Contents

The MGAPS website compiler

This is the official manual for the `mgaps-website` compiler.

This compiler takes source material and turns it into a static website, automatic as many tasks as possible for easiest maintainability. The resulting website works on both desktop *and* mobile.

If you have any questions, feel free to e-mail me at laurent.renedecotret@mail.mcgill.ca.

Usage

The `mgaps-website` compiler is a command-line program with a few options. An overview of the available commands is available as follows:

```
> mgaps-website --help
Usage: mgaps-website.exe [-v|--verbose] COMMAND
  mgaps-website.exe - Static site compiler created with Hakyll

Available options:
  -h,--help                Show this help text
  -v,--verbose             Run in verbose mode
```

```
Available commands:
  build                    Generate the site
  check                    Validate the site output
  clean                    Clean up and remove cache
  deploy                   Upload/deploy your site
  rebuild                  Clean and build again
  server                   Start a preview server
  watch                    Autocompile on changes and start a preview server.
```

Building the website

The most useful command is `build`, used to compile the website source into a built website:

```
> mgaps-website build
Initialising...
  Creating store...
  Creating provider...
  Running rules...
Checking for out-of-date items
Compiling
  updated templates/default.html
  (... omitted ...)
  updated static/workspace/fridges.md
  updated sitemap.xml
  updated static/index.html
Success
```

This command will update the last built website version into a folder called `/_rendered/`. The compiler will only look at items that have changed; to trigger a rebuild from scratch, you can use `mgaps-website rebuild`.

Testing the website

Once the website has been built using `mgaps-website build` or `mgaps-website rebuild`, the rendered website can be checked for common mistakes (such as nonexistent links) using the `check` command:

```
> mgaps-website check
Checking file _rendered/workspace.html
(... omitted ...)
Checking file _rendered/finances.html
Checking file _rendered/events.html
```

The rendered website should be checked before publishing!

The website source

The `mgaps-website` compiler expects certain things from your website source. You need at least:

- A `static/` directory, containing static website content;
  - A `static/quick-links/` directory containing quick links information;
- An `announcements/` directory containing announcements (more on this later);
- A `template/` directory containing HTML templates;
- An `image/` directory containing images;
  - An `image/profile/` directory containing profile pictures;
- `people/council/` and `people/officers/` directories containing profiles (more on this later);
- `js/` and `css/` directory containing the appropriate files;

Updating static content

Static content is written in Markdown format. Using this format, you have access to many features, including links, images, bullet lists, tables, code blocks, quotes, and many more.

To update a static page, simply edit the source Markdown file. `mgaps-website` will compile Markdown source to HTML during compilation.

Statis pages have metadata at the top. At this time, the following metadata is available:

- title (required): title of the page. This title will appear in the top banner.
- withtoc (optional): if `withtoc: yes`, a table of content will be automatically inserted at the top of the page.
- tocdepth (optional): Table of content depth. For example, if `tocdepth: 2`, level 1 and level 2 headings are taken into account in the generation of the table of content.
- contact (optional): the `contact` field specifies what MGAPS VP or officer is responsible for this page. This field must match at least one of the VPs or Officers. Contact information will be automatically added to the page.

Here's an example:

```
---
title: Test Page
contact: President
withtoc: yes
tocdepth: 4
---

# Level 1 heading
```

Updating profiles

`mgaps-website` will automatically create a webpage called `people.html` based on the content of the `people/council/` and `people/officers/` directories.

The files in these directories should all be Markdown files with the following metadata:

- name (required): Person name;
- email (required): email address;
- position (required): Position within MGAPS. If a person holds more than one position, many profiles will have to be created.
- picture (optional): path to a profile picture, usually of the form `images/profiles/something.jpg`.

The body of the file is anything that will appear as profile description. Here's an example of a profile:

```
---
name: Mohit Bhardwaj
email: work.mohitbhardwaj@gmail.com
position: VP Academic
picture: images/profiles/mohit-bhardwaj.jpg
---

As the VP Academic, it is my job to contribute to a positive academic
experience for MGAPS members.
```

The name of the files is completely irrelevant. Profiles can be added by adding a new markdown file. The distinction between `people/council/` and `people/officers/` is only so that the `people.html` is easier to navigate.

Profiles NOT IN `people/council/` or `people/officers/` will be ignored.

Linking to a profile

An anchor is automatically created for every profile. For example, a profile with `position: President` can be reached at the location `people.html#President`, while a profile with `position: VP Academic` can be reached at location `people.html#VP Academic`.

In case where there are two positions with the same name (e.g. two PGSS representatives), the anchor will be placed on the profile that comes alphabetically first.

Updating announcements

To create a new announcements, add a markdown file to the directory `announcements/`. `mgaps-website` will create one announcement per markdown file in `announcements/`. An announcement file can have the following metadata:

- title (required): title of the announcement;
- date (required): date of the announcement **in the YYYY-MM-DD format**;
- updated (optional): date of the last update to the announcement **in the YYYY-MM-DD format**;
- summary (optional): summary of the announcement. This summary will appear on front page `index.html`.

Here's an example of an announcement:

```
---
title: Example announcement
date: 2018-11-29
updated: 2018-11-30
summary: This is the summary of this announcement.
---

Mauris in lorem nisl. Maecenas tempus facilisis ante,
eget viverra nisl tincidunt et. Donec turpis lectus,
mattis ac malesuada a, accumsan eu libero. Morbi condimentum,
tortor et tincidunt ullamcorper, sem quam pretium nulla,
id convallis lectus libero nec turpis. Proin dapibus nisi
id est sodales nec ultrices tortor pellentesque.
```

Announcements will be sorted by date. Only the most recent annoucements will be shown on the front page (`index.html`), but all announcements are visible at `announcements.html`. This page is linked to the front page.

Updating quick links

The front page `index.html` has five quick links. These links are specified in Markdown files, in the `static/quick-links/` directory, with the following metadata:

- title (required): Title of the quick link
- link (required): Link to the resource

An example of a quick-link file is presented below:

```
---
name: McGill Society of Physics Students
link: http://msps.sus.mcgill.ca/
---

McGill's offical organisation representing the undergraduates in physics
```

The name of the files in `static/quick-links/` directory are not important. However, strange things might happen if the number of quick links changes to anything else than **five**.

Updating files

Files can be hosted on this website, as long as they are stored in the `files/` folder. Any file in this folder will be included in the website as-is. These files can be linked to from any Markdown file using the link syntax, for example:

```
...
The GA minutes are available [here](/files/ga_minutes_2018.pdf)
...
```

Updating images

Images must be stored in the `images/` directory. Subdirectories, like `images/profiles/`, are also supported. There are two cases:

- JPEG images (`*.jpg` and `*.jpeg`) are compressed automatically;
- Non-JPEG images are copied as-is.

I urge you to use JPEGs as much as possible, especially for pictures/photos, as the compression pipeline in `mgaps-website` is very space-efficient.

What cannot be changed

For ease of use, the `mgaps-website` compiler forces certain things to stay the same. These are:

- The overall page template cannot be changed. This includes navigation bar, top banners, and page footer. `mgaps-website` will generated the layout `templates/default.html` automatically, and all changes to it will be overwritten;
- The website schema, or website map. Navigation links (visible in the navigation bar) are generated by the compiler;
- The order of profiles in `people.html`. For technical reasons, the only guaranteed order is that the President comes first.

Aknowledgements

This manual has been built using the pandoc LaTeX template Eisvogel.