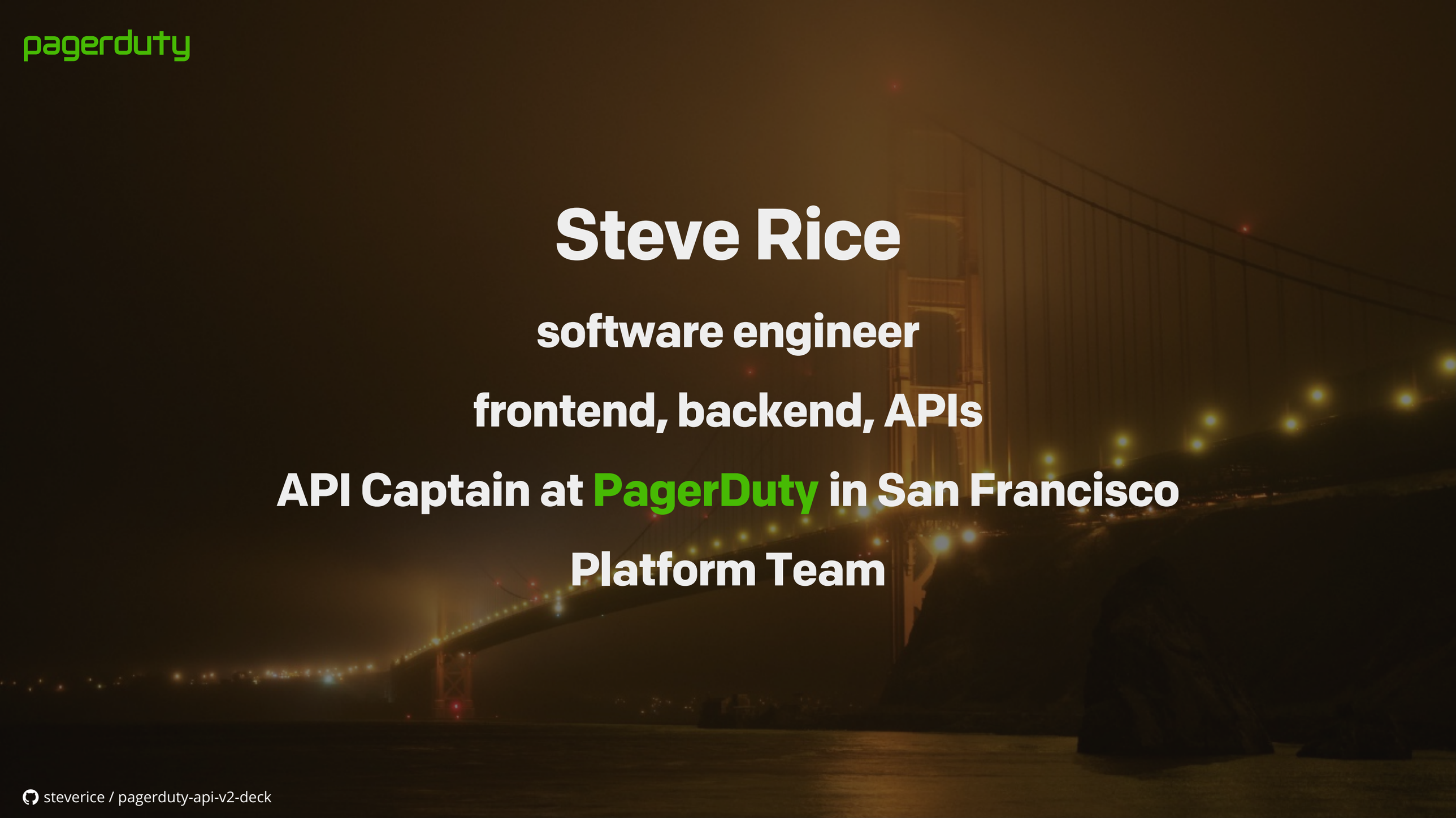
pagerduty

API Creation to Iteration

Without the Frustration

steverice / pagerduty-api-v2-deck

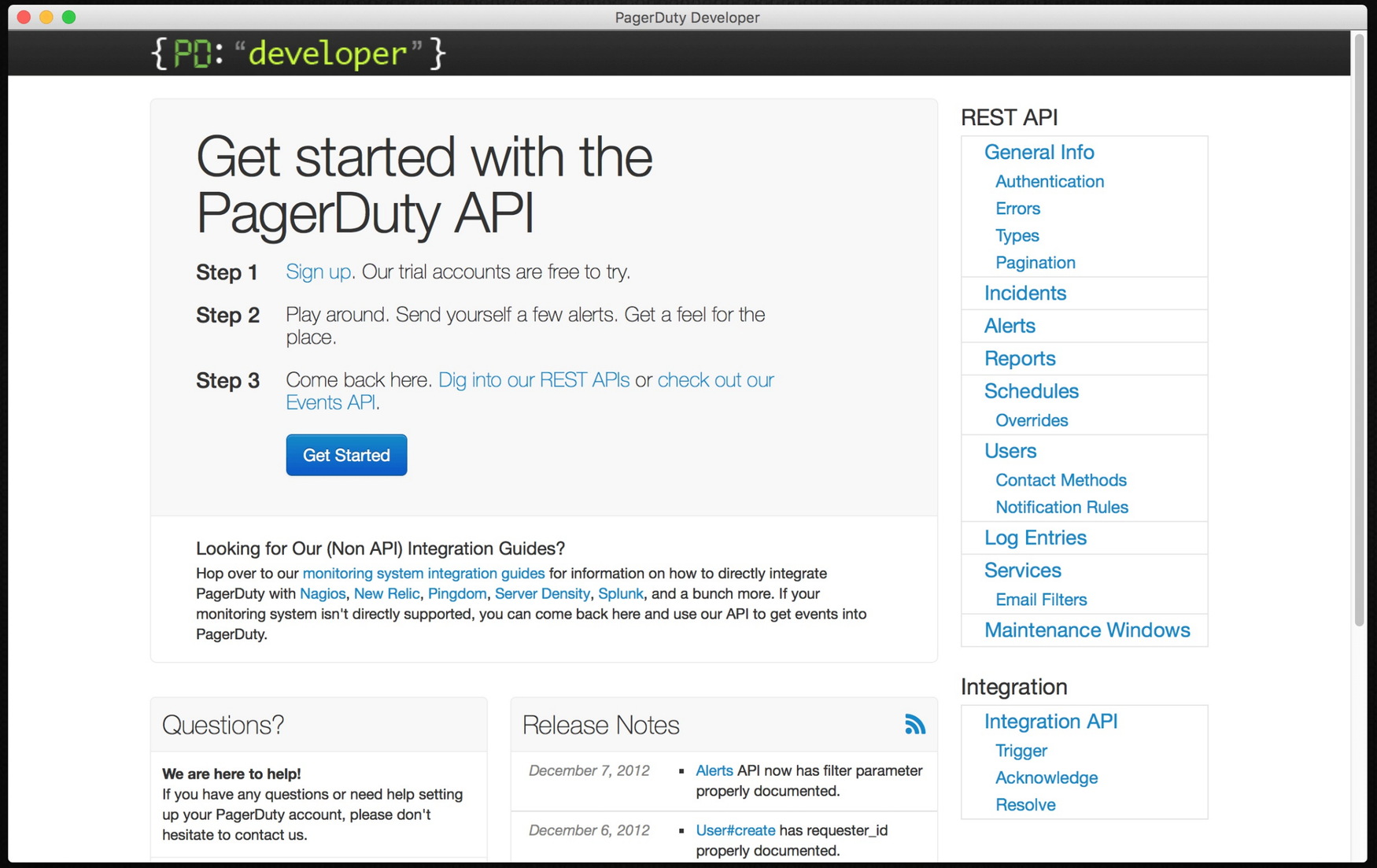

pagerduty
Steve Rice
software engineer
frontend, backend, APIs
API Captain at PagerDuty in San Francisco
Platform Team
steverice / pagerduty-api-v2-deck

pagerduty

What is PagerDuty?

- incident resolution platform
- triage events
- notify responders
- manage incident response
- SaaS product
- ~250 employees, started with half that
- Rails, MySQL, Scala, Cassandra, Kafka, etc.
- Backbone, Ember.js, Android + iOS

pagerduty

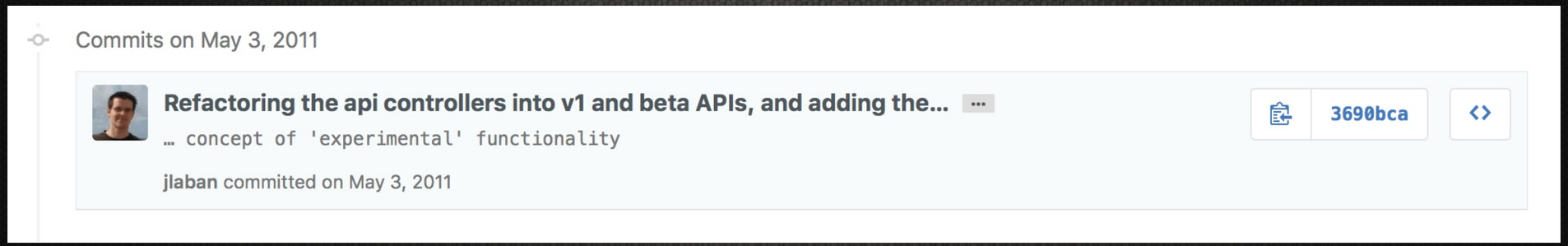


https://web.archive.org/web/20130116145245/http://developer.pagerduty.com/

steverice / pagerduty-api-v2-deck

pagerduty

```
# This controller is for communication between the main app
# server and other processes in the system. The idea here is
# that this shouldn't be exposed to the outside world.
```



pagerduty

https://web.archive.org/web/20130312085114/http://blog.pagerduty.com/2012/09/you-have-the-power/

pagerduty

PagerDuty Developer

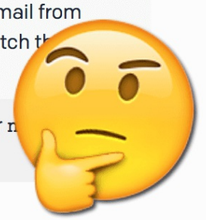
Update an existing Email Filter.

Resource URL

```
PUT https://<subdomain>.pagerduty.com/api/v1/services/:service_id/email_filters/:id
```

Parameters

Name	Type	Required	Description
subject_mode	String	No	One of `always`, `match`, `no-match`, which, respectively, means to not filter the email trigger by subject, filter it if the email subject matches the given regex, or filter if it doesn't match the given regex. Defaults to `always`.
subject_regex	String	No	The regex to be used when subject_mode is `match` or `no-match`. It is a required parameter on such cases.
body_mode	String	No	One of `always`, `match`, `no-match`, which, respectively, means to not filter the email trigger by body, filter it if the body email matches the given regex, or filter if it doesn't match the given regex. Defaults to `always`.
body_regex	String	No	The regex to be used when body_mode is `match` or `no-match`. It is a required parameter on such cases.
from_email_mode	String	No	One of `always`, `match`, `no-match`, which, respectively, means to not filter the email trigger by its from address, filter it if the email from address matches the given regex, or filter if it doesn't match th regex. Defaults to `always`.
from_email_regex	String	No	The regex to be used when from_email_mode is `match` or n is a required parameter on such cases.



