

Shifra - Technical Manual

This is a project which has been forked off the minimal Gatsby starter which includes Bulma and Storybook. It leverages the Gatsby Contentful plugin to pull in the site content. Styling is done primarily with scss.

Features

- Storybook
- Bulma (with font-awesome icons)
- Contentful
- (S)CSS modules
- ESLint
- Jest

Install

Clone the repository and install dependencies

```
$ git clone https://github.com/ShifraIO/shifra-refresh.git
Cloning into 'shifra-refresh'...
remote: Enumerating objects: 164, done.
remote: Counting objects: 100% (164/164), done.
remote: Compressing objects: 100% (158/158), done.
remote: Total 602 (delta 85), reused 51 (delta 2), pack-reused 438
Receiving objects: 100% (602/602), 12.62 MiB | 3.03 MiB/s, done.
Resolving deltas: 100% (250/250), done.
```

```
$ cd shifra-refresh/
```

```
$ yarn
yarn install v1.9.4
$ node ./preinstall-check.js
[1/4] Resolving packages...
[2/4] Fetching packages...
```

```
[3/4] Linking dependencies...
```

```
[4/4] Building fresh packages...
Done in 73.51s.
```

Please note that this project uses yarn selective dependency resolutions, so you **must** use yarn as an npm client.

Before running the project, please update the .env file with the correct Contentful API Key and ID. You can copy the example file like so;

```
$ cp .envexample .env
```

Useful yarn commands

Execute the web application dev-server (on port 8000):

```
$ yarn develop
yarn run v1.9.4
$ gatsby develop
success delete html and css files from previous builds - 0.016 s
success open and validate gatsby-config - 0.501 s
success copy gatsby files - 0.048 s
success onPreBootstrap - 0.578 s
· Starting to fetch data from Contentful
Fetching default locale
· source and transform nodes default locale is : en-US
· source and transform nodes contentTypes fetched 6
Updated entries 148
Deleted entries 0
Updated assets 50
Deleted assets 0
Fetch Contentful data: 2423.100ms
success source and transform nodes - 2.605 s
success building schema - 0.400 s
success createLayouts - 0.071 s
success createPages - 0.091 s
success createPagesStatefully - 0.038 s
success onPreExtractQueries - 0.029 s
success update schema - 0.085 s
success extract queries from components - 0.147 s
success run graphql queries - 0.681 s
success write out page data - 0.010 s
success write out redirect data - 0.001 s
success onPostBootstrap - 0.000 s

info bootstrap finished - 8.012 s

DONE Compiled successfully in 7221ms 2:04:54 PM

You can now view gatsby-starter-bulma-storybook in the browser.

  http://localhost:8000/

View GraphiQL, an in-browser IDE, to explore your site's data and schema

  http://localhost:8000/___graphql

Note that the development build is not optimized.
To create a production build, use gatsby build

WAIT Compiling... 2:04:54 PM

DONE Compiled successfully in 950ms 2:04:55 PM
```

Run the Storybook dev-server (on port 9001):

```
$ yarn storybook
yarn run v1.9.4
$ start-storybook -p 9001 -c .storybook
info @storybook/react v3.4.3
info
info => Loading custom addons config.
info => Loading custom webpack config (full-control mode).
webpack built 7bd3498d2380ae525ce1 in 6342ms
info Storybook started on => http://localhost:9001/
info
```

Execute the tests:

```
$ yarn test
yarn run v1.9.4
$ jest
PASS src\__test__\Storyshots.test.js
  Storyshots
    Footer
      ✓ default (28ms)
    Header
      ✓ default (25ms)
    Navbar
      ✓ default (15ms)
    NavbarBrand
      ✓ default (2ms)
    NavbarDropdown
      ✓ default (1ms)
    NavbarEnd
      ✓ default (1ms)
    NavbarHamburger
      ✓ default (1ms)
    NavbarItem
      ✓ default (1ms)
    NavbarMenu
      ✓ default (1ms)
    NavbarStart
      ✓ default (1ms)
    NavDropdown
      ✓ default (1ms)

Test Suites: 1 passed, 1 total
Tests:       11 passed, 11 total
Snapshots:   11 passed, 11 total
Time:        5.791s, estimated 6s
Ran all test suites.
Done in 6.44s.
```

