

Web Design
& Development

Hello API / ui-stack

Feb-22

Kick-off

UI-Stack

1. UI-Stack
2. Fetch Status

UI-Stack

**Every screen you interact
with in a digital product
has multiple personalities.**

Scott Hurff

UI-Stack

How to fix a bad user interface

The screenshot shows a web browser window displaying a blog post. The header includes the site name 'Scott Hurff' and navigation links for 'HOME', 'BOOK', and 'WRITING'. A 'Subscribe' button and social sharing icons are also present. The main content area has a sub-header 'ESSAYS' and a large, bold title 'How to fix a bad user interface'. Below the title is a author bio section featuring a small profile picture of Scott Hurff, his name, and a '20 MIN READ' estimate. The post begins with a descriptive paragraph about the book 'Designing Products People Love'. It then poses a question about UI experiences and introduces the concept of 'Awkward UI'.

www.scotthurff.com/posts/why-your-user-interface-is-awkward ↗

Scott Hurff HOME BOOK WRITING

ESSAYS

How to fix a bad user interface

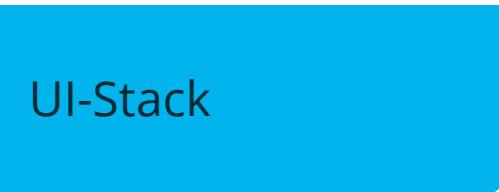
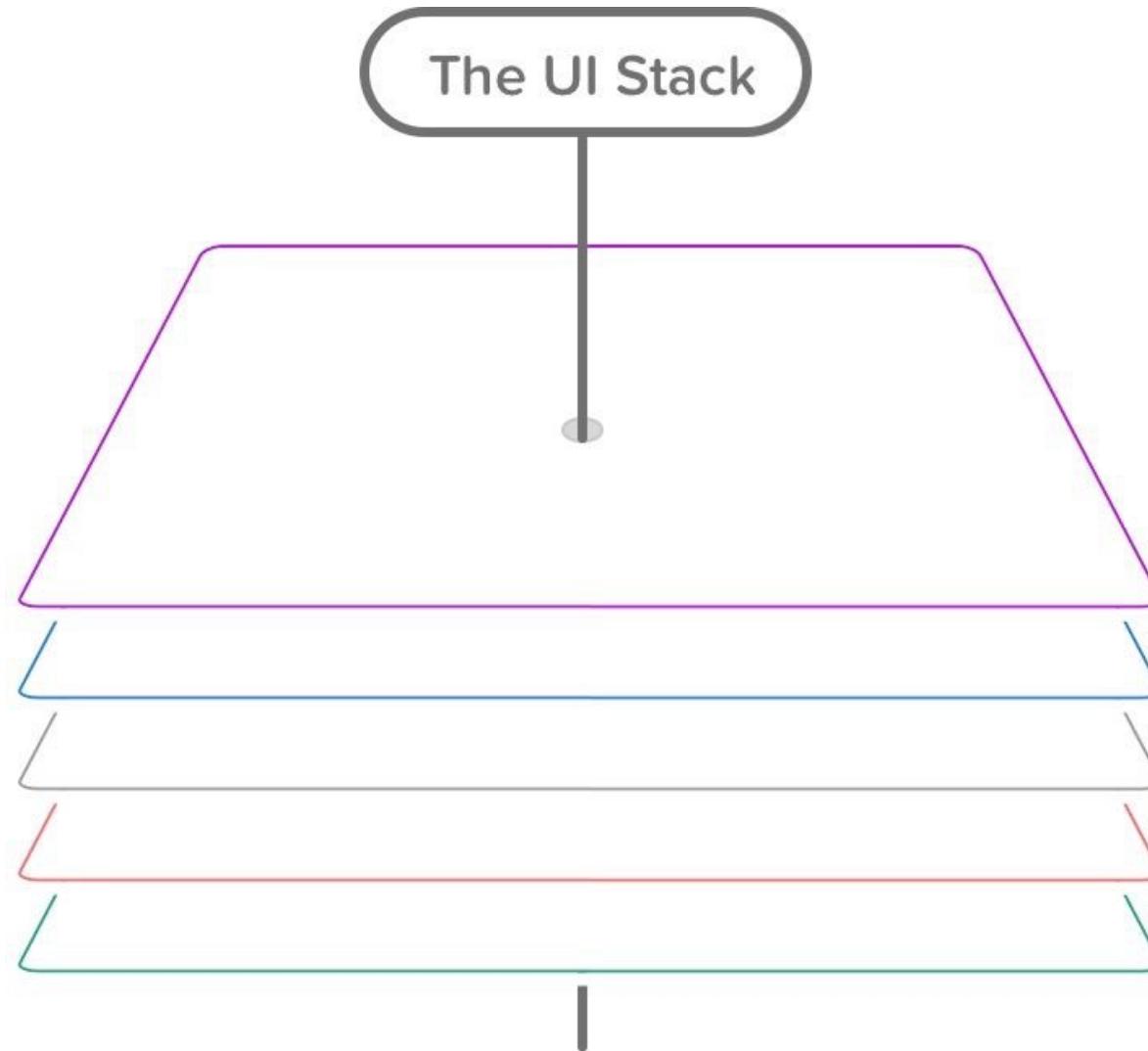
SCOTT HURFF
20 MIN READ

Hey! This is an excerpt from my book [Designing Products People Love](#), which was published by O'Reilly in January 2016. Learn more about the book and the 20+ product designers from Facebook, Twitter, Slack, etc. who were interviewed about how they work.

Have you ever experienced a user interface that feels lifeless? Have you created a UI that just seems to be missing...*something*?

If that's the case, you've probably experienced a case of Awkward UI.

Awkward UI is a missing loading indicator. It's forgetting to tell your customer where something went wrong. Bonus points for doing so with a scary error message. It's a graph that looks weird with only a few data points. It's a linear snap into place when introducing a new piece of data.



IDEAL STATE



EMPTY STATE

De eerste keer dat een gebruiker op je website komt of een app gebruikt.