Example

steverice / pagerduty-api-v2-deck

pagerduty

Research


pagerduty
Stand on the Shoulders of Giants
steverice / pagerduty-api-v2-deck

pagerduty

Conferences and Talks

https://www.heavybit.com/library/video/move-fast-dont-break-api/
http://amberonrails.com/move-fast-dont-break-your-api/


pagerduty
Dropbox Developer Blog
Topics
Subscribe
Dropbox blogs
A preview of the new Dropbox API v2
Leah Culver | April 8, 2015
0
4
9
[EDIT June 3, 2015] This post has been updated to reflect the latest API v2 syntax.
structural changes underway, and we'd like
know what you think!
Developer Blogs
https://blogs.dropbox.com/developers/2015/04/a-preview-of-the-new-dropbox-api-v2/
Overall, we've simplified our use of HTTP. For example, most endpoints always use HTTP POST, including those that return structured data. Requests take JSON in the body and responses return JSON in the body.
We will continue to use other HTTP features in specific cases where they provide concrete benefits. For example, endpoints that return bulk binary data (e.g. file contents) support HTTP GET and ETag-based caching. This allows browsers and HTTP client libraries to transparently cache that data for you.
We've also simplified our use of HTTP status codes for errors. For errors that are common to all API calls, we do the same thing as before: 400 for bad request, 401 for auth failure, 429 for rate limiting, etc. But if a request fails for some call-specific reason, v1 might have returned any of 403, 404, 406,
steverice / pagerduty-api-v2-deck

pagerduty

Protocols and Hypermedia Types

- HAL: http://stateless.co/hal_specification.html
- OData: http://www.odata.org
- Collection+JSON: https://github.com/collection-json/spec
- JSON-API: http://jsonapi.org
- JSON-LD: http://json-ld.org
- GData: https://developers.google.com/gdata/
- GraphQL: http://graphql.org
- Siren: https://github.com/kevinswiber/siren

https://sookocheff.com/post/api/on-choosing-a-hypermedia-format/

pagerduty

Plan


pagerduty
Make decisions.
steverice / pagerduty-api-v2-deck

pagerduty

Branch: master ▾ pygerduty / pygerduty / __init__.py Find file Copy path

gmjosack Fix bug in datetime encoding 05326d5 on Aug 30

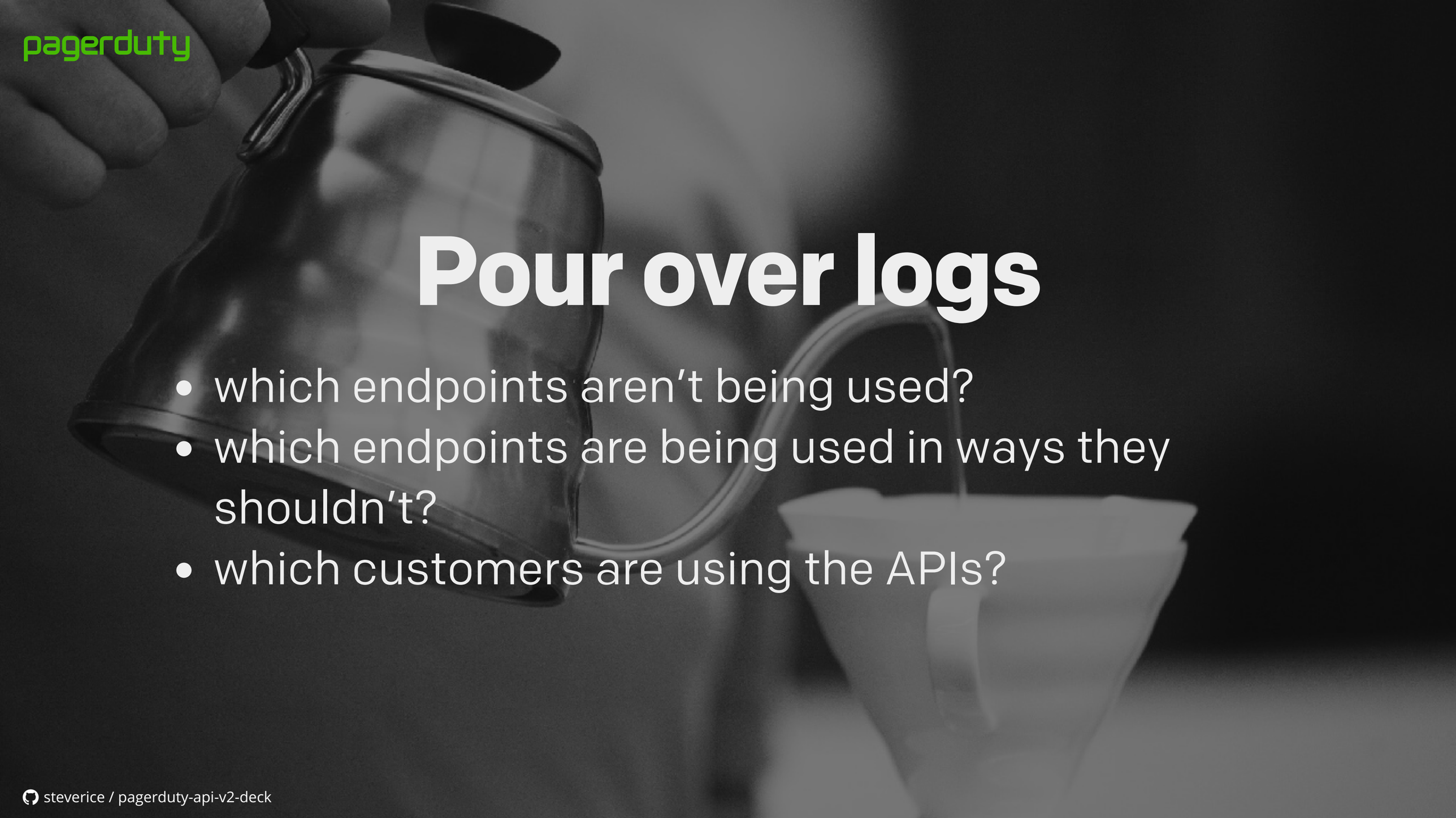
14 contributors

Executable File | 762 lines (562 sloc) | 23.3 KB Raw Blame History

```
396 class Incident(Container):
397     def __init__(self, *args, **kwargs):
398         Container.__init__(self, *args, **kwargs)
399         self.log_entries = LogEntries(self.pagerduty, self)
400         self.notes = Notes(self.pagerduty, self)
401
402     def _do_action(self, verb, requester_id, **kwargs):
403         path = '{0}/{1}/{2}'.format(self.collection_name, self.id, verb)
404         data = {'requester_id': requester_id}
405         data.update(kwargs)
406         return self.pagerduty.request('PUT', path, data=_json_dumper(data))
407
408     def has_subject(self):
409         return hasattr(self.trigger_summary_data, 'subject')
410
411     def resolve(self, requester_id):
412         self._do_action('resolve', requester_id=requester_id)
413
414     def acknowledge(self, requester_id):
415         self._do_action('acknowledge', requester_id=requester_id)
416
417     def snooze(self, requester_id, duration):
418         self._do_action('snooze', requester_id=requester_id, duration=duration)
419
420     def get_trigger_log_entry(self, **kwargs):
421         match = TRIGGER_LOG_ENTRY_RE.search(self.trigger_details_html_url)
422         return self.log_entries.show(match.group('log_entry_id'), **kwargs)
```

Go to the Source

https://github.com/dropbox/pygerduty


pagerduty
Pour over logs
• which endpoints aren’t being used?
• which endpoints are being used in ways they shouldn’t?
• which customers are using the APIs?
steverice / pagerduty-api-v2-deck

pagerduty

An example of a relationship is Notification Rules, whose primary meaning is as a link between a user and a contact method. No individual notification rule has a unique, name-worthy meaning, and so they are handled (read and written) only as a collection. There is no compelling reason to update a single notification rule in place rather than replace the entire object with a new one. While this might seem like a significant loss of functionality, data shows that a trivially small number of clients outside of the PagerDuty web app are using it:

- over the last 30 days, only 11 requests were made to `PUT notification_rules/id` using non-cookie authentication, and all by the subdomain `redacted`. While the mobile team experimented with it in the latest "Edit Notification Rules" update in 3.6, they've confirmed that that endpoint isn't being used.

Similarly, the escalation rules relationship sees little in-place use:

- over the last 30 days, only one request was made to `PUT escalation_rules/id` using non-cookie authentication.

pagerduty

This document outlines the vision and scope for v2 of the API.

Ordered lists represent prioritized order. We recannot accomplish everything at once, and thus everything needs to be prioritized sooner or later. I choose sooner.

This document does not attempt to present a comprehensive record of all the changes that will be made, as those will be captured by stories in JIRA. Instead, it establishes the goals, principles, and patterns that we'll use to make the decisions about what needs to change.

It is a living document and will evolve as we discover new edge cases and outliers that weren't previously considered.

Goals

What are we trying to accomplish with API v2?

These are the high-level goals, in priority order:

1. iteration
    - leave well enough alone
2. adoption
    - prove the effectiveness of our versioning strategy
3. consistency
    - provide consistency across endpoints
4. performance
    - fix design decisions that have led to performance problems
        - see Deprecated API Functionality
    - enable greater cacheability
5. semantics
    - better use of generic HTTP libraries and patterns

pagerduty

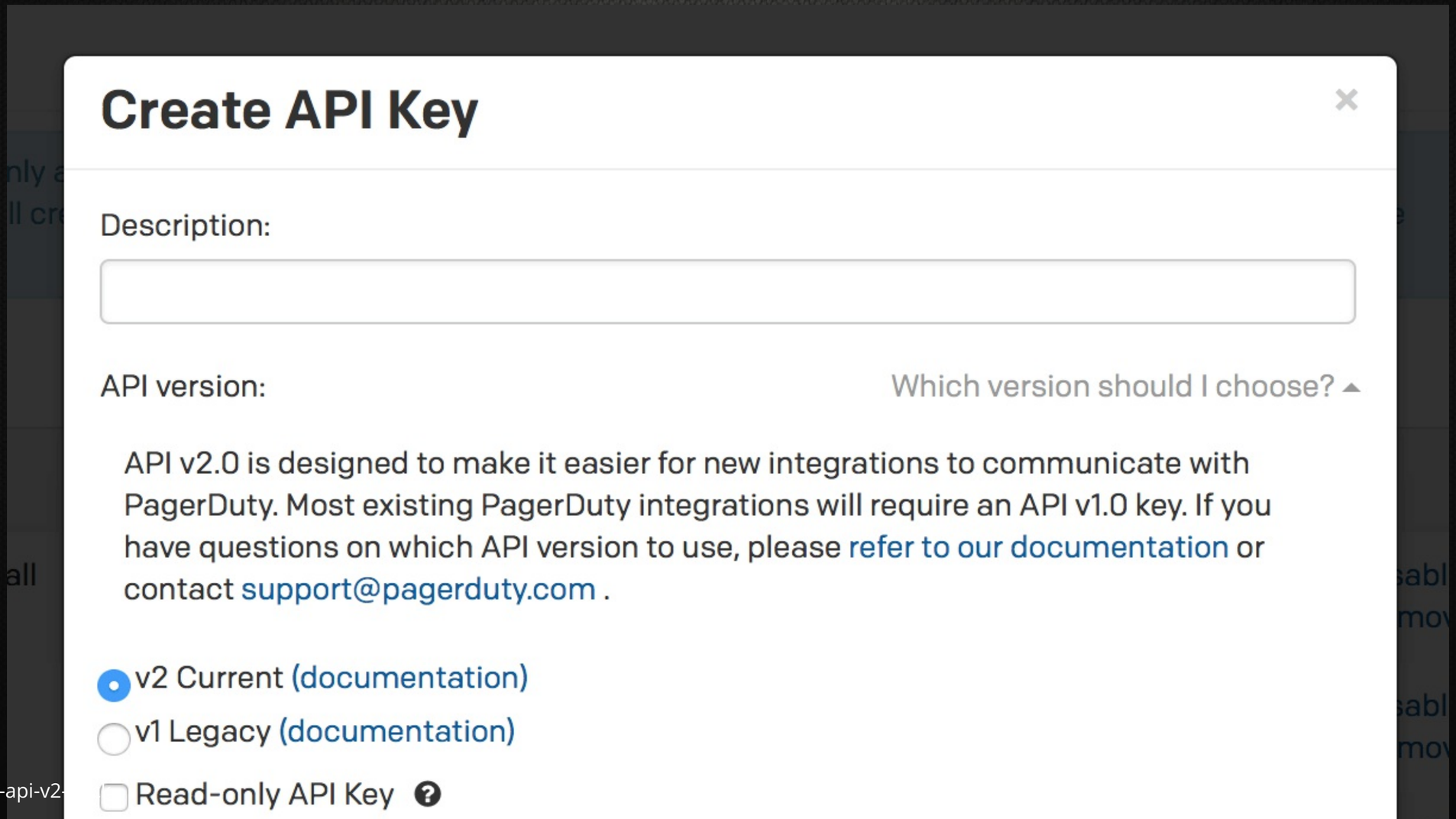
Pick your Battles

- Don’t change what works
- Prioritize
- Keep bullets away from feet

pagerduty

header- and key-based versioning

```
Accept: application/vnd.pagerduty+json;version=2
```



