

# React + typescript + antd开发前端应用（六）应用基本框架组件

 ep76

关注

上一篇：[React + typescript + antd开发前端应用（五）React路由](#)

▲

赞同

🚩

分享

对于一些事务性的后台管理系统，我们经常需要类似一个多页签页面的UI界面，如下图所示：



后台管理系统页面分区图

分析上图，根据主键分布，可以把页面组件划分为Layout、menu和tabs三个组件。antd已经提供了对应的组件，只是需要我们一步步组装成我们想要的系统主界面。

为了让and的框架组件高度占据怎么屏幕，需要在index.tsx文件中引入antd的reset组件：

```
import 'antd/dist/reset.css';
```

该CSS会对HTML元素样式进行重置，比如设置margin、padding为0、html、body元素高度为100等等。

## 1、构建Layout组件

创建src\layout\AppLayout.tsx文件，编辑文件内容如下：

```
import { Layout, ConfigProvider, theme } from "antd";
const { Header, Sider, Content } = Layout;

function AppLayout() {
  return (
    <ConfigProvider theme={{ algorithm: theme.darkAlgorithm }}>{/** 使用antd的暗黑模式
      <Layout>
        <Header>Header部分</Header>
        <Layout>
          <Sider width={190} style={{overflow: "auto"}}>
            <span style={{color: "white"}}>菜单区</span>
          </Sider>
          <Content
```

▲ 赞同 ▼

● 添加评论

🚩 分享

♥ 喜欢

★ 收藏

📄 申请转载

...

```

        <span style={{color: "white"}}>页签区</span>
      </Content>
    </Layout>
  </Layout>
</ConfigProvider>
);
};

export default AppLayout;

```

修改应用入口文件index.tsx文件内容如下:

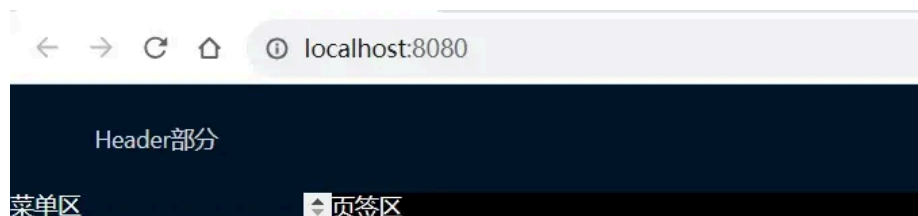
```

import React from 'react';
import { createRoot } from 'react-dom/client';
import AppLayout from './layout/AppLayout';//引入Layout组件
import 'antd/dist/reset.css';//引入antd的reset.css重置样式

createRoot(document.getElementById('root') as HTMLElement).render(
  <React.StrictMode>
    <AppLayout /> {/** 使用Layout组件 */}
  </React.StrictMode>
);

```

回到浏览器查看结果,发现Layout控件并没有像期望的那样占满整个屏幕,如下图所示:



知乎 @ep76

Layout组件未占满body剩余空间

创建src\layout\AppLayout.css文件,并添加CSS代码如下:

```

#root, main, .ant-layout {
  height: 100%;
}

```

这里之所有多个选择器的样式高度需要设置为100%是因为子元素高度是百分比,而这个百分比是相对于父元素的,所以需要把Layout组件的父元素的高度也要设置为100%。修改src\layout\AppLayout.tsx文件,导入刚刚定义的AppLayout.css文件:

```

import { Layout, ConfigProvider, theme } from "antd";
import './AppLayout.css';//添加import css代码
const { Header, Sider, Content } = Layout;

```

再次回到浏览器页面可以看到Layout组件占满了剩余的屏幕空间。

## 2、构建菜单组件

现在，创建src\layout\AppMenu.tsx文件，文件内容如下：

```
import React from 'react';
//import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-design/icons';
import type { MenuProps } from 'antd';//引入菜单项的数据类型
import { Menu } from 'antd';

type MenuItem = Required['items'][number];

function getItem(label: React.ReactNode, key: React.Key, icon?: React.ReactNode, child
    return {key, icon, children, label, type} as MenuItem;
}

const items: MenuProps['items'] = [
    getItem('系统菜单', 'systemMenu', null, [
        getItem('功能一', 'systemMenu_menu1'),
        getItem('功能二', 'systemMenu_menu2'),
        getItem('功能三', 'systemMenu_menu3')
    ])
];

function AppMenu() {
    const onClick: MenuProps['onClick'] = (e) => {
        console.log('click ', e);
    };
    return (
        <Menu onClick={onClick} defaultSelectedKeys={['systemMenu_menu1']} defaultOpen
    );
}

export default AppMenu;
```

这个文件中出现了很对typescript的语法形式，希望大家仔细分析以下这些代码的细节，好好理解“typescript是JavaScript的超集，是具有强类型的JavaScript”这句话的含义。修改src\layout\Applayout.tsx文件，引入AppMenu组件：

```
import { Layout, ConfigProvider, theme } from "antd";
import './Applayout.css';
import AppMenu from "../AppMenu";//引入菜单组件
const { Header, Sider, Content } = Layout;

function Applayout() {
    return (
        <ConfigProvider theme={{ algorithm: theme.darkAlgorithm }}>
            <Layout>
                <Header>Header部分</Header>
                <Layout>
                    <Sider width={190} style={{overflow: "auto"}}>
                        <AppMenu />{/* 在左侧边栏应用菜单组件 */}
                    </Sider>
                    <Content>
                        <span style={{color: "white"}}>页签区</span>
                    </Content>
                </Layout>
            </Layout>
        </ConfigProvider>
    );
};

export default Applayout;
```

在浏览器中查看AppMenu菜单组件是否被成功引入AppLayout组件中。

### 3、创建多页签组件

创建src\layout\AppTabs.tsx文件，编辑文件内容如下：



```
import React, { useRef, useState } from 'react';
import { Tabs } from 'antd';

type TargetKey = React.MouseEvent | React.KeyboardEvent | string;

const initialItems = [
  { label: '页签一', children: '这是页签一的内容', key: '1', closable: false },
  { label: '页签二', children: '页签只能是文本内容吗?', key: '2' },
  { label: '页签三', children: '还需要改进啊', key: '3' },
];

function AppTabs() {
  const [activeKey, setActiveKey] = useState(initialItems[0].key);
  const [items, setItems] = useState(initialItems);
  const newTabIndex = useRef(0);

  const onChange = (newActiveKey: string) => {
    setActiveKey(newActiveKey);
  };

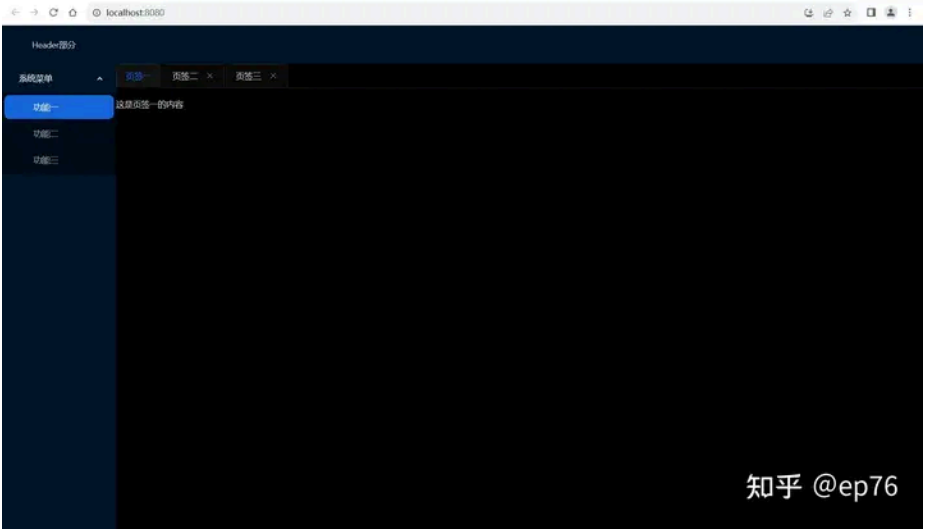
  function add() {
    const newActiveKey = `newTab${newTabIndex.current++}`;
    const newPanes = [...items];
    newPanes.push({ label: 'New Tab', children: 'Content of new Tab', key: newActiveKey });
    setItems(newPanes);
    setActiveKey(newActiveKey);
  };

  function remove(targetKey: TargetKey) {
    let newActiveKey = activeKey;
    let lastIndex = -1;
    items.forEach((item, i) => {
      if (item.key === targetKey) {
        lastIndex = i - 1;
      }
    });
    const newPanes = items.filter((item) => item.key !== targetKey);
    if (newPanes.length && newActiveKey === targetKey) {
      if (lastIndex >= 0) {
        newActiveKey = newPanes[lastIndex].key;
      } else {
        newActiveKey = newPanes[0].key;
      }
    }
    setItems(newPanes);
    setActiveKey(newActiveKey);
  };

  function onEdit(targetKey: React.MouseEvent | React.KeyboardEvent | string, action: string) {
    if (action === 'add') {
      add();
    } else {
      remove(targetKey);
    }
  };
}
```

```
return (  
  <Tabs  
    type="editable-card"  
    onChange={onChange}  
    activeKey={activeKey}  
    onEdit={onEdit}  
    items={items}  
    hideAdd={true}  
  />  
);  
};  
  
export default AppTabs;
```

然后修改src\layout\AppLayout.tsx文件，引入AppTabs组件，并在<Content>标签中使用这个组件，代码和在AppLayout中使用AppMenu类似，就不重复粘贴代码了。最终效果如下图所示：





完成以上工作后，应用的基本框架就开发完成了。但者只是万里长征完成了第一步，后续还有很多改进的点，比如点击菜单后要添加页签并在页签体中显示记录，页签体不能只显示文本内容，要能动态展现子组件等等。

下一篇：[React + typescript + antd开发前端应用（七）添加菜单点击功能](#)

编辑于 2023-10-06 04:01 · IP 属地上海

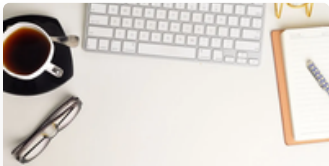
[React](#)   [Ant Design](#)   [TypeScript](#)

 欢迎参与讨论





推荐阅读



Vite + React 组件开发实践

阿里云开发者



我的react组件化开发道路(一)  
一脸懵比的踩坑中

四目大大

发表于极乐科技



React 之我见：JSX，虚拟  
DOM，Diff 算法，setState...

LeviDing

【腾  
框架

作者：  
出一  
方介  
ui把r  
等同  
dom  
腾讯i