

TIM GOURLEY - OKC RUBY

INTRO TO APIS

ABOUT TIM


PHASE 2

- ▶ *Currently a developer at Phase 2*
<https://www.phase2online.com>
- ▶ *Professional Rubyist for over a decade*

 @docbratta

 therealtimgourley

 bratta

 @tgourley@ruby.social
<https://timgourley.com>
<https://night.coffee>





BUZZ BUZZ BUZZ

ALL ABOUT APIS

APPPLICATION **P**ROGRAMMING **I**NTERFACE

INSERT BORING LIST OF COMMON TYPES OF APIS HERE

▶ **Libraries & Frameworks**

- ▶ *Working with Ruby Gems*
- ▶ *iOS/Android Development*

▶ **Operating System APIs**

- ▶ *Windows API, POSIX with Linux/BSD*

▶ **Remote APIs**

- ▶ *Remote Method Invocation (e.g. Java RMI)*

▶ **Web APIs**

- ▶ *HTTP*
- ▶ *Web Services via SOAP*
- ▶ *REST web services*

**LET'S FOCUS ON
THE WEB FOR FUN
AND PROFIT!! ***

* Profit not guaranteed. Your mileage may vary. Batteries not included. A Møøse once bit my sister... No realli! She was Karving her initials on the møøse with the sharpened end of an interspace tøøthbrush given her by Svenge - her brother-in-law - an Oslo dentist and star of many Norwegian møvies: "The Høt Hands of an Oslo Dentist", "Fillings of Passion", "The Huge Mølars of Horst Nordfink"

WEB API TERMS

- ▶ **Endpoint** - A URI pointing to an HTTP-based web service, where requests are made and responses are expected.
- ▶ **SOAP** - An XML-based messaging protocol for exchanging information, over the application layer. Uses WSDL to group and describe endpoints related to the services offered. This is a scary place typically used in the soul-crushing enterprise.
- ▶ **REST** - (i.e. RESTful) represents web resources with uniform and predefined stateless operations, utilizing the URI and HTTP verbs to define both the actions and the objects used in the communication. Generally a happier approach to web services.

(MORE) WEB API TERMS

- ▶ **Auth/Access Token** - Identifier providing proof that the user has permission to access and/or manipulate data over the web service.
- ▶ **OAuth 2.0** - An open standard for token-based authentication/authorization so third parties can securely gain access to content without knowledge of a user's password.
- ▶ **OpenID Connect** - Allows a user to **authenticate** into multiple websites or web services with a single set of credentials. OpenID Connect builds on OAuth2 to provide both authentication and authorization and issues an identity JSON web token.
- ▶ **Access Key ID/Secret Access Key Authentication** - A pair of keys are used to calculate a hash of data used to “sign” the request sent to the server. This is how Amazon authenticates requests.

CONSUMING APIS

vs.