pagerduty

canonical API subdomain

~~initech.pagerduty.com~~

~~acmemonitoring.pagerduty.com~~

~~evilcorp.pagerduty.com~~

api.pagerduty.com

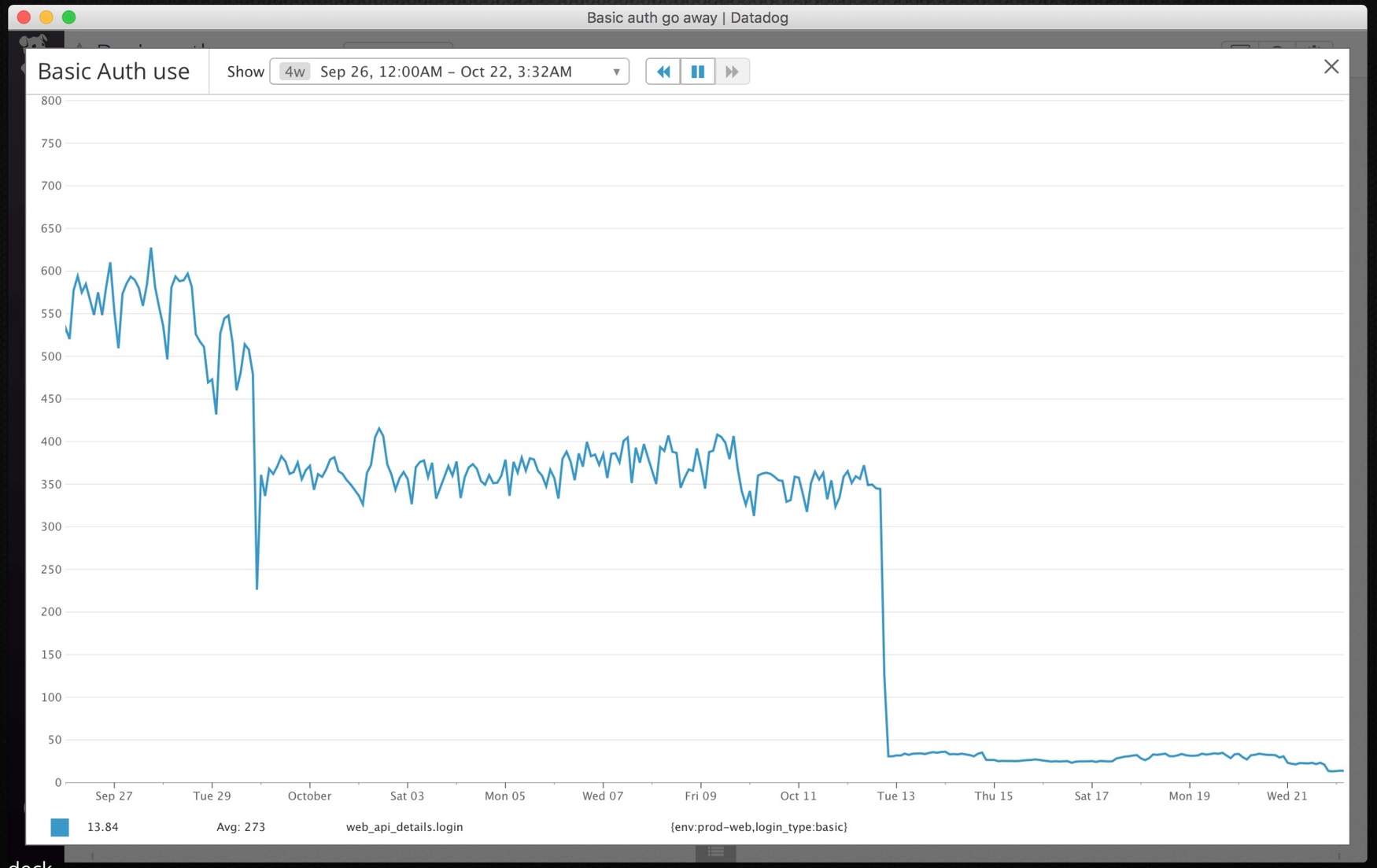
pagerduty

use standards

- respect HTTP semantics
- ISO 8601
- IANA tzinfo

pagerduty

deprecate basic auth



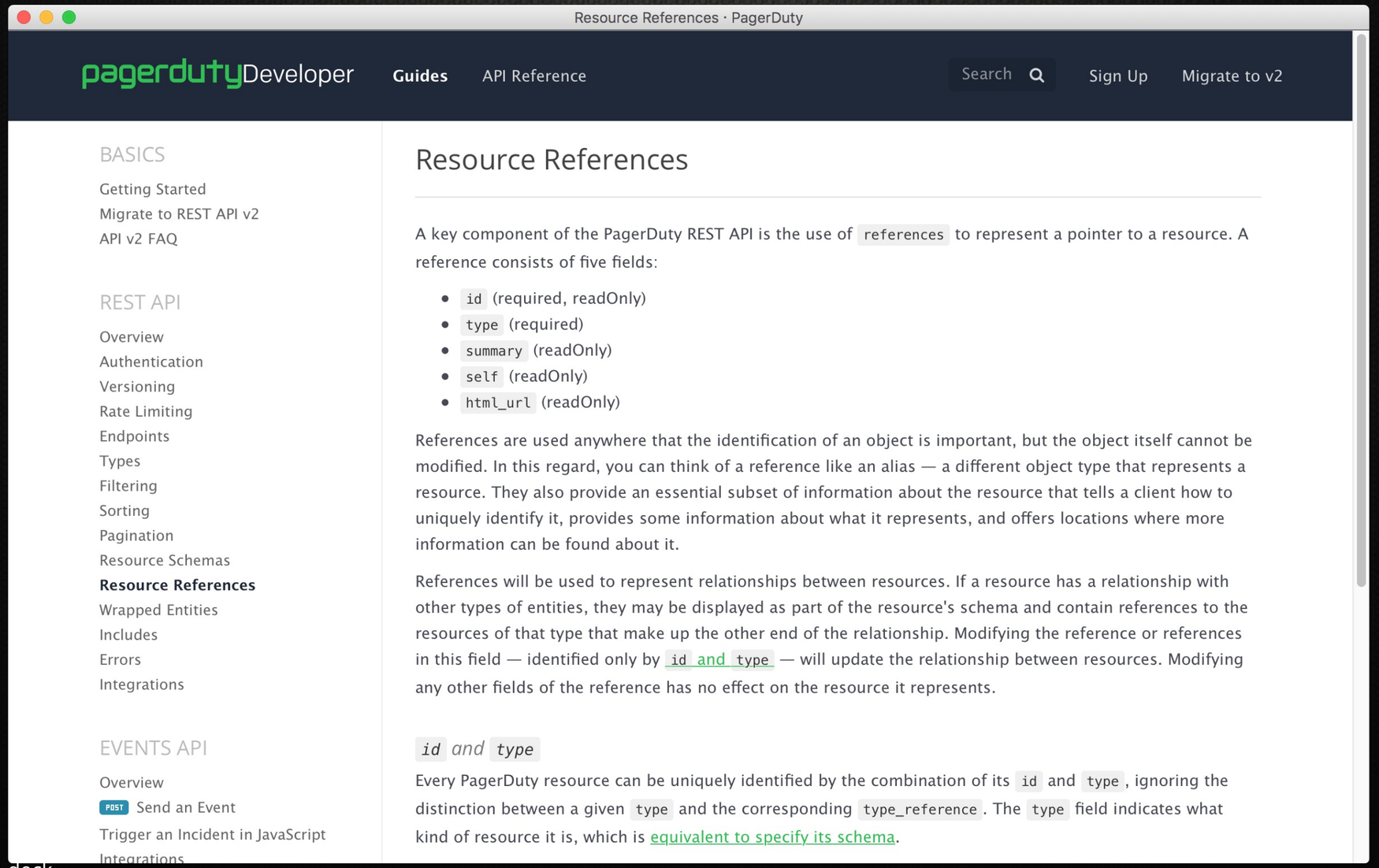
pagerduty

performance tweak

```
{
  "incidents": [{
    "id": "P1R8NC6",
    "type": "incident",
    "summary": "[#7893] There are 2 Charizard nearby!",
    "self": "https://api.pagerduty.com/incidents/P1R8NC6",
  }],
  "limit": 1,
  "offset": 0,
  "total": null, //don't care about this
  "more": true //more resources available
}
```

pagerduty

consistency



pagerduty

Change the Plan

Clarify: include[] includes resources throughout the object graph · PagerDuty/captains-log@e67771b

Showing **1 changed file** with **3 additions** and **1 deletion.**

Unified | Split

4 proposals/api-v11-vision.md

View

An `include` array must be used to specify any and all related resources that should be used in a response

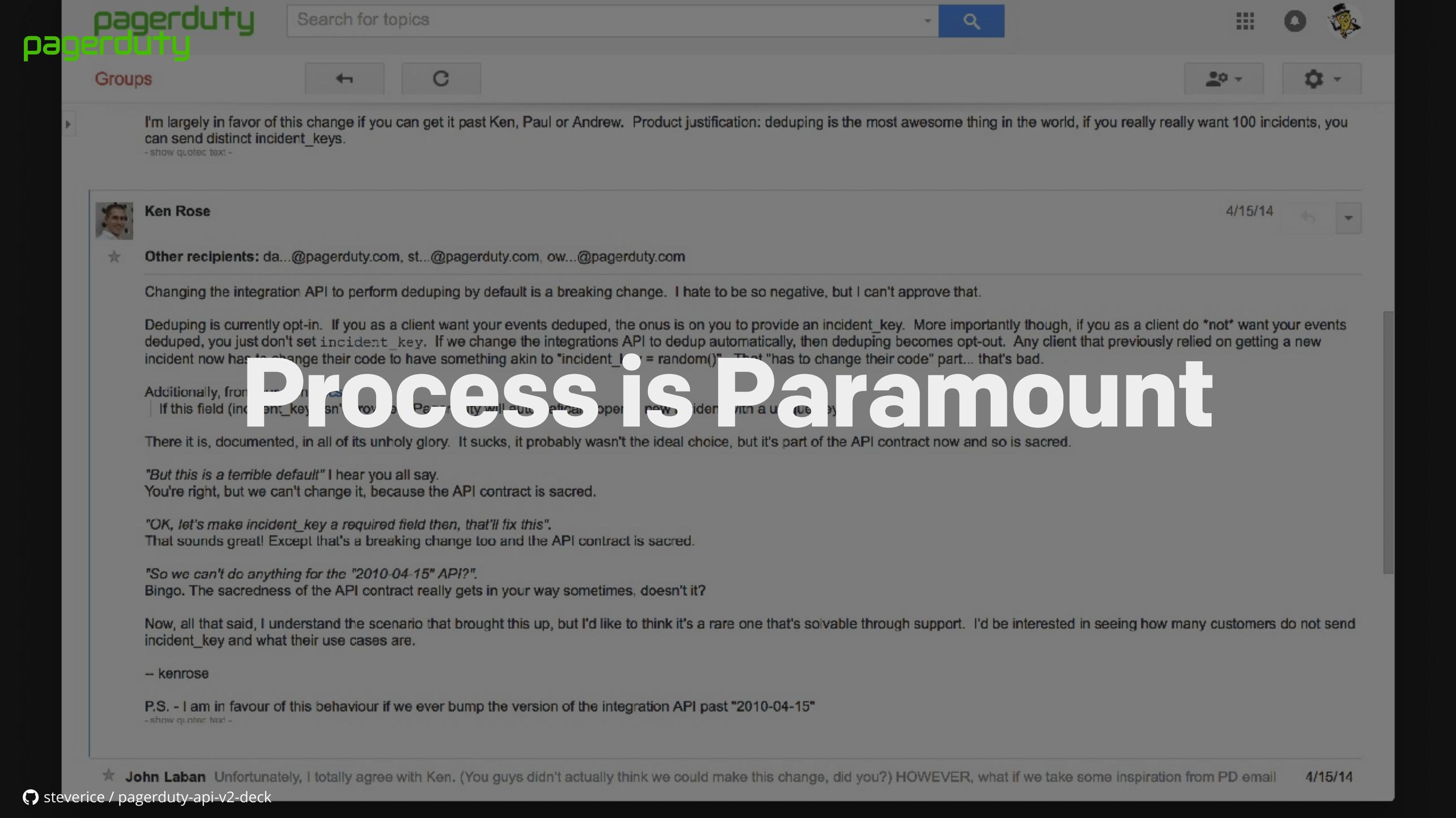
- `include` **must** be an array
- `include` must **only** be used for including resources, not other entities nor extra fields/metadata
- ~~`include` is one level only, there are no nested includes (e.g. `include[]=incidents.trigger_log_entry`)~~
- an `include` includes resources at all levels
  - there is no syntax for specifying includes at a specific level (e.g. `include[]=incidents.trigger_log_entry` or `include[]=services/incidents/log_entries`)
  - the API will prevent infinite recursion, e.g. `include[]=users` on a `/teams` endpoint will show full `users` inside that team, and each user will have an array of `teams` references. However, those `teams` references will not show full `users`.
- `include` cannot have any default values; i.e. no resources can be explicitly included by default.
  - the exception to this is that when `POST`ing our `PUT`ing a resource containing nested resources that would ordinarily require `include`-ing, those resources will be included in the response
- `include` must be populated solely by either the *pluralized* name of a resource or polymorphic relationship

For example, the v1.0 schedules API allows including a user's next `oncall` via:

0 comments on commit `e67771b`

Lock conversation

steverice / pagerduty-api-v2-



Process is Paramount

pagerduty

- your prototype is not making any breaking changes to existing contracted APIs

What constitutes approval?

