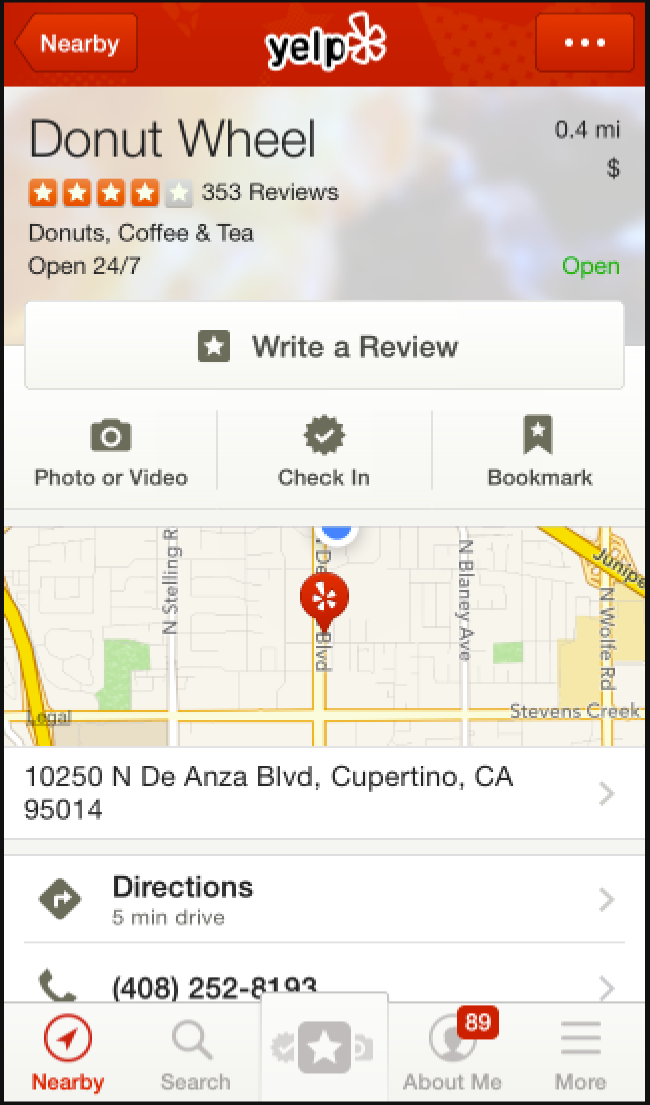
Building Mobile APIs with Services

by Stephan Jaensch - sjaensch@yelp.com

What's Yelp?

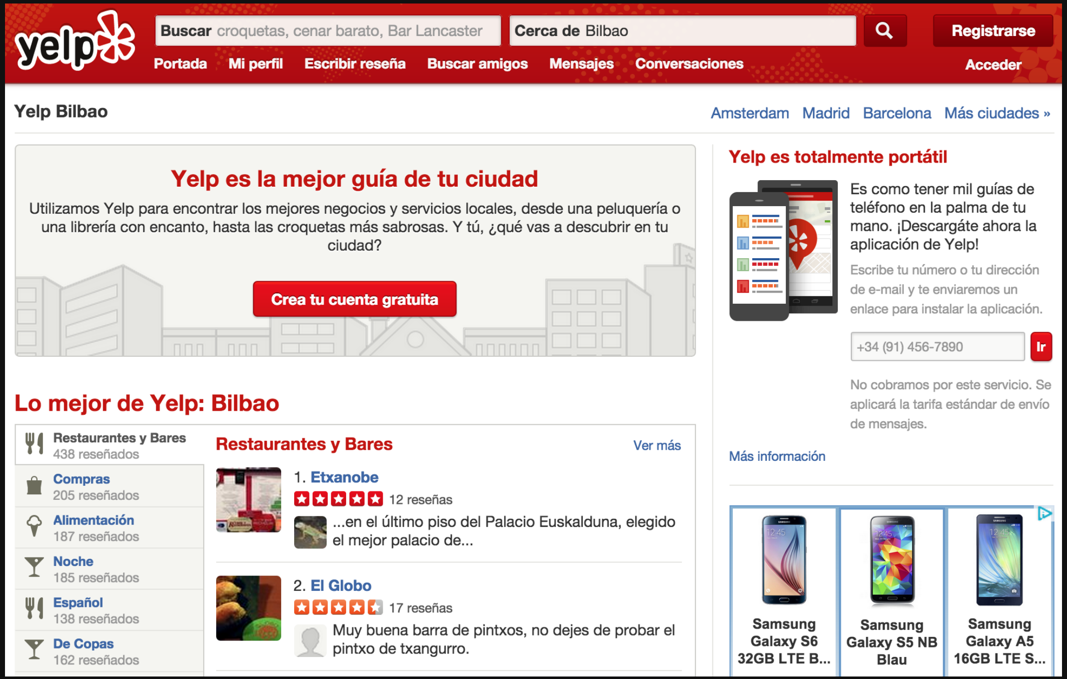


Donut Wheel 0.4 mi
$
353 Reviews
Donuts, Coffee & Tea
Open 24/7 Open

10250 N De Anza Blvd, Cupertino, CA 95014

Directions
5 min drive

(408) 252-8193



Yelp Bilbao — Amsterdam Madrid Barcelona Más ciudades »

Yelp es la mejor guía de tu ciudad

Utilizamos Yelp para encontrar los mejores negocios y servicios locales, desde una peluquería o una librería con encanto, hasta las croquetas más sabrosas. Y tú, ¿qué vas a descubrir en tu ciudad?

