How TypeScript Made Me a Better JS Developer

Kylie Stewart


TypeScript
Developer
JAVASCRIPT
JS

What is TypeScript, exactly?

Statically Typed

Dynamically Typed

Compared to JavaScript

```
class Button extends Component {
  render() {
    return (
      <button {...this.props}>{children}</button>
    );
  }
}

Button.defaultProps = {
  href: '',
  disabled: false
};

export default Button;
```

```
interface IButtonProps {
  href: string;
  disabled?: boolean;
  large?: boolean;
}

class Button = React.Component<IButtonProps> => {
  public static defaultProps: Partial<IButtonProps> = {
    href: '',
    disabled: false
  };

  public render() {
    return (
      <button {...this.props}>{children}</button>
    );
  }
};

export default Button;
```

But How Does TypeScript Make You a Better Developer?


TS

📦 Create more portable code

📓 Write better documentation

📈 Increase dev speed

Write Better Component APIs

Using Interfaces

```
interface IButtonProps {
  href: string;
  disabled?: boolean;
  large?: boolean;
}

class Button = React.Component<IButtonProps> => {
  public static defaultProps: Partial<IButtonProps> = {
    href: '',
    disabled: false
  };

  public render() {
    return (
      <button {...this.props}>{children}</button>
    );
  }
};

export default Button;
```

PropTypes vs. Interfaces

```
class Button extends Component {
  render() {
    return (
      <button {...this.props}>{children}</button>
    );
  }
}

Button.defaultProps = {
  href: '',
  disabled: false
};

Button.PropTypes = {
  children: PropTypes.node,
  href: PropTypes.string,
  disabled: PropTypes.boolean,
  large: PropTypes.boolean
};

export default Button;
```

```
interface IButtonProps {
  href: string;
  disabled?: boolean;
  large?: boolean;
}

class Button = React.Component<IButtonProps> => {
  public static defaultProps: Partial<IButtonProps> = {
    href: '',
    disabled: false
  };

  public render() {
    return (
      <button {...this.props}>{children}</button>
    );
  }
};

export default Button;
```

Composing Better Props and State

HOC

Higher Order Component

Extending Interfaces

```
interface IWithUserDataProps {
  id: number;
  age: number;
  firstName: string;
  lastName: string;
  nickname?: string;
}
```

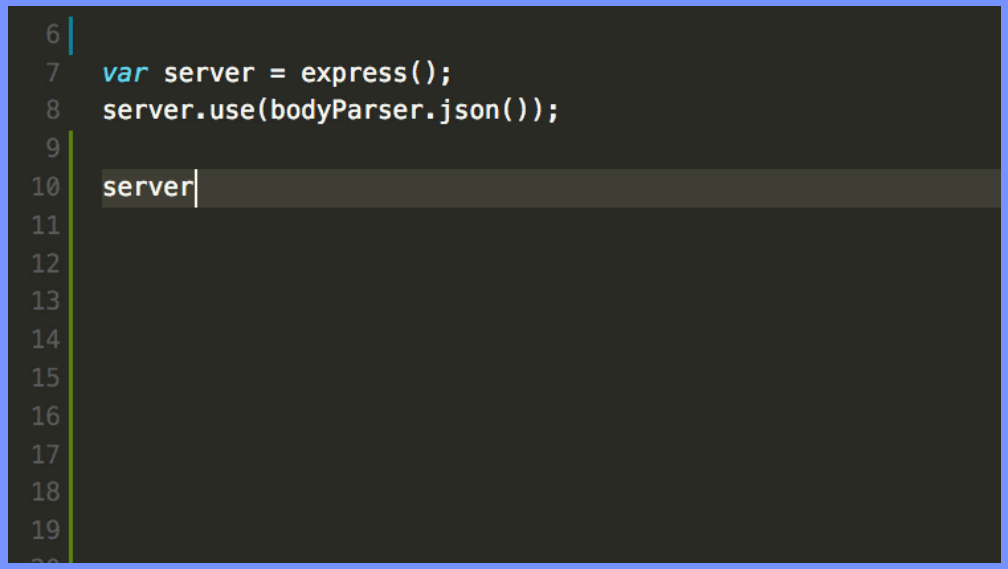
```
interface IGreetingProps extends IWithUserDataProps {
  message: string;
}
```

Interfaces x TDD

✅ Create more portable code

✅ Write better documentation

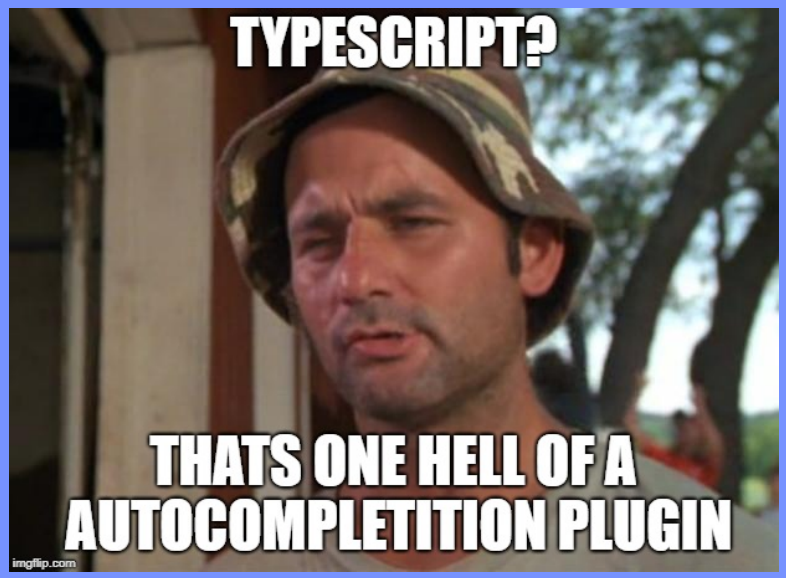
❓ Increase dev speed


var server = express();
server.use(bodyParser.json());
server

Reads 3rd party declaration files

Gives you cleaner errors

Helps you refactor


TYPESCRIPT?
THATS ONE HELL OF A
AUTOCOMPLETITION PLUGIN
imgflip.com

Try it for yourself!

https://typescriptcourses.com/typescript-fundamentals

https://www.typescriptlang.org/play

https://basarat.gitbooks.io/typescript

TypeScript support using Babel 7 #4837

Merged **Timer** merged 1 commit into `facebook:master` from `brunolemos:next-typescript` 4 days ago

THANK YOU!

@KYLIESTEW ON TWITTER

`github.com/kale-stew/typescript-is-awesome`