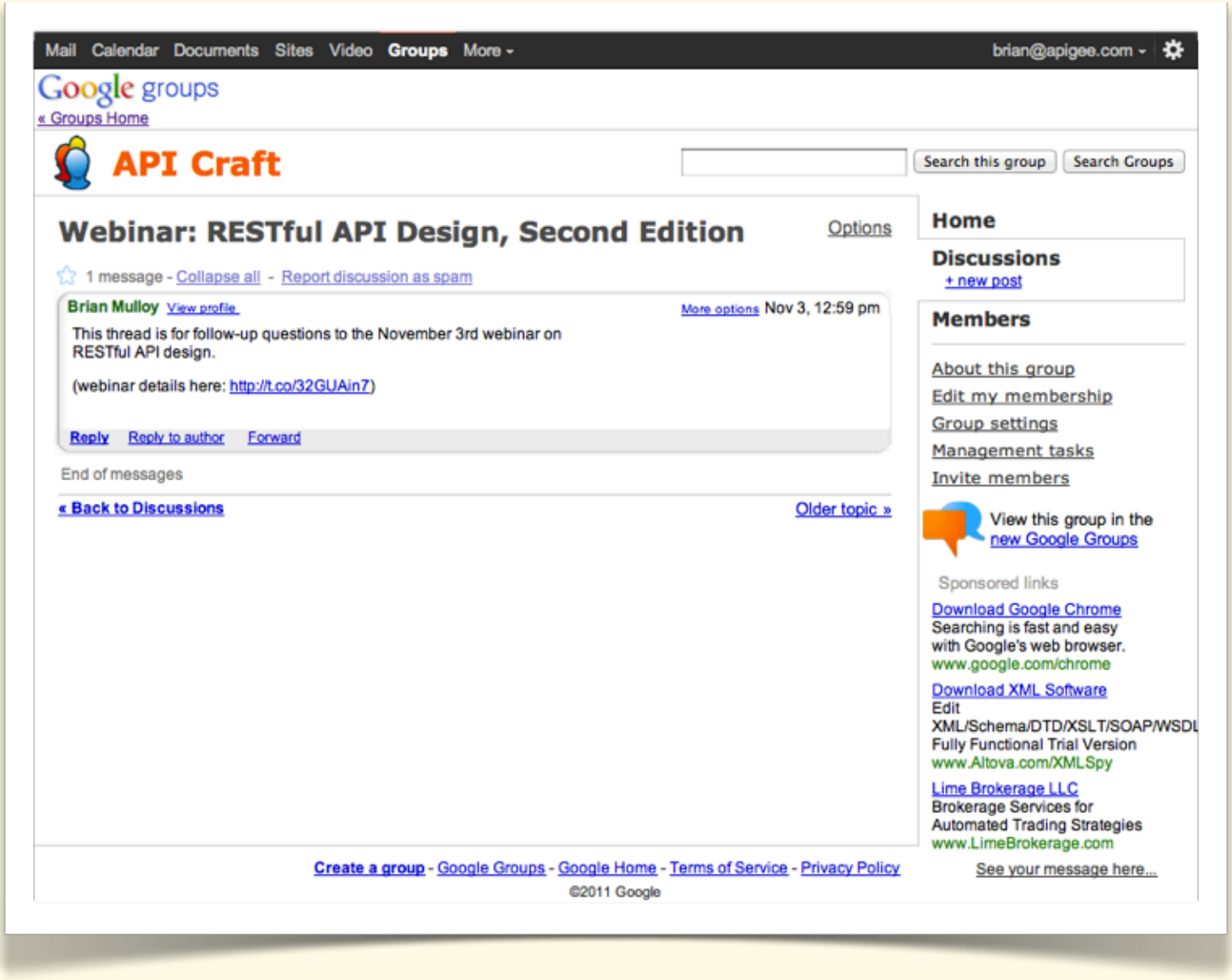
RESTful API Design
Second Edition

Brian Mulloy
@landlessness

Apigee
@apigee

11.03.11 @ 11:03:11 PST
Dial-in or VoIP
See Details in Chat Window

groups.google.com/group/api-craft



youtube.com/apigee

To be a RESTafarian or a Pragmatist?

