

Module 2

Week 1 | Day 4 | **HTTP & the Anti-Corruption Layer**

What are we going to do tonight, Pinky?



What are we going to do tonight, Pinky?

- Explore HTTP and how requests work.

What are we going to do tonight, Pinky?

- Explore HTTP and how requests work.
- Work with 3rd party requests

What are we going to do tonight, Pinky?

- Explore HTTP and how requests work.
- Work with 3rd party requests
- Implement 3rd party them into our applications

Question of the Day.....

Question of the Day.....

Back to the request!

Back to the request!

Client

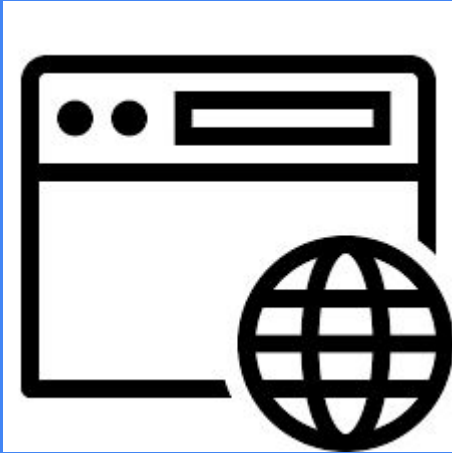


Server



Back to the request!

Client



Can I have a website please?

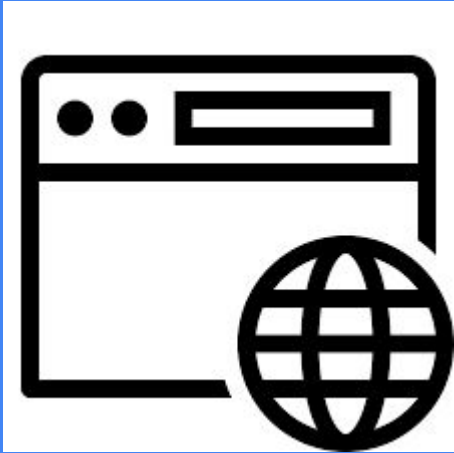


Server



Back to the request!

Client



Server



Can I have a website please?

Back to the request!

Client



Server



Now that I have it. I might as well show it!!!

Back to the request!

Client



Server



<<<<Response

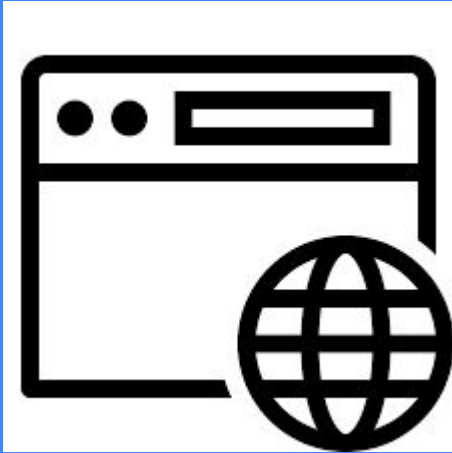


Request>>>>



Back to the request!

Client



Server



<<<<Response : HTML



Request>>>>



Back to the request!

Client



Server



<<<<Response : Document File



Request>>>>



Back to the request!

Client



Server



<<<<Response : MP3

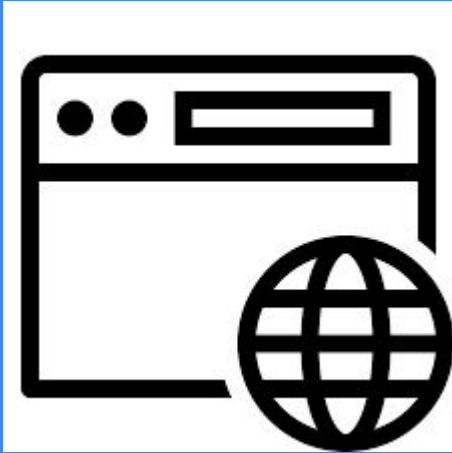


Request>>>>



Back to the request!