An API is not approved until it's been merged into the Captain's Log.

Wild Clients

Any client that PagerDuty cannot, at a whim, revoke or modify without customer impact, is a **"wild client"**. These clients are outside PagerDuty's control. These include:

- Mobile apps that have been shipped to a public app store
- Clients or client libraries written by third parties based on our published documentation
- Client libraries that have been written and released by PagerDuty

Managed Clients

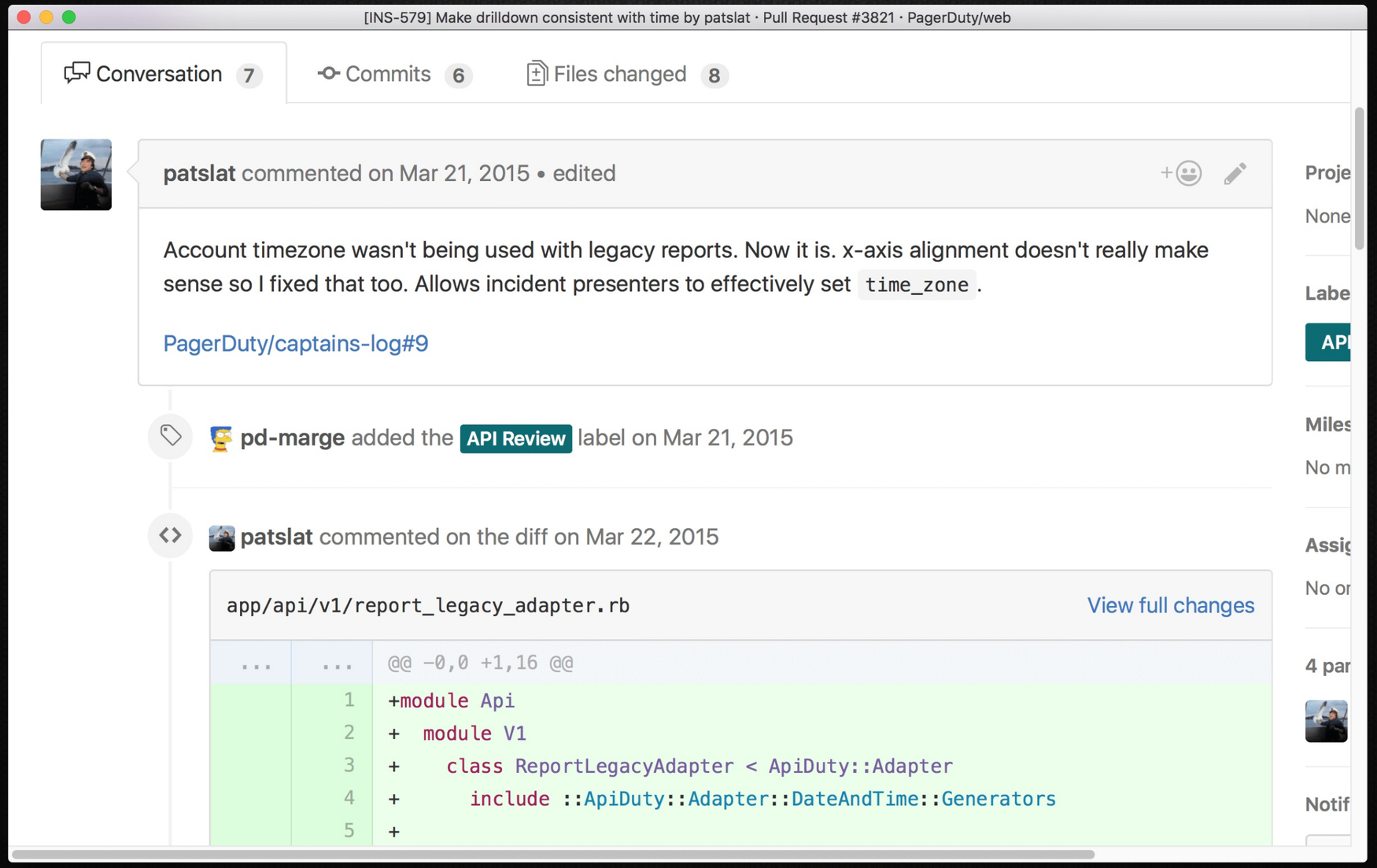
Any client whose distribution is controlled entirely by PagerDuty is a **"managed client"**. This control includes the ability to stop users of the client from being able to use it at any time.

Clients that *are* under PagerDuty's control include:

- the Backbone-based Web app
- the Ember-based Web UI app
- the mobile-web app
- Watchdog
- Mobile apps that are only available on an internal testing platform (Crashlytics Beta, Deploygate)

However, you should still expect to modify managed clients if you're making a change to an API change they depend upon — otherwise the client will break!

pagerduty



https://github.com/PagerDuty/flagger

pagerduty

Document

steverice / pagerduty-api-v2-deck


pagerduty
Keep it maintainable
steverice / pagerduty-api-v2-deck

```
          </td>
          <td>
            <a href="/documentation/rest/types#object">Object</a>
          </td>
          <td>
            The PagerDuty <a href="/documentation/rest/services/">service</a> that the incident belongs to. The service will contain fields of i
          </td>
        </tr>
        <tr>
          <td>
            escalation_policy
          </td>
          <td>
            <a href="/documentation/rest/types#object">Object</a>
          </td>
          <td>
            The <a href="/documentation/rest/escalation_policies/">escalation policy</a> that the incident belongs to. The policy will contain f
          </td>
        </tr>
        <tr>
          <td>
            teams
          </td>
          <td>
            <a href="/documentation/rest/types#array">Array</a>
          </td>
          <td>
            <p>
              The <a href="/documentation/rest/teams/">teams</a> involved in the incident’s lifecycle. The teams will contain fields of their ow
            </p>
            <p>
              <strong>NOTE:</strong> You must explicitly include <code>teams</code> in the <code>fields</code> parameter in order to include it
```

pagerduty

Domain-specific technologies

pagerduty

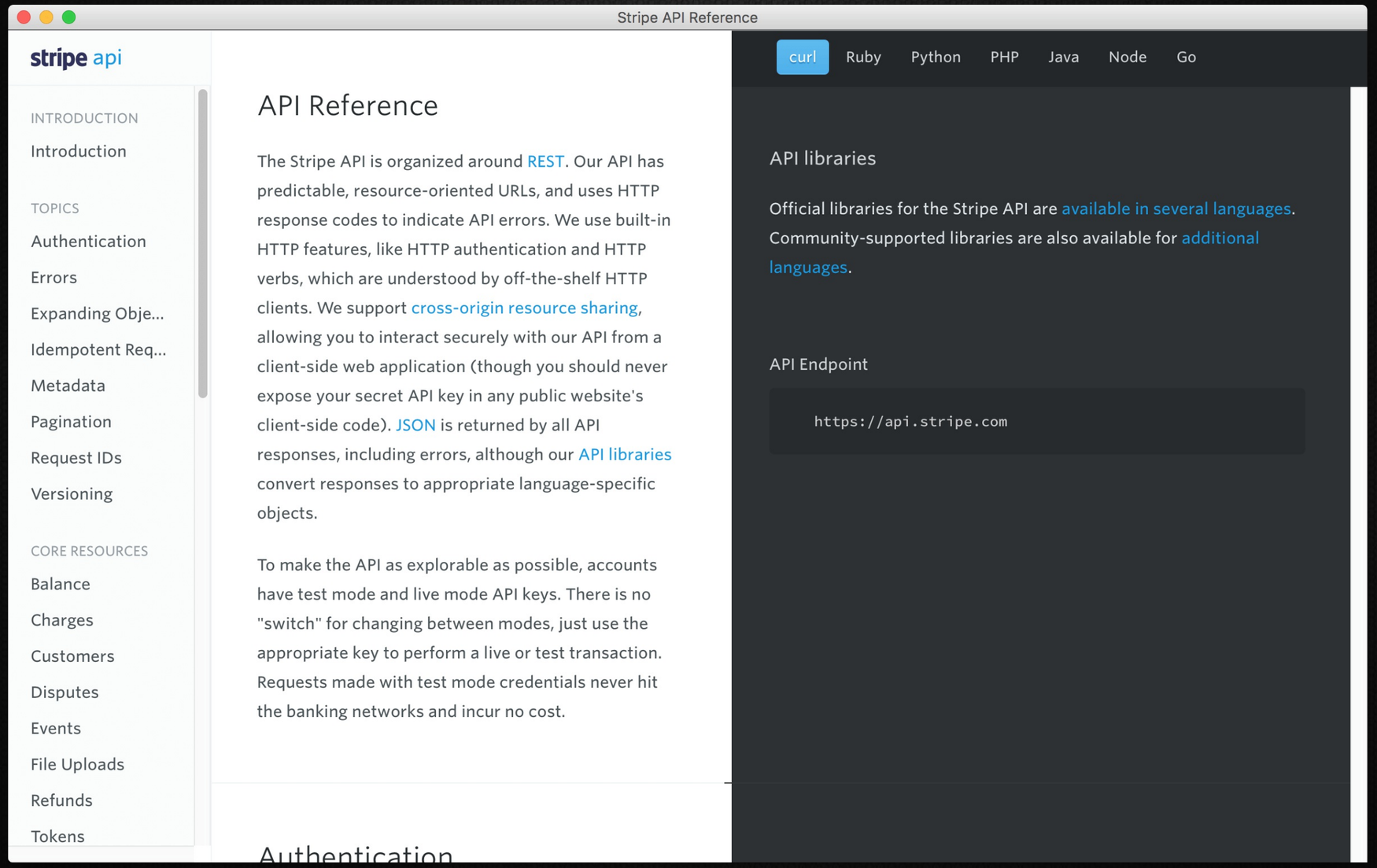
```
4283                "parameters": [{
4284                        "name": "team",
4285                        "in": "body",
4286                        "description": "The team to be created.",
4287                        "schema": {
4288                                "type": "object",
4289                                "properties": {
4290                                        "team": {
4291                                                "$ref": "#/definitions/Team"
4292                                        }
4293                                },
4294                                "required": ["team"]
4295                        }
4296                }],
4297                "responses": {
4298                        "201": {
4299                                "description": "The team that was created.",
4300                                "schema": {
4301                                        "type": "object",
4302                                        "properties": {
4303                                                "team": {
4304                                                        "$ref": "#/definitions/Team"
4305                                                }
4306                                        },
4307                                        "required": ["team"]
4308                                },
4309                                "examples": {
4310                                        "application/json": {
4311                                                "team": {
4312                                                        "id": "PQ9K7I8",
4313                                                        "type": "team",
4314                                                        "summary": "Engineering",
```

pagerduty

Consider the Humans

steverice / pagerduty-api-v2-deck

pagerduty



https://stripe.com/docs/api

pagerduty

List incidents GET

REQUEST URL

```
https://api.pagerduty.com/incidents?time_zone=UTC
```

"CURL" COMMAND

```
curl -X GET --header 'Accept: application/vnd.pagerduty+json;version=2' --header 'Authorization: Token token=w_8PcN
uhHa-y3xYdmc1x' 'https://api.pagerduty.com/incidents?time_zone=UTC'
```

REQUEST HEADERS

```
{
  "Accept": "application/vnd.pagerduty+json;version=2",
  "Authorization": "Token token=w_8PcNuhHa-y3xYdmc1x"
}
```

RESPONSE BODY


```
{
  "incidents": [
    {
      "incident_number": 54147,
      "description": "Rats chewed through the fiber.",
      "created_at": "2016-09-10T10:19:11Z",
      "status": "resolved",
      "pending_actions": [],
      "incident_key": "srv01/HTTP",
      "service": {
        "id": "P3U7V58",
        "type": "service_reference",
        "summary": "Website",
```


Make it interactive

steverice / pagerduty-api-v2-deck

pagerduty

pagerdutyDeveloper Guides API Reference Search Sign Up Migrate to v2

BASICS

Getting Started
Migrate to REST API v2
API v2 FAQ

REST API

Overview
Authentication
Versioning
Rate Limiting
Endpoints
Types
Filtering
Sorting
Pagination
Resource Schemas
Resource References
Wrapped Entities
Includes
Errors
Integrations

EVENTS API

Overview
POST Send an Event
Trigger an Incident in JavaScript

Migrate to REST API v2

Migrating your existing REST API v1 client to API v2

Migrating your existing PagerDuty REST API v1 integration to API v2 will let you leverage greatly increased consistency to simplify your code. It will also allow your integration to take advantage of new v2-only features and functionality coming to the API in the future.

