Construction of User Interfaces : COMS 3190

11/6/24

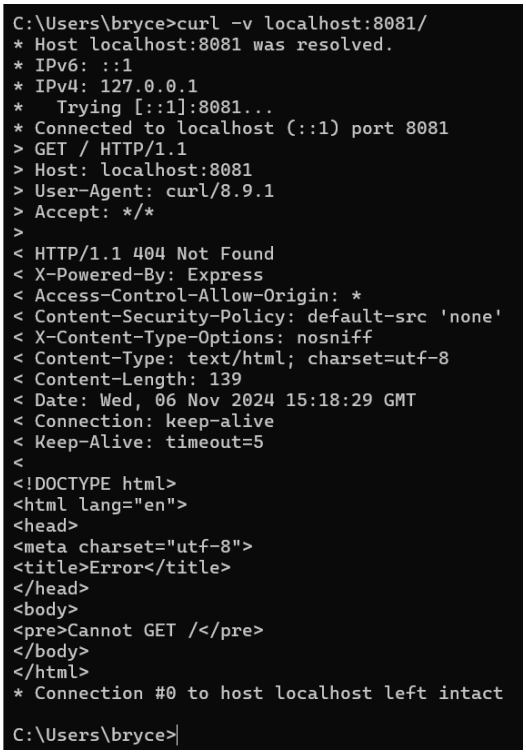
Bryce Jensenius

brycejen@iastate.edu

REST applications is an architectural style of connecting and exchanging information between different systems through the internet. They are stateless meaning everything needed must be contained within a request. There is no context around it so you just send a request and get a response and that is it.

An API is rules that must be followed for communicating between different software systems. It works as the gateway between the client and the server on the web. They are specific URL endpoints you define to perform specific actions within an application. They can be GET, POST, PUT, or DELETE. You send a request and then receive a response with information on how the request went with a code as well as any information needed for a get request.

This command returned a result code of 404 indicating that the GET request could not be found/complete. Because the connection of this specific API endpoint could not be found, the connection couldn't be made. Specifically, we were trying to get the endpoint of just / but this API endpoint is not set up in our program. We only have a get request setup for /listRobots so the request goes to an endpoint that isn't setup on our app.



The top portion of this curl request indicates information about the request itself. It indicates the host it is trying to communicate with which is localhost:8081 since our program is running on port 8081 locally. It gives the unique IP address for the local computer to communicate with, which is 127.0.0.1. More specifically for the request, it indicates it was a GET, trying to retrieve information. Also it was to endpoint /listRobots within our program and sent through version 1.1 of HTTP. This is shown in the line GET /listRobots HTTP/1.1.

The bottom portion of what is returned indicates the response of the request. In this case the response had a status code of 200 indicating that it went OK, successful. The body returned is at the bottom an is the JSON objects representing the data we requested. The content type is in the header and indicates the type of media returned. In this case it was simply text, JSON text. HTTP/1.1 was used meaning the protocol version 1.1 of HTTP.

← Address communicated with

← Indicator this was a GET request, with HTTP version

← Response code and Protocol

← Type of content returned

← Body of JSON data text returned

This is the request with the status code changed to 404, Not Found. Even though the request does return the data, we told the program to return 404. This is meant to indicate that something was not found while processing the request, so even with not successful status codes, you can still return content.

```
C:\Users\bryce>curl -v localhost:8081/listRobots
* Host localhost:8081 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:8081...
* Connected to localhost (::1) port 8081
> GET /listRobots HTTP/1.1
> Host: localhost:8081
> User-Agent: curl/8.9.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 404 Not Found
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Type: text/html; charset=utf-8
< Content-Length: 531
< ETag: W/"213-+DfUaChmBi6qwTYadUimQmO1pg4"
< Date: Wed, 06 Nov 2024 18:35:40 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
[
    {
    "id": 1,
    "name": "Robot 1",
    "price":24.50,
    "description": "This is a description of Robot 1",
    "imageUrl": "https://robohash.org/robot1"
    },
    {
    "id": 2,
    "name": "Robot 2",
    "price":32.90,
    "description": "This is a description of Robot 2",
    "imageUrl": "https://robohash.org/robot2"
    },
    {
    "id": 3,
    "name": "Robot 3",
    "price":20.50,
    "description": "This is a description of Robot 3",
    "imageUrl": "https://robohash.org/robot3"
    }
]* Connection #0 to host localhost left intact
```

