

Introducción a React.Js

Módulo 05

Material Design

Material-UI es una librería complementaria de React que nos provee de componentes que implementan el diseño de materiales de Google.

Instalación

Instale los archivos fuente de Material-UI a través de npm 3 :

npm install --save @material-ui/core

Material-UI fue diseñado con la fuente Roboto en mente. Así que asegúrese de seguir estas instrucciones. Por ejemplo, a través de Google Web Fonts:

***< link rel = "stylesheet"
href = "https://fonts.googleapis.com/css?family=Roboto:300,400,500" >***

Instalación

Para usar el componente de icono de fuente, primero debe agregar la fuente de iconos de Material. Aquí hay algunas instrucciones sobre cómo hacerlo. Por ejemplo, a través de Google Web Fonts:

```
< link rel = "stylesheet"  
href = "https://fonts.googleapis.com/icon?family=Material+Icons" >
```

Podemos sino optar por los otros tipos de iconos. Para utilizar los iconos de Material SVG, primero debe instalar el paquete @ material-ui / icons:

```
npm install @ material-ui / icons
```

Uso

Los componentes de Material UI funcionan de forma aislada.

Son autosustentables e inyectarán, y solo inyectarán, los estilos que necesitan para mostrar. No dependen de ninguna hoja de estilo global, como `normalize.css`, aunque Material-UI proporciona un componente `CssBaseline` opcional.

Aquí hay un ejemplo rápido para comenzar, es literalmente todo lo que necesita:

Uso

```
import React from 'react';
import ReactDOM from 'react-dom';
import Button from '@material-ui/core/Button';

function App() {
  return (
    <Button>
      Hola Mundo!
    </Button>
  );
}
```

Themes

El tema especifica el color de los componentes, la oscuridad de las superficies, el nivel de sombra, la opacidad adecuada de los elementos de tinta, etc.

Los temas te permiten aplicar un tono uniforme a tu aplicación. Le permite personalizar todos los aspectos de diseño de su proyecto para cumplir con las necesidades específicas de su negocio o marca.

Para promover una mayor coherencia entre las aplicaciones, los tipos de tema claro y oscuro están disponibles para elegir. Por defecto, los componentes usan el tipo de tema claro.

Theme Provider

Si desea personalizar el tema, debe usar el componente `MuiThemeProvider` para insertar un tema en su aplicación. Sin embargo, esto es opcional; Los componentes Material-UI vienen con un tema predeterminado.

`MuiThemeProvider` se basa en la función de contexto de React para pasar el tema a los componentes, por lo que debe asegurarse de que `MuiThemeProvider` sea uno de los componentes que está tratando de personalizar.

Cambiar las variables de configuración del tema es la forma más efectiva de adaptar la IU de material a sus necesidades. Las siguientes secciones cubren las variables temáticas más importantes:

Theme Provider

Si desea personalizar el tema, debe usar el componente `MuiThemeProvider` para insertar un tema en su aplicación. Sin embargo, esto es opcional; Los componentes Material-UI vienen con un tema predeterminado.

`MuiThemeProvider` se basa en la función de contexto de React para pasar el tema a los componentes, por lo que debe asegurarse de que `MuiThemeProvider` sea uno de los componentes que está tratando de personalizar.

Theme Provider

Cambiar las variables de configuración del tema es la forma más efectiva de adaptar la IU de material a sus necesidades. Las siguientes secciones cubren las variables temáticas más importantes:

- Palette
- Type (light / dark theme)
- Typography
- Other variables
- Custom variables

Theme Provider

```
import React from 'react';
import { render } from 'react-dom';
import { MuiThemeProvider, createMuiTheme } from '@material-ui/core/styles';
import Root from './Root';

const theme = createMuiTheme();

function App() {
  return (
    <MuiThemeProvider theme={theme}>
      <Root />
    </MuiThemeProvider>
  );
}

render(<App />, document.querySelector('#app'));
```

Podemos crear un tema personalizado cambiando las propiedades de cualquiera de estas variables o usar un tema predefinido por Material UI utilizando el método `createMuiTheme()`

Podemos crear un tema personalizado cambiando las propiedades de cualquiera de estas variables o usar un tema predefinido por Material UI utilizando el método `createMuiTheme()`

¡Muchas gracias!

¡Sigamos trabajando!