

硬件开发工具链平台云服务

开发指南

文档版本 03
发布日期 2024-08-08



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 卡片配置指南	1
1.1 关于卡片	1
1.2 开发概述	1
1.3 总体开发思路	2
1.4 准备工作	3
1.5 外部无 AUI 依赖开发示例	4
1.5.1 开发指导	4
1.5.2 注意事项	9
1.6 卡片上传使用	9
1.7 常用调试方法	13
1.7.1 前端调试方法	13
1.7.2 调试工具	16
2 模板卡片配置指南	17
2.1 模板卡片概述	17
2.2 通用清单模板卡片配置	17
2.3 通用外部链接卡片配置	22
2.4 生命周期数据卡片配置	25
3 统一认证配置指南	28
3.1 基于 OrgID 的单点登录配置	28
3.2 基于 IPDCenter 的单点登录配置	32
3.3 IPDCenter 外部集成开发	34
3.3.1 开发概述	34
3.3.2 总体开发思路	38
3.3.3 准备工作	39
3.3.4 快速开始	41
3.3.5 实现一个认证 API	43
3.3.5.1 开发指导	43
3.3.5.2 注意事项	43
3.3.6 代码示例	43
3.3.7 常用调试方法	44
3.3.7.1 查看日志	44
4 统一数据模型指南	45

4.1 对外 API 规范.....	45
5 流程配置指南.....	47
5.1 快速入门.....	47
5.1.1 入门指引.....	47
5.1.2 步骤一：创建申请单及流程节点.....	48
5.1.3 步骤二：配置服务定义.....	50
5.1.4 步骤三：创建属性及模型.....	57
5.1.5 步骤四：绘制流程页面.....	62
5.1.6 步骤五：发布电子流.....	72
5.2 服务定义.....	72
5.3 流程管理.....	76
5.3.1 创建流程申请单.....	76
5.3.2 流程设计.....	76
5.3.3 导航树配置.....	81
5.3.4 发布审批.....	82
5.4 数据模型.....	83
5.4.1 创建数据实体模型.....	83
5.4.2 申请单类型.....	84
5.4.3 属性库.....	85
5.4.4 数据实体.....	88
5.4.4.1 创建数据实体.....	88
5.4.4.2 作废数据实体.....	91
5.4.4.3 查询数据实体.....	91
5.4.4.4 查看数据实体.....	92
5.4.4.5 修改数据实体.....	93
5.4.5 同步应用.....	94
5.4.5.1 创建同步应用.....	94
5.4.5.2 管理同步应用.....	95
5.4.5.3 查询同步应用.....	97
5.4.5.4 查看同步应用.....	98
5.5 应用配置.....	99
5.5.1 基本配置.....	99
5.5.2 团队角色管理.....	101
5.6 UI 编辑器使用帮助.....	101
5.6.1 设计器界面.....	102
5.6.2 物料及模板.....	108
5.6.3 业务组件.....	111
5.6.3.1 添加部件.....	111
5.6.3.2 流程团队.....	113
5.6.3.3 流程指引.....	114
5.6.3.4 评审三件套.....	115
5.6.3.5 流程记录.....	117

5.6.3.6 评审记录.....	117
5.6.3.7 转审记录.....	118
5.6.3.8 附件.....	118
5.6.4 IPD 组件.....	119
5.6.4.1 容器组件.....	119
5.6.4.1.1 BOX（盒子）.....	119
5.6.4.1.2 栅格布局（列容器）.....	120
5.6.4.1.3 栅格布局（行容器）.....	122
5.6.4.1.4 折叠面板.....	123
5.6.4.1.5 页签.....	125
5.6.4.2 信息组件.....	128
5.6.4.2.1 文本展示.....	128
5.6.4.2.2 状态标识.....	131
5.6.4.2.3 帮助文档.....	132
5.6.4.2.4 弹框.....	133
5.6.4.2.5 消息提示框.....	135
5.6.4.3 表单组件.....	136
5.6.4.3.1 按钮.....	137
5.6.4.3.2 单选框.....	138
5.6.4.3.3 多选框.....	139
5.6.4.3.4 输入框.....	140
5.6.4.3.5 选择器.....	142
5.6.4.3.6 时间选择器.....	143
5.6.4.3.7 日期选择器.....	145
5.6.4.3.8 选人组件.....	147
5.6.4.4 数据组件.....	148
5.6.4.4.1 表单.....	149
5.6.4.4.2 表格.....	151
5.6.4.4.3 树形控件.....	154
5.6.5 流程模板.....	155
5.6.5.1 提交记录模板.....	156
5.6.5.2 评审页面模板.....	157
5.6.5.3 流程页面模板.....	158
5.6.6 属性面板设置.....	158
5.6.6.1 属性面板.....	158
5.6.6.2 设置器.....	159
5.6.7 高级面板设置.....	165
5.6.8 国际化词条.....	166
5.6.8.1 添加国际化词条.....	166
5.6.8.2 绑定国际化词条.....	168
5.6.8.3 管理国际化词条.....	170
5.6.9 页面 JS 及事件绑定.....	171

5.6.9.1 页面 JS 基本使用与管理.....	171
5.6.9.2 绑定 API.....	175
5.6.10 状态管理及变量绑定.....	177
5.6.10.1 状态管理.....	177
5.6.10.2 变量绑定及使用.....	179
5.6.11 资源管理.....	180
5.7 电子流应用.....	181
A 修订记录.....	182

1 卡片配置指南

1.1 关于卡片

- 卡片是指具有一定业务意义的页面元素及功能组合，通常由技术组件和业务服务组成。
- 卡片最终在门户页面上使用，管理员可以在卡片管理界面管理上传卡片，供用户选择配置到页面上。

1.2 开发概述

卡片库简介

卡片是一种新的服务形态，提供一种比页面更细粒度的可复用的原子化服务能力，能直观呈现独立的业务含义，使用户能够精准快速地获取想要的服务。目前卡片定义是狭义定义，不含报表、中大屏、页面、移动屏等由卡片编排的消费产物。

1. 作为一种入口方式。如在一站式应用，以图标/链接/图表/列表等形式作为子应用的入口，提供比菜单更丰富和聚合的呈现和交互效果。
2. 作为一种可视方式。如在销售项目空间，以图表/列表/表格等形式提供各项指标数据，用户可以根据需要删减指标卡片，调整指标卡片位置；系统也可以根据User Profile动态推送内容。