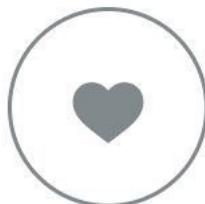
- Een lege *Wish List*



Wish Lists

My Wish Lists

Friends



Your Wish Lists will live
here

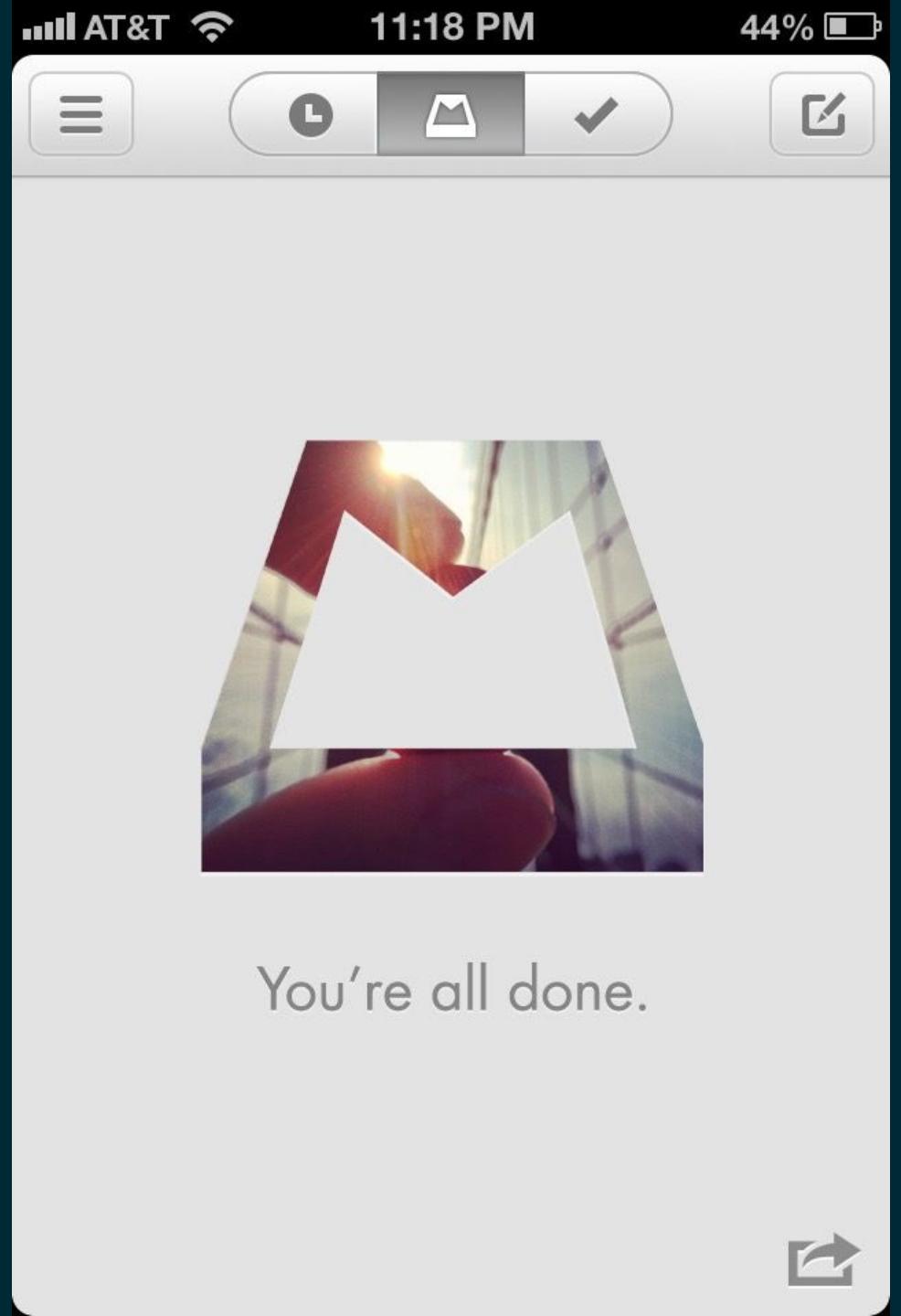
Explore thousands of fairytale
destinations and add your favorite
spaces from around the world.

Start Exploring

EMPTY STATE

Wanneer een gebruiker data of informatie van een scherm haalt.

- Inbox Zero



EMPTY STATE

Wanneer je niets kan tonen bij een zoekactie of filter/sorteer.

- No result Amazon

••○○○ Verizon 3:40 PM 98% 🔋

< ⌂ amazon Prime 0

🔍 metallikka

Showing results for "metallikka".
Did you mean: metallica

See more results for "metallica" in All Departments

Metallica
by Metallica
Audio CD
\$9.00 ✓Prime
Get it by Tuesday, Mar 31
62 offers from \$4.37
★★★★★ (1528)

Master Of Puppets
by Metallica
MP3 Music
This item is not available for purchase from your device.
★★★★★ (1477)

...And Justice for All
by Metallica
Audio CD
\$9.20 ✓Prime
Get it by Tuesday, Mar 31
50 offers from \$4.81
★★★★★ (997)

