API Design for ETL

Lessons from Nearly 100 Data Integrations in the Wild

Dan Mosora
Sources Software Engineer
Talend – Stitch Data Loader

github.com/dmosorast/api-design-for-etl

Backstory

Data is complicated

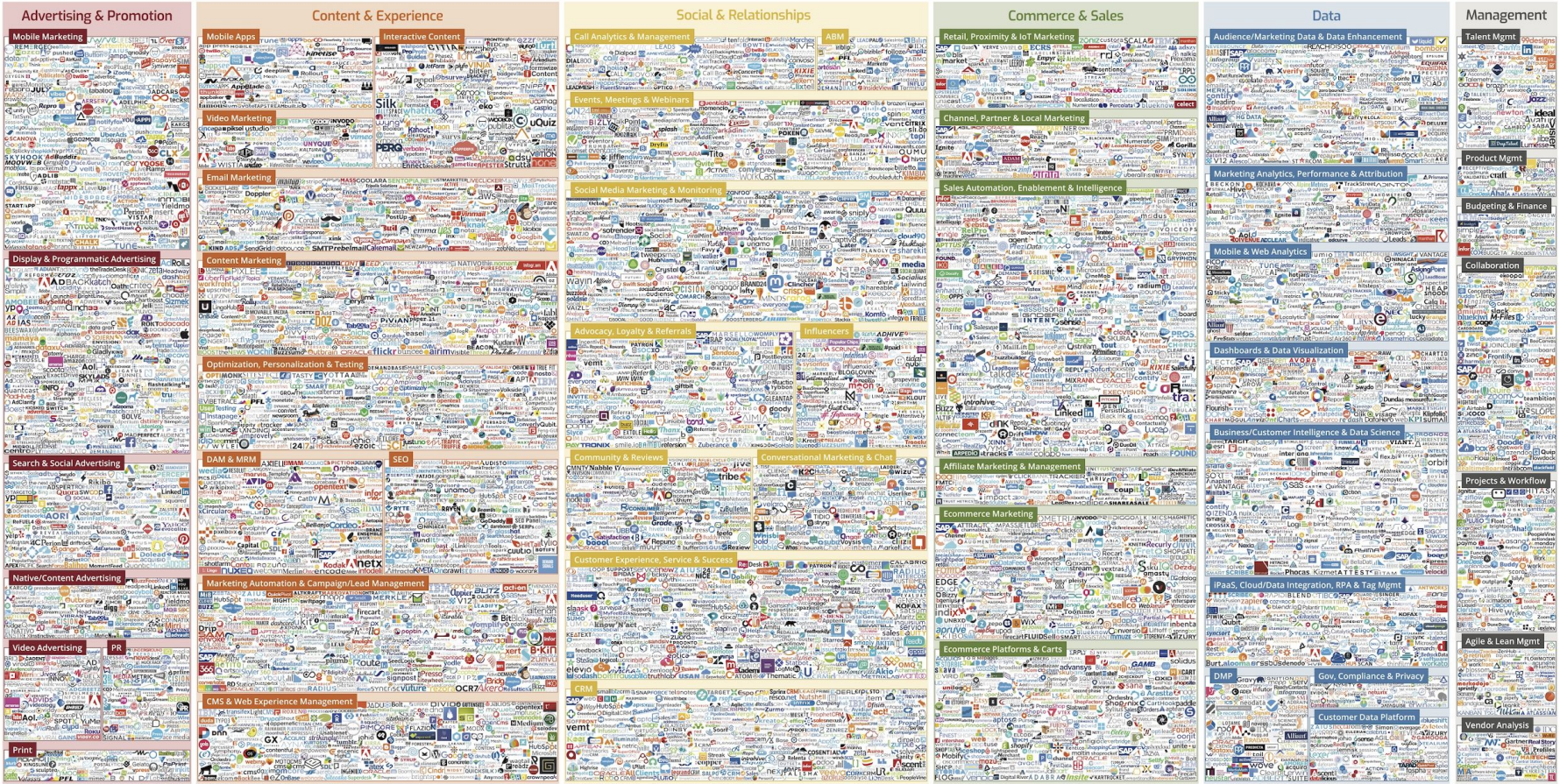
SINGER

Level Setting

E

chiefmartec.com Marketing Technology Landscape ("Martech 5000")

April 2019



Produced by Scott Brinker (@chiefmartec) and Blue Green Brands (@bluegreenbrands). blue green

What this talk is...

