Web Service

* Software that you can interact with via http or https.
  + The web in a web service
* Web Application vs Web Service
  + Web Application is a finished product that people use.
  + Web Service is a software that is designed by other pieces of software and interacted with programmatically.
    - REST services communication in JSON is technically human readable but not meant for humans to directly use.
* RESTful Web Services are the dominant type of web service on the internet.
  + REpresentational State Transfer.
* A truly RESTful web service conforms to these 6 constraints.
  + Stateless
    - There is no session on your REST service.
    - There is no information stored in memory about the user or what they are doing.
    - Think about how we use JWTs to identify who made a request.
      * There is no JWT information stored on the server other than the secret.
  + Cacheable
    - Can choose to cache information on the server.
      * If you are getting a ton of requests for a specific resource than you can temporarily store it on the server rather than fetching from the DB each time.
  + Client-server separation
    - Front-end does not need to know the implementation of the service.
    - A RESTful web service should not be tied to a specific client.
      * You are sending JSONs that any client could parse and use as it sees fit.
      * CONTRAST with sending back information in a pre-formatted HTML table.
      * Should not have server-side rendering.
      * Client agnostic.
  + Layered
    - REST services often call one another.
      * In large applications and in particular microservices to complete a request at one endpoint the service might call a different RESTful web service located on another server.
      * EX
        + Some companies have a RESTful web service dedicated entirely to creating JWT’s and doing initial logins
  + Code on demand (Optional)
    - Optionally a RESTful web service could send back executable code as the response.
  + Uniform interface.
    - Uri routes are used to define resources
      * /books/50
    - Verbs are used to define the action you are performing
      * Get
      * Post
      * Put
      * Delete
    - HATEOAS
      * Hypermedia As the engine of the application state
      * A resource should send back links to related resources.
    - Requests are independent (self contained)
      * You do not need to set up an HTTP call with another one.
      * Every request has all the information necessary to be processed.