

# Solution: Can you fetch data?

Here is the completed solution code for the `App.js` file:

```
1  import React from "react";
2
3  function App() {
4    const [user, setUser] = React.useState([]);
5
6    const fetchData = () => {
7      fetch("https://randomuser.me/api/?results=1")
8        .then((response) => response.json())
9        .then((data) => setUser(data));
10   };
11
12   React.useEffect(() => {
13     fetchData();
14   }, []);
15
16   return Object.keys(user).length > 0 ? (
17     <div style={{padding: "40px"}}>
18       <h1>Customer data</h1>
19       <h2>Name: {user.results[0].name.first}</h2>
20       <img src={user.results[0].picture.large} alt="" />
21     </div>
22   ) : (
23     <h1>Data pending...</h1>
24   );
25 }
26
27 export default App;
```

Here is the sample output from the solution code for the `App.js` file. Note that the lab produces a different random image and name with each refresh, so your output is likely to have a different name and a different customer image than the one shown below.

## Customer data

**Name: Liam**



## Step-by-step solution

### Step 1

Inside the `fetchData()` function's code block, you executed the `fetch()` function, passing it a single string argument: `"https://randomuser.me/api/?results=1"`.

```

1  const fetchData = () => {
2    fetch("https://randomuser.me/api/?results=1")
3  };

```

## Step 2

Next, inside the `fetchData()` function, under the `fetch()` function call, you added the following piece of code:

```

1  const fetchData = () => {
2    fetch("https://randomuser.me/api/?results=1")
3      .then((response) => response.json())
4  };

```

## Step 3

You then added another `then()` call, which takes an arrow function.

The passed-in arrow function, receives a `data` argument, and using that `data` argument, it invokes the `setUser()` function, with the data passed to it.

```

1  const fetchData = () => {
2    fetch("https://randomuser.me/api/?results=1")
3      .then((response) => response.json())
4      .then((data) => setUser(data));
5  };

```

## Step 4

In the `return` statement of the App component, under the `h1` heading that reads “Customer data”, you added an `h2` heading, with the following code: `Name: {user.results[0].name.first}`

```

1  return Object.keys(user).length > 0 ? (
2    <div style={{padding: "40px"}}>
3      <h1>Customer data</h1>
4      <h2>Name: {user.results[0].name.first}</h2>
5    </div>
6    ) : (
7    <h1>Data pending...</h1>
8    );

```

## Step 5

You then updated the `return` statement of the App component by adding another line of code under the newly-added `h2`.

You added an `img` element, with the `src` attribute and an `alt` attribute holding the following code:

`{user.results[0].picture.large}`

```

1  return Object.keys(user).length > 0 ? (
2    <div style={{padding: "40px"}}>
3      <h1>Customer data</h1>
4      <h2>Name: {user.results[0].name.first}</h2>
5      <img src={user.results[0].picture.large} alt="" />

```

```
6 </div>
7   ) : (
8   <h1>Data pending...</h1>
9   );
```