工作原理

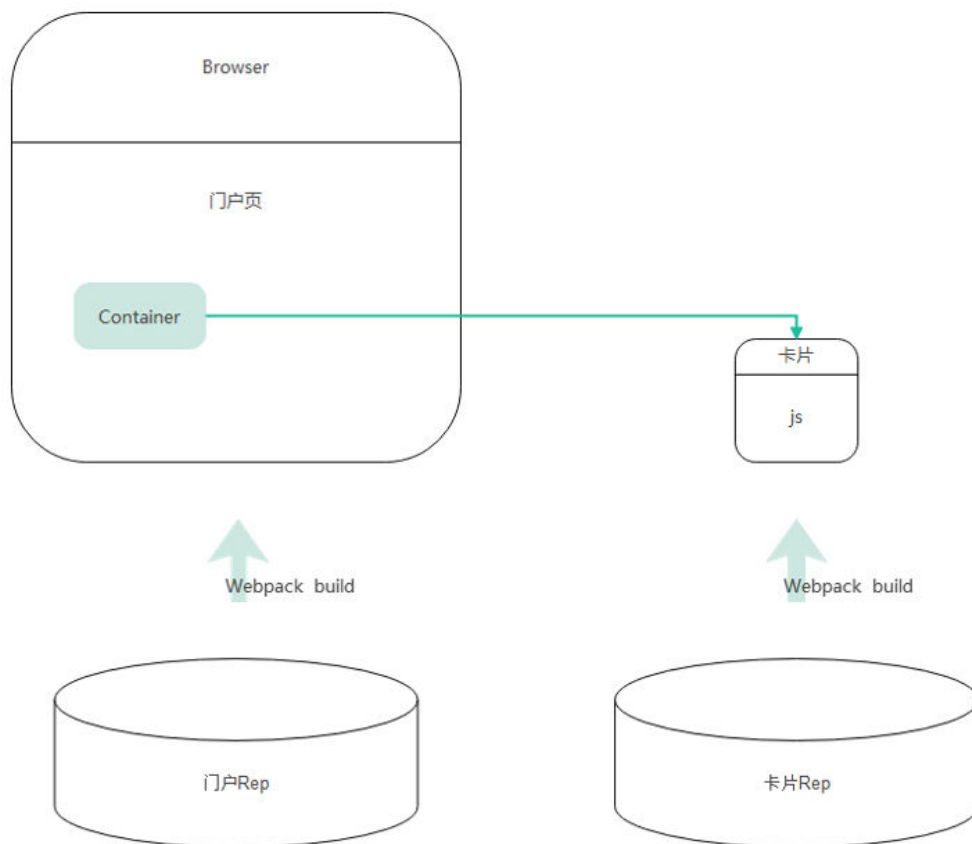
1. 组件化开发思想

采用Vue.js的组件化开发思想，将整体功能模块包装成一个个独立且可复用的组件，独立维护自身的状态及样式，更好地隔离多方合作伙伴的不同诉求，提升代码的可维护性。

2. 基于Webpack的构建过程

使用Webpack作为构建工具，通过对源代码进行打包、压缩、合并等操作，最终生成可运行的JS和CSS文件，将整体项目包装成一个独立单元模块，以组件的形式暴露。

图 1-1 工作原理

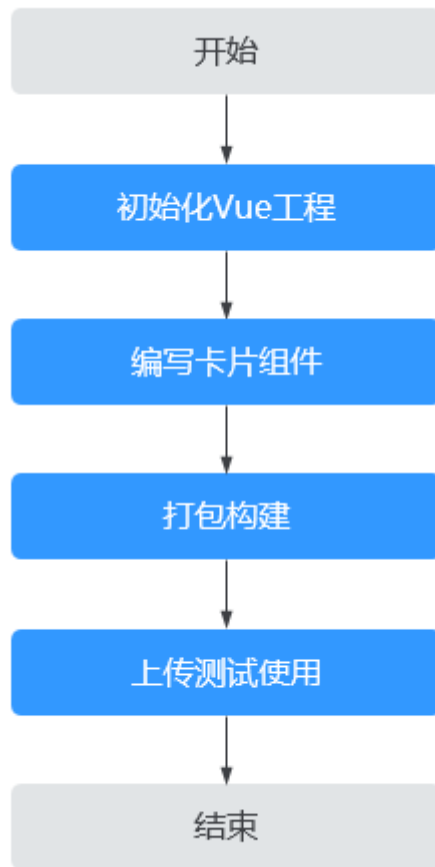


1.3 总体开发思路

开发流程

编写卡片的工作流程如下：

图 1-2 开发流程



1. 创建一个新的工程目录。
使用Vue CLI脚手架创建Demo前端工程，能够自动创建一个简单的样例工程，在此基础上开发能够减少很多繁琐的工作。
2. 编写构建Vue前端组件。
我们可以把功能或者需求类似的有机体封装成一个业务组件，并对外暴露接口来实现灵活的定制性，这样就可以在不同页面不同子系统中，复用同样的逻辑和功能了。
3. 通过Webpack进行组件打包构建成为独立JS。
独立JS能够被懒加载。
4. 在管理页面进行卡片创建，上传管理。
创建卡片实例，关联独立JS包，进行版本控制。

1.4 准备工作

开发技能要求

- 熟悉前端JS，能够编写JS代码。
- 了解基本的Vue框架，熟悉前端应用构建。

环境要求

已安装Node.js环境。

安装 Node 及 Npm

Node.js是一个JavaScript的运行环境，它能帮我们在服务器端运行JavaScript代码。我们需要Node.js来运行一些命令行工具。

首先，我们需要下载Node.js的安装包。你可以在Node.js的官方网站（<https://nodejs.org/>）下载安装包。

安装完成后，打开命令行工具，输入以下命令来检查Node.js是否成功安装：

```
node -v
```

如果安装成功，应该可以看到Node.js的版本号。

npm是Node.js的包管理器，用于安装第三方包。同样地，在命令行中输入以下命令来检查npm是否成功安装：

```
npm -v
```

如果安装成功，应该可以看到npm的版本号。

安装 Vue CLI

Vue CLI是Vue.js的脚手架工具，可以帮助我们快速创建Vue.js应用。要使用它，我们需要先安装它。

在命令行中输入以下命令来安装Vue CLI：

```
npm install -g @vue/cli
```

这个命令会全局安装Vue CLI，所以安装成功后，你可以在命令行中输入以下命令来检查是否安装成功：

```
vue --version
```

如果安装成功，可以看到Vue CLI的版本号。

1.5 外部无 AUI 依赖开发示例

1.5.1 开发指导

AUI为内部开源使用，外部无法访问，底层逻辑使用Vue组件暴露封装，因此在不使用AUI情况下，通过原生Vue CLI脚手架搭建组件库形式暴露卡片组件，也能够正常使用。

开发过程

准备工作中已经正常安装了Vue CLI脚手架，因此可以快速搭建Vue项目。

步骤1 初始化Vue项目。

```
vue create my-app
```

选择Vue3。

图 1-3 选择 Vue3

```
Vue CLI v5.0.8
? Please pick a preset: (Use arrow keys)
> Default ([Vue 3] babel, eslint)
  Default ([Vue 2] babel, eslint)
  Manually select features
```

步骤2 运行项目。

```
npm run serve
```

图 1-4 项目目录

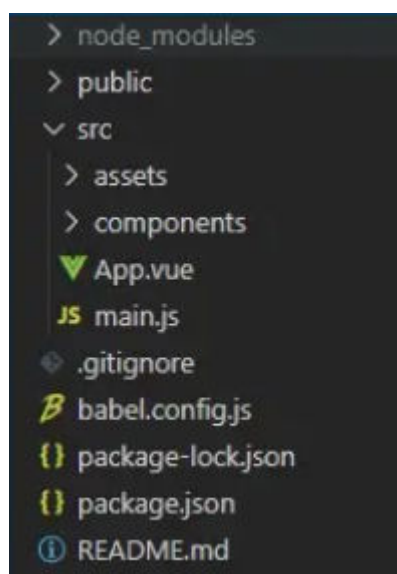
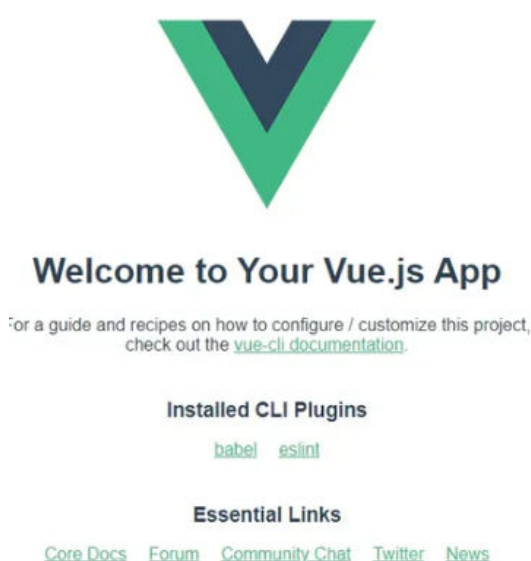
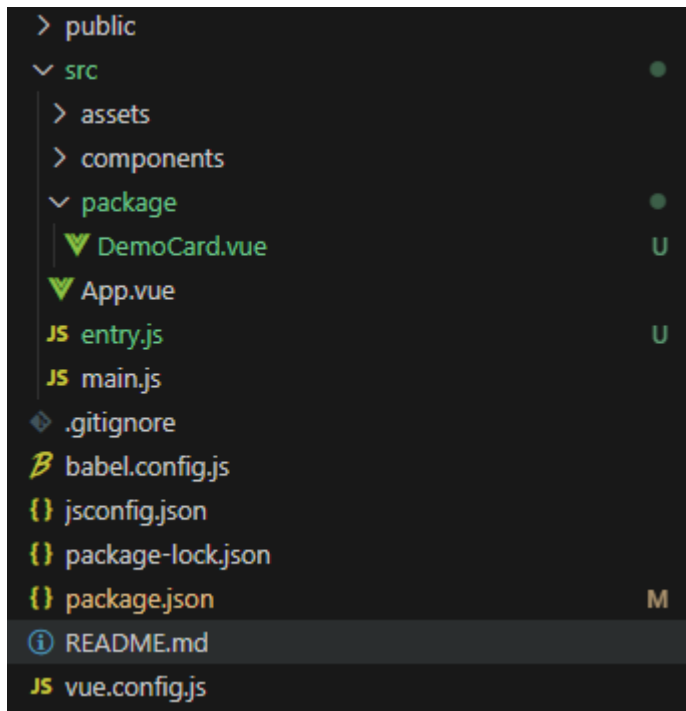


