

## gatsby-transformer-screenshot

Plugin for creating screenshots of website URLs using an AWS Lambda Function. This plugin looks for `SitesYaml` nodes with a `url` property, and creates `Screenshot` child nodes with a `screenshotFile` field.

Live demo (source)

Data should be in a YAML file named `sites.yml` and look like:

```
- url: https://reactjs.org/
  name: React
- url: https://about.sourcegraph.com/
  name: Sourcegraph
- url: https://simply.co.za/
  name: Simply
```

### Install

```
npm install gatsby-transformer-screenshot
```

### How to use

```
// in your gatsby-config.js
module.exports = {
  plugins: [`gatsby-transformer-screenshot`],
}
```

By default, the plugin will target nodes sourced from a YAML file named `sites.yml`.

To source additional node types, supply an array of the types to a `nodeTypes` option on the plugin.

```
// in your gatsby-config.js
module.exports = {
  plugins: [
    {
      resolve: `gatsby-transformer-screenshot`,
      options: {
        nodeTypes: [`StartersYaml`, `WhateverType`],
      },
    },
  ],
}
```

### How to query

You can query for screenshot files as shown below:

```

{
  allSitesYaml {
    edges {
      node {
        url
        childScreenshot {
          screenshotFile {
            id
          }
        }
      }
    }
  }
}

```

screenshotFile is a PNG file like any other loaded from your filesystem, so you can use this plugin in combination with `gatsby-image`.

## Lambda setup

Gatsby provides a hosted screenshot service for you to use; however, you can run the service yourself on AWS Lambda.

AWS Lambda is a “serverless” computing platform that lets you run code in response to events, without needing to set up a server. This plugin uses a Lambda function to take screenshots and store them in an AWS S3 bucket.

First, you will need to create a S3 bucket for storing screenshots. Once you have done that, create a Lifecycle Policy for the bucket that sets a number of days before files in the bucket expire. Screenshots will be cached until this date.

To build the Lambda package, run `npm run build-lambda-package` in this directory. A file called `lambda-package.zip` will be generated - upload this as the source of your AWS Lambda. Finally, you will need to set `S3_BUCKET` as an environment variable for the lambda.

To set up the HTTP interface, you will need to use AWS API Gateway. Create a new API, create a new resource under `/`, select “Configure as proxy resource”, and leave all the settings with their defaults. Create a method on the new resource, selecting “Lambda Function Proxy” as the integration type, and fill in the details of your lambda.

## Placeholder image

If your site pulls a lot of screenshots it might be beneficial to use placeholder image instead of downloading and processing all the screenshots. It will help with data sourcing and query running times.

You can use placeholder image by setting `GATSBY_SCREENSHOT_PLACEHOLDER` environment variable when running `npm run develop`:

```
GATSBY_SCREENSHOT_PLACEHOLDER=true gatsby develop
```

or by using `dotenv` in your `gatsby-config.js` and adding `GATSBY_SCREENSHOT_PLACEHOLDER` to `.env.development` file in root of your project:

```
GATSBY_SCREENSHOT_PLACEHOLDER=true
```