
Documentation
and
Open-Source

When Brandon asked me to do a presentation on the role of documentation in open-source, I wasn't sure what to make it on, whether it'd be open source tooling or documentation best practices.

So I started thinking about what I would talk about the issue from the very top:

Before I talk about what documentation does for open source, I want to first philosophize on why we do open-source at all.

And I’m not talking about the pros and cons, or the successes associated with open-source software. I mean, more fundamentally, what’s the point of sitting down and shipping some open-source project?

Is the main reason to work on OSS to consider it “good practice” for your “real job”? Maybe. If you code in the open, you’ll get feedback (and critique) on your style from all sorts of internet trolls.

That will in turn force you to apply those same skills to your closed-source office job. Or maybe…

Maybe we’re motivated to do open-source because it quells our guilt. Maybe we believe in open systems because it’s the right thing to do, you can’t put a price tag on knowledge, share and share alike, fuck the man, information wants to be free…

… I just felt my beard grow a little bit there.

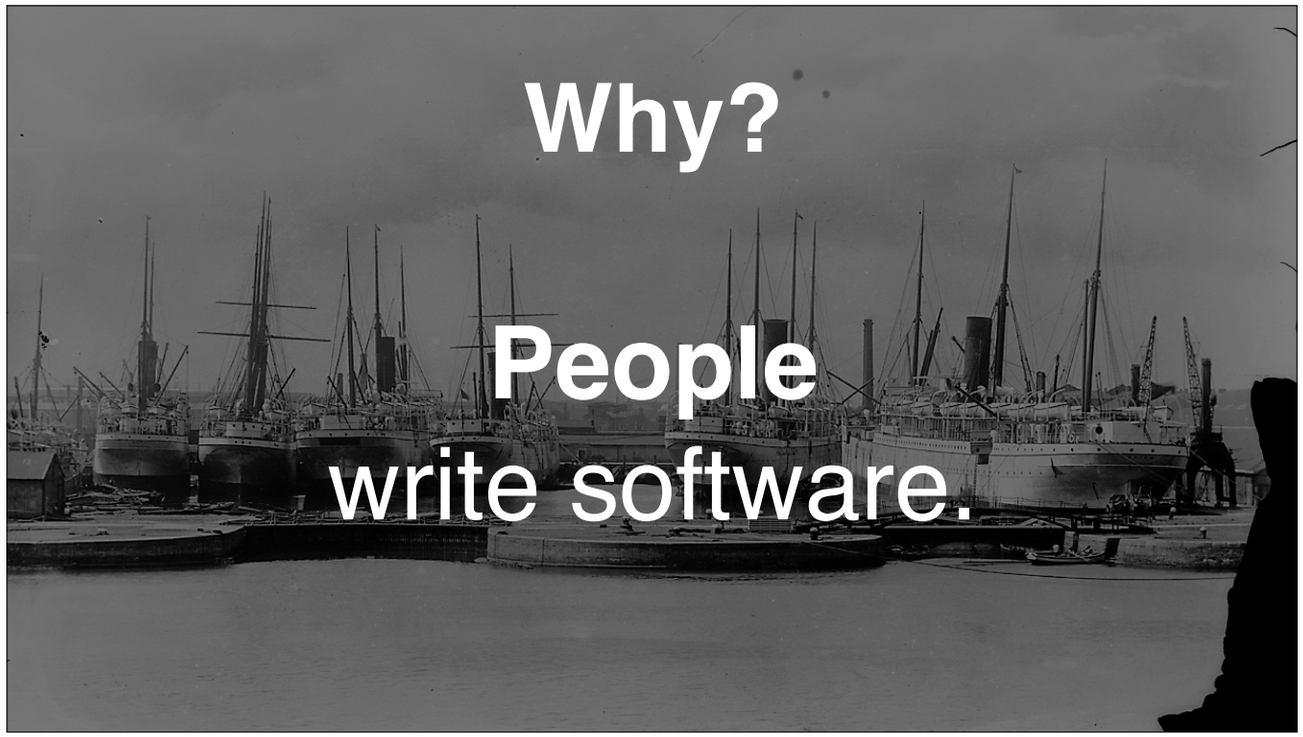
If you’re doing OSS to achieve fame and fortune and recognition, you’re probably doing it for the wrong reason.

No, I think the real motivator behind doing open source is…

To help. More specifically:

To help people.


Why?
People
use software.



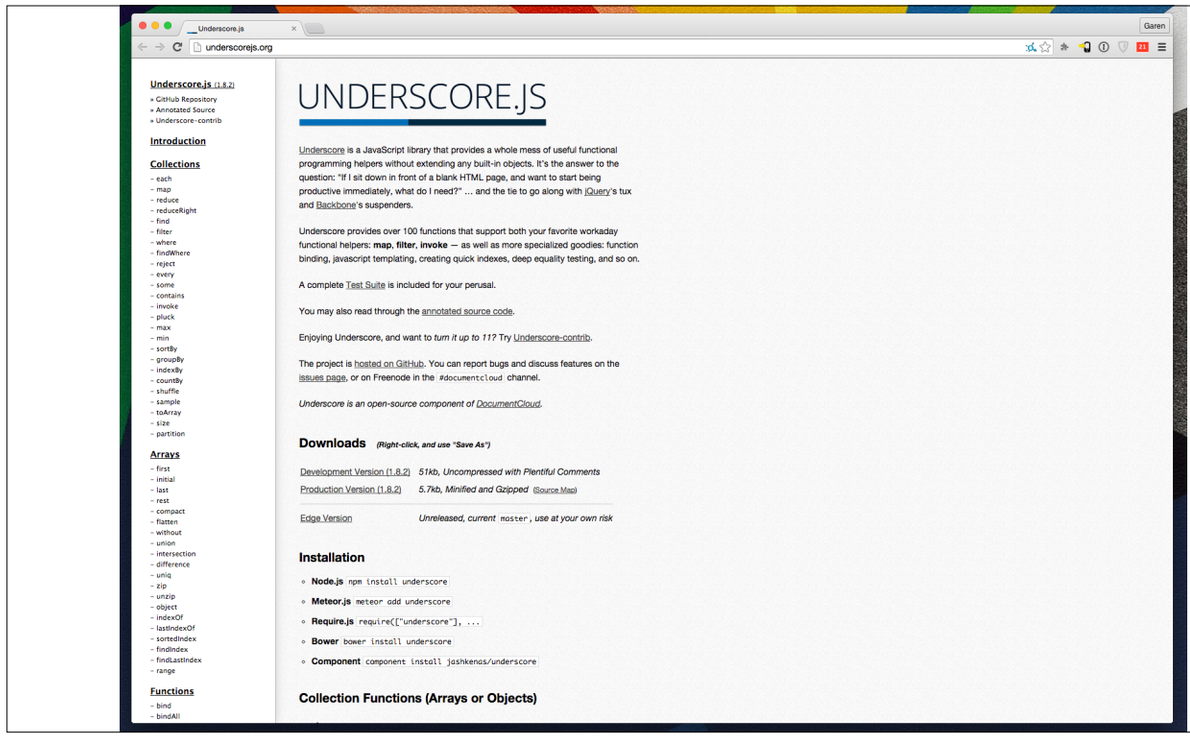
When we talk about open-source projects gaining momentum in banks, in governments, in corporations, what we're really talking about are people that make up those organizations using open source software.

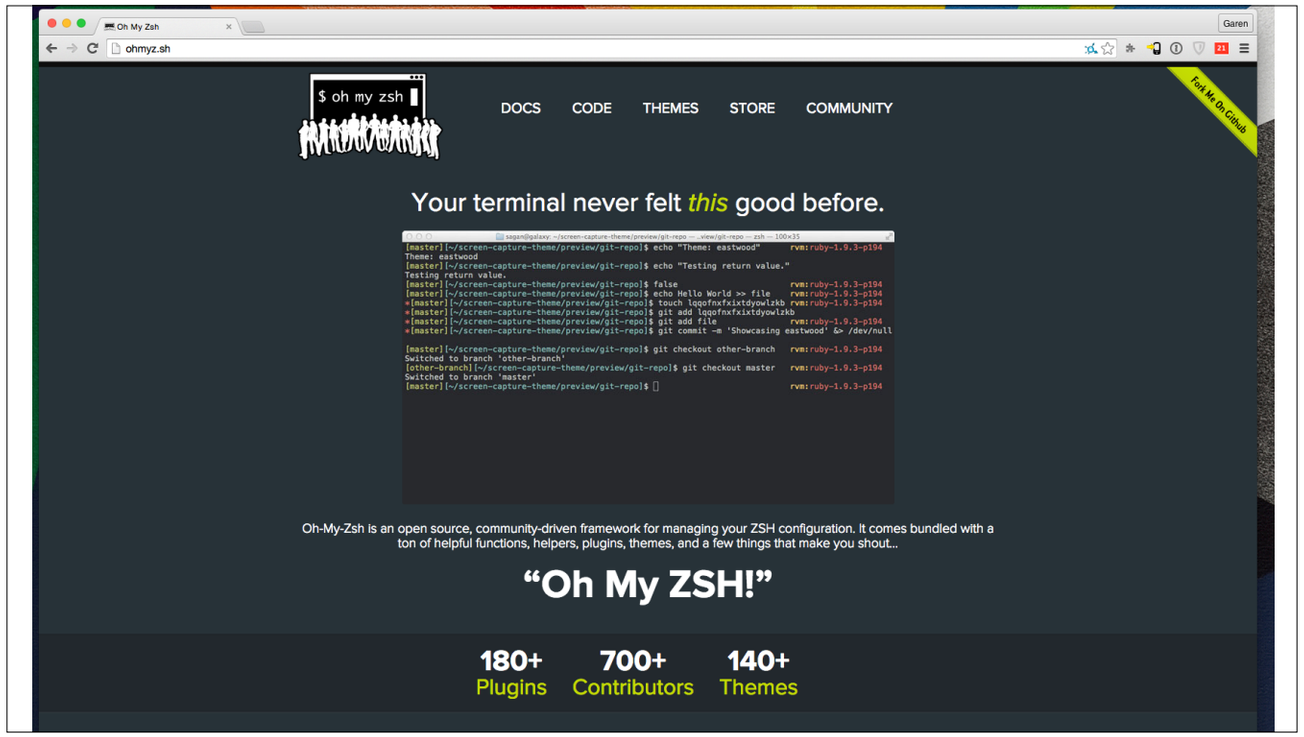
Just as regular software is consumed by users, the main consumer of open source is people writing software.

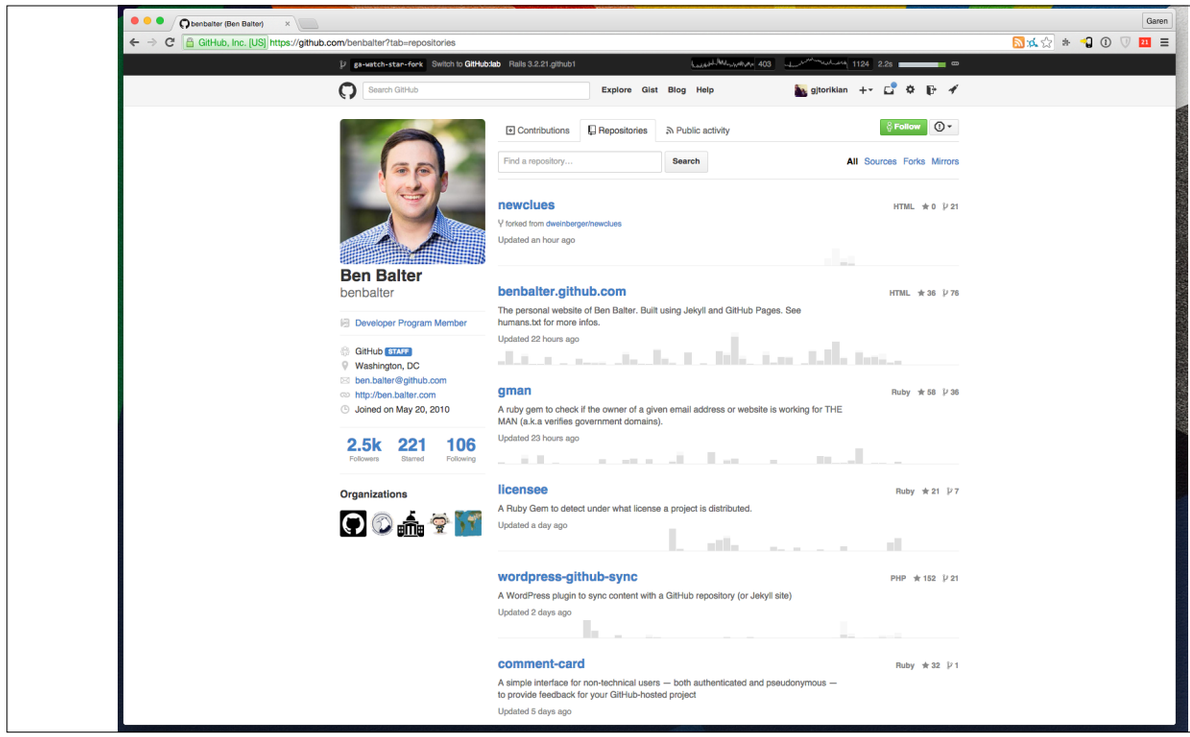
We write open source software for a variety of reasons. Perhaps there's an interesting problem to be solved, or you're dissatisfied with a set of existing solutions, but ultimately, I think, the best open source projects are those that are motivated by wanting to help people.

