

1.7 TeslaCar Component

Now let's render a nice Tesla car image with wheel animation.

Create the `src/components/TeslaCar` directory, create a `TeslaCar.js` file in it, and inside your `TeslaCar.js` file :

```
import React from 'react';
import './TeslaCar.css';
const TeslaCar = (props) => (
  <div className="tesla-car">
    <div className="tesla-wheels">
      <div className={`tesla-wheel tesla-wheel--front tesla-wheel--${props.wheelsize}`}></div>
      <div className={`tesla-wheel tesla-wheel--rear tesla-wheel--${props.wheelsize}`}></div>
    </div>
  </div>
);
TeslaCar.propTypes = {
  wheelsize: React.PropTypes.number
}
export default TeslaCar;
```

Here we specify `propTypes` using the React built-in typechecking. In development mode, React checks `props` passed to the component. (Only in development mode for performance reasons) For each `props` attribute, React attempts to find it in the component's `propTypes` object to determine whether (1) prop is expected (2) prop is the correct type. In this case, the `TeslaCar` component expects the `props` attribute `wheelsize` and specifies that it is a `number` type. If the wrong value is provided, a warning appears in the JavaScript console, which is useful for fixing potential bugs in early stage.

More information on `React.PropTypes` can be found [here](#)

Update: New Deprecation Warnings in React 15.5

In 15.5, instead of accessing `PropTypes` from the main React object, install the `prop-types` package and import them from [here](#)

```
// Before (1.7 and below) import React from 'react';
```

```
import React from 'react';
import './TeslaCar.css';
.....

TeslaCar.propTypes = {
  wheelsize: React.PropTypes.number
}
export default TeslaCar;

// After (1.7)
import React from 'react';
import PropTypes from 'prop-types';
import './TeslaCar.css';
.....
TeslaCar.propTypes = {
  wheelsize: PropTypes.number
}

export default TeslaCar;
```

1.7.1 TeslaCar Component Style

Next, create a `TeslaCar.css` file in the `src/components/TeslaCar` directory and give it the following style. Since the code is long and omitted here, let's check the [source code](#).

```
.tesla-car {
  width: 100%;
  min-height: 350px;
  background: #fff url(../../assets/tesla.jpg) no-repeat top center;
  background-size: contain; }
.tesla-wheels {
  height: 247px;
  width: 555px;
  position: relative;
  margin: 0 auto; }
...
```

This gives us our animations and the component base for the car, which is displayed as background images.

1.7.2 Import TeslaCar component in TeslaBattery Container

Next, we need to add this component to our container again. Import `TeslaNotice` component in `TeslaBattery.js`:

```
...
import TeslaCar from '../components/TeslaCar/TeslaCar';
class TeslaBattery extends React.Component {
  render() {
    return (
```

$$\begin{array}{l} \dots \\ \} \\ \} \end{array} \quad \begin{array}{l}) \\ \} \\ \} \end{array}$$

</g>

</svg>