A number of things have changed between API v1 and API v2. While this document covers the majority of the [illegible] are not covered [illegible] how API v2 endpoints behave.

Patterns followed in API v2 were typically present in API v1, but may have been followed inconsistently across different endpoints. Review your code to determine where a pattern is already being followed and where it needs to be [illegible]

Remember your audience

> **REST API Only**
>
> This guide applies only to requests made against the PagerDuty REST API. The format and function of the Events API has not changed.

Versioned Accept Header

To make your code talk with API v2 instead of API v1, you'll need to add the versioned PagerDuty `Accept` header to any API requests being made by your code:

```
Accept: application/vnd.pagerduty+json;version=2
```

pagerduty

Develop

pagerduty

Keep it Simple

```
class Api::V2::EscalationPolicyAdapter::Base < ApiDuty::Adapter
  root_attribute :escalation_policy

  validate :name, validator: :is_string
  validate :escalation_rules, validator: :is_array
  validate :description, validator: :is_string
  validate :num_loops, validator: :is_integer
  validate :teams, validator: :is_array
  validate :teams, validator: make_validator_each(
      make_validator_object_resource(%w(team).freeze)
    )

  passthrough :name, :description
  generate :num_repeats, from: :num_loops do |num|
    num.to_i + 1
  end
```

pagerduty

Keep it Simple

```
module Api
  module V2
    class SchedulesController < Api::V1::SchedulesController
    end
  end
end
```

pagerduty

Be pragmatic

DRY is not a god

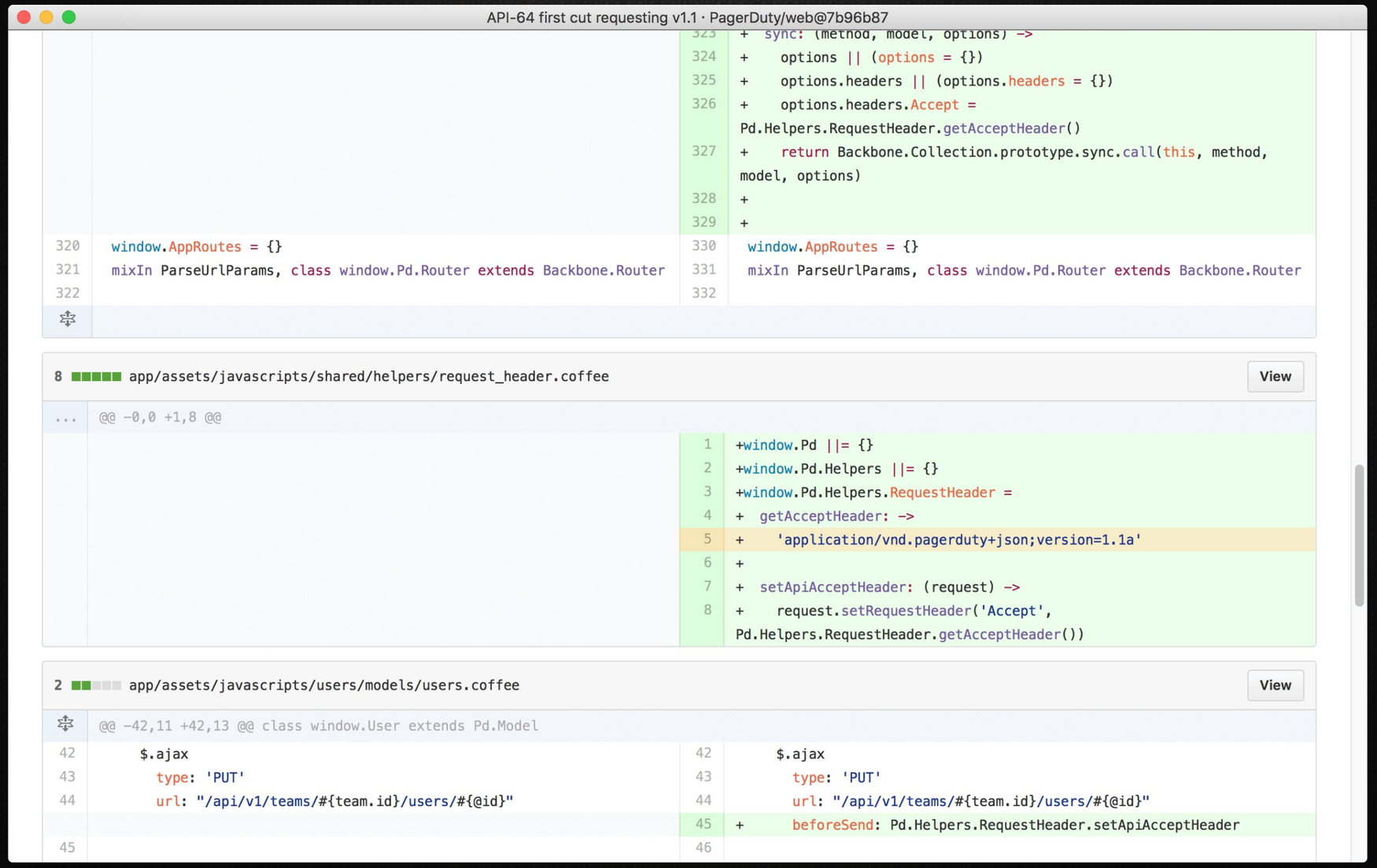
```
def unwrapped_incident_json
  # V2+: incident object is wrapped.
  json[:incident]
end

def assert_assigned_to_user(json, user)
  # V2+: assigned_to_user is removed
  assert_not_include json.keys, 'assigned_to_user'
end

def assert_reassigned_to_user(json, user)
  # V2+: assigned_to_user is removed
  assert_not_include json.keys, 'assigned_to_user'
end

def assert_incident_count(expected, json)
```

pagerduty

Test

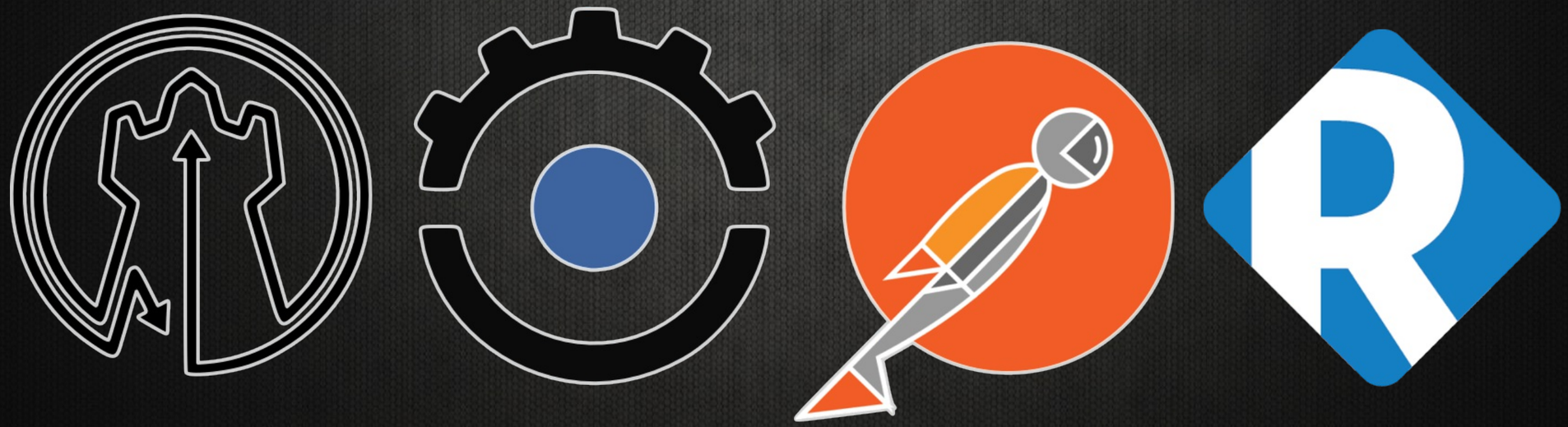

pagerduty
Dogfood.
Everything.
steverice / pagerduty-api-v2-deck

pagerduty


API-64 first cut requesting v1.1 · PagerDuty/web@7b96b87
+ sync: (method, model, options) ->
+ options || (options = {})
+ options.headers || (options.headers = {})
+ options.headers.Accept = Pd.Helpers.RequestHeader.getAcceptHeader()
+ return Backbone.Collection.prototype.sync.call(this, method, model, options)
+
+
window.AppRoutes = {}
mixIn ParseUrlParams, class window.Pd.Router extends Backbone.Router
window.AppRoutes = {}
mixIn ParseUrlParams, class window.Pd.Router extends Backbone.Router
8 app/assets/javascripts/shared/helpers/request_header.coffee
View
@@ -0,0 +1,8 @@
+window.Pd ||= {}
+window.Pd.Helpers ||= {}
+window.Pd.Helpers.RequestHeader =
+ getAcceptHeader: ->
+ 'application/vnd.pagerduty+json;version=1.1a'
+
+ setApiAcceptHeader: (request) ->
+ request.setRequestHeader('Accept', Pd.Helpers.RequestHeader.getAcceptHeader())
2 app/assets/javascripts/users/models/users.coffee
View
@@ -42,11 +42,13 @@ class window.User extends Pd.Model
$.ajax
type: 'PUT'
url: "/api/v1/teams/#{team.id}/users/#{@id}"
+ beforeSend: Pd.Helpers.RequestHeader.setApiAcceptHeader