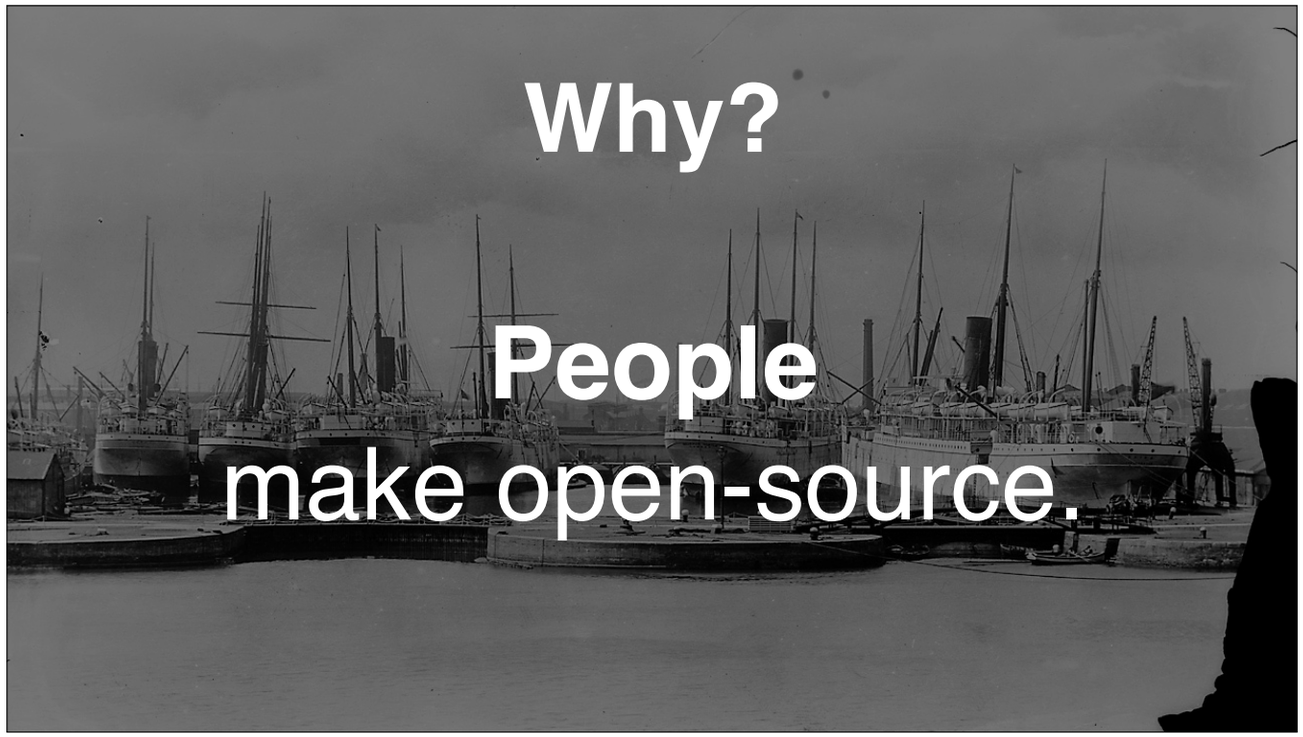
You think, "I've gone through this horrendous task, and I want to make sure no one else needs to."
Or maybe you think: "The bullshit people have to put up with is ridiculous!"

As an example, you only need to look at a handful of popular projects on GitHub, like


Underscore.js
underscorejs.org
Garen
Underscore.js (1.8.2)
» GitHub Repository
» Annotated Source
» Underscore-contrib
Introduction
Collections
– each
– map
– reduce
– reduceRight
– find
– filter
– where
– findWhere
– reject
– every
– some
– contains
– invoke
– pluck
– max
– min
– sortBy
– groupBy
– indexBy
– countBy
– shuffle
– sample
– toArray
– size
– partition
Arrays
– first
– initial
– last
– rest
– compact
– flatten
– without
– union
– intersection
– difference
– uniq
– zip
– unzip
– object
– indexOf
– lastIndexOf
– sortedIndex
– findIndex
– findLastIndex
– range
Functions
– bind
– bindAll
UNDERSCORE.JS
Underscore is a JavaScript library that provides a whole mess of useful functional programming helpers without extending any built-in objects. It's the answer to the question: "If I sit down in front of a blank HTML page, and want to start being productive immediately, what do I need?" ... and the tie to go along with jQuery's tux and Backbone's suspenders.
Underscore provides over 100 functions that support both your favorite workaday functional helpers: map, filter, invoke — as well as more specialized goodies: function binding, javascript templating, creating quick indexes, deep equality testing, and so on.
A complete Test Suite is included for your perusal.
You may also read through the annotated source code.
Enjoying Underscore, and want to turn it up to 11? Try Underscore-contrib.
The project is hosted on GitHub. You can report bugs and discuss features on the issues page, or on Freenode in the #documentcloud channel.
Underscore is an open-source component of DocumentCloud.
Downloads (Right-click, and use "Save As")
Development Version (1.8.2) 51kb, Uncompressed with Plentiful Comments
Production Version (1.8.2) 5.7kb, Minified and Gzipped (Source Map)
Edge Version Unreleased, current master, use at your own risk
Installation
Node.js npm install underscore
Meteor.js meteor add underscore
Require.js require(["underscore"], ...
Bower bower install underscore
Component component install jashkenas/underscore
Collection Functions (Arrays or Objects)


Oh My Zsh
ohmyz.sh
Garen
$ oh my zsh
DOCS
CODE
THEMES
STORE
COMMUNITY
Fork Me On Github
Your terminal never felt this good before.
Oh-My-Zsh is an open source, community-driven framework for managing your ZSH configuration. It comes bundled with a ton of helpful functions, helpers, plugins, themes, and a few things that make you shout...
“Oh My ZSH!”
180+ Plugins
700+ Contributors
140+ Themes


benbalter (Ben Balter)
GitHub, Inc. [US] https://github.com/benbalter?tab=repositories
Search GitHub
Explore Gist Blog Help
gjtorikian
Contributions Repositories Public activity
Follow
Find a repository...
Search
All Sources Forks Mirrors
Ben Balter
benbalter
Developer Program Member
GitHub STAFF
Washington, DC
ben.balter@github.com
http://ben.balter.com
Joined on May 20, 2010
2.5k Followers
221 Starred
106 Following
Organizations
newclues
forked from dweinberger/newclues
Updated an hour ago
HTML ★ 0 ⑂ 21
benbalter.github.com
The personal website of Ben Balter. Built using Jekyll and GitHub Pages. See humans.txt for more infos.
Updated 22 hours ago
HTML ★ 36 ⑂ 76
gman
A ruby gem to check if the owner of a given email address or website is working for THE MAN (a.k.a verifies government domains).
Updated 23 hours ago
Ruby ★ 58 ⑂ 36
licensee
A Ruby Gem to detect under what license a project is distributed.
Updated a day ago
Ruby ★ 21 ⑂ 7
wordpress-github-sync
A WordPress plugin to sync content with a GitHub repository (or Jekyll site)
Updated 2 days ago
PHP ★ 152 ⑂ 21
comment-card
A simple interface for non-technical users — both authenticated and pseudonymous — to provide feedback for your GitHub-hosted project
Updated 5 days ago
Ruby ★ 32 ⑂ 1



all of these projects were started because they were solving problems for other people.

Now, when we talk about people making open-source, what I really want to emphasize is

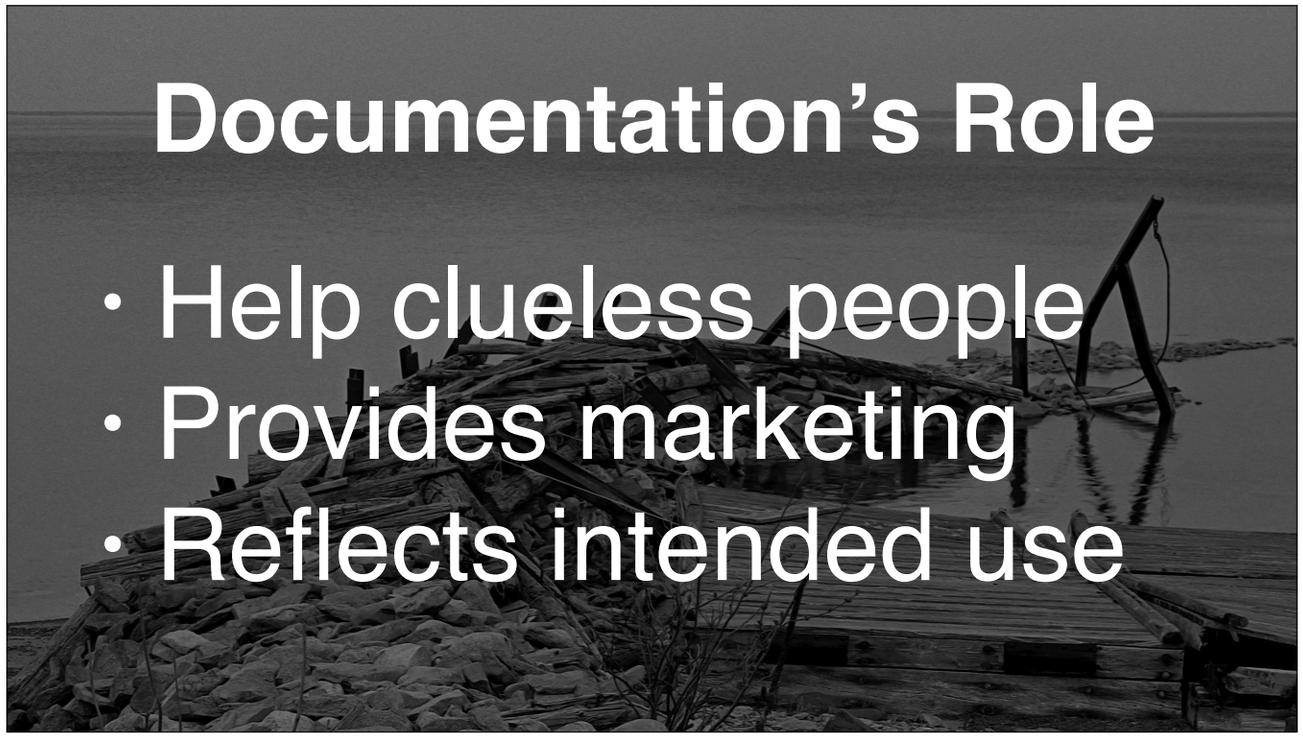
that communities of people make open source. The most successful open source projects aren’t the ones where one guy sits off in a cave with an internet connection and churns out lines of code (even if that guy is still Ben).

The most successful projects are the ones where a community is formed and develops, and contributors go back and forth between the project and their implementations of their project. Communities in open-source are both the users and writers of software.

So that gets us to the point of this talk: what’s documentation’s role here?

I’m not going to talk about why you should write README, or how great GitHub Pages are great for your projects, or any of that crap.

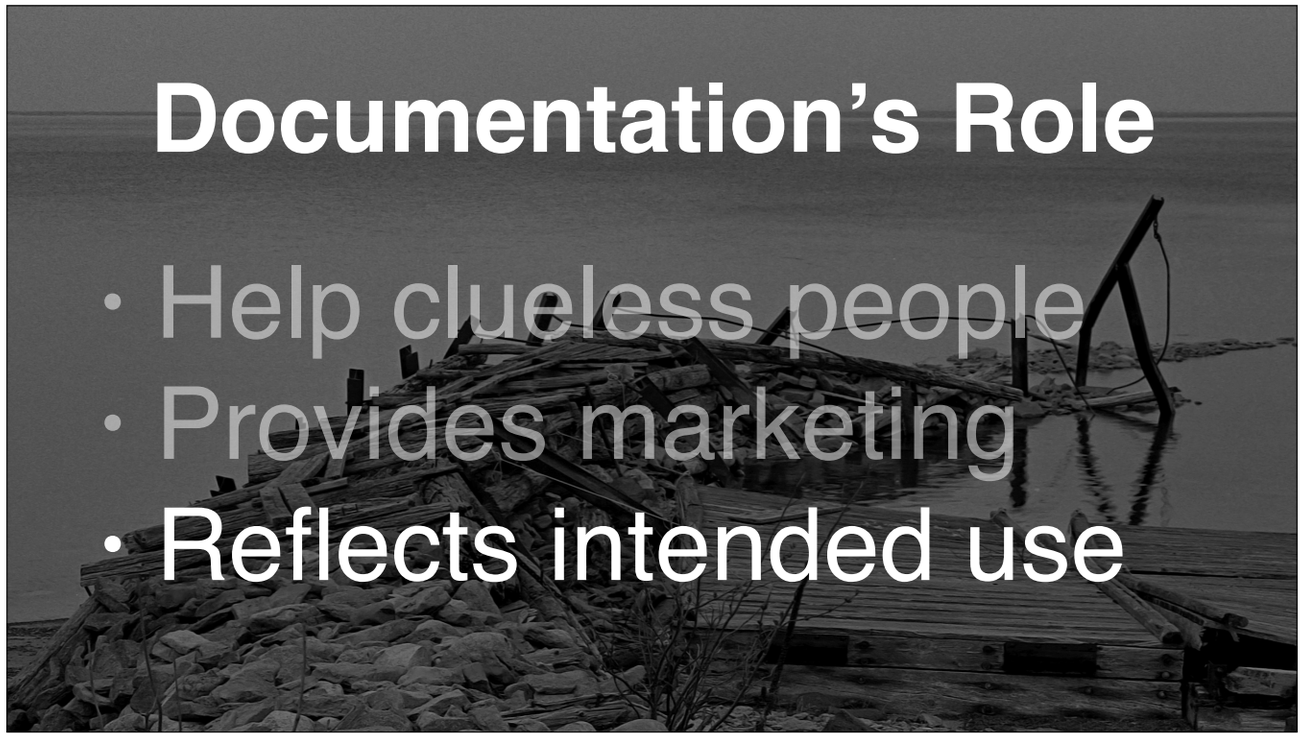
I want to get to a deeper fundamental question of what documentation does.



