Visual Testing with Storybook

> **Note:** Make sure that you use Storybook version `>=6.3.0` before following the instructions below. For older versions of Storybook, you can visit the Gatsby v2 documentation.

Knowing your components look as intended in every permutation is an excellent way to test them visually and provides "living documentation" for them. It makes it easier for teams to:

1. know what components are available to them in a given project and
2. what props do those components accept and what all of the states of that component are.

As your project grows over time, having this information available will be invaluable. This is the function of Storybook. Storybook is a UI development environment for your UI components. With it, you can visualize different states of your UI components and develop them interactively.

Setting up your environment

> **Note:** The following instructions are using npx. `npx` is a part of npm. In this case, it allows you to automatically generate a file/folder structure complete with the default configuration. If you're running an older version of `npm` (`<5.2.0`), you should run the following command instead: `npm install -g @storybook/cli`. You can then run `sb init` from your Gatsby root directory to initialize Storybook.

To set up Storybook, you need to install dependencies and do some custom configuration. You can get started quickly by using the automated command line tool from your Gatsby root directory:

```
npx sb init --builder webpack5
```

> **Note:** Using the builder flag will adjust Storybook's bootstrap process to include webpack 5 support out of the box.

Running this command adds the necessary Storybook dependencies, configuration files, and a set of boilerplate stories that you can browse.

Updating from a previous version

If you're upgrading from a previous Storybook version, be advised that Storybook relies on webpack 4, and Gatsby is currently supporting webpack 5. To update your Storybook version, run the following command:

```
npx sb upgrade
```

Add the following development dependencies to enable webpack 5 with Storybook:

```
npm i -D @storybook/builder-webpack5 @storybook/manager-webpack5
```

Then, update your `.storybook/main.js` to the following:

```
module.exports = {
  stories: [],
  addons: [],
  // highlight-start
  core: {
    builder: "webpack5",
  },
  // highlight-end
}
```

Configuration

Additional configuration is required to allow Gatsby's components to be manually tested with Storybook. If you want to learn more about Storybook's configuration, continue reading. If you wish to streamline the process, jump to the add-on section below.

Manual configuration

Storybook's webpack configuration will require adjustments to allow you to transpile Gatsby's source files and ensure the proper Babel plugins are used.

In your Storybook configuration file (i.e., `.storybook/main.js`) add the following:

```
module.exports = {
  webpackFinal: async config => {
    // Transpile Gatsby module because Gatsby includes un-transpiled ES6 code.
    config.module.rules[0].exclude = [/node_modules\/(?!(gatsby)\/)/]

    // Use babel-plugin-remove-graphql-queries to remove static queries from components whe
    config.module.rules[0].use[0].options.plugins.push(
      require.resolve("babel-plugin-remove-graphql-queries")
    )

    return config
```

```
  },
}
```

The final `.storybook/main.js` should look something like this:

```
module.exports = {
  // You will want to change this to wherever your Stories will live
  stories: ["../src/**/*.stories.mdx", "../src/**/*.stories.@(js|jsx|ts|tsx)"],
  addons: ["@storybook/addon-links", "@storybook/addon-essentials"],
  // highlight-start
  core: {
    builder: "webpack5",
  },
  webpackFinal: async config => {
    // Transpile Gatsby module because Gatsby includes un-transpiled ES6 code.
    config.module.rules[0].exclude = [/node_modules\/(?!(gatsby)\/)/]

    // Use babel-plugin-remove-graphql-queries to remove static queries from components whe
    config.module.rules[0].use[0].options.plugins.push(
      require.resolve("babel-plugin-remove-graphql-queries")
    )

    return config
  },
  // highlight-end
}
```

Next, inside your `.storybook/preview.js` add the following:

```
import { action } from "@storybook/addon-actions"

// Gatsby's Link overrides:
// Gatsby Link calls the `enqueue` & `hovering` methods on the global variable ___loader.
// This global object isn't set in storybook context, requiring you to override it to empty
// so Gatsby Link doesn't throw errors.
global.___loader = {
  enqueue: () => {},
  hovering: () => {},
}
// This global variable prevents the "__BASE_PATH__ is not defined" error inside Storybook.
global.__BASE_PATH__ = "/"

// Navigating through a gatsby app using gatsby-link or any other gatsby component will use
// In Storybook, it makes more sense to log an action than doing an actual navigate. Check

window.___navigate = pathname => {
  action("NavigateTo:")(pathname)
}
```

> **Note:** Storybook's `preview.js` file allows you to set up decorators, parameters, and other aspects that affect the story rendering in the Canvas. Read more about it in the official documentation.

Using an addon

You can streamline the entire configuration process by adding the `storybook-addon-gatsby` to your Gatsby project. Run the following command:

```
npm i -D storybook-addon-gatsby
```

Next, register the addon within Storybook's main configuration file (i.e., `.storybook/main.js`).

```
module.exports = {
  stories: ["../src/**/*.stories.mdx", "../src/**/*.stories.@(js|jsx|ts|tsx)"],
  addons: [
    "@storybook/addon-links",
    "@storybook/addon-essentials",
    // highlight-start
    "storybook-addon-gatsby",
    // highlight-end
  ],
}
```

TypeScript Support

Storybook v6 has out-of-the-box support for TypeScript. Your components and stories can be authored with the `.tsx` extension.

Writing stories

A complete guide to writing stories is beyond the scope of this guide, but we'll take a look at creating a story.

> **Note:** Writing stories for components that use `StaticImage` from `gatsby-plugin-image` is currently not supported. Contributions to making this possible are welcome.

First, create the story file. Storybook looks for all files with a `.stories.js` extension and loads them into Storybook for you. Generally, you will want your stories near where the component is defined. However, since this is Gatsby, you will have to create those files outside the `pages` directory if you want stories for your pages.

A good solution is to use the `stories` directory which was created when Storybook was installed and put any page stories in there.

```
import React from "react"

import Header from "./header"

export default {
  component: Header,
  title: "Components/Header",
}

const Template = args => <Header {...args} />

export const Default = Template.bind({})
Default.args = {
  siteTitle: "Something",
}
```

The example above is a straightforward story with Storybook's args. If you want to learn more about how Storybook works and what you can do with it, check out some of the resources listed below.

Additional resources

- For more information on Storybook, visit the Storybook site
- Get started with Storybook with the Intro to Storybook tutorial
- To learn more about Visual Testing, check out the Visual Testing handbook