Run the linter:

```
$ yarn lint
yarn run v1.9.4
$ eslint src
Done in 2.31s.
```

Reformat the code:

```
$ yarn format
yarn run v1.9.4
$ prettier --write "src/**/*.js"
src\__test__\Storyshots.test.js 23ms
src\components\Footer\Footer\Footer.js 33ms
src\components\Footer\Footer\Footer.stories.js 5ms
src\components\Footer\Footer\index.js 1ms
src\components\Header\Header.js 6ms
src\components\Header\Header.stories.js 4ms
src\components\Header\index.js 2ms
src\components\Map\map-marker\MapMarker.js 11ms
src\components\Map\map-result\MapResult.js 16ms
src\components\Map\MapContainer.js 81ms
src\components\Navigation\Navbar\index.js 5ms
src\components\Navigation\Navbar\Navbar.js 60ms
src\components\Navigation\Navbar\Navbar.stories.js 6ms
src\components\Navigation\NavbarBrand\index.js 2ms
src\components\Navigation\NavbarBrand\NavbarBrand.js 6ms
src\components\Navigation\NavbarBrand\NavbarBrand.stories.js 8ms
src\components\Navigation\NavbarDropdown\index.js 2ms
src\components\Navigation\NavbarDropdown\NavbarDropdown.js 5ms
src\components\Navigation\NavbarDropdown\NavbarDropdown.stories.js 3ms
src\components\Navigation\NavbarEnd\index.js 2ms
src\components\Navigation\NavbarEnd\NavbarEnd.js 3ms
src\components\Navigation\NavbarEnd\NavbarEnd.stories.js 2ms
src\components\Navigation\NavbarHamburger\index.js 2ms
src\components\Navigation\NavbarHamburger\NavbarHamburger.js 5ms
src\components\Navigation\NavbarHamburger\NavbarHamburger.stories.js 4ms
src\components\Navigation\NavbarItem\index.js 6ms
src\components\Navigation\NavbarItem\NavbarItem.js 10ms
src\components\Navigation\NavbarItem\NavbarItem.stories.js 3ms
src\components\Navigation\NavbarMenu\index.js 2ms
src\components\Navigation\NavbarMenu\NavbarMenu.js 6ms
src\components\Navigation\NavbarMenu\NavbarMenu.stories.js 3ms
src\components\Navigation\NavbarStart\index.js 2ms
src\components\Navigation\NavbarStart\NavbarStart.js 2ms
src\components\Navigation\NavbarStart\NavbarStart.stories.js 3ms
src\components\Navigation\NavDropdown\index.js 1ms
src\components\Navigation\NavDropdown\NavDropdown.js 6ms
src\components\Navigation\NavDropdown\NavDropdown.stories.js 3ms
src\html.js 9ms
src\layouts\index.js 26ms
src\pages\404.js 4ms
src\pages\about.js 23ms
src\pages\clinic-locations.js 2ms
src\pages\index.js 6ms
src\templates\contentPage.js 50ms
src\templates\menuPage.js 35ms
src\utils\search-query.js 11ms
Done in 0.96s.
```

