Mime

A comprehensive, compact MIME type module.

Install

NPM

```
npm install mime
```

Browser

It is recommended that you use a bundler such as webpack or browserify to package your code. However, browser-ready versions are available via skypack.dev as follows:

```
// Full version
<script type="module">
import mime from "https://
cdn.skypack.dev/mime";
</script>
```

```
// "lite" version
<script type="module">
import mime from "https://
cdn.skypack.dev/mime/lite";
</script>
```

Quick Start

For the full version (800+ MIME types, 1,000+ extensions):

```
const mime = require('mime');

mime.getType('txt');                    //
        ⇨ 'text/plain'
mime.getExtension('text/
        plain');        // ⇨ 'txt'
```

See Mime API below for API details.

Lite Version

The “lite” version of this module omits vendor-specific (`*/vnd.*`) and experimental (`*/x-*`) types. It weighs in at ~2.5KB, compared to 8KB for the full version. To load the lite version:

```
const mime = require('mime/lite');
```

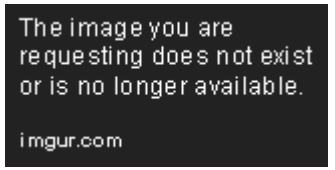

Command Line

```
mime [path_or_extension]
```

E.g.

```
> mime scripts/jquery.js
application/javascript
```

Markdown generated from src/README_js.md by



mime.getExtension(type)

Get extension for the given mime type. Charset options (often included in Content-Type headers) are ignored.

```
mime.getExtension('text/
        plain');                // ⇨
        'txt'
mime.getExtension('application/
        json');          // ⇨ 'json'
mime.getExtension('text/html;
        charset=utf8');  // ⇨ 'html'
```

mime.define(typeMap[, force = false])

Define [more] type mappings.

`typeMap` is a map of type -> extensions, as documented in `new Mime`, above.

By default this method will throw an error if you try to map a type to an extension that is already assigned to another type. Passing `true` for the `force` argument will suppress this behavior (overriding any previous mapping).

```
mime.define({'text/x-abc': ['abc',
        'abcd']});

mime.getType('abcd');             // ⇨
        'text/x-abc'
mime.getExtension('text/x-abc')   // ⇨
        'abc'
```


Mime .vs. mime-types .vs. mime-db modules

For those of you wondering about the difference between these [popular] NPM modules, here's a brief rundown …

`mime-db` is "the source of truth" for MIME type information. It is not an API. Rather, it is a canonical dataset of mime type definitions pulled from IANA, Apache, NGINX, and custom mappings submitted by the Node.js community.

`mime-types` is a thin wrapper around mime-db that provides an API drop-in compatible(ish) with `mime@<v1.3.6` API.

`mime` is, as of v2, a self-contained module bundled with a pre-optimized version of the `mime-db` dataset. It provides a simplified API with the following characteristics:

- Intelligently resolved type conflicts (See mime-score for details)
- Method naming consistent with industry best-practices
- Compact footprint. E.g. The minified+compressed sizes of the various modules:

Module	Size
`mime-db`	18 KB
`mime-types`	same as mime-db
`mime`	8 KB
`mime/lite`	2 KB

Mime API

Both `require('mime')` and `require('mime/lite')` return instances of the MIME class, documented below.

Note: Inputs to this API are case-insensitive. Outputs (returned values) will be lowercase.

new Mime(typeMap, … more maps)

Most users of this module will not need to create Mime instances directly. However if you would like to create custom mappings, you may do so as follows …

```
// Require Mime class
const Mime = require('mime/Mime');

// Define mime type -> extensions map
const typeMap = {
  'text/abc': ['abc', 'alpha', 'bet'],
  'text/def': ['leppard']
};

// Create and use Mime instance
const myMime = new Mime(typeMap);
myMime.getType('abc');            // ⇨
        'text/abc'
myMime.getExtension('text/def');  // ⇨
        'leppard'
```


If more than one map argument is provided, each map is `define()`ed (see below), in order.

mime.getType(pathOrExtension)

Get mime type for the given path or extension. E.g.

```
mime.getType('js');              // ⇨
        'application/javascript'
mime.getType('json');            // ⇨
        'application/json'

mime.getType('txt');             // ⇨
        'text/plain'
mime.getType('dir/text.txt');    // ⇨
        'text/plain'
mime.getType('dir\\text.txt');   // ⇨
        'text/plain'
mime.getType('.text.txt');       // ⇨
        'text/plain'
mime.getType('.txt');            // ⇨
        'text/plain'
```

`null` is returned in cases where an extension is not detected or recognized

```
mime.getType('foo/txt');         // ⇨
        null
mime.getType('bogus_type');      // ⇨
        null
```