Lessons from real interactions with APIs

Inspiration to start a discussion about these things

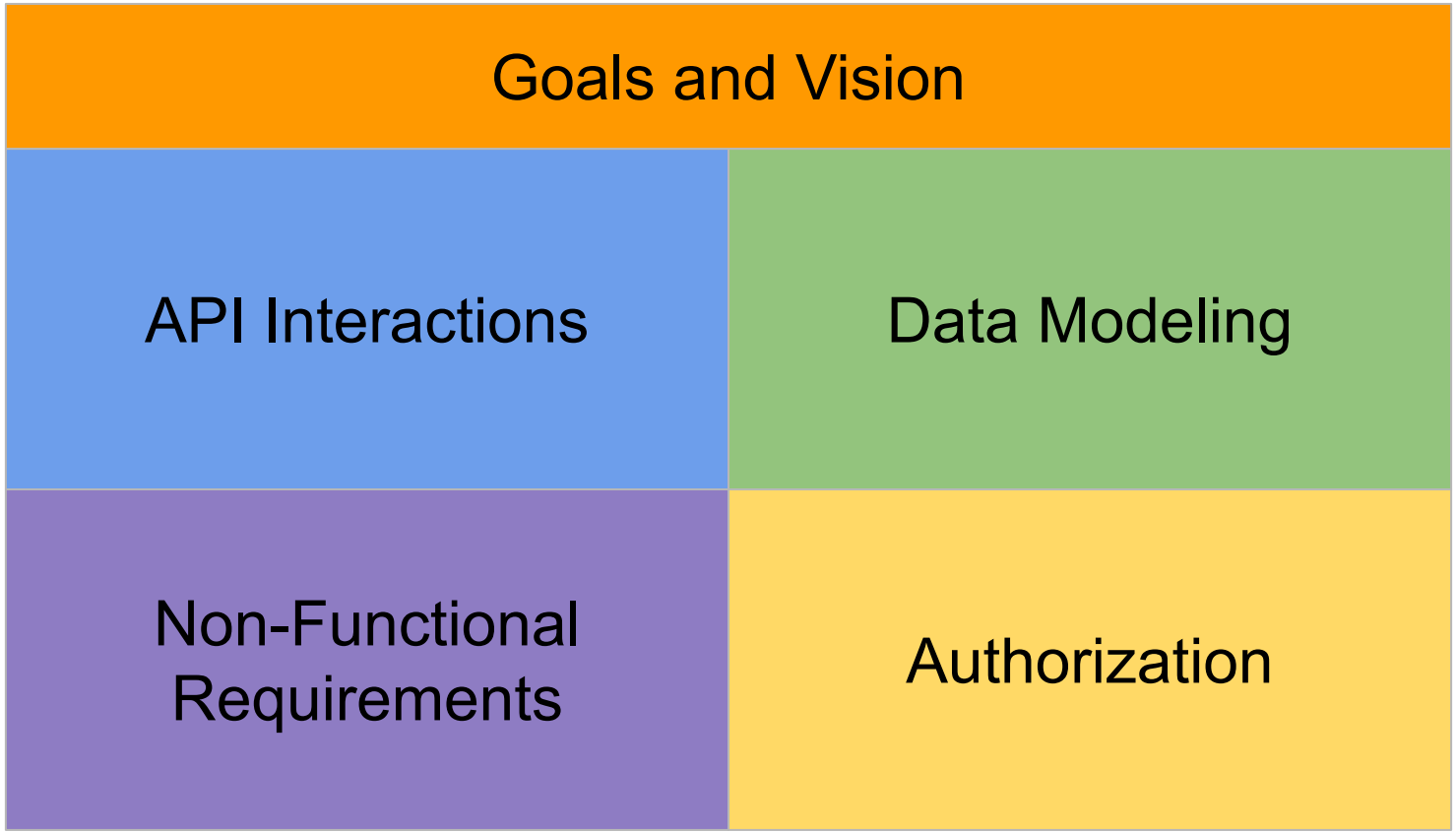
My black-box **perspective** as a USER of APIs

What this talk is **NOT**...

