tamMax ) {

    int longitudString =  strlen( pString );
    int longitudCopiar = ( longitudString > tamMax ?  tamMax :  longitudString );
    // copio nombreServicio -> uuidServicio pero al revés
    for( int i=0; i<=longitudCopiar-1; i++ ) {
      pUint[ tamMax-i-1 ] = pString[ i ];
    } // for

    return pUint;
} // ()

// -----------------------------------------------------------
// -----------------------------------------------------------
class ServicioEnEmisora {

public:


  // --------------------------------------------------------
  // --------------------------------------------------------

  // .........................................................
  // .........................................................
  using CallbackCaracteristicaEscrita = void ( uint16_t conn_handle,
                                              BLECharacteristic * chr,
                                              uint8_t * data, uint16_t len);
  // .........................................................
  // .........................................................
  class Caracteristica {
  private:
    uint8_t uuidCaracteristica[16] = { // el uuid se copia aquí (al revés) a partir de un string-c
      // least signficant byte, el primero
      '0', '1', '2', '3',
      '4', '5', '6', '7',
      '8', '9', 'A', 'B',
      'C', 'D', 'E', 'F'
    };

    //
    //
    //
    BLECharacteristic laCaracteristica;

  public:

    // .........................................................
    // .........................................................
    Caracteristica( const char * nombreCaracteristica_ )
      :
      laCaracteristica( stringAUint8AlReves( nombreCaracteristica_, &uuidCaracteristica[0], 16 ) )
    {

    } // ()

    // .........................................................
    // .........................................................
    Caracteristica( const char * nombreCaracteristica_ ,
                    uint8_t props,
                    SecureMode_t permisoRead,
                    SecureMode_t permisoWrite,
                    uint8_t tam )
      :
      Caracteristica( nombreCaracteristica_ ) // llamada al otro constructor
    {
      (*this).asignarPropiedadesPermisosYTamanyoDatos( props, permisoRead, permisoWrite, tam );
    } // ()

  private:
    // .........................................................
    // CHR_PROPS_WRITE , CHR_PROPS_READ ,  CHR_PROPS_NOTIFY
    // .........................................................
    void asignarPropiedades ( uint8_t props ) {
      // no puedo escribir AUN si el constructor llama a esto:  Serial.println( "
laCaracteristica.setProperties( props ); ");
      (*this).laCaracteristica.setProperties( props );
    } // ()

    // .........................................................
    // SecureMode_t::SECMODE_OPEN  , SecureMode_t::SECMODE_NO_ACCESS
    // .........................................................
    void asignarPermisos( SecureMode_t permisoRead, SecureMode_t permisoWrite ) {
      // no puedo escribir AUN si el constructor llama a esto:  Serial.println(
  "laCaracteristica.setPermission( permisoRead, permisoWrite ); " );
      (*this).laCaracteristica.setPermission( permisoRead, permisoWrite );
    } // ()

    // .........................................................
    // .........................................................
    void asignarTamanyoDatos( uint8_t tam ) {
      // no puedo escribir AUN si el constructor llama a esto:  Serial.print( "
  (*this).laCaracteristica.setFixedLen( tam = " );
      // no puedo escribir AUN si el constructor llama a esto:  Serial.println( tam );
      // (*this).laCaracteristica.setFixedLen( tam );
      (*this).laCaracteristica.setMaxLen( tam );
    } // ()

  public:
    // .........................................................
    // .........................................................
    void asignarPropiedadesPermisosYTamanyoDatos( uint8_t props,
                                                  SecureMode_t permisoRead,
                                                  SecureMode_t permisoWrite,
                                                  uint8_t tam ) {
      asignarPropiedades( props );
      asignarPermisos( permisoRead, permisoWrite );
      asignarTamanyoDatos( tam );
    } // ()


    // .........................................................
    // .........................................................
    uint16_t escribirDatos( const char * str ) {
      // Serial.print( " return (*this).laCaracteristica.write( str  = " );
      // Serial.println( str );

      uint16_t r = (*this).laCaracteristica.write( str );

      // Serial.print( "»>Escritos " ); Serial.print( r ); Serial.println( " bytes con write() " );

      return r;
    } // ()

    // .........................................................
    // .........................................................
    uint16_t notificarDatos( const char * str ) {

      uint16_t r = laCaracteristica.notify( &str[0] );

      return r;
    } //  ()