ERROR STATE

Create an account

Correct the following errors: username and password

Name

Stephanie

Smith

Username

stephsm

@email.com

That name is already being used

Password

•••••

Confirm password

•••••

Passwords don't match

Birthday

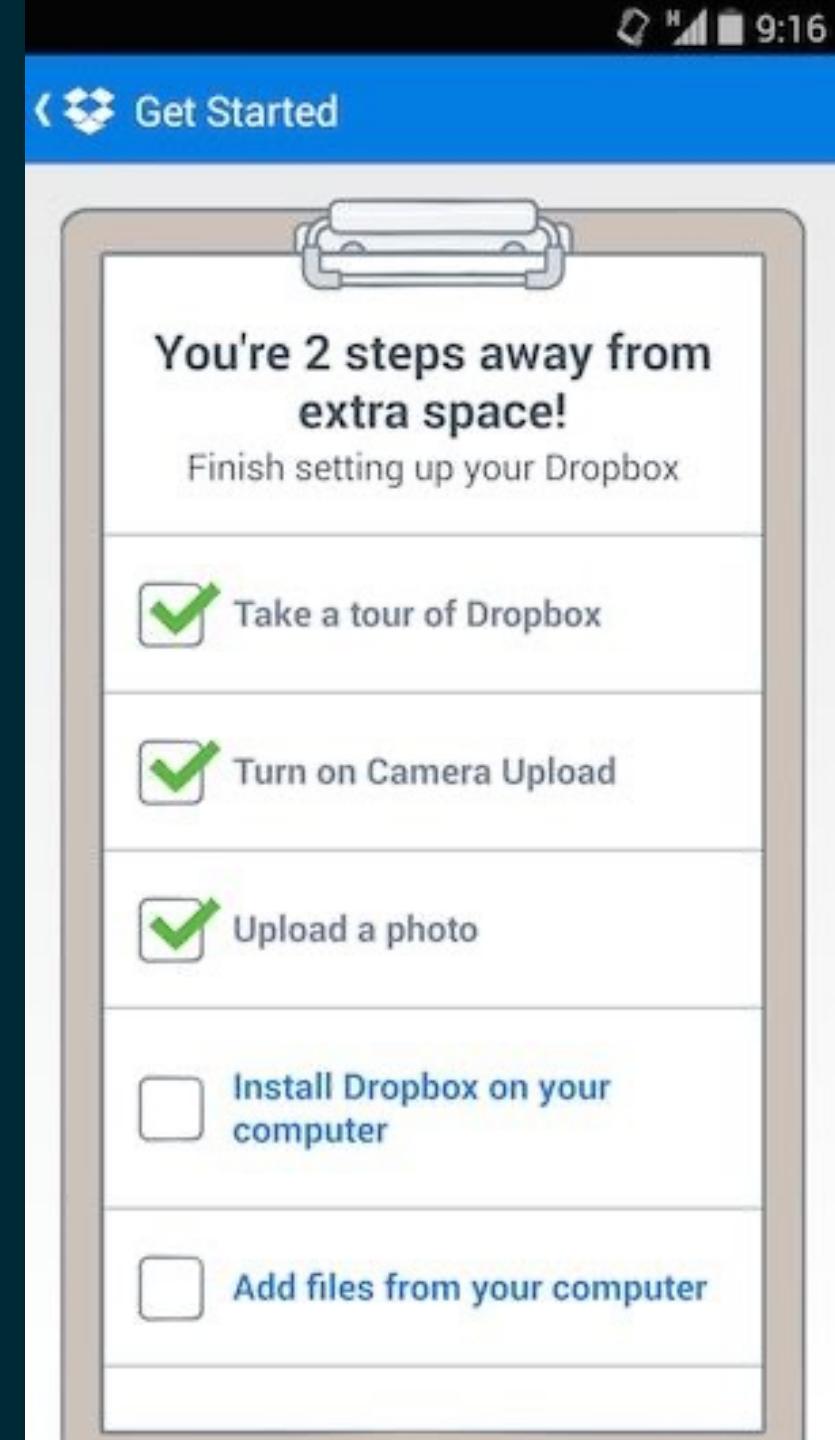
November

▼ 23

1990

Gender

PARTIAL STATE



PARTIAL

nl.pinterest.com/ko2641/_saved/

Home Search your Pins **Your Pins** **K**

ko2641
@ko2641
0 following

Share **Edit Profile**

Created **Saved**

All Pins
11 Pins Just now

house
8 Pins Just now

Skateboard
1 Pin 2m

+

?

UI-Stack

LOADING STATE

77% 07:59



General

FDND (docenten) (1 guest)



POSTS

FILES

MORE

C





General

FDND (docenten) (1 guest)



POSTS

FILES

MORE



General

FDND (docenten) (1 guest)



POSTS

FILES

MORE

No internet connection

Recordings
517.26 MB, Modified by Charley Muhren on 04/10/2...

PiE
232.43 KB, Modified by Joost Faber on 29/09/2021

Beeld FDND Kickoff
155.51 MB, Modified by Justus Sturkenboom on 06...

Beeld Voorhoede
11.71 MB, Modified by Justus Sturkenboom on 30/...

Beeld TheValley
2.57 MB, Modified by Justus Sturkenboom on 30/0...

Beeld Lifely
820.58 MB, Modified by Justus Sturkenboom on 17...

Beeld minor web
14.17 MB, Modified by Justus Sturkenbo...

Beeld IWB



General

FDND (docenten) (1 guest)