pagerduty

Automate



pagerduty

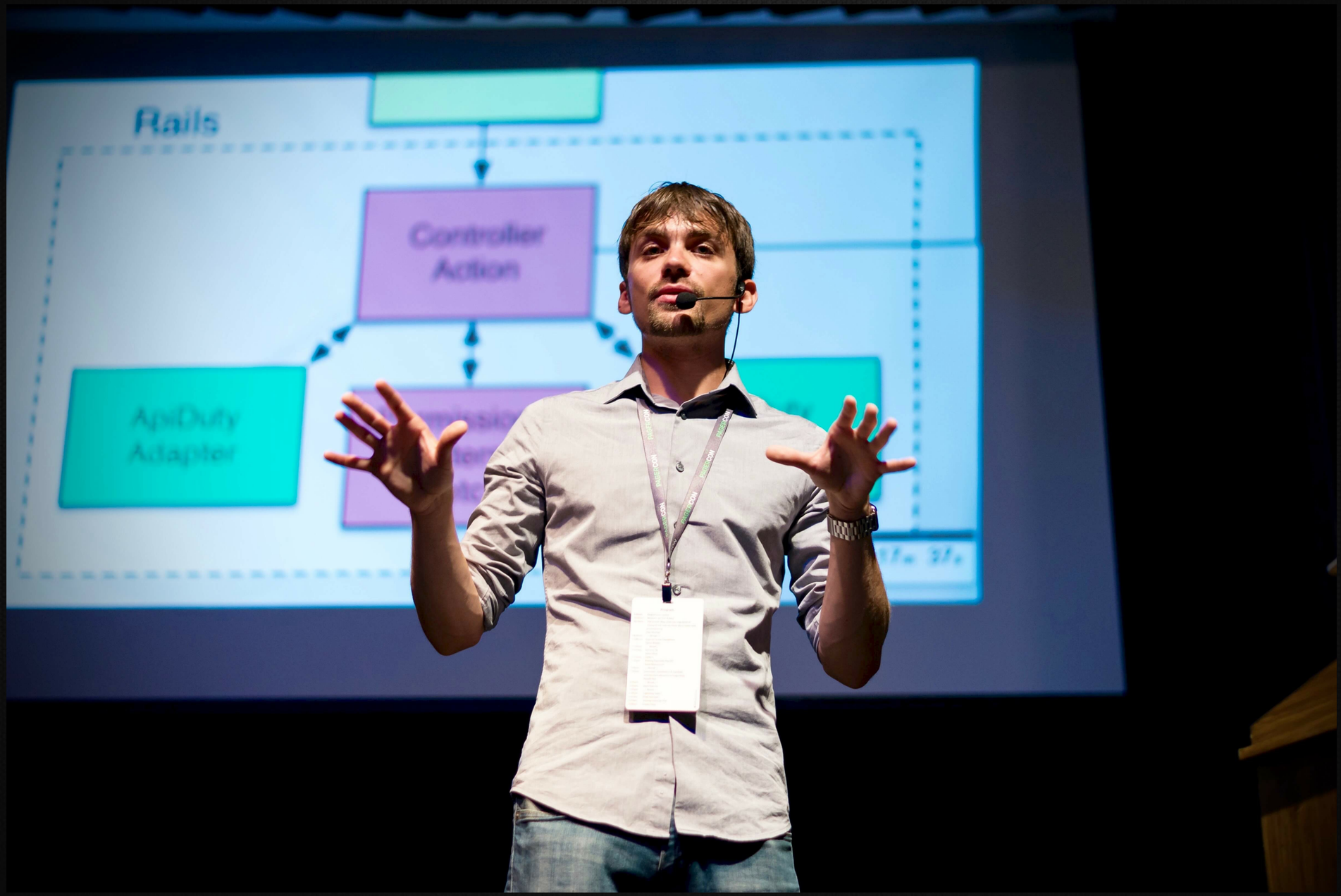
Trust, but Verify

```ruby
def self.compare_json(old, new, consider_ordering = false)
  return true if old == new

  old, new = *[old, new].map(&JSON.method(:parse))

  unless consider_ordering
    # Recursively sort hash keys so the diff ignores ordering
    original = self.sort_recursively(original)
    modified = self.sort_recursively(modified)

    # Quick test, in case the sorted hashes are equal
    return true if original == modified
  end
  return false
end
```

pagerduty

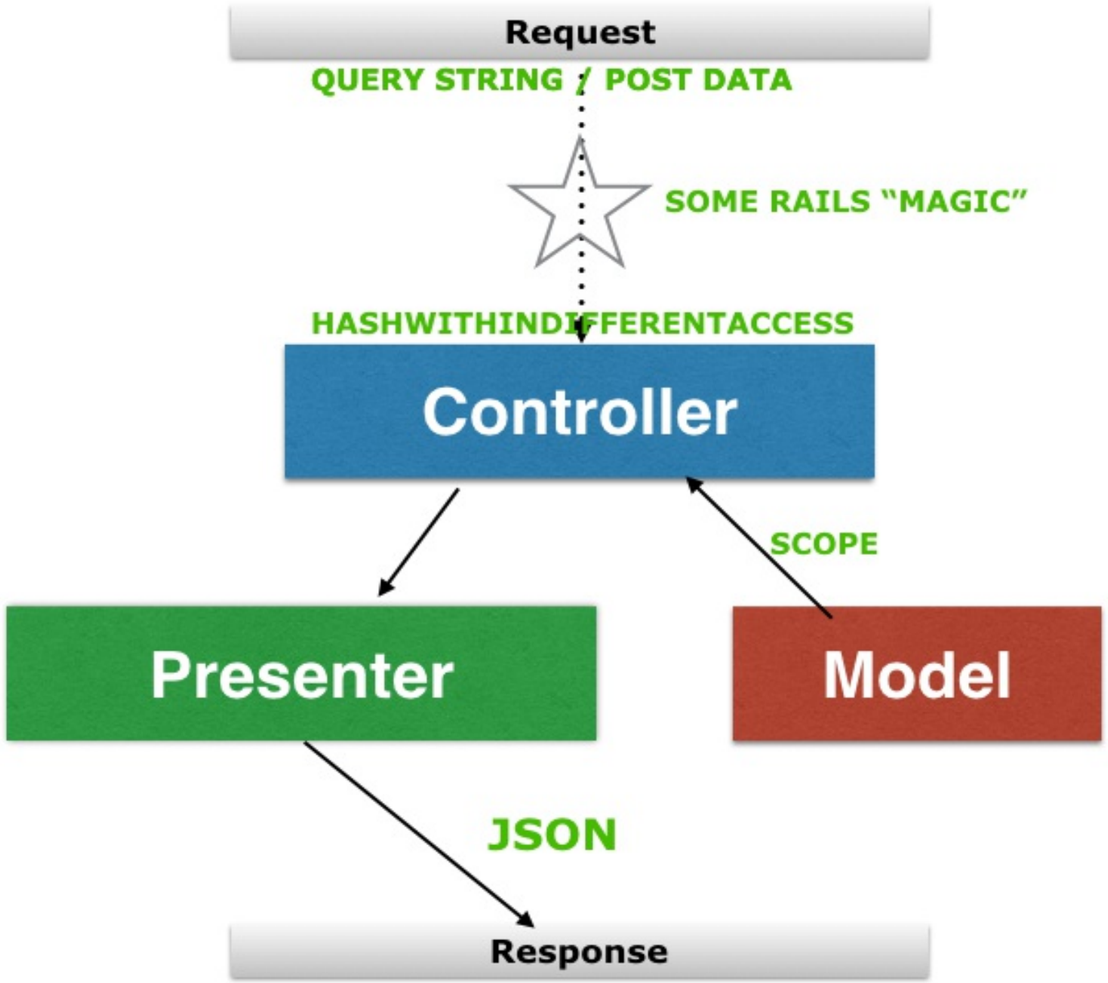
Connect

pagerduty



steverice / pagerduty-api-v2-deck

Why Adapt?


Request
QUERY STRING / POST DATA
SOME RAILS "MAGIC"
HASHWITHINDIFFERENTACCESS
Controller
SCOPE
Presenter
Model
JSON
Response

pagerduty

Hello,

Thanks for your continued interest and use of the PagerDuty API.

We're writing to let you know that as of November 4th, 2015, we will be discontinuing the ability to authenticate against the API using HTTP Basic Authentication (PagerDuty username and password), which has been deprecated since mid-2014.

We are doing this to improve security of PagerDuty accounts by limiting the spread of PagerDuty user passwords, which are difficult to audit and revoke should they become compromised.

Additionally, the need to authenticate a user via HTTP Basic Authentication imposes a performance penalty on every API request due to the robust hashing techniques we use to store and validate user passwords. This is done by design to make brute forcing passwords computationally infeasbile.

Switching away from basic authentication will improve the security of your users' passwords and make your API requests faster.

We've identified your PagerDuty account as one that is likely using HTTP Basic Authentication for automated HTTP requests against our API. The following PagerDuty users on your account are making recurring Basic Authentication requests: ${USER_NAME} ${USER_EMAIL}

If you don't recognize any of these users as being used with a PagerDuty integration or don't believe you are using HTTP Basic Authentication, please let us know and we can investigate further to help identify your integration.

If you are using a third-party client, tool, or library to connect to PagerDuty, check with the vendor/maintainer for an updated version that supports API Token Authentication. If you maintain such an integration internally, your development team can refer to our authentication documentation at https://developer.pagerduty.com/documentation/rest/authentication to learn more about using API Tokens.

Should you have any questions, don't hesitate to reach out. We recommend making the necessary changes well in advance of the November 4th deadline to provide ample time for testing and prevent any interruption in service.

pagerduty

Instead, we've focused on refining, standardizing, and simplifying the patterns that appear across our API. We've gotten a lot of great feedback over the years and are addressing a lot of the challenges developers have faced. And we're providing developers with better tools to build on and interact with the API, starting with a brand new developer portal and the ability to play with API v2 right from a web browser.

Here are just a few of the highlights:

- We've simplified our API URLs. Instead of needing to know an account's subdomain, just point all your requests to `api.pagerduty.com` and we'll figure out what account you wanted from the authentication. No more `/api/v1` root path either — it's as simple as `/incidents` or `/schedules`!
- We've standardized all of our resource types so that a user is always a user and a service is always a service, no matter where they appear in the API.
- We're launching a new `/oncalls` API that makes accessing information about who is on call, when, and for what more straightforward than it's ever been.
- We're adhering even more closely to REST architectural constraints. HTTP verbs work as you'd expect, and URIs are provided for every resource.
- We've given every resource a `summary` that describes what it is. It's your one-stop-shop for getting text to name and identify a resource.
- Across the board, we've taken steps to simplify or eliminate things in the v1 API that weren't strictly necessary, weren't being used, or caused developer confusion.

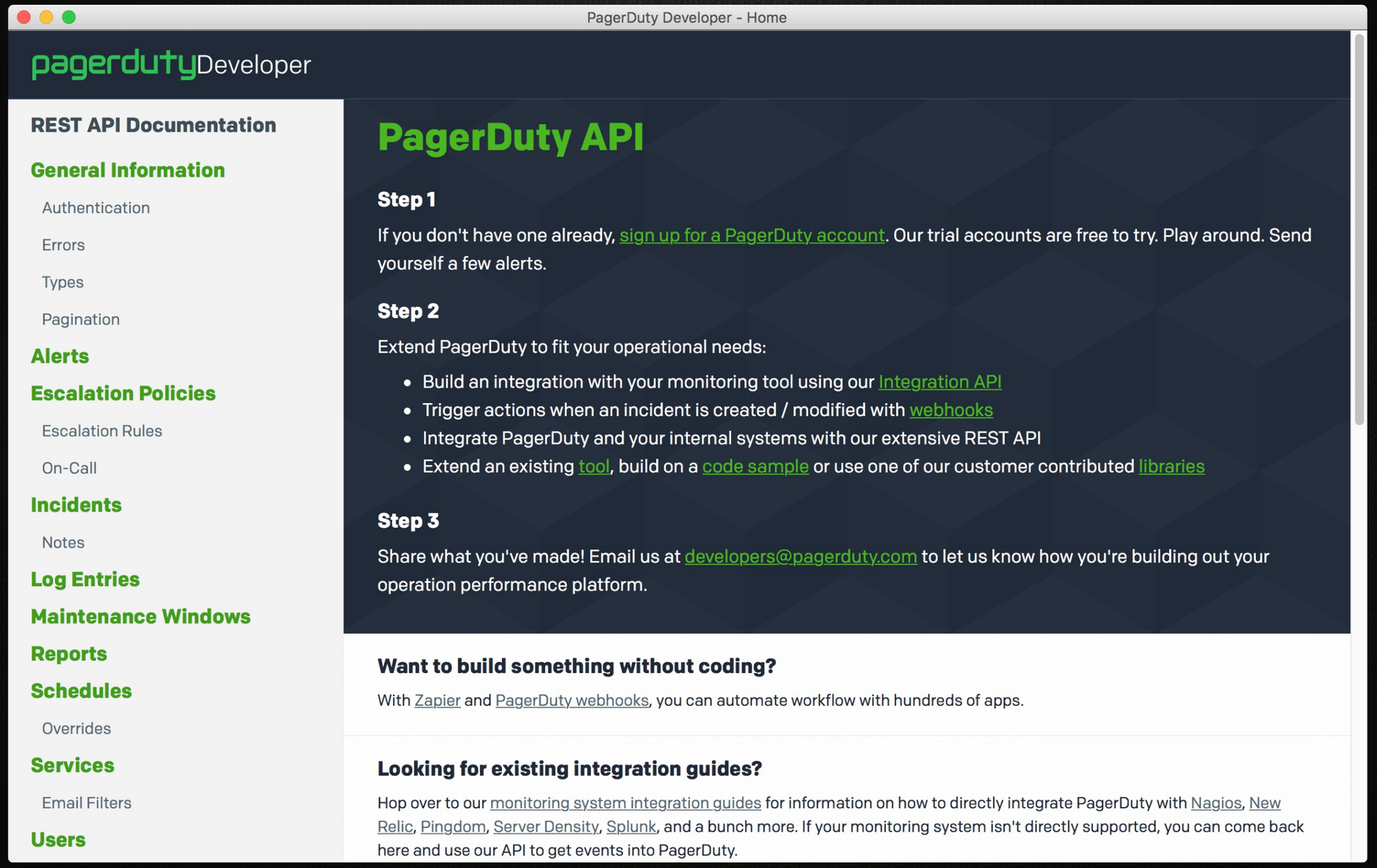
Our v1 to v2 migration guide has all the details about what sorts of things have changed, and our new reference documentation backed by a comprehensive Open API Initiative Specification contains complete specifics about all of our supported endpoints.

