Intro to RESTful APIs

A 10 slide example-based primer on REST.

Joshua Austin
jbaustin.com

Digital follows Physical

- Concepts in the digital world usually follow a metaphor that exists in the physical/analog world; RESTful APIs are no different.
- An API, short for Application Programming Interface, is the way that two distinct entities digitally carry out work with each other by exposing and sharing services and data.
- There are many ways to implement an API. A few are: REST, GraphQL, SOAP, and RPC.

Why REST?

- REST is an approach to architecting an API. The main ingredients are: Resources, Requests, and Responses (stay tuned for more on these).
- It stands for REpresentational State Transfer and is driven by a handful of conventions that make it very powerful and flexible. The conventions are (LUCCCS):
  - Layered System
  - Uniform Interface
  - Client – Server
  - Cacheable
  - **Code on demand (optional)**
  - Stateless
- Don’t worry! We'll cover each of these through examples.

The Three "R's" of REST

- Resource – an entity on the server that is available for interaction.
  - Examples: Restaurant, Menu Item, Order, or Customer
- Request – the act of the client interacting with a resource on the server.
  - Consists of a Method and Content (optional)
  - Methods: GET, PUT, POST, DEL, PATCH, OPTIONS, HEAD, TRACE, CONNECT
  - Content: typically JSON, XML, or Text data
- Response - the result that the server passes to the client in response to a request.
  - Consists of a Status Code and Content (optional)
  - Status Codes: 2xx, 3xx, 4xx, 5xx
  - Content: typically JSON, XML, or Text data

Our Example

You are dinning in at a restaurant with five friends. The restaurant is located in the United States and you have already been seated by a hostess.

Example RRRs

These examples show how a physical example maps to a digital resource, request and response.

Physical Concept	Digital Resource Request - Client		Digital Response - Server
Ask for Menu	GET	myapi.com/Menu	200 [{MenuItem1}, {MenuItem2}]
Place an order	POST	myapi.com/Order	201 {OrderID: 1}
Change entire order	PUT	myapi.com/Order/1	200
Update salad dressing	PATCH	myapi.com/Order/1	200
Cancel order	DELETE	myapi.com/Order/1	204
Check if order was placed	HEAD	myapi.com/Order/1	200
Ask for order details	GET	myapi.com/Order/1	200 {Order Details}

RESTful Conventions

- Layered-System: you (client) are not concerned with the mechanics of how the restaurant (server) makes your meal. They are able to change their approach at any time and use as many resources as needed.
- Uniform Interface – The way in which you interact with the wait staff doesn't change with each interaction or visit. There is consistency across how you and your friends interact with the staff.
- Client – Server -> You and your friends are the "Client" & the Restaurant Staff is the "Server." By using menus and wait staff (the UI) and a centralized kitchen, the restaurant is able to scale and serve many clients at the same time.
- Cacheable - Common results are able to be stored in between the client and server (for perf reasons). A good example of this is the waitress keeping a note of the specials for the day. Imagine if she had to go and ask the chef each time if you and all five friends asked for the specials.
- Stateless – If you ate at the restaurant each day, you would still need to make a complete order each time and not rely on the restaurant remembering "your usual"

Say what?

Each response from the server has a Status Code that signals to the client the result of the Request. Here are example Status Codes for situations from our dining example.

- 2xx – These are the good. 200 for "OK" and 201 for "Created"
- 3xx – These are for redirects / more actions needed
- 4xx - These usually indicate "less than ideal" situations.
  - 400 means a bad request - would be one of your friends ordering in Russian at an English-speaking restaurant,.
  - 401 is unauthorized – would be your friend trying to order before they were seated.
  - 403 means forbidden - would be another friend trying to order off a secret menu.
  - 404 means not found - would be you ordering an item that doesn't exist
- 5xx – these are error cases
  - 500 – general error – would be a case where the kitchen staff messed up your order and wasn't able to complete it.
  - 503 – service unavailable – would be you ordering after the kitchen closed for the day.

Summary

- RESTful APIs are ways for distinct entitles to conduct some form of business with each other in a consistent and scalable manner.
- REST hinges on LUCCCS and the three Rs: Resource, Request, & Response.
- When in doubt, think about going out to dinner :)

Additional Resources

- RESTful APIs
- HTTP Methods
- HTTP Status Codes
- RESTful APIs for Beginners