Documentation, of course, helps orient people in the right direction, and helps them make use of your code

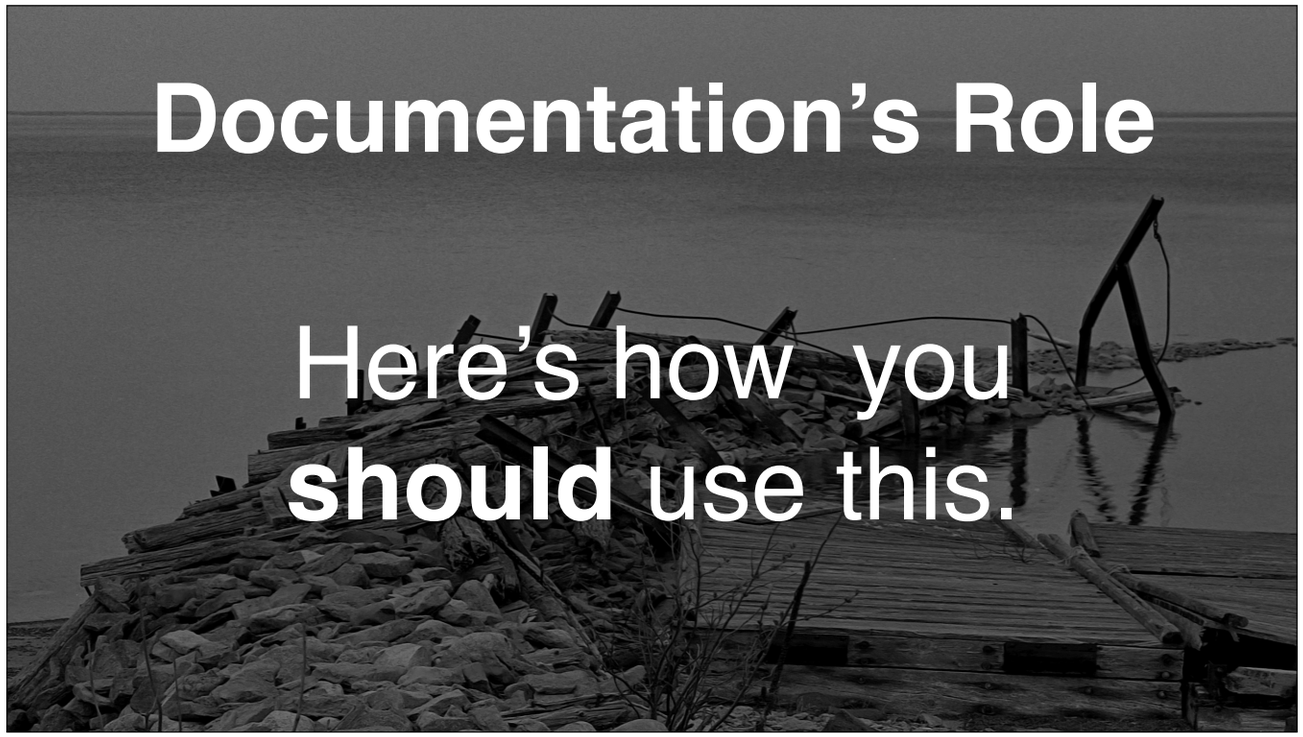
It also provides some splashy marketing. It shows off your project with a branded URL to pass around.

Most importantly, though, the documentation you provide shows how you intend your project to be used.

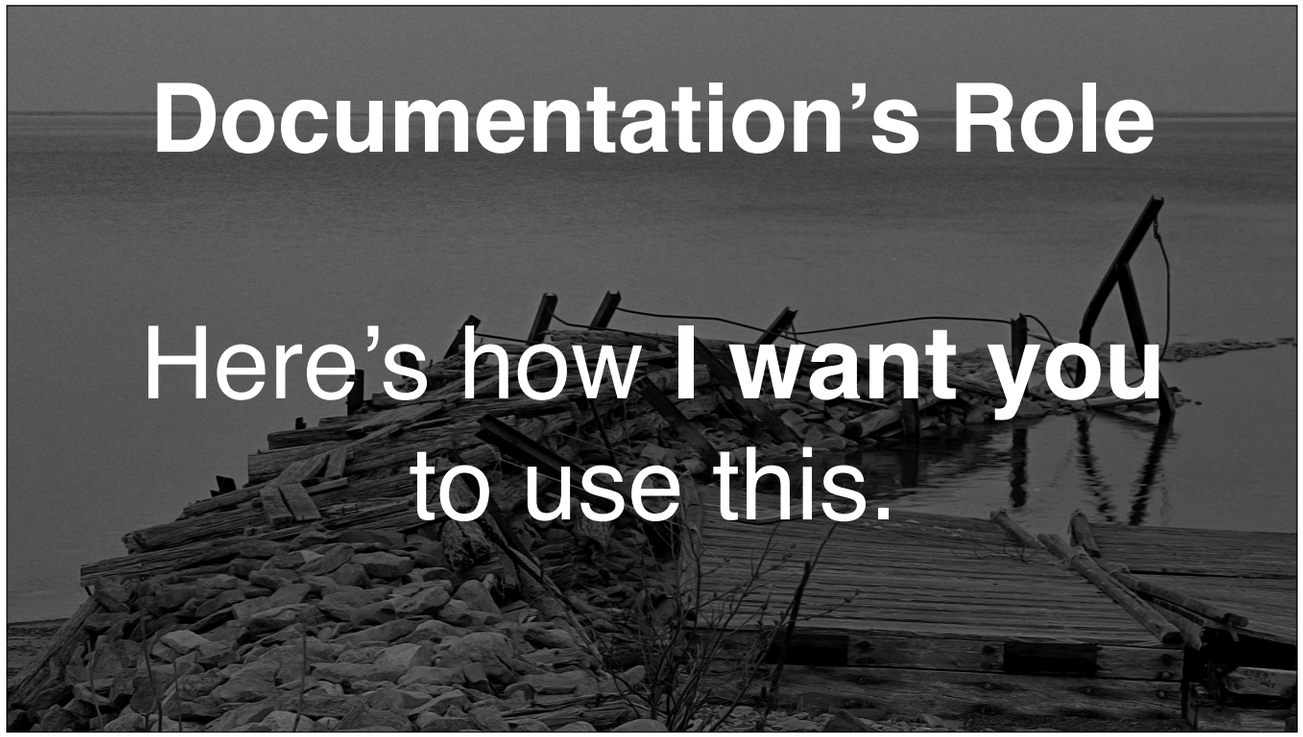


I'm short on time, but I really want to hone in on this final point of reflecting intended use.

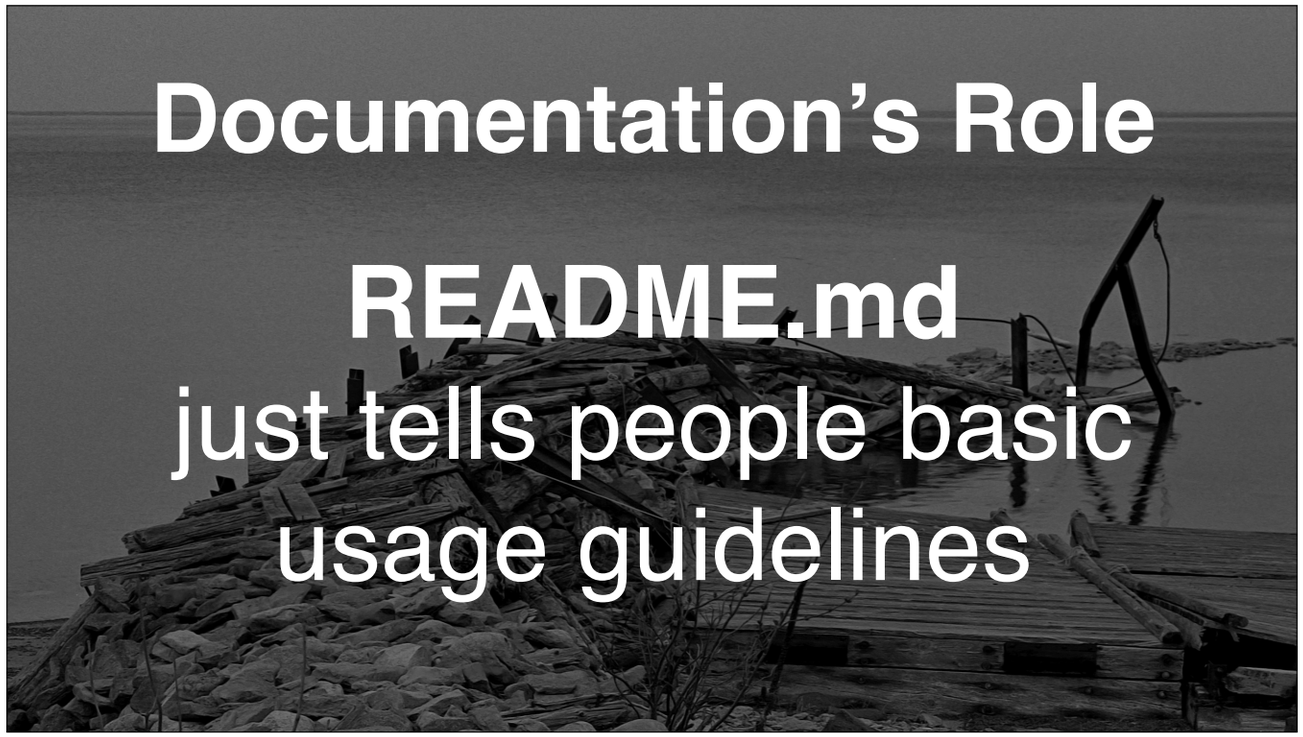
Whether you consciously intend to or not, what you write about your project represents an image to people of how you want them to use your project.



Documentation in an open source project goes beyond just “how you should use this”

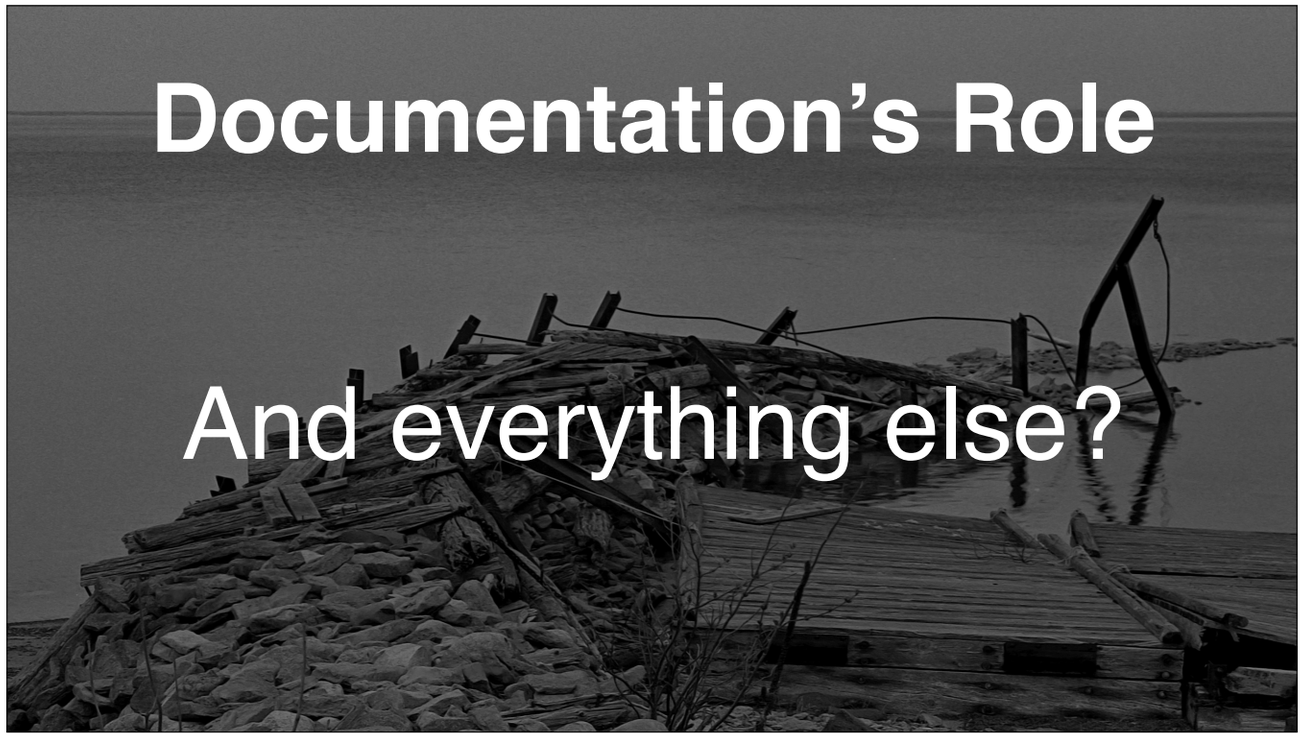


Documentation tells the consumers of your project, “This is how *I want you* to use this.”

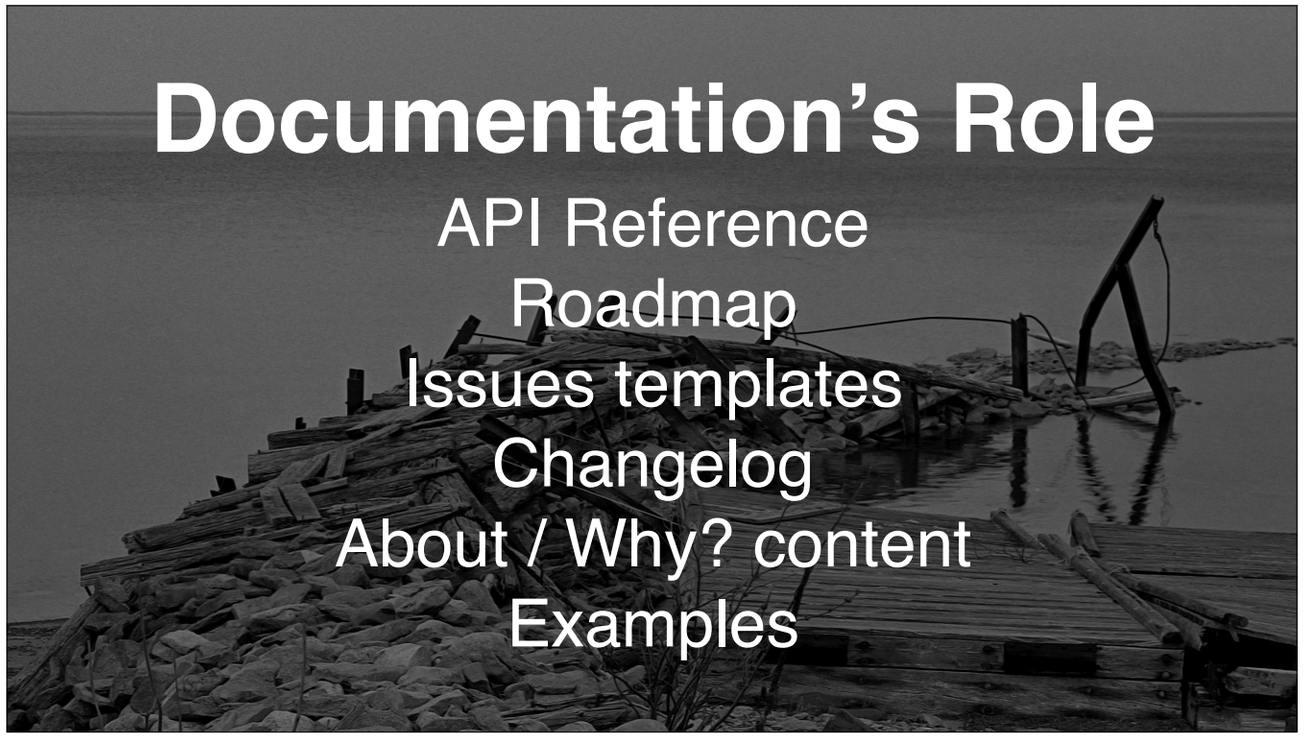


If you take a look at the most basic form of documentation on GitHub, the README, it's a simple set of quick instructions that offer some basic guidelines.

