frog-doc

Frank Smith

13 June 2013

frog

Introduction

Increasingly more software projects require project documentation written for low level components. Writing these types of documents manually is not feasible due to the changing nature of source code.

Generated documents often look unprofessional due to the lack of template application. frog-doc tries to address some of these concerns by using a few tools together.

- `pandoc`
- LaTeX
- `sed`

Use cases include:

- Application Programming Interface docs
- Automatic release note generation based on specially crafted `VCS` log messages
- Complex document that can be authored by many people at a time
- The need to simultaneously target many formats such as `html`, `epub`, `mobi`, `pdf`, `docx`

How to use this document

Choose one of the examples below which best matches your needs and use it as a base.

Requirements

You need the following:

- `pandoc`
- A basic LaTeX distribution such as `TeXLive`
- A Unix tool chain (Windows support will be added in future if there is a need)

Examples

The following examples serve to show how automation can be uses to write different classes of documents. Each document will be produced using

First you may want to check out some examples, this is what the example branches are for:

Simple

The most simple example is this file itself. It's not a very complicated document and the hierarchy is pretty well defined. We don't make use of intra doc references or have the need for any fancy tables:

```
$ git checkout eg-simple
$ ./write_documentation
```

Peruse the contents of `write_documentation.sh`

Like what you see? Go print that doc (or put it on your tablet/reader, you tree hugger) so you can follow along for the rest of the examples without having to open this file.

Logo

```
$ git checkout eg-logo
$ ./write_documentation
```

Glorious, my company logo!

How it's done

Skip this section if you don't care how the saussage is made

Okay, so we want to put the company logo in our file, or the customer's logo? No problem, we need to tweak the latex that `pandoc` spits out is all. We could give `pandoc` a fancy new LaTeXtemplate if we're up to it. I'm not up to it.

Let's make use of the `titlepic` package and some `sed` magic to manipulate the LaTeXthat `pandoc` spits out.

1. We make a file for custom included packages for `pandoc`: `custom_includes.tex`.
2. In it we put `\include{titlepic}`
3. Instruct `pandoc` to include that file in generated `tex`.

4. Manipulate the generated `tex` file to include the logo how and when we want it.

   1. Include the `titlepage` option in the article class.
   2. Insert the `\titlepic{\includegraphics{../frog_log.pdf}}` statement right after the doc date