Yelp es totalmente portátil

Es como tener mil guías de teléfono en la palma de tu mano. ¡Descargáte ahora la aplicación de Yelp!

Escribe tu número o tu dirección de e-mail y te enviaremos un enlace para instalar la aplicación.

+34 (91) 456-7890

No cobramos por este servicio. Se aplicará la tarifa estándar de envío de mensajes.

Más información

Lo mejor de Yelp: Bilbao

Restaurantes y Bares	438 reseñados
Compras	205 reseñados
Alimentación	187 reseñados
Noche	185 reseñados
Español	138 reseñados
De Copas	162 reseñados

Restaurantes y Bares

1. Etxanobe — 12 reseñas
...en el último piso del Palacio Euskalduna, elegido el mejor palacio de...

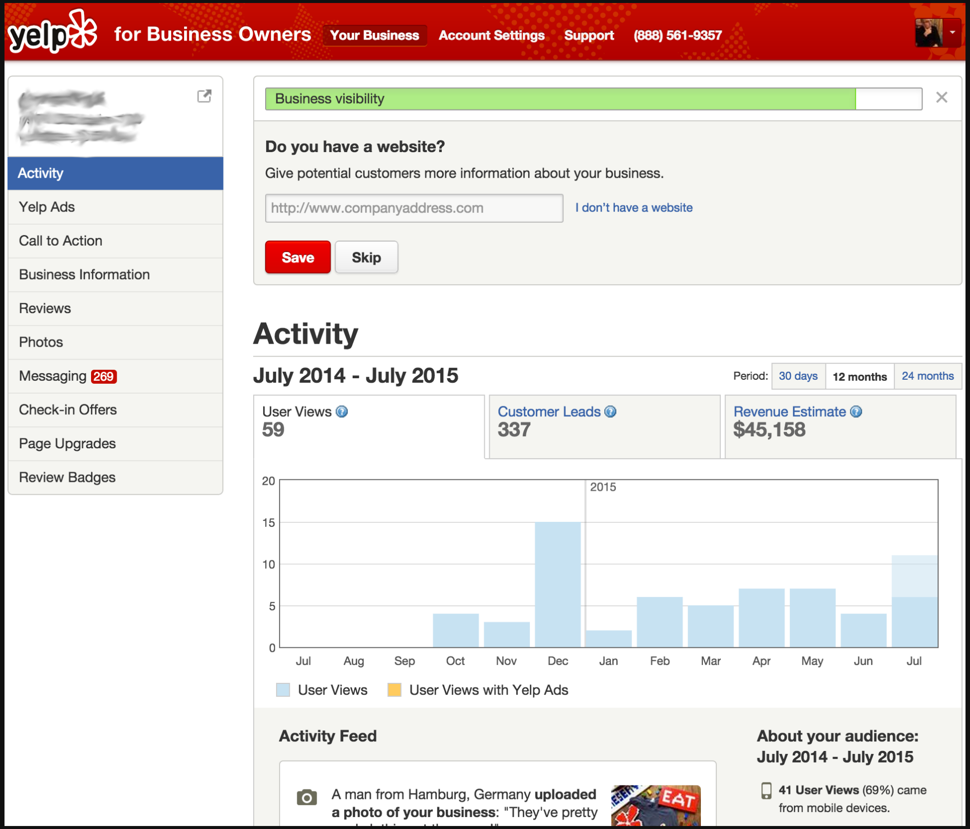
2. El Globo — 17 reseñas
Muy buena barra de pintxos, no dejes de probar el pintxo de txangurro.

Samsung Galaxy S6 32GB LTE B...
Samsung Galaxy S5 NB Blau
Samsung Galaxy A5 16GB LTE S...

- connect people with great local businesses
- website, apps, mobile site
- 142 million monthly unique visitors
- 77 million reviews

Yelp for Biz Owners

- measure visitor activity on your page
- interact with customers
- upload photos



[redacted]

Do you have a website?