Client



Server



<<<<Response : Streaming Packet

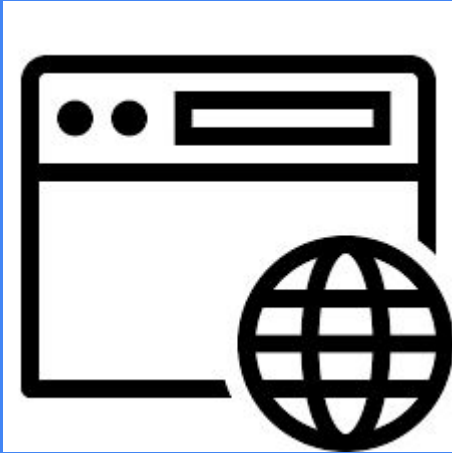


Request>>>>



Back to the request!

Client



Server



<<<<Response: JSON

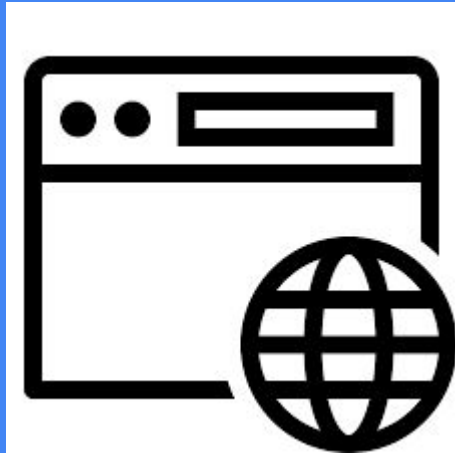


Request>>>>



Back to the request!

Client: Browser



Server



<<<<Response: JSON

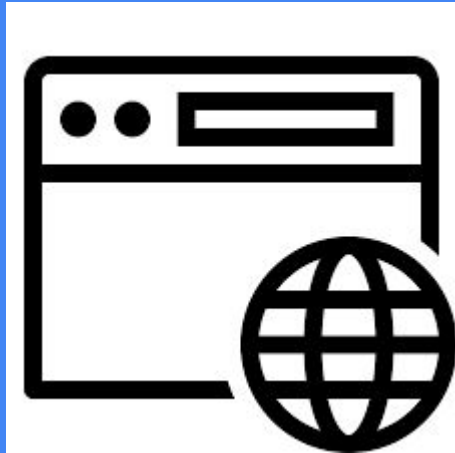


Request>>>>



Back to the request!

Client: Smart Phone/Watch



Server



<<<<Response: JSON

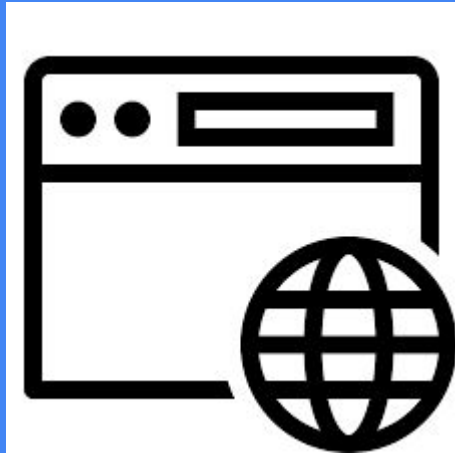


Request>>>>



Back to the request!

Client: Any Internet Aware Device



Server



<<<<Response: JSON

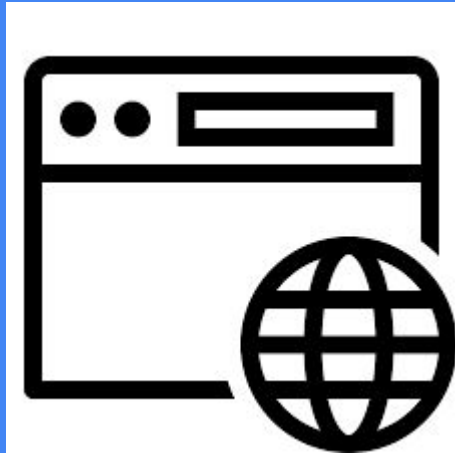


Request>>>>



Back to the request!

Client: Server



Server



<<<<Response: JSON



Request>>>>



Back to the request!

Client: API Client



Server



<<<<Response: JSON



Request>>>>



Back to the request!

Client: API Client



Server:

Computer running code out there.



<<<<Response: JSON



Request>>>>



Back to the request!

Client: API Client



Server:
Node App, Django App



<<<<Response: JSON



Request>>>>



Back to the request!

