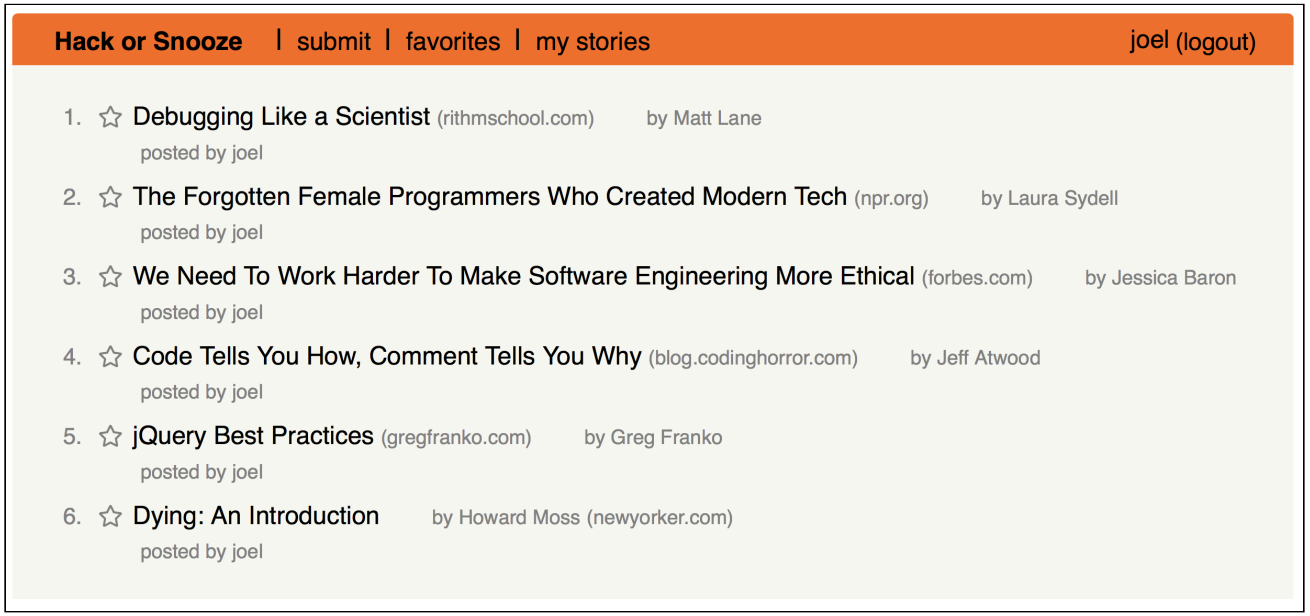
AJAX with jQuery Exercise: Hack-or-Snooze

Download code <../hack-or-snooze-ajax-api.zip>

In this exercise, we'll make a clone of the popular news-aggregator site Hacker News. It will allow users to create accounts, log in, create articles, mark articles as favorites, and more!



<_images/hack-or-snooze.png>

You can see a working copy of our solution <http://hack-or-snooze.surge.sh>.

We've already build the backend server API, so this exercises focuses on how to use an existing API to build out the functionality in front-end Javascript using jQuery and AJAX.

Part 0: Read the docs!

Head over to https://hackorsnoozev3.docs.apiary.io/ <https://hackorsnoozev3.docs.apiary.io/> and walk through the quickstart. Make all of these requests in Insomnia so you can save them and see what data needs to be sent to the server and what responses look like.

Make sure you are comfortable with the routes required to retrieve data from the API.

Part 1: Reading the starter code

Download the starter code and open up the ***index.html*** file. You will see that stories are displayed and there is functionality to log in and create a user.

There are two JavaScript files:

api-classes.js

this file contains the classes you will use and should be where you place all the logic for communicating with the API. **Make sure to read this file thoroughly.** There is a new keyword here, ***static***, which you want to make sure you understand before moving on.

ui.js

this file makes use of the classes in ***api-classes.js*** and contains all of the DOM manipulation and jQuery code necessary.

When a user has successfully logged in, a secured string or "token" is sent back from the server. This is the mechanism we are going to use to mark a user as logged in. The type of token we are using is called a JSON Web Token or JWT—we will discuss this at great length later on in the course. For now, all you need to know is that this token is a special string that we will use to mark a user as logged in.

We will also need this token when we make future API requests that are "Auth Required." You can see in the quickstart <https://hackorsnoozev3.docs.apiary.io/#introduction/quickstart> in the API docs that to make an API request that is "Auth Required" we must send the token in the body or query string.

Think of the token like a handstamp when you're trying to get into an exclusive club. The first time you go in, somebody stamps your hand. If you exit the club and come back, you can show your stamp at the door to prove you should be allowed in. No stamp, no access.

Note also that when the page refreshes, the user can stay logged in. This is because we are storing the token in localStorage!

Make sure you read through the code and play around with it before moving on! **Be sure to start this application using** `python3 -m http.server` or by using the VSCode live server extension.

Part 2: Creating new stories

Once a user has logged in, allow them to create a new story. This will involve adding a new form that when submitted will send the data in the form to the API, which will respond with the newly created story.

Make sure you append the newly created story to the DOM and only allow logged in users to create a new story.

The method for creating a story should be defined in the StoryList class.

As a note - there is a task on the backend that will delete stories every day so do not be alarmed if your story is deleted after a day! Since many people are sharing this API, that task is meant to clean things up daily.

Part 3: Favoriting stories

Allow logged in users to “favorite” and “un-favorite” a story. These stories should remain favorited when the page refreshes.

Allow logged in users to see a separate list of favorited stories.

The methods for favoriting and unfavoriting a story should be defined in the User class.

Part 4: Removing Stories

Allow logged in users to remove a story. Once a story has been deleted, remove it from the DOM and make sure when the page refreshes, the story is deleted.

Further Study

- Add some error handling for when a username has already been taken or if credentials are incorrect!
- Allow users to edit stories they have created.
- Add a section for a “user profile” where a user can change their ***name*** and ***password*** in their profile.
- Style the application so that it is presentable on mobile devices.
- Add infinite scroll! When a user scrolls to the bottom of the page, load more stories.
- Come up with some other features you can build using what our Hack or Snooze API makes available to you!

Solution

View our solution <solution/index.html>