Give potential customers more information about your business.

http://www.companyaddress.com

Activity

July 2014 - July 2015

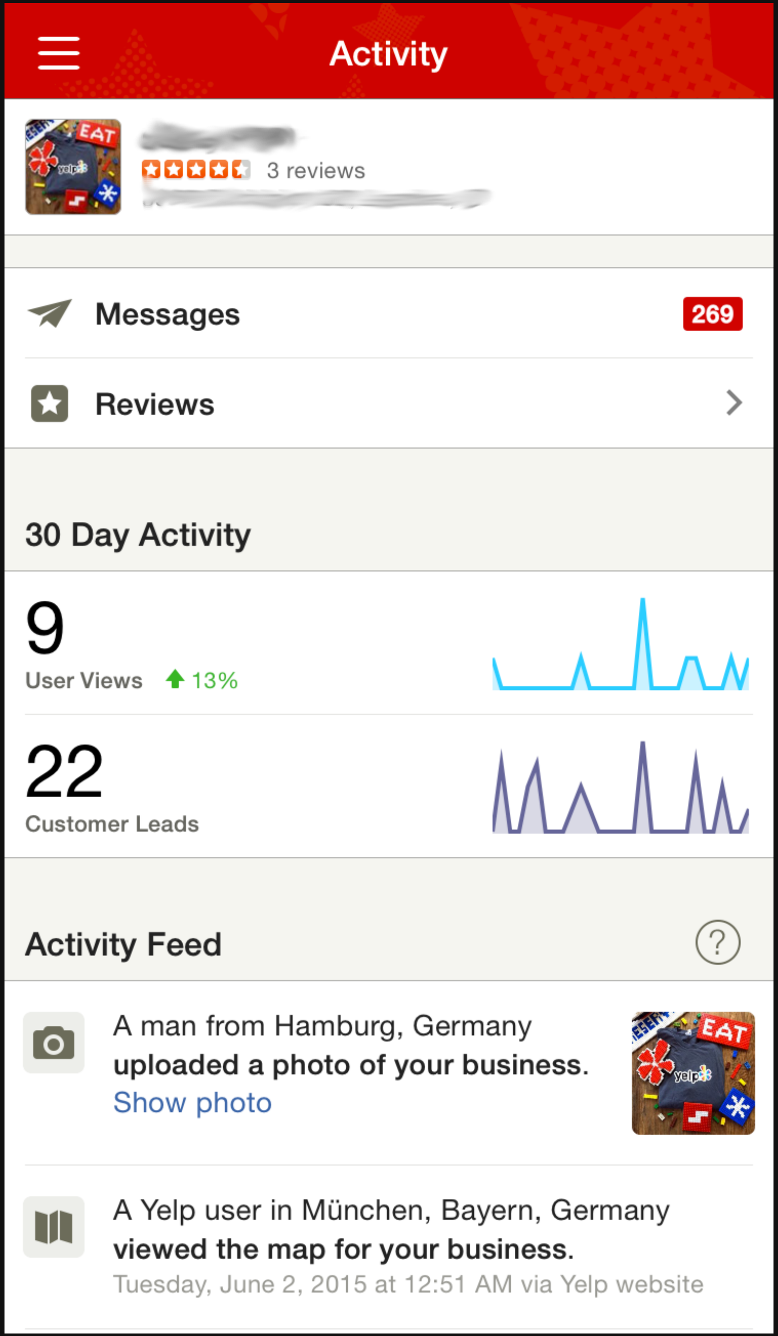
User Views	Customer Leads	Revenue Estimate
59	337	$45,158

Activity Feed

A man from Hamburg, Germany **uploaded a photo of your business**: "They've pretty

About your audience: July 2014 - July 2015

41 User Views (69%) came from mobile devices.



Activity

[redacted]

3 reviews

30 Day Activity

9	User Views ↑ 13%
22	Customer Leads

Activity Feed

A man from Hamburg, Germany **uploaded a photo of your business.** Show photo

A Yelp user in München, Bayern, Germany **viewed the map for your business.**
Tuesday, June 2, 2015 at 12:51 AM via Yelp website

whoami

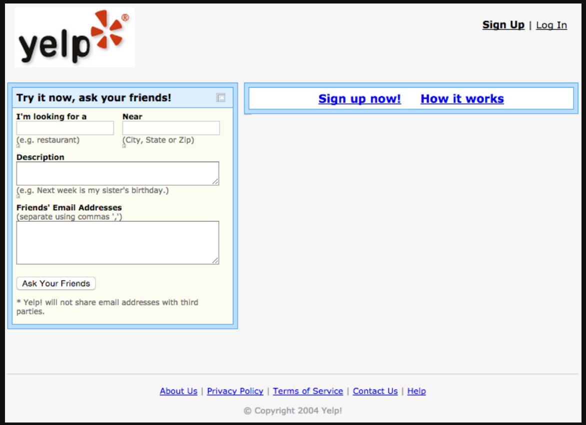
backend developer for the Biz Owner App

worked on the main Yelp app backend before that

Python user since 2008

did a lot of Django work in the past

Yelp: a brief history lesson


Sign Up | Log In

Try it now, ask your friends!

