RZZT CIC Website Documentation

Mozart Olbrycht-Palmer

6 March 2019

Contents

Chapter 1

Introduction

The website of RZZT CIC has developed and grown significantly since the current framework was launched in February 2018.

A website can fulfil many purposes. In broad terms, a website for an organisation like RZZT is intended to provide information about the organisation and its activities to a general audience. However, because the Company places an emphasis on transparency, a significant number of records are published via the website.

The website is a static website that is compiled using Jekyll. Jekyll is an exceptionally versatile Ruby-based platform for creating websites. Although it has usually been used as a framework for blogging, it has enormous potential for processing other types of data, and can be used to create sophisticated static websites.

The source files for the website are hosted by GitHub, which Netlify accesses to recompile and serve the website automatically when changes are made.

As a result of the policy of transparency and the power of Jekyll to process different sorts of data, the website has become not merely a way of presenting information, but of storing and processing it too.

The source files for the website include files that generate 'normal' content — pages, blog posts, press releases and announcements. However, they also include files that generate accounting records, minutes of meetings and decisions, structural documents, and even a timeline function. The website not only publishes this information, it *generates it*. The accounts, for example, are entirely generated using Jekyll.

This book, with the invigorating title 'RZZT CIC Website Documentation', covers all the custom functionality that has been incoporated to make such a website possible, and increase the bus factor beyond one.

Mozart Olbrycht-Palmer
Director & Company Secretary
6 March 2019

Chapter 2

Overview

It is not necessary to know the intimate details of how Jekyll or the underlying technologies work. Different people will have different purposes when modifying the website, whether that be publishing a press release or updating the accounting data. Most information can be found online. However, it is useful to have an overview.

Chapter 3

Pages

Pages are sections of the website that can be used to display information about the Company or for grouping other content, such as blog posts.

3.1 Making a page

Standard pages are made by creating a Markdown file somewhere in the website source directory. This can be top-level or it can be in a subdirectory (or subdirectories):

- `website/page.markdown`
- `website/pages/page.markdown`
- `website/pages/category/page.markdown`

Note that the path of the file will determine its location (URL) on the website, unless a permalink is specified in the file itself (see below). In the above examples, without a permalink the pages would be accessible at:

- `rzzt.io/page.html`
- `rzzt.io/pages/page.html`
- `rzzt.io/pages/category/page.html`

3.2 Standard page frontmatter

Each page requires frontmatter at the start of the file to determine its layout and title, and other parameters. This is in YAML format and will typically look like this:

```
---
layout: page
title: "Page Title"
---
```

It is a good idea to put the page title in inverted commas to help Jekyll process it without being confused by apostrophes and special characters.

3.3 Page URL

The frontmatter of a page can also be used to specify its URL, which will override its location in the source files:

```
---
layout: page
title: "Page Title"
permalink: "/page-url/"
---
```

This will create a page that can be accessed by visiting:

- `rzzt.io/page-url/index.html`; or
- `rzzt.io/page-url/`

3.4 Adding a page to the menu

The RZZT website is able to construct a sitewide navigation menu that supports pages nested to three levels deep. It does this by using the YAML frontmatter of each page to determine whether it should be included and in what order:

```
---
layout: page
title: "A"
permalink: "/page-a/"
parent: "Home"
subpages: "yes"
order: 1
---
```

Any page with the parent 'Home' will be placed at the top level of the menu.

If other pages should be nested under it in the menu, specify that it has subpages as illustrated in the example. That line can be omitted if no nesting is needed and, even if it is included, will be ignored below the third level.

The 'order' parameter determines the position of the page in relation to other pages with the same parent: in the example above, this page will be positioned after any page with the order value '0' and before any page with the order value '2'.

3.4.1 Nesting pages

Pages can be nested in the menu by specifying the title of the parent page in the current page's frontmatter. In the following example, this page will be nested below the page with the title 'A':

```
---
layout: page
title: "B"
permalink: "/page-b/"
parent: "A"
subpages: "yes"
order: 1
---
```

Pages can be nested to one level further than the example above. Any page with the parent 'B' will be nested below the above page.

