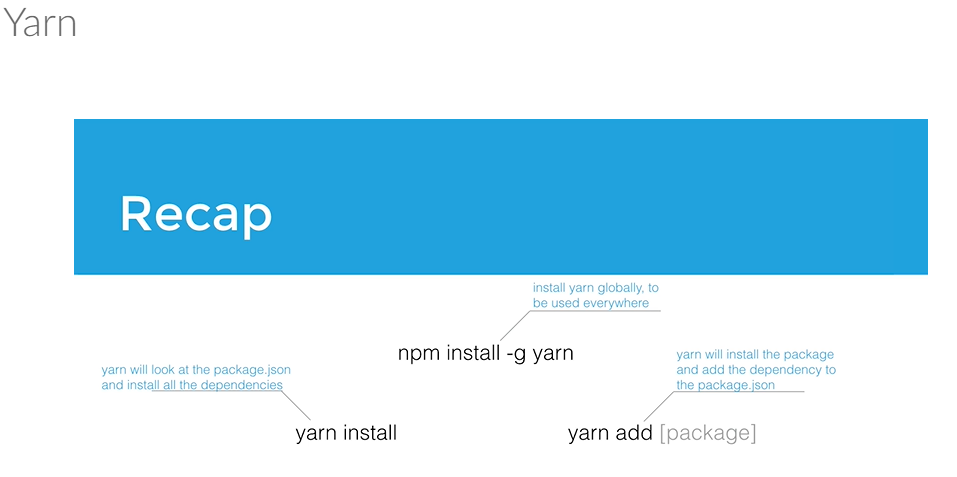
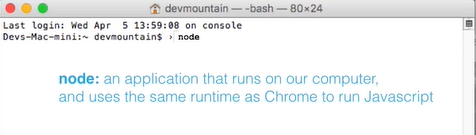
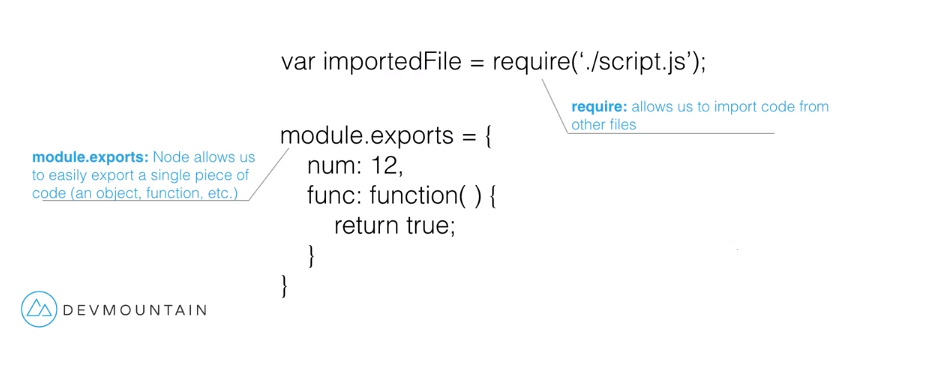
YARN

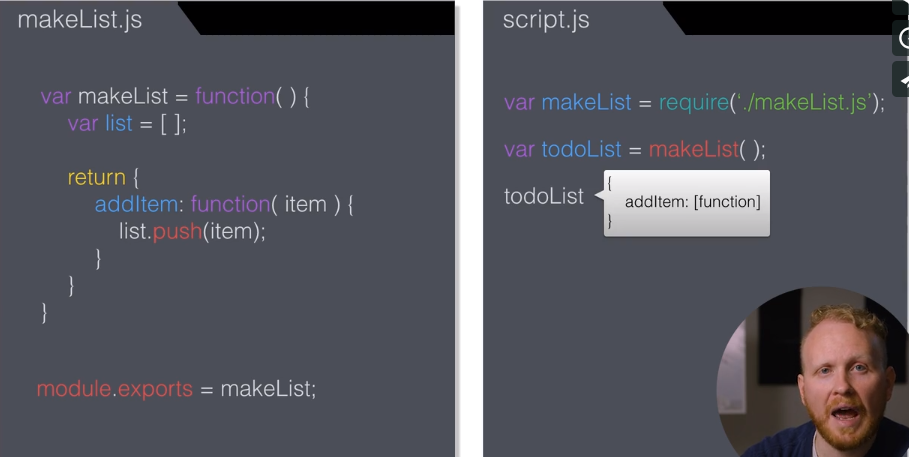


NODE:



