

```
1  import React, { useEffect, useState } from "react";
2
3  function App() {
4    // the data response from the API - initially empty array
5    const [data, setData] = useState([]);
6    // a flag to indicate the data is loading - initially false
7    const [loading, setLoading] = useState(false);
8    // a flag to indicate an error, if any - initially null.
9    const [error, setError] = useState(null);
10
11    //useEffect is a React Hook
12    // lets you synchronize a component with an external system.
13    useEffect(() => {
14      // This is just a static JSON file - so not a 'real' API
15      const URL =
16        "https://raw.githubusercontent.com/petermooney/cs385/main/flights/today.json";
17
18      async function fetchData() {
19        try {
20          const response = await fetch(URL);
21          const json = await response.json(); // wait for the JSON response
22          setLoading(true);
23          // IMPORTANT - look at the JSON response - look at the structure
24          setData(json); // IMPORTANT - the JSON response is
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      fetchData(); // invoke fetchData in useEffect
33    }, []); // end of useEffect
34    // notice the empty square brackets here?
35    // This means that the useEffect hook does not depend
36    // on any other variable. Therefore, here it will only
37    // happen once.
38
39    if (error) {
40      return <h1>Oops! An error has occurred: {error.toString()}</h1>;
41    } else if (loading === false) {
42      return <h1>Loading Data..... please wait!</h1>;
43    } else {
44      return (
45        // send the data to the ResultsComponent for render
46        <>
47          <ResultsComponent APIData={data} />
48        </>
49      );
50    } // end else
51  } // end App() function or component
52
53  // This is our Results Component
```

```

54 // This component will display the contents
55 // the response of the API call from above.
56
57 function ResultsComponent(props) {
58   // To reduce the array size - let's just take
59   // a smaller number of elements.
60   let dataLength = props.APIData.length;
61   let n = Math.floor(Math.random() * dataLength);
62   // now use slice() to take the first n objects in the array
63   let localAPIData = props.APIData.slice(0, n);
64   return (
65     <>
66       <h1>Number of flights returned: {localAPIData.length}</h1>
67       <table border="1">
68         <thead>
69           <tr>
70             <th>Flight No.</th>
71             <th>Dept.</th>
72             <th>Dest</th>
73             <th>Arrival</th>
74           </tr>
75         </thead>
76         <tbody>
77           {localAPIData.map((p, index) => (
78             <tr key={index}>
79               <td>
80                 <b>{p.id}</b>
81               </td>
82               <td>
83                 <b>{p.dept}</b>
84               </td>
85               <td>
86                 <b>{p.dest}</b>
87               </td>
88               <td>{p.arrival}</td>
89             </tr>
90           )]}
91         </tbody>
92       </table>
93     </>
94   );
95 }
96 export default App;
97

```