Products Pricing Documentation Community

npm

Sign Up

Sign In

Search packages

Search

Have ideas to improve npm? Join in the discussion! »

node-html-to-image TS

3.1.0 • Public • Published 4 months ago

Readme

Explore BETA

3 Dependencies

8 Dependents

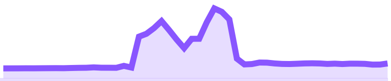
9 Versions

Install

```
> npm i node-html-to-image
```

Weekly Downloads

3,490



Version

3.1.0

License

Apache-2.0

Unpacked Size

23.8 MB

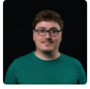
Total Files

30

Last publish

4 months ago

Collaborators



>_ Try on RunKit

Report malware

Welcome to node-html-to-image

version 3.1.0 documentation yes License Apache-2.0 Follow @yvonnickfrin 514

> A Node.js library that generates images from HTML

🏠 Homepage

Description

This module exposes a function that generates images (png, jpeg) from HTML. It uses puppeteer in headless mode to achieve it. Additionally, it embarks **Handlebars** to provide a way to add logic in your HTML.

Install

```
npm install node-html-to-image
# or
yarn add node-html-to-image
```

Usage

- Simple example
- TypeScript Support
- Options
- Setting output image resolution
- Example with Handlebars
- Dealing with images
- Using the buffer instead of saving to disk

- Generating multiple images

Simple example

```
const nodeHtmlToImage = require('node-html-to-image')

nodeHtmlToImage({
  output: './image.png',
  html: '<html><body>Hello world!</body></html>'
})
  .then(() => console.log('The image was created successfully!'))
```

TypeScript support

Types are included in the package. Enable the `esModuleInterop` compiler flag then change all references of `require` with `import` statements and you should be good to go:

```
import nodeHtmlToImage from 'node-html-to-image'
```

Options

List of all available options:

option	description	type	required
output	The ouput path for generated image	string	optional
html	The html used to generate image content	string	required
type	The type of the generated image	jpeg or png (default: png)	optional
quality	The quality of the generated image (only applicable to jpg)	number (default: 80)	optional
content	If provided html property is considered an handlebars template and use content value to fill it	object or Array	optional
waitUntil	Define when to consider markup succeded. Learn more.	string or Array (default: networkidle0)	optional

option	description	type	required
puppeteerArgs	The puppeteerArgs property let you pass down custom configuration to puppeteer. Learn more.	object	optional
beforeScreenshot	An async function that will execute just before screenshot is taken. Gives access to puppeteer page element.	Function	optional
transparent	The transparent property lets you generate images with transparent background (for png type).	boolean	optional
encoding	The encoding property of the image. Options are `binary` (default) or `base64`.	string	optional

Setting output image resolution

`node-html-to-image` takes a screenshot of the body tag's content. If you want to set output image's resolution you need to set its dimension using CSS like in the following example.

```
const nodeHtmlToImage = require('node-html-to-image')

nodeHtmlToImage({
  output: './image.png',
  html: `<html>
    <head>
      <style>
        body {
          width: 2480px;
          height: 3508px;
        }
      </style>
    </head>
    <body>Hello world!</body>
  </html>
  `
})
  .then(() => console.log('The image was created successfully!'))
```

Example with Handlebars

Handlerbars is a templating language. It generates HTML from a template and an input object. In the following example we provide a template to `node-html-to-image` and a content object to fill the template.

```
const nodeHtmlToImage = require('node-html-to-image')

nodeHtmlToImage({
  output: './image.png',
  html: '<html><body>Hello {{name}}!</body></html>',
  content: { name: 'you' }
})
  .then(() => console.log('The image was created successfully!'))
```

Handlebars provides a lot of expressions to handle common use cases like conditions or loops.

Dealing with images

If you want to display an image which is stored remotely do it as usual. In case your image is stored locally I recommend having your image in `base64`. Then you need to pass it to the template with the content property. Here is an example:

```
const nodeHtmlToImage = require('node-html-to-image')
const fs = require('fs');

const image = fs.readFileSync('./image.jpg');
const base64Image = new Buffer.from(image).toString('base64');
const dataURI = 'data:image/jpeg;base64,' + base64Image

nodeHtmlToImage({
  output: './image.png',
  html: '<html><body><img src="{{imageSource}}" /></body></html>',
  content: { imageSource: dataURI }
})
```

Using the buffer instead of saving to disk

If you don't want to save the image to disk and would rather do something with it immediately, you can use the returned value instead! The example below shows how you can generate an image and send it back to a client via using **express**.

```
const express = require('express');
const router = express.Router();
const nodeHtmlToImage = require('node-html-to-image');

router.get(`/api/tweet/render`, async function(req, res) {
  const image = await nodeHtmlToImage({
    html: '<html><body><div>Check out what I just did! #cool</div></b
  });
  res.writeHead(200, { 'Content-Type': 'image/png' });
  res.end(image, 'binary');
});
```

Generating multiple images

If you want to generate multiple images in one call you must provide an array to the content property.

Saving to disk

To save on the disk you must provide the output property on each object in the content property.

```
nodeHtmlToImage({
  html: '<html><body>Hello {{name}}!</body></html>',
  content: [{ name: 'Pierre', output: './image1.png' }, { name: 'Paul
})
  .then(() => console.log('The images were created successfully!'))
```

Using buffers

If you don't want to save the images to disk you can use the returned value instead. It returns an array of Buffer objects.

```
const images = await nodeHtmlToImage({
  html: '<html><body>Hello {{name}}!</body></html>',
  content: [{ name: 'Pierre' }, { name: 'Paul' }, { name: 'Jacques' }
})
```

Related

- node-html-to-image-cli - CLI for this module

Run tests

```
yarn test
```

Author

👤 **FRIN Yvonnick frin.yvonnick@gmail.com**

- Website: https://yvonnickfrin.dev
- Twitter: @yvonnickfrin
- Github: @frinyvonnick

🤝 Contributing

Contributions, issues and feature requests are welcome!

Feel free to check **issues page**.

Show your support

Give a ⭐ if this project helped you!

📝 License

Copyright © 2019 **FRIN Yvonnick frin.yvonnick@gmail.com**.

This project is **Apache--2.0** licensed.

---

*This README was generated with ❤️ by **readme-md-generator***

Keywords

none

Support

Help

Community

Advisories

Status

Contact npm

Company

About

Blog

Press

Terms & Policies

Policies

Terms of Use

Code of Conduct

Privacy