Client: API Client



Server:

(cloud) hosted code with an ip



<<<<Response: JSON



Request>>>>



Cloud Server:

AWS Availability Zone 1 : West Virginia



Cloud Server:

AWS Availability Zone 1 : West Virginia



Back to the request!

Client



Server



<<<<Response



Request>>>>

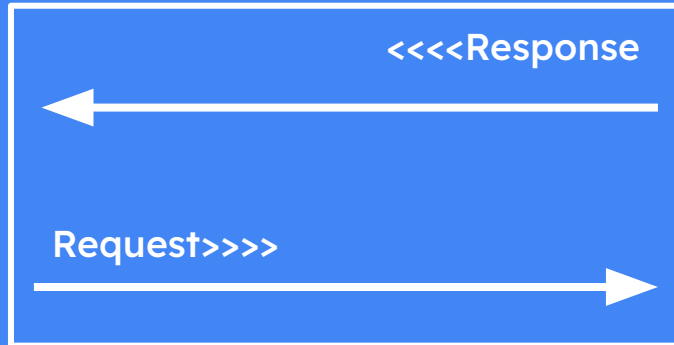


Back to the request!

Client:



HTTP



Server

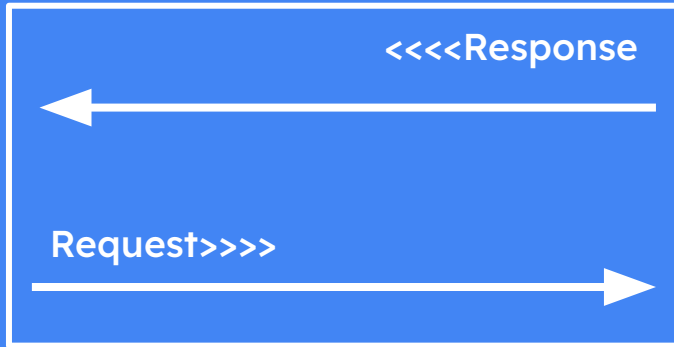


Back to the request!

Client:



Hypertext Transfer Protocol



Server



Back to the request!

Client:



Hypertext Transfer **Protocol**



Server

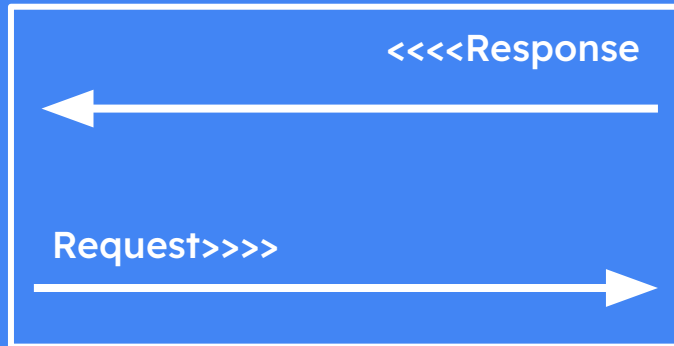


Back to the request!

Client:




Hypertext Transfer **Protocol**
Common structure for making a request




Server




What's in a HTTP Request?

- Where is this message going?
 - What kind of message is this?
 - Was the message successfully received?
 - What information am I sending?
 - What information am I getting back?
 - Extras?
- 


What's in a HTTP Request?

- Request URL
 - What kind of message is this?
 - Was the message successfully received?
 - What information am I sending?
 - What information am I getting back?
 - Extras?
- 

What's in a HTTP Request?

- Request URL
 - Request Method
 - Was the message successfully received?
 - What information am I sending?
 - What information am I getting back?
 - Extras?
- 

What's in a HTTP Request?

- Request URL
 - Request Method
 - Status Code: <https://httpstatusdogs.com>
 - What information am I sending?
 - What information am I getting back?
 - Extras?
- 


What's in a HTTP Request?

- Request URL
- Request Method
- Status Code
- Body
- What information am I getting back?
- Extras?

What's in a HTTP Request?

- Request URL
- Request Method
- Status Code
- Body
- Response
- Extras?

What's in a HTTP Request?

- Request URL
 - Request Method
 - Status Code
 - Body
 - Response
 - Request and Response Headers
- 

What's in a HTTP Request?