NOT criticism of APIs that have been hard to work with

NOT advice about the ***meaning*** of the data

NOT advice on specific implementation details

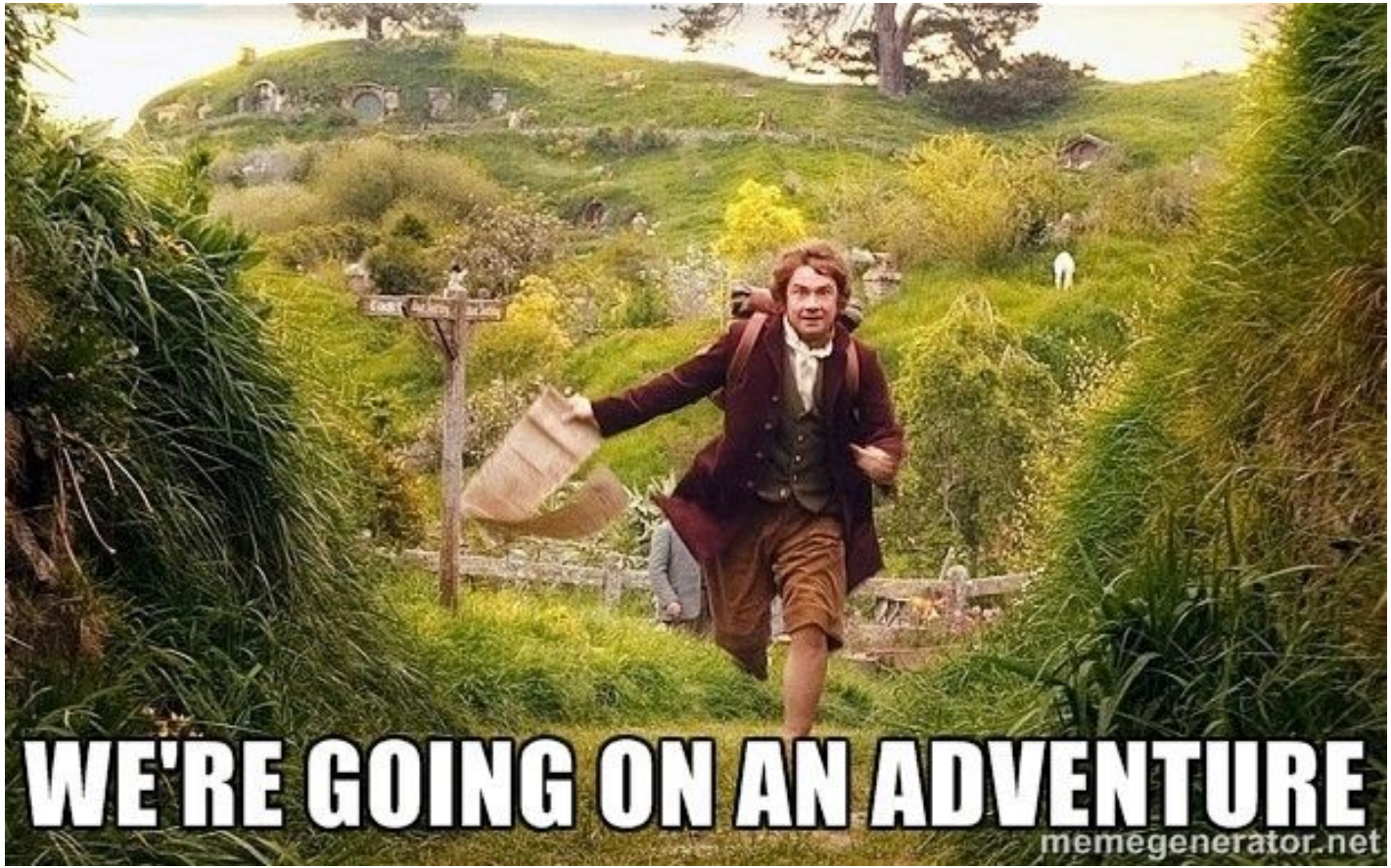

Goals and Vision
API Interactions
Data Modeling
Non-Functional Requirements
Authorization