I'm looking for a (e.g. restaurant)

Near (City, State or Zip)

Description (e.g. Next week is my sister's birthday.)

Friends' Email Addresses (separate using commas ',')

Ask Your Friends

* Yelp! will not share email addresses with third parties.

Sign up now! How it works

About Us | Privacy Policy | Terms of Service | Contact Us | Help

© Copyright 2004 Yelp!

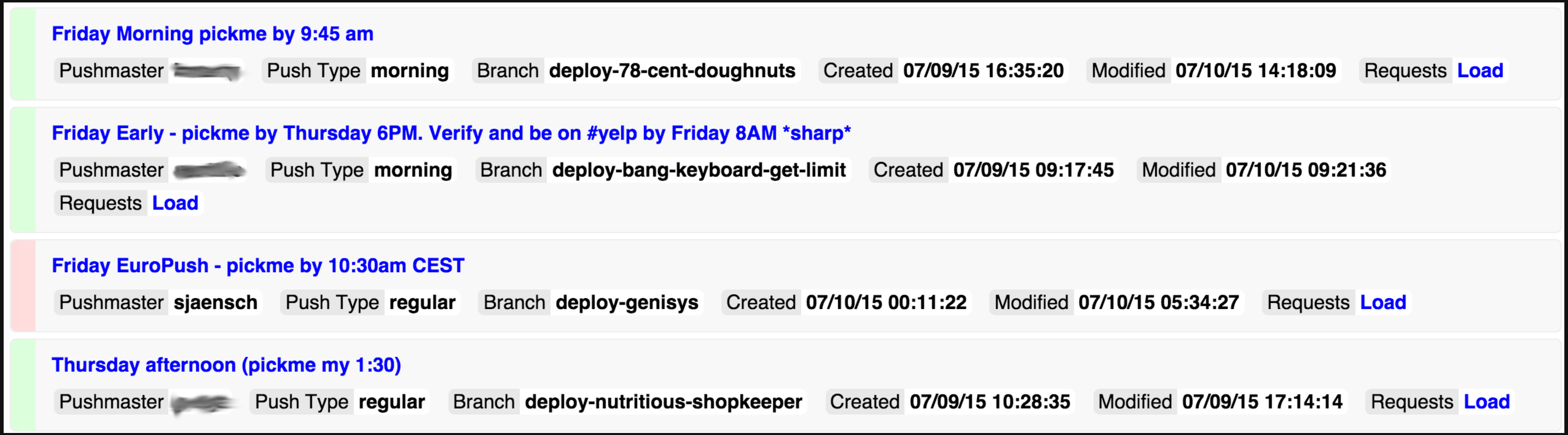
- founded in 2004
- all code in one central repository ('yelp-main')
- web, mobile web, mobile backend, business owner site
- a lot of homegrown code
- new abstractions introduced without removing the old ones
- as Yelp grew, this started to become a bottleneck

The Yelp push process

Code deployments ("pushes") are done several times a day

Run by a pushmaster, an engineer with production system access

People join a push ("pickme")



Friday Morning pickme by 9:45 am

Pushmaster [redacted] Push Type **morning** Branch **deploy-78-cent-doughnuts** Created **07/09/15 16:35:20** Modified **07/10/15 14:18:09** Requests **Load**

Friday Early - pickme by Thursday 6PM. Verify and be on #yelp by Friday 8AM *sharp*

Pushmaster [redacted] Push Type **morning** Branch **deploy-bang-keyboard-get-limit** Created **07/09/15 09:17:45** Modified **07/10/15 09:21:36** Requests **Load**

Friday EuroPush - pickme by 10:30am CEST

Pushmaster **sjaensch** Push Type **regular** Branch **deploy-genisys** Created **07/10/15 00:11:22** Modified **07/10/15 05:34:27** Requests **Load**

Thursday afternoon (pickme my 1:30)

Pushmaster [redacted] Push Type **regular** Branch **deploy-nutritious-shopkeeper** Created **07/09/15 10:28:35** Modified **07/09/15 17:14:14** Requests **Load**

