Colombia Observatory of the Digital Economy

Technical Documentation

Overview

The Colombia Observatory site is a combination of front-end user experience and back-end data. In order to simplify hosting and use modern website development technique, the site uses the VueJS framework for a reactive user experience and Google Sheets for easy to understand data storage.

VueJS is part of a family of modern web frameworks that change the user interface by reacting to changes in the data or changes in state initiated by the user (e.g. selecting menu items or clicking buttons). For this project, VueJS is used to as a Single Page Application (SPA) to completely encapsulate all site functions on the client. This enables easy hosting by simply installing the files on any web server. All static content on the site is neatly separately into configuration files for each editing.

Google Sheets is an economic and efficient way to store tabular data, almost identical to a standard Excel spreadsheet. This means the Observatory data used by the site can be managed in its natural form as a spreadsheet. Adding and editing is easy without resorting to additional database forms. Google Sheets allows applications to subscribe to the spreadsheet via dynamic CSV export. It's dynamic because changes to the spreadsheet are almost immediately available in the export.

The VueJS application uses the GoogeSheet export to pull the data into the site. Any changes to the data on Google Sheets are immediately available to the VueJS site via a simply browser reload.

The following sections provide more detail on the site implementation, data specifications and hosting requirements.

Implementation Details

The source code for the site is stored on GitHub in a private repository at:

https://github.com/cleverglue/colombia

Note: *This repository should be transferred to MinTIC*.

A current snapshot (as of 12/17/2017) of the site for direct installation on a production server is available at:

http://colombia-test.s3-website-us-east-1.amazonaws.com/colombia-observatory.zip

Organization

The application files are organized using the standard VueJS 2 template. Please refer to Vue2 documentation for details at https://vuejs.org/v2/guide/.

Key Directories and Files

The `src` directory contains all the source files for the site. VueJS uses a component architecture and the individual components can be found in `/src/components`. Some of these components are configured using data from the `/src/lib/data` directory.

Navigation

Overall site navigation is managed by the `/src/components/NavBar.vue` and `/src/components/SubNav.vue` components which uses `/src/lib/data/nav.js` to configure the various navigation elements and their links.

Text Pages

The text pages of the site are managed by the `/src/components/Static.vue` component which uses `/src/lib/data/static.js` to configure which text appears. The `static.js` file references local images (charts and tables) that are stored in the `/static/images` directory.

Link Pages

The surveys and links pages contain list of links to other resources. The pages are managed by the `/src/components/Links.vue` component which uses `/src/lib/data/links.js` to configure the text and URLs of the links. While the surveys pages link to external sites, the links page links to local documents stored in the `/static/docs` directory.

Chart Pages

The chart pages (sector, region and company size) are managed by the `/src/components/Chart.vue` component which uses `/src/lib/chart/bar.js` to to connect to the ChartJS library used for charting services. Only a bar chart is used. All chart data is pulled from Google Sheets and therefore is not part of the code.

Routing

All URL routing is handled by the VueJS framework in `/src/router/index.js`.

Setup

This site uses the npm package manager to handle dependencies, run a local server of development, run tests, and build the final distribution to deployment to production servers. Npm must already be installed locally in order to run the following commands.

After cloning the source code from GitHub, all dependencies should be installed via:

```
npm install
```

Once all dependencies are installed, a local server can be started to enable development and local review of the site:

```
npm run dev
```

A production build of the site creates/updates a `/dist` (for distribution) directory with all the files necessary to run the site on a server:

```
npm run build
```

Development

Developing using the VueJS framework is the same as any normal Javascript development work. The .vue files contain the html, css and javascript for each component. The separate .js files in the `src/lib` directory contain code shared among several components and additional data structures and code separated from the .vue files for readability.

The following sections describe some expected scenarios for site changes and how to handle them.

