

CS385 Lecture 10 – Working with API data in our applications



Situation – we now can obtain EXTERNAL data from an API

- What can we do with the data within our React Application?
- **ANSWER** – Whatever we need to do with the data!
- The JSON response, from the API, is now just a JSON object array.
- We can map, filter, reduce, sort, and so on. Essentially, we can perform any array-based operation we require.

Let's use `times.json` (from GitHub as our API data source)

JS App.js

```
7 [loading, setLoading] = useState(false);
8 flag to indicate an error, if any - initially null.
9 [error, setError] = useState(null);
10
11 Effect is a React Hook
12 ts you synchronize a component with an external system.
13 fect(() => {
14 This is just a static JSON file - so not a 'real' API
15 st URL =
16 https://raw.githubusercontent.com/petermooney/cs385/main/flights/times.json
17
18 nc function fetchData() {
19 ry {
20 const response = await fetch(URL);
21 const json = await response.json(); // wait for the JSON
22 setLoading(true);
23 // IMPORTANT - look at the JSON response - look at the str
24 setData(json.flightTimes);
25 // assigned to 'data' as an array of JSON objects
26 catch (error) {
27 setError(error); // take the error message from the system
28 setLoading(false);
29 // end try-catch block
30 / end of fetchData
31
32 chData(); // invoke fetchData in useEffect
33 ); // end of useEffect
34 tice the empty square brackets here?
35 is means that the useEffect hook does not depend
36 any other variable. Therefore, here it will only
37 open once.
38
39 rror) {
40 urn ch1>Oops! An error has occurred: [error.toString()]</h1>
```

Browser Tests

https://7gr67n.csb.app/

Number of flights returned: 1200

Pilot	Flight No.	Dept.	Dest	Arrival	
Lotti Quilter	LH-6204	Prague	Antwerpen	17:52	
Imogen Karppi	BA-651	Munich	Athens	11:26	
Warde Urch	SK-526	Barcelona	Dortmund	21:48	
Brew Roizn	FH-284	Sofia	Naples	16:50	
Willi Fittes	IE-750	Las Palmas de Gran Canaria	Oslo	12:35	
			Naples	14:54	
			Bristol	21:52	
			Thessaloniki	23:03	
			Las de Gran	Essen	18:38
			Marseille	12:30	
			Shannon	19:52	
			Genoa	09:42	
			Pisa	13:53	
			Antwerpen	17:40	
			Edinburgh	22:29	
			Porto	08:44	
			Marseille	23:12	
			Frankfurt am	10:14	

Code Blame

14404 lines (14404 loc) · 379 KB

```
1 {
2   "flightTimes": [
3     {
4       "id": "LH-6204",
5       "dept": "Prague",
6       "dest": "Antwerpen",
7       "passengers": 107,
8       "arrival": "17:52",
9       "aircraft": {
10         "type": "CRJ-900",
11         "crew": 10,
12         "captain": "Lotti Quilter"
13       }
14     },
15     {
16       "id": "BA-651",
17       "dept": "Munich",
```

Example 1: Filter by time

- Screenshot taken at 13:50 – we FILTER using the arrival property of the data

```
55 // the response of the API call from above.
56
57 function ResultsComponent(props) {
58   // Creating a basic filter for times
59   // for our purposes we can use very simple time comparison.
60   // We assume all flights are TODAY
61
62   function flightTimeFilter() {
63     return function (flightObject) {
64       // The arrival property of the data is HH:MM
65       const timeParts = flightObject.arrival.split(":");
66       let h = timeParts[0]; // extract the hour
67       let m = timeParts[1]; // extract the minute
68       // create a Javascript Date Object
69       let flightDate = new Date();
70       flightDate.setHours(h);
71       flightDate.setMinutes(m);
72
73       return flightDate >= new Date();
74     };
75   }
76
77   // here we apply the filter directly to the API response
78   let localAPIData = props.APIData.filter(flightTimeFilter());
79   return (
80     <>
81     <h1>Number of flights returned: {localAPIData.length}</h1>
82     <table border="1">
83       <thead>
```

