

Web Services

Web Services are flexible

- Human conventions on what/how you send/get
- Normal requests to the server
- Not technical differences

Different Conventions

- SOAP is web requests with lots of XML
- GraphQL is a "query language"
 - send JSON-like "query" text
 - Get JSON back
- REST is HTTP as the query language
 - URL is the "resource"
 - HTTP Method is the "interaction"
 - HTTP Status Code is the result
 - Send whatever body
 - get whatever response

REST Services

This course will use REST services

- most common
- not hard to transition away

GraphQL is growing in popularity

- but heavier pieces

REST Urls

The path of the URL for web requests:

- NOT a static file
- NOT an action
- Is an idea of a thing
- Generally a noun
- May use subdirectories for granularity

REST Sample URLs

- **good** /students
- **good** /students/courses
- **good** /cats/internet/Maru
- **good** /cats/habits
- **Bad** /addCat
- **Bad** /callDogs

The URL is something you can do different actions on

HTTP Methods

In REST the HTTP Method in REST is

- what action you are doing
- to the resource

-
- GET - Read
 - POST - Create
 - PUT - Replace
 - PATCH - Update
 - DELETE - Delete

HTML Forms can only do GET/POST

Difference between POST/PUT/PATCH

- POST - Creating where there was no record before
- PUT - Replacing the entire record
- PATCH - Changing parts of record, the rest stays

```
{  
  name: "Jorts",  
  age: 3,  
  color: "Orange Tabby"  
}
```

- POST to create record initially
- PUT to replace it with a new record
- PATCH to change/add/delete a field

Sending Data

Path can have details

- example: `/cats/internet/Maru`
- example: `/students/Svathi/grades`

Query params might be used

- GET has no body
 - search params, filters, etc

Body Params have details

- Usually the record changes

Getting Results

Returned body can be anything

- Often JSON data, even with non-JS server
- Flexible, easy to parse

This Course

We will be mostly

- sending JSON in requests
- getting JSON in response

Will parse JSON and use to rerender HTML

But we will only use a few services anyway