Build the web application in production mode:

```
$ yarn build
yarn run v1.9.4
$ gatsby build
success delete html and css files from previous builds - 0.011 s
success open and validate gatsby-config - 1.000 s
success copy gatsby files - 0.070 s
success onPreBootstrap - 2.528 s
· Starting to fetch data from Contentful
Fetching default locale
· source and transform nodesdefault local is : en-US
· source and transform nodescontentTypeTypes fetched 6
Updated entries 0
Deleted entries 0
Updated assets 0
Deleted assets 0
Fetch Contentful data: 204.422ms
success source and transform nodes - 0.286 s
success building schema - 0.365 s
success createLayouts - 0.075 s
success createPages - 0.061 s
success createPagesStatefully - 0.024 s
success onPreExtractQueries - 0.033 s
success update schema - 0.083 s
success extract queries from components - 0.173 s
success run graphql queries - 0.010 s
success write out page data - 0.009 s
success write out redirect data - 0.001 s
success onPostBootstrap - 0.000 s

info bootstrap finished - 11.781 s

success Building CSS - 13.640 s
success Building production JavaScript bundles - 10.397 s
success Building static HTML for pages - 3.566 s
info Done building in 39.389 sec
Done in 39.58s.
```

Run a local HTML server to test the web application in production mode (on port 9000):

```
$ yarn serve
yarn run v1.9.4
$ gatsby serve

UPDATE AVAILABLE: The latest version of 'serve' is 10.0.2

Serving!
- Local: http://localhost:9000
- On Your Network: http://192.168.20.11:9000
Copied local address to clipboard!
```

Important Notes

When adding new dependencies, please be aware that the current version of Gatsby does **not** support ES6 and above, as it does not implement babel. The code will run fine during development, but it **will not** compile on the production build task.

If you really need a feature included in a new dependency, please make sure it is **pre-compiled** to ES5.

Deployment

This project is built on a framework (GatsbyJS) that builds static websites. This means all of the content and pages are programmatically generated as static files at compile time. This means that when you run `'yarn server'` the Gatsby CLI will fetch any content it needs and generate the pages with this. Be aware that you **need** an internet connection for the files to be generated.

As this is a static website, it can be hosted on any service that offers static website hosting. This particular project takes advantage of Netlify.

Webhooks are used to trigger deployments when the **master** branch is updated. Netlify can also be configured to build different versions of the site from different branches (test, develop, staging, etc).

Gatsby

Gatsby is a framework for building static websites. It uses GraphQL to fetch data and programmatically build out websites. Gatsby can fetch data from markdown files, json, and even external API's.

Useful Links

- <https://www.gatsbyjs.org/>
- <https://www.gatsbyjs.org/docs/>
- <https://github.com/gvaldambrini/gatsby-starter-bulma-storybook>

Bulma

Bulma is a no JavaScript CSS framework that takes a mobile first approach to development. By extending the provided styles it provides a great framework for mobile friendly views that also look great on desktop clients.

Useful Links

- <https://bulma.io/>
- <https://bulma.io/documentation/>

Contentful

Contentful is a headless Content Management System (CMS). It allows complete freedom of data structures and types and provides a GraphQL endpoint that makes fetching data incredibly easy.

Contentful is used in Shifra to manage the different content and menu options, as well as the different languages it supports. It is worth reviewing the different types in the Shifra Contentful project and how they relate to the GraphQL queries included in this project.

Useful Links

- <https://www.contentful.com/>
- <https://www.contentful.com/r/knowledgebase/contentful-101/>
- <https://www.contentful.com/guides/>

GraphQL

GraphQL is an alternative to RESTful API's. It provides the front-end code complete control of the data that it requests. No RESTful endpoints need to be built for different data, as each request is made to the GraphQL endpoint using a GraphQL specific query language.

GraphQL is the main method used in the project for fetching the content.



Useful Links

- <https://graphql.org/>
- <https://graphql.org/learn/>

Netlify

Netlify is a super simplistic static website hosting service.

Useful Links

- <https://www.netlify.com/>
- <https://www.netlify.com/docs/>

React

React is a component focussed JavaScript library for building user interfaces. GatsbyJS is a static website generator that is built around React.

Useful Links

- <https://reactjs.org/>
- <https://reactjs.org/docs/>
- <https://reactjs.org/docs/getting-started.html>
- <https://reactjs.org/tutorial/tutorial.html>