POSTS

FILES

MORE



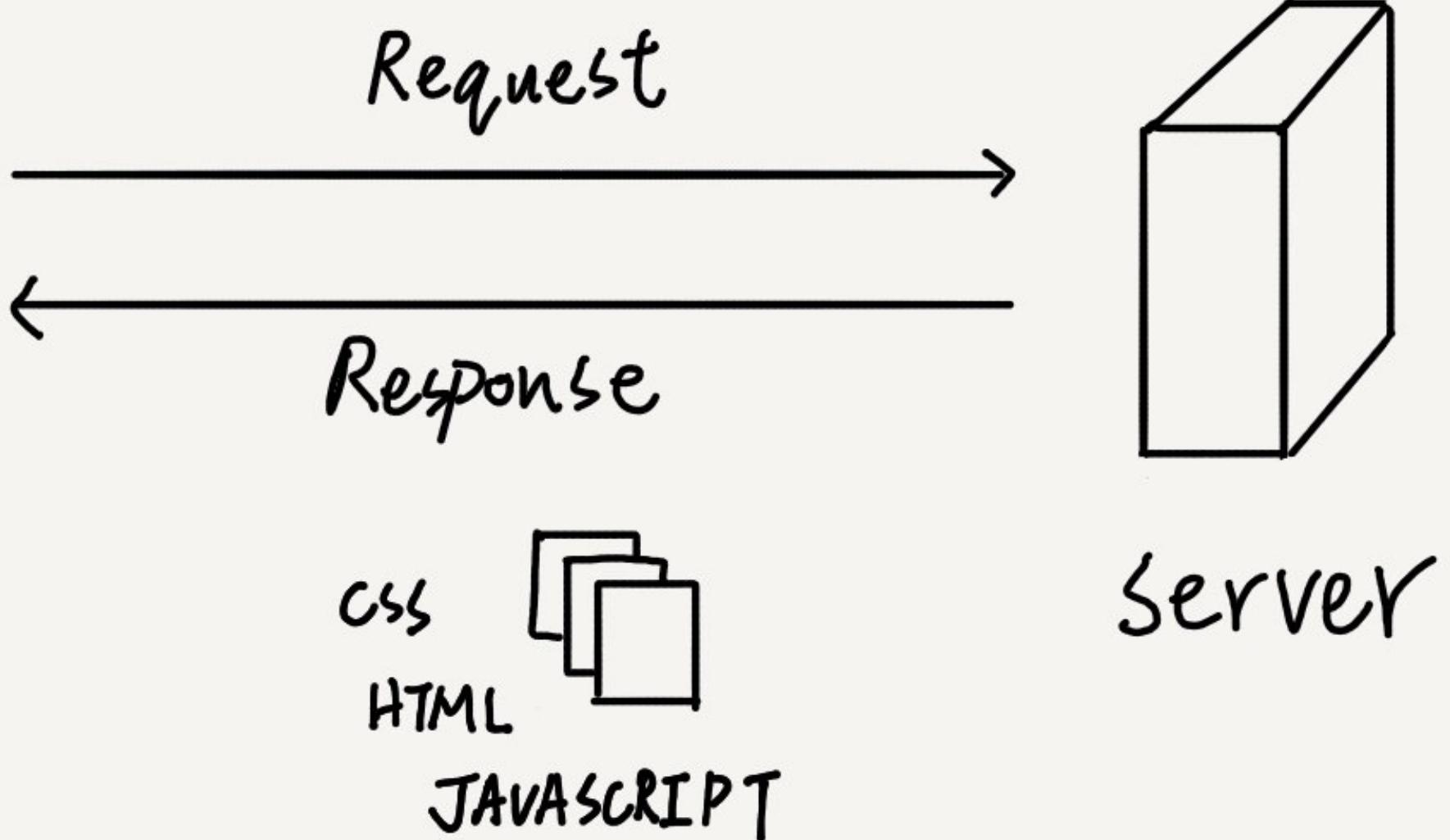
Share files in this channel

Everyone can work together on files you upload or create.





