Training TypeScript

Module: React + TypeScript

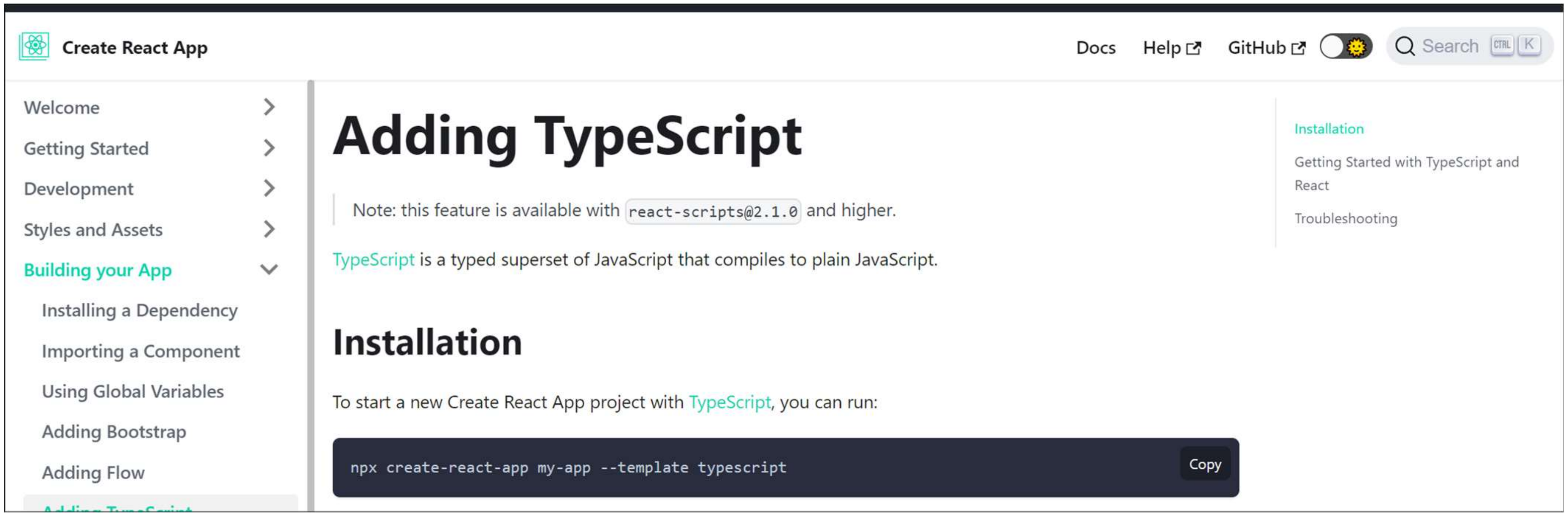
ORBIT GT BELGIUM

Peter Kassenaar

info@kassenaar.com

Not so much slides...

...mostly code



Adding TypeScript

> Note: this feature is available with `react-scripts@2.1.0` and higher.

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript.

