Dashboard / ... / Old Stuff

I - Landscape

Created by Chris Ernst, last modified on 2016-10-17

Similar Products

The following items are standards for developers in regards to Developer Portals. These are deployments that exist across industry. Items in green work well and we should model after them. Items in yellow work OK, but could be good with a little work. Items in red do not work and give us an opportunity to offer a better experience than our competitors.

Payeezy

https://developer.payeezy.com

This site was used as an example on what we should do. A lot of really great features!

- Copy/pasteable example code
- Informative
- Sandbox as part of the entry
- Interactive demo
- Some weirdness ("click here")
- Layout
- Collapsed (and defaults)
- Linking to similar/related entries

List View, Individual View

Brain Tree

https://developers.braintreepayments.com/ios+ruby/

This is a company owned by Paypal.

- Step-by-step guides
- Layout
- Choose your language
- Guide-based documentation
- Not super informative (or Braintree is fairly simple)
- Taxonomy not really clear (but maybe it's not important for Braintree)
- Sandbox is a separate area

Landing Page

Unity3D

http://docs.unity3d.com/ScriptReference/

This is the documentation for a very popular video game engine.

- Searchable
- Informative
- Tutorials
- Choose your language
- Examples not on every entry
- Layout
- Related entries only via taxonomy

Navigation

Stripe

https://stripe.com/docs

A lightweight payments processing tool.

- High-level overview
- Choose your language
- Community contributions
- Guide- or action-based reference
- No sandbox?

Landing Page

AngularJS

https://docs.angularjs.org/api

AngularJS API Docs

- Layout
- Copy/pasteable example code
- Sandbox in entry, editable via Plunkr (but opens in same tab!)
- Good search
- Navigation, related entries via taxonomy
- Steep learning curve (could just be Angular)

Simplify

https://www.simplify.com/commerce/docs/api/index

API/SDK documentation/examples for Simplify Commerce by Mastercard

- Copy/pasteable example code
- Choose your langauge
- Layout
- IA
- No plain description in entry
- Scrolling seems non-native (or this page is really heavy)
- No scrollspy when it seems like it would have it
- No sandbox

Apple

https://developer.apple.com/

Apple dev portal: references, account, announcements, etc

- Choose your language
- Choose your version
- Related documentation
- Verbose (tables, diagrams, prose)
- Verison availability
- Large-scope portal for many purposes
- Layout (could be my unfimilairity with xcode, swift, etc)
- Taxonomy
- Monolithic

Mozilla Developers Network

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/split

General web dev API reference for HTML, CSS, JS, etc.

- Copy/pasteable example code
- High-level description
- Verbose
- Several kinds of examples
- Metadata (spec it comes from, browser compatibility)
- Clear taxonomy
- Layout
- No scrollspy when it seems like it would have it

Potential Tools

These are some tools, frameworks, etc that might be useful to make Bleach.

Swagger

http://java.dzone.com/articles/swagger-great

A breakdown of Swagger.

Akana

API management system that has built in:

- Authorization
- Documentation
- Versioning
- Forum
- Rating
- Sandbox

Hugo

Static site generator that takes in markdown and updates in realtime.

Considerations

This information should not be considered definitive. We are building a list of what is needed for Bleach.

Bleach should include API, Code, Apple/Samsung Pay, a sandbox, style guides, and component libraries. The first two products should be Converge and Commerce SDK, with a focus on the former.

We need to consider the type of API we are dealing with, such as REST or Java. We also must provide endpoints and credentials.

API Docs

- Name / Description (What does this do)
- Resource URL
- Resource Parameters
- Resource Summary
    - Request
        - Model Schema
        - Property name
        - Expected Property type (string, object, etc.)
        - Property description
    - Response

- Model Schema
- Property name
- Expected Property type (string, object, etc.)
- Property description
- Errors
  - Model Schema
  - HTTP status code (400, 500, etc.)
  - HTTP status code descriptions
  - API error codes
  - API error code descriptions

API Testing

- Includes test cards data (other such content required) if applicable
- Editable/Copy/Paste-able parameter/request values

App Management

- Devs (and others) can see their different project/products
  - Performance
  - Metrics
  - Reports
- App-level changes
  - Stop/start
  - Fees?
  - Etc?

No labels

ATLASSIAN