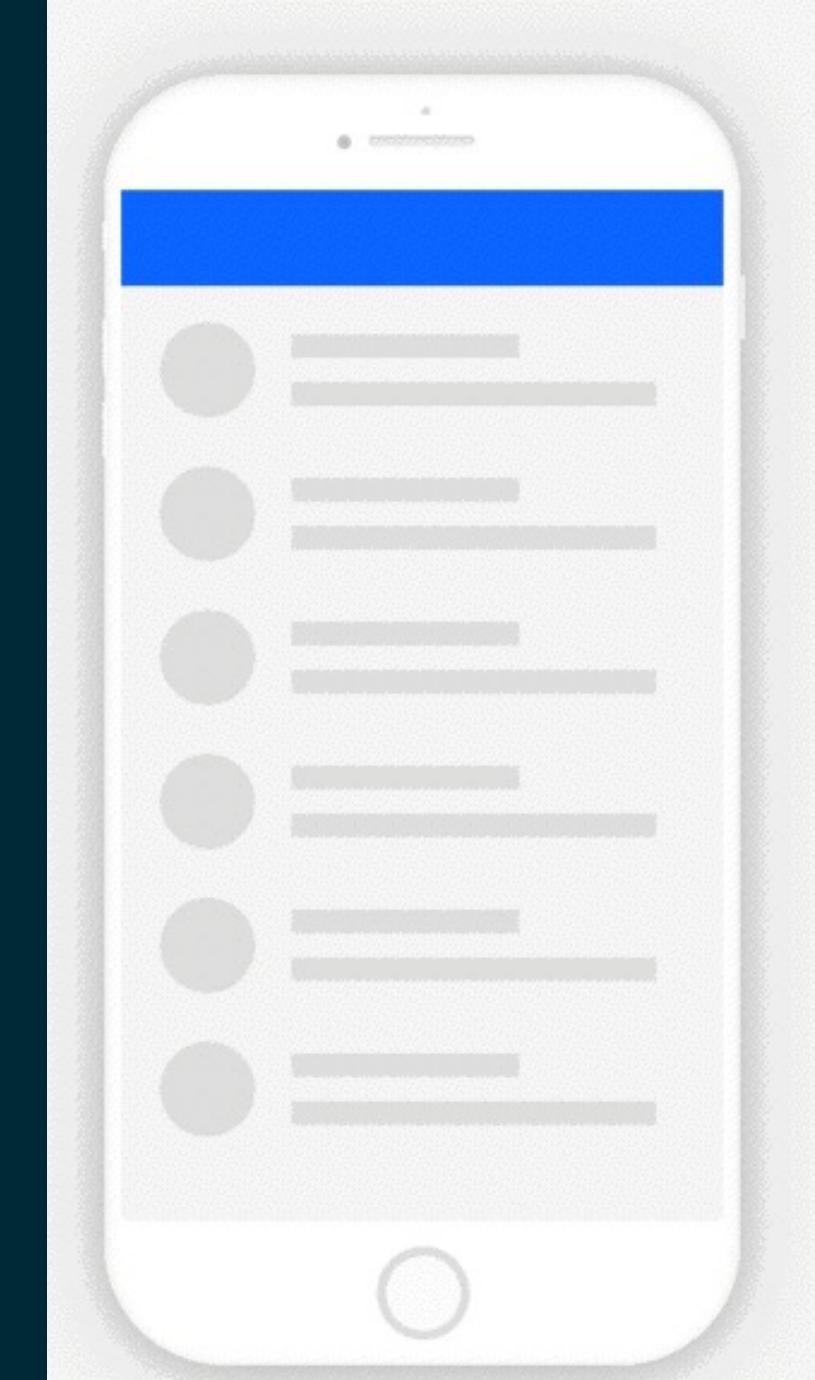
Client

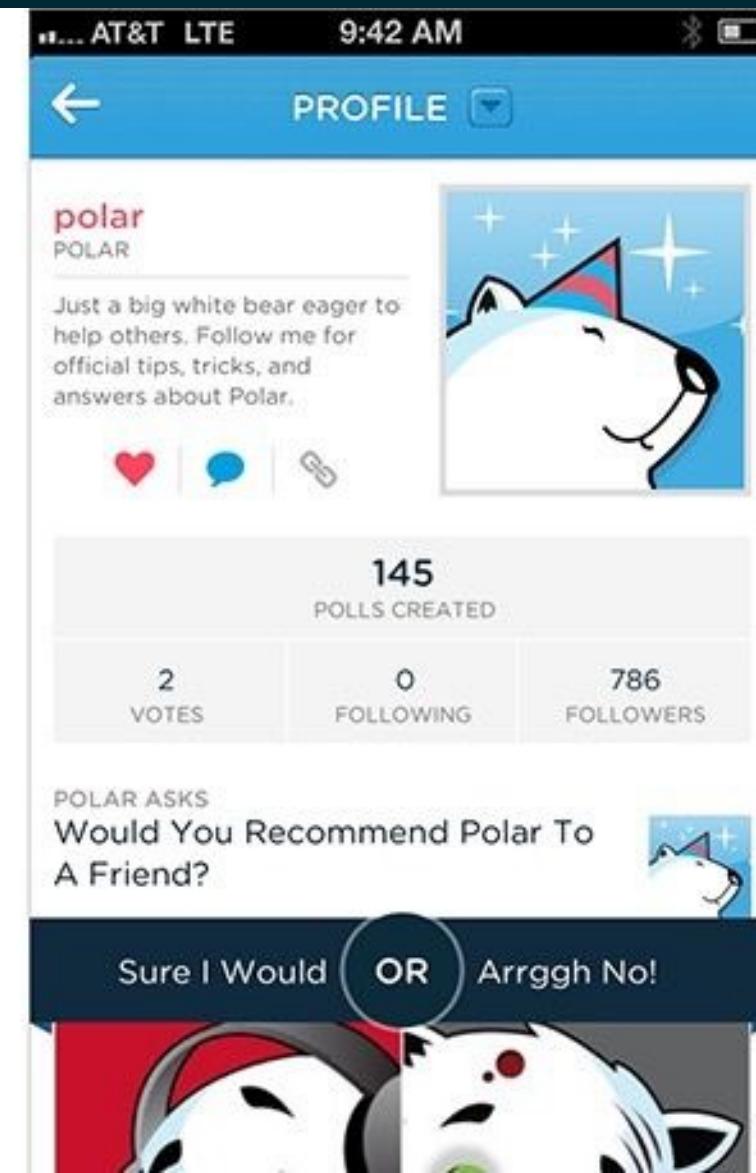
