Homework #1

CSE312

Homework Topics

- Handling Request, Response
- Router Class
- Hosting Static Files
- Handout Code

Homework Topics

Request class

```
class Request:

    def __init__(self, request: bytes):
        # TODO: parse the bytes of the request and populate the following instance variables

        self.body = b""
        self.method = ""
        self.path = ""
        self.http_version = ""
        self.headers = {}
        self.cookies = {}
```

```
def test1():
    request = Request(b'GET / HTTP/1.1\r\nHost: localhost:8080\r\nConnection: keep-alive\r\n\r\n')
    assert request.method == "GET"
    assert "Host" in request.headers
    assert request.headers["Host"] == "localhost:8080"  # note: The leading space in the header val
    assert request.body == b""  # There is no body for this request.
```

Homework Topics

Request Class
Self.body

POST /api/chats HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Content-Length: 18
Cookie: id=123; theme=dark
Origin: http://localhost:8080

{"content":"asdf"}

```
lass Request:

    def __init__(self, request: bytes):
        # TODO: parse the bytes of the re

        self.body = b""
        self.method = ""
        self.path = ""
        self.http_version = ""
        self.headers = {}
        self.cookies = {}
```

Homework Topics

Request Class
Self.body

“POST /api/chats HTTP/1.1\r\n
Host: localhost:8080\r\n
Content-Type: application/json\r\n
Content-Length: 18\r\n
Cookie: id=123; theme=dark\r\n
Origin: http://localhost:8080\r\n
\r\n
{"content":"asdf"}”

-Request as a single string, would be single line if wasn’t on slide
-Notice “\r\n\r\n” between headers and body

```
lass Request:

    def __init__(self, request: bytes):
        # TODO: parse the bytes of the re

        self.body = b""
        self.method = ""
        self.path = ""
        self.http_version = ""
        self.headers = {}
        self.cookies = {}
```

Homework Topics

Request Class
Self.method

POST /api/chats HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Content-Length: 18
Cookie: id=123; theme=dark
Origin: http://localhost:8080

{"content":"asdf"}

```
lass Request:

    def __init__(self, request: bytes):
        # TODO: parse the bytes of the re

        self.body = b""
        self.method = ""
        self.path = ""
        self.http_version = ""
        self.headers = {}
        self.cookies = {}
```

Homework Topics

Request Class
Self.path

POST /api/chats HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Content-Length: 18
Cookie: id=123; theme=dark
Origin: http://localhost:8080

{"content":"asdf"}

```
lass Request:

    def __init__(self, request: bytes):
        # TODO: parse the bytes of the re

        self.body = b""
        self.method = ""
        self.path = ""
        self.http_version = ""
        self.headers = {}
        self.cookies = {}
```

Homework Topics

Request Class
Self.http_version

POST /api/chats HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Content-Length: 18
Cookie: id=123; theme=dark
Origin: http://localhost:8080

{"content":"asdf"}

```
lass Request:

    def __init__(self, request: bytes):
        # TODO: parse the bytes of the re

        self.body = b""
        self.method = ""
        self.path = ""
        self.http_version = ""
        self.headers = {}
        self.cookies = {}
```

Homework Topics

Request Class
Self.headers

POST /api/chats HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Content-Length: 18
Cookie: id=123; theme=dark
Origin: http://localhost:8080

{"content":"asdf"}

```
lass Request:

    def __init__(self, request: bytes):
        # TODO: parse the bytes of the re

        self.body = b""
        self.method = ""
        self.path = ""
        self.http_version = ""
        self.headers = {}
        self.cookies = {}
```

Homework Topics

Request Class
Self.cookies

POST /api/chats HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Content-Length: 18
Cookie: id=123; theme=dark
Origin: http://localhost:8080

{"content":"asdf"}

-Notice that Cookies are Also a Header

```
lass Request:

    def __init__(self, request: bytes):
        # TODO: parse the bytes of the re

        self.body = b""
        self.method = ""
        self.path = ""
        self.http_version = ""
        self.headers = {}
        self.cookies = {}
```

Homework Topics

Request Class

Show Request handout code

Homework Topics

Response Class

- Helps you construct a response to send to client
- You will construct responses at every endpoint you have, this makes your code and life easier

```
class Response:
    def __init__(self):
        pass

    def set_status(self, code, text):
        pass

    def headers(self, headers):
        pass

    def cookies(self, cookies):
        pass

    def bytes(self, data):
        pass

    def text(self, data):
        pass

    def json(self, data):
        pass

    def to_data(self):
        pass
```

Homework Topics

Response Class

__init__()

- Will be called on creation of class
- Create variables that will be used in other methods, before finally calling ***to_data()***

```
class Response:
    def __init__(self):
        pass

    def set_status(self, code, text):
        pass

    def headers(self, headers):
        pass

    def cookies(self, cookies):
        pass

    def bytes(self, data):
        pass

    def text(self, data):
        pass

    def json(self, data):
        pass

    def to_data(self):
        pass
```

Homework Topics

Response Class
set_status()

- Takes an int **(code)** and a str **(text)**
- returns self (This will be true for most of these methods. Returning a reference to the calling object allows you to chain together calls)
- Sets the status code and message for the response.
- If this method is never called, the code and message should default to **"200 OK"**

```
class Response:
    def __init__(self):
        pass

    def set_status(self, code, text):
        pass

    def headers(self, headers):
        pass

    def cookies(self, cookies):
        pass

    def bytes(self, data):
        pass

    def text(self, data):
        pass

    def json(self, data):
        pass

    def to_data(self):
        pass
```

Homework Topics

Response Class
headers()

- Takes a dict of str to str
- returns self
- Adds all the key-value pairs from the input dict as headers to the response
- If this method is called multiple times, all headers across all calls must be part of the response

```
class Response:
    def __init__(self):
        pass

    def set_status(self, code, text):
        pass

    def headers(self, headers):
        pass

    def cookies(self, cookies):
        pass

    def bytes(self, data):
        pass

    def text(self, data):
        pass

    def json(self, data):
        pass

    def to_data(self):
        pass
```

Homework Topics

Response Class
cookies()

- Takes a dict of str to str
- returns self
- Adds all the key-value pairs from the input dict as cookies to the response
- If this method is called multiple times, all cookies across all calls must be part of the response

```
class Response:
    def __init__(self):
        pass

    def set_status(self, code, text):
        pass

    def headers(self, headers):
        pass

    def cookies(self, cookies):
        pass

    def bytes(self, data):
        pass

    def text(self, data):
        pass

    def json(self, data):
        pass

    def to_data(self):
        pass
```

Homework Topics

Response Class
bytes()

- Takes bytes
- returns self
- Appends the input to the end of the body of the response
- If this method is called multiple times, all bytes must be appended to the body. This method can be combined with the text method

```
class Response:
    def __init__(self):
        pass

    def set_status(self, code, text):
        pass

    def headers(self, headers):
        pass

    def cookies(self, cookies):
        pass

    def bytes(self, data):
        pass

    def text(self, data):
        pass

    def json(self, data):
        pass

    def to_data(self):
        pass
```

Homework Topics

Response Class
text()

- Takes a str
- returns self
- Appends the input to the end of the body of the response as bytes
- If this method is called multiple times, all text must be appended to the body. This method can be combined with the bytes method (ie. calling both text and bytes should result in the text and bytes from all calls appearing in the body as bytes)

```
class Response:
    def __init__(self):
        pass

    def set_status(self, code, text):
        pass

    def headers(self, headers):
        pass

    def cookies(self, cookies):
        pass

    def bytes(self, data):
        pass

    def text(self, data):
        pass

    def json(self, data):
        pass

    def to_data(self):
        pass
```

Homework Topics

Response Class

json()

- Takes either a dict or a list
- returns self
- Set the body of the response to the input converted to json as bytes and sets the Content-Type to "application/json"
- This method can only be called once. Calling it again should replace the old body

```
class Response:
    def __init__(self):
        pass

    def set_status(self, code, text):
        pass

    def headers(self, headers):
        pass

    def cookies(self, cookies):
        pass

    def bytes(self, data):
        pass

    def text(self, data):
        pass

    def json(self, data):
        pass

    def to_data(self):
        pass
```

Homework Topics

Response Class
to_data(self)

- Does not take any parameters
- returns the entire response in bytes. This is the final response that will be sent to the client over the TCP socket
- The returned bytes must be properly formatted according the HTTP protocol and must contain all headers, cookies, the status code and message, and the body of the response along with the Content-Length header

```
class Response:
    def __init__(self):
        pass

    def set_status(self, code, text):
        pass

    def headers(self, headers):
        pass

    def cookies(self, cookies):
        pass

    def bytes(self, data):
        pass

    def text(self, data):
        pass

    def json(self, data):
        pass

    def to_data(self):
        pass
```

