Understanding the benefits of TypeScript with React

In this lesson, we are going to discover the benefits that TypeScript brings when developing React apps.

WE'LL COVER THE FOLLOWING

Using CodeSandbox

CodeSandbox is a browser-based IDE that supports React and TypeScript apps. When you create a project in CodeSandbox, the basic configuration is automatically done, so we can start writing code straight away. It's an excellent tool for learning React with TypeScript!

We are going to use CodeSandbox in some of the lessons in this course to try out different aspects of React with TypeScript.

The following link takes us to CodeSandbox:

https://codesandbox.io/

Consuming and refactoring a component prop without TypeScript

In order to get a sense of how TypeScript increases productivity, we are going to carry out an exercise in CodeSandbox.

First, we will consume and refactor a React component that doesn't use TypeScript. Let's carry out the following steps:

- Open the exercise in CodeSandbox: https://codesandbox.io/s/typescript-

benefits-js-978y6

- In `index.js`, add the `Header` component to the `App` component inside the `div` element:

```
function App() {
  return (
    <div>
      <Header />
    </div>
  );
}
```

The `App` component won't render anything. Try to pin the problem down. How easy is it to understand the problem?

- Open `header.js` and read the code. We can see that the `Header` expects a `profile` prop. This `profile` prop is an object that also expects props. How easy is it to work out the props that need to be passed to the `Header` component?

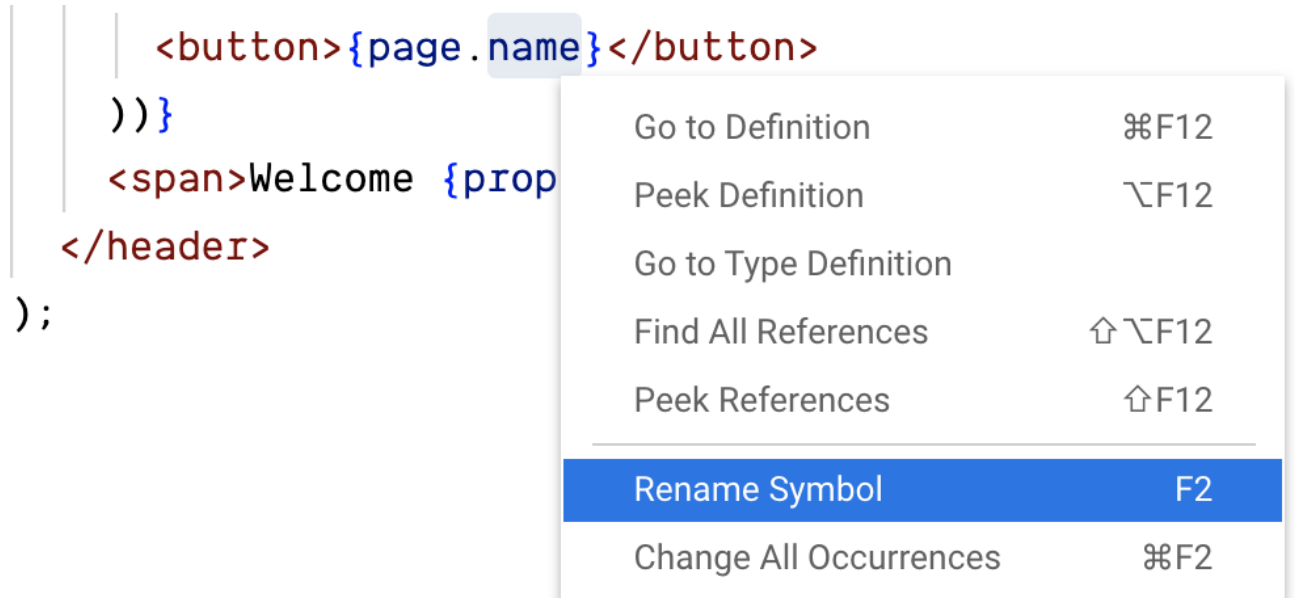
- Let's move back to `index.js` and supply a `profile` prop to the `Header` component:

```
<Header
  profile={{
    pages: [{ name: "Home" }, { name: "products" }],
    user: { name: "Bob" }
  }}
/>
```

Did the editor provide any IntelliSense when entering the `profile` prop?

Show Answer

- Move back to `header.js`. We are going to change the page `name` property to `caption`.
- Right-click on the pages `name` property and choose the **Rename Symbol** option:



Were you able to rename the `name` property using the **Rename Symbol** feature? If so, did it automatically correct the references in the `App` component?

Show Answer

That completes this exercise with the plain React component. Keep the experience fresh in your mind and quickly move on to the next section.

Consuming and refactoring a component prop with TypeScript

Let's try this same exercise with TypeScript:

- Open the exercise in CodeSandbox: https://codesandbox.io/s/typescript-benefits-ts-bm2qx
- In `index.tsx`, add the `Header` component to the `App` component inside the `div` element:

```
function App() {
  return (
    <div>
      <Header />
    </div>
  );
```

```
}
```

Notice the red squiggly line that appears under the `Header` component in the JSX. Hover over it:

```
Property 'profile' is missing in type '{}' but required in type
'Props'. ts(2741)

header.tsx(5, 3): 'profile' is declared here.

Quick Fix...  Peek Problem

    <Header />
```

A tooltip informs us that the `Header` component expects a `profile` prop. Neat!

- Let's add a `profile` prop to the `Header` component:

```
<Header
  profile={{
    pages: [{ name: "Home" }, { name: "products" }],
    user: { name: "Bob" }
  }}
/>
```

Did the editor provide any IntelliSense when entering the `profile` prop?

Show Answer

- Move to `header.tsx` and use the **Rename Symbol** feature to change the page `name` property to `caption`.

Did this work this time? If so, did it automatically correct the references in the `App` component?

Show Answer

Wrap up

TypeScript has many benefits when using it to develop React apps:

- Sophisticated type checking
- Accurate IntelliSense
- Accurate code refactoring
- Accurate code navigation

These benefits increase our productivity.

Excellent - that’s another lesson completed!

Now that we have started to understand and experience the benefits of TypeScript, we’ll learn about some of the types that are available in TypeScript in the next chapter. Before that, we’ll test what we have learned in this chapter with a quiz.