图 1-5 访问页面



步骤3 卡片封装，新建package文件夹。

图 1-6 新增 package 包



添加样例组件DemoCard.vue，添加任意样例代码如下：

```
<template>
  <div class="hello">
    <h1>我是卡片</h1>
    
  </div>
</template>
<script>
export default {
  name: 'DemoCard',
  props: {
    url: String // url接收来自门户关于静态资源的路径
  }
}
</script>
<!-- Add "scoped" attribute to limit CSS to this component only -->
<style scoped>
h3 {
  margin: 40px 0 0;
}
</style>
```

然后我们引用到App.vue组件里面验证一下，看组件是否可用，代码如下：

```
<template>
  
  <HelloWorld msg="Welcome to Your Vue.js App"/>
</template>
<script>
import HelloWorld from './package/demoCard.vue'
export default {
  name: 'App',
  components: {
    HelloWorld
  }
}
</script>
```

```
<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

最终效果显示如下:

图 1-7 组件预览



步骤4 使用Vue打包的lib库模式，添加入口文件。

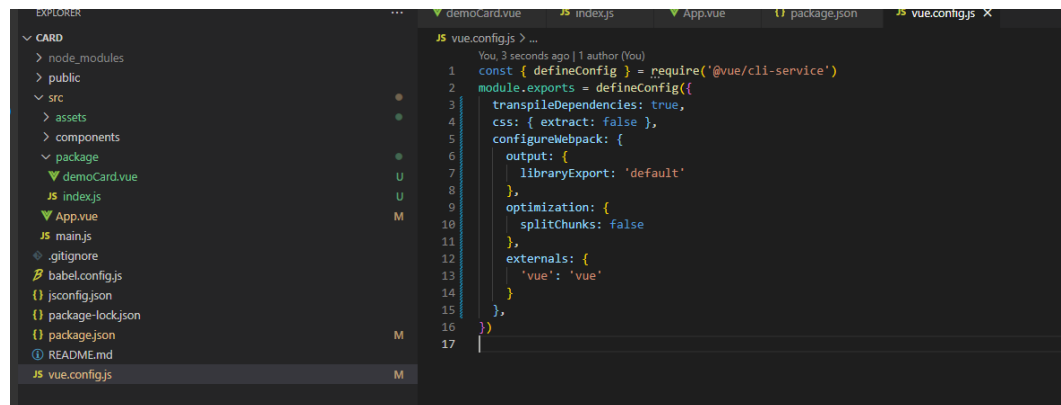
打包后的js文件运行逻辑已在门户专门处理，按照下列方式配置打包入口文件即可。

在package目录下新建index.js文件，代码如下：

```
import demoCard from './demoCard.vue'
const component = {
  default: demoCard
};
export default component
```

步骤5 打包配置。

图 1-8 vue.config.js 配置



配置示例：

```
const { defineConfig } = require('@vue/cli-service')
module.exports = defineConfig({
  transpileDependencies: true,
  css: { extract: false },
  configureWebpack: {
    output: {
      libraryExport: 'default'
    },
    optimization: {
      splitChunks: false
    },
  },
  externals: {
    'vue': 'vue'
  }
},
})
```

步骤6 卡片打包。

到这里为止，我们的卡片封装基本就完成了，当然卡片封装成什么样得看自己的业务需求了，接下来我们就需要将卡片进行打包了。

修改我们项目的package.json文件，配置打包命令：

```
"build-lib": "vue-cli-service build --target lib --name mylib --dest lib src/package/index.js"
```

图 1-9 npm 命令

```
"scripts": {
  "serve": "vue-cli-service serve",
  "build": "vue-cli-service build",
  "build-lib": "vue-cli-service build --target lib --name mylib --dest lib src/package/index.js",
  "lint": "vue-cli-service lint"
},
```

打包命令解释：

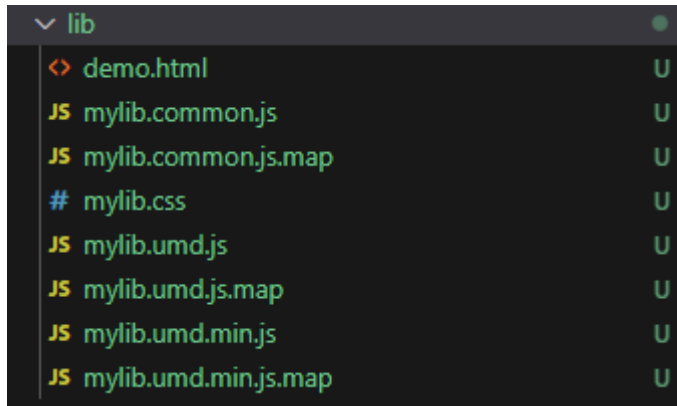
- --target lib 关键字指定打包的目录。
- --name 打包后的文件名字。
- --dest 打包后的文件夹的名称。

然后执行打包命令：