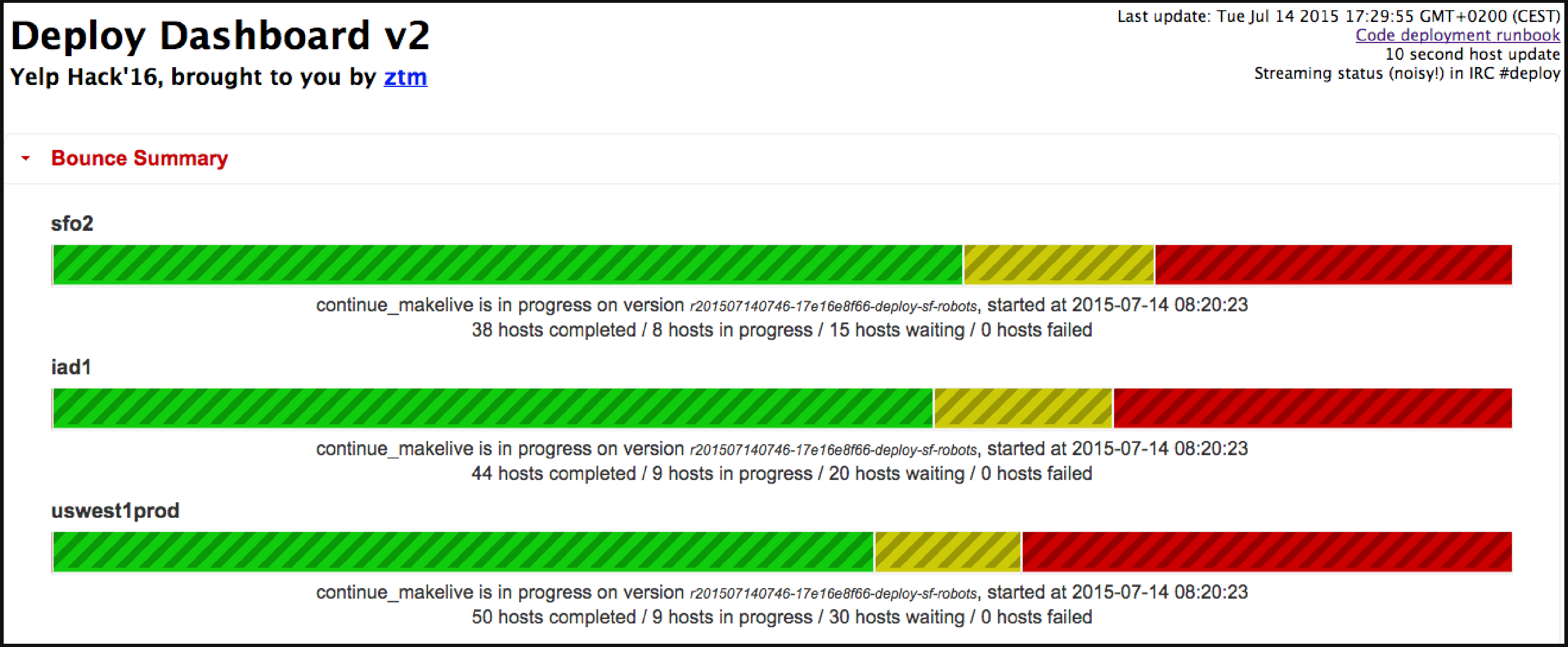
Running a push

Automatic checks make sure there are no merge conflicts

deployment branch is deployed to a stage system

after verification, it’s sent to production

~2 hour process, with no upper bound



Deploy Dashboard v2

Yelp Hack'16, brought to you by ztm

Last update: Tue Jul 14 2015 17:29:55 GMT+0200 (CEST)
Code deployment runbook
10 second host update
Streaming status (noisy!) in IRC #deploy

Bounce Summary

sfo2

continue_makelive is in progress on version *r201507140746-17e16e8f66-deploy-sf-robots*, started at 2015-07-14 08:20:23
38 hosts completed / 8 hosts in progress / 15 hosts waiting / 0 hosts failed

iad1

continue_makelive is in progress on version *r201507140746-17e16e8f66-deploy-sf-robots*, started at 2015-07-14 08:20:23
44 hosts completed / 9 hosts in progress / 20 hosts waiting / 0 hosts failed

uswest1prod

continue_makelive is in progress on version *r201507140746-17e16e8f66-deploy-sf-robots*, started at 2015-07-14 08:20:23
50 hosts completed / 9 hosts in progress / 30 hosts waiting / 0 hosts failed

I HAVE A SOLUTION
FOR THAT

Modularize

you can run only so many pushes a day

so let’s build services!

Why Services?

each service is developed and deployed independently

services are usually small, covering only one aspect or set of features

easy to parallelize thanks to async HTTP requests, so it might even speed your code up

http://bit.do/fowler-service
http://bit.do/microservices
https://github.com/Yelp/service-principles

Why Not Services?

Why Not Services?

consistency is really hard

no clear dependency / usage graph

need to maintain interfaces “forever”

testing one huge, mostly self-contained codebase is easy; how do you test services?