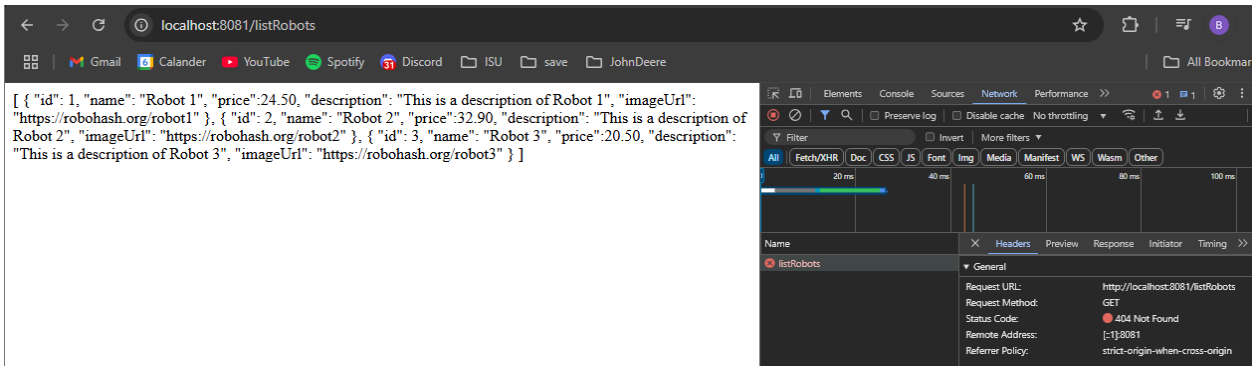
← Same Get Request

← Response Code of 404 Not Found as we set the program to do

← JSON text content still returned

Changing Status code to 404 and using Web Browser, you can see in network how the response is actually processed. As before, the data is still returned but a 404 Not found code is retrieved. In the network section, the request to listRobots is in red indicating some kind of

failure. Additionally, to the right you can see the actual status code, 404 Not Found indication the reason it was unsuccessful.



Returning HTML Element from the curl request

```
C:\Users\bryce>curl -v localhost:8081/
* Host localhost:8081 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:8081...
* Connected to localhost (::1) port 8081
> GET / HTTP/1.1
> Host: localhost:8081
> User-Agent: curl/8.9.1
> Accept: */*
>
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Type: text/html; charset=utf-8
< Content-Length: 94
< ETag: W/"5e-+H4LCicvu5dBtmvvThMcK3gAJHs"
< Date: Wed, 06 Nov 2024 18:57:59 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
<h1 style='color:Green; background-color:
    black;border: 0px; '>Hello World From Node </h1>* Connection #0 to host localhost left intact

C:\Users\bryce>
```

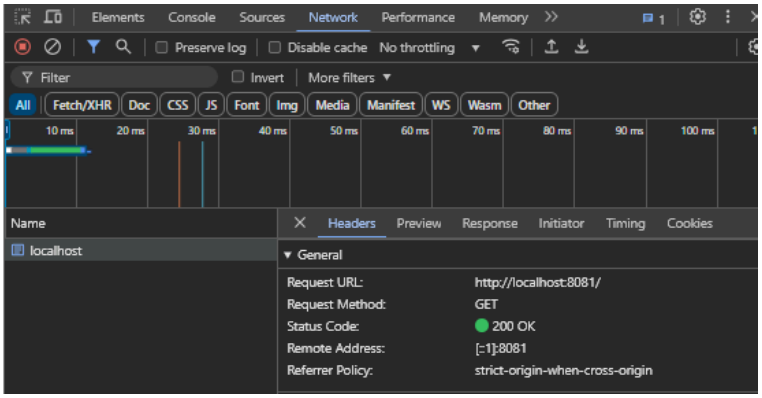
← Request to localHost

← A Get Request to '/' endpoint

← OK Status code 200 returned

← Text representing an HTML h1 element returned in the body

Returning HTML Element Through web browser. The web browser sends a GET request to the endpoint of just / which we set up to return an HTML header element. The status code shown on the right for this request is successful, 200, or OK. This means it retrieves this HTML and can render it on screen since it knows how to process HTML unlike the terminal. On the left you can se the HTML h1 element returned by the get requests response body.



After adding additional code for the /person endpoint to get a Json Data response we get this. Request to get persons JSON object through the console still has a successful 200 status code.

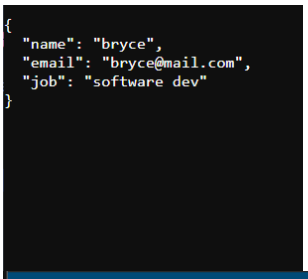
```
C:\Users\bryce>curl -v localhost:8081/person
* Host localhost:8081 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:8081...
* Connected to localhost (::1) port 8081
> GET /person HTTP/1.1
> Host: localhost:8081
> User-Agent: curl/8.9.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Type: application/json; charset=utf-8
< Content-Length: 60
< ETag: W/"3c-BeqiMqtijU9zNSyXx+uf+ZbBKAc"
< Date: Wed, 06 Nov 2024 20:26:35 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
{"name":"alex","email":"alex@mail.com","job":"software dev"}* Connection #0 to host localho
st left intact

C:\Users\bryce>
```

←Sent get request to /person endpoint

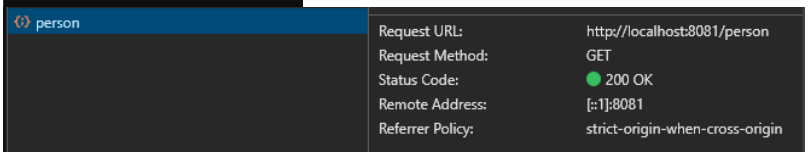
←Received successful code of 200
←ContentType this time is returned as a json object

←The actual object returned, in json string to represent each of the persons entries

This is the image of the same request but instead in the web browser. It had a successful status code as expected. The Json object was correctly returned and then the result displayed on screen.



← Json Object returned by the request with each of the values



← Endpoint sent to /person endpoint
← Get request
← Successful status code of 200