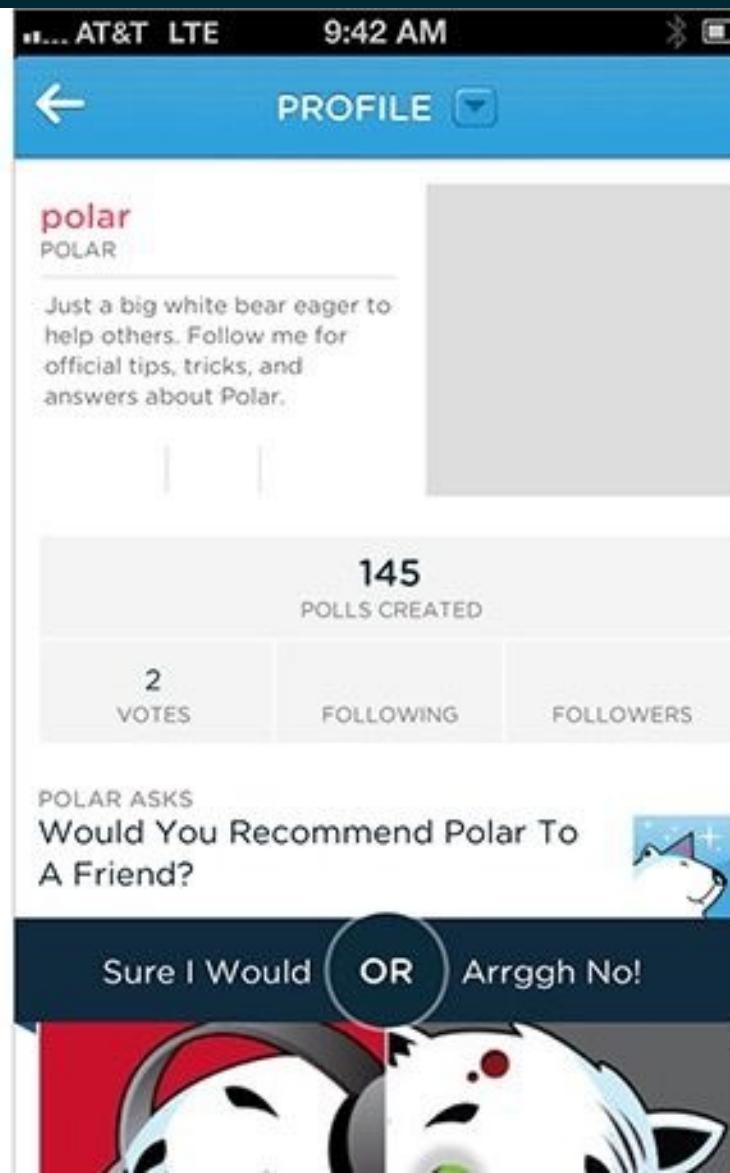
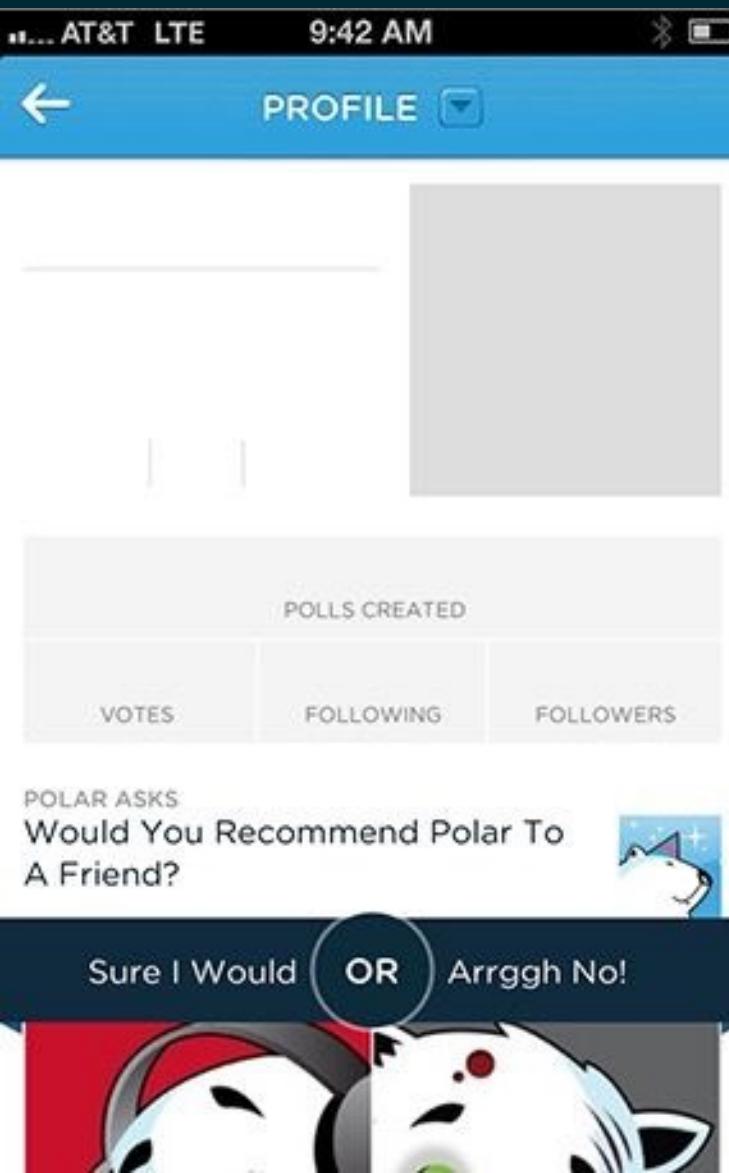