How to make sure it doesn't break

unittests

...are great, but not enough

a lot of breakage if interfaces change

our solution: acceptance tests

as close to production as possible without using dedicated stage environments

Testing SOA at Yelp

spin up all components you need, using production code

done with docker-compose

heavyweight: take time to run, setup grows with the number of services you call

Setting up acceptance testing

```
configs:
build: acceptance/configs/
volumes:
        - "./logs:/tmp/logs"

bizapp:
build: .
links:
        - bizfeed
        - businessmedia
        - internalapi
        - sessionsservice
        - ruleserv
volumes_from:
        - configs
ports:
        - 13849
```

```
internalapi:
image: docker-dev/internalapi-testing
links:
        - gearman
        - memcache
        - databaseprimary
        - databaseaux
        - databasebatch
        - geocoderservice
environment:
        YELP_USE_GEARMAND: True
```

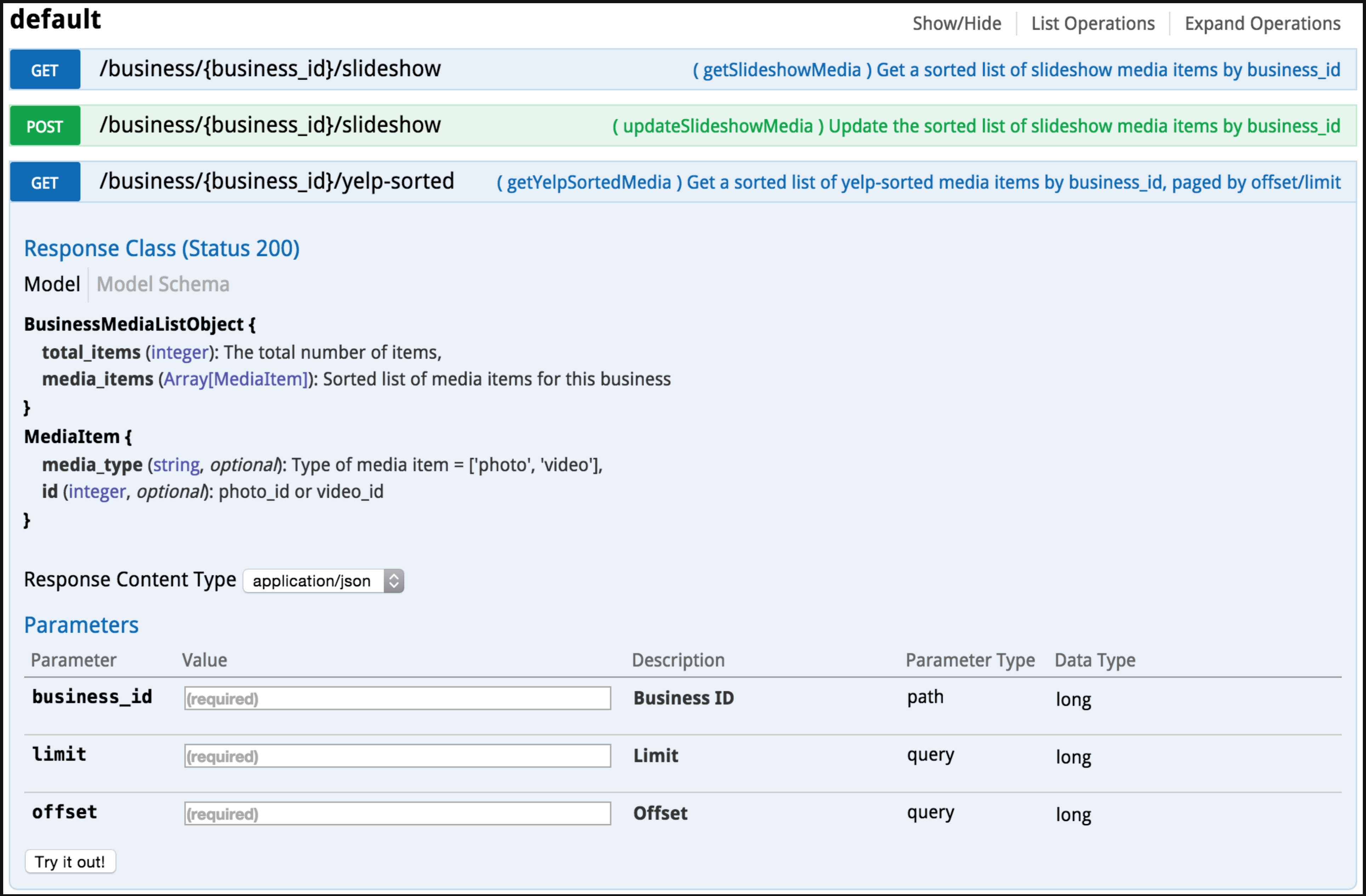
The Yelp service stack

originally we used tornado; didn’t work well

now: Pyramid, uWSGI, SQLAlchemy

HTTP and JSON for communication

Swagger to specify the API and do the inter-service calls


default

Show/Hide | List Operations | Expand Operations

GET /business/{business_id}/slideshow (getSlideshowMedia) Get a sorted list of slideshow media items by business_id

POST /business/{business_id}/slideshow (updateSlideshowMedia) Update the sorted list of slideshow media items by business_id

GET /business/{business_id}/yelp-sorted (getYelpSortedMedia) Get a sorted list of yelp-sorted media items by business_id, paged by offset/limit

Response Class (Status 200)

Model | Model Schema

BusinessMediaListObject {
  total_items (integer): The total number of items,
  media_items (Array[MediaItem]): Sorted list of media items for this business
}
MediaItem {
  media_type (string, *optional*): Type of media item = ['photo', 'video'],
  id (integer, *optional*): photo_id or video_id
}

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
business_id	(required)	**Business ID**	path	long
limit	(required)	**Limit**	query	long
offset	(required)	**Offset**	query	long

Try it out!

