# mime

Comprehensive MIME type mapping API based on mime-db module.

## Install

Install with [npm](http://github.com/isaacs/npm):

npm install mime

## Contributing / Testing

npm run test

## Command Line

mime [path\_string]

E.g.

> mime scripts/jquery.js

application/javascript

## API - Queries

### 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'

### mime.default\_type

Sets the mime type returned when mime.lookup fails to find the extension searched for. (Default is application/octet-stream.)

### mime.extension(type)

Get the default extension for type

mime.extension('text/html'); // => 'html'

mime.extension('application/octet-stream'); // => 'bin'

### 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.)

## API - Defining Custom Types

Custom type mappings can be added on a per-project basis via the following APIs.

### 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'

### 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.