Change Home Page

The home page is managed by the `/src/components/Home.vue` component. In this file is everything referenced on the home page. Note that all pages use the `/src/components/Page.vue` template to wrap every page with the top navigation bar and a footer.

Change Static Text and Images

The static text on the Visión and Medición pages are managed by the `/src/components/Static.vue` file in terms of layout and by the `/lib/data/static.js` file in terms of content. Content changes, which must be simple HTML, can be made in `/lib/data/static.js` without needing to change any code in `/src/components/Static.vue`. Any images referenced in /lib/data/static.js should be placed in the `/static/images` directory.

Change Navigation

The top navigation bar (main nav links and their sub nav links) are controlled by `/src/components/NavBar.vue` (layout) and `/src/lib/nav.js` (content). The lib file contains a JSON representation of the navigation. Editing this file will change the navigation text and structure. Make sure that all navigation in this file has a valid route so the application know where to go. Here is an example entry for the first nav section:

```
{
  title: 'Visión',
  route: 'Static',
  levels: [
    { title: 'Que es la economía digital?' },
    { title: 'Tendencias mundiales de digitalización de las economías' },
    { title: 'La importancia económica de la digitalización' },
    { title: 'La economía digital y Colombia' }
  ]
}
```

Add New Pages

New static text pages can be added by adding new entries in `/src/lib/nav.js` and matching entries in `/src/lib/static.js`. The entries in /src/lib/static.js are arrays that match the navigation route seen in the URL for a page. For example, *The Visión > La economía digital y Colombia* has the URL /page/1/4 because it is section 1 (Visión) and subsection 4 (La economía digital y Colombia). The content for that page in `/src/lib/static.js` is stored in the 4th entry of the 1st array.

Add New Data

All data for charts and tables comes from Google Sheets. Please see the section below on Data for instructions on modifying the statistical data on the site.

Deployment

As mentioned in the Setup, building the site for production use begins with running the `npm run build` command in order to create/update the `/dist` directory with the latest code. The `/dist` directory will then contain a new `index.html` file and a `static` directory with all the assets and code to run the site. This directory should then be copied to the appropriate place on a production server that is hosting the site.

Data

All data represented in the charts and tables for the Sector, Región and Dimensiones sections of the site are drawn from CSV files hosted on Google Drive. These CSV files are sheets created and managed via Google Docs/Sheets. Any changes made to the Google spreadsheet is reflected in the CSV used by the site. Every time the user navigates to a different level (of the 4 available levels) of the above sections, the CSV is pulled to include the latest data.

The Google spreadsheet require authentication to access directly, but not via CSV so the application does not require any special authentication. The spreadsheet should be moved to a MinTIC Google Drive account. It is current hosted at:

https://docs.google.com/spreadsheets/d/1hBGzynQZlG6evDlLZoIpGBGVdeoIEIv7UrKKUQUGnXs/edit#gid=75109715

It is crucial to preserve the structure of the Google spreadsheet since the application code depends on it. Rows can be added, deleted or modified, but the column order and title MUST remain the same. Furthermore, the row values for chart dimensions (Año, Region Geografica, Sector, Sector Economico, Sub sector, Tamaño de Empresa) must be consistent. The application is coded specifically for those values. The only new values supported are for Año. Any other changed, such as a new sub sector or change to the name of a sub sector, would require an appropriate code change in `/src/lib/chart-nav.js`.

There is an individual CSV file for each of the 4 tabs in the spreadsheet. The URLs for these are identified in `/src/lib/data/chart-nav.js`. The `chart-nav.js` file also contains the name, description, allowed years and index names for each CSV. Furthermore, this file also manages the internal (as seen in spreadsheet) and external (as seen in application user interface) names for all the selectable dimensions for controlling chart display.

Hosting

In order to host the site, the files in the `/dist` directory after run `npm run build` must be copied to the host. Once this are installed, navigating to the path containing the `index.html` file should access the site. Keep in mind that data is pulled from Google Sheets, so the server must have access to google services.