Homework Topics

Response Class

Show Response handout code

Homework Topics

TCP Handler Overview

```
class MyTCPHandler(socketserver.BaseRequestHandler):

    def __init__(self, request, client_address, server):
        self.router = Router()
        self.router.add_route("GET", "/hello", hello_path, True)
        # TODO: Add your routes here
        super().__init__(request, client_address, server)

    def handle(self):
        received_data = self.request.recv(2048)
        print(self.client_address)
        print("--- received data ---")
        print(received_data)
        print("--- end of data ---\n\n")
        request = Request(received_data)

        self.router.route_request(request, self)
```

Show in handout code

Homework Topics

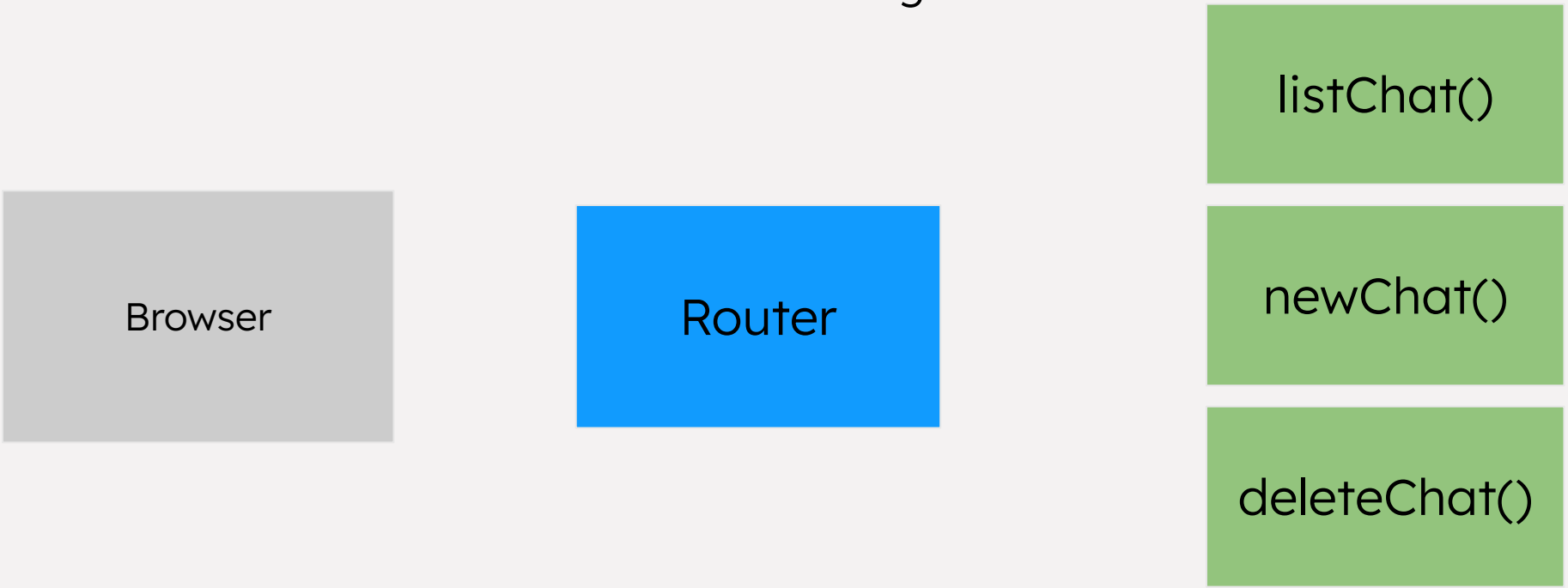
Router Class

- Allows your server to route requests
- Given a certain **method** and **path** will call a certain function (**action**)

```
class Router:

    def __init__(self):
        pass

    def add_route(self, method, path, action, exact_path=False):
        pass

    def route_request(self, request, handler):
        pass
```

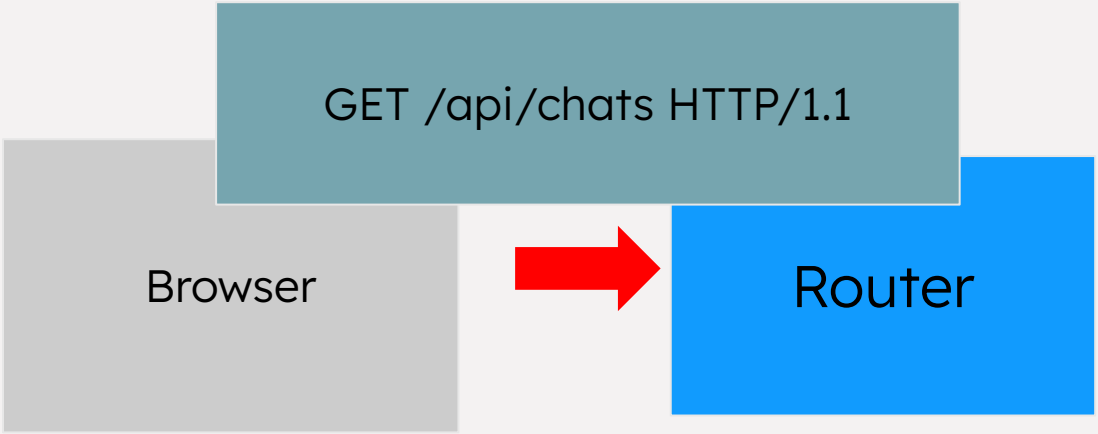
Homework Topics

Router Class - What that looks like at high level



Homework Topics

Router Class - What that looks like at high level



listChat()

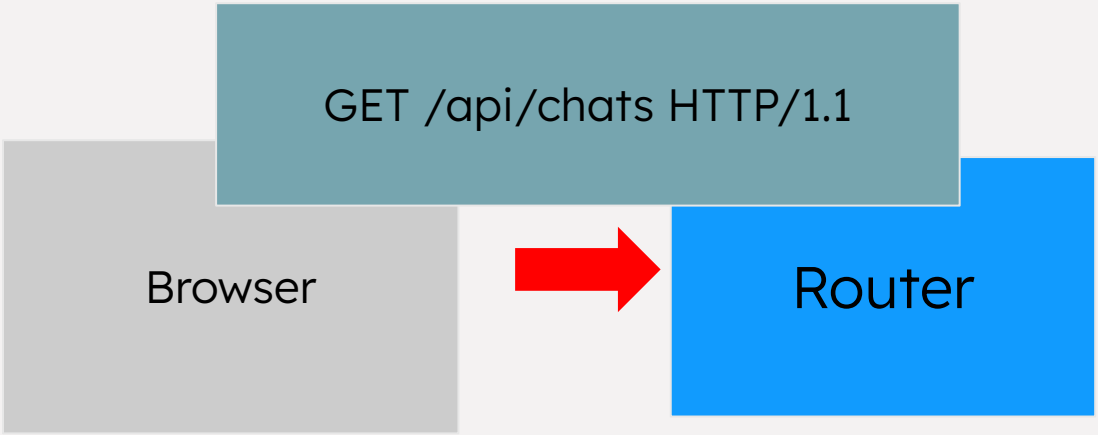
newChat()

deleteChat()

- Browser makes request to your TCP server in server.py

Homework Topics

Router Class - What that looks like at high level



listChat()

