

mime-types

npm v3.0.2

downloads 685.4M/month

node >=18

ci success

coverage 100%

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.

Install

This is a [Node.js](#) module available through the [npm registry](#).

Installation is done using the [npm install command](#):

```
$ npm install mime-types
```

Note on MIME Type Data and Server

This package considers the programmatic api as the server compatibility. Additionally, the package which provides the MIME data for this package (mime-db) *also* considers it's programmatic api as the server contract. This means the MIME type resolution is *not* considered in the server bumps.

In the past the version of mime-db was pinned to give two decision points when adopting MIME data changes. This is no longer true. We still update the mime-db package here as a minor release when necessary, but will use a ^ range going forward. This means that if you want to pin your mime-db data you will need to do it in your application. While this expectation was not set in docs until now, it is how the package operated, so we do not feel this is a breaking change.

If you wish to pin your mime-db version you can do that with overrides via your package manager of choice. See their documentation for how to correctly configure that.

All mime types are based on [mime-db](#), so open a PR there if you'd like to add mime types.

Adding Types

API

```
var mime = require('mime-types')
```

All functions return `false` if input is invalid or not found.

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
```

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'
```

mime.extension(type)

Get the default extension for a content-type.

```
mime.extension('application/octet-  
stream') // 'bin'
```

mime.charset(type)

Lookup the implied default charset of a content-type.

```
mime.charset('text/markdown') // 'UTF-8'
```

```
var type = mime.types[extension]
```

A map of content-types by extension.

```
[extensions...] = mime.extensions[type]
```

A map of extensions by content-type.

License

[MIT](#)