    // .........................................................
    // .........................................................
    void instalarCallbackCaracteristicaEscrita( CallbackCaracteristicaEscrita cb ) {
      (*this).laCaracteristica.setWriteCallback( cb );
    } // ()

    // .........................................................
    // .........................................................
    void activar() {
      err_t error = (*this).laCaracteristica.begin();
      Globales::elPuerto.escribir(  " (*this).laCaracteristica.begin(); error = " );
      Globales::elPuerto.escribir(  error );
    } // ()

  }; // class Caracteristica

  // --------------------------------------------------------
  // --------------------------------------------------------
private:

  uint8_t uuidServicio[16] = { // el uuid se copia aquí (al revés) a partir de un string-c
    // least signficant byte, el primero
    '0', '1', '2', '3',
    '4', '5', '6', '7',
    '8', '9', 'A', 'B',
    'C', 'D', 'E', 'F'
  };

  //
  //
  //
  BLEService elServicio;

  //
  //
  //
  std::vector< Caracteristica * > lasCaracteristicas;

public:

  // .......................................................
  // .......................................................
  ServicioEnEmisora( const char * nombreServicio_ )
    :
    elServicio( stringAUint8AlReves( nombreServicio_, &uuidServicio[0], 16 ) )
  {

  } // ()

  // .......................................................
  // .......................................................
  void escribeUUID() {
    Serial.println ( "**********" );
    for (int i=0; i<= 15; i++) {
      Serial.print( (char) uuidServicio[i] );
    }
    Serial.println ( "\n**********" );
  } // ()

  // .......................................................
  // .......................................................
  void anyadirCaracteristica( Caracteristica & car ) {
    (*this).lasCaracteristicas.push_back( & car );
  } // ()

  // .......................................................
  // .......................................................
  void activarServicio( ) {
    // entiendo que al llegar aquí ya ha sido configurado
    // todo:  características y servicio

    err_t error = (*this).elServicio.begin();
    Serial.print( " (*this).elServicio.begin(); error = " );
    Serial.println( error );

    for( auto pCar :  (*this).lasCaracteristicas ) {
      (*pCar).activar();
    } // for

  } // ()

  // .......................................................
  // .......................................................
  operator BLEService&() {
    // "conversión de tipo":  si pongo esta clase en un sitio donde necesitan un BLEService
    return elServicio;
  } // ()

}; // class

#endif

// ---------------------------------------------------------
// ---------------------------------------------------------
// ---------------------------------------------------------
// ---------------------------------------------------------

```

210.12 Medidor.h

```
// -*- mode:  c++ -*-

#ifndef MEDIDOR_H_INCLUIDO
#define MEDIDOR_H_INCLUIDO
#define pin_gas 28
#define pin_temp 29
#define pin_vref 5
#define TIAGain 499
float sensitivityCode = -41.26;


class Medidor {
  private:

```

```
  public:
  // ....................................................
  // constructor
  // ....................................................
  Medidor(  ) {
  } // ()

  // ....................................................
  // ....................................................
  void iniciarMedidor() {
    // las cosas que no se puedan hacer en el constructor, if any
  } // ()

  float medirCO2() {
    /*float Vgas = analogRead(pin_gas);
float Vref = analogRead(pin_vref);

float M = 1 / (sensitivityCode * TIAGain * 1000 * 0.000000001);
float DiferenciaVgasVref = Vgas - Vref;
float Vtotal = DiferenciaVgasVref * 3.3 / 1024;
float ValorPPM = M * Vtotal;
float ValorPPMpor10paraBeacon = ValorPPM * 10;
return ValorPPMpor10paraBeacon;  */
    return 33;
  } // ()

  // ....................................................
  // ....................................................
  float medirTemperatura() {
       float Vtemp = analogRead(pin_temp); // Leer el valor analógico
     float voltage = (analogRead(pin_vref) * Vtemp) / ADC_RESOLUTION; // Convertir a voltaje
     float temperaturaC = voltage * 100; // Convertir el voltaje a grados Celsius (LM35:  10 mV/°C)
     return temperaturaC;
  } // ()

  float getVgas(){
    float Vgas = analogRead(pin_gas);
    return Vgas;
  }

   float getVref(){
    float Vref = analogRead(pin_vref);
    return Vref;
  }

}; // class

// ------------------------------------------------------
// ------------------------------------------------------
// ------------------------------------------------------
// ------------------------------------------------------
#endif
```