Data Fetching with React

We are currently fetching data, but it's still pseudo data coming from a promise we set up ourselves. The lessons up to now about asynchronous React and advanced state management were preparing us to fetch data from a real third-party API. We will use the reliable and informative Hacker News API to request popular tech stories.

Instead of using the `initialStories` array and `getAsyncStories` function (you can remove these), we will fetch the data directly from the API:

```
const API_ENDPOINT = 'https://hn.algolia.com/api/v1/search?query=';
const App = () => {
  ...

  React.useEffect(() => {
    dispatchStories({ type: 'STORIES_FETCH_INIT' });

    fetch(`${API_ENDPOINT}react`) // B
      .then(response => response.json()) // C

      .then(result => {
        dispatchStories({
          type: 'STORIES_FETCH_SUCCESS',

          payload: result.hits, // D
        });
      })
      .catch(() =>
        dispatchStories({ type: 'STORIES_FETCH_FAILURE' })
      );
  }, []);

  ...
};
```

src/App.js

First, the `API_ENDPOINT` (A) is used to fetch popular tech stories for a certain query (a search topic). In this case, we fetch stories about React (B). Second, the native browser's fetch API is used to make this request (B). For the fetch API, the response needs to be translated into JSON (C). Finally, the returned

API, the response needs to be translated into JSON ©. Finally, the returned

result follows a different data structure (D), which we send as payload to our component's state.

In the previous code example we used JavaScript's Template Literals for a string interpolation. When this feature wasn't available in JavaScript, we'd have used the + operator on strings instead:

```
const greeting = 'Hello';

// + operator
const welcome = greeting + ' React';
console.log(welcome);
// Hello React

// template literals
const anotherWelcome = `${greeting} React`;
console.log(anotherWelcome);
// Hello React
```

src/App.js

The complete code:

```
  □ □ □□  □   ã□  F   □□  □   □  )□     □   9□  5□  @@  □   °□  n□  □PNG
□
   IHDR   □   □□□   (-□S   äPLTE"""""""""""""""""""2PX=r□)7;*:>H□¤-BGE□□8do5Xb6[eK□®K□¯1MU
□
   IHDR   □   □□□   ×©ÍÊ  □ePLTE"""""""""""""""""""""""""2RZN¢¹J□«3R[J□¬)59YÁÞ0KS4W`Q«ÄL□²%
?^q÷ñíÛ□ï.},□isæÝ_TttÔ¾ □1#□□/(ì□-[□□□è`□è`Ì□ÚïÅðZ□d5□□□□?ÎebZ¿Þ□i.Ûæ□□□ìqÎ□+1°□}Â□5
□
   IHDR       □□   D¤□Æ  □APLTE   """""""""""""""""""""""""""""2RZVºÖ_ÔôU·Ñ=r□$()'25]ÎíC□□0
□
   IHDR   @   @□□   □·□ì  □:PLTE   """""""""""""""""""""""""""""""""""""""""""""""""""""""""""
¢ßqÇ8Ù□´□mKË±mÆ¶mÛü·yi!è□ÎªYÏuë ÀÏ_Àï?i÷□ý+ò□□ÄA□|□ù{□□´?¿□_En□).□JËD¤<□
©¬¢Z\Ts©R*□(□  ¯©□J□□□□u□X/□4J□9□¡5·DEµ4kÇ4□&i¥V4Ú□¡®Ð□□¯□vsf:àg,□¢èBC»î$¶□ºÍùî□□á□@□
-ê>Û□º«¢XÕ¢î}ß¨ëÛÑ;□ÃöN´□ØvÅý□Î¸ÿ1 □ë×ÄO@&v/Äþ_□ö\ô□Ç\í.□□¾+0□□;□□□!□fÊ□¦´Ó%Â JY·O□Â□'
```

Check your browser to see stories related to the initial query fetched from the Hacker News API. Since we used the same data structure for a story for the sample stories, we didn't need to change anything, and it's still possible to filter the stories after fetching them with the search feature. We will change this behavior in one of the next sections. For the App component, there wasn't much data fetching to implement here, though it's all part of learning how to manage asynchronous data as state in React.

Exercises:

- Confirm the changes from the last section.
- Read through Hacker News and its API.
- Read more about the browser native fetch API for connecting to remote APIs.
- Read more about JavaScript's Template Literals.