The other form of “simple documentation” we have is the CONTRIBUTING guide. With this file, you’re basically stepping beyond “here’s how I want you to use this,” and veering into “Here is how I want you to help.”



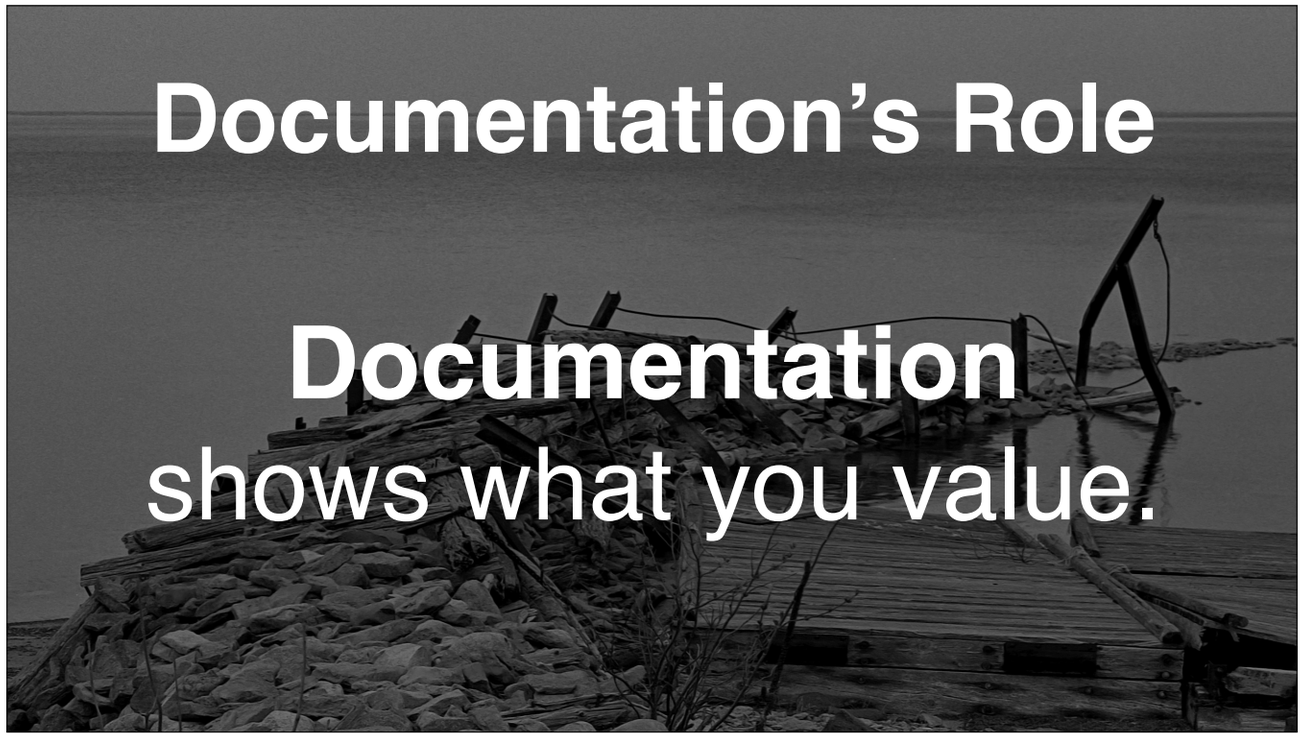
But what about everything else?



In terms of docs for a OSS project, an API references, roadmaps, templates for issues, Changelogs, website content, examples…the list goes on and on.

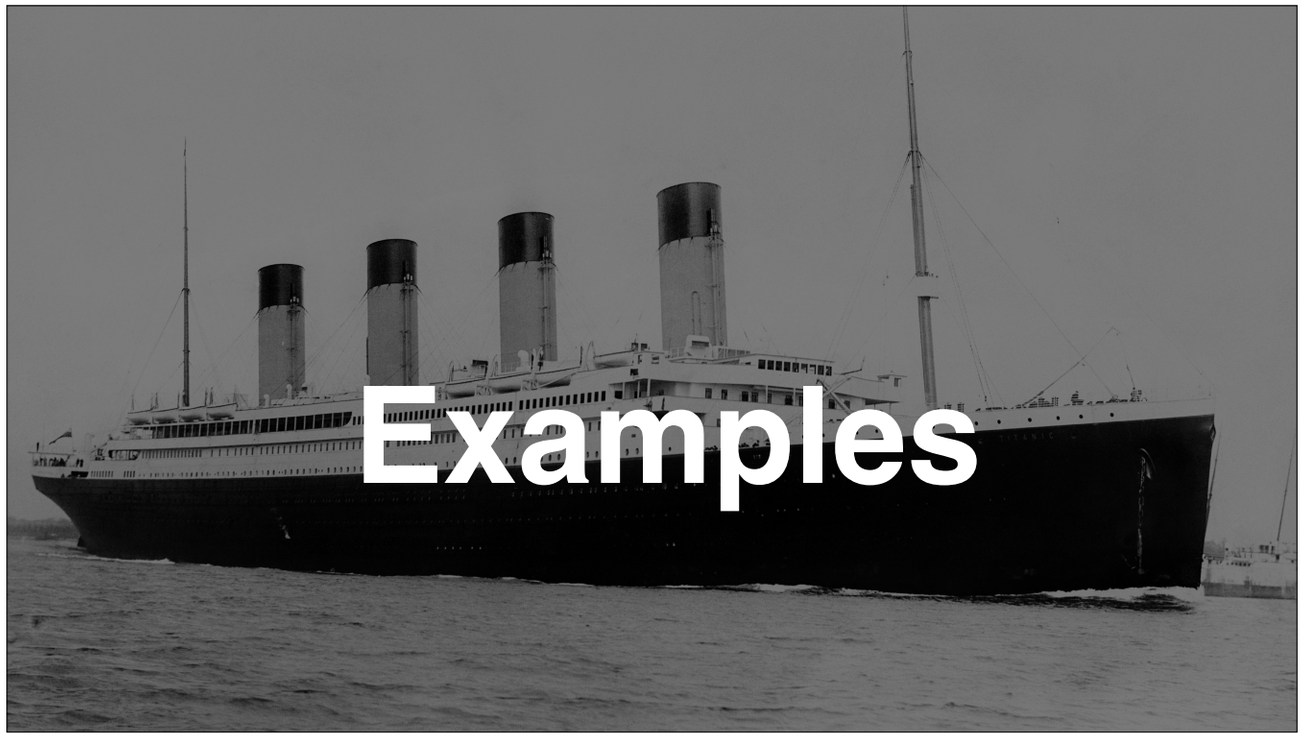
All of these are types of documentation. And the type of documentation you choose to highlight, the one you place value in, is the one you want a community to spring forth from.

Providing a roadmap shows that you have a projected set of goals your care about communicating.
A changelog shows that you care about telling people what's changed in each release.
Issues templates shows how you prefer feedback to come in (and that you like feedback).

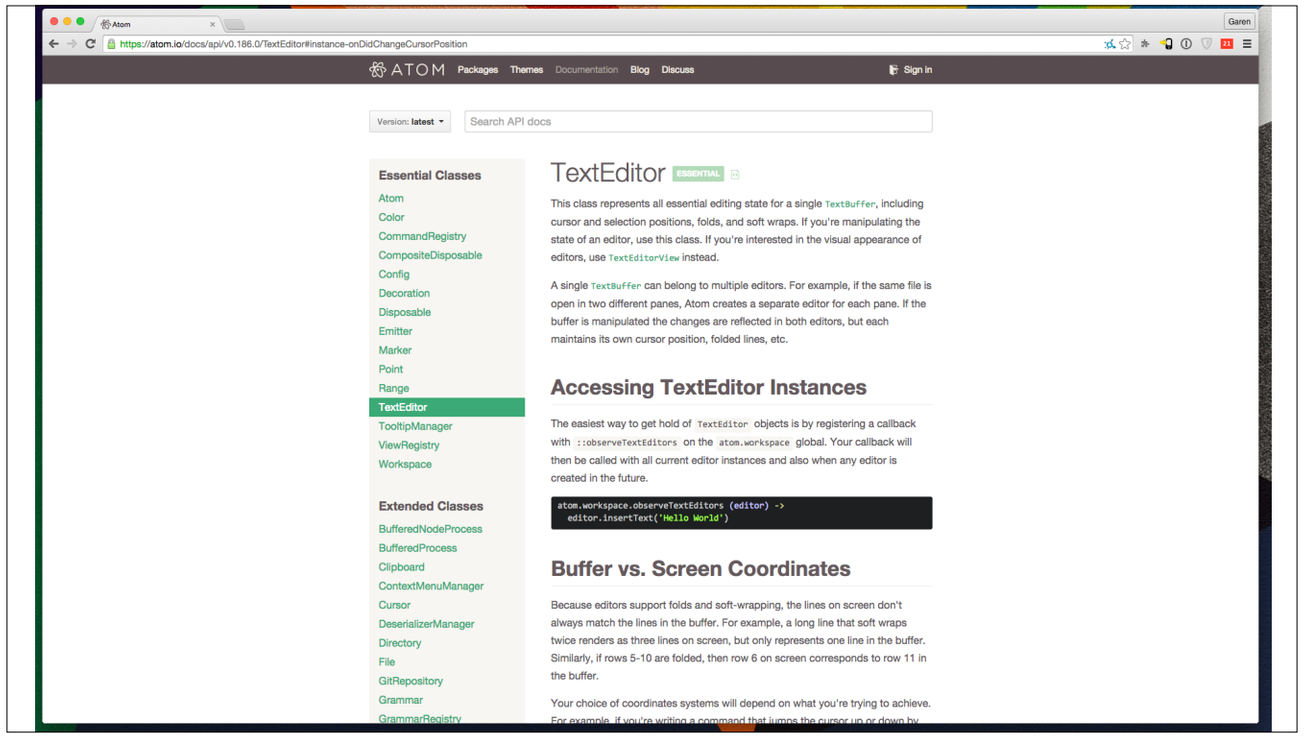


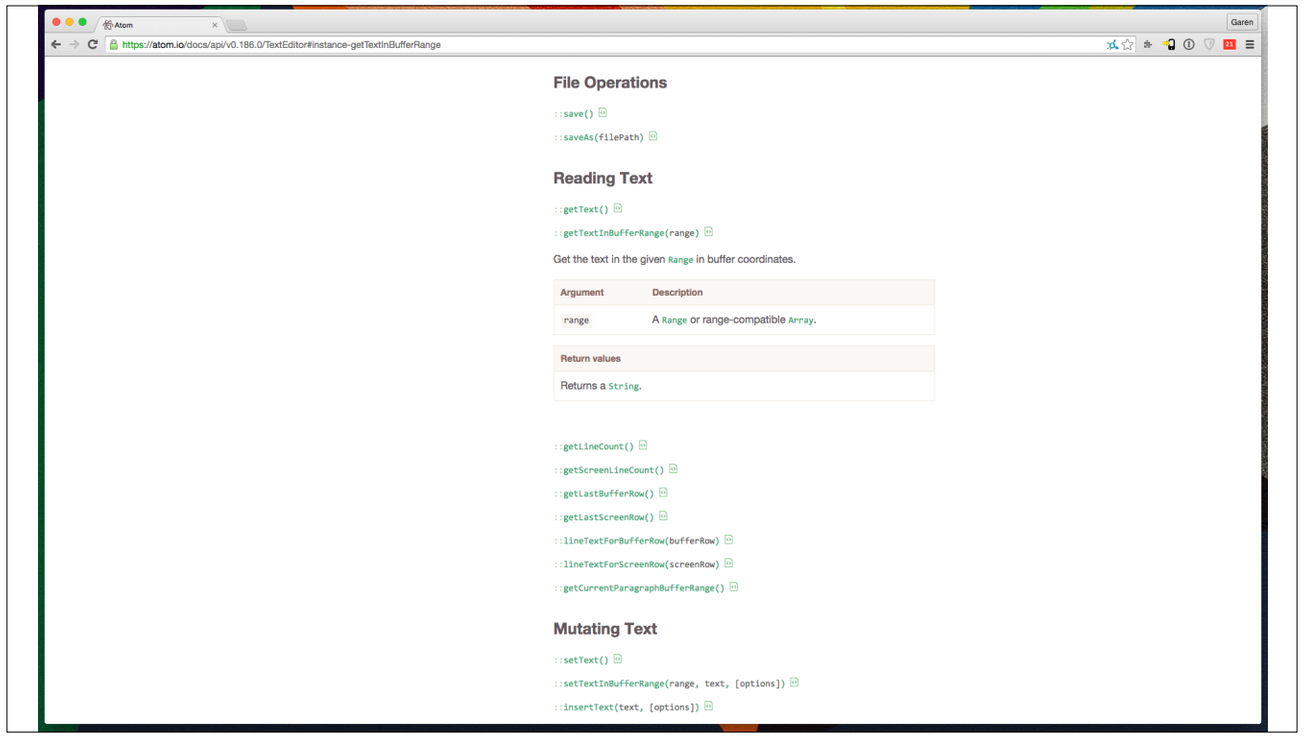
Your documentation shows people what you think is important.