https://7gr67n.csb.app/

Number of flights returned: 740

Pilot	Flight No.	Dept.	Dest	Arrival
Lotti Quilter	LH-6204	Prague	Antwerpen	17:52
Warde Urch	SK-526	Barcelona	Dortmund	21:48
Brew Roizn	FH-284	Sofia	Naples	16:50
Celinda Grint	PM-69853	Madrid	Naples	14:54
Agace Patron	CS-428	Budapest	Bristol	21:52
Harlan Domino	UA-5044	Bucharest	Thessaloniki	23:03
Ruby Cochern	FR-04515	Las Palmas de Gran Canaria	Essen	18:38
Dar Swallowell	UA-77749	Manchester	Shannon	19:52
Devy Lally	IE-9173	Bochum	Antwerpen	17:40
Hobey Welbeck	PM-4494	Tirana	Edinburgh	22:29
Bernie Lindemann	BA-865	Paris Orly	Marseille	23:12
Tabatha Tregaskis	UA-407	Bremen	Frankfurt am Main	19:14
Carlie Cundict	DK-2486	Amsterdam Schiphol	Krakow	20:20
Christiana Valeri	FH-6644	Belgrade	Zagreb	18:35
Vale Larrie	UA-673	Copenhagen	Antwerpen	21:16
Leah Simmell	IE-7725	Vienna	Zagreb	18:10
Carlyn Foddy	SY-39825	Sofia	Belfast	16:17
Charmine Tringham	LH-48361	Bucharest	Stuttgart	15:06
Clotilda Alstead	FH-24973	Vienna	Istanbul Atatürk	20:20
Caresse Lowery	SK-4030	Barcelona	Antwerpen	23:22
Justis Struijs	DK-6401	Milan	Oslo	21:31
Meredith Olczyk	FH-54237	Pristina	Genoa	23:05

Example 2: Filter by dept/dest

```
74   };
75   }
76
77   // A standard search for a string within the
78   // departure or destination property values
79   function flightDeptDestFilter(searchTerm) {
80     return function (flightObject) {
81       let dept = flightObject.dept.toLowerCase();
82       let dest = flightObject.dest.toLowerCase();
83
84       return (
85         searchTerm !== "" &&
86         (dept.includes(searchTerm.toLowerCase()) ||
87          dest.includes(searchTerm.toLowerCase()))
88       );
89     };
90   }
91
92   // here we apply the filter directly to the API response
93   let localAPIData = props.APIData.filter(flightDeptDestFilter("Sofia"));
94   return (
95     <>
96     <h1>Number of flights returned: {localAPIData.length}</h1>
97     <table border="1">
98       <thead>
99         <tr>
100           <th>Pilot</th>
101           <th>Flight No.</th>
102           <th>Dept.</th>
103           <th>Dest</th>
104           <th>Arrival</th>
105         </tr>
106       </thead>
```



Number of flights returned: 24

Pilot	Flight No.	Dept.	Dest	Arrival
Brew Roizn	FH-284	Sofia	Naples	16:50
Carlyn Foddy	SY-39825	Sofia	Belfast	16:17
Malinda McCamish	CS-444	Sofia	Thessaloniki	08:38
Henka Rainbird	LH-6737	Sofia	Turin	23:28
Darda Roe	SY-096	Sofia	Wroclaw	13:18
Kaylyn Prewer	SK-11726	Sofia	Krakow	21:32
Cherry Poff	SY-96306	Sofia	Riga	20:30
Arlene Prudence	UA-5528	Sofia	Zaragoza	16:51
Kimmy Roskelley	PM-01553	Sofia	Frankfurt am Main	12:36
Emyle Winborn	UA-3789	Sofia	Dublin	09:23
Chris Ovitz	FH-93961	Sofia	Goeteborg	17:56
Hasty Ingall	SK-617	Sofia	Naples	15:11
Urban Tocqueville	LH-05588	Sofia	Naples	19:01
Janey Josephy	BA-1781	Sofia	Riga	13:48
Rudy Habbijam	LH-4823	Sofia	Zagreb	17:18
Paulie Behrens	SK-98011	Sofia	Pisa	09:44
Lenna Caps	UA-8638	Sofia	Antwerpen	22:40
Karlee Marnes	IE-2861	Sofia	Dortmund	11:54
Deni Antoszczyk	UA-273	Sofia	Essen	21:41
Cristal Glazebrook	IE-22201	Sofia	Glasgow	09:21
Rorie Tabbitt	LH-469	Sofia	Frankfurt am Main	09:30
Vinnie Lanchester	IE-3660	Sofia	Sarajevo	23:11
Quinlan MacGuinness	FH-238	Sofia	Koeln	14:04
Portia Trainer	BA-1569	Sofia	Edinburgh	10:13

