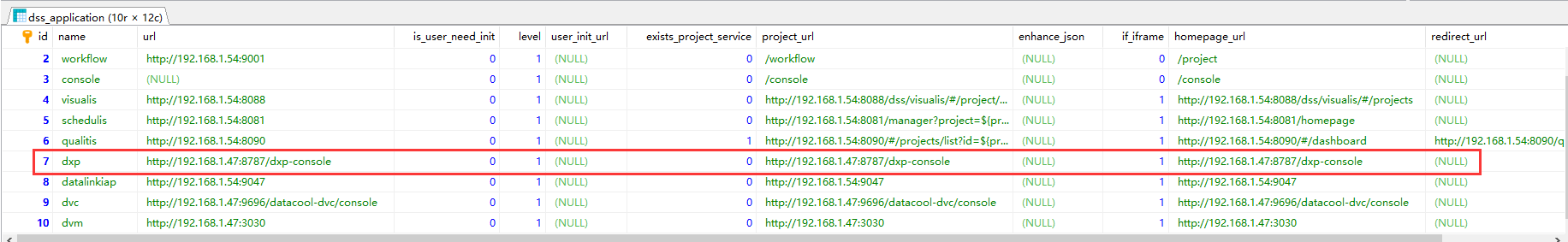
**DSS集成DXP**

1. **前端集成**
2. **数据库集成**
3. 在数据库的dss\_application表中增加dxp的信息，如下图所示：

****

解释：在登陆dss和点击下拉框或首页图标进行跳转时，dss会通过dss-application模块中的

com.webank.wedatasphere.dss.application.restful.ApplicationRestfulApi

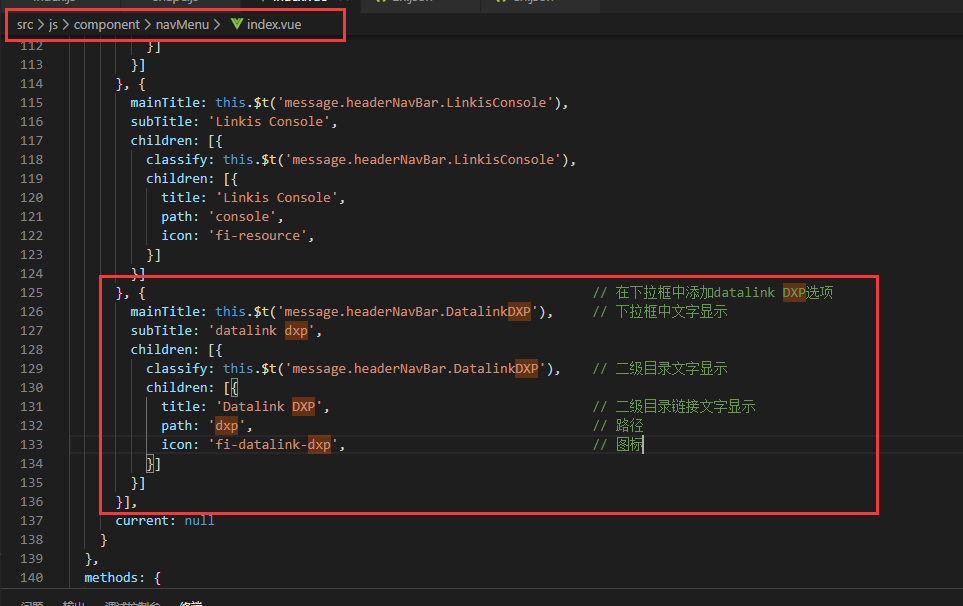
获取dss\_application的所有信息，并通过name字段匹配不同系统。

**二、下拉框集成**

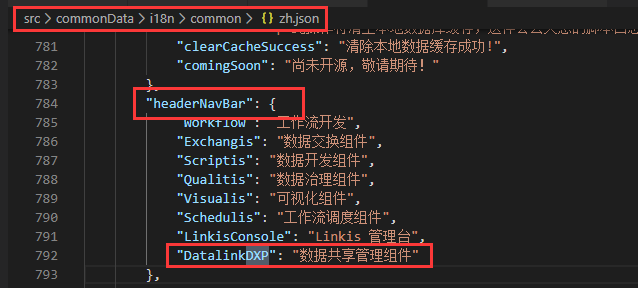
实现下拉框点击跳转到DXP页面，在下拉框中增加DatalinkDXP选项并支持点击跳转。