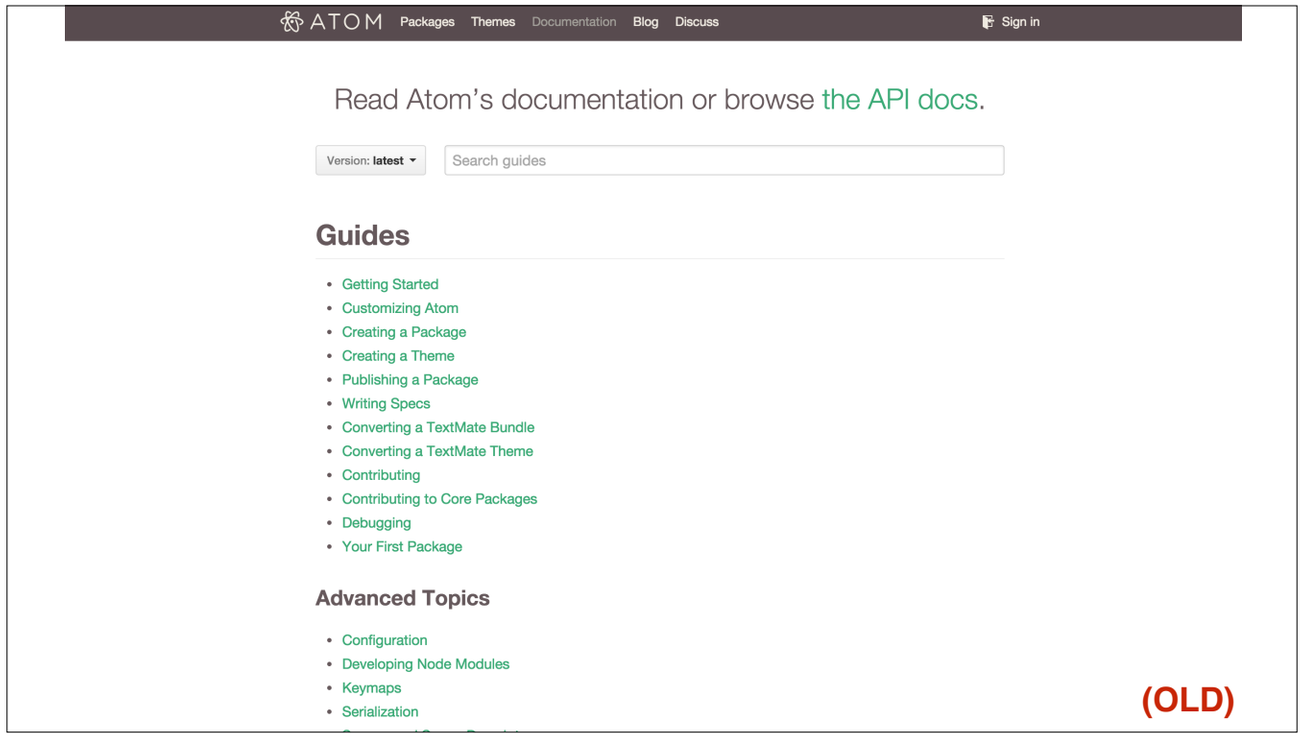
And what you value is what your community will grow from.



So let’s show off with some examples.


Atom
Garen
https://atom.io/docs/api/v0.186.0/TextEditor#instance-onDidChangeCursorPosition
ATOM
Packages
Themes
Documentation
Blog
Discuss
Sign in
Version: latest
Search API docs
Essential Classes
Atom
Color
CommandRegistry
CompositeDisposable
Config
Decoration
Disposable
Emitter
Marker
Point
Range
TextEditor
TooltipManager
ViewRegistry
Workspace
Extended Classes
BufferedNodeProcess
BufferedProcess
Clipboard
ContextMenuManager
Cursor
DeserializerManager
Directory
File
GitRepository
Grammar
GrammarRegistry
TextEditor
ESSENTIAL
This class represents all essential editing state for a single TextBuffer, including cursor and selection positions, folds, and soft wraps. If you're manipulating the state of an editor, use this class. If you're interested in the visual appearance of editors, use TextEditorView instead.
A single TextBuffer can belong to multiple editors. For example, if the same file is open in two different panes, Atom creates a separate editor for each pane. If the buffer is manipulated the changes are reflected in both editors, but each maintains its own cursor position, folded lines, etc.
Accessing TextEditor Instances
The easiest way to get hold of TextEditor objects is by registering a callback with ::observeTextEditors on the atom.workspace global. Your callback will then be called with all current editor instances and also when any editor is created in the future.
atom.workspace.observeTextEditors (editor) ->
editor.insertText('Hello World')
Buffer vs. Screen Coordinates
Because editors support folds and soft-wrapping, the lines on screen don't always match the lines in the buffer. For example, a long line that soft wraps twice renders as three lines on screen, but only represents one line in the buffer. Similarly, if rows 5-10 are folded, then row 6 on screen corresponds to row 11 in the buffer.
Your choice of coordinates systems will depend on what you're trying to achieve. For example, if you're writing a command that jumps the cursor up or down by


Atom
https://atom.io/docs/api/v0.186.0/TextEditor#instance-getTextInBufferRange
Garen
File Operations
::save()
::saveAs(filePath)
Reading Text
::getText()
::getTextInBufferRange(range)
Get the text in the given Range in buffer coordinates.
Argument
Description
range
A Range or range-compatible Array.
Return values
Returns a String.
::getLineCount()
::getScreenLineCount()
::getLastBufferRow()
::getLastScreenRow()
::lineTextForBufferRow(bufferRow)
::lineTextForScreenRow(screenRow)
::getCurrentParagraphBufferRange()
Mutating Text
::setText()
::setTextInBufferRange(range, text, [options])
::insertText(text, [options])

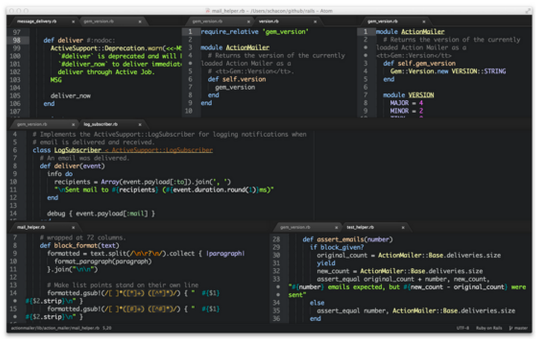

ATOM
Packages
Themes
Documentation
Blog
Discuss
Sign in
Read Atom's documentation or browse the API docs.
Version: latest
Search guides
Guides
Getting Started
Customizing Atom
Creating a Package
Creating a Theme
Publishing a Package
Writing Specs
Converting a TextMate Bundle
Converting a TextMate Theme
Contributing
Contributing to Core Packages
Debugging
Your First Package
Advanced Topics
Configuration
Developing Node Modules
Keymaps
Serialization
(OLD)

Atom Flight Manual

1 : Getting Started
1.1 Why Atom
1.2 Installing Atom
1.3 Atom Basics
1.4 Summary
2 : Using Atom
2.1 Atom Packages
2.2 Moving in Atom
2.3 Atom Selections
2.4 Editing and Deleting Text
2.5 Find and Replace
2.6 Snippets
2.7 Autocomplete
2.8 Folding
2.9 Panes
2.10 Grammar
2.11 Version Control in Atom
2.12 Writing in Atom
2.13 Basic Customization
2.14 Summary
3 : Hacking Atom
3.1 Tools of the Trade
3.2 Package: Word Count
3.3 Package: Modifying Text
3.4 Creating a Theme
3.5 Debugging
3.6 Writing specs
3.7 Converting from TextMate
3.8 Summary
4 : Behind Atom
4.1 Configuration API
4.2 Keymaps In-Depth

Panes

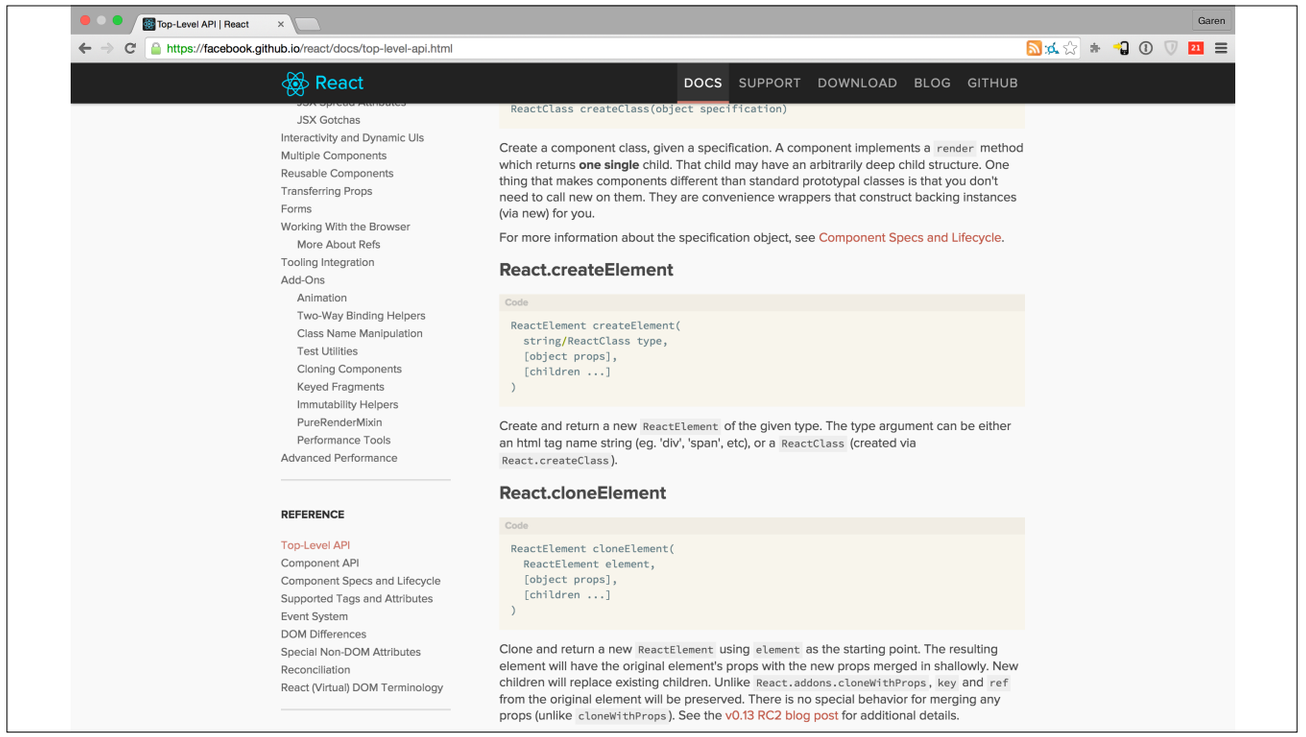
You can split any editor pane horizontally or vertically by using `cmd-k arrow` where the arrow is the direction to split the pane . Once you have a split pane, you can move focus between them with `cmd-k cmd-arrow` where the arrow is the direction the focus should move to.

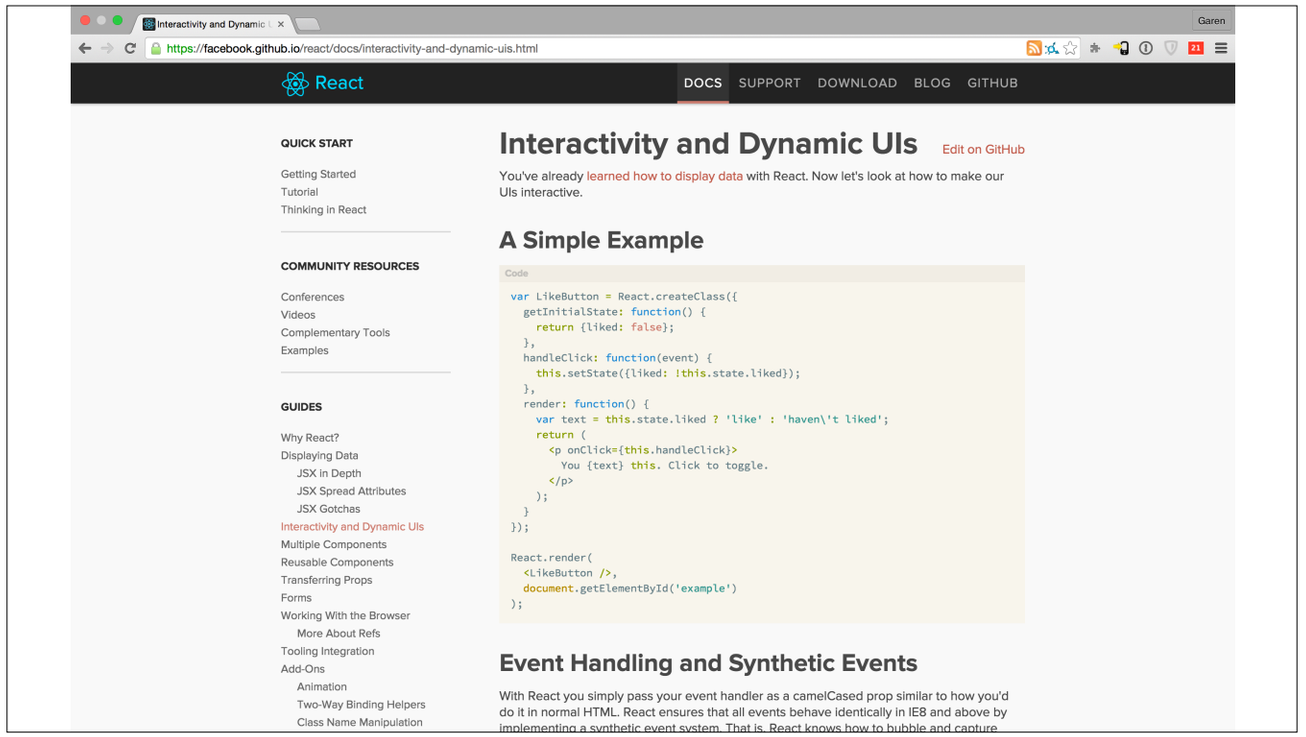


Multiple panes

Each pane has it's own "items" or files, which are represented by tabs. You can move the files from pane to pane by dragging them with the mouse and dropping them in the pane you want that file to be in.