Console 0

Problems 1

React DevTools 0

Example 3- chain or compose MULTIPLE filters together

- Let's combine our time filter (screenshot 14:05) and our dept/dest filter

```
    );  
  };  
}  
  
// here we apply the filter(s) directly to the API response  
let localAPIData =  
  props.APIData.filter(flightDeptDestFilter("Sofia")).filter(flightTimeFilter());  
return (  
  <>  
    <h1>Number of flights returned: {localAPIData.length}</h1>  
    <table border="1">  
      <thead>  
        <tr>  
          <th>Pilot</th>  
          <th>Flight No.</th>  
          <th>Dept.</th>  
          <th>Dest</th>  
          <th>Arrival</th>  
        </tr>  
      </thead>  
      <tbody>  
        {localAPIData.map((p, index) => (  
          <tr key={index}>  
            <td>  
              <i>{p.aircraft.captain}</i>  
            </td>  
            <td>  
              <i>{p.flightNo}</i>  
            </td>  
            <td>  
              <i>{p.dept}</i>  
            </td>  
            <td>  
              <i>{p.dest}</i>  
            </td>  
            <td>  
              <i>{p.arrival}</i>  
            </td>  
          </tr>  
        )}  
      </tbody>  
    </table>  
  </>  
);
```

<https://7gr67n.csb.app/>

Number of flights returned:
13

Pilot	Flight No.	Dept.	Dest	Arrival
Brew Roizn	FH-284	Sofia	Naples	16:50
Carlyn Foddy	SY-39825	Sofia	Belfast	16:17
Henka Rainbird	LH-6737	Sofia	Turin	23:28
Kaylyn Prewer	SK-11726	Sofia	Krakow	21:32
Cherry Poff	SY-96306	Sofia	Riga	20:30
Arliene Prudence	UA-5528	Sofia	Zaragoza	16:51
Chris Ovitz	FH-93961	Sofia	Goeteborg	17:56
Hasty Ingall	SK-617	Sofia	Naples	15:11
Urban Tocqueville	LH-05588	Sofia	Naples	19:01
Rudy Habbijam	LH-4823	Sofia	Zagreb	17:18
Lenna Caps	UA-8638	Sofia	Antwerpen	22:40
Deni Antoszczyk	UA-273	Sofia	Essen	21:41
Vinnie Lanchester	IE-3660	Sofia	Sarajevo	23:11

Challenges to think about for your project or larger apps

- Could you add a textbox to the application to allow someone to search the RESPONSE Data from the API?
- Could you write other types of filter functions (depending on the data)?
- How would you use multiple components – Rendering the output, obtaining user input, and so on?

Lecture 9 + 10 – THE END for Javascript coding today

Let's talk about the project

- Example project screencasts from 2022/2023

CS385 Project Update

- Are you working as a group (2, 3, or 4) or as an individual?
- YOU need to inform me [via email only] of your project status before lectures on Tuesday 7th November 2023 (after mid-term break)
- One person per group can email.
- It is YOUR RESPONSIBILITY to inform me.



Let's talk about Lab Exam 1