```
import ReactMarkdown from "react-markdown";
import rehypeRaw from "rehype-raw";
import remarkGfm from "remark-gfm";
import {Prism as SyntaxHighlighter} from "react-syntax-highlighter";
// 设置高亮样式 coldarkDark ,
import {dracula} from "react-syntax-highlighter/dist/cjs/styles/prism";
import React, {useEffect, useState} from "react";
import {Spin} from 'antd'
import axios from "axios";
import {BASE_URL} from '../../utils/request'
import {withRouter} from 'react-router-dom'
import {connect} from "react-redux";
// The default style of markdown-navbar should be imported additionally
import 'markdown-navbar/dist/navbar.css';
import HeadingBlock from "../../components/HeadingBlock";
import MarkdownNavbar from "../../components/MarkDownNavbar";

function FirstLevelNavigation(props) {
  const {mainDataMd, mainDataLoading, location, rootState} = props;
  const [mainDataMdState, setMainDaraMdState] = useState('');
  const [mainDataLoadingState, setMainDataLoadingState] = useState(false);
  useEffect(() => {
    // document.onmousemove = function () {
    //   setShowHomeIcon(true);
    //   setTimeout(() => setShowHomeIcon(false), 2000);
    // };
    async function fetchData() {
      let res = await axios({
        url: `${BASE_URL}/readme.md`,
        method: 'GET'
      });
      setMainDaraMdState(res.data);
    }

    fetchData();
  }, []);
  useEffect(() => {
    setMainDataLoadingState(mainDataLoading);
  }, [mainDataLoading]);
  useEffect(() => {
    let splitPathNameArr = location.pathname.split('/');
    if (splitPathNameArr[2]) {
      let oneTitle = '';
      let secondTitle = '';
      let children = [];
      rootState.forEach(item => {
        if (item.uid.includes(splitPathNameArr[1])) {
          oneTitle = item.title;
          children = item.children;
        }
      });
      children && children.forEach(item => {
        if (item.uid.includes(splitPathNameArr[2])) {
          secondTitle = item.title
        }
      });
      if (oneTitle && secondTitle) {
        async function fetchData() {
          setMainDataLoadingState(true);
          let res = await axios({
            url: `${BASE_URL}/${encodeURIComponent(oneTitle)}/$
{encodeURIComponent(secondTitle)}`,
            method: 'GET'
          });
          setMainDataLoadingState(false);
          setMainDaraMdState(res.data);
        }
        fetchData();
      }
    }
  }, [mainDataMd, location.pathname, rootState]);
  return <>
    <img className={`homeIcon homeIcon_active`}
         onClick={() => window.location.href = '/'}
         src="https://cdn.jsdelivr.net/gh/IceRain-mvc/cdn/img/home.svg" alt=""/>
    <Spin
      delay={'200ms'}
      tip="加载中..."
      spinning={mainDataLoadingState}
    >
      <div className='md_container'>
        <div className='md_content'>
          <ReactMarkdown
            children={mainDataMdState}
            rehypePlugins={[rehypeRaw]}
            remarkPlugins={[remarkGfm]}
            components={{
              h2: ({node, ...props}) => (<HeadingBlock {...props}/>),
              h3: ({node, ...props}) => (<HeadingBlock {...props}/>),
              h4: ({node, ...props}) => (<HeadingBlock {...props}/>),
              h5: ({node, ...props}) => (<HeadingBlock {...props}/>),
              h6: ({node, ...props}) => (<HeadingBlock {...props}/>),
              img: ({node, ...props}) => (<img alt={'图片'} style={{width: '50%'}}
{...props}/>),
              code({node, inline, className, children, ...props}) {
                const match = /language-(\w+)/.exec(className || '');
                return !inline && match ? (
                  <SyntaxHighlighter
                    children={String(children).replace(/\n$/, '')}
                    style={dracula}
                    language={match[1]}
                    PreTag="div"
                    {...props}
                  />
                ) : (
                  <code className={className} {...props}>
                    {children}
                  </code>
                )
              }
            }}
          >{mainDataMdState}</ReactMarkdown>
        </div>
        <div className="md_navigation">
          <div className='md_nav_content'>
            <MarkdownNavbar
              onNavItemClick={(event, element, text) => {
                // console.log(text);
                // window.location.href = `#${text}`
              }}
              updateHashAuto={true}
              declarative={true}
              source={mainDataMdState}/>
          </div>
        </div>
      </div>
    </Spin>
  </>
```

```
}

const mapStateToProps = state => {
  return {
    rootState: state.rootReducers
  }
};
export default connect(mapStateToProps)(withRouter(FirstLevelNavigation));
```