API Authentication

* None, usually limits #times API is used in a timeframe
* Basic, provide username & password when request is made to API
  + Usually done by passing base64 encoded string in the header of the request
* Key, a string, kind of like a password
  + Authorization, allows you to use API
  + Authentication, be identified as a user to the API
  + Added to Header of request / query Param
  + More secure, even though a request can be intercepted and used by others, they will not be able to get a hold of your acc details
* Token, use logs in->token is gen->use token to interact with API
  + Usually you (3rd party) gets user to login -> user gives token to you ->you use token to interact with API

RESTful API

1. Uses standard HTTP methods, GET, POST, PUT, PATCH, DELETE
2. Has a standard data format that it responds with (JSON, XML)

* Representational part of Representational state transfer
* Resources are represented in a specific format that is sent as a response to the client

1. Clients and Servers are completely separate

* Able to msg each other over a network (req, res)
* Allows each side to scale independently from each other, evolve separately, built by diff people
* RESTful API allows the whole system to scale easily

1. Stateless, each req from client to server should contain all info that’s needed to understand and process the req

* Server shouldn’t be storing any client side, state or data between requests
* Each request, response is complete, without needing to know what happened prev
  + Each req, res has all the info needed
* Allows better scalability and simplifies server-side implementation

1. Resource-based, centred around resources, uses a unique resource ID, a Universal Resource Locator/ID (URL)

* URL, address for a particular resource in the API

API Notes

* an interface that two computer systems use to exchange information securely over the internet
* Endpoint, a route on the API provider server, its like a function of the API
  + https://bored-api.appbrewery.com/random
* Query Param, name of param and its value
  + “?” after the endpoint
  + https://x.com/filter?query=value
  + Multiple Params: xxxx&xxxx
    - <https://x.com/EP?q1=val&q2=val>
* Path Param, identifies a resource in the API
  + Each resource has a unique key/ID
  + Works same way as endpoints, have to check docs of API to use

JSON(JavaScript Object Notation)

* JS object written shorter, each key & value is a “” string
* Lacks a object name, enclosed by { }
* {“doors: 2”, “drawers”: 2, “colour”: red}
* To turn JS Object into JSON
  + const jsonData = JSON.stringify(object);
* To turn JSON into JS Object
  + const data = JSON.parse(jsonObject)
* need to define data(value) with a key to access on .ejs
* JS:

app.get("/", (req, res) => {

  res.render("index.ejs", {recipe: data});

});

* EJS:

<% if(locals.recipe) {%>

<h2>

<%=recipe.name%>

Axios

* npm package to make a request to an api endpoint to get information back/render a page, etc
* when getting JSON data, returns it as as JS obj
* await, render a page and get data

app.get('/get-stock', async (req, res) => {

try {

  //get the array of JSON objects

  const response = await axios.get("link");

  //give the array a name

  const stockList = response.data;

//get user input, user query as this is a .get

 const stockSymbol = req.query.sym;

 //search for a specific object in the array

 const foundStock = stockList.find(stock => stock.symbol === stockSymbol.toUpperCase());

* to access params of in endpoint of URL:

req.params.paramName

* to define a function: async function name(){}