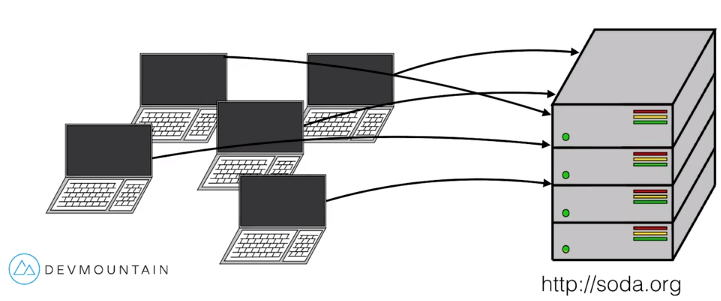
Node Modules:



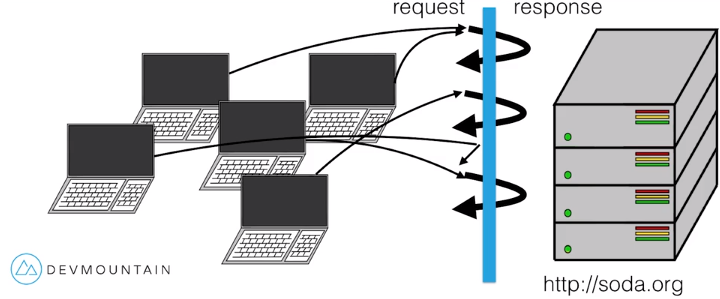


ENDPOINTS:

* Points in our code that we have defined to listen for user requests.
  + For example: We have many computers trying to ask for things from our Server.
  + It is a bad idea to just let them in to mess around with whatever they wanted.

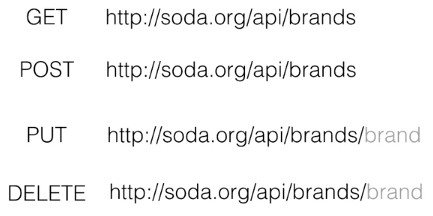
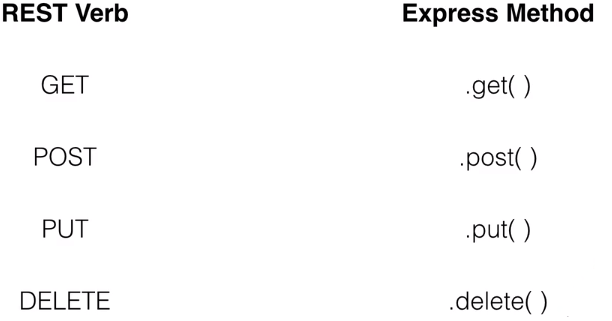


* To fix this we create a layer, often referred to or used as an API portion of our server.
  + The API’s job is to receive the requests, put them in little boxes, and send the answers back out to the requesting computers.
  + A bad request to the API will not get into our code, or receive a package/response back. Requests can only go where we want them to thanks to the API.



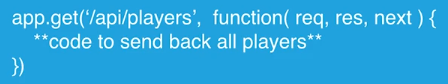
API

* We can create RESTFUL endpoints to REQUEST, ADD, EDIT\UPDATE, or REMOVE data. (GET, POST, PUT, DELETE).
  + The REST verbs are used with matching methods.

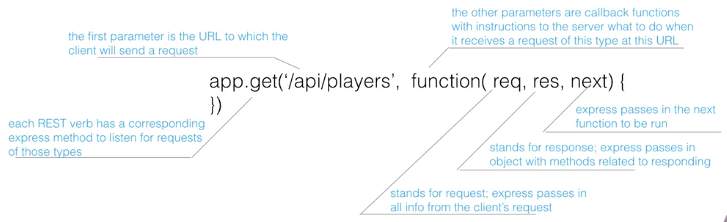
USING REST METHODS:

* We use the EXPRESS APP from and the verb our API is listening for in React to write the endpoint.
  + We are at **BASEBALL.NET**, we are connected to the internet, are running our server, and told the web that if anyone looks for BASEBALL.NET, come to our server, they are looking for the ROOT url.
* **app.get(‘/api/players’)** = Once a request is made, the server will narrow the request down to this to GET the *requested* information from its location root and *respond.* The user will need to type out the full URL <http://baseball.net/api/players>
  + **We don’t know when the users are going to make this request or how many at once. So we need to give it a callback function with the correct code inside to get the information and send it back.**



* + **req:** 1st parameter representing the **request** information that the users sent us when they started the request.
  + **res:** 2nd parameter represents the **response** of information we will be sending back to the user. Express is putting the information on this variable and sending ‘res’ back to the user.
  + **next:** 3rd parameter is a function that will automatically force code to go to the **next** function in the chain, and is used to bring multiple functions and middleware into together.
* app.post, app.put, app.delete are formatted the same way, but perform different actions.

BASIC ENDPOINT STRUCTURE:



SIMPLE SERVER SET UP:

