



Rapid React Part 6: Passing Data with Props

In [Part 5](#), we introduced you to the basics of CSS and how to implement them with Bootstrap. In this installment, we look at how to pass data between components with props (properties) and how we can pass properties using ReactRouter's location hooks. The sample code for this installment is available on [GitHub](#).

Key Concepts

In this section, we explain what are props and ReactRouter's useLocation hook.

Props

In React, props (properties) pass data between components. In frontend frameworks, such as Angular, data can be passed between parent and child components and back again. This data is mutable and can be changed at either end. React does things differently and has a stricter approach to passing data. For a start, data passed between parent and child components only flows in one direction (parent to child). Furthermore, passed props are read-only (immutable) this means that the child component is not unable to change (mutate) the state of a passed prop.

Location Hook

As explained in [Part 3: User Interaction and Forms](#), Hooks are a mechanism for collecting data and managing user interaction. In addition to the hooks provided by React, many React libraries and packages provide their own hooks. The ReactRouter provides a useLocation hook to pass props and state. We will use this hook to pass props between the Login and Tasks components.

Displaying Profile Information

Let's start by adding a component to the Tasks page that displays the name of an authenticated user. The page also provides a button that displays the full profile.





In the src/components folder, create a new file called Info.js. Paste the following code into the file.

```
import Alert from "react-bootstrap/Alert";

const Welcome = (user) => {
  return (
    <div>
      Welcome: {user.name}
      <span style={{ float: "right" }}>
        <button
          className="btn btn-sm btn-primary"
          onClick={(e) => {
            alert(JSON.stringify(user));
          }}
        >
          View Profile
        </button>
      </span>
    </div>
  );
};

const Info = (props) => {
  return (
    <Alert variant={props.variantInfo}>
      <Welcome {...props.user} />
    </Alert>
  );
};

export default Info;
```

The component references the Bootstrap Alert component, a bar that displays information to the user. The Info function receives properties from the Tasks component. This information is used to set the Alert's variant property to a specific color. The Alert uses the Welcome function to display a welcome message to the authenticated user. The component receives the prop's user object. When the user clicks View Profile, an alert displays the user profile.

Passing User Profile Props from the Login Component

When a user logs in with the correct credentials, the Login form locates the profile and navigates to the tasks page. Let's modify Login.js to pass this data to Tasks.js. To do this, we will need to replace the Navigate with the useNavigate hook. At the top of the page, comment add the following code.

```
// import { Navigate } from "react-router-dom";
import { useNavigate } from "react-router-dom";
```

In FormBody, add the following code to pass the user profile props to the Tasks component.

```
{/* {user} && (<Navigate to="/tasks" { state: { user:user } />)} */}
{ user ? navigate('/tasks', { state: { user:user } }) : null}
```



Displaying Props in the Tasks Component

Open Tasks.js. At the top of the page, add a reference to useLocation hook and the Info component.

```
import { useLocation } from "react-router-dom";

import userTasks from "../data/tasks";
import tableCols from "../data/cols";

import Table from 'react-bootstrap/Table';
import Container from 'react-bootstrap/Container';

import NavbarComponent from '../components/Navigation';
import Footer from '../components/Footer';
import Info from "../components/Info";
```

In the Tasks component function, add a location variable that references userLocation. After location, add a user variable that retrieves the user data from the location's object's state property. Now, we can reference the Info component and use it to pass. The Info component passes the user profile and variantInfo (color) from the Tasks component.

```
const Tasks = () => {

  const location = useLocation();
  const user = location.state;

  return (
    <div>
      <NavbarComponent />
      <Container style={{ marginTop: "50px" }}>
        < Info {...user} variantInfo="info" />
        <TaskTable />
      </Container>
      <Container style={{ paddingTop: '20%' }} >
        <Footer />
      </Container>
    </div>
  );
};
```

Refresh the page, and the Info component will be displayed.

Conclusion and What's Next

In this installment, we looked at how to pass data between components with props (properties) and how we can pass properties using ReactRouter's location hooks. In the next installment, we look at how we retrieve data from an API and use that data in our app.