Server

LOADING STATE

- Skeleton screens







⌘1



⌘2



#random

☆ | 8 - | 0 | #minorvormgeving



Search



Gift 2

Connection trouble

Apologies, we're having some trouble with your WebSocket connection.

We've seen this problem clear up with a restart of Slack, a solution which we suggest to you now only with great regret and self-loathing.

OK

**Titus Wormer** 09:40

<https://github.com/CMDA/minor-everything-web/graphs/contributors>, due @koop, @joost, en @vasilis!

[CMDA/minor-everything-web](#)

Contribute to minor-everything-web development by creating an account on GitHub.

**Anne Marleen** 10:32

Minoren lijken soms wel de beste gloei & bloei omgevingen. Waarom noemen we ze niet playgrounds? Mooi gedaan!

**Koop Reynders** 10:56

Mischien kunnen jullie voor de minor UX een



Kick-off

UI-Stack

1. ~~UI-Stack~~
2. Fetch Status

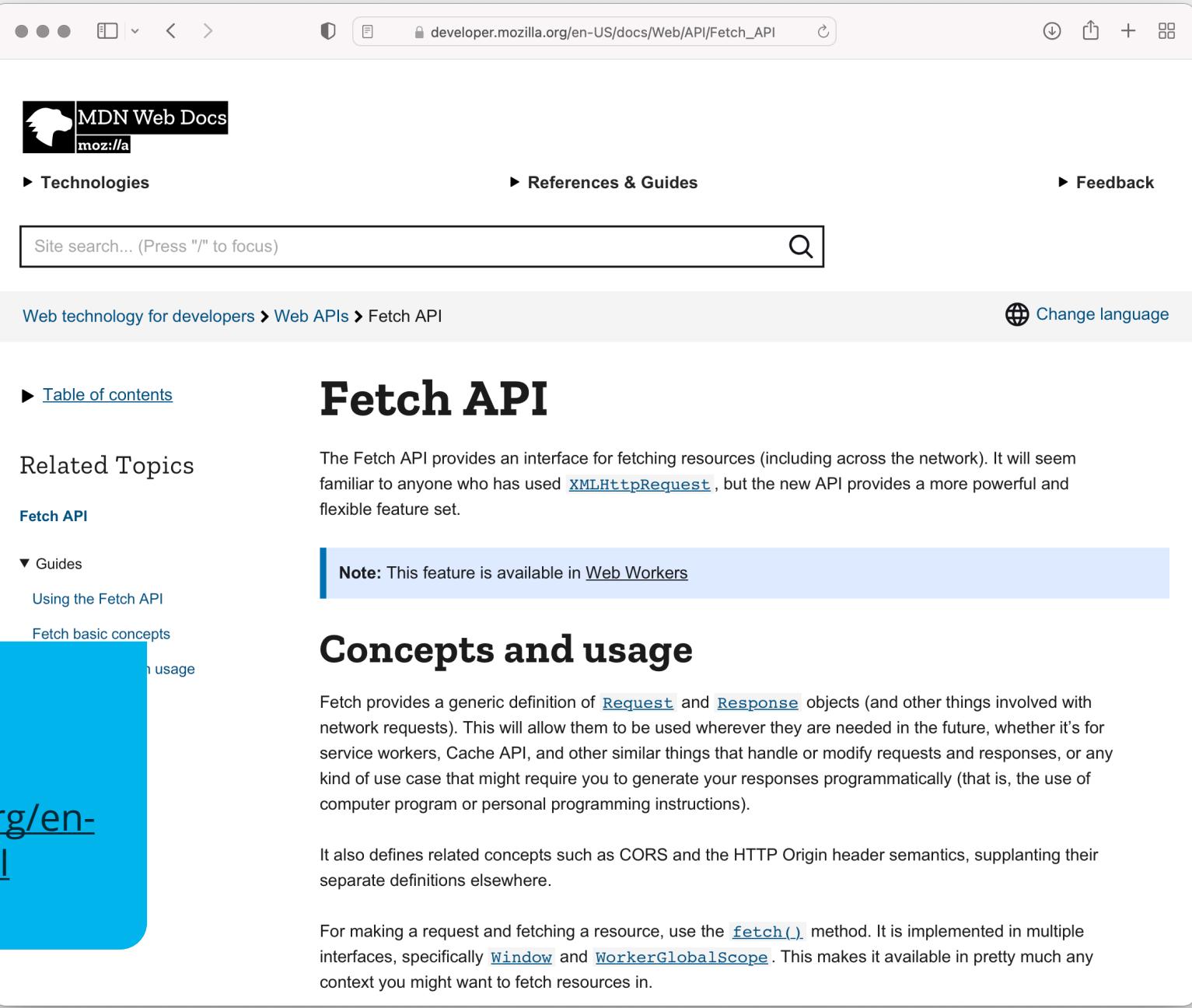
Fetch Status

Design and implement the UI-Stack

- Design an Empty state, Loading state and/or Error state and add them to your Wireflow.
- Sketch a Breakdown-graph: break the Wireflow down in HTML, CSS and JS.
- Implement the new code and test with a peer students.

Fetch API op MDN

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API



The screenshot shows the MDN Web Docs page for the Fetch API. The page has a header with the MDN logo, a search bar, and navigation links for Technologies, References & Guides, and Feedback. Below the header, there's a breadcrumb trail: Web technology for developers > Web APIs > Fetch API. A "Change language" button is also present. On the left, there's a sidebar with a "Table of contents" link and sections for Related Topics (Fetch API), Guides (Using the Fetch API, Fetch basic concepts), and Usage. The main content area features a large h1 heading "Fetch API". Below it, a text block explains that the Fetch API provides an interface for fetching resources, mentioning XMLHttpRequest as a familiar alternative. A note states that the feature is available in Web Workers. The "Concepts and usage" section follows, explaining that Fetch provides generic Request and Response objects and supplants CORS and HTTP Origin semantics. It also notes the use of the fetch() method across multiple interfaces like Window and WorkerGlobalScope.

MDN Web Docs
moz://a

▶ Technologies ▶ References & Guides ▶ Feedback

Site search... (Press "/" to focus)

Web technology for developers > Web APIs > Fetch API

Change language

▶ Table of contents

Related Topics

Fetch API

▼ Guides

Using the Fetch API

Fetch basic concepts

Usage

Fetch API

The Fetch API provides an interface for fetching resources (including across the network). It will seem familiar to anyone who has used [XMLHttpRequest](#), but the new API provides a more powerful and flexible feature set.

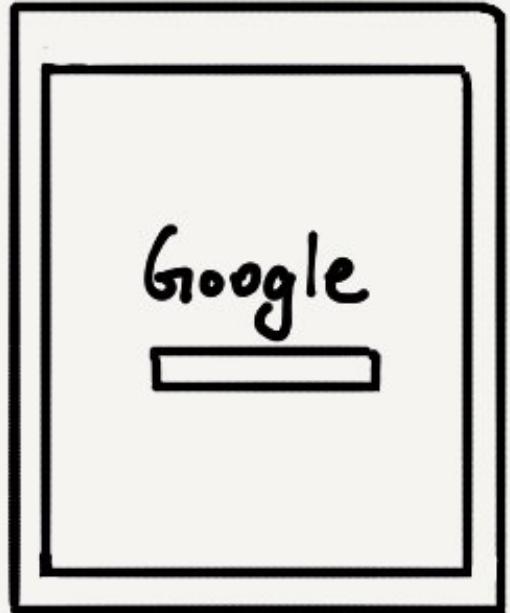
Note: This feature is available in [Web Workers](#)