Goals and Vision

Efficiently Use Resources

https://knowyourmeme.com/photos/16265-you-must-construct-additional-pylons

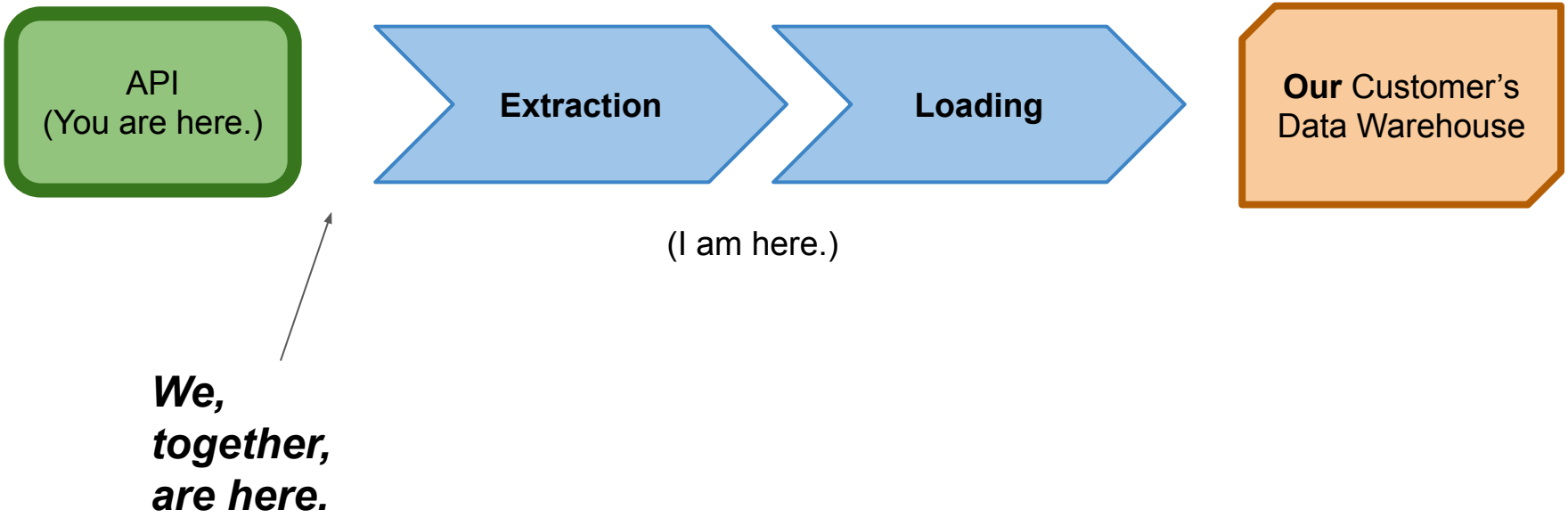
Spread the Word



https://www.pinterest.com/pin/839499186761318979/

Lesson 1 in API Interactions

Where We Are


API
(You are here.)
Extraction
Loading
Our Customer's
Data Warehouse
(I am here.)
We,
together,
are here.

Anatomy of an ELT API Interaction

- ALL the data
- Reliably
- Efficiently

Case Study – CodeMash Attendees

id	name	ctf_rank
be57ca75	Alister Pawson	3
da7acafe	Sam Wisen	1
feedaca7	Jordan Sanovich	2

How can we get ALL this data?

```
-- # FULL TABLE

-- Simple, but inefficient

SELECT id, name, ctf_rank
    FROM CodeMash.Attendees
```

Remember...