```
npm run build-lib
```

打包执行完成后我们项目目录下就会多出一个“lib”文件夹，存放的是打包后的文件。

图 1-10 输出件



----结束

1.5.2 注意事项

注意事项

卡片运行资源大小会影响资源下载的速度，也会影响浏览器的解析执行，卡片资源体积不宜太大，不能超过5M。

尽量用最小的代码实现卡片功能，例如：

1. 优先考虑使用公共资源。
2. 使用构建工具对代码进行压缩混淆。
3. 持续重构，改善代码质量。
4. 打包时排除vue、axios。

1.6 卡片上传使用

整体卡片管理功能权限受控，应当由具备管理员权限的账号进行统一管理维护，普通账号无权限进行操作。

上传卡片

- 步骤1** 访问CraftArts IPDCenter门户网站，登录管理员账号，点开左上角功能扩展按钮，进入管理中心。

图 1-11 进入管理中心



步骤2 绑定应用。

进入管理中心，选择“集成管理 > 应用集成”，单击“添加应用”。

图 1-12 添加应用



步骤3 新增密钥。

选择“集成管理 > 密钥管理”，单击“创建密钥”。

密钥作为卡片认证信息唯一标识，在卡片开发过程中使用当前密钥进行统一认证，生成token进行前后端数据请求。详情参见[统一认证配置指南](#)。

图 1-13 新增密钥



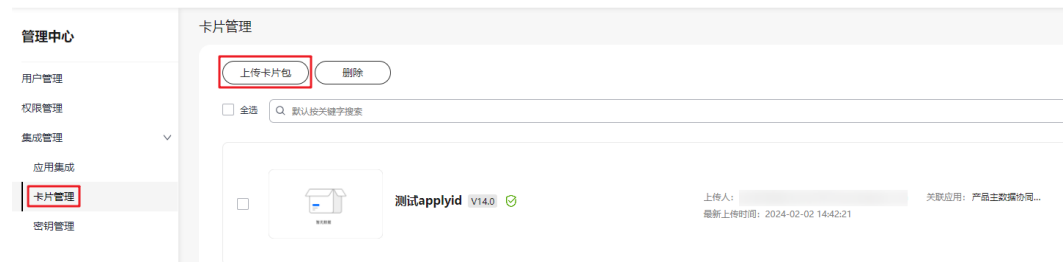
步骤4 将[开发指导](#)中生成的卡片JS进行上传管理。

将打包后的mylib.umd.min文件名修改为index，并同需要引用的静态资源压缩在同一级目录。

图 1-14 zip 示例截图

名称	类型	压缩大小	密码保护	大小	比率	修改日期
index	JavaScript 源文件	8 KB	否	20 KB	65%	2024/4/1 16:09
logo	PNG 文件	7 KB	否	7 KB	5%	2024/4/1 15:44

图 1-15 上传卡片



说明

Vue CLI生成的卡片输出为完整js，需要将其重命名为index.js，放置在zip压缩包内。否则会出现验证不通过的异常。

---结束

使用卡片

卡片上传完成之后就可以在角色桌面中查看并使用了。

步骤1 进入管理中心，选择“集成管理 > 卡片管理”，选择需要使用的卡片，单击“使用卡片”并“确定”。

图 1-16 使用卡片



步骤2 进入门户编辑页面。

图 1-17 进入门户编辑



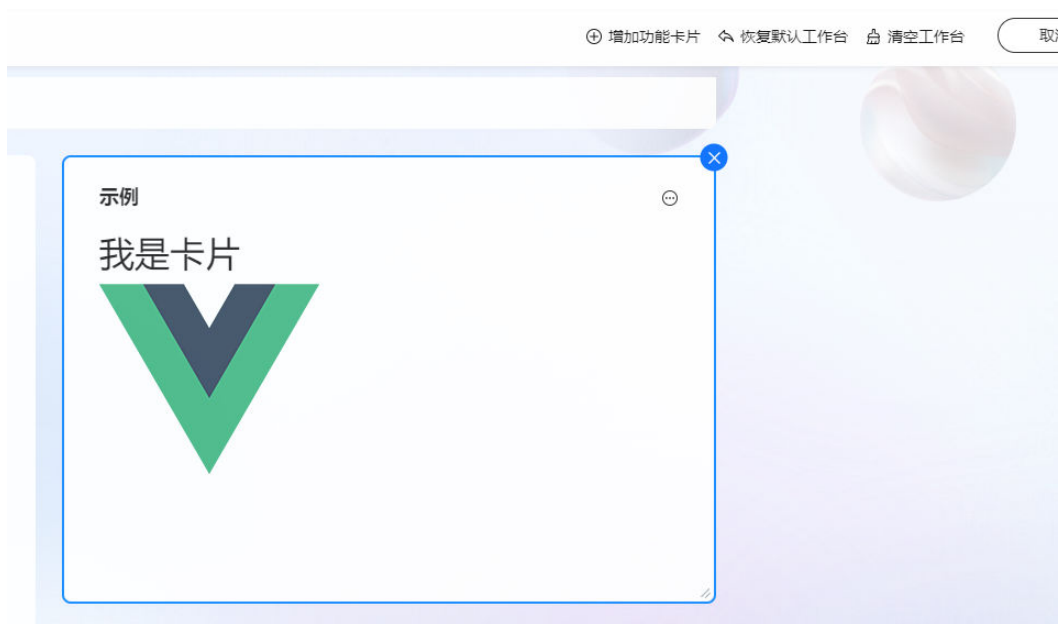
步骤3 单击“增加功能卡片”展开卡片列表，拖动创建好的卡片到角色桌面中。

图 1-18 添加专属卡片



步骤4 成功引入卡片展示，效果如图。

图 1-19 卡片渲染成功样例



----结束

1.7 常用调试方法

1.7.1 前端调试方法

查看 console.log

通常我们遇到BUG，第一直觉肯定看console的报错信息，这种报错，错误原因和报错位置一目了然，很方便就定位到出问题的代码，console.log简单实用，简单的缺陷还是可以通过这种方法定位的，生产环境利用webpack去掉，就不会影响性能方面。

图 1-20 console 报错



全局查找

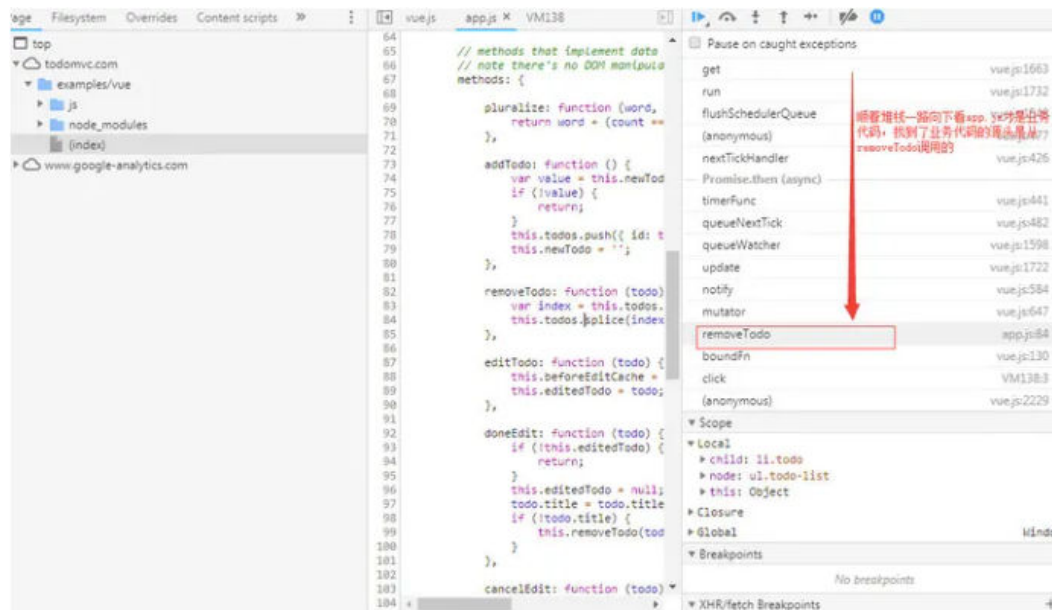
可以根据报错关键字定位问题，一般是dom元素的id、class、name或者页面中出现的唯一性的中文汉字，搜索就可以直接定位到代码。

调用堆栈

堆栈是一个数据结构，每一个函数调用时都会将函数的指针和参数值保存到堆栈中，后进先出，最后调用的函数最先出栈。

可以通过打断点，查看Chrome开发面板的sources的call stack，可以查看追溯到源代码的位置，这个特别适合console报错位置不明确的情况（比如报错指向vue.esm.js）。

图 1-21 代码追溯



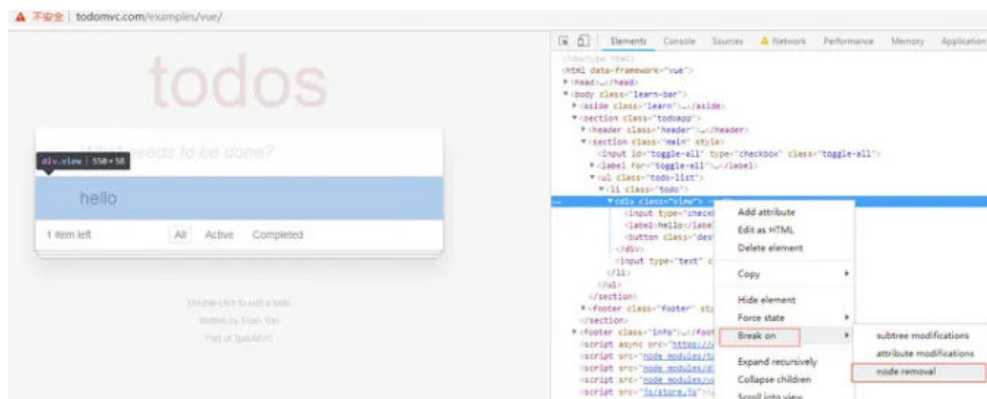
分析代码

分析代码最常用的就是断点，一步步调试查看结果，定位问题。

断点

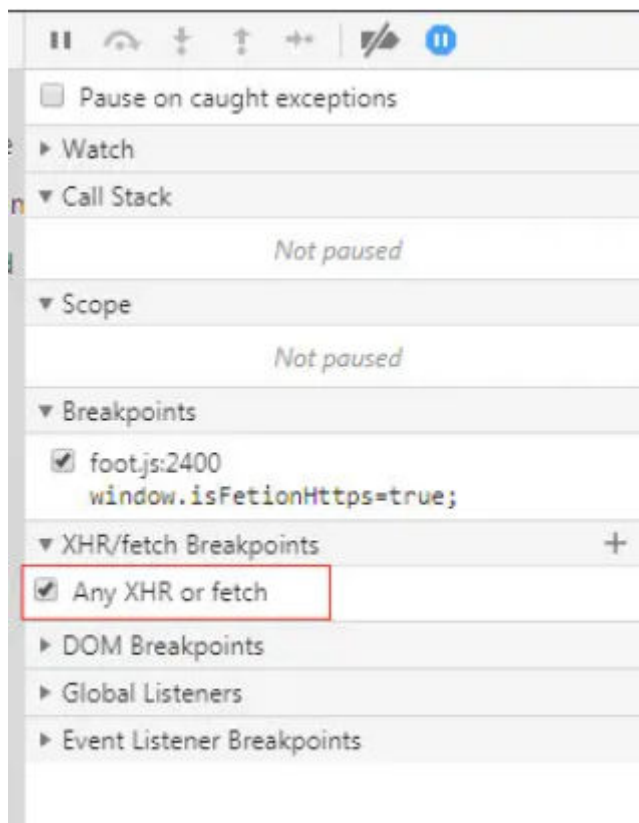
1. 直接在代码里Debugger。
在代码运行到该处就能触发断点，这对于Sources面板来说不容易寻找源码中的断点。
2. Sources源码中加断点。
这种打断点非常方便，但有时候没那么容易找到源码。
3. Dom修改时加断点。

图 1-22 Dom 断点



