**蓝源卫士：功能描述**

（本文档不外流）

整个项目共有7大页面正在启用，1个页面暂时未启用。项目实现了国际化，支持简体中文和English两种语言。

项目所有的请求都基于https://www.deepfoss.com这个baseURL。

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **页面** | **View name In English** | **描述** |
| 1 | 登录 | Login | 账号登录 |
| 2 | 注册 | Sign\_Up | 账号注册 |
| 3 | 仪表盘（首页） | main | 展示“我的项目”，并能转到单项目管理页面进行项目的管理。 |
| 4 | 项目细节 | Detail | 通过仪表盘页面展现，主要功能为删、查 |
| 5 | 扫描项目 | scans | 扫描项目文件，并上传 |
| 6 | 项目报告 | reports | 查看项目报告列表，支持导出Excel |
| 7 | 用户信息 | admin | 账号管理页面，预配备有修改密码功能 |
| 8 | 管理仓库 | manage | 搭建中，未启用 |

下面是对项目功能具体的理解。

1. **登录**

为React.FC泛型Login函数组件类型为PageProps（继承UmiJS的ConnectProps）。填写登录信息后点击登录触发onFinsh事件。

》》onFinsh事件中：if判断信息是否为空，是否勾选协议-->dispatch进行登录请求【参数为一个{}】

》》》》：dispatch后面加“！”是因为定义的类型PageProps中dispatch()并不是必有项，加“！”是人为确定必有该项。

》》》》》》Dispatch 参数对象中包含type：请求路径，payload：请求负载，callback方法。

》》》》》》》》Callback接收响应信息，判断请求状态，如成功则提示成功信息，替换当前的 history 记录，跳转到指定的 url：'/'，不会向 history 添加新的记录。否则提示报错信息。

1. **注册**

》》全局就一个Login函数组件，包裹antd的标签。

》》完成输入之后，点击按钮，触发onFinish事件，传入表单props.value。

》》》》onFinish方法中，首先进行if判断验证.

》》》》》》触发Toast.show（属于Antd），轻提示。

【参数为一个对象】

{

icon: 'loading', // 图标

maskClickable: false, // 轻提示不可点击

duration: 0, // 持续时间

}

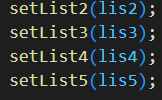
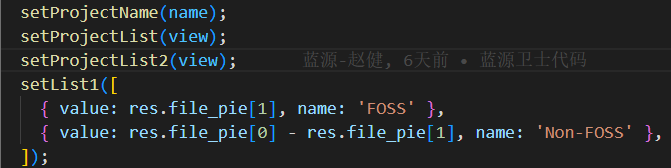
》》》》》》Request 【参数有二：url，params】发送POST请求，路径是‘/local2/signup’，附带数据：email，passwd，user（使用邮箱作为用户名传递）。

》》》》》》接收响应，解除加载状态Toast.clear()。判断状态是否为success，是则有提示信息，并在路由中添加一条记录，并转到登录界面。否则抛出异常并给出错误提醒。

》》这项功能用到了useIntl中的formatMessage进行具体的值绑定。Classnames进行classname拼接。参数校验(validator)

1. **主页功能——“我的项目”管理，主要功能为删、查**

getList方法，无参，向'/local2/getallprojectsview'发送POST请求，负载user和token，将返回成功的数据附加上用户user识别存入localStorage，并展示项目信息。这里的信息会存入5个list中，并传入到Echarts的仪表盘组件中用来展示。



getProjectStatus，无参，向'/local2/checkuserprojects'发送POST请求，负载user和token，将返回成功的数据存入state之statuslist和projectname2中，组件通过状态值绑定数据，状态更新组件更新。

useEffect

DashboardInfoPage（仪表盘信息页面），设置一个state值为‘all’

useEffect首先验证登录成员，确认后添加（push）首页路由（‘/’）到history，

仪表盘界面的重新扫描就是更新ScanVisible状态值，触发组件重渲染和请求，删除操作还会会更新localStorage中存放的数据。

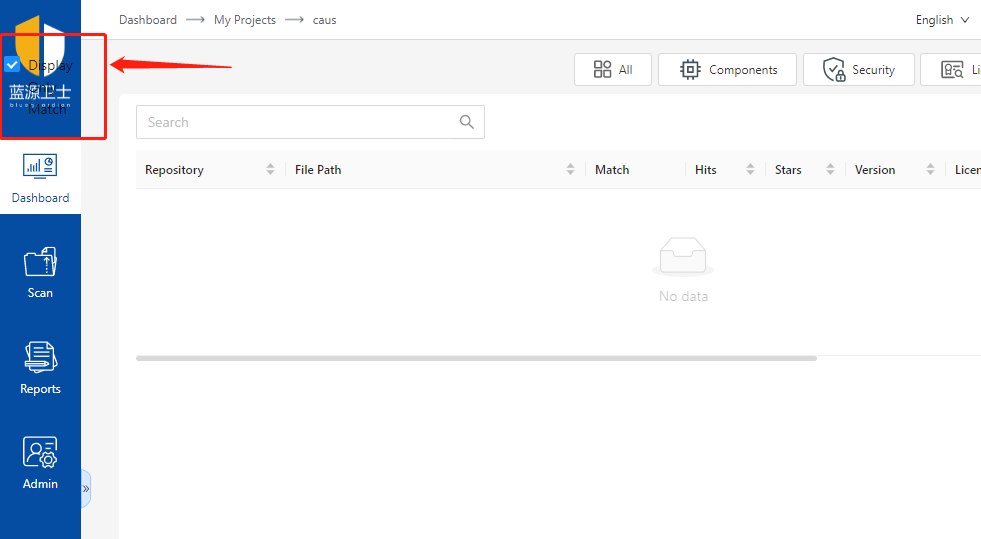
》》Ditail中主要为针对单项目管理的页面组件。

包括all、code、component、CVE、licenses、reports六个方面，每次切换都是从sessionStorage、和localStorage中读取数据，然后把数据通过state接收然后渲染到组件。