MikeSchinkel.com

Jack of all Trades — Serial Master of One

home blog portfolio projects consulting about login

What is a RESTafarian?

Nov 12th, 2006 | Programming, Web ShareThis

A **RESTifarian** is a zealous proponent of the REST *software architectural style* as defined by Roy T. Fielding in Chapter 5 of his PhD. dissertation at UCIrvine. You can find RESTifarians in the wild on the REST-discuss mailing list. But be careful, RESTifarians can be extremely meticulous when discussing the finer points of REST, as I learned recently while participating on the list. :)

Tags: *people, rest, royfielding, webservices*

people, rest, royfielding, webservices

Search

To search, type and hit enter

Me

ADD THIS BLOG TO MY Technorati FAVORITES

Categories

- Atlanta
- Contrarianism

App User

App Store

App

App Developer

World of APIs

API

API Team

Internal Systems

Application developers are *raison d'être* for APIs.

Be pragmatic.

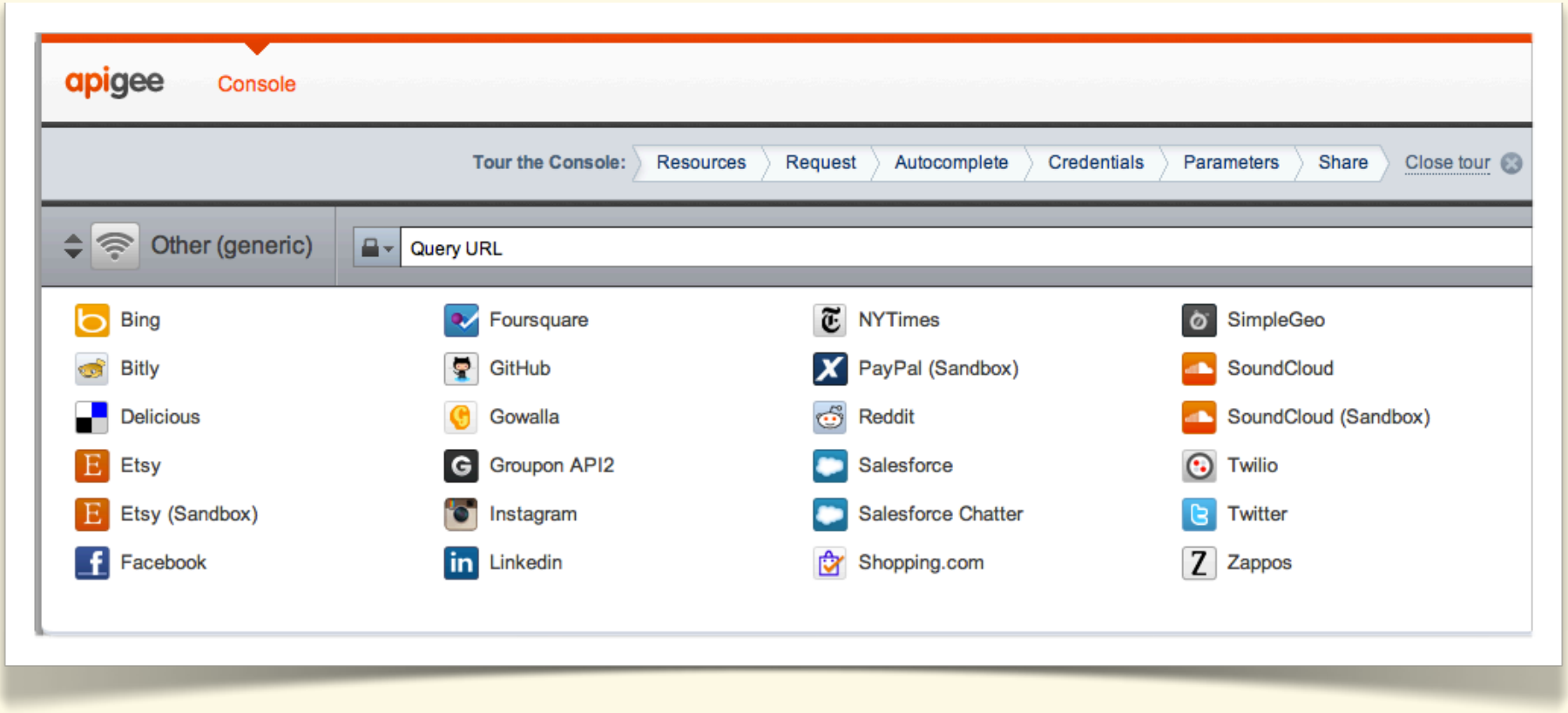
For the benefit of application developers.

