

REST stands for Representational State Transfer. It uses HTTP to transfer data over the internet, from one program to another. It uses unique URLs and commands like GET, POST, PUT, and DELETE to designate the action of the receive program. The programs communicating can range from servers to websites to databases and more.

API stands for Application Program Interface. It is a set of guidelines governing communication between two pieces of software. These are critical to make sure behavior is intended on both ends. It describes both the behavior of the request and the response. An API is usually created to allow outside programs to communicate with it and request some data or the result of some process.

```
Last login: Mon Apr 21 12:53:45 on ttys000
petercollins@Peters-MacBook-Pro Activity_12_Apr_24 % curl -v http://localhost:8080
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
* Trying [::1]:8080...
* Connected to localhost (::1) port 8080
> GET / HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.7.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 404 Not Found
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Security-Policy: default-src 'none'
< X-Content-Type-Options: nosniff
< Content-Type: text/html; charset=utf-8
< Content-Length: 139
< Date: Thu, 24 Apr 2025 20:57:18 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>Cannot GET /</pre>
</body>
</html>
* Connection #0 to host localhost left intact
petercollins@Peters-MacBook-Pro Activity_12_Apr_24 %
```

The connection could not be made because there is no endpoint to hit. The program accepts requests at <http://localhost:8080/listRobots> and the part after the slash is very

important. There is nothing at the URL we hit initially because we didn't have anything running there.

```
* Connection #0 to host localhost left intact
petercollins@Peters-MacBook-Pro Activity_12_Apr_24 % curl -v http://localhost:8080/list
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
* Trying [::1]:8080...
* Connected to localhost (::1) port 8080
> GET /listRobots HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.7.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Type: text/html; charset=utf-8
< Content-Length: 498
< ETag: W/"1f2-aRXw+ZTqKd91IWuHkAN4BE6QcFo"
< Date: Thu, 24 Apr 2025 21:00:54 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
[
  {
    "id": 1,
    "name": "Robot 1",
    "price": 24.5,
    "description": "This is a description of Robot 1",
    "imageUrl": "https://robohash.org/robot1"
  },
  {
    "id": 2,
    "name": "Robot 2",
    "price": 32.9,
    "description": "This is a description of Robot 2",
    "imageUrl": "https://robohash.org/robot2"
  },
  {
    "id": 3,
    "name": "Robot 3",
    "price": 20.5,
    "description": "This is a description of Robot 3",
    "imageUrl": "https://robohash.org/robot3"
  }
]
* Connection #0 to host localhost left intact
petercollins@Peters-MacBook-Pro Activity_12_Apr_24 %
```