》》》》这里每次通过CodePreview查看有关代码的时候依然是会发送请求，接收响应数据渲染到CodePreview组件中。在code、reports、licenses、components中使用到了这个组件。

》》》》除了all外，其他五个部分都有查询的功能

》》》》code页面有一个侧边栏，支持鼠标拖动。但收起来的时候发现有个小问题（如下图所示）。另外这个侧边栏用鼠标拖动的时候有卡顿的感觉。页面中还可以选择展示数量，通过一个下拉列表选择。



1. **登出**

Components》Top》index.tsx

为登出设置了一个state值：logoutVisible，默认值为false。当点击“退出登录”时触发按钮点击事件，修改state值为true。

》》logoutVisible同时也作为了Modal组件的visible值，由false变为true后退出登录的确认提示转为可见。

》》》》当在Modal中点击取消时，修改logoutVisible为false。

》》》》当点击确认后，触发点击时间onOk，向后台发出post请求，路径为‘/local2/logout’，附载为user、token。替换当前的 history 记录，跳转到指定的 url：'/login'。并使用sessionStorage.clear()清除session中存放的数据，完成登出。

1. **上传项目**

功能体现在ScansPage函数组件中。定义了两个useEffect，触发isFinished时会立即执行以下操作（但是这个方法暂时没有启用）：

1. 从session中解析出member，判断成员变量menber，若为空或null，则跳转到登录界面。若不为空，则继续下面的步骤。这一步只要member改变都会执行一次
2. 使用user和token信息向'/local2/checkuserprojects'发送POST请求。开启轻提醒。
3. 接收到response信息后，判断res是否包含started字符串，若为真则将state之up由默认值false改为true，否则依然赋值为false
4. 异步执行catch，抛出错误信息。

》》上传文件会触发beforeUpload通过setFile改变state之file的值，并促使Dragger（Antd组件组分）渲染，另外提供了删除方法onRemove（就是把file重新设为undefined）。

点击提交之后会触发onFinish：

判断标题和文件是否为空、创建formData对象，并为其添加属性，作为请求的负载。开启轻提示，并向'/local2/submitproject'发送POST请求。上传成功后重置组件以及两个state值。若请求失败则抛出错误，不改变state。

这里使用了两个state和两个useEffect。*其中一个useEffect没有作用，内部代码被注释掉了，就是isFinished。*使用到了Antd的form实例。

1. **导出报告为Excel**

》》一个名为CVEPage的函数式组件。渲染时直接使用api：use Effect，请求后端项目数据，并渲染到页面。**这里的数据是一次性全部获取**。后面的查询其实也是通过请求查询的接口，然后再进行数据对接。实现报告查询功能。

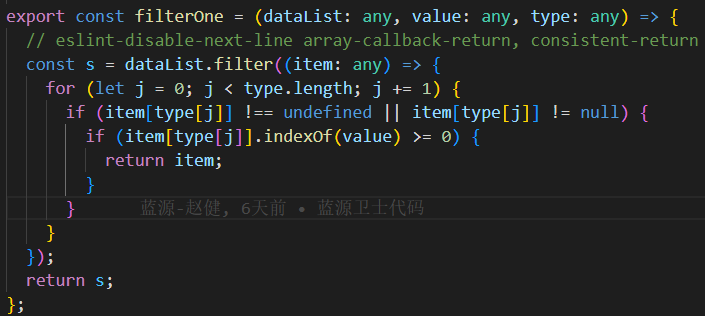
》》导出为Excel，则是引入了一个js包协助实现的（import Table2Excel from 'table2excel.js';）

通过table2Excel实例的导出方法export来导出为xlsx文件。在使用该方法的时候，通过worksheetCreated对表格进行绘制。

1. **一维数组模糊搜索filterOne**

传入数据源（dataList）、搜索值（value，为输入框输入的）、指定想要搜索的字段名（type，类型为数组）。通过循环查询键值，匹配出现搜索框输入的字符串，如果indexOf返回的索引值大于等于0，说明字符串存在，返回对应的字段。这是一个高阶方法。

另外还有一个fileterOne2，多了一个pro参数和判断



1. **重置密码**

功能还没有实现。点击提交只会在控制台输出表单value对象。

1. **代码预览**

用到了react-syntax-highlighter库，用于代码高亮展示，通过props（prop.code即为包含的代码）将接收到的参数渲染到组件中。这里没有使用state等。

以上就是对“蓝源卫士”功能的基本理解。（本文档不会外流）