Pragmatic RESTful APIs are a design problem.

baddesigns.com


PUSH
PUSH

Paul Mijksenaar Design for Function Award 2011


apigee
Console
Tour the Console:
Resources
Request
Autocomplete
Credentials
Parameters
Share
Close tour
Other (generic)
Query URL
Bing
Bitly
Delicious
Etsy
Etsy (Sandbox)
Facebook
Foursquare
GitHub
Gowalla
Groupon API2
Instagram
Linkedin
NYTimes
PayPal (Sandbox)
Reddit
Salesforce
Salesforce Chatter
Shopping.com
SimpleGeo
SoundCloud
SoundCloud (Sandbox)
Twilio
Twitter
Zappos

Let’s look at puppies.

hackett

hackett

hackett

hackett

```
...
/getAllDogs
/locationVerify
/foodNeeded
/createRecurringDogWalk
/giveDirectOrder
/healthCheck
/getRecurringDogWalkSchedule
/getLocation
/getDog
/massDogParty
/getNewDogsSince
/getRedDogs
/getSittingDogs
/dogStateChangesSearch
/replaceSittingDogsWithRunningDogs
/saveDog
...
```

API	NVP	SOAP
AddressVerify	NVP	SOAP
BillOutstandingAmount	NVP	SOAP
Callback	NVP	
CreateRecurringPaymentsProfile	NVP	SOAP
DoAuthorization	NVP	SOAP
DoCapture	NVP	SOAP
DoDirectPayment	NVP	SOAP
DoExpressCheckoutPayment	NVP	SOAP
DoNonReferencedCredit	NVP	SOAP
DoReauthorization	NVP	SOAP
DoReferenceTransaction	NVP	SOAP
DoVoid	NVP	SOAP
GetBalance	NVP	SOAP
GetBillingAgreementCustomerDetails	NVP	SOAP
GetExpressCheckoutDetails	NVP	SOAP
GetRecurringPaymentsProfileDetails	NVP	SOAP
GetTransactionDetails	NVP	SOAP
ManageRecurringPaymentsProfileStatus	NVP	SOAP
ManagePendingTransactionStatus	NVP	SOAP
MassPayment	NVP	SOAP
RefundTransaction	NVP	SOAP
SetCustomerBillingAgreement	NVP	SOAP
SetExpressCheckout	NVP	SOAP
TransactionSearch	NVP	SOAP
UpdateRecurringPaymentsProfile	NVP	SOAP

A puppy’s world is big.

JnL

law_keven

Smithsonian's National Zoo

hackett

```
...
/getAllDogs
/verifyLocation
/feedNeeded
/createRecurringWakeUp
/giveDirectOrder
/checkHealth
/getRecurringWakeUpSchedule
/getLocation
/getDog
/newDog
/getNewDogsSince
/getRedDogs
/getSittingDogs
/setDogStateTo
/replaceSittingDogsWithRunningDogs
/saveDog
...
```

```
...
/getAllLeashedDogs
/verifyVeterinarianLocation
/feedNeededFood
/createRecurringMedication
/doDirectOwnerDiscipline
/doExpressCheckupWithVeterinarian
/getRecurringFeedingSchedule
/getHungerLevel
/getSquirrelsChasingPuppies
/newDogForOwner
/getNewDogsAtKennelSince
/getRedDogsWithoutSiblings
/getSittingDogsAtPark
/setLeashedDogStateTo
/replaceParkSittingDogsWithRunningDogs
/saveMommaDogsPuppies
...
```

We are on a slippery slope.

Keep the simple things simple.

Hopkinsii

We only need two base URLs per resource.

The first is for a collection.

`/dogs`

The second is for an element.