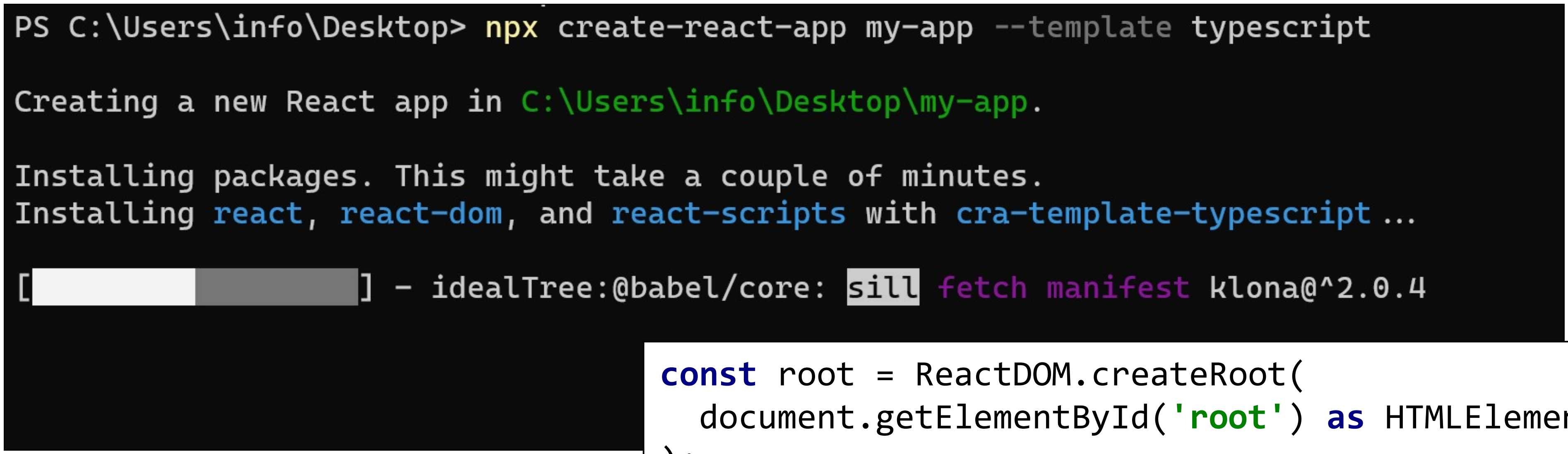
Installation

To start a new Create React App project with TypeScript, you can run:

```
npx create-react-app my-app --template typescript
```

Official docs on React + TypeScript

- You can use CRA to create a React + TypeScript App
  - `npx create-react-app my-app --template typescript`



```
PS C:\Users\info\Desktop> npx create-react-app my-app --template typescript

Creating a new React app in C:\Users\info\Desktop\my-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template-typescript...

[                ] - idealTree:@babel/core: sill fetch manifest klona@^2.0.4
```

```
const root = ReactDOM.createRoot(
  document.getElementById('root') as HTMLElement
);
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

https://create-react-app.dev/docs/adding-typescript/

Using Vite

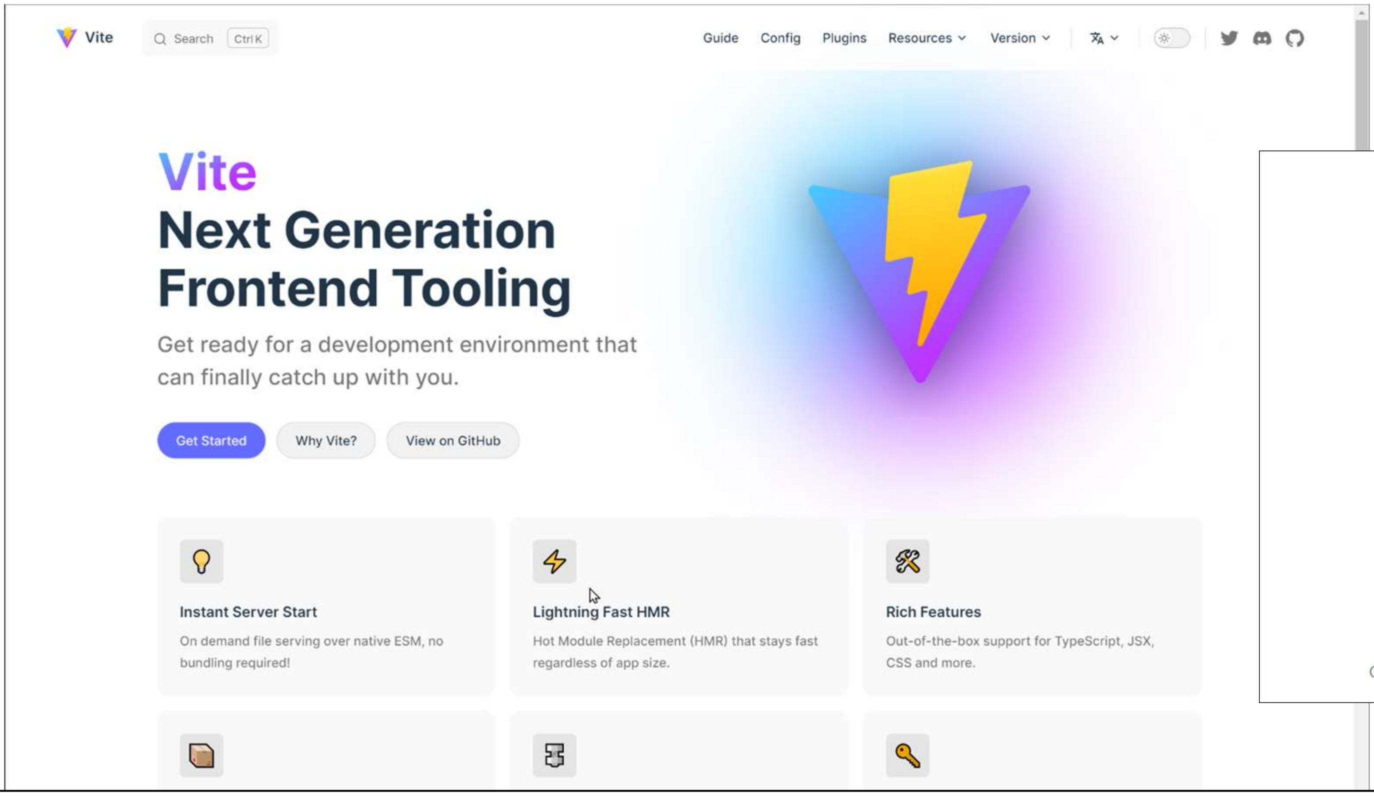
- Vite is a popular tool for scaffolding new projects lately. It also supports React + TypeScript.
- https://vitejs.dev/

