RESTFUL API

REPRESENTATIONAL STATE TRANSFER

REST is Resource based. Things versus actions. Common to use HTTP. But can use other protocols as well such as FTP, SMTP, SNMP.

- Uniform Interface.
- Stateless
- Cacheable
- Client – Server
- Layered System
- Code on Demand

HTTP VERBS

- GET
- PUT
- POST
- DELETE

REST CALLS

- URI's are resource names
- HTTP Response (Status and Body)

STATELESS

- Server contains no client state.
- Each request contains enough context to process the message.
- Any session state is held on the client. Once you have a framework, it's easy to figure out.

CLIENT SERVER

- Assume a disconnected system
- Separation of concerns. Client does not connect directly to the resources but via the provider of the service.
- Uniform interface is the link between the two.

CACHEABLE

- Server Reponses are cacheable
  - Implicit – client decides
  - Explicit – server tells client how long to cache.
  - Negotiated

LAYERED SYSTEM

- Client can’t assume direct connection to server.
- Details hidden. Separation of concerns.
- Improves Scalability with caching, layered architecture etc.

CODE ON DEMAND

- Server can extend client.
- Optional constraint.

ADVANTAGES

- Scalable
- Simple
- Modifiable
- Visible
- Portable
- Reliable

HTTP METHODS

HTTP METHOD	CRUD
GET	read
POST	create
PUT	update/replace
PATCH	partial update
DELETE	delete

HTTP CODES

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

EXERCISE TIME!

- Download postman if you haven't already from https://www.postman.com/