Full Stack Development

Lecture 11: APIs implemented with PHP

What is an API?

- Application Programming Interface
  - a set of functions and procedures allowing access to the features or data of some other software service.
  - Routes http://example.com/class/action/values
  - Methods GET POST PUT PATCH DELETE...
  - Expectation of JSON return values

Routes/Endpoints

- Generally Follow CRUD design with addition of searching and listing
  - Create – PUT
  - Show – GET
  - Update – PATCH/POST
  - Delete – DELETE
  - Search – POST/GET
  - List/Show all – GET/POST

- Routes are usually denoted by query parameters. More on this in a second
- Endpoints are the full URI with sort/searching params
- Built on larger to smaller result set premise
  - ex: http://somesite.com**/API/v2/User/Show/{user_id}**

Query Strings vs Query Parameters

- Query strings are the key value pairs that appear in the url AFTER the question mark
  - Used to sort and filter results
  - Ex http://example.com/user/search?**id=1234&fname=john&lname=smith**
- Query parameters (aka path parameters) appear BEFORE the question mark
  - Used to access a specific resource or set of resources
  - Ex http://example.com/**user/search**?id=1234&fname=john&lname=smith

Rate Limits

- What is a rate limit
  - A rate limit is generally a limit on requests that can be made per account/user in a given amount of time.
  - Maintaining an api and the infrastructure for storing and serving data can be costly so companies implement rate limits
  - Most API’s have dev accounts whice are a free tier with a very small limit useful for building and testing but not high enough to serve a functioning site.
  - Example: https://smartystreets.com/pricing

Authorization

- Keys – API keys are the most common way to authorize access to an API
  - Can be set up to limit various aspects
  - Managed by the API owner
  - Stored in DB and programatically checked against
- Bearer Tokens – Usually generated per call and expire after a short amount of time or once a transaction is complete
  - Most often used with payment systems and merchant accounts usually requires a call to an OPTIONS method first
- AuthCodes (old school bearer tokens) – These are still in use but have no real standard in practice. Good luck. RTFM for sure!
- Login/Authorization routes with UN/PW (don't build these unless you have to)
  - Sometimes used with the bearer token method but pretty insecure since you have to send the pw in the request.

Using an API PT 1

1) RTFM! Read the &%($#@* Manual!!
   a. Docs are the source of all truth with an api
   b. They tell you how to set it up what routes do what and what parameters are needed for each request
2) Get an API key
3) Find the route you need
   a. /User/Search?id=, /Users/search, /User/id/
4) Test the basic route with no filtering or sorting to make sure your connection is valid
5) Understand the API’s return structure and formatting
   a. JSON or XML or something else. Old api’s do some weird things…
6) Know your rate limits! Most places will charge you instantly. See AWS...

Using an API PT.2

Lets try one!

1) Open PostMan
2) Open smarty streets US Street Address api docs: https://smartystreets.com/docs/cloud/us-street-api
3) Make a request!

Building an API

- Understand your data!
  - What do you *want* to server the user, specifically?
  - Make sure you can convert your data to JSON
- Decide on route template design
  - User/ or Users/ for show all?
  - GET vs POST and for which routes?
  - Query strings or query params?
  - Every route should follow suit
- Decide on Resource (top level) routes
  - User
  - Account
  - Post
  - Admin
  - Orders
  - etc

Building an API cont.

- Let your routes do the navigation and your classes to the work
  - /User/find/{user_id} route
    - // validate params
    - // find requested thing
    - // format requested thing
    - // return response
  - User class
    - // connect to db
    - // query db
    - // validate results
    - // return result object

For More Info:

- https://codeofaninja.com/2017/02/create-simple-rest-api-in-php.html