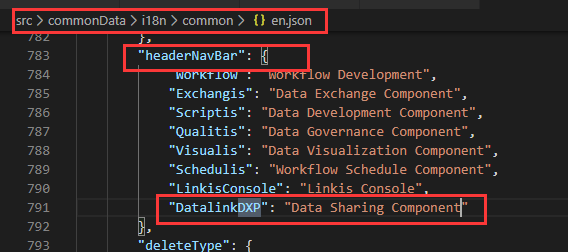


1. 在是src\js\component\navMenu\index.vue中添加dxp的下拉选项

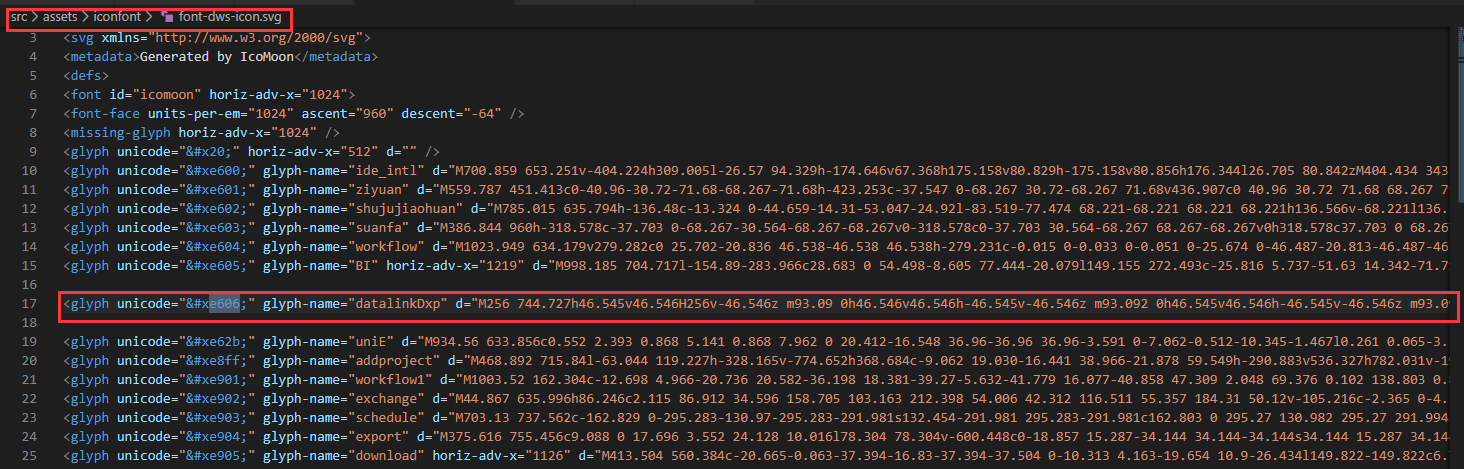


1. 在zh.json和en.json中添加dxp的文字描述

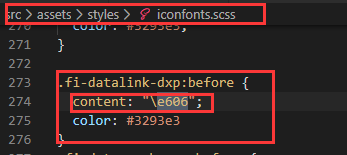




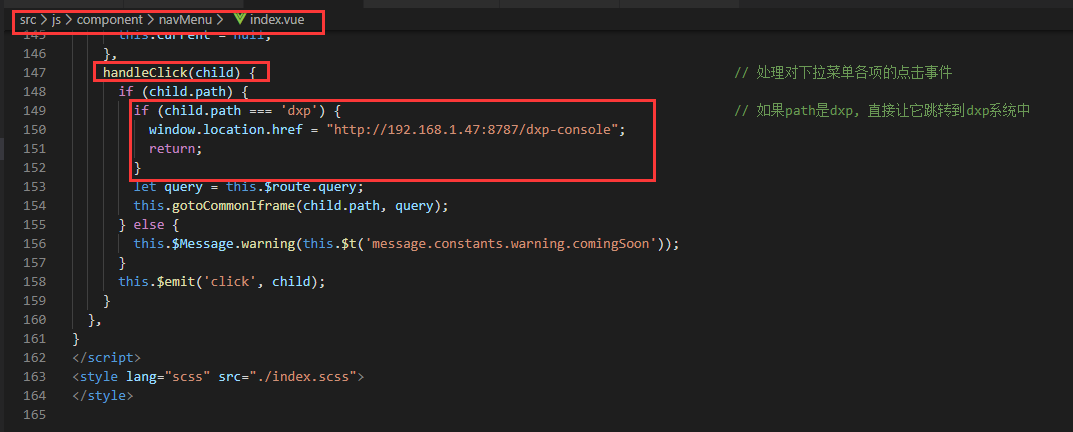
1. 添加图标（SVG图标）
2. 、将图标添加准备好的svg图标添加到src\ assets\ iconfont\ font-dws-icon.svg中