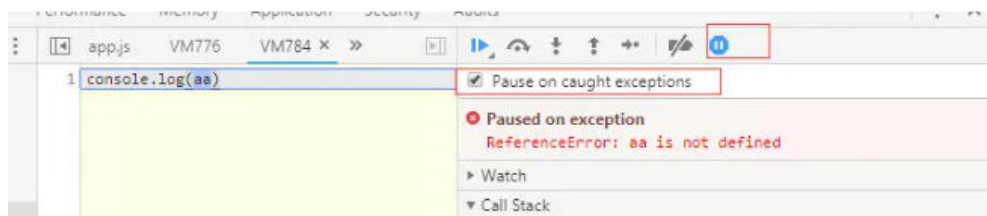
4. Ajax请求断点。

图 1-23 Ajax 断点



5. 异常时断点。

图 1-24 异常断点



调试代码

调试代码最好保持本地环境和线上环境一致，这样基本保证修改缺陷不会被测试验证不通过。那怎么保持本地环境和线上保持一致呢？

1. Vue CLI的本地代理功能，在配置文件`devServer.proxy`中配置反向代理，这种方式最好弄一个网站作为中介代理，开发的时候只需要修改网站上的配置，本地服务不用重复启动。
2. 自己搭建Nginx反向代理，效果同上。
3. 后端接口设置Access-Control-Allow-Origin，利用cors跨域本地直接访问线上数据，这种方法最方便，但需要后端同事配合。
4. 使用Fiddler AutoResponder功能，直接把服务器重定向到本地目录（需要编写一个正则表达式），修改本地文件刷新浏览器就能看的修改效果。

5. 使用Chrome DevTools的Overrides功能，把服务器的文件映射到本地，可以直接在sources面板中修改文件中的方法，这种最适合调试jsp这种不用webpack压缩的老项目。

1.7.2 调试工具

Vue Devtools是一个专为Vue.js开发者设计的浏览器扩展。它提供了一系列强大的功能，帮助开发者更容易地调试和优化他们的Vue应用程序。

Vue Devtools

以下是Vue Devtools的主要特点和功能：

1. 组件树查看：允许用户查看完整的Vue组件层次结构，以及每个组件的属性、数据、计算属性和插槽。
2. 实时编辑：在Devtools中直接修改组件的数据，可以立即在应用中看到变化。
3. 事件追踪：查看组件之间的事件传递，帮助理解和调试组件间的交互。
4. Vuex集成：如果应用使用了Vuex，可以在Devtools中查看、追踪和编辑应用的状态。
5. 性能分析：提供组件渲染的性能数据，帮助找到和解决性能瓶颈。
6. 插件支持：可以集成第三方Vue插件，为开发者提供更多的调试工具。
7. 定制设置：允许开发者根据自己的喜好调整Devtools的外观和行为。

总的来说，Vue Devtools是每个Vue开发者必备的工具，无论是新手还是经验丰富的开发者，都可以从中受益。它简化了调试过程，提高了开发效率和应用质量。

2 模板卡片配置指南

2.1 模板卡片概述

模板卡片是特定的专属卡片，它可以跳过环境准备、创建项目、编码、打包发布等过程，通过上传配置文件的方式生成自定义或标准化的内容。

目前系统中存在如下模板卡片：

- 通用清单模板卡片：用于以表格形式展示分页数据，可以通过配置文件配置表头和表格数据源接口。
- 通用外部链接模板卡片：用于以iframe形式展示嵌入式页面，可以通过配置文件配置需要展示URL地址。
- 生命周期数据模板卡片：用于展示统计目标实体各个生命周期状态的实例个数及完成率，可以通过配置文件配置源实体，目标实体、关系实体和计算规则等。

2.2 通用清单模板卡片配置

前提条件

已按[IPDCenter外部集成开发](#)规范完成认证逻辑。

适用场景

通用清单卡片适用于展示表格分页数据，可以通过配置表格的表头、数据源来快速自定义不同的表格。支持配置单个表格，也支持配置多个表格，多个表格以tab页签切换的方式展示在卡片中。

通用清单卡片类似如[图2-1](#)所示，卡片中展示三个表格，您可以单击页签名称切换查看相应的表格信息。

图 2-1 通用清单卡片模板预览效果

产品主数据资产

我的部件 我的文档 我的图纸

编码	名称	生命周期状态	流程状态	创建时间
0302HW6PX5	单板	Develop	In Work	2024-05-13 19:5...
0302HW6PX3	单板	Develop	In Work	2024-05-09 11:0...
0302HW6PX1	单板	Develop	In Work	2024-05-08 15:3...
0302HW1199...	制成板(华为云)	Develop	Released	2024-05-06 16:4...

总条数: 832

10 < 1 2 3 4 5 6 ... 84 >

配置模板卡片的配置文件

步骤1 打开文本编辑工具，新建一个文件。

您也可以在开发工具新建一个文件作为配置文件。

步骤2 在文件中填写配置文件信息，配置文件的参数配置说明请参见[通用清单卡片配置文件参数配置说明](#)。

配置文件格式如下：

