

第48期-基于React的前端用户登录与授权

2021年11月30日 20:27

添加 IdentityServer 客户端

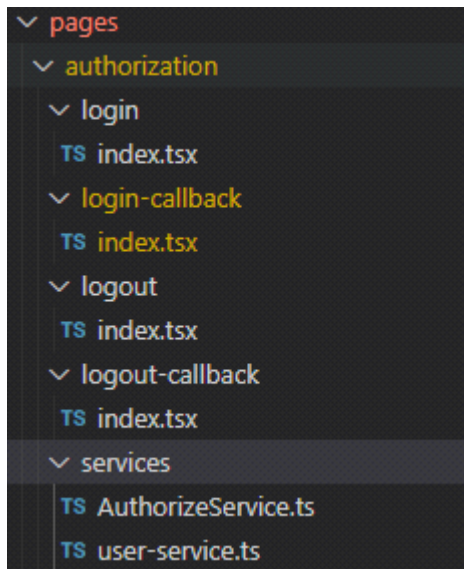
```
new Client
{
    ClientId = "devicecenterweb",
    ClientName = "Device Center Web",
    AllowedGrantTypes = GrantTypes.Code,
    AllowAccessTokensViaBrowser = true,
    RequireClientSecret=false,
    AlwaysSendClientClaims=true,
    AlwaysIncludeUserClaimsInIdToken=true,
    RequireConsent = false,
    RedirectUri = {
        "http://localhost:8000/authorization/login-callback",
    },
    PostLogoutRedirectUri = {
        "http://localhost:8000/authorization/logout-callback",
    },
    AllowOfflineAccess=true,
    RequirePkce = true,
    AllowedScopes =
    {
        IdentityServerConstants.StandardScopes.OpenId,
        IdentityServerConstants.StandardScopes.Profile,
        "userinfo",
        "openapi",
        "identityserver",
        "devicecenter"
    },
    AllowedCorsOrigins = { "http://localhost:8000"},
    AccessTokenLifetime = 600
}
```

设计客户端登录与回调

<https://github.com/IdentityModel/oidc-client-js>

```
npm install oidc-client --save
```

```
"oidc-client": "^1.11.5",
```



屏幕剪辑的捕获时间: 2021-12-02 21:32

```
{
  path: '/authorization',
  layout: false,
  routes: [
    {
      name: 'login',
      path: '/authorization/login',
      component: './authorization/login',
    },
    {
      path: '/authorization/login-callback',
      component: './authorization/login-callback',
    },
    {
      path: '/authorization/logout',
      component: './authorization/logout',
    },
    {
      path: '/authorization/logout-callback',
      component: './authorization/logout-callback',
    },
    {
      component: './404',
    }
  ],
},
```

未登陆时跳转到登录页面

```
export const layout: RunTimeLayoutConfig = ({ initialState }) => {
  return {
    headerTitleRender: (logo) => (<a>{logo}<h1><FormattedMessage
id='site.title' /></h1></a>),
    pageTitleRender: (_props, _defaultPageTitle, info) => `
${info?.pageName}-${useIntl().formatMessage({ id: 'site.title' })}`,
    rightContentRender: () => <RightContent />,
    disableContentMargin: false,
    footerRender: () => <Footer />,
    onPageChange: () => {
      const { location } = history;
```

```

        // 如果没有登录, 重定向到 login
        if (!initialState?.currentUser && !
location.pathname.startsWith('/authorization')) {
            history.push(`/authorization/login?returnUrl=
${encodeURIComponent(window.location.href)}`);
        }
    },
    links: [],
    menuHeaderRender: undefined,
    // 自定义 403 页面
    // unAccessible: <div>unAccessible</div>,
    ...initialState?.settings,
  };
};
};

```

```

export async function getInitialState(): Promise<{
  settings?: Partial<LayoutSettings>;
  appConfigs?: API.ApplicationConfiguration;
  currentUser?: User;
}> {
  const appConfigs = await getConfigurations();
  return {
    currentUser: await userManager.getUser() ?? undefined,
    settings: {},
    appConfigs: appConfigs,
  };
}

```

登录后显示用户名和退出登录

```
const loginOut = async () => {
```

```
  history.push(`/authorization/logout?returnUrl=
```

```
  ${encodeURIComponent(window.location.href)}`);
```

```
};
```

在 Visual Studio Code 中调试 React 应用

<https://www.cnblogs.com/luxx/p/6708895.html>