Documentation Basics Documentation

Release 0.1

Eric Holscher

Apr 21, 2017

Contents

When a user comes to your documentation, it's important to present them with a set of information that is useful to them. This will hopefully be a good starting point of what to include in your documentation. Either on the front page, or linked in an obvious fashion.

This is a living document, and you should help contribute to it. The code lives at Github, and we accept pull requests with love.

CHAPTER 1

What to include

What problem your project solves.

A lot of people will come to your docs trying to figure out what exactly your project is. Someone will mention it, or they'll google a phrase randomly. You should explain what your project does and why it exists. Fabric does a great job of this.

A small code example

Show a telling example of what your project would normally be used for. Requests does a great example of this.

A link to your code & issue tracker

People like to browse the code sometimes. They might be interested in filing bugs against the code for issues they've found. Make it really easy for people who want to contribute back to the project in any way possible. I think the Python Guide does a good job with the link to the code portion.

A FAQ

A lot of people have the same problems. If things happen all the time, you should probably fix your documentation or the code, so that the problems go away. However, there are always questions that get asked about your project, things that can't be changed, etc. Document those, and **keep it up to date**. FAQs are generally out of date, but when done well, they are a golden resource. Tastypie did a great job with this, with their "Cookbook" concept.

How to get support

Mailing list? IRC Channel? Document how to get help and interact with the community around a project. Django does a great job with this.

Information for people who want to contribute back

People usually have standards for how they expect things to be done in their projects. You should document these so that if people write code, they can do things in the norm of the project. Open Comparison does a great job of this.

Installation instructions

Once people figure out whether they want to use your code or not, they need to know how to actually get it and make it run. Hopefully your install instructions should be a couple lines for the basic case. A page that gives more information and caveats should be linked from here if necessary. I think at Read the Docs we do a good job with this.

Your project's license

BSD? MIT? GPL? This stuff might not matter to you, but the people who want to use your code will care about this a whole lot. Think about what you want to accomplish with your license, and please only pick one of the standard licenses that you see around the web.

Contents: