Let's create a TypeScript project by using `create-react-app`, then import `antd` step by step.

> *We build `antd` based on latest stable version of TypeScript (`>=4.0.0`), please make sure your project dependency matches it.*

Install and Initialization

Ensure your system has installed latest version of yarn or npm.

Create a new cra-template-typescript project named `antd-demo-ts` using yarn.

```
$ yarn create react-app antd-demo-ts --template typescript
```

If you are using npm (we will use yarn in the following instructions, it's ok to replace yarn with npm)

```
$ npx create-react-app antd-demo-ts --template typescript
```

Then we go inside `antd-demo-ts` and start it.

```
$ cd antd-demo-ts
$ yarn start
```

Open browser at http://localhost:3000/, it renders a header saying "Welcome to React" on the page.

Import antd

```
$ yarn add antd
```

Modify `src/App.tsx`, import Button component from `antd`.

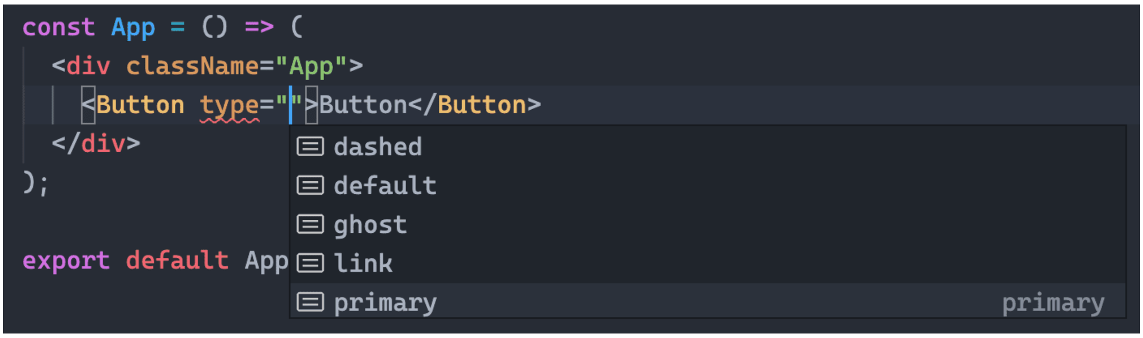
```
import React, { FC } from 'react';
import { Button } from 'antd';
import './App.css';

const App: FC = () => (
  <div className="App">
    <Button type="primary">Button</Button>
  </div>
);

export default App;
```

Add `antd/dist/antd.css` at the top of `src/App.css`.

```
@import '~antd/dist/antd.css';
```

OK, reboot with `yarn start`, you should now see a blue primary button displayed on the page. Next you can choose any components of `antd` to develop your application. Visit other workflows of `create-react-app` at it's User Guide.

`antd` is written in TypeScript with complete definitions, try out and enjoy the property suggestion and typing check.



```
const App = () => (
  <div className="App">
    <Button type="">Button</Button>
  </div>
);

export default App
```

> *Don't install `@types/antd`.*

Advanced Guides

In the real world, we usually have to modify default webpack config for custom needs such as themes. We can achieve that by using craco which is one of create-react-app's custom config solutions.

Install craco and modify the `scripts` field in `package.json`.

```
$ yarn add @craco/craco
```

```
/* package.json */
"scripts": {
-   "start": "react-scripts start",
-   "build": "react-scripts build",
-   "test": "react-scripts test",
+   "start": "craco start",
+   "build": "craco build",
+   "test": "craco test",
}
```

Then create a `craco.config.js` at root directory of your project for further overriding.

```
/* craco.config.js */
module.exports = {
  // ...
};
```

Customize Theme

According to the Customize Theme documentation, we need to modify less variables via loader like less-loader. We can use craco-less to achieve that,

First we should modify `src/App.css` to `src/App.less`, then import less file instead.

```diff
/* src/App.ts */
- import './App.css';
+ import './App.less';
```

```diff
/* src/App.less */
- @import '~antd/dist/antd.css';
+ @import '~antd/dist/antd.less';
```

Then install `craco-less` and modify `craco.config.js` like below.

```
$ yarn add craco-less
```

```js
const CracoLessPlugin = require('craco-less');

module.exports = {
  plugins: [
    {
      plugin: CracoLessPlugin,
      options: {
        lessLoaderOptions: {
          lessOptions: {
            modifyVars: { '@primary-color': '#1DA57A' },
            javascriptEnabled: true,
          },
        },
      },
    },
  ],
};
```

By adding `modifyVars` option of less-loader here, we should see a green button rendered on the page after rebooting the server now.

We provide built-in dark theme and compact theme in antd, you can reference to Use dark or compact theme.

> *You could also try react-app-rewired and customize-cra to customize create-react-app webpack config like craco did.*

Alternative ways

Follow manual in Adding TypeScript to setup TypeScript development environment if you already create a project by Use in create-react-app.

- Create React apps (with Typescript and antd) with no build configuration
- react-app-rewire-typescript

- ts-import-plugin
- Migrating from create-react-app-typescript to Create React App