ADDITIONAL PYLONS
You must construct them.
not
DIYDESPAIR.COM

Case Study – CodeMash Attendees

id	uniqueidentifier
name	varchar(255)
ctf_rank	byte
registered_at	datetime
last_slept	datetime

```
-- # INCREMENTAL

-- Ideal, "Updated" Value
-- Reliably Sortable by Updated Value

SELECT id, name, ctf_rank, last_slept
    FROM CodeMash.Attendees
    WHERE last_slept >= '2020-01-07 09:00:00'
      AND last_slept <= NOW()
    ORDER BY last_slept ASC
```

Lesson 1: Shoot for Database-like Interactions

Field Selection

Resuming

Sorting

```
SELECT id, name, ctf_rank, last_slept
    FROM CodeMash.Attendees
    WHERE last_slept >= '2020-01-07 09:00:00'
    ORDER BY last_slept ASC
```

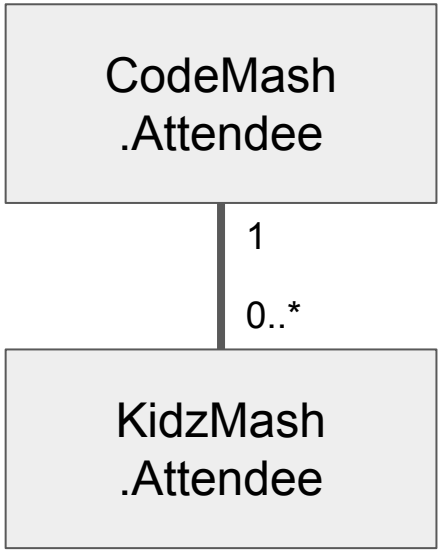
Lesson 1: In The Wild

```
xql_filter = "updated >= '%s' order by
updated asc" % "2019-01-08T07:00:00Z"
```

```
GET /tickets?
updated_since=2019-01-08T07:00:00Z
order_by=updated
order_type=asc
include=time_to_resolution,assignee,tags
HTTP/1.1
Host: agilely.io
```