```
npm init vite@latest vite-react-app -- --template react-ts
```

```
cd to correct directory
…
Npm install
```

```
npm -E react-router-dom@5.3.0 @types/react-router-dom@5.3.3
```

More on Vite...



Vite

Next Generation Frontend Tooling

Get ready for a development environment that can finally catch up with you.

Instant Server Start

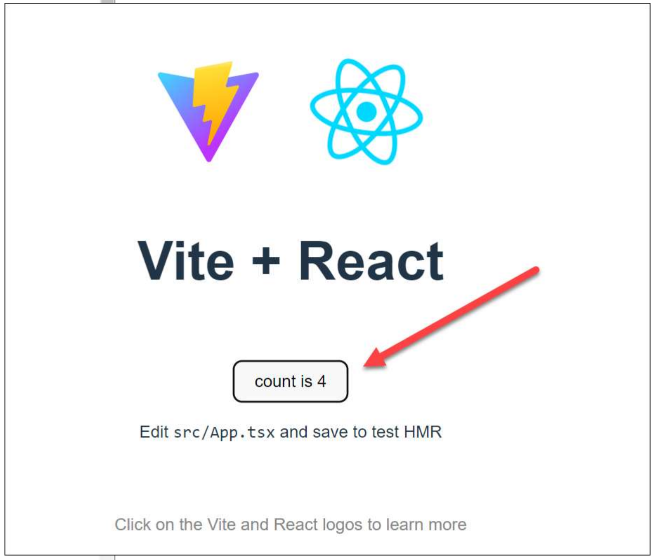
On demand file serving over native ESM, no bundling required!

Lightning Fast HMR

Hot Module Replacement (HMR) that stays fast regardless of app size.

Rich Features

Out-of-the-box support for TypeScript, JSX, CSS and more.



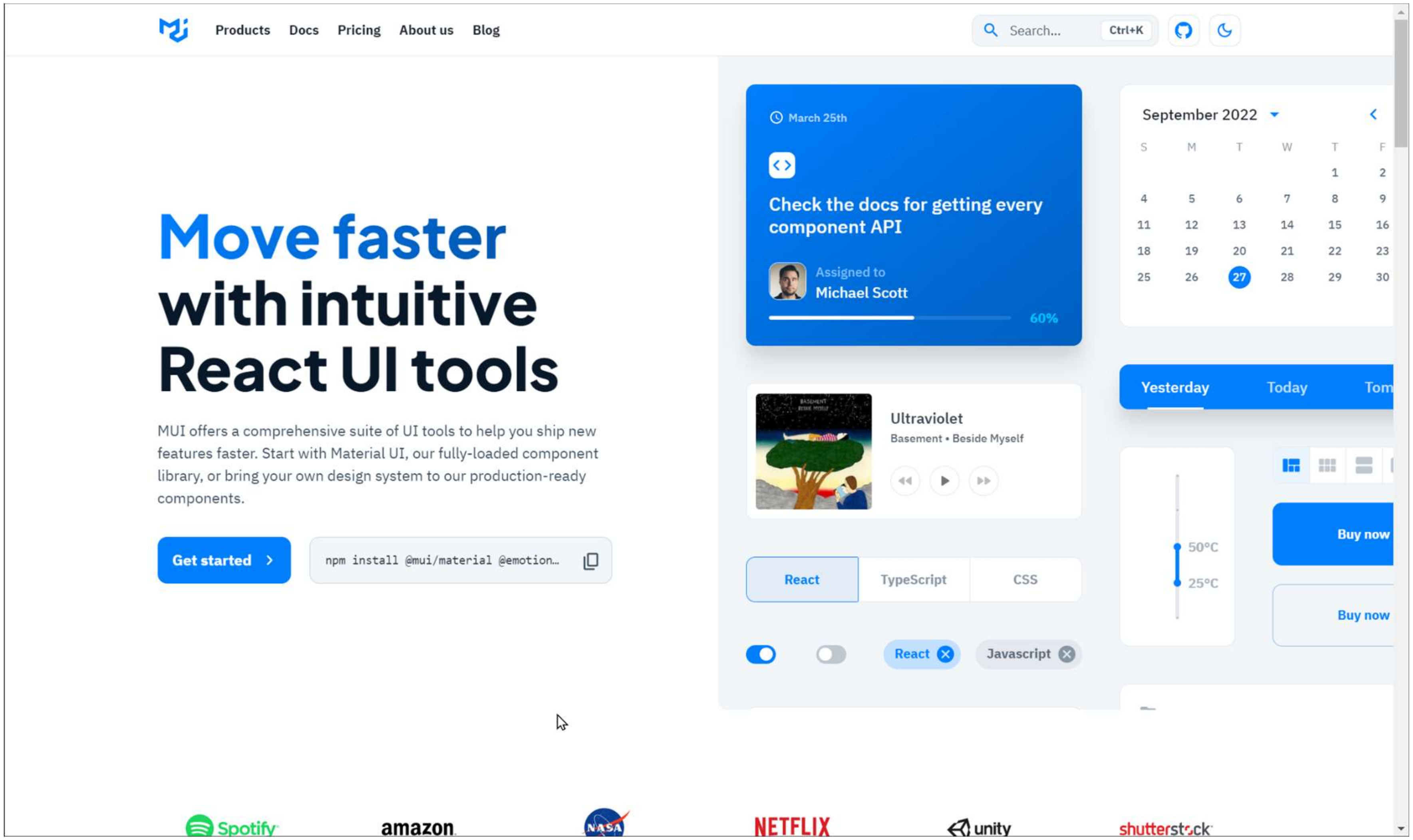
Vite + React

Edit `src/App.tsx` and save to test HMR

Click on the Vite and React logos to learn more