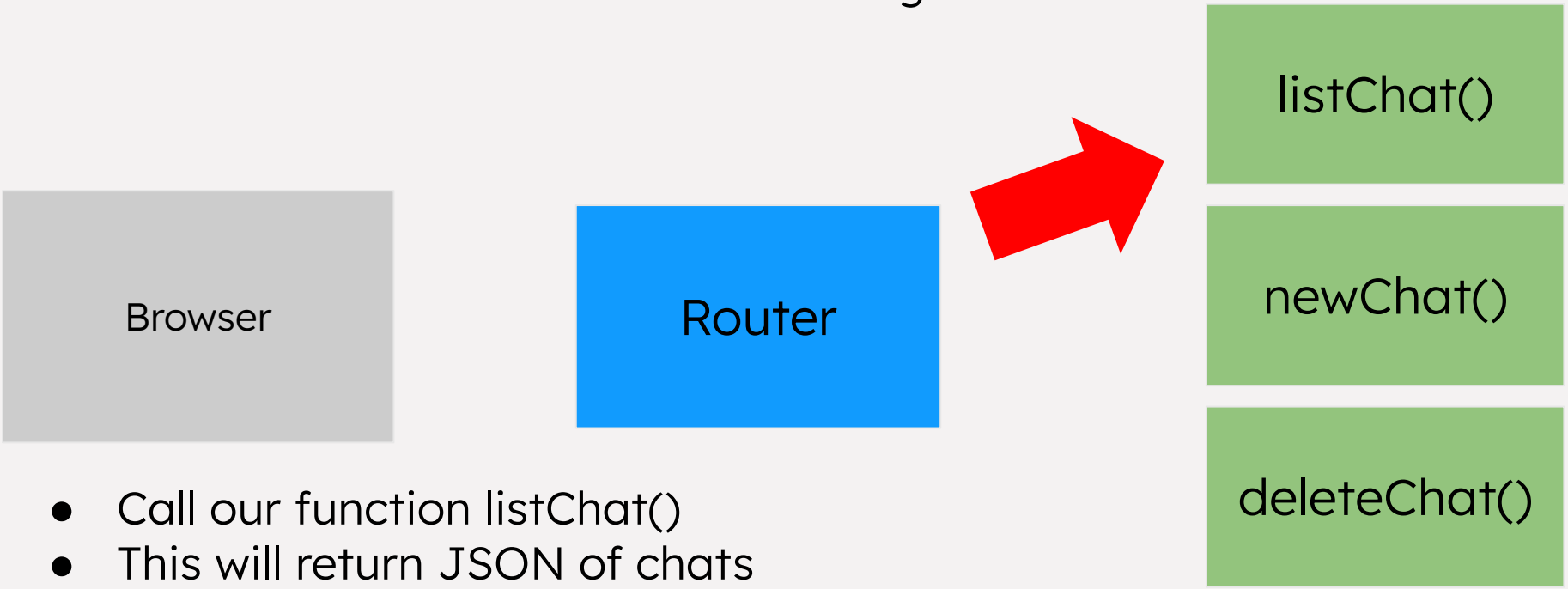
newChat()

deleteChat()

- API routes
- These aren't returning files but data from functions.
- Will have same path but different method

Homework Topics

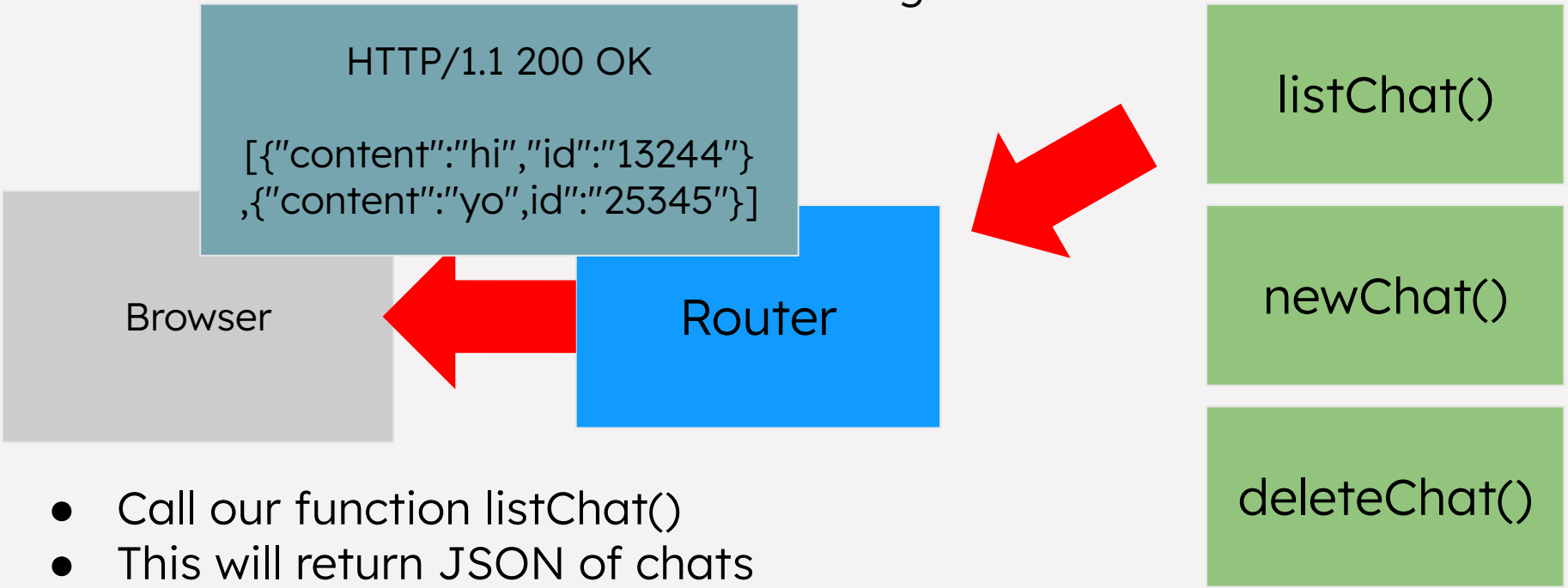
Router Class - What that looks like at high level



- Call our function listChat()
- This will return JSON of chats

Homework Topics

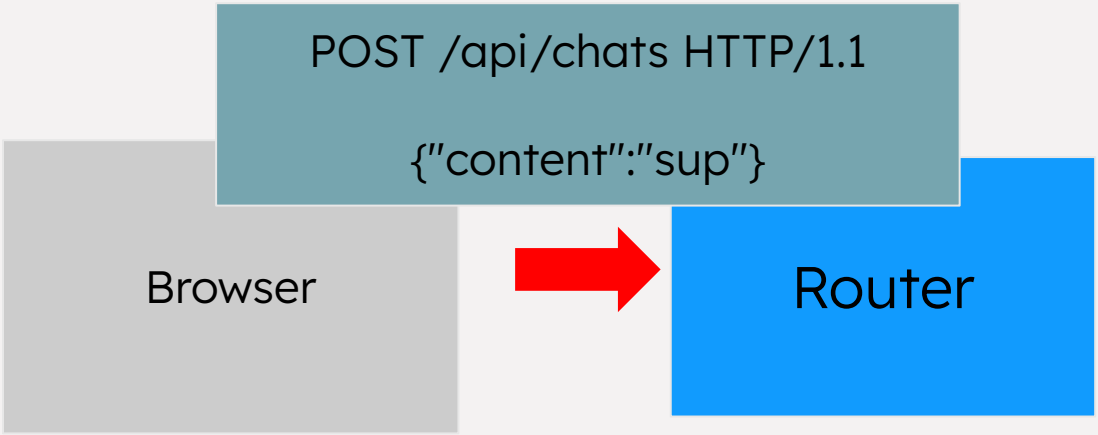
Router Class - What that looks like at high level



- Call our function listChat()
- This will return JSON of chats

Homework Topics

Router Class - What that looks like at high level



listChat()

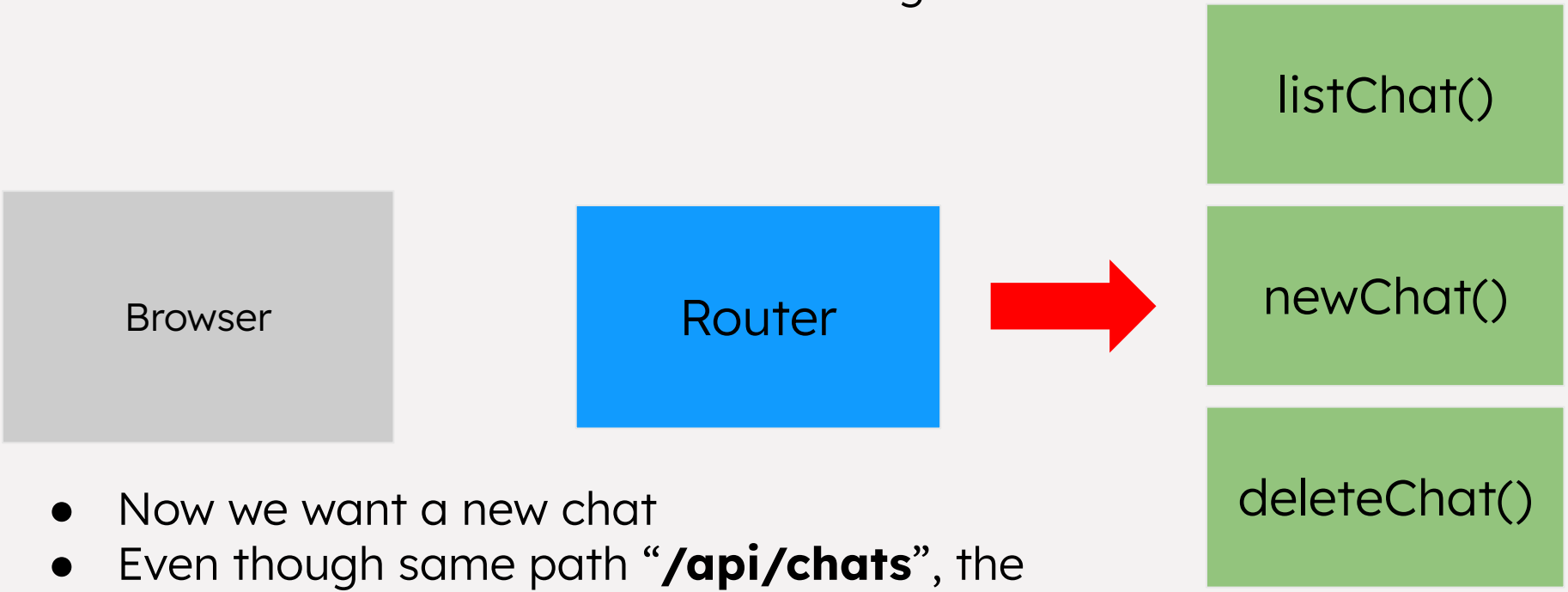
newChat()

deleteChat()