/dogs/**1234**

POST
GET
PUT
DELETE

CREATE

READ

UPDATE

DELETE

Resource	**POST** create	**GET** read	**PUT** update	**DELETE** delete
/dogs	create a new dog	list dogs	replace dogs with new dogs	delete all dogs
/dogs/1234	treat as a collection create new dog in it	show Bo	if exists update Bo if not create Bo	delete Bo

Wikipedia

Resource	**POST** create	**GET** read	**PUT** update	**DELETE** delete
/dogs	create a new dog	list dogs	~~**replace dogs with new dogs**~~	delete all dogs
/dogs/1234	~~**treat as a collection create new dog in it**~~	show Bo	if exists update Bo ~~**if not create Bo**~~	delete Bo

Wikipedia

Resource	**POST** create	**GET** read	**PUT** update	**DELETE** delete
/dogs	create a new dog	list dogs	**bulk update dogs**	delete all dogs
/dogs/1234	**error**	show Bo	if exists update Bo **if not error**	delete Bo

Wikipedia

Verbs are bad.

Nouns are good.

Plurals or singulars?

Foursquare

```
/checkins
```

GroupOn

```
/deals
```

Zappos

```
/Product
```

Plurals are better.

```
/dogs
```

Abstract or concrete naming?

Super High

```
/things
```

High

```
/animals
```

Medium

```
/dogs
```

Low

```
/beagles
```

Concrete is better than abstract.

```
/dogs
```

What about associations?

```
GET /owners/5678/dogs
```

```
POST /owners/5678/dogs
```

What about complex variations?

Cody Simms

Sweep variations under the ‘?’

```
/dogs?color=red&state=running&location=park
```

Hopkinsii

/dogs

What about errors?

meredithfarmer

Facebook

HTTP Status Code: **200**

{"**type**":"OAuthException","**message**":"(**#803**) Some of the aliases you requested do not exist: foo.bar"}

Twilio

HTTP Status Code: **401**

{"**status**":401,"**message**":"Authenticate","**code**":20003,"**more_info**":"http://www.twilio.com/docs/errors/20003"}

SimpleGeo

HTTP Status Code: **401**

{"**code**":401,"**message**":"Authentication Required"}

Code for code

```
200 - OK
```

```
401 - Unauthorized
```

http://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Message for people

```
{"message" : "Verbose, plain language
description of the problem with hints about
how to fix it."
"more_info" : "http://dev.teachdogrest.com/
errors/12345"}
```

What about versioning?

Twilio

```
/2010-04-01/Accounts/
```

salesforce.com

```
/services/data/v20.0/sobjects/Account
```

Facebook

```
?v=1.0
```

`/v1/dogs`

Please give me exactly what I need.

LinkedIn

```
/people:(id,first-name,last-name,industry)
```

Facebook

```
/joe.smith/friends?fields=id,name,picture
```

Google (partial response)

```
?fields=title,media:group(media:thumbnail)
```

```
/dogs?fields=name,color,location
```

What about pagination?

Facebook

```
offset
limit
```

Twitter

```
page
rpp
```

LinkedIn

```
start
count
```

```
/dogs?limit=25&offset=50
```

What about formats?

Google Data

```
?alt=json
```

Foursquare

```
/venue.json
```

Digg*

```
Accept: application/json
```

```
?type=json
```

* The type argument, if present, overrides the Accept header.

`/dogs.json`

`/dogs/1234.json`

What about defaults?

Format

```
json
```

Pagination (depends on data size)

```
limit=10&offset=0
```

What about attribute names?

Twitter

```
"created_at": "Thu Nov 03 05:19:38 +0000 2011"
```

Bing

```
"DateTime": "2011-10-29T09:35:00Z"
```

Foursquare

```
"createdAt": 1320296464
```

JSON is for JavaScript Objects

```
var myObject = JSON.parse(response);
```

These looks funny in JavaScript

```
timing = myObject.created_at;
```

```
timing = myObject.DateTime;
```

JavaScript Convention

```
"createdAt": 1320296464
```

```
timing = myObject.createdAt;
```

Medial Capitalization aka CamelCase

What about non-resource-y stuff?