Lesson 2 in Data Modeling

Anatomy of an ELT Data Model

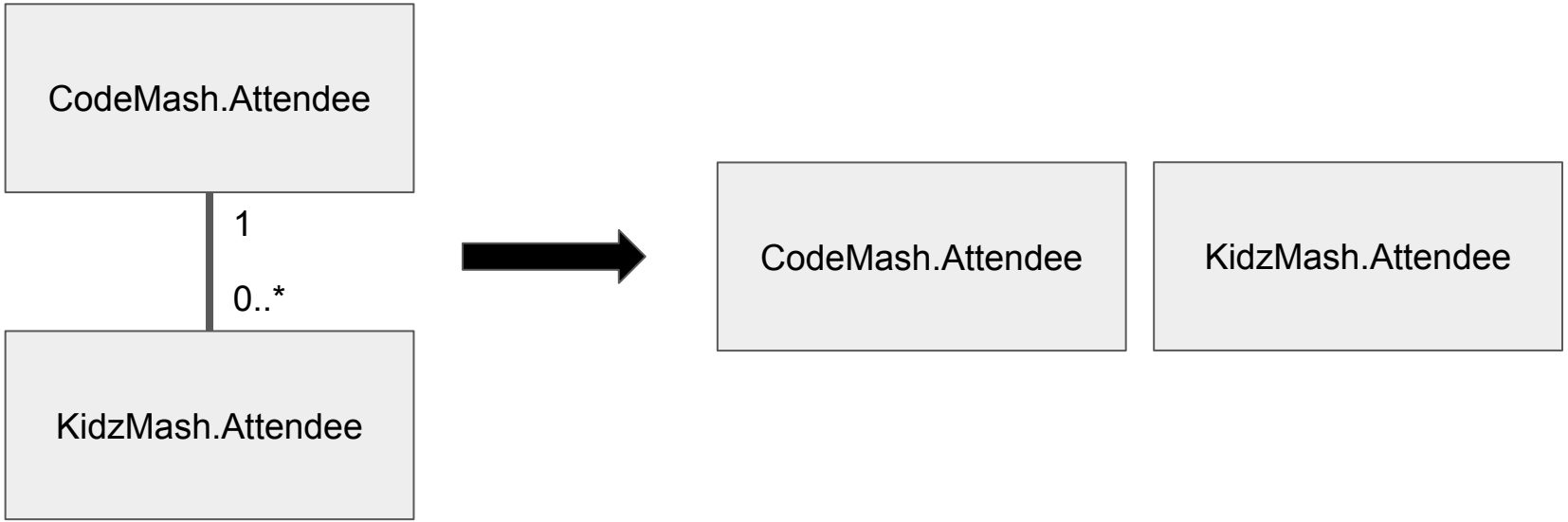


GET /attendees/{id}/kidz

Anatomy of an ELT Data Model

```
GET /attendees/1234/kidz/
GET /attendees/1235/kidz/
GET /attendees/1236/kidz/
GET /attendees/1237/kidz/
GET /attendees/1238/kidz/
GET /attendees/1239/kidz/
GET /attendees/1240/kidz/
GET /attendees/1241/kidz/
...
```

Anatomy of an ELT Data Model



Anatomy of an ELT Data Model

```
GET /attendees/1234/kidz/
GET /attendees/1235/kidz/
GET /attendees/1236/kidz/
GET /attendees/1237/kidz/
GET /attendees/1238/kidz/
GET /attendees/1239/kidz/
GET /attendees/1240/kidz/
GET /attendees/1241/kidz/
...
```

```
GET /attendees/
GET /kidz/
```

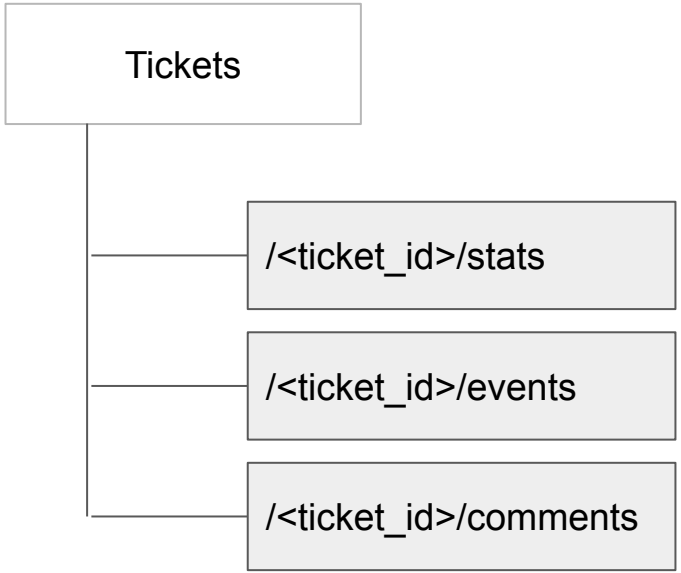
...fin

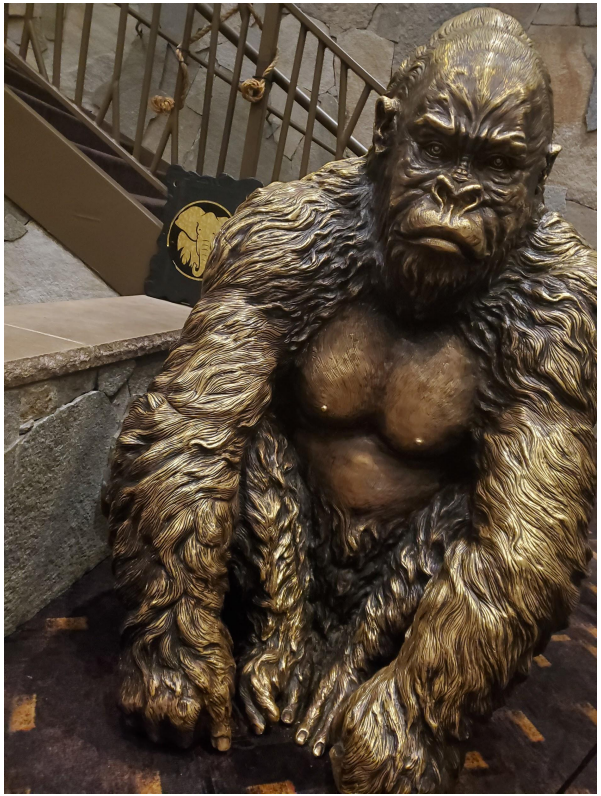
Lesson 2 – Avoid Sub-Resources

```
for account in get_accounts() {
    for department in get_departments(account.id) {
        for user in get_users(account.id,department.id) {
            for invoice in get_invoices(account.id, department.id, user.id) {
                write_to_pipeline(invoice)
            }
        }
    }
}
```