PROVIDING APIS

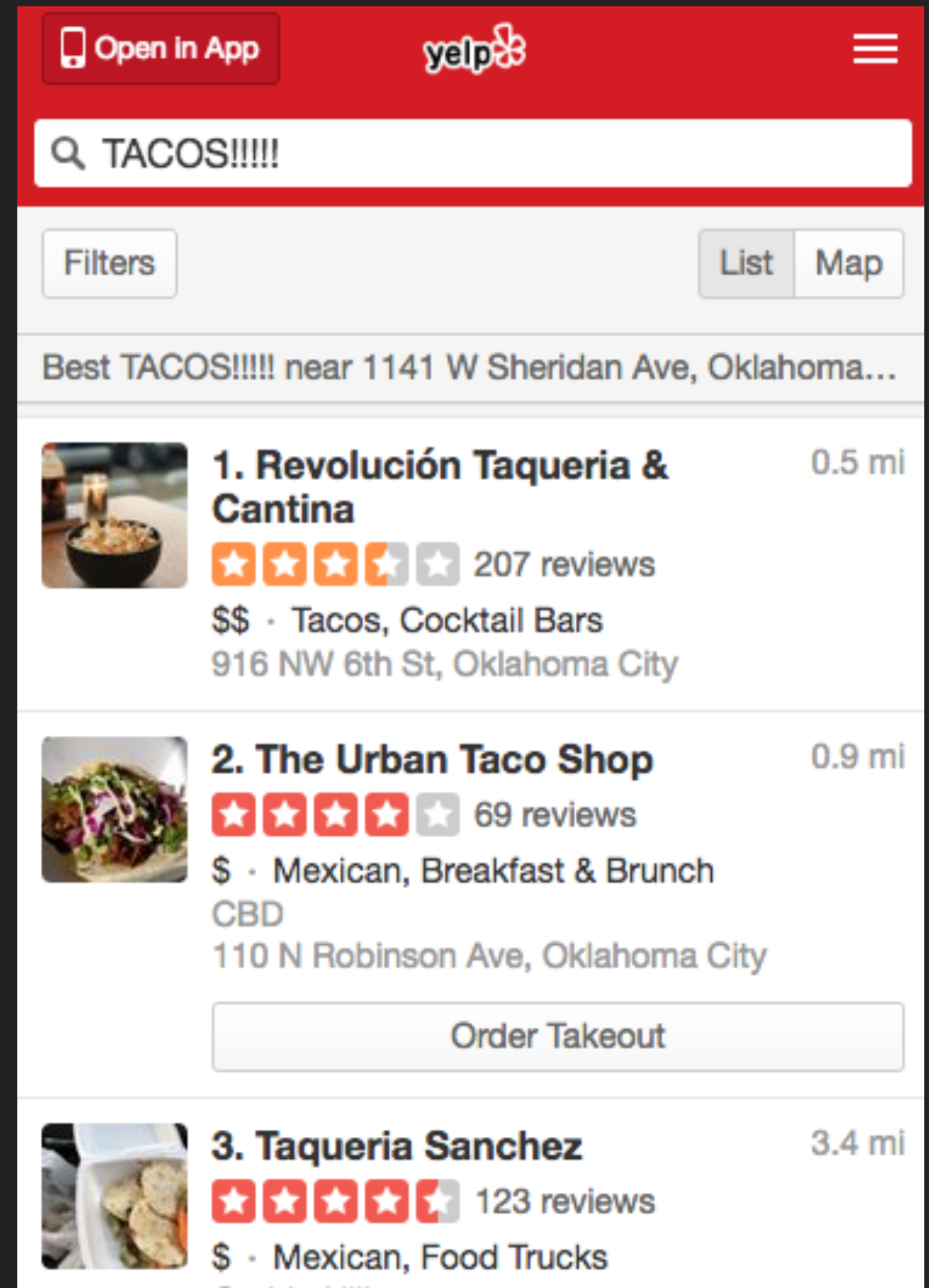
INTRODUCTION TO APIS - CONSUMING

THE PROBLEM:

- ▶ **WHERE THE HECK DO WE EAT?**
There are too many choices. They. Must. Battle.

THE SOLUTION:

- ▶ *Build an app that lets restaurants battle for dining supremacy.*
- ▶ **YELP!**
- ▶ <https://www.yelp.com/developers>



SEARCHING FOR A BUSINESS

▶ **Url:**

- ▶ `https://www.yelp.com/developers/documentation/v3/business_search`

▶ **Endpoint:**

- ▶ `GET https://api.yelp.com/v3/businesses/search`

▶ **Parameters:**

- ▶ **term** - “restaurants”, “tacos”, or “Starbucks”
- ▶ **location** - Geographical address or partial address; zip codes work
- ▶ **limit** - Default of 20, max is 50

SUPER HAPPY FUN CODING TIME WITH YOU & ME!

WHEN YOU DON'T CREATE THINGS, YOU BECOME DEFINED BY YOUR TASTES RATHER THAN ABILITY. YOUR TASTES ONLY NARROW AND EXCLUDE PEOPLE. SO CREATE.



_why the lucky stiff

CREATING AN API WITH RUBY

CREATING A WEB API WITH RAILS

- ▶ *`rails new wicked_cool_api --api`*
- ▶ Uses a limited set of middleware than normal. Run *`rails middleware`* to see exactly what is loaded.
- ▶ Controllers inherit from *`ActionController::API`* instead of *`ActionController::Base`*
- ▶ Generators skip views, helpers, and assets.

CREATING A WEB API WITH SINATRA

- ▶ *gem install sinatra*
- ▶ *Not tied to any particular ORM*
- ▶ *Clean DSL for RESTful service creation*
- ▶ *Don't get a lot of the niceties of Rails (i.e. IP Spoofing detection, strong parameters, conditional GETs/ETag support) without additional configuration.*

ALEXA, WHAT DO I DO WITH MY API NOW?

- ▶ Add **rspec** (or some other testing library). Do this first.
- ▶ Use **ActiveModel::Serializers** to customize your JSON.
- ▶ Enable CORS (Cross-Origin Resource Sharing) by installing the **rack-cors** middleware gem.
- ▶ Consider namespaced URIs to version your API.
- ▶ Add rate limiting/request throttling middleware with the **rack-attack** gem.
- ▶ Add authentication/authorization with something like JSON Web Tokens (JWT) with the **jwt** gem.

[HTTPS://GITHUB.COM/BRATTA/WICKED_COOL_API](https://github.com/bratta/wicked_cool_api)

[HTTPS://GITHUB.COM/BRATTA/CLINCH_CLASH](https://github.com/bratta/clinch_clash)