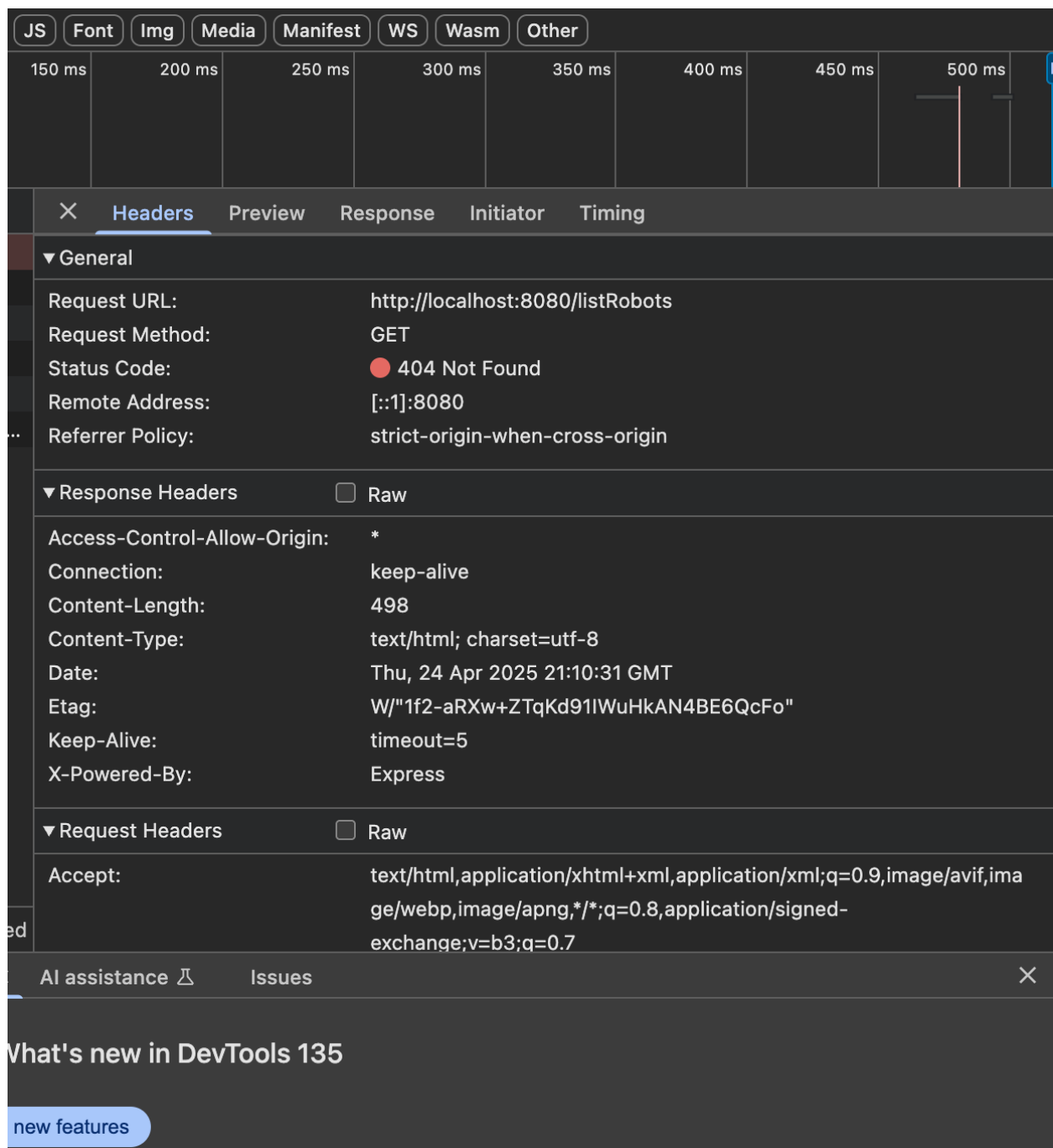
The request is everything above the line “Request completely sent off” and the response is the rest. The request includes the method, the IP and port, and who sent it. The response contains the JSON, the status code, the headers, and more. The JSON is the core of what the response is, called the body. This curl worked because we do have an endpoint at the URL we hit. Our program is designed to respond to hits at this URL.

```

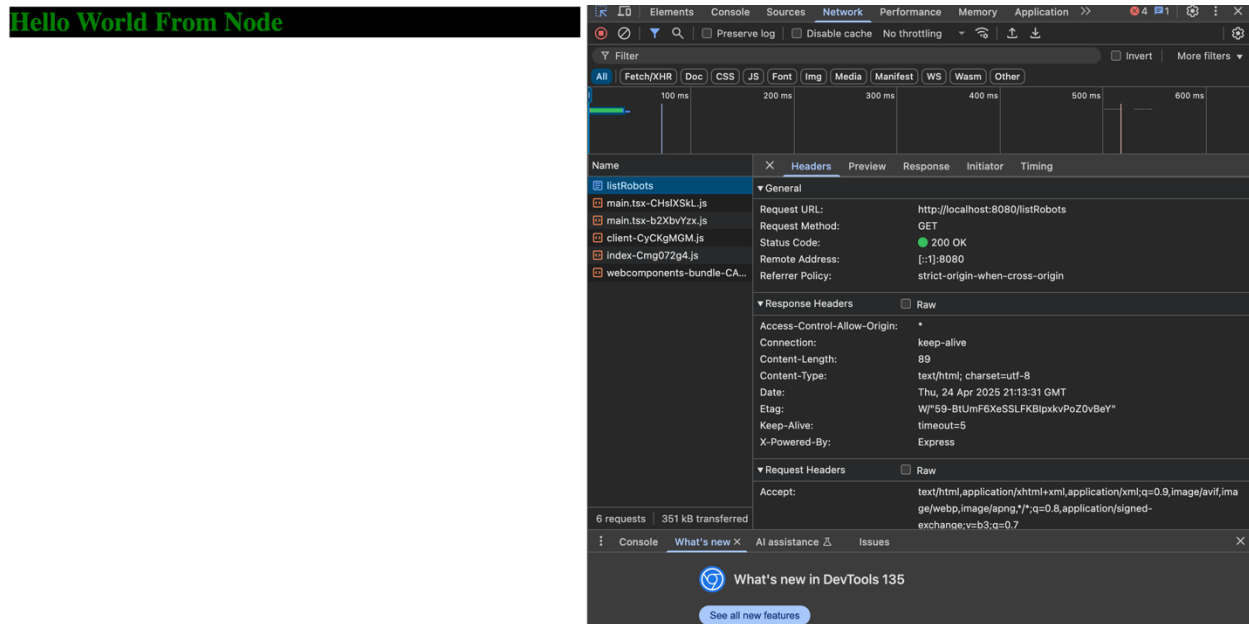
* Connection #0 to host localhost left intact
petercollins@Peters-MacBook-Pro Activity_12_Apr_24 % curl -v http://localhost:8080/list
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
* Trying [::1]:8080...
* Connected to localhost (::1) port 8080
> GET /listRobots HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.7.1
> 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: 498
< ETag: W/"1f2-aRXw+ZTqKd91IWuHkAN4BE6QcFo"
< Date: Thu, 24 Apr 2025 21:04:39 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
[
  {
    "id": 1,
    "name": "Robot 1",
    "price": 24.5,
    "description": "This is a description of Robot 1",
    "imageUrl": "https://robohash.org/robot1"
  },
  {
    "id": 2,
    "name": "Robot 2",
    "price": 32.9,
    "description": "This is a description of Robot 2",
    "imageUrl": "https://robohash.org/robot2"
  },
  {
    "id": 3,
    "name": "Robot 3",
    "price": 20.5,
    "description": "This is a description of Robot 3",
    "imageUrl": "https://robohash.org/robot3"
  }
]
* Connection #0 to host localhost left intact
petercollins@Peters-MacBook-Pro Activity_12_Apr_24 %

```

Everything here is the same except the status code. We manually changed it to respond to this request with 404. Each HTTP status code has a specific meaning, and 404 is “not found” so this is generally reserved for when things go badly. All 400 levels and above generally mean something bad.



This is another example of the response for a request. The red circle indicates that this is a 400 or higher status code, meaning something went wrong (in most programs). However, since this is intended, this is correct behavior for the program.



Now we have changed the response and the status code and this is reflected in the response. The HTML has loaded and created a website and we now get a 200 OK status code, meaning everything went fine on our end.

```
petercollins@Peters-MacBook-Pro Activity_12_Apr_24 % curl -v http://localhost:8080/person
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
* Trying [::1]:8080...
* Connected to localhost (::1) port 8080
> GET /person HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.7.1
> 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: Thu, 24 Apr 2025 21:15:26 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
* Connection #0 to host localhost left intact
{"name":"alex","email":"alex@mail.com","job":"software dev"}%
petercollins@Peters-MacBook-Pro Activity_12_Apr_24 %
```

You can see at the bottom that we got an object back. It looks similar to the JSON response because that was an array of objects. Since our program responds with that at the URL <http://localhost:8080/person>, that is what happens when it receives a request.