```
{
  "tableList": [
    {
      "tableName": "表格1",
      "columns": [
        {
          "title": "title1",
          "field": "field1"
        },
        {
          "title": "title2",
          "field": "field2",
          "type": "link",
          "jumpUrl": "${detailPageUrl}"
        },
        {
          "title": "title3",
          "field": "field3",
          "type": "date",
          "format": "yyyy-MM-dd HH:mm:ss"
        }
      ]
    },
    {
      "queryDataType": "api",
      "apiConfig": {
        "url": "url",
        "method": "GET",
        "headers": {
          "authorizationKey": "",
          "applyId": ""
        }
      },
      "body": {},
      "pageNoField": "pageNo",
      "pageSizeField": "pageSize",
      "dataBody": "resp.data.data",
    }
  ]
}
```



```
"totalBody": "resp.data.totalCount"  
  }  
}  
]  
}
```

步骤3 将配置文件保存在本地，文件需要保存为json格式。

----结束

通用清单卡片配置文件参数配置说明

表 2-1 配置项

名称	类型	默认值	说明
tableList	tableConfig[]	[]	表格项的配置信息，支持配置多个表格项。 表格项超过1个时，各个表格以tab页签方式组合，可以切换查看每个表格。最多支持配置5个表格项，表格项中的具体配置请参考 表2-2 。

表 2-2 表格项配置

名称	类型	默认值	说明
tableName	string	--	表格名称，展示为tab页签的标题。 当表格项个数为1个时不展示。
columns	columnConfig	--	表格列的配置信息，具体配置请参考 表2-3 。
queryDataTyp e	string	--	获取表格数据源的方式，当前仅支持配置为“api”。
apiConfig	apiConfig	--	数据源接口的配置，具体配置请参考 表2-4 。

表 2-3 表格列配置

名称	类型	默认值	说明
title	string	--	列标题，展示为表格中各列的参数名称。
field	string	--	列的key值，用于映射数据源中的字段。

名称	类型	默认值	说明
type	string	--	数据处理方式，包括如下选项： <ul style="list-style-type: none">• link：链接方式，配置此种方式时，需要配置“jumpUrl”参数。• date：日期方式，配置此种方式时，需要配置“format”参数。
jumpUrl	string	--	跳转链接，只在“type”配置为“link”时有效。 若使用\${fieldName}，则取表格数据中的fieldName字段的值替换\${fieldName}生成跳转链接。

名称	类型	默认值	说明
format	string	'yyyy-MM-dd HH:mm:ss'	<p>时间转换格式，只在“type”配置为“date”时列有效。</p> <p>日期格式配置规则如下：</p> <ul style="list-style-type: none"> • y: 四位数的年份 • yy: 两位数的年份 • yyy: 三位数的年份 • yyyy: 四位数的年份 • M: 不带前导0的月份（例如：1-12） • MM: 带前导0的月份（例如：01-12） • MMM: 月份的缩写（例如：Jan、Feb） • d: 不带前导0的日期（例如：1-31） • dd: 带前导0的日期（例如：01-31） • h: 12小时制的小时，不带前导零（例如：1-12） • hh: 12小时制的小时，带前导零（例如：01-12） • H: 24小时制的小时，不带前导零（例如：0-23） • HH: 24小时制的小时，带前导零（例如：00-23） • m: 不带前导0的分钟（例如：0-59） • mm: 带前导0的分钟（例如：00-59） • s: 不带前导0的秒钟（例如：0-59） • ss: 带前导0的秒钟（例如：00-59） • SSS: 毫秒（例如：000-999）

表 2-4 数据源接口配置

名称	类型	默认值	说明
url	string	"	接口的URL地址。

名称	类型	默认值	说明
method	string	'GET'	接口的请求方式，包括如下选项： <ul style="list-style-type: none"> • GET • POST • DELETE • PUT
headers	Object	{}	接口请求头配置，支持HTTP标头配置项，同时支持机机认证逻辑。如果配置为机机认证，需要配置机机认证信息，具体配置请参考表2-5。
body	Object	{}	接口参数。
pageNoField	string	'pageNo'	接口分页参数中页码对应的名称。
pageSizeField	string	'pageSize'	接口分页参数中分页条数对应的名称。
dataBody	string	"	接口响应返回值中表格数据链式路径。
totalBody	string	"	接口响应返回值中表格总条数链式路径。

表 2-5 机机认证配置

名称	类型	默认值	说明
applyId	string	--	接入机机认证机制的第三方接口对应的应用ID，用于获取token。
authorization Key	string	--	接入机机认证机制的第三方接口自定义的请求头字段，用于携带token。

2.3 通用外部链接卡片配置

前提条件

已按[IPDCenter外部集成开发](#)规范完成认证逻辑。

适用场景

通用外部链接卡片适用于展示项目页面，可以通过配置不同的项目URL地址来展示不同的页面。如果需要展示多个项目页面，您可以通过配置一组项目的数据源来实现，展示时通过选择项目查看所选项目页面。

通用外部链接卡片展示效果如图2-2所示。

图 2-2 通用外部链接卡片模板预览效果



配置模板卡片的配置文件

步骤1 打开文本编辑工具，新建一个文件。

您也可以在开发工具新建一个文件作为配置文件。

步骤2 在文件中填写配置文件信息，配置文件的参数配置说明请参见[通用外部链接卡片配置文件参数配置说明](#)。

配置文件格式如下：

```
{
  "label": "标签名称",
  "labelField": "${labelField1}",
  "isShowForm": true,
  "queryDataType": "api",
  "apiConfig": {
    "url": "http://www.example.com",
    "method": "GET",
    "headers": {
      "authorizationKey": "X-Pdm-Isc-Common-Token",
      "applyId": "1"
    }
  },
  "body": {
    "params1": "value1"
  }
}
```

```
"dataBody": "resp.data.data"
},
"iframeUrl": "${iframeUrl}"
}
```

步骤3 将配置文件保存在本地，文件需要保存为json格式。

----结束

通用外部链接卡片配置文件参数配置说明

表 2-6 配置项

名称	类型	默认值	说明
iframeUrl	string	"	用于配置待展示页面的完整URL地址。支持直接配置为完整的URL地址，也支持配置为数据映射来获取数据链中的某个变量。如配置为\${iframeUrl}，则会获取apiConfig配置的数据源接口的返回值中“ifreamUrl”的值。
isShowForm	Boolean	false	是否展示表单下拉框，包括如下选项： <ul style="list-style-type: none"> ● 不配置 ● false ● true 不配置或者配置值为“false”时，数据源下拉框不展示。
label	string	"	下拉框前显示的标签名称。
labelField	string	'\${label}'	下拉框选项中显示的内容，必须使用数据映射方式进行自定义配置。如配置为“\${labelField1}”，界面中展示时会从接口返回值中取“labelField1”字段的值进行替换。不配置时会默认获取数据源中“label”字段的值。
apiConfig	apiConfig	--	用于配置数据源接口，具体配置请参考 表2-7 。

表 2-7 数据源接口配置

名称	类型	默认值	说明
url	string	"	接口地址。

名称	类型	默认值	说明
method	string	'GET'	接口请求方式，包括如下选项： <ul style="list-style-type: none">• GET• POST• DELETE• PUT
headers	Object	{}	接口请求头配置，支持HTTP 标头配置项，同时支持机机认证逻辑。 如果配置为机机认证，需要配置机机认证信息，具体配置请参考 表2-8 。
body	Object	{}	接口参数。
dataBody	string	"	接口响应返回值中表格数据链式路径。

表 2-8 机机认证配置

名称	类型	默认值	说明
applyId	string	--	接入机机认证机制的第三方接口对应的应用ID，用于获取 token。
authorization Key	string	--	接入机机认证机制的第三方接口自定义的请求头字段，用于携带 token。

2.4 生命周期数据卡片配置

前提条件

IPDCenter基础服务的unidata实体中存在待配置的数据实体、关系实体、目标实体。

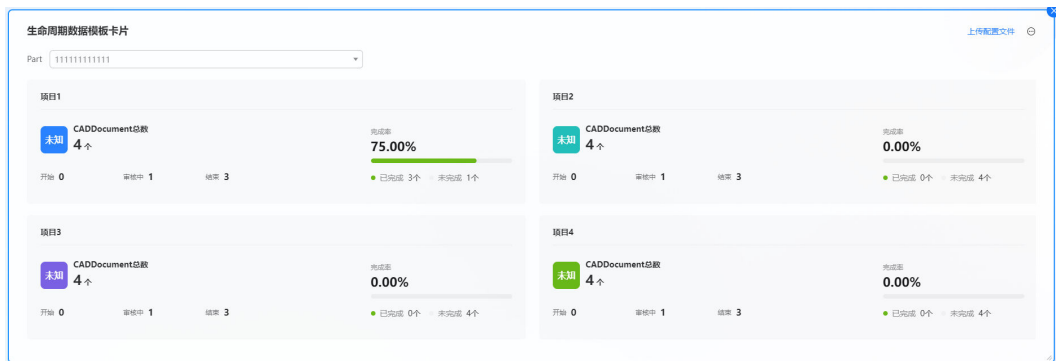
适用场景

生命周期数据模板卡片，适用于展示、统计，目标实体在各个生命周期状态的，实例个数及完成率等信息。

通过配置不同的源数据实体、关系数据实体和目标数据实体，计算目标数据实体的实例在各个生命周期状态个数，并根据自定义规则计算出完成率。

生命周期卡片的展示效果如[图2-3](#)所示。

图 2-3 生命周期卡片预览效果



配置模板卡片的配置文件

步骤1 打开文本编辑工具，新建一个文件。

您也可以在开发工具新建一个文件作为配置文件。

步骤2 在文件中填写配置文件信息，配置文件的参数配置说明请参见[生命周期数据卡片配置文件参数配置说明](#)。

配置文件格式如下：