× Headers Preview Response Initiator Timing Cookies

▼ General

Request URL: `https://randomuser.me/api/`

Request Method: GET

Status Code: 🟢 200

Remote Address: `[2606:4700:e6::ac40:cd10]:443`

Referrer Policy: `strict-origin-when-cross-origin`

Request as API

Client:



Server: API



<<<<Response



Request>>>>

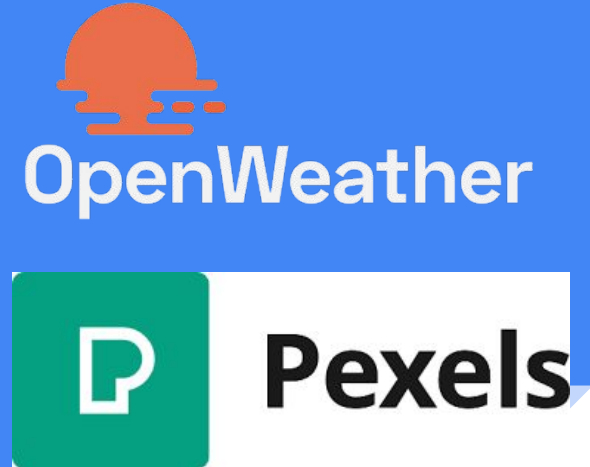


Request as API

Client:



