

# readme.md for @push.rocks/smartmime

a module to detect mime types

## Install

Installing `@push.rocks/smartmime` is as simple as running the following npm command in your terminal. Make sure you have Node.js (version 10.x or newer) installed on your machine.

```
npm install @push.rocks/smartmime --save
```

This command will download `@push.rocks/smartmime` and add it to your project's `package.json` file.

## Usage

`@push.rocks/smartmime` is a versatile TypeScript module designed for mime type detection, supporting a variety of file types including images (JPG, PNG), text (HTML, SVG), application data (JSON), and scripts (TS, JS). Utilizing ESM syntax, this guide demonstrates its comprehensive feature set.

## Basic Mime Type Detection

To start, you can use the `detectMimeType` function to analyze a file path and return its mime type. Supported mime types include `image/jpeg`, `image/svg+xml`, `application/json`, `text/html`, etc.

```
import { detectMimeType } from '@push.rocks/smartmime';

// Example: Detecting the mime type of a JPEG image
const imagePath = 'path/to/image.jpg';
const imageMimeType = detectMimeType(imagePath);
console.log(imageMimeType); // Output: 'image/jpeg'
```

```
// Example: Detecting the mime type of an SVG
const svgPath = 'path/to/design.svg';
const svgMimeType = detectMimeType(svgPath);
console.log(svgMimeType); // Output: 'image/svg+xml'
```

## Checking if a File is Binary

Determining whether a file is binary (e.g., images, PDFs) or text-based (e.g., HTML, TypeScript) is crucial for data handling. The `isBinary` function facilitates this by returning a boolean value.

```
import { isBinary } from '@push.rocks/smartmime';

// Example: Checking if a PNG image is binary
const binaryCheckPath = 'path/to/image.png';
console.log(isBinary(binaryCheckPath)); // Output: true

// Example: Checking if a CSS file is binary
const cssPath = 'path/to/styles.css';
console.log(isBinary(cssPath)); // Output: false
```

## Getting File Encoding

Knowing a file's encoding is essential for reading or writing operations. The `getEncoding` function returns 'binary' for binary files and 'utf8' for text-based files.

```
import { getEncoding } from '@push.rocks/smartmime';

// Example: Getting encoding for a PDF document
const pdfPath = 'path/to/document.pdf';
console.log(getEncoding(pdfPath)); // Output: 'binary'

// Example: Getting encoding for a JavaScript file
const scriptPath = 'path/to/script.js';
console.log(getEncoding(scriptPath)); // Output: 'utf8'
```

## Supported File Types

`@push.rocks/smartmime` has predefined support for a select list of binary and text file types, including JSON, HTML, SVG, JPG, TS, and JS. This predefined list ensures quick integration for common file handling scenarios, but the library's core functions can be leveraged for a broader set of file types as it relies on the comprehensive `mime-types` library for mime type detection.

## Advanced Usage

Given the modular and straightforward design of `@push.rocks/smartmime`, it is seamlessly integrable into file processing pipelines, web server implementations, or any application requiring mime type detection. Its functionality extends beyond simple file type checking, allowing for sophisticated file type validations, content-based routing, or conditional processing based on detected mime types.

By utilizing TypeScript, `@push.rocks/smartmime` not only offers strong typing advantages for development time introspection but also ensures compatibility with modern JavaScript projects taking advantage of ES modules.

## Conclusion

`@push.rocks/smartmime` encapsulates a powerful set of functionalities for mime type detection, addressing common needs in file handling with an easy-to-use API. Whether for basic checks or integrated into larger processing pipelines, it stands as a valuable tool for developers needing reliable mime type detection in their Node.js or TypeScript projects.

## License and Legal Information

This repository contains open-source code that is licensed under the MIT License. A copy of the MIT License can be found in the [license](#) file within this repository.

**Please note:** The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

## Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH and are not included within the scope of the MIT license granted herein. Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines, and any usage must be approved in writing by Task Venture Capital GmbH.

# Company Information

Task Venture Capital GmbH

Registered at District court Bremen HRB 35230 HB, Germany

For any legal inquiries or if you require further information, please contact us via email at [hello@task.vc](mailto:hello@task.vc).

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

---

Revision #3

Created 2026-03-28 11:11:31 UTC by foss.global Team

Updated 2026-03-28 12:18:21 UTC by foss.global Team