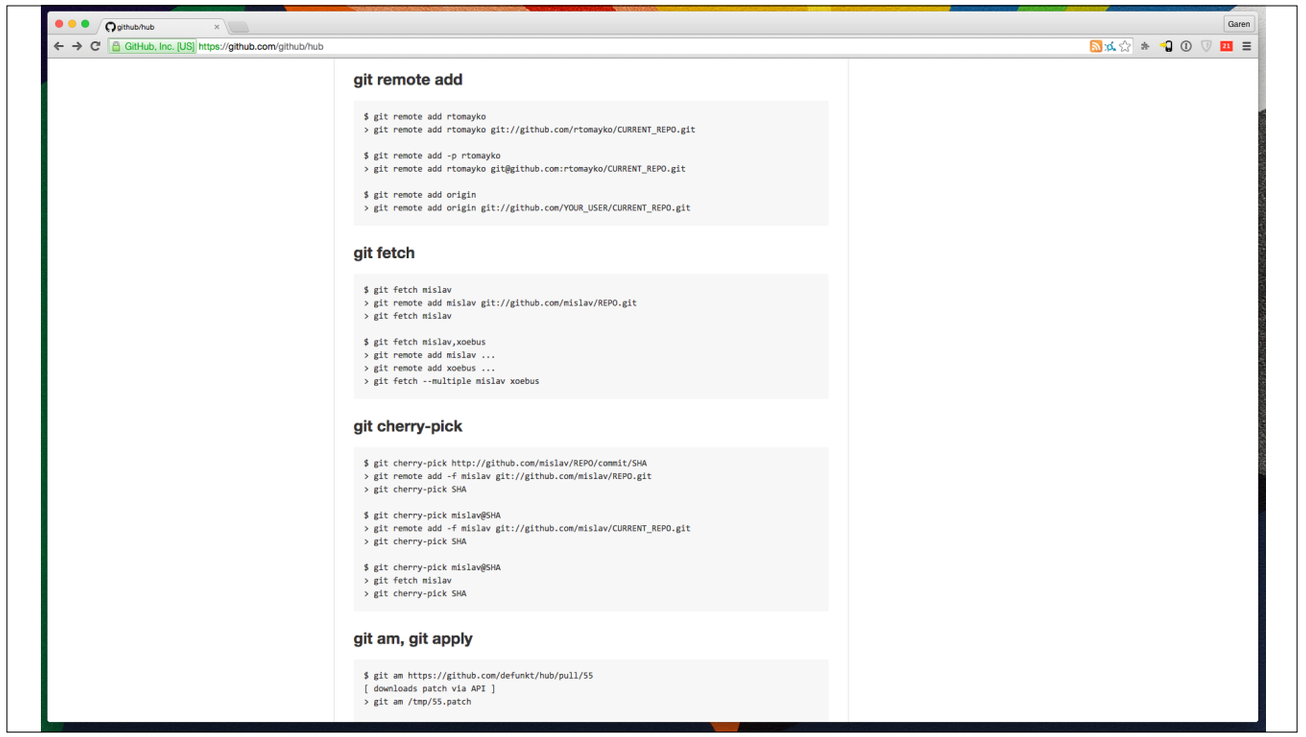
To close a pane, close all its editors with `cmd-w` , then press `cmd-w` one more time to close the pane. You can configure panes to auto-close when empty in the Settings view. **(NEW)**

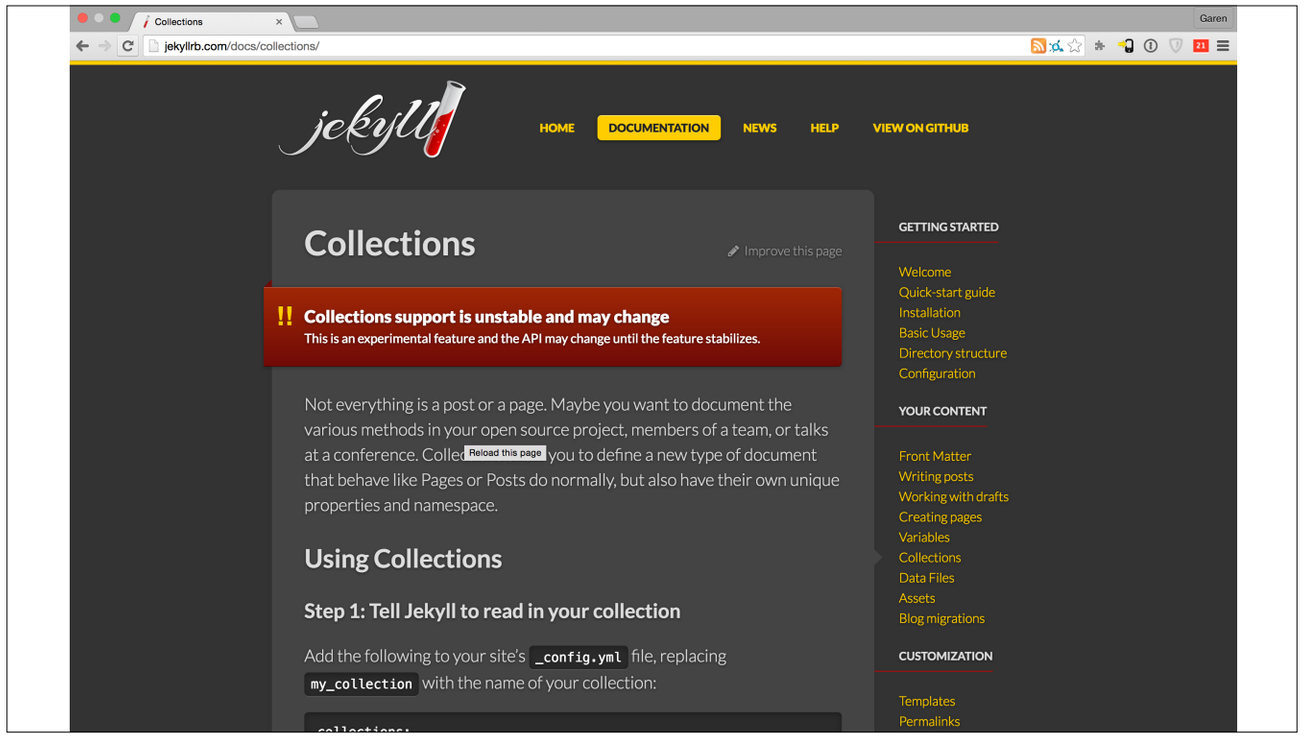

Top-Level API | React
Garen
https://facebook.github.io/react/docs/top-level-api.html
React
DOCS
SUPPORT
DOWNLOAD
BLOG
GITHUB
JSX Gotchas
Interactivity and Dynamic UIs
Multiple Components
Reusable Components
Transferring Props
Forms
Working With the Browser
More About Refs
Tooling Integration
Add-Ons
Animation
Two-Way Binding Helpers
Class Name Manipulation
Test Utilities
Cloning Components
Keyed Fragments
Immutability Helpers
PureRenderMixin
Performance Tools
Advanced Performance
REFERENCE
Top-Level API
Component API
Component Specs and Lifecycle
Supported Tags and Attributes
Event System
DOM Differences
Special Non-DOM Attributes
Reconciliation
React (Virtual) DOM Terminology
ReactClass createClass(object specification)
Create a component class, given a specification. A component implements a render method which returns one single child. That child may have an arbitrarily deep child structure. One thing that makes components different than standard prototypal classes is that you don't need to call new on them. They are convenience wrappers that construct backing instances (via new) for you.
For more information about the specification object, see Component Specs and Lifecycle.
React.createElement
Code
ReactElement createElement(
string/ReactClass type,
[object props],
[children ...]
)
Create and return a new ReactElement of the given type. The type argument can be either an html tag name string (eg. 'div', 'span', etc), or a ReactClass (created via React.createClass).
React.cloneElement
Code
ReactElement cloneElement(
ReactElement element,
[object props],
[children ...]
)
Clone and return a new ReactElement using element as the starting point. The resulting element will have the original element's props with the new props merged in shallowly. New children will replace existing children. Unlike React.addons.cloneWithProps, key and ref from the original element will be preserved. There is no special behavior for merging any props (unlike cloneWithProps). See the v0.13 RC2 blog post for additional details.


Interactivity and Dynamic
Garen
https://facebook.github.io/react/docs/interactivity-and-dynamic-uis.html
React
DOCS
SUPPORT
DOWNLOAD
BLOG
GITHUB
QUICK START
Getting Started
Tutorial
Thinking in React
COMMUNITY RESOURCES
Conferences
Videos
Complementary Tools
Examples
GUIDES
Why React?
Displaying Data
JSX in Depth
JSX Spread Attributes
JSX Gotchas
Interactivity and Dynamic UIs
Multiple Components
Reusable Components
Transferring Props
Forms
Working With the Browser
More About Refs
Tooling Integration
Add-Ons
Animation
Two-Way Binding Helpers
Class Name Manipulation
Interactivity and Dynamic UIs
Edit on GitHub
You've already learned how to display data with React. Now let's look at how to make our UIs interactive.
A Simple Example
Code
var LikeButton = React.createClass({
getInitialState: function() {
return {liked: false};
},
handleClick: function(event) {
this.setState({liked: !this.state.liked});
},
render: function() {
var text = this.state.liked ? 'like' : 'haven\'t liked';
return (

You {text} this. Click to toggle.

);
}
});
React.render(
<LikeButton />,
document.getElementById('example')
);
Event Handling and Synthetic Events
With React you simply pass your event handler as a camelCased prop similar to how you'd do it in normal HTML. React ensures that all events behave identically in IE8 and above by implementing a synthetic event system. That is, React knows how to bubble and capture


github/hub
GitHub, Inc. [US] https://github.com/github/hub
Glaren
git remote add
$ git remote add rtomayko
> git remote add rtomayko git://github.com/rtomayko/CURRENT_REPO.git
$ git remote add -p rtomayko
> git remote add rtomayko git@github.com:rtomayko/CURRENT_REPO.git
$ git remote add origin
> git remote add origin git://github.com/YOUR_USER/CURRENT_REPO.git
git fetch
$ git fetch mislav
> git remote add mislav git://github.com/mislav/REPO.git
> git fetch mislav
$ git fetch mislav,xoebus
> git remote add mislav ...
> git remote add xoebus ...
> git fetch --multiple mislav xoebus
git cherry-pick
$ git cherry-pick http://github.com/mislav/REPO/commit/SHA
> git remote add -f mislav git://github.com/mislav/REPO.git
> git cherry-pick SHA
$ git cherry-pick mislav@SHA
> git remote add -f mislav git://github.com/mislav/CURRENT_REPO.git
> git cherry-pick SHA
$ git cherry-pick mislav@SHA
> git fetch mislav
> git cherry-pick SHA
git am, git apply
$ git am https://github.com/defunkt/hub/pull/55
[downloads patch via API]
> git am /tmp/55.patch


Collections
Garen
jekyllrb.com/docs/collections/
jekyll
HOME
DOCUMENTATION
NEWS
HELP
VIEW ON GITHUB
Collections
Improve this page
!! Collections support is unstable and may change
This is an experimental feature and the API may change until the feature stabilizes.
Not everything is a post or a page. Maybe you want to document the various methods in your open source project, members of a team, or talks at a conference. Colle
Reload this page
you to define a new type of document that behave like Pages or Posts do normally, but also have their own unique properties and namespace.
Using Collections
Step 1: Tell Jekyll to read in your collection
Add the following to your site's _config.yml file, replacing my_collection with the name of your collection:
collections:
GETTING STARTED
Welcome
Quick-start guide
Installation
Basic Usage
Directory structure
Configuration
YOUR CONTENT
Front Matter
Writing posts
Working with drafts
Creating pages
Variables
Collections
Data Files
Assets
Blog migrations
CUSTOMIZATION
Templates
Permalinks

branch: master ▾ **jekyll** / lib / jekyll / **document.rb**

willnorris 8 days ago always include file extension on destination files

6 contributors 

285 lines (254 sloc) 8.723 kb **Raw** **Blame** **History**

```
# encoding: UTF-8

module Jekyll
  class Document
    include Comparable

    attr_reader   :path, :site, :extname, :output_ext
    attr_accessor :content, :collection, :output

    YAML_FRONT_MATTER_REGEXP = /\A(---\s*\n.*?\n?)^((---|\.\.\.)\s*$\n?)/m

    # Create a new Document.
    #
    # site - the Jekyll::Site instance to which this Document belongs
    # path - the path to the file
    #
    # Returns nothing.
    def initialize(path, relations)
      @site = relations[:site]
      @path = path
      @extname = File.extname(path)
      @output_ext = Jekyll::Renderer.new(site, self).output_ext
      @collection = relations[:collection]
      @has_yaml_header = nil
    end

    # Fetch the Document's data.
    #
    # Returns a Hash containing the data. An empty hash is returned if
    #   no data was read.
    def data
      @data ||= Hash.new
    end

    # The path to the document, relative to the site source.
```

Active Record Basics

This guide is an introduction to Active Record.

After reading this guide, you will know:

- **What Object Relational Mapping and Active Record are and how they are used in Rails.**
- **How Active Record fits into the Model-View-Controller paradigm.**
- **How to use Active Record models to manipulate data stored in a relational database.**
- **Active Record schema naming conventions.**
- **The concepts of database migrations, validations and callbacks.**

Chapters

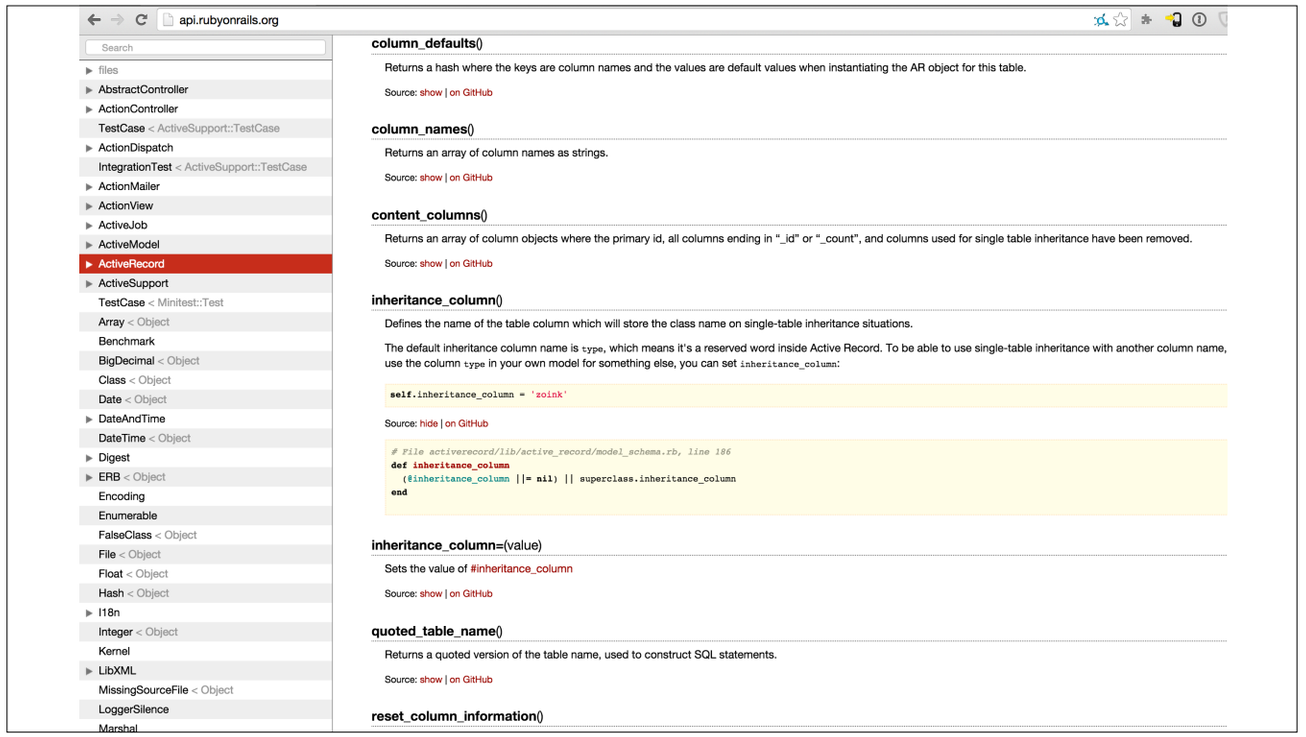
1. **What is Active Record?**
   - The Active Record Pattern
   - Object Relational Mapping
   - Active Record as an ORM Framework
2. **Convention over Configuration in Active Record**
   - Naming Conventions
   - Schema Conventions
3. **Creating Active Record Models**
4. **Overriding the Naming Conventions**
5. **CRUD: Reading and Writing Data**
   - Create
   - Read
   - Update
   - Delete
6. **Validations**
7. **Callbacks**
8. **Migrations**