The PagerDuty Events API is versioned separately and is unaffected by these changes. All of your PagerDuty integrations will be able to continue sending events to PagerDuty as they always have.

pagerduty

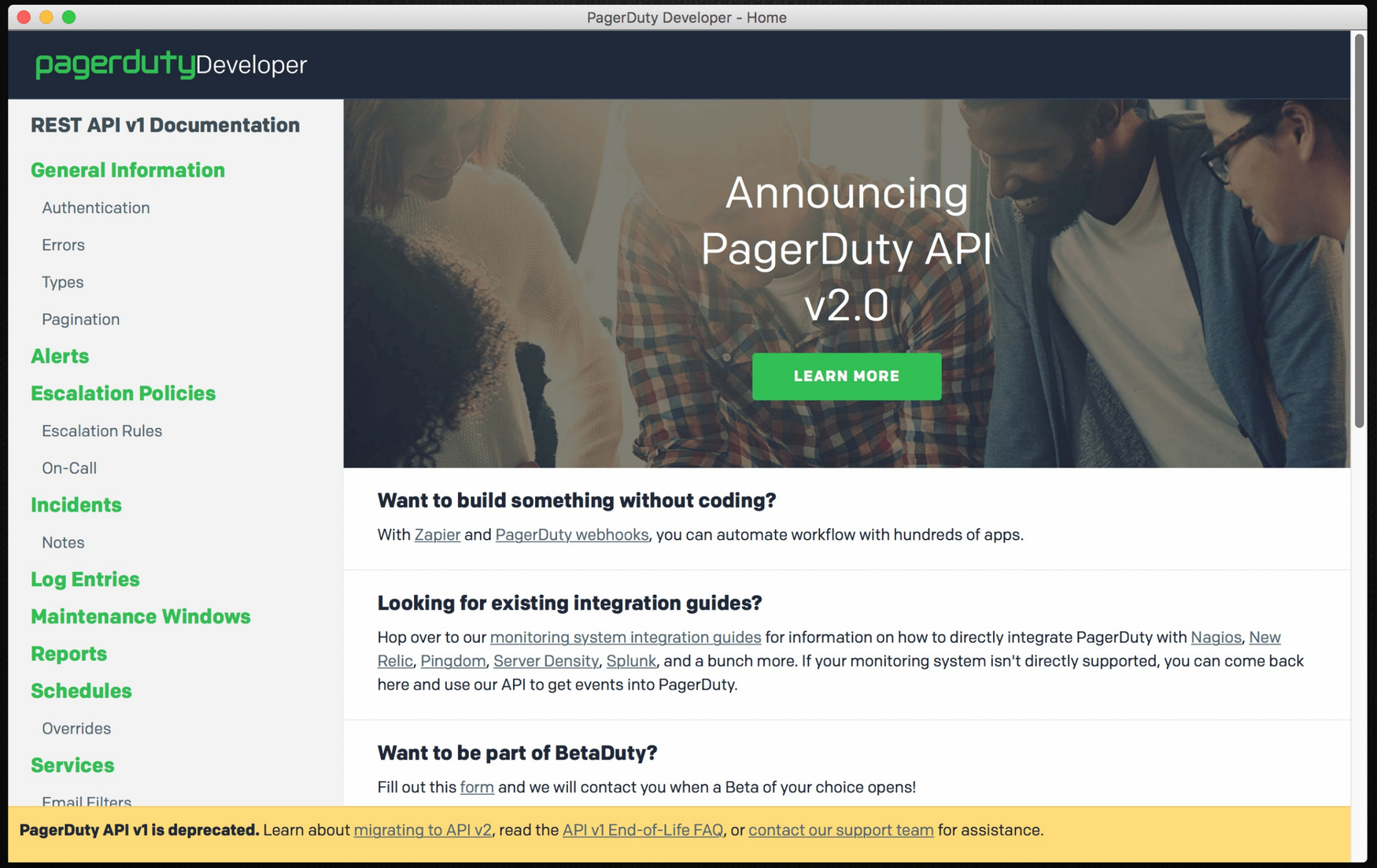
Market

pagerduty



steverice / pagerduty-api-v2-deck

pagerduty



https://v1.developer.pagerduty.com/

steverice / pagerduty-api-v2-deck

pagerduty

Demonstrate

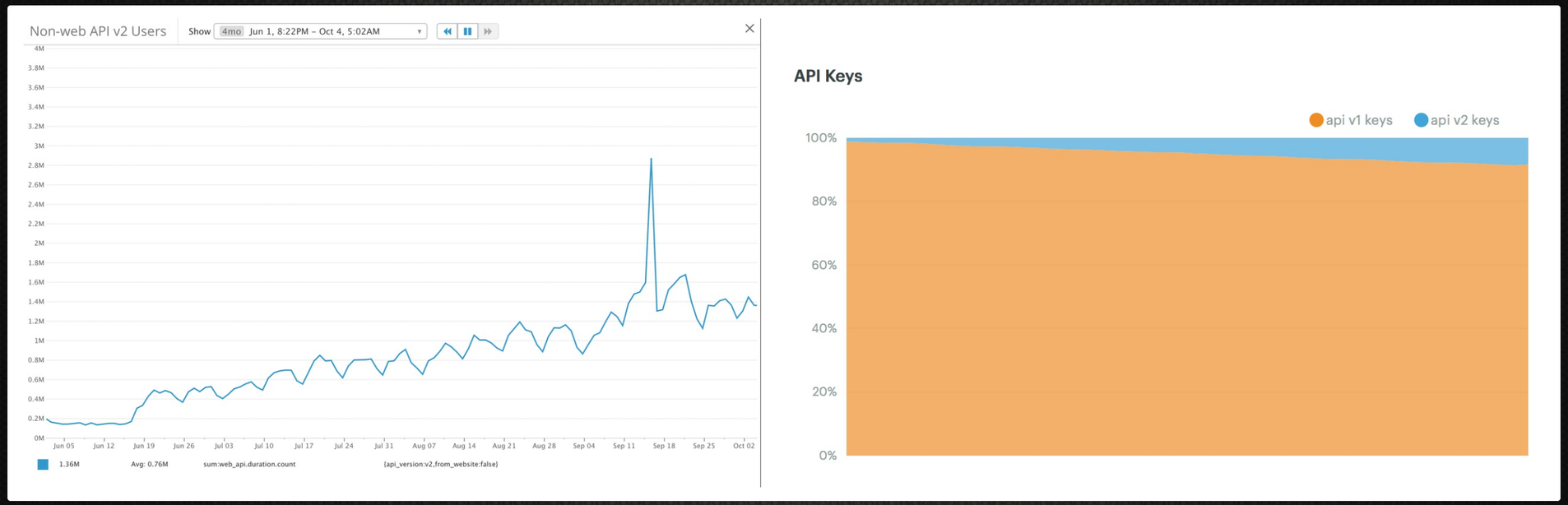
webinars

sales demos

live demos

customer success

pagerduty



steverice / pagerduty-api-v2-deck

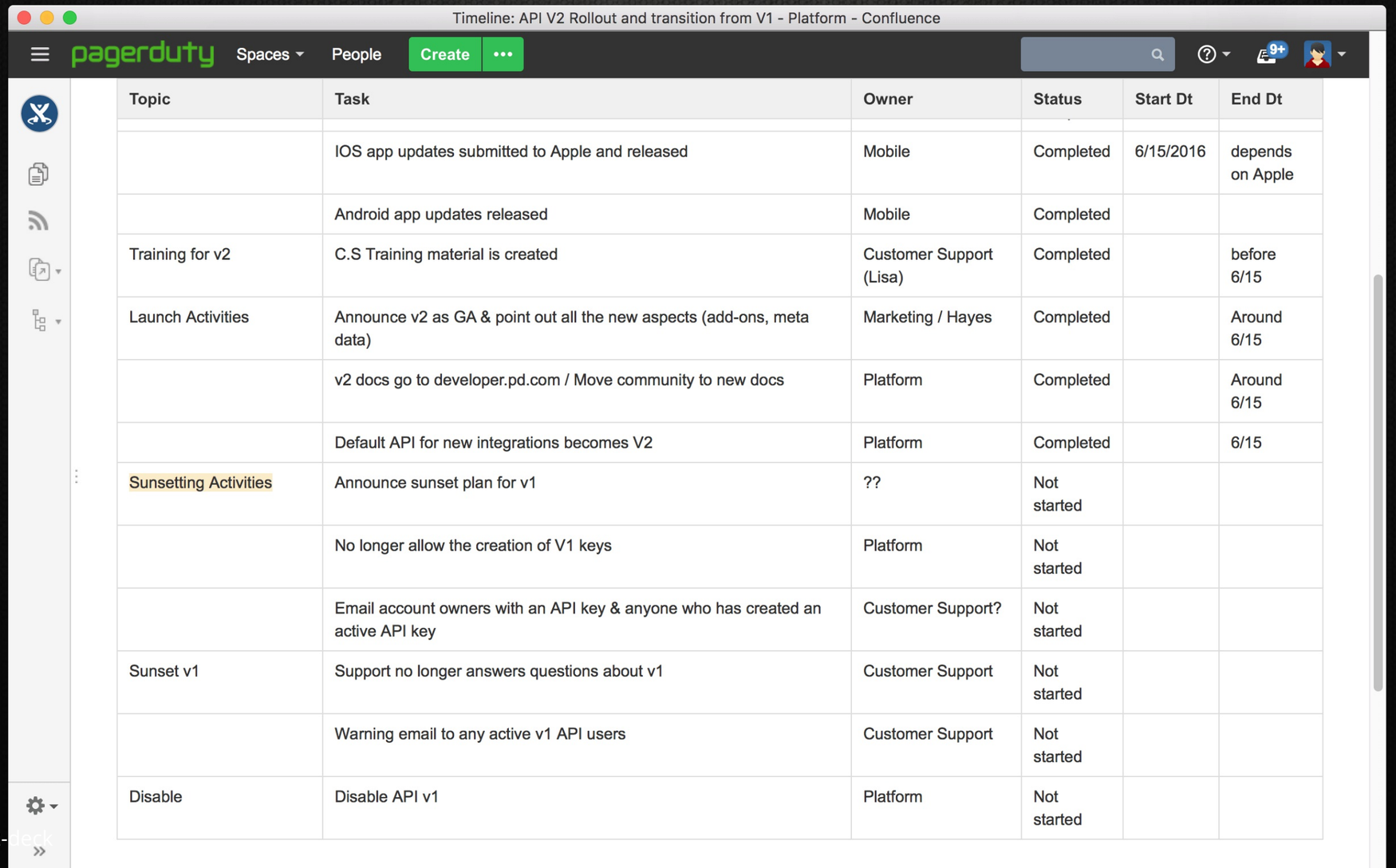
pagerduty

What's next?

steverice / pagerduty-api-v2-deck

pagerduty

Deprecation



Topic	Task	Owner	Status	Start Dt	End Dt
	IOS app updates submitted to Apple and released	Mobile	Completed	6/15/2016	depends on Apple
	Android app updates released	Mobile	Completed		
Training for v2	C.S Training material is created	Customer Support (Lisa)	Completed		before 6/15
Launch Activities	Announce v2 as GA & point out all the new aspects (add-ons, meta data)	Marketing / Hayes	Completed		Around 6/15
	v2 docs go to developer.pd.com / Move community to new docs	Platform	Completed		Around 6/15
	Default API for new integrations becomes V2	Platform	Completed		6/15
Sunsetting Activities	Announce sunset plan for v1	??	Not started		
	No longer allow the creation of V1 keys	Platform	Not started		
	Email account owners with an API key & anyone who has created an active API key	Customer Support?	Not started		
Sunset v1	Support no longer answers questions about v1	Customer Support	Not started		
	Warning email to any active v1 API users	Customer Support	Not started		
Disable	Disable API v1	Platform	Not started		

