GET and POST requests

Web Dev, Spring 2021

Last time

We introduced the server-side of web development

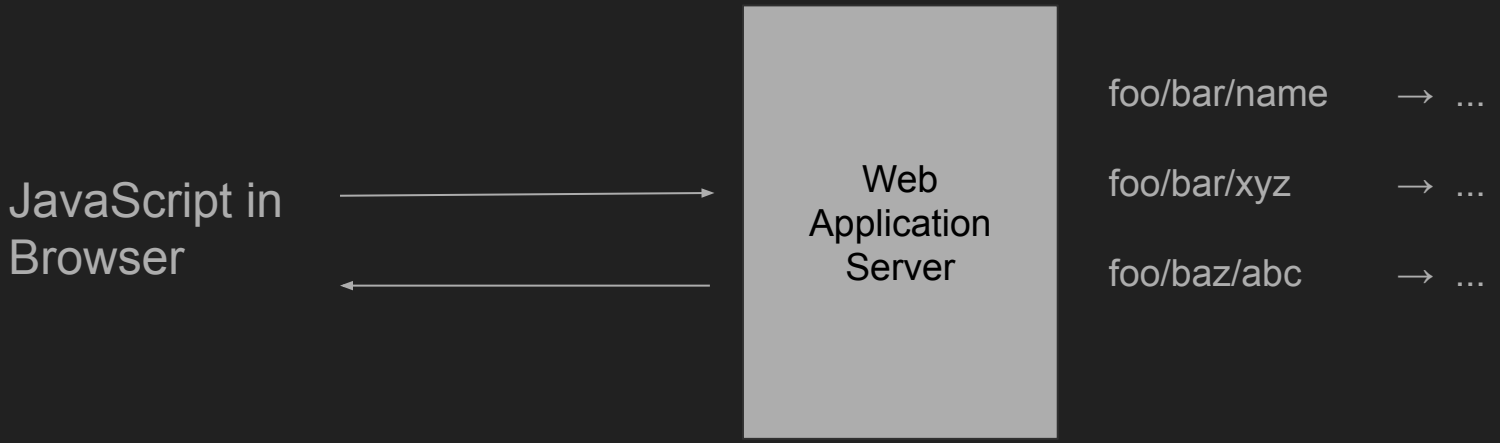
- Web Application Servers (aka servers)

Idea:

- associate with each route a function to call that creates the response to the request corresponding to the route
- we focused on GET requests

What can we do in a Web Application Server? Anything we want, really

Web Application Server



You can think of it as JavaScript making function calls to the server

- more expensive than normal function calls though
- why? server has more storage, shared between clients, survives refreshes...

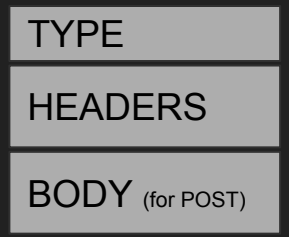
HTTP requests and responses

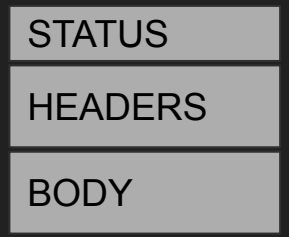
HTTP is a text-based protocol

- GET requests
  - used for requests that don't change the server state
  - minimal argument passing (query parameters in the route)
  - no body
  - what a browser sends from the browser bar
- POST requests
  - used for requests that change the server state
  - can pass "arbitrary" data in the body

Responses:

- status [200 = OK, 3xx / 4xx / 5xx = errors]
- response data in the body





GET requests

```
GET /index.html HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.feedbacknow.com
Connection: Keep-Alive
```

POST requests

```
POST /api/settime HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.feedbacknow.com
Content-Type: application/json
Content-Length: 34
Connection: Keep-Alive

{"DateTime":"2021-03-03 20:12:34"}
```

POST requests

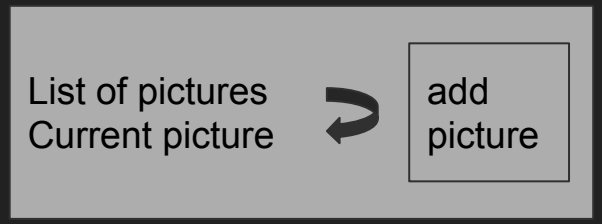
```
POST /api/settime HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.feedbacknow.com
Content-Type: application/json
Content-Length: 34
Connection: Keep-Alive

{"DateTime":"2021-03-03 20:12:34"}
```

MIME type —
describes the content of the body

Long catalog of MIME types

Our running demo



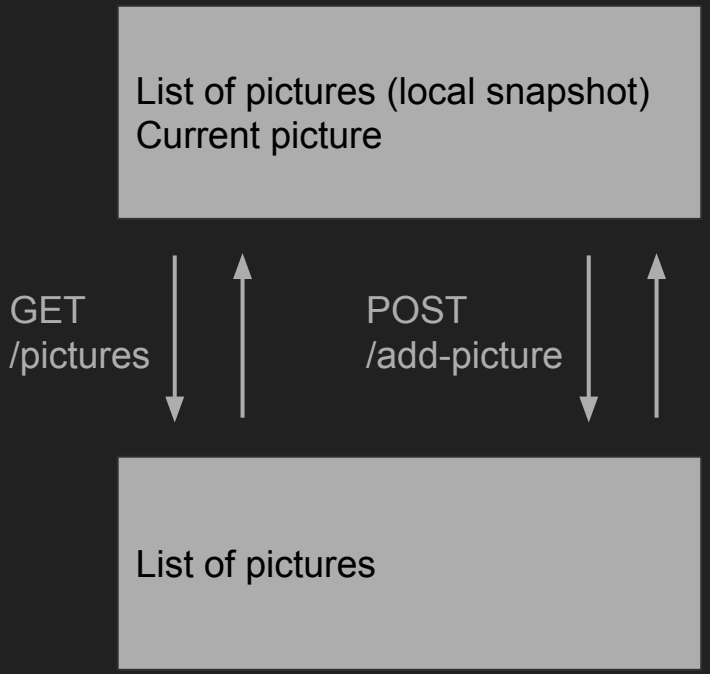
Maintain a list of pictures in the document (browser)

Can add a picture to the browser list

Any picture added is lost on browser refresh

What if we wanted to keep pictures across refreshes or over time?

Our running demo



One solution — keep the list of pictures on the server

When document loads, fetch a copy of the list from the server

When adding a picture, send it to the server

Demo

Pay attention to:

- actions `fetchPictures` and `addPictures`
- how JavaScript gets data back from the `/pictures` endpoint
- how JavaScript sends a POST request to the `/add-pictures` endpoint with a JSON body
- how Flask gets the JSON body out of the request

Distraction: where do you get the frontend?

Right now — we're opening the HTML document from the file system

Better to make it available via a URL

Who delivers it?

- Another web server
- The web application server itself