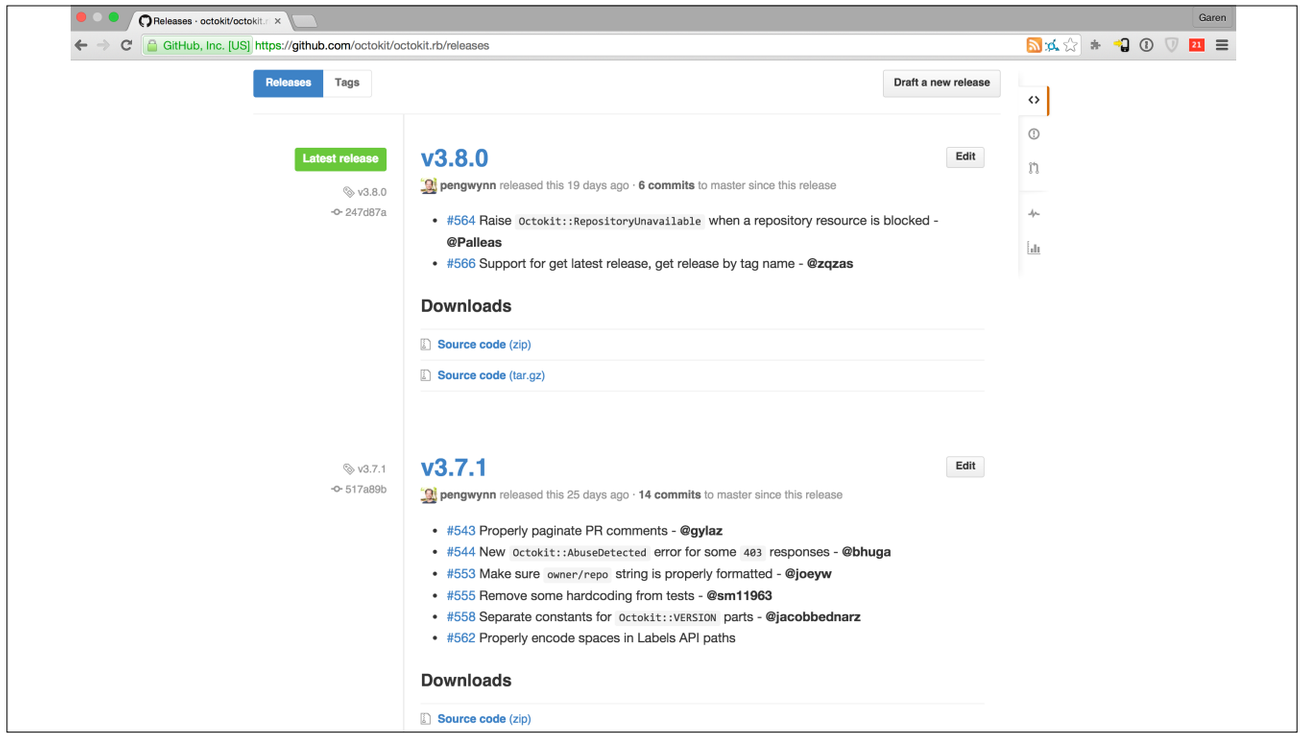
1 What is Active Record?

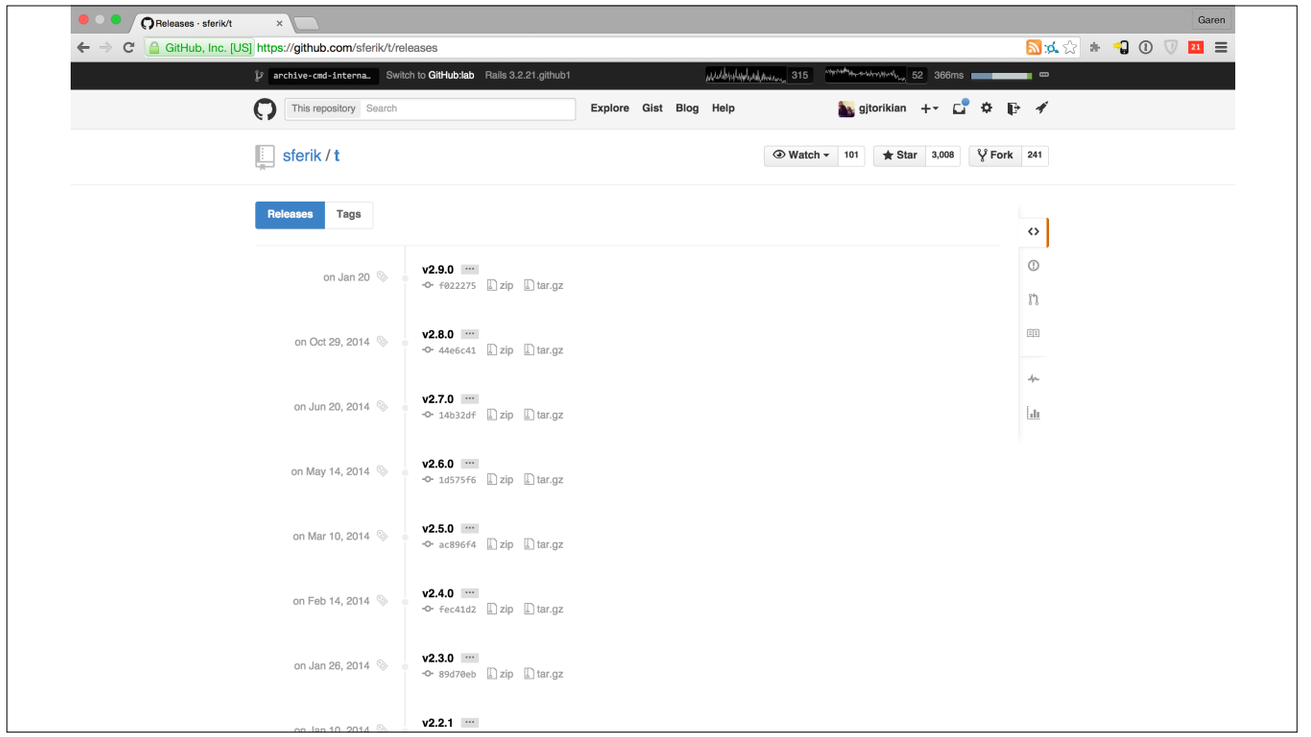
Active Record is the M in MVC - the model - which is the layer of the system responsible for representing business data and logic. Active Record facilitates the creation and use of business objects whose data requires persistent storage to a database. It is an implementation of the Active Record pattern which itself is a description of an Object Relational Mapping system.

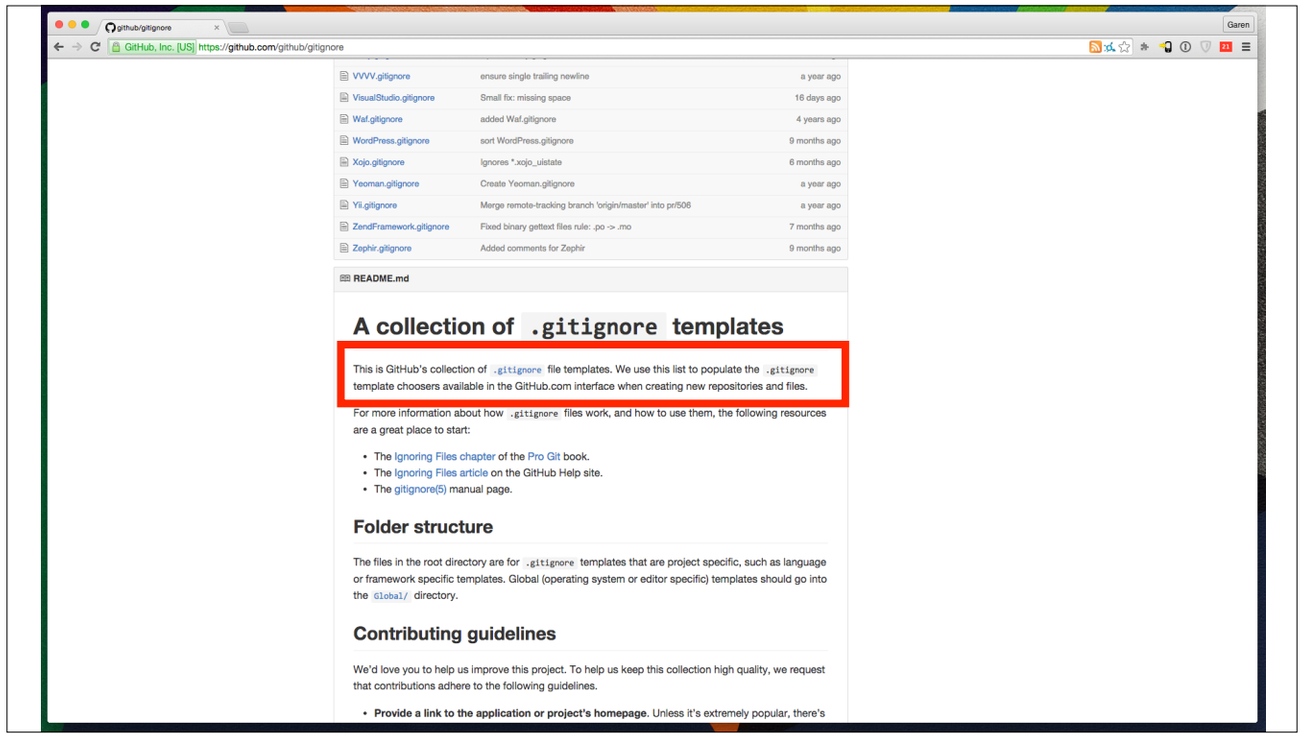
1.1 The Active Record Pattern

Active Record was described by Martin Fowler in his book *Patterns of Enterprise Application Architecture*. In Active Record, objects carry both persistent data and behavior which operates on that data. Active Record takes the opinion that ensuring data access logic as part of the object will educate users of that object on how to write to and read from the database.


api.rubyonrails.org
Search
files
AbstractController
ActionController
TestCase < ActiveSupport::TestCase
ActionDispatch
IntegrationTest < ActiveSupport::TestCase
ActionMailer
ActionView
ActiveJob
ActiveModel
ActiveRecord
ActiveSupport
TestCase < Minitest::Test
Array < Object
Benchmark
BigDecimal < Object
Class < Object
Date < Object
DateAndTime
DateTime < Object
Digest
ERB < Object
Encoding
Enumerable
FalseClass < Object
File < Object
Float < Object
Hash < Object
I18n
Integer < Object
Kernel
LibXML
MissingSourceFile < Object
LoggerSilence
Marshal
column_defaults()
Returns a hash where the keys are column names and the values are default values when instantiating the AR object for this table.
Source: show | on GitHub
column_names()
Returns an array of column names as strings.
Source: show | on GitHub
content_columns()
Returns an array of column objects where the primary id, all columns ending in "_id" or "_count", and columns used for single table inheritance have been removed.
Source: show | on GitHub
inheritance_column()
Defines the name of the table column which will store the class name on single-table inheritance situations.
The default inheritance column name is type, which means it's a reserved word inside Active Record. To be able to use single-table inheritance with another column name, use the column type in your own model for something else, you can set inheritance_column:
self.inheritance_column = 'zoink'
Source: hide | on GitHub
File activerecord/lib/active_record/model_schema.rb, line 186
def inheritance_column
(@inheritance_column ||= nil) || superclass.inheritance_column
end
inheritance_column=(value)
Sets the value of #inheritance_column
Source: show | on GitHub
quoted_table_name()
Returns a quoted version of the table name, used to construct SQL statements.
Source: show | on GitHub
reset_column_information()


Releases · octokit/octokit.r
Garen
GitHub, Inc. [US] https://github.com/octokit/octokit.rb/releases
Releases
Tags
Draft a new release
Latest release
v3.8.0
247d87a
v3.8.0
Edit
pengwynn released this 19 days ago · 6 commits to master since this release
#564 Raise Octokit::RepositoryUnavailable when a repository resource is blocked - @Palleas
#566 Support for get latest release, get release by tag name - @zqzas
Downloads
Source code (zip)
Source code (tar.gz)
v3.7.1
517a89b
v3.7.1
Edit
pengwynn released this 25 days ago · 14 commits to master since this release
#543 Properly paginate PR comments - @gylaz
#544 New Octokit::AbuseDetected error for some 403 responses - @bhuga
#553 Make sure owner/repo string is properly formatted - @joeyw
#555 Remove some hardcoding from tests - @sm11963
#558 Separate constants for Octokit::VERSION parts - @jacobbednarz
#562 Properly encode spaces in Labels API paths
Downloads
Source code (zip)


