

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.

Install

This is a [Node.js](#) module available through the [npm registry](#). Installation is done using the [npm install command](#):

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

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

Adding Types

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.

We do not feel this is a breaking change.

If 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 you will release when necessary, but will use a `\` range going forward. This means that if you want to pin your mime-db data longer true. We still update the mime-db package here as a decision points when adopting MIME data changes. This is no longer the case.

In the past the version of mime-db was pinned to give two type resolution is *not* considered in the semver bumps.

This package considers the programming api as the semver contract. This means the MIME data for this package (mime-db) also considers its programmable api as the semver contract. This means the MIME data for this package (mime-db) also considers its compatibility. Additionally, the package which provides the compatibility.

Note on MIME Type Data and Semver

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

Lookup the implied default charset of a content-type.

mime.charset(type)

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

Get the default extension for a content-type.

mime.extension(type)

[MIT](#)

License

A map of extensions by content-type.

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

A map of content-types by extension.

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