Concepts and usage

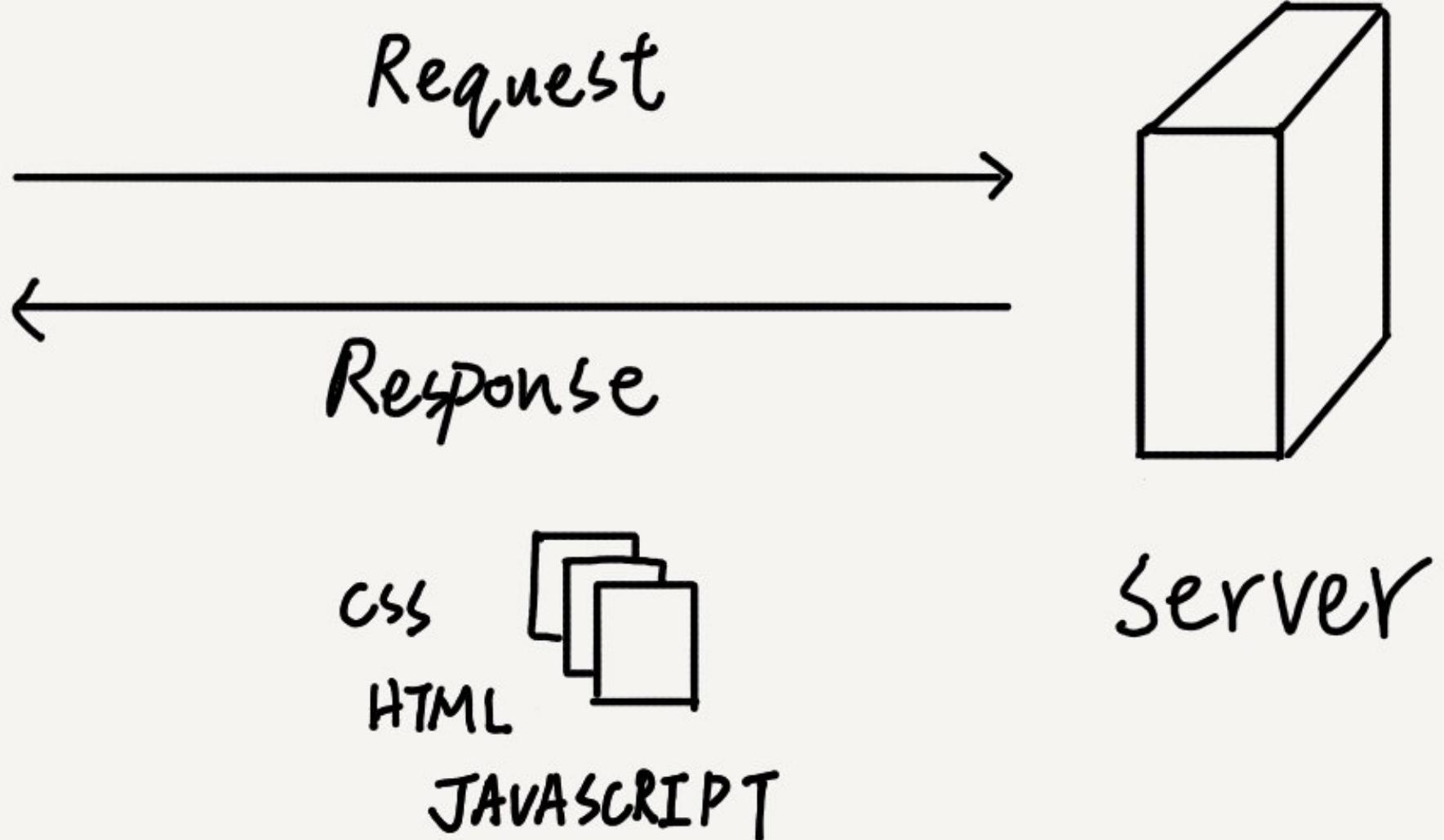
Fetch provides a generic definition of [Request](#) and [Response](#) objects (and other things involved with network requests). This will allow them to be used wherever they are needed in the future, whether it's for service workers, Cache API, and other similar things that handle or modify requests and responses, or any kind of use case that might require you to generate your responses programmatically (that is, the use of computer program or personal programming instructions).

It also defines related concepts such as CORS and the HTTP Origin header semantics, supplanting their separate definitions elsewhere.

For making a request and fetching a resource, use the [fetch\(\)](#) method. It is implemented in multiple interfaces, specifically [Window](#) and [WorkerGlobalScope](#). This makes it available in pretty much any context you might want to fetch resources in.



Client



Server

Waar komt de loading state?

Voor de Fetch

```
<script>
  // FDND API
  const display = document.querySelector('section');
  display.textContent = "Loading quotes, even geduld a.u.b"

  fetch("https://quote.api.fdnd.nl/v1/quote")
    .then(response => {
      return response.json();
    })
    .then(json => {
      display.textContent = "";
      //data
      let data = json.data[5];
      //title
      const title = document.createElement("h2");
      title.textContent = data.text;
      display.appendChild(title);
      //auteur
      const naam = document.createElement("h3");
      naam.textContent = data.name;
      display.appendChild(naam);
    });
  
```

Na response opruimen

Waar komt de empty state?

```
<body>
  <h1>Als student digital design, wil ik inspirerende web design quotes kunnen vinden</h1>
  <section>
    <p>Er is nog niets geladen ...</p>
  </section>

<script>
  // FDND API
  const display = document.querySelector('section');
  display.textContent = "Loading quotes, even geduld a.u.b";

  fetch("https://quote.api.fdnd.nl/v1/quote")
    .then(response => {
      return response.json();
    })
    .then(json => {
      display.textContent="";
      //data
      let data = json.data[5];
      //title
      const title = document.createElement("h2");
      title.textContent = data.text;
      display.appendChild(title);
    })
</script>
```

In de HTML

Waar komt de
error state?

Response.status

HTTP response
status codes

```
fetch(url)
  .then((response) => {
    if (response.status >= 200 && response.status <= 299) {
      return response.json();
    } else {
      throw Error(response.statusText);
    }
  })
  .then((jsonResponse) => {
    // do whatever you want with the JSON response
  })
  .catch((error) => {
    // Handle the error
    console.log(error);
  });
}
```

HTTP response status codes

Response.status

The screenshot shows a web browser displaying a blog post. The header of the browser window reads "learnwithparam.com/blog/how-to-handle-fetch-errors/". The main content area has a white background with a dark blue header bar at the top containing the site's logo "LEARN WITH PARAM" and navigation links for "Blog", "Videos", and a Twitter icon.

The main title of the post is "Error handling while using native fetch API in JavaScript". Below the title, the date "September 16, 2019" is followed by a popcorn icon and "2 min read". There is also a link "Edit this post on Github".

A yellow sidebar on the right side of the post contains the text: "Follow me on [twitter](#). I share quick tips, my reading list and also about free workshop and webinars around web and mobile development everyday."

The main content of the post discusses the Fetch API and its disadvantages, specifically mentioning error handling. It includes a code example:

```
fetch(url).then((response) => {
  // Always gets a response, unless there is network error
  // It never throws an error for 4xx or 5xx response 😞
```

Error handling while using native Fetch API in JavaScript.

<https://learnwithparam.com/blog/how-to-handle-fetch-errors/>



Shaun Moynihan

@moynihan



Follow

An SEO expert walks into a bar, bars, beer garden, hangout, lounge, night club, mini bar, bar stool, tavern, pub, beer, wine, whiskey...

Reply Retweet Favorite More

RETWEETS

2,386

FAVORITES

630