Releases · sferik/t
Garen
GitHub, Inc. [US] https://github.com/sferik/t/releases
archive-cmd-interna…
Switch to GitHub:lab
Rails 3.2.21.github1
315
52
366ms
This repository
Search
Explore
Gist
Blog
Help
gjtorikian
sferik / t
Watch 101
Star 3,008
Fork 241
Releases
Tags
on Jan 20
v2.9.0
f022275 zip tar.gz
on Oct 29, 2014
v2.8.0
44e6c41 zip tar.gz
on Jun 20, 2014
v2.7.0
14b32df zip tar.gz
on May 14, 2014
v2.6.0
1d575f6 zip tar.gz
on Mar 10, 2014
v2.5.0
ac896f4 zip tar.gz
on Feb 14, 2014
v2.4.0
fec41d2 zip tar.gz
on Jan 26, 2014
v2.3.0
89d70eb zip tar.gz
v2.2.1


github/gitignore
GitHub, Inc. [US] https://github.com/github/gitignore
Garen
VVVV.gitignore ensure single trailing newline a year ago
VisualStudio.gitignore Small fix: missing space 16 days ago
Waf.gitignore added Waf.gitignore 4 years ago
WordPress.gitignore sort WordPress.gitignore 9 months ago
Xojo.gitignore Ignores *.xojo_uistate 6 months ago
Yeoman.gitignore Create Yeoman.gitignore a year ago
Yii.gitignore Merge remote-tracking branch 'origin/master' into pr/506 a year ago
ZendFramework.gitignore Fixed binary gettext files rule: .po -> .mo 7 months ago
Zephir.gitignore Added comments for Zephir 9 months ago
README.md
A collection of .gitignore templates
This is GitHub's collection of .gitignore file templates. We use this list to populate the .gitignore template choosers available in the GitHub.com interface when creating new repositories and files.
For more information about how .gitignore files work, and how to use them, the following resources are a great place to start:
The Ignoring Files chapter of the Pro Git book.
The Ignoring Files article on the GitHub Help site.
The gitignore(5) manual page.
Folder structure
The files in the root directory are for .gitignore templates that are project specific, such as language or framework specific templates. Global (operating system or editor specific) templates should go into the Global/ directory.
Contributing guidelines
We'd love you to help us improve this project. To help us keep this collection high quality, we request that contributions adhere to the following guidelines.
Provide a link to the application or project's homepage. Unless it's extremely popular, there's

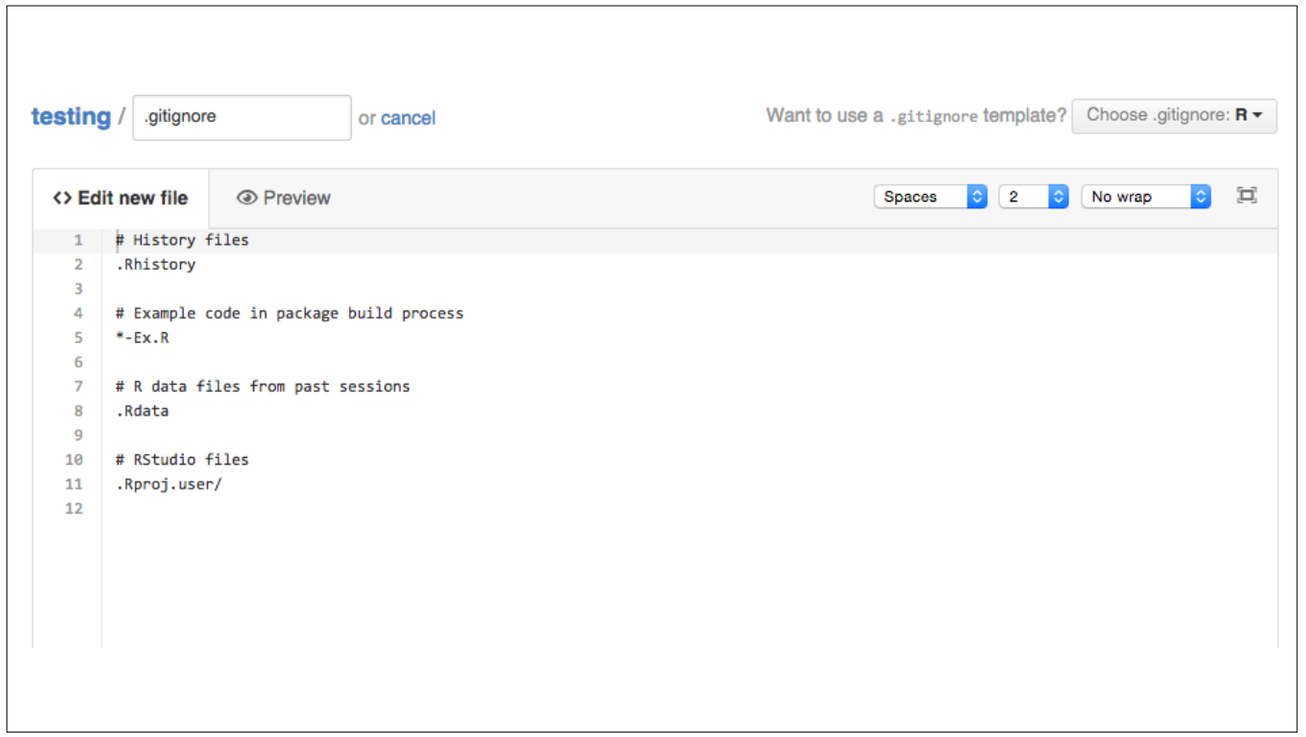
branch: **master** ▾ **gitignore** / **R.gitignore**

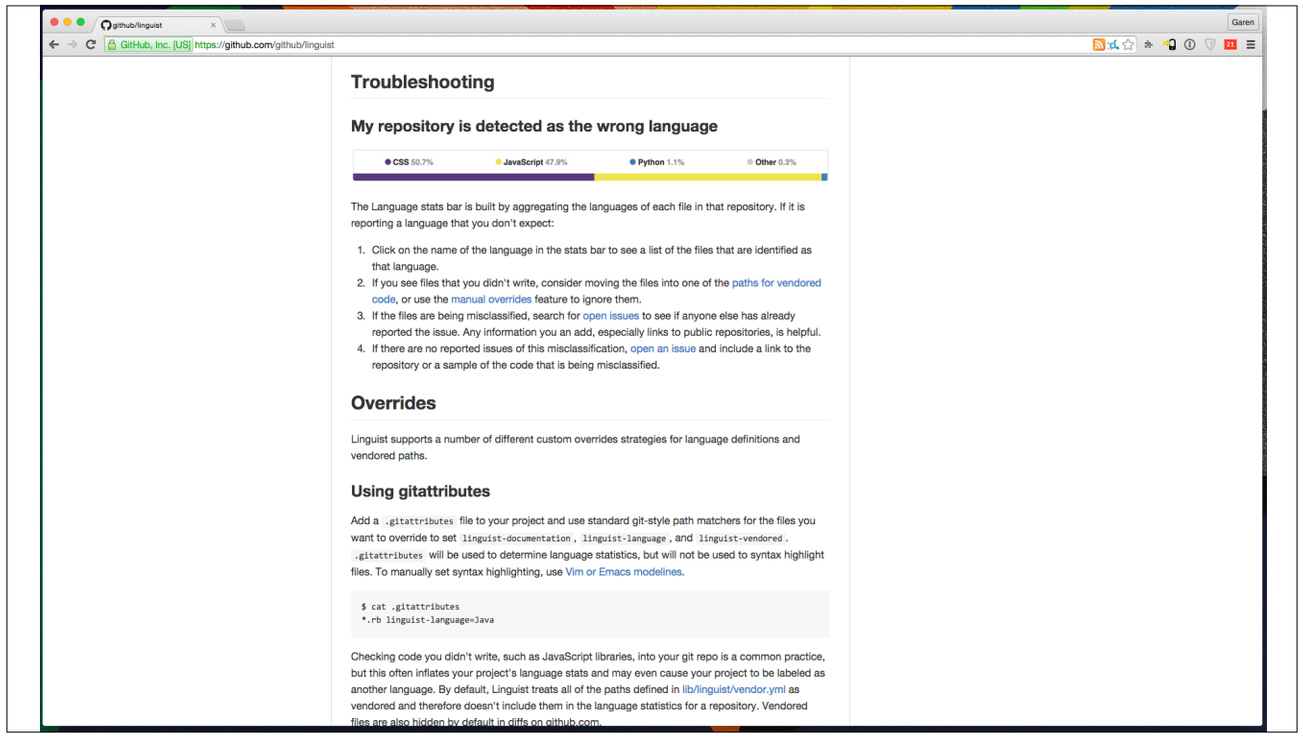
arcresu on Nov 28, 2014 Merge pull request #1263 from jangorecki/master

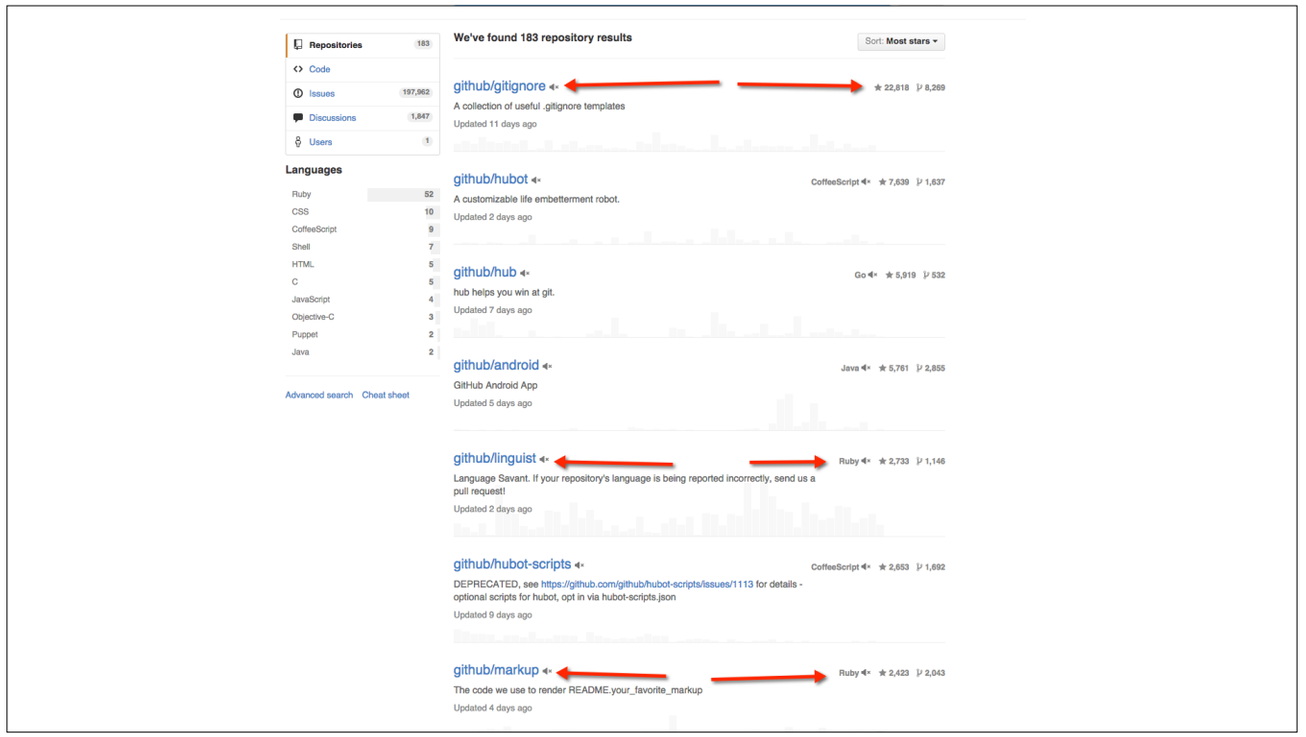
7 contributors

14 lines (10 sloc) | 0.173 kb | **Raw** | **Blame** | **History**

```
1  # History files
2  .Rhistory
3  .Rapp.history
4
5  # Example code in package build process
6  *-Ex.R
7
8  # RStudio files
9  .Rproj.user/
10
11 # produced vignettes
12 vignettes/*.html
13 vignettes/*.pdf
```


testing /
.gitignore
or cancel
Want to use a .gitignore template?
Choose .gitignore: R
Edit new file
Preview
Spaces
2
No wrap
History files
.Rhistory
Example code in package build process
*-Ex.R
R data files from past sessions
.Rdata
RStudio files
.Rproj.user/


github/linguist
Garen
GitHub, Inc. [US] https://github.com/github/linguist
Troubleshooting
My repository is detected as the wrong language
CSS 50.7%
JavaScript 47.9%
Python 1.1%
Other 0.3%
The Language stats bar is built by aggregating the languages of each file in that repository. If it is reporting a language that you don't expect:
1. Click on the name of the language in the stats bar to see a list of the files that are identified as that language.
2. If you see files that you didn't write, consider moving the files into one of the paths for vendored code, or use the manual overrides feature to ignore them.
3. If the files are being misclassified, search for open issues to see if anyone else has already reported the issue. Any information you an add, especially links to public repositories, is helpful.
4. If there are no reported issues of this misclassification, open an issue and include a link to the repository or a sample of the code that is being misclassified.
Overrides
Linguist supports a number of different custom overrides strategies for language definitions and vendored paths.
Using gitattributes
Add a .gitattributes file to your project and use standard git-style path matchers for the files you want to override to set linguist-documentation, linguist-language, and linguist-vendored. .gitattributes will be used to determine language statistics, but will not be used to syntax highlight files. To manually set syntax highlighting, use Vim or Emacs modelines.
$ cat .gitattributes
*.rb linguist-language=Java
Checking code you didn't write, such as JavaScript libraries, into your git repo is a common practice, but this often inflates your project's language stats and may even cause your project to be labeled as another language. By default, Linguist treats all of the paths defined in lib/linguist/vendor.yml as vendored and therefore doesn't include them in the language statistics for a repository. Vendored files are also hidden by default in diffs on github.com.


Repositories 183
Code
Issues 197,962
Discussions 1,847
Users 1
Languages
Ruby 52
CSS 10
CoffeeScript 9
Shell 7
HTML 5
C 5
JavaScript 4
Objective-C 3
Puppet 2
Java 2
Advanced search Cheat sheet
We've found 183 repository results
Sort: Most stars
github/gitignore
★ 22,818 ⑂ 8,269
A collection of useful .gitignore templates
Updated 11 days ago
github/hubot
CoffeeScript ★ 7,639 ⑂ 1,637
A customizable life embetterment robot.
Updated 2 days ago
github/hub
Go ★ 5,919 ⑂ 532
hub helps you win at git.
Updated 7 days ago
github/android
Java ★ 5,761 ⑂ 2,855
GitHub Android App
Updated 5 days ago
github/linguist
Ruby ★ 2,733 ⑂ 1,146
Language Savant. If your repository's language is being reported incorrectly, send us a pull request!
Updated 2 days ago
github/hubot-scripts
CoffeeScript ★ 2,653 ⑂ 1,692
DEPRECATED, see https://github.com/github/hubot-scripts/issues/1113 for details - optional scripts for hubot, opt in via hubot-scripts.json
Updated 9 days ago
github/markup
Ruby ★ 2,423 ⑂ 2,043
The code we use to render README.your_favorite_markup
Updated 4 days ago