# mime-db

[NPM Version](https://npmjs.org/package/mime-db) [NPM Downloads](https://npmjs.org/package/mime-db) [Node.js Version](https://nodejs.org/en/download) [Build Status](https://github.com/jshttp/mime-db/actions?query=workflow%3Aci) [Coverage Status](https://coveralls.io/r/jshttp/mime-db?branch=master)

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>

## Installation

npm install mime-db

### Database Download

If you're crazy enough to use this in the browser, you can just grab the JSON file using [jsDelivr](https://www.jsdelivr.com/). It is recommended to replace master with [a release tag](https://github.com/jshttp/mime-db/tags) as the JSON format may change in the future.

https://cdn.jsdelivr.net/gh/jshttp/mime-db@master/db.json

## Usage

var db = require('mime-db')

// grab data on .js files

var data = db['application/javascript']

## 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](http://svn.apache.org/repos/asf/httpd/httpd/trunk/docs/conf/mime.types)
  + iana - [IANA-defined media types](http://www.iana.org/assignments/media-types/media-types.xhtml)
  + nginx - [nginx media types](http://hg.nginx.org/nginx/raw-file/default/conf/mime.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.

## 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](https://en.wikipedia.org/wiki/Primary_source); links to type aggregating sites and Wikipedia are *not acceptable*.

To update the build, run npm run build.

### 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](http://tools.ietf.org/html/rfc6838#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.