Server: API



<<<<Response



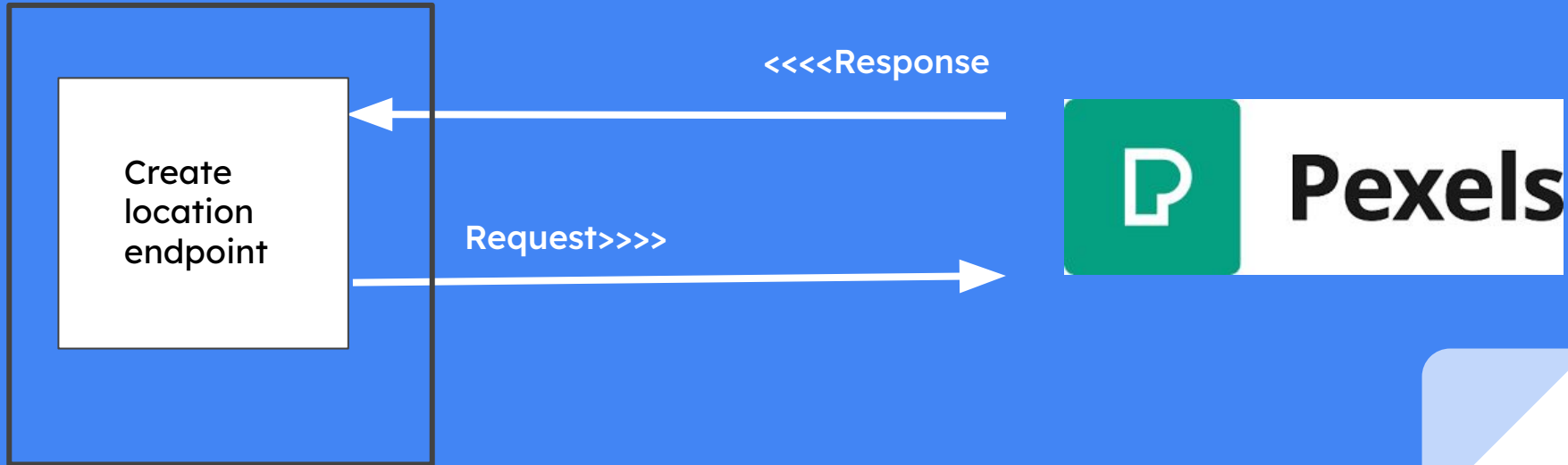
Request>>>>



Request as API

Client: Django Server

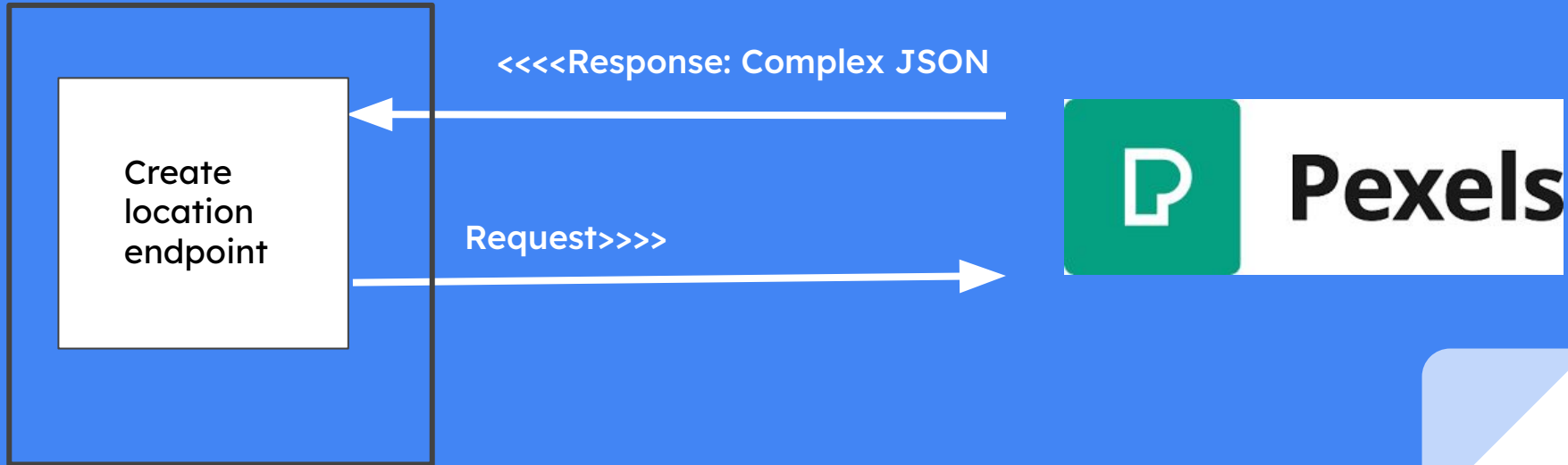
Server: API



Request as API

Client: Django Server

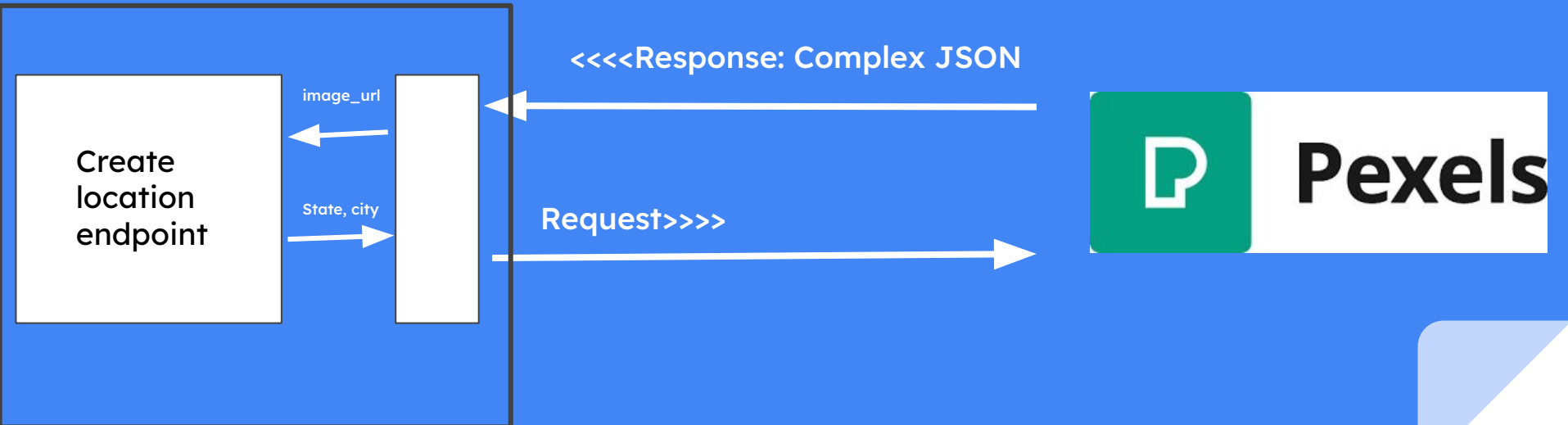
Server: API



Request as API

Client: Django Server

Server: API



Request as API

Client: Django Server

Server: API



Anti-Corruption Layers

Just a function that makes requests and “cleans” the data for us.

