Practical Exercise

Based on the demos, particularly fetchDemo1, and previous lab exercises, create a web API to serve up jokes using static web server middleware and the front end based on what you designed in a previous exercise. The joke type is selected from the dropdown menu based on the types in the requirements below.

Use the jokes file kindly made available by David Katz in jokes.json which I've given you as the data source.

Requirements:

Using the provided json file as your data source, create the following API GET endpoints:

/jokes This needs a path parameter as described below. Without one it should return code 400

/type can be **general | programming | dad | knock-knock | any** Where | means **or**.

For example, **/jokes/dad** will return dad jokes whereas **/jokes/any** will return no specific type

Query: **count=n** An optional query parameter where n is the number of jokes you want to return. If it is not provided, one joke should be returned.

If count = 1 then return one **random** joke of the selected type. Write the setup, then a 2 second delay, then the punchline to the web page.

If count is greater than one, you should return that number of jokes of the specified type and output all their attributes in a table.

You need to account for the user asking for more jokes than there are available.

/types this will request all joke types in the data source to be returned from the server. These will be used to populate the types dropdown list

Don't worry for this exercise about testing for stupid inputs or case sensitivity as this just takes time and you will have spent a lot of time on this in your programming modules. However, be aware that you must take testing seriously and account for all eventualities – and prove it for an assignment. In this case, just prove that all requirements are satisfied if you enter sensible values

The following is an example. Don't spend ages making it look nice, it's the concepts that are important

As an optional improvement, return the joke in a **non-repeating random order** so the same joke isn't displayed more than once.

The dropdown menu displaying the available joke types should be populated by calling the types API GET endpoint called /types and added to the menu dynamically using JavaScript so if a new type is added to the list of jokes, your list would be updated on the next request and available for selection. This provides good maintainability in that the code doesn't need to change if the data does.


Joke Central
Choose a joke type: Dad
Number of jokes: 1
Get Joke
Why do fathers take an extra pair of socks when they go golfing?
In case they get a hole in one!


Joke Central
Choose a joke type: Knock Knock
Number of jokes: 100
Get Joke
ID Type Setup ID
12 knock-knock Knock knock. Who's there? A broken pencil. A broken pencil who? Never mind. It's pointless.
13 knock-knock Knock knock. Who's there? Cows go. Cows go who? No, cows go moo.
34 knock-knock Knock knock. Who's there? Opportunity. That is impossible. Opportunity doesn't come knocking twice!
14 knock-knock Knock knock. Who's there? Little old lady. Little old lady who? I didn't know you could yodel!
61 knock-knock Knock knock. Who's there? Hatch. Hatch who? Bless you!