Calculate

Translate

Convert

Use verbs **not** nouns

```
/convert?from=EUR&to=CNY&amount=100
```

What about searching?

Global

```
/search?q=fluffy+fur
```

Scoped

```
/owners/5678/dogs/search?q=fluffy+fur
```

Formatted

```
/search.xml?q=fluffy+fur
```

What about counts?

/dogs/**count**

What about the rest of the URL?

Facebook	**graph**.facebook.com
	api.facebook.com
	developers.facebook.com
Foursquare	api.foursquare.com
	developers.foursquare.com
Twitter	api.twitter.com
	search.twitter.com
	stream.twitter.com
	dev.twitter.com

API gateway

api.teachdogrest.com

Developer connection

developers.teachdogrest.com

Do web redirects

api → developers (if from browser)

dev → developers

developer → developers

What about exceptional stuff?

Client intercepts HTTP error codes

Twitter

```
/public_timelines.json?
suppress_response_codes=true
```

```
HTTP Status Code: 200
```

```
{"error" : "Could not authenticate
you." }
```

Always returns **OK**

```
/dogs?suppress_response_codes=true
```

Code for ~~code~~ **ignoring**

```
200 - OK
```

Message for people **& code**

```
{"response_code" : "401", "message" :
"Verbose, plain language description of the
problem with hints about how to fix it."
"more_info" : "http://dev.teachdogrest.com/
errors/12345", "code" : 12345}
```

Client supports limited HTTP methods

Method Parameter

create

```
/dogs?method=post
```

read

```
/dogs
```

update

```
/dogs/1234?method=put&location=park
```

delete

```
/dogs/1234?method=delete
```

What about authentication?

PayPal

```
Permissions Service API
```

Facebook

```
OAuth 2.0
```

Twitter

```
OAuth 1.0a
```

Use latest and greatest OAuth

```
OAuth 2.0
```

Don’t do something close, but different.

How do application developers use the API?

What about chatty applications?

First be complete & RESTful.

Then provide shortcuts.

Partial response syntax can help.

```
/owners/5678?fields=name,dogs(name)
```

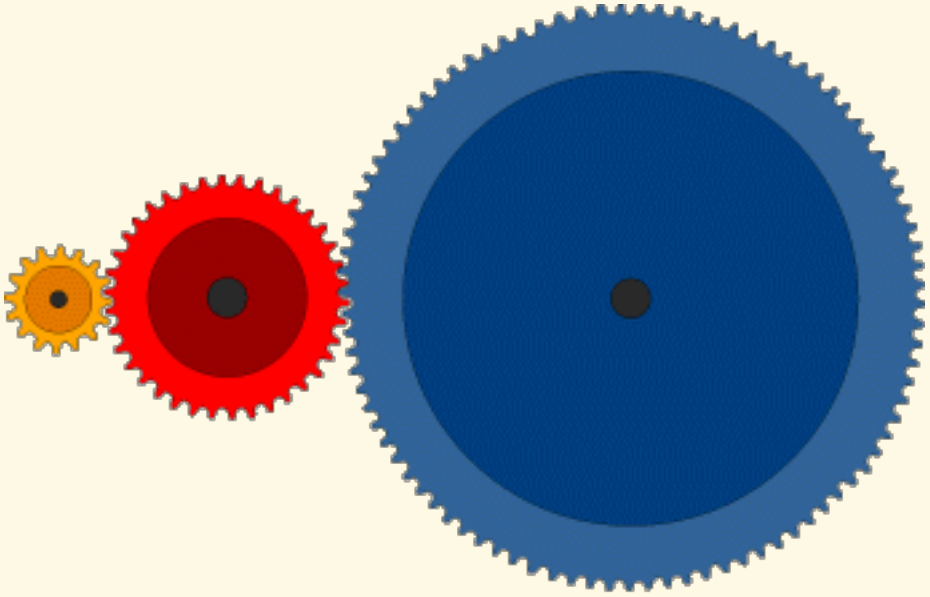
What about when building an UI requires a lot of domain knowledge?