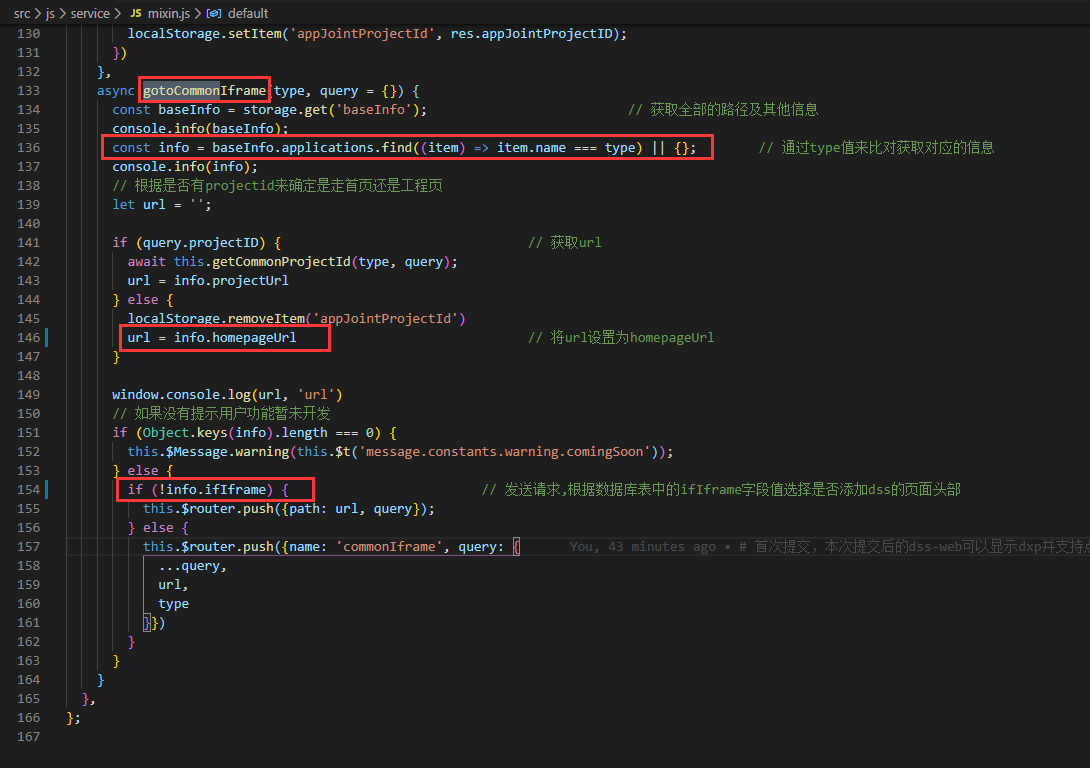
1. 、给图标设置属性通过它的unicode进行关联



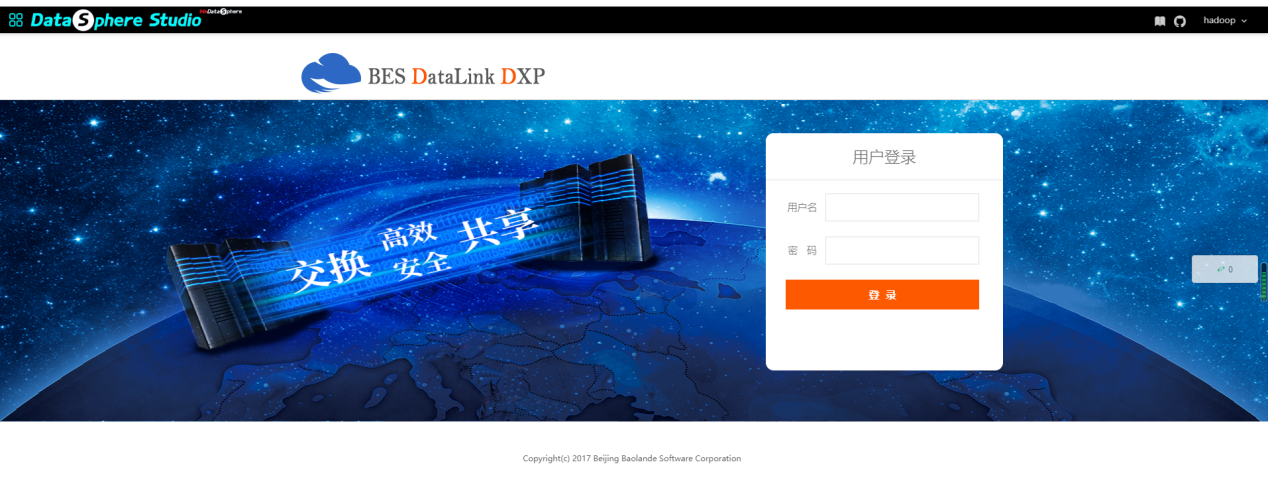
1. 点击下拉框中的Datalink DXP实现页面跳转过程



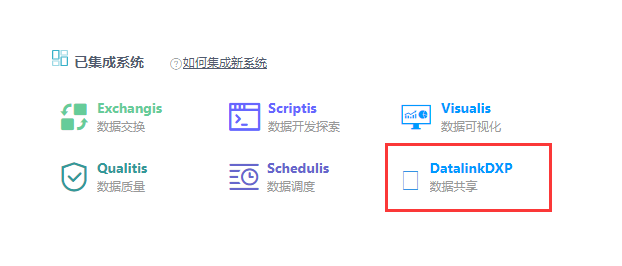
在点击下拉框中的选项后会调用handleClick()方法，若此相应下拉框组件的path值存在，handleClick()方法会以此path值为参数执行gotoCommonIframe()方法。



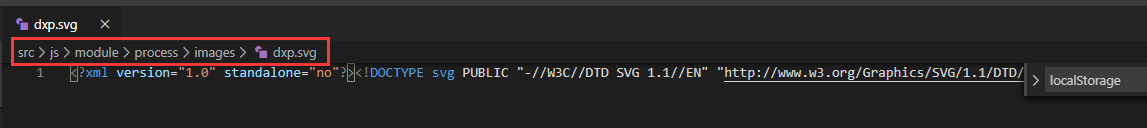
在gotoCommonIframe()方法中handleClick()传入的path赋值给type,此方法会根据type值与dss\_application表中的name字段进行匹配，匹配成功会将相应信息赋值给info。在下面进行判断是否传入了query.projectID，若是则url设置为info.projectUrl也就是dss\_application表中projectUrl值，否则将url的值设置为info.homepageUrl也就是dss\_application表中的homepageUrl字段的值。再然后根据表中info.ifIframe的值（dss\_application表中ifIframe字段的值）决定是否给跳转后的页面加上dss系统的页面头。



