Catbook Workshop 1: Client Side JS

Danny Tang
Previously...

Some frontend

- Profile page
- Nav bar
- Stories feed page

Techniques

- DOM manipulation with JS
In this workshop...

More frontend

- Stories feed page (fancier!)

Techniques

- DOM manipulation with JS
- API requests with JS*
- Tracking specific items with unique IDs

*aside: learn about asynchronicity
Code repository

go.6148.io/workshop1
Overview

index.html - the page for our stories

api.js - methods to perform get/post requests (provided)

feed.js - contains renderStories and helper methods (outline provided)

index.js - calls required javascript methods for final page render
It’s important to understand the tools that we are using
Exercise 1: understanding api.js

Take a few minutes to look at and try to understand api.js

Questions:

1) What parameters do the "get" and "post" methods take?
2) How are requests actually made (what object/tool is used)?
3) What happens when a request returns with a success?
4) What happens when a request returns with an error?
Exercise 1: understanding api.js

This time we’ve provided you with methods to make get and post requests in JS.

However, many frameworks have their own methods for making API requests.
Understanding our api
Understanding our api

```javascript
const API_ENDPOINT_START = 'https://mit-wdc-workshop-1.herokuapp.com';

// /api/stories
// /api/comment
```
Aside: why callback functions?

JavaScript is asynchronous

Asynchronous === one line of code does not wait for the previous one to finish
Aside: why callback functions?

HTTP requests take a while to finish
Exercise 2: feed.js

Goal:

To fetch stories and comments from our API and create HTML blocks to display their data

Breakdown:

1) renderStories makes API calls and passes responses to helper functions
2) storyDOMObject creates a story block
3) commentDOMObject creates a comment block
Exercise 2: feed.js

Part one (stories only):

1) Write code to fetch all stories with an API request (renderStories)
2) Write code to create an HTML block for a story (storyDOMObject)
3) Write code to prepend an HTML block for each story (renderStories)

```html
<div id=STORY-ID-HERE class="story card">
  <div class="card-body">
    <a class="story-creator card-title">STORY-CREATOR-HERE</a>
    <p class="story-content card-text">STORY-CONTENT-HERE</p>
  </div>
  <div class="card-footer">
    <div id=STORY-ID-HERE + "-comments" class="story-comments">
      <!-- All comments will go here-->
    </div>
  </div>
</div>
```
Exercise 2: feed.js

Part one (stories only):

1) Write code to fetch all stories with an API request (renderStories)
2) Write code to create an HTML block for a story (storyDOMObject)
3) Write code to prepend an HTML block for each story (renderStories)
Exercise 2: feed.js

Part one (stories only):

1) Write code to fetch all stories with an API request (renderStories)
2) Write code to create an HTML block for a story (storyDOMObject)
3) Write code to prepend an HTML block for each story (renderStories)
Exercise 2: feed.js

Part one (stories only):

1) Write code to fetch all stories with an API request (renderStories)
2) Write code to create an HTML block for a story (storyDOMObject)
3) Write code to prepend an HTML block for each story (renderStories)

```html
<div id="STORY-ID-HERE" class="story card">
  <div class="card-body">
    <a class="story-creator card-title">STORY-CREATOR-HERE</a>
    <p class="story-content card-text">STORY-CONTENT-HERE</p>
  </div>
  <div class="card-footer">
    <div id="STORY-ID-HERE -comments" class="story-comments">
      <!-- All comments will go here -->
    </div>
  </div>
</div>
```
Bookkeeping

To view current changes:

1) Add a call to `renderStories` in `index.js`

2) Load in all of your scripts at the bottom of `index.html`
Let’s get on the same page

git reset --hard

git checkout ex2_part1
Exercise 2: feed.js

Part two:

1) Write code to fetch all comments for each story (renderStories)
2) Write code to create an HTML block for a comment (commentDOMObject)
3) Write code to add an HTML block for each story’s comments (renderStories)
Exercise 2: feed.js

Part two:

1) Write code to fetch all comments for each story (renderStories)
2) Write code to create an HTML block for a comment (commentDOMObject)
3) Write code to add an HTML block for each story’s comments (renderStories)
Exercise 2: feed.js

Part two:

1) Write code to fetch all comments for each story (renderStories)
2) Write code to create an HTML block for a comment (commentDOMObject)
3) Write code to add an HTML block for each story’s comments (renderStories)
Let’s get on the same page

git reset --hard

git checkout complete
Recap

We...

1) Learned how to make API calls in JS
2) Learned about asynchronicity and callback functions
3) Learned how to track items with specific id’s
4) Gained more experience in creating/manipulating DOM elements
5) Made a pretty cool story feed page!