Swagger

does request and response validation

data structure and basic type checking of the individual fields

works dynamically by reading a service's spec, no need to generate and update client libraries

The Biz App service

a special snowflake since it's one of the very few services reachable from the outside

not constrained to one area (like business media)

no local datastore

oftentimes just a proxy, calling yelp-main and other services

The Biz App [illegible]ervice API

[illegible]ESTy model

[illegible]ne resou[illegible]e per endpo[illegible]nt

do multiple calls (to [illegible]fferent endp[illegible]ints) to [illegible]tch related resources

get concurr[illegible]ncy for free (i[illegible]in[illegible] async calls)

some say a lot of simpl[illegible]lls ar[illegible]asier to scale than fewer complicated ones

The Biz App service API

one endpoint per client (app) page

for write (POST) endpoints, also send the client the data it needs to display the follow-up page

aggregate and enrich data we retrieve from yelp-main and other services

a high-level interface that translates to our low-level service APIs

Developing a mobile app backend

mobile apps have releases

in our case, they’re synchronized, both in time and in features

iOS apps need to be reviewed; might take 10+ days

you probably also want to test before releasing

meaning: API needs to be done sooner than client implementation

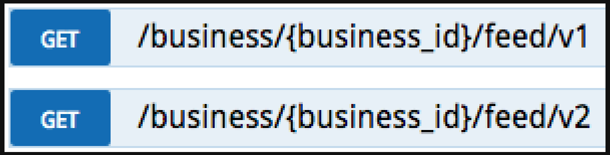
way sooner than release date

It's not web development

you can’t upgrade apps whenever you upgrade the server

actually, some users never upgrade

so your APIs need to be backwards compatible - forever



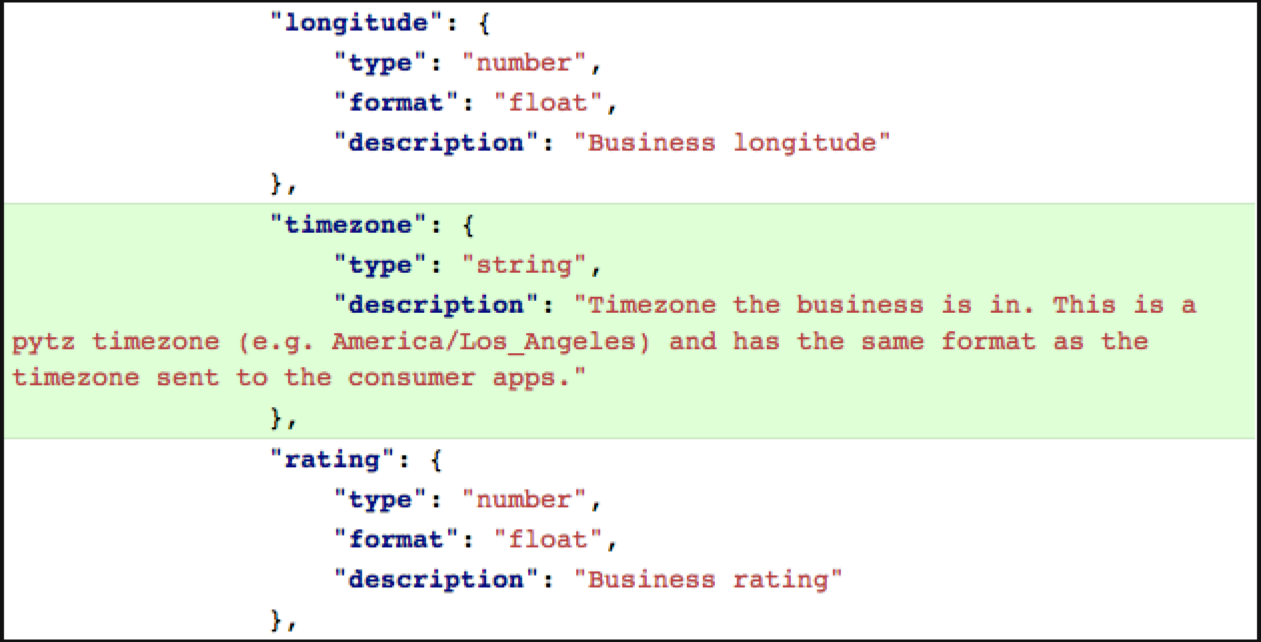
GET /business/{business_id}/feed/v1

GET /business/{business_id}/feed/v2

Multi-version API

maintaining multiple versions can become costly

adding fields is backwards compatible



```
                "longitude": {
                    "type": "number",
                    "format": "float",
                    "description": "Business longitude"
                },
                "timezone": {
                    "type": "string",
                    "description": "Timezone the business is in. This is a
pytz timezone (e.g. America/Los_Angeles) and has the same format as the
timezone sent to the consumer apps."
                },
                "rating": {
                    "type": "number",
                    "format": "float",
                    "description": "Business rating"
                },
```