- Now we want a new chat
- Even though same path "**/api/chats**", the method is now "**POST**"
- Call correct function **newChat()**

Homework Topics

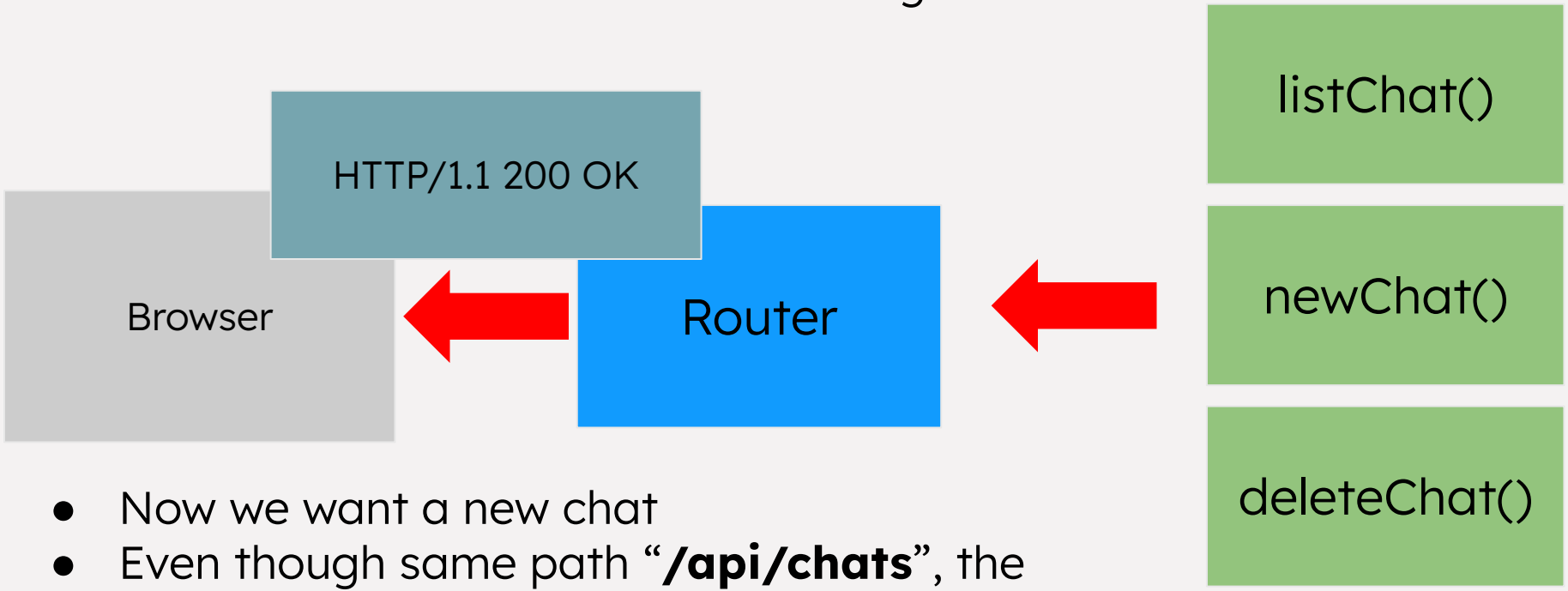
Router Class - What that looks like at high level



- Now we want a new chat
- Even though same path "**/api/chats**", the method is now "**POST**"
- Call correct function **newChat()**

Homework Topics

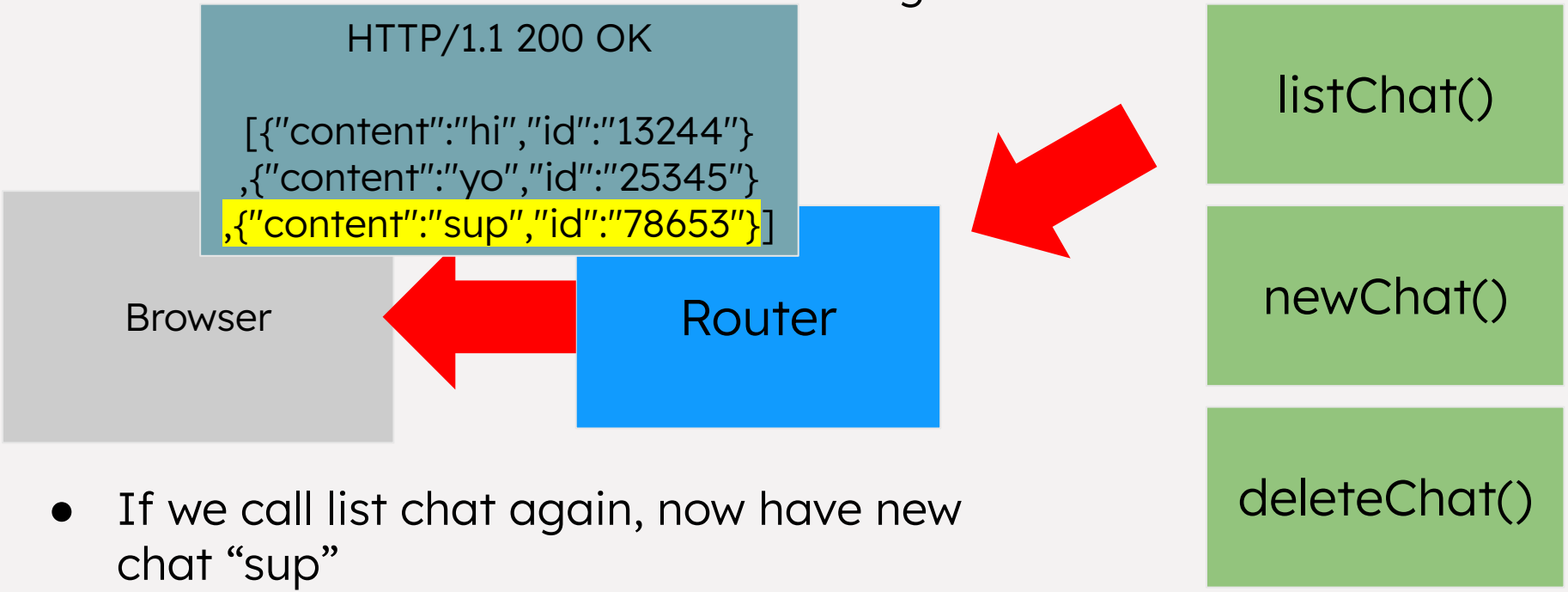
Router Class - What that looks like at high level



- Now we want a new chat
- Even though same path "**/api/chats**", the method is now "**POST**"
- Call correct function **newChat()**

Homework Topics

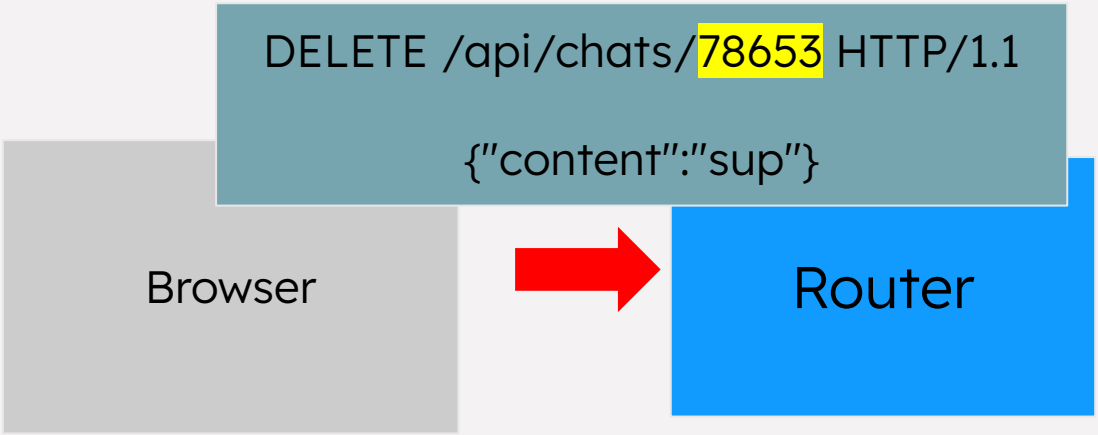
Router Class - What that looks like at high level