3.4.2 Diagrammatic example of nested pages

- `title: "Our Work"`
  `parent: "Home"`
  `subpages: "yes"`
  `order: 0`

  - `title: "Funding"`
    `parent: "Our Work"`
    `subpages: "yes"`
    `order: 0`

    * `title: "Human Rights & Civil Liberties"`
      `parent: "Funding"`

  - `title: "Awards"`
    `parent: "Our Work"`
    `subpages: "yes"`
    `order: 1`

    * ...

  - `title: "Our Work"`
    `parent: "Our Work"`
    `subpages: "yes"`
    `order: 2`

    * ...

- `title: "Get Involved"`
  `parent: "Home"`
  `subpages: "yes"`
  `order: 1`

  - ...

- `title: "About"`
  `parent: "Home"`
  `subpages: "yes"`
  `order: 2`

  - ...

3.5 Page content

Page content begins after the frontmatter and is written using standard Markdown syntax:

```
---
layout: page
title: "Page Title"
---

*Cum sociis natoque** penatibus et magnis dis parturient montes, [nascetur
ridiculus mus](https://rzzt.io/another-page/). Vestibulum _id ligula porta_
felis euismod semper.
```

Chapter 4

Posts

A post is a piece of content that is created and published on a regular basis. They are similar to pages, but are better suited to publishing information sorted by date order, such as blog posts, announcements and press releases.

4.1 Making a post

Posts are made by creating a Markdown file in the posts directory:

- `website/_posts/2018-01-01-a-post.markdown`
- `website/_posts/2018-07-01-another-post.markdown`
- `website/_posts/2019-01-01-yet-another-post.markdown`

4.2 Post frontmatter

Like pages, all posts require frontmatter at the start of the file to determine their layout and title, as well as their date. The post frontmatter can also include the author, categories, and an excerpt. This is in YAML format and will typically look like this:

```
---
layout: post
title: "Post Title"
date: YYYY-MM-DD
author: "Post Author"
categories:
- category1
- category2
excerpt: "Sed posuere consectetur est at lobortis. Cum sociis natoque penatibus
et magnis dis parturient montes, nascetur ridiculus mus. Lorem ipsum dolor
sit amet, consectetur adipiscing elit."
---
```

4.3 Blog posts, announcements and press releases

All posts are published at `rzzt.io/blog/` regardless of category, but adding the category 'blog' helps to identify blog posts. Posts with the category 'announcement' are published at `rzzt.io/announcements/` and posts with the category 'media' (for press releases) are published at `rzzt.io/media/`. The relevant category ('blog', 'announcement' or 'media') should be placed first on a post's list of categories. Other categories may be added as appropriate.

Chapter 5

Timelines

Timelines are a feature that can be used to illustrate the progress of something, such as the development of a project, the history of an organisation, or the progress toward a funding goal.

A page or post can include a timeline in its YAML frontmatter (refer to the previous chapters):

```
---
layout: page
title: "Project Name"
permalink: "/project-name/"
timeline:
  complete: false
  milestones:
  - milestone: "Milestone Description"
    date: YYYY-MM-DD
  - milestone: "Another Milestone Description"
    date: YYYY-MM-DD
---
```

In this example, the timeline is not complete and so 'false' is specified. If the timeline has completed (for example, the project has been finished) this should be replaced with 'true'.

A timeline can be included anywhere in the body of the page or post using the following Liquid tag:

```
{% include timeline.html %}
```

Only one timeline per page or post is supported.

Chapter 6

Data files

Jekyll can use data files for various purposes. The Jekyll documentation for this feature is extensive and won't be explained here, but two data files are used to store information about RZZT's officers and positions vacant that should be explained.

6.1 Officers

The file `officers.yml` contains entries formatted as follows:

```
- role: "Director"
  surname: "Sem"
  forename: "Nibh"
  bio: "Maecenas sed diam eget risus varius blandit sit amet non magna."
```

Each officer

6.2 Positions vacant