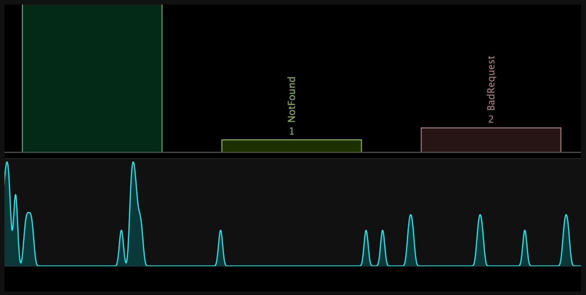
Monitoring

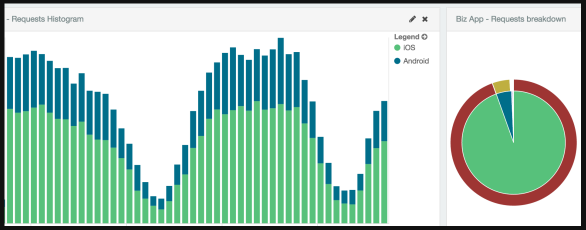
number of requests, server errors, task queues, sent push notifications…

ElastAlert: it's open source!

app crashes: Crashlytics

you need an on-call rotation: we use PagerDuty





Gearman queue sizes
Push notification queues
Gearman task rates
Push notifications task rates

More about services @Yelp

Scott Triglia: Arrested Development - surviving the awkward adolescence of a microservices-based application

Friday, 11am, *Python Anywhere* room

The shameless plugs

We're hiring! Check out yelp.com/careers

Interested? Contact me even if you don't find an open job position that fits you, we're always looking for talented people!

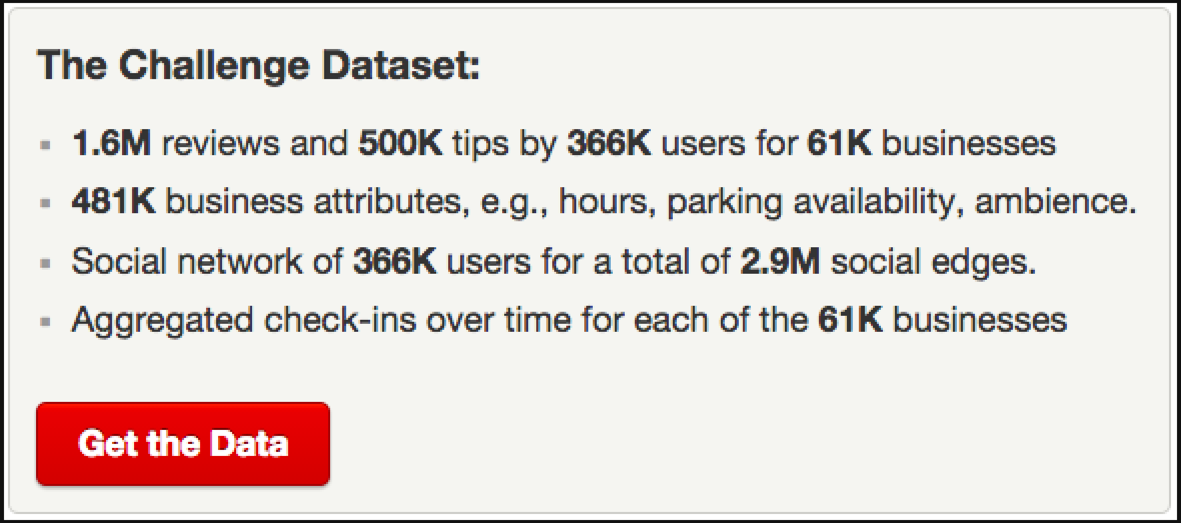
yelp.com/engineering aggregates the blog posts, open source projects and more

follow us on Twitter: @Yelp, @YelpEngineering

Have fun and win prizes

The Yelp Dataset Challenge: yelp.com/dataset_challenge

Want to work with data, but have no data lying around?



The Challenge Dataset:

- **1.6M** reviews and **500K** tips by **366K** users for **61K** businesses
- **481K** business attributes, e.g., hours, parking availability, ambience.
- Social network of **366K** users for a total of **2.9M** social edges.
- Aggregated check-ins over time for each of the **61K** businesses

THANK YOU

questions?