Lesson 2: In The Wild

- First Attempt: 4 Requests Each



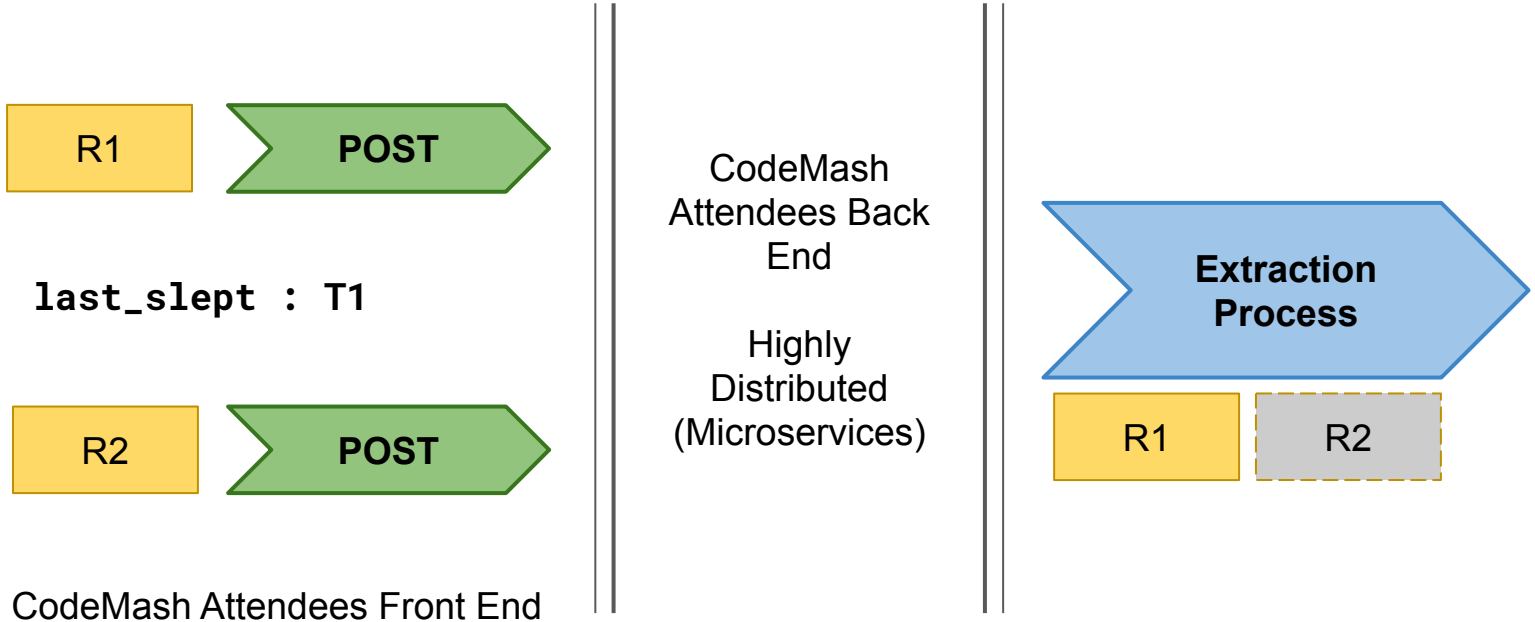


Lesson 3 in Non-Functional Requirements

Non-Functional Requirements

- Reliability
- Data Typing
- Error Messaging

Reliability



Data Typing

Discoverable

GET /api/describe

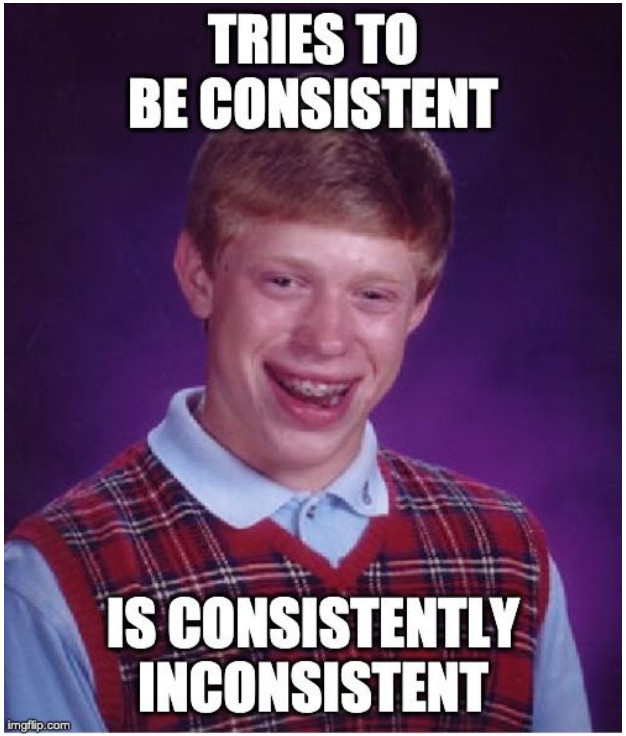
Consistent

12 vs. 0.0

Error Messaging

```
HTTP 200 OK
{“status”: “error”}
```

Lesson 3 – Be Consistent



https://imgflip.com/memegenerator/Bad-Luck-Brian

Lesson 3: In The Wild

```
[error for error
 in ex.fault.detail.FaultDetail.errors
 if error.ErrorType == “AuthError”]
```

Lesson 4 in Authorization

Authorization

Token-Based	...user generated, easy to revoke and recreate
OAuth1	...encrypted, nonced, signed, validated requests
OAuth2	...three-legged handshake, no password exchanged
Basic Auth	...credentials appear directly in request headers
One of these and then some?	… … it happens!

It really depends on your security needs...

Authorization – Guidance

Token-Based ...give users the ability to easily revoke tokens

OAuth1 ...complicated, keep to spec and *document*

OAuth2 ...allow viewing/control over redirect_urls, client ID, secret...

Basic Auth ...please don't put credentials in the URL string

One of these and then some? … … ***document*** it!

API Documentation

Scopes

GET /oauth/token
&scope=invoices.read,
invoiceItems.read,...

Examples

All Images

Lesson 4 – Empower Your API Users

https://imgflip.com/memetemplate/196061783/hide-the-pain-harold-drummer

Lesson 4: In The Wild

Scopes are period-separated strings with a pluralized resource name (e.g. "leads") and a level of access (e.g., read, write, delete or all). For example, …

(Condensed and Paraphrased for brevity and to protect the identities of those involved.)

API Interactions **Shoot for Database-Like Interactions**	Data Model **Avoid Sub-Resources**
Non-Functional Requirements **Be Consistent**	Authorization **Empower Your API Users**

Conclusion

- Everything’s a balance. It’s tough.
  - Between your App’s needs, your customer’s needs, your infrastructure, security, the nature of the data, etc.

- These are just a few of the big points to help guide you when making decisions, so that we can work together for the benefit of our mutual customers as the amount of APIs grows every year

END

(thank you!)

(please leave feedback in the app)

Slack (singer.io) – @dmosora
Slack (CodeMash) – @danmosora

Slides: github.com/dmosorast/api-design-for-etl