```
{
  "label": "标签名称",
  "sourceEntityName": "Source",
  "labelField": "${labelField2}",
  "dataCardList": [
    {
      "cardName": "xxx",
      "relationEntityName": "Relation",
      "targetEntityName": "Target",
      "completionStatusList": ["state1"]
    }
  ]
}
```

步骤3 将配置文件保存在本地，文件需要保存为json格式。

----结束

生命周期数据卡片配置文件参数配置说明

表 2-9 配置项

名称	类型	默认值	说明
label	string	"	下拉框的标签名称。
labelField	string	'\${name}'	下拉框选项显示的文本，必须使用数据映射方式进行自定义配置。 如配置为“\${labelField2}”，界面中将会展示从unidata获取源实体实例的接口返回值中取“labelField2”字段的值。

名称	类型	默认值	说明
sourceEntityName	string	"	源数据实体英文名称，需要配置为已存在于unidata系统中的数据实体。
dataCardList	dataCardConfig[]	[]	数据项配置，支持配置多个数据项，详细配置请参考表2-10。

表 2-10 数据项配置

名称	类型	默认值	说明
cardName	string	"	数据项名称。
iconText	string	"	数据项图标文字。
relationEntityName	string	"	关系实体英文名称，需要配置为已存在于unidata系统中的关系实体。
targetEntityName	string	"	目标数据实体英文名称，需要配置为已存在于unidata系统中的数据实体。
completionStatusList	array	[]	生命周期状态中已完成状态的英文名称，用于配置完成率计算规则。可以在该配置项中配置所有已完成状态的类型，其他状态为未完成类型。

3 统一认证配置指南

3.1 基于 OrgID 的登录配置

背景介绍

组织成员帐号 (OrgID) 是面向企业提供组织管理、企业成员帐号管理以及SaaS应用授权管理能力的云服务，将Huawei ID帐号体系延伸到企业用户，统一华为云面向生态SaaS服务的组织、帐号，同时面向生态伙伴推出SaaS服务帐号集成规范。当前 CraftArts IPDCenter在公有云场景使用OrgID作为认证登录服务。

前提条件

1. 已开通**OrgID服务**。
2. 邀请注册帐号加入CraftArts IPDCenter组织，并授予管理员权限（与CraftArts IPDCenter集成时，采用此种方式，各应用共用一个组织）。

基本信息配置

- 步骤1 登录管理中心。**
- 步骤2** 选择左侧导航栏的“应用管理”。
- 步骤3** 单击“添加自建应用”。

图 3-1 添加自建应用



1. 输入应用名称，如“自建App”。
2. 上传应用图标，图标要求必须为JPG或PNG格式，大小不超过20KB，尺寸240*240px。
3. 选择应用类型，当前仅支持选择“Web”。
4. 设置应用负责人，输入并选择成员姓名，将成员设置为应用负责人。

📖 说明