- If we call list chat again, now have new chat “sup”

Homework Topics

Router Class - What that looks like at high level



listChat()

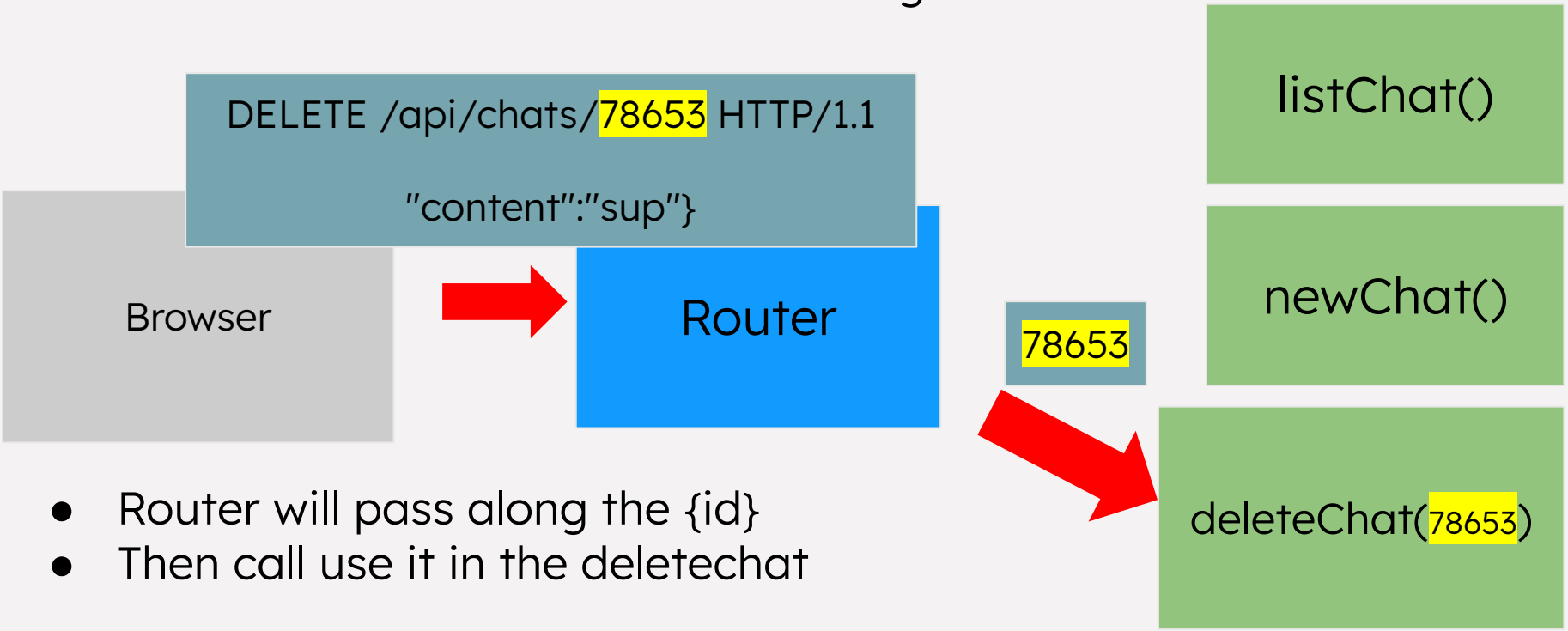
newChat()

deleteChat()

- Now we want to delete last message
- However, we need to specify ID of message
- Router needs to support any path with variable ID ”**/api/chats/{id}**”

Homework Topics

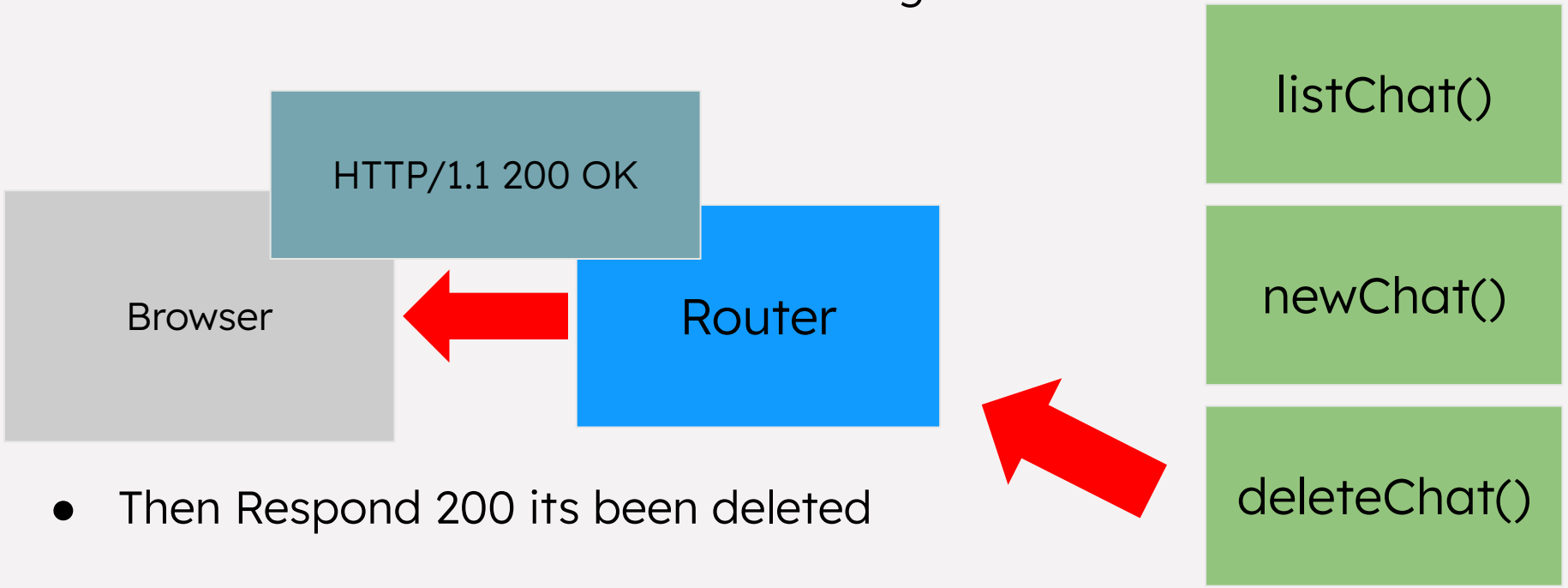
Router Class - What that looks like at high level



- Router will pass along the {id}
- Then call use it in the deletechat

Homework Topics

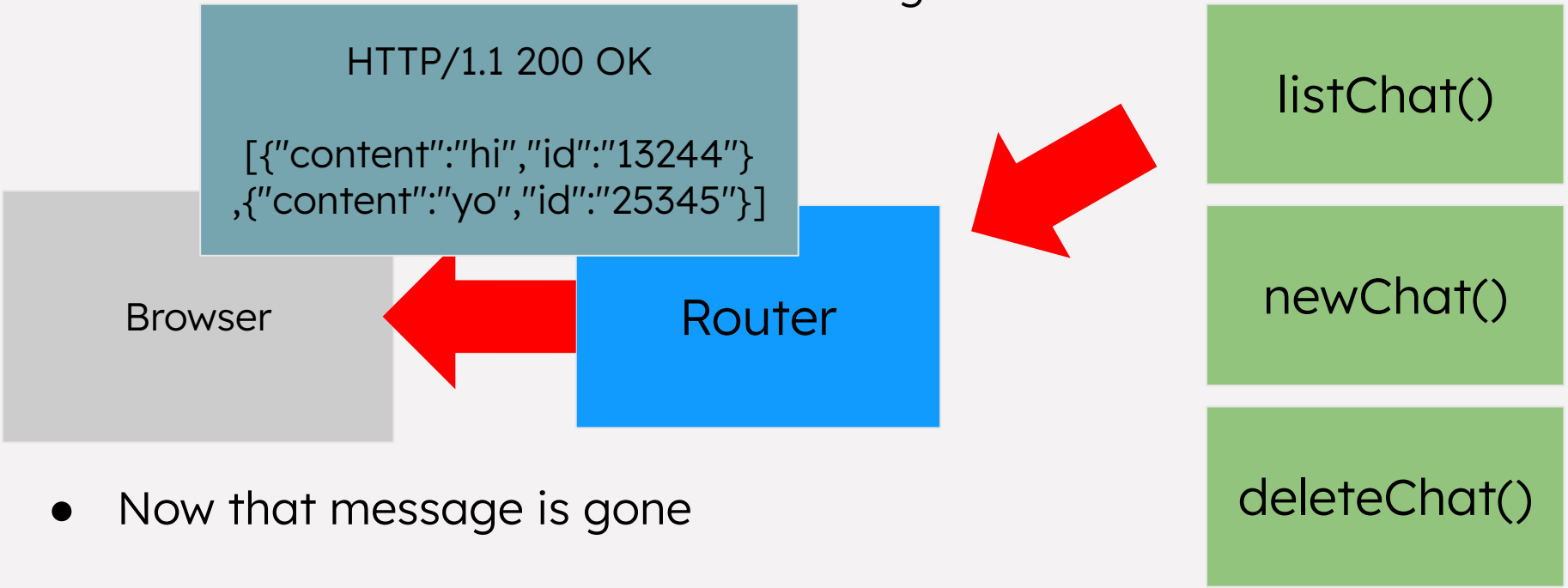
Router Class - What that looks like at high level



