

React TypeScript HackJam

@Hackages

Tuesday, March 19, 2019

TypeScript

- Superset of JavaScript which provides optional static typing (classes, interfaces, ...)
- Gives compile time guarantees (doesn't check anything at runtime)

Example

```
interface User {  
  name: string;  
  age: number;  
  role: Role;  
}
```

```
type Role = 'Admin' | 'Basic';
```

```
const myUser: User = { name: 'Bouba', age: 28, role: 'Admin' };
```

How it works ?

TypeScript checks the code (types) and **analyses it during the compilation.** Then it strips all the type stuff from the code, and what is left is JavaScript. The **output is untyped.**

Advantages

- Quick feedback loop on a lot of errors
- Code base becomes much more self-documenting and discoverable
- Helps refactoring/renames, moving stuff around
- Auto-completion and error messages
- Gradual adoption

React SpaceX

Mission

Important Stats

TOTAL LAUNCH **92**

PERCENTAGE OF SUCCESS **74 %**

PERCENTAGE OF FAILURE **26 %**

ALL

PAST

UPCOMING

Filter mission name



2006 - FalconSat



2007 - DemoSat



2008 - Trailblazer



2008 - RatSat



2009 - RazakSat



2010 - Falcon 9 Test Flight

Example: Function Component

```
type Props = {  
  message: string;  
};  
  
const App: React.FC<Props> = ({ message }) => (  
  <div>{message}</div>
```

Example: Class Component

```
type Props = {  
  message: string;  
};  
type State = {  
  count: number;  
};  
class App extends React.Component<Props, State> {  
  state { count: 0 };  
  render() {  
    return <div>{this.props.message} {this.state.count}</div>  
  }  
}
```


Instructions

1. Add types for the `launches` (in the data folder) we'll use in the application. You can add these types in the `types.ts` file

```
const allLaunches = [  
  {  
    flight_number: 1,  
    launch_year: '2006',  
    mission_name: 'FalconSat',  
    details: 'Engine failure at 33 seconds and loss of vehicle',  
    launch_success: false,  
  },  
];
```

Instructions

2. Add types for the components:

MyComponent.jsx -> MyComponent.tsx

```
// import MyComponent from './MyComponent.jsx'  
import MyComponent from './MyComponent';
```

Let's go!

Github: <https://github.com/hackages/react-spacex>

Working version: <https://reactspacex.netlify.com/>

Cheatsheets: <https://devhints.io/typescript> / <https://github.com/sw-yx/react-typescript-cheatsheet>