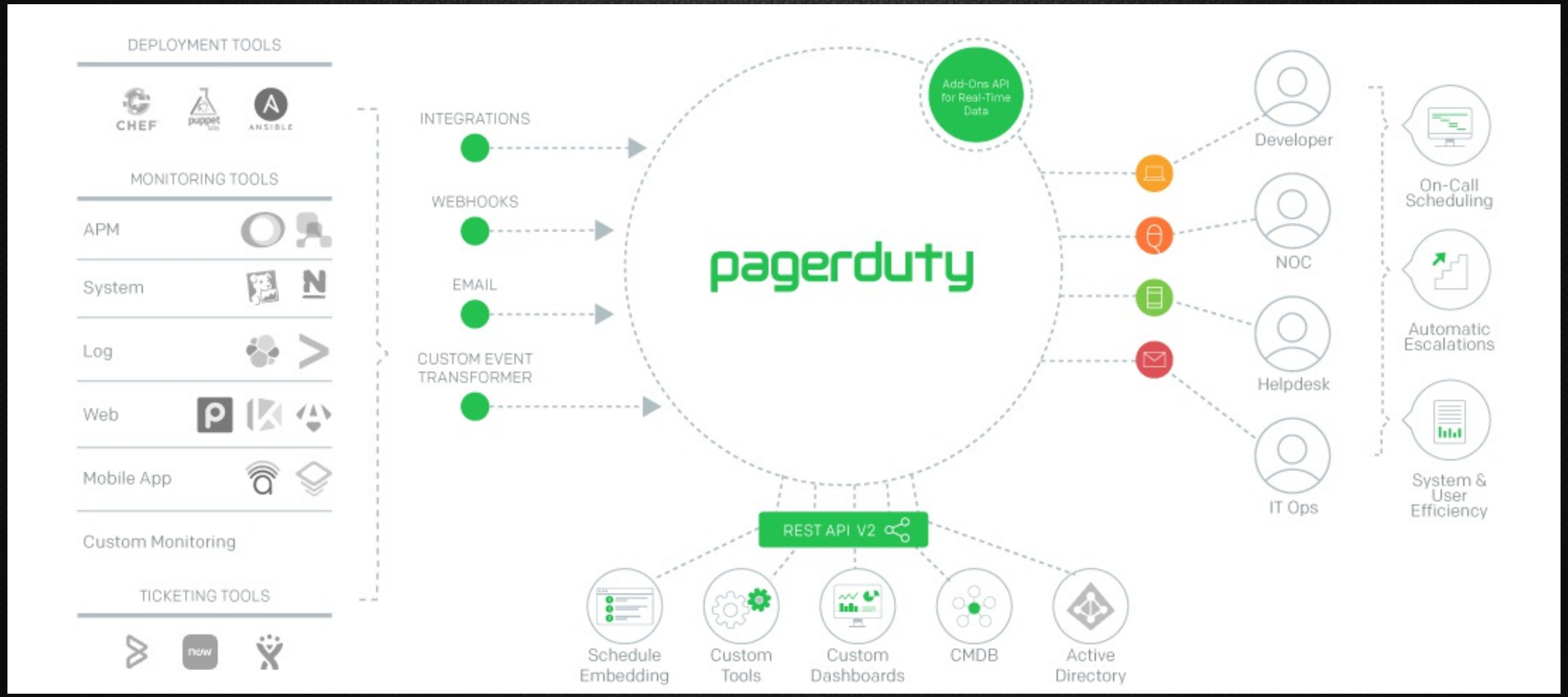
pagerduty

BFF

steverice / pagerduty-api-v2-deck

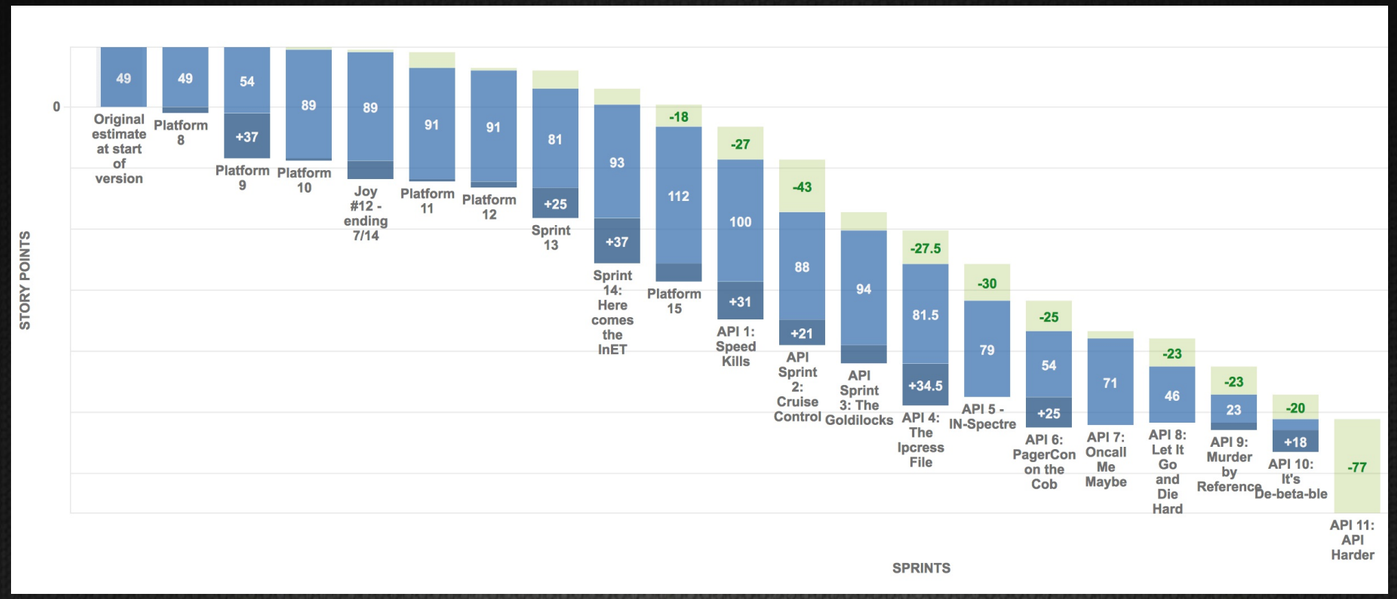
pagerduty

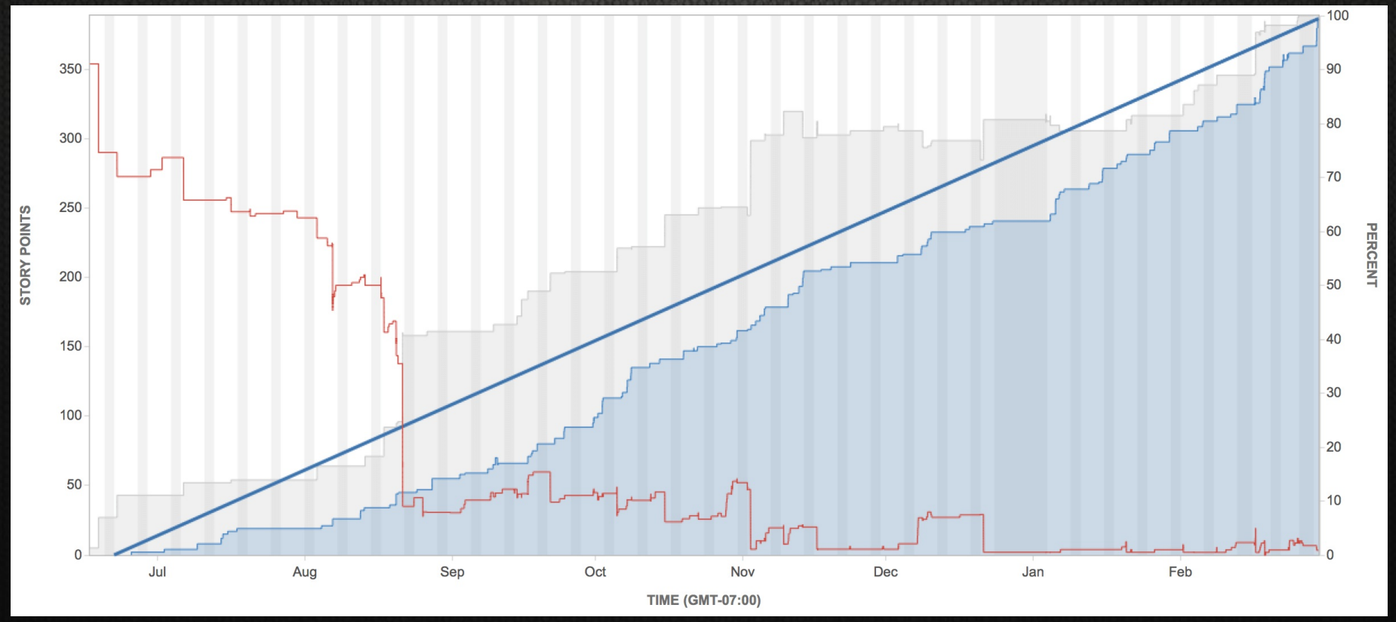
The Platform Strategy



https://www.pagerduty.com/features/platform-extensibility/

pagerduty


STORY POINTS
0
49
Original estimate at start of version
49
Platform 8
54
+37
Platform 9
89
Platform 10
89
Joy #12 - ending 7/14
91
Platform 11
91
Platform 12
81
+25
Sprint 13
93
+37
Sprint 14: Here comes the InET
-18
112
Platform 15
-27
100
+31
API 1: Speed Kills
-43
88
+21
API Sprint 2: Cruise Control
94
API Sprint 3: The Goldilocks
-27.5
81.5
+34.5
API 4: The Ipcress File
-30
79
API 5 - IN-Spectre
-25
54
+25
API 6: PagerCon on the Cob
71
API 7: Oncall Me Maybe
-23
46
API 8: Let It Go and Die Hard
-23
23
API 9: Murder by Reference
-20
+18
API 10: It's De-beta-ble
-77
API 11: API Harder
SPRINTS


STORY POINTS
350
300
250
200
150
100
50
0
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
TIME (GMT-07:00)
PERCENT
100
90
80
70
60
50
40
30
20
10
0

pagerduty

It takes a village


pagerduty
Fin?
developer.pagerduty.com
steverice / pagerduty-api-v2-deck

pagerduty

Appendix

pagerduty

Image Credits

- "Steve Rice" By Steve Rice © 2016 (Own work), all rights reserved
- "Stand on the Shoulders of Giants" By Samir Eberlin [CC0], via Wikimedia Commons — https://commons.wikimedia.org/wiki/File:Turtles.on.a.stone.in.brazil.jpg
- "Make Decisions." By SeppVei (Own work) [CC0], via Wikimedia Commons — https://commons.wikimedia.org/wiki/File:Bicycle_shed.JPG
- "Be Pragmatic" By Liveon001 © Travis K. Witt (Own work) [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0) or GFDL (http://www.gnu.org/copyleft/fdl.html)], via Wikimedia Commons — https://commons.wikimedia.org/wiki/File:Farfalle_Pasta.JPG
- "Dogfood. Everything." Original source not found. Used without permission.
- "Keep it maintainable" By David Jones from Isle of Wight, United Kingdom [CC BY 2.0], via Wikimedia Commons — https://commons.wikimedia.org/wiki/File:Mike_O%27Callaghan-Pat_Tillman_Bridge_construction.jpg
- "Pour over logs" by Timothy Marsee [CC BY 2.0], via Flickr — https://www.flickr.com/photos/tmarsee530/9899684046
- "BFF" By Guillaume Paumier (Own work) [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons — https://commons.wikimedia.org/wiki/File:Best_Frends_Forever_-_Golden_Gate_bridge_guard_rail_166.jpg
- "Fin?" Copyright 1984 Warner Bros. Pictures, via Cinema Autopsy — https://blog.cinemaautopsy.com/2013/04/02/film-review-the-neverending-story-1984/