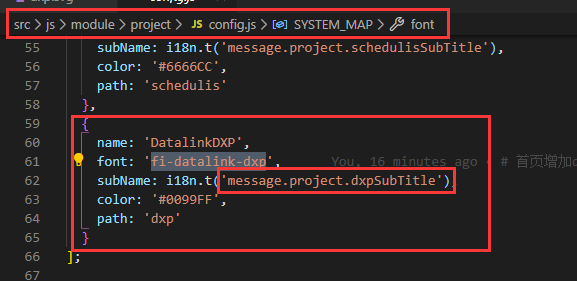
1. **首页集成DXP并实现点击跳转**

****

1. 添加图标，将准备好的dxp图标添加到是src\ js\ moudle\ process\ images目录下。

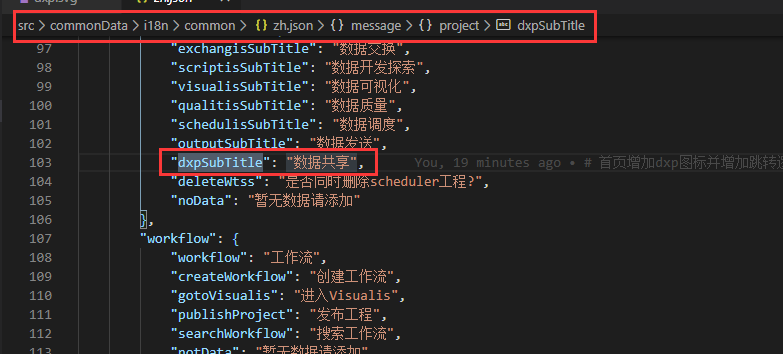


1. 在首页增加dxp的图标，在src\js\module\project\config.中进行配置。



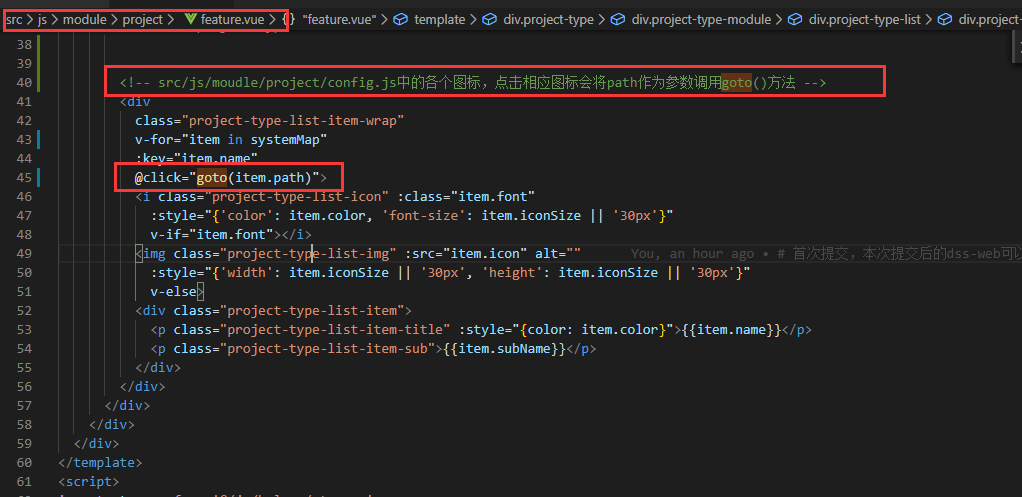
上图中的path要与数据库中表dss\_application的name字段向匹配。

1. 在zh.json中增加dxp的汉化描述dxpSubTitle。

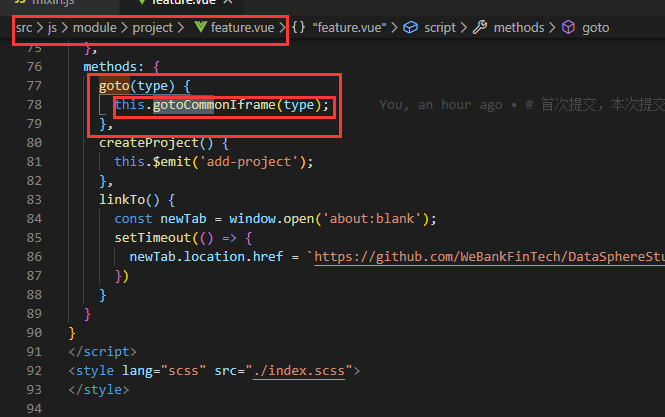


1. 增加点击dxp图标进行跳转的逻辑，在是src\ js\ module\ project\

Feature.vue进行配置。



点击相应图标后会以图标组件的path为参数调用goto()方法，goto()方法会以path为type参数调用gotoCommonIframe()方法。



而后gotoCommonIframe()方法的处理逻辑与上述下拉框集成的逻辑一致。

1. 后端集成
2. 前端在初始化的时候会向前端发送一个请求，获取dss\_application中各个结点的相应信息。此请求会交由dss-application处理，最终由

com.webank.wedatasphere.dss.application.restful.ApplicationRestfulApi负责。

1. dxp用户鉴权
2. 实现dss-dxp-appjoin后在相应目录中控制台中执行mvn clean install生成jar包们，Jar包需要放置到dss-server和linkis-appjoint-entrance两个微服务中。
3. 以linkis-appjoint-entrance 为例(dss-server与linkis-appjoint- entrance一致)，在linkis-appjont-entrance下面的lib的同级目录有一个appjoints目录， 在appjoints目录下面新建一个visualis目录。visualis目录下面要求有lib目录，lib目录存放的是visualis在实现VisualisAppJoint的编译的jar包，当然如果有引入dss系统没有带入的jar包，也需要放置到lib目录中，如sendemail Appjoint需要发送邮件功能的依赖包，所以需要将这些依赖包和已经实现的jar包统一放置到lib目录中。另外可以将本AppJoint所需要的一些配置参数放置到appjoints.properties,DSS系统提供的AppJointLoader会将这些配置的参数读取，放置到一个Map中，在AppJoint调用init方法的时候传入。