- Then Respond 200 its been deleted

Homework Topics

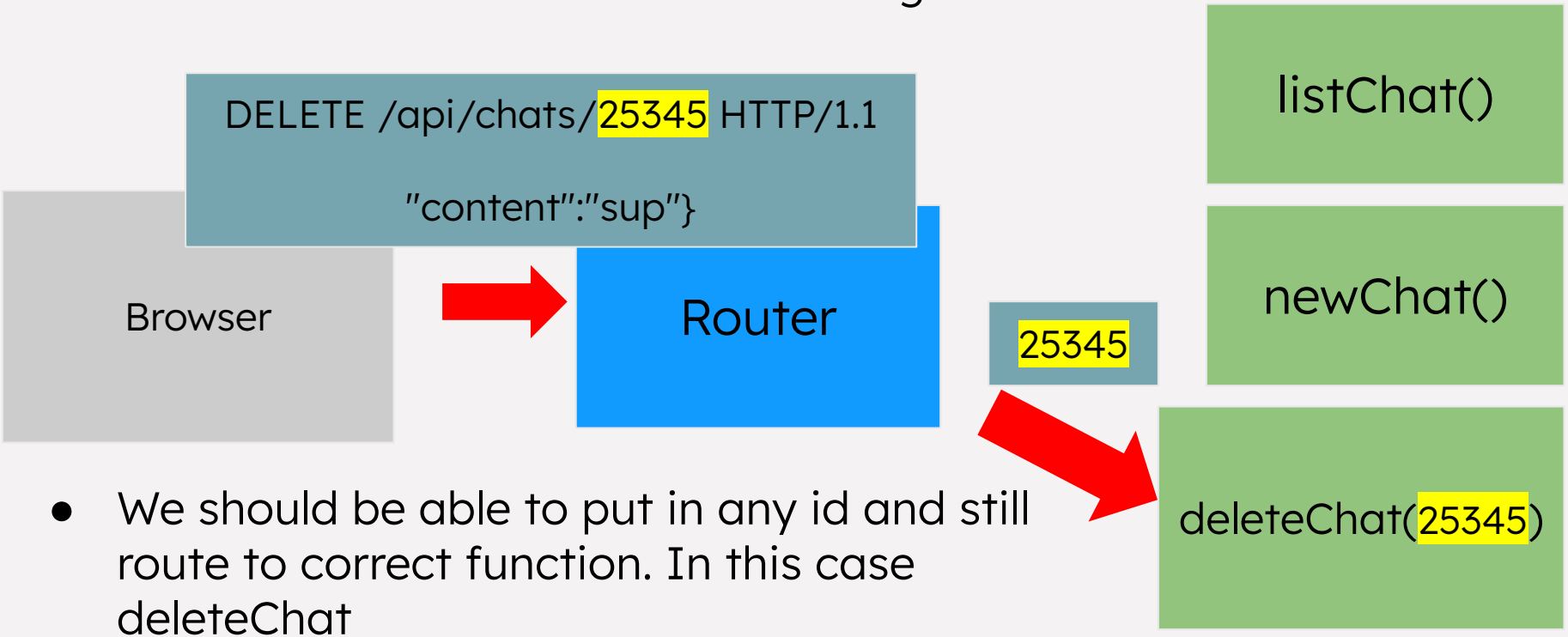
Router Class - What that looks like at high level



- Now that message is gone

Homework Topics

Router Class - What that looks like at high level



- We should be able to put in any id and still route to correct function. In this case deleteChat

Homework Topics

Router Class - What that looks like in code

Recall

```
class Router:

    def __init__(self):
        pass

    def add_route(self, method, path, action, exact_path=False):
        pass

    def route_request(self, request, handler):
        pass
```

Homework Topics

Router Class - What that looks like in code

```
class Router:

    def __init__(self):
        pass

    def add_route(self, method, path, action, exact_path=False):
        pass

    def route_request(self, request, handler):
        pass
```

```
self.router.add_route("GET", "/hello", hello_path, True)
```

Homework Topics

Router Class - What that looks like in code

```
class Router:

    def __init__(self):
        pass

    def add_route(self, method, path, action, exact_path=False):
        pass

    def route_request(self, request, handler):
        pass
```

```
self.router.add_route("GET", "/hello", hello_path, True)
```

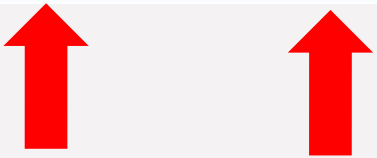
METHOD

Homework Topics

Router Class - What that looks like in code

```
class Router:

    def __init__(self):
        pass

    def add_route(self, method, path, action, exact_path=False):
        pass

    def route_request(self, request, handler):
        pass
```

```
self.router.add_route("GET", "/hello", hello_path, True)
```



METHOD

PATH

Homework Topics

Router Class - What that looks like in code

```
class Router:

    def __init__(self):
        pass

    def add_route(self, method, path, action, exact_path=False):
        pass

    def route_request(self, request, handler):
        pass
```

```
self.router.add_route("GET", "/hello", hello_path, True)
```

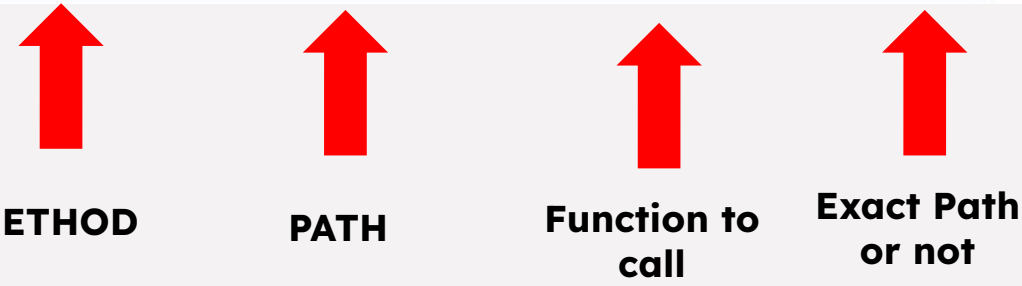
METHOD **PATH** **Function to call**

Homework Topics

Router Class - What that looks like in code

```
class Router:

    def __init__(self):
        pass

    def add_route(self, method, path, action, exact_path=False):
        pass

    def route_request(self, request, handler):
        pass
```

```
self.router.add_route("GET", "/hello", hello_path, True)
```



Homework Topics

Router Class - What that looks like in code

- Then when handling new requests
- Route request will look up function to call associated with path and method

```
def handle(self):
    received_data = self.request.recv(2048)
    print(self.client_address)
    print("--- received data ---")
    print(received_data)
    print("--- end of data ---\n\n")
    request = Request(received_data)

    self.router.route_request(request, self)
```

```
class Router:

    def __init__(self):
        pass

    def add_route(self, method, path, action, exact_path=False):
        pass

    def route_request(self, request, handler):
        pass
```

Homework Topics

Router Class - What that looks like in code

- If “/hello” is requests, call function “hello_path”

```
def hello_path(request, handler):
    res = Response()
    res.text("hello")
    handler.request.sendall(res.to_data())
```

```
class Router:

    def __init__(self):
        pass

    def add_route(self, method, path, action, exact_path=False):
        pass

    def route_request(self, request, handler):
        pass
```

Homework Topics

Router Class

Show in handout code

Homework Topics

Router Class

Quick Recap

- Depending on **method** (GET, POST … DELETE) and **Path** (“/api/chats”). Route to different functions
- Variable routes allowing variables to be passed to function /api/chats/{id}
  - /api/chats/25345
  - /api/chats/78653