```
ReactDOM.createRoot(document.getElementById('root') as HTMLElement)
  .render(
    <React.StrictMode>
      <App />
    </React.StrictMode>
  )
```

Adding a UI library - MUI



Products Docs Pricing About us Blog

Search... Ctrl+K

Move faster with intuitive React UI tools

MUI offers a comprehensive suite of UI tools to help you ship new features faster. Start with Material UI, our fully-loaded component library, or bring your own design system to our production-ready components.

Get started

```
npm install @mui/material @emotion…
```

https://mui.com/

Adding the library to your project

```
npm install @mui/material @emotion/react @emotion/styled
```

```
"dependencies": {
  "@emotion/react": "^11.10.4",
  "@emotion/styled": "^11.10.4",
  "@mui/material": "^5.10.5",
…
```

See Docs for more information on using fonts, icons, components and so on…

https://mui.com/material-ui/getting-started/overview/

Creating a Custom Component

- See code:

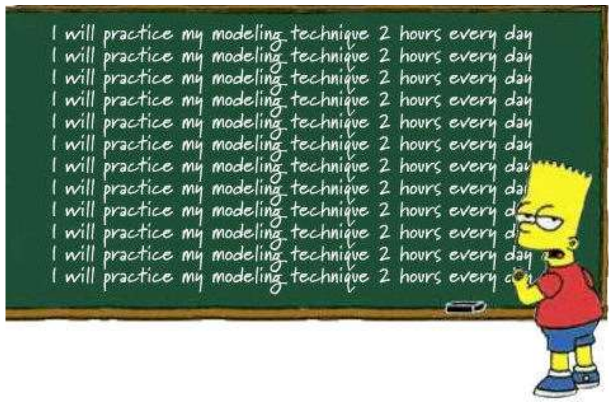
```
import React from 'react';
import {Button, ButtonProps} from "@mui/material";
// 0. If we want to type the attributes for our button,
// it is very common to use types like these:
type Props={
    color: 'primary' | 'secondary' | 'success',
    children: React.ReactNode // text inside <BrandButton>...</BrandButton>
}

// …
```

../sample-projects/90-react-typescript-brandbutton-mui/BrandButton.tsx

Workshop

- Create a new TypeScript project using CRA or Vite
- Add the MUI library + icons
- Create a new component (for instance a <CompanyButton>)
- Use the `ButtonProps` and a Mapped type. We used `Omit<T>` and/or `Required<T>`
- Experiment with some other components like
  - `Slider`
  - `Dialog`
  - `Accordion`
- Create custom properties, using Utitlity Types
- Example code: `../sample projects/90-react….`
- Docs: https://mui.com/material-ui/



More info on React + TypeScript

- …tbd

https://www.youtube.com/results?search_query=typescript+conditional+types