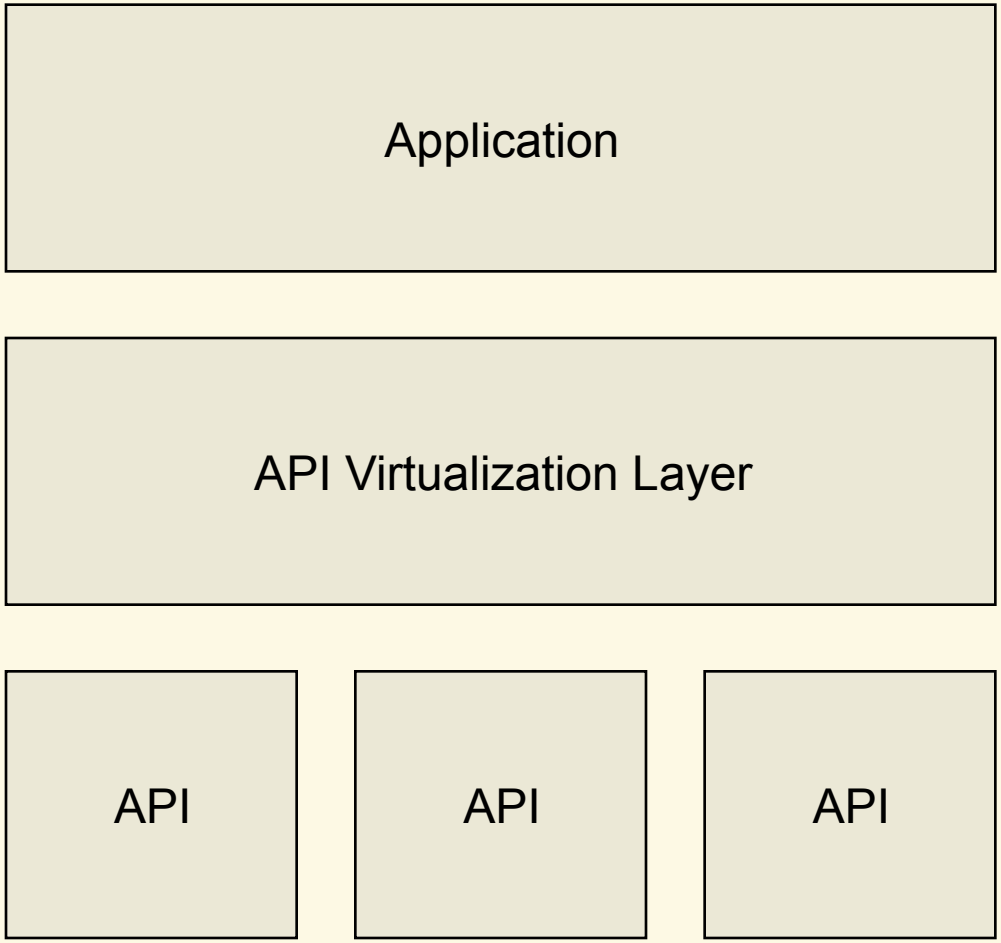

firesideint @cdellaverson Felt that, loud and clear. We're in Cap Haitien. House (really the whole world) shook for about 30 seconds.
From Haiti
Cap-Haitien
Grande Rivière du Nord
Pepillo Salcedo
Fort-Liberte
Dajabón
Ennery
Saint-
Lafond
Saint-Marc
Hinche
Verrettes
Dominican Republic
Haiti
Montrouis
Canal de Saint-Marc
La Cayenne
Anse-a-Galets
Mirebalais
Comendador
San Juan
Manneville
Croix-de-Bouquets
Port-au-Prince
Petionville
Leogane
Kenscoff
twitter

Complement your API with code libraries and SDK.

Really? All of this? And iterate it?


normal
cartilage


©2000 How Stuff Works


Application
API Virtualization Layer
API
API
API

Be RESTful
Only 2 URLs
No verbs
Use nouns as plurals
Concrete over abstract
For JSON follow JavaScript conventions
Sweep complexity behind the ‘?’
Borrow from leading APIs
Account for exceptional clients
Add virtualization layer

Questions?

THANK YOU

Subscribe to API webinars at:

youtube.com/apigee

THANK YOU

Questions and ideas to:

groups.google.com/group/api-craft

THANK YOU

Contact me at:
@landlessness
brian@apigee.com