Homework Topics

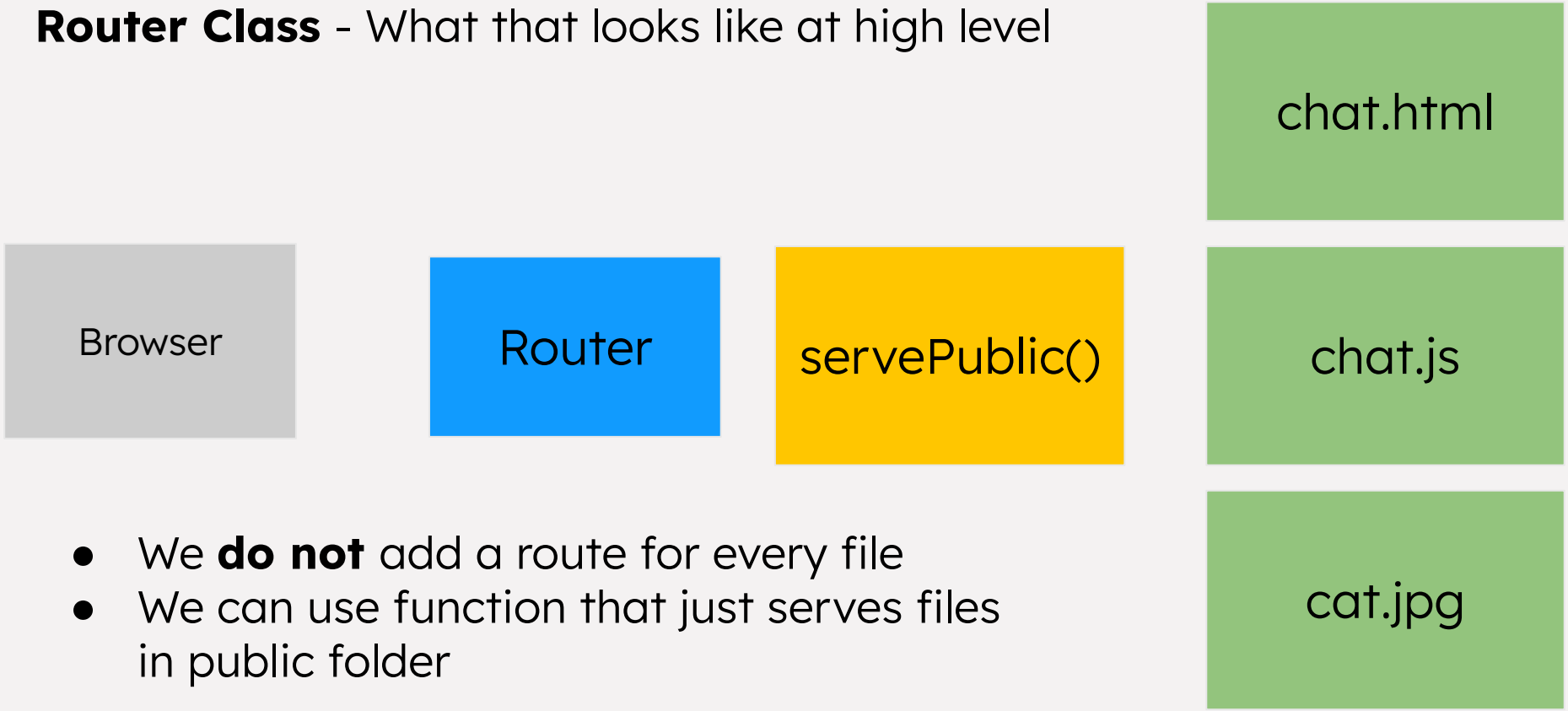
Router Class

Now what else do we need router for

- How do we handle static files?
  - Should we add a route for every file?
- What else can variable routes be used for besides chat ids?

Homework Topics

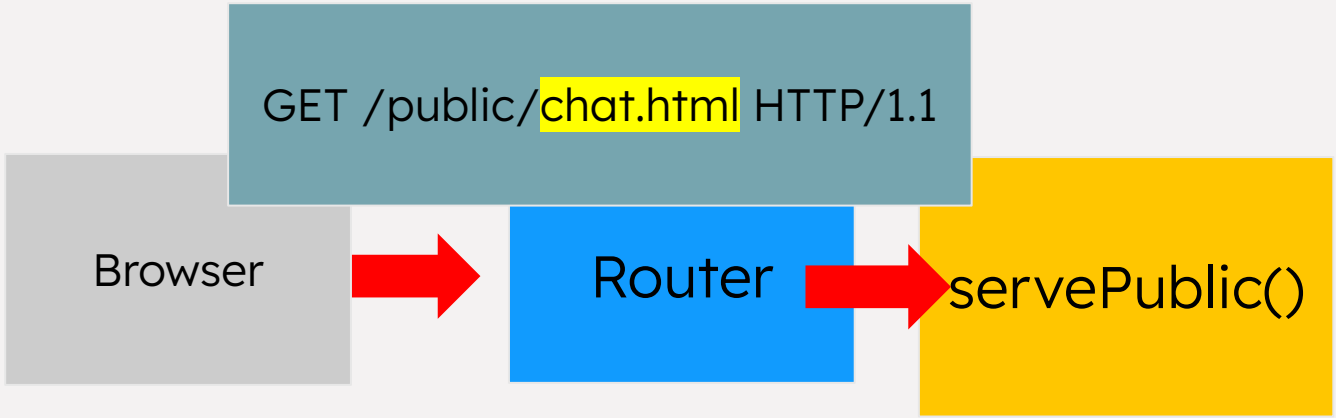
Router Class - What that looks like at high level



- We **do not** add a route for every file
- We can use function that just serves files in public folder

Homework Topics

Router Class - What that looks like at high level



chat.html

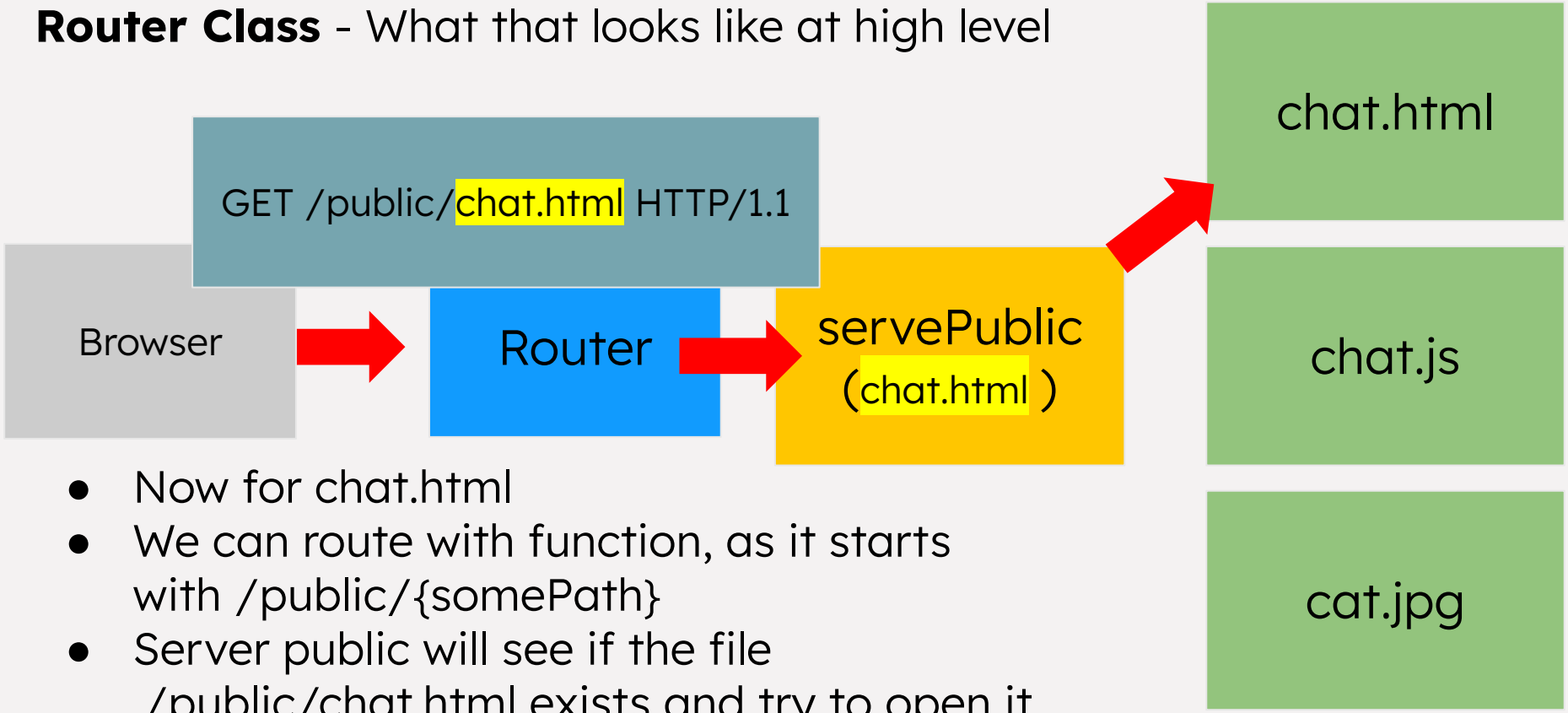
chat.js

cat.jpg

- Now for chat.html
- We can route with function, as it starts with /public/{somePath}

Homework Topics

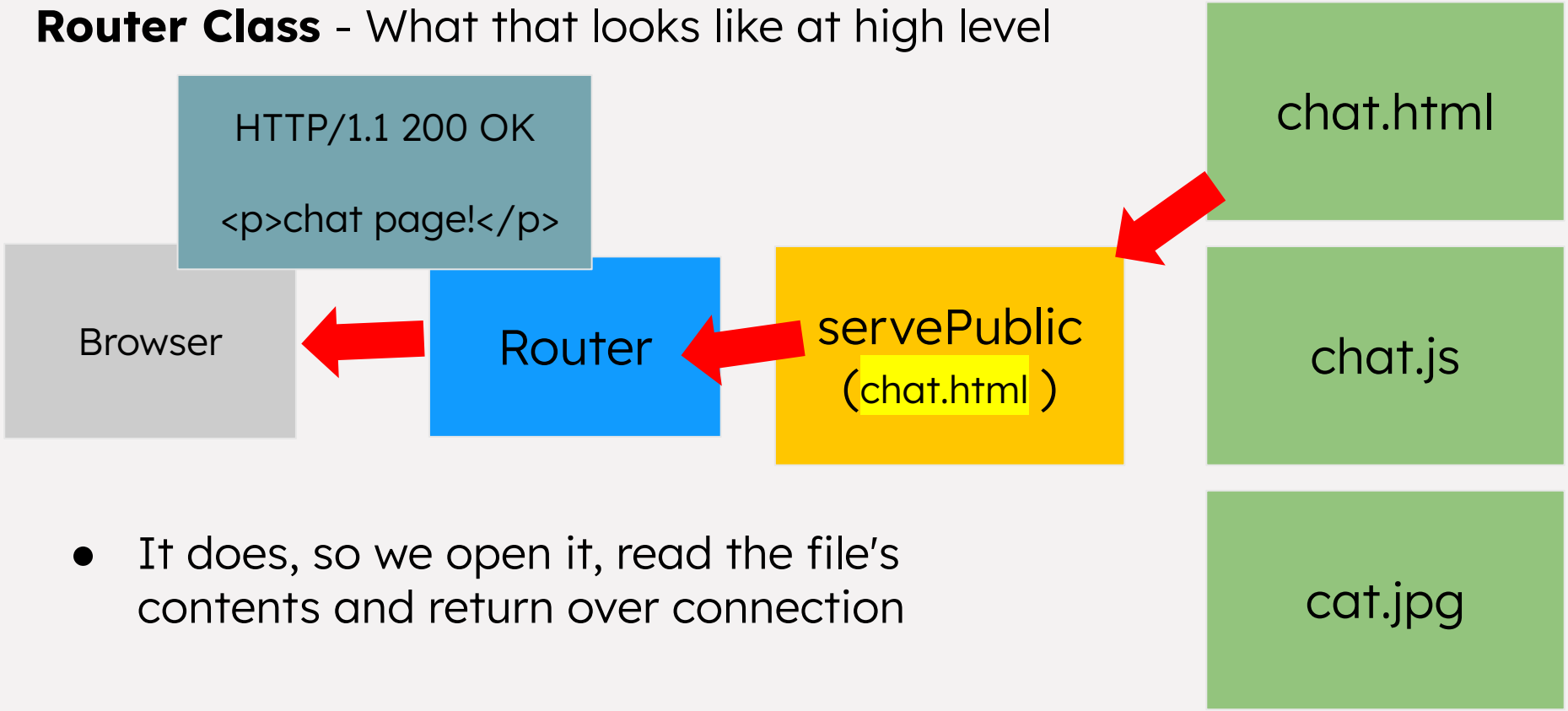
Router Class - What that looks like at high level



- Now for chat.html
- We can route with function, as it starts with /public/{somePath}
- Server public will see if the file ./public/chat.html exists and try to open it

Homework Topics

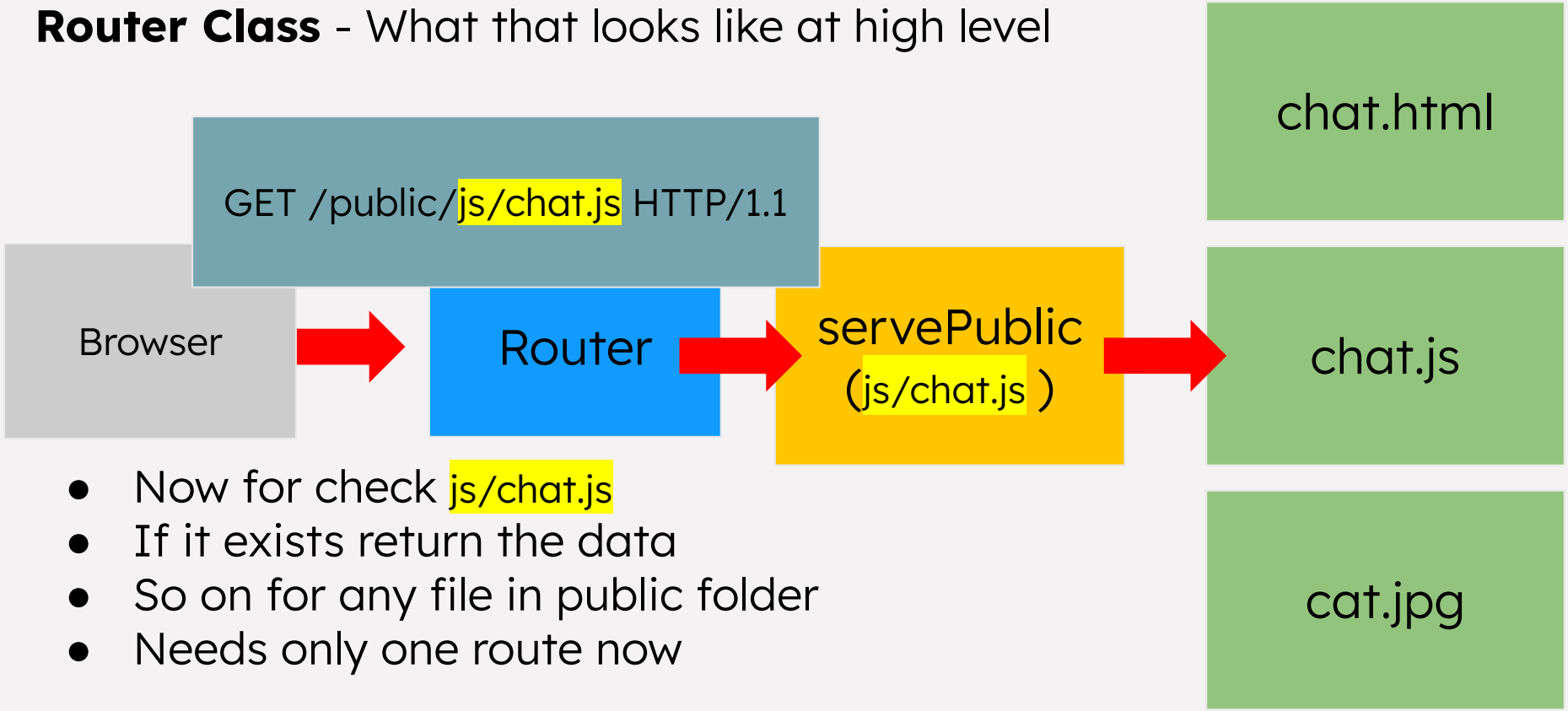
Router Class - What that looks like at high level



- It does, so we open it, read the file's contents and return over connection

Homework Topics

Router Class - What that looks like at high level



- Now for check js/chat.js
- If it exists return the data
- So on for any file in public folder
- Needs only one route now

Homework Topics

Router Class

- Depending on **method** (GET, POST … DELETE) and **Path** (“/api/chats”). Route to different functions
- Variable routes allowing variables to be passed to function /api/chats/{id}
  - /api/chats/25345
  - /api/chats/78653
- Serving files with router taking use of variable routes

Get things running

- Server receives requests when visiting http://localhost:8080/
  - Hard code response
- How to start database
  - “docker compose -f docker-compose.db-only.yml up --build”
- Demo page