- 应用负责人即该应用的应用管理员，只有应用管理员才能更新该应用配置，其他管理员没有操作该应用的权限。
- 普通成员不能成为应用负责人，需先成为组织管理员、部门管理员才能被设置为应用负责人。

步骤4 单击“确定”，进入认证集成页面。

----结束

认证集成配置

OrgID的登录集成方式推荐使用OAuth2。

图 3-2 选择认证方式

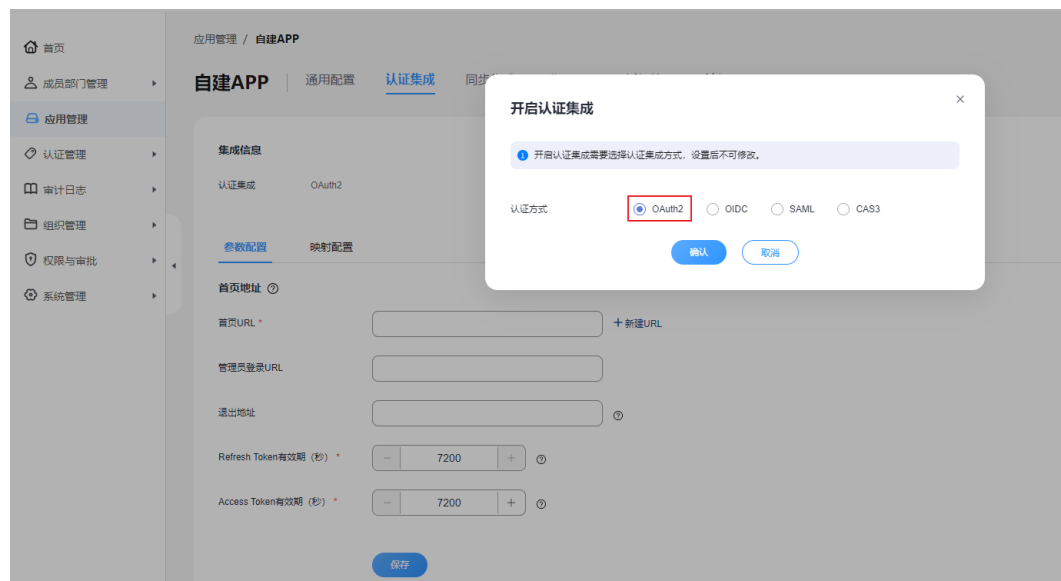


表 3-1 参数说明

认证集成方式	参数名称	参数说明
OAuth2	首页URL	应用首页的URL地址，例：https://xx.xx。 支持设置多个首页的URL地址，可单击“新建URL”，添加新的URL地址。
	管理员登录URL	可选项，管理员登录应用的URL地址。
	退出地址	可选项，应用的退出地址，请以http或https开头，例：https://xx.xx/logout。

认证集成方式	参数名称	参数说明
	Refresh Token有效期（秒）	允许用户在多久时间内不用重新登录应用的时间。
	Access Token有效期（秒）	允许用户在多久时间内保持登录应用的时间。

说明

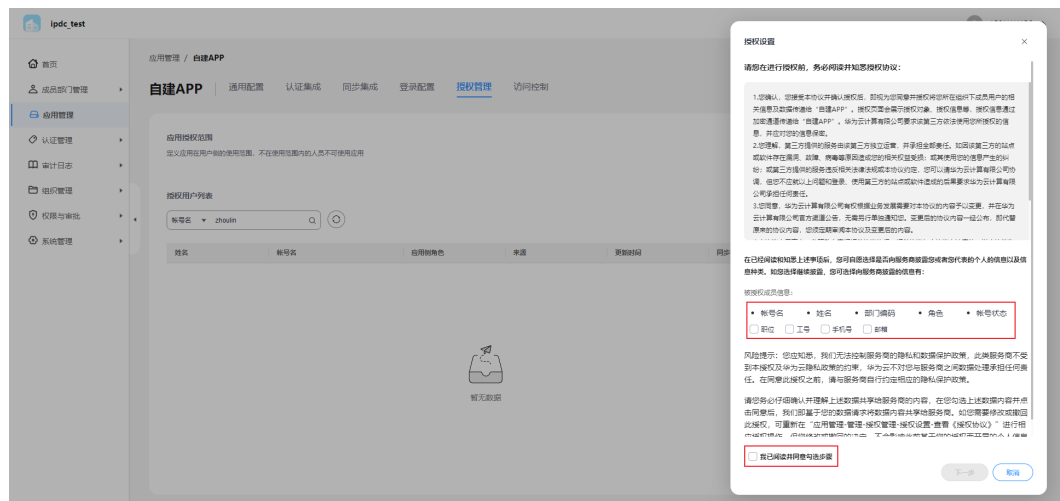
配置“首页URL”，此参数配置为产品首页的URL，集成OrgID后，在OrgID登录成功后会跳转到此地址，并携带认证参数。

授权管理配置

在授权管理页面对用户进行授权（配置哪些账号可以登录此应用）。

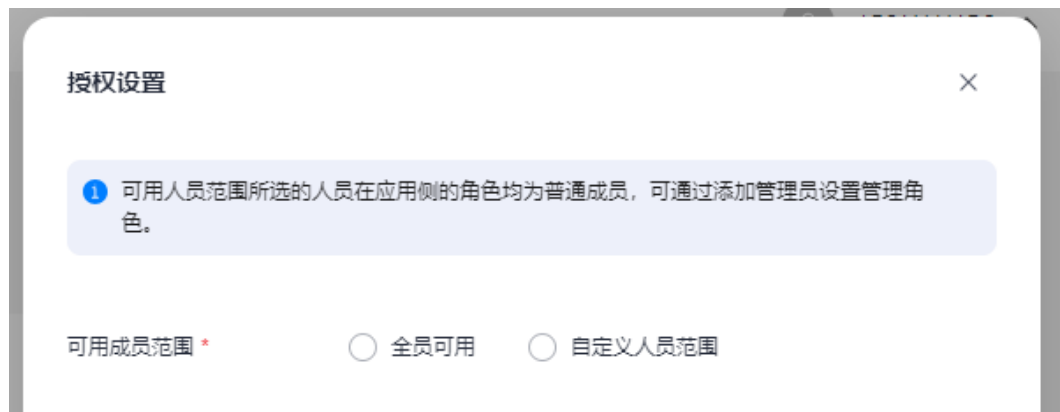
步骤1 单击“授权设置”，在“授权设置”界面中选择被授权成员信息，单击“下一步”。

图 3-3 授权设置-1



步骤2 选择可用成员范围，可勾选“全员可用”或“自定义人员范围”，勾选“自定义人员范围”后还需要选择指定的部门与成员或者用户组。

图 3-4 授权设置-2



📖 说明

设置后，应用授权范围中会显示授权部门、授权成员或授权用户组信息。同时，授权用户列表中也会展示授权账号的详细情况（包括姓名、账号名、应用侧角色、来源、更新时间和同步状态），支持按照时间或账号名进行过滤查询。

步骤3 单击“确定”。

----结束

接口对接配置

应用需要完成的逻辑

1. 在用户访问应用时，应用需要判断是否已经登录，如果没有登录，需要重定向到 OrgID 登录页面，获取[认证集成配置](#)中的“首页URL”地址。
2. 用户在 OrgID 登录成功后，OrgID 会自动重定向到应用页面，同时会携带授权码，此时，应用在后端需要通过授权码调用 OrgID 的 2 个接口（getToken 接口、getUserInfo 接口），换取具体的用户信息。

OrgID 接口信息地址

- [获取AccessToken](#)
- [获取用户信息](#)

图 3-5 通过授权码获取 token 接口

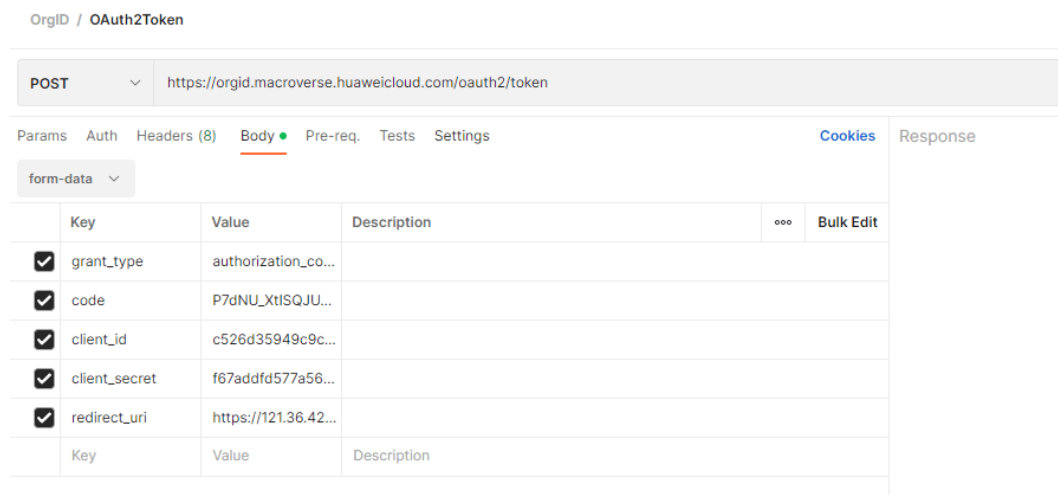
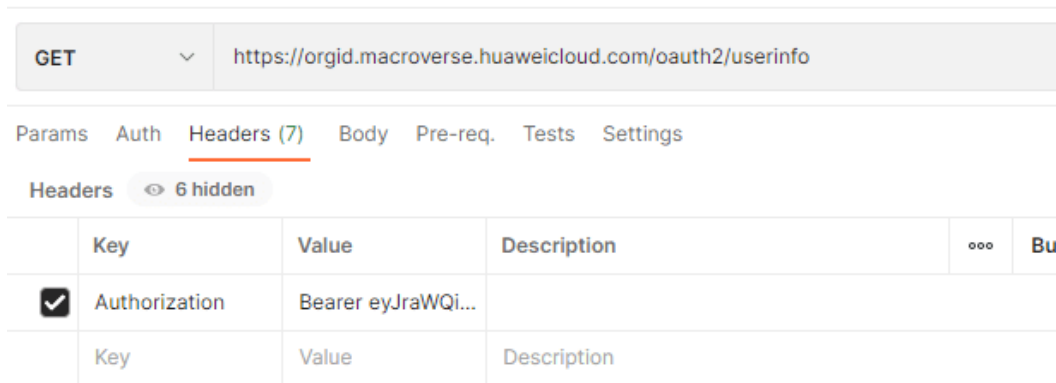


图 3-6 通过 token 获取用户信息接口



3.2 基于 IPDCenter 的单点登录配置

背景介绍

为实现IPDCenter与各个产品之间的集成，当前CraftArts IPDCenter提供单点登录能力，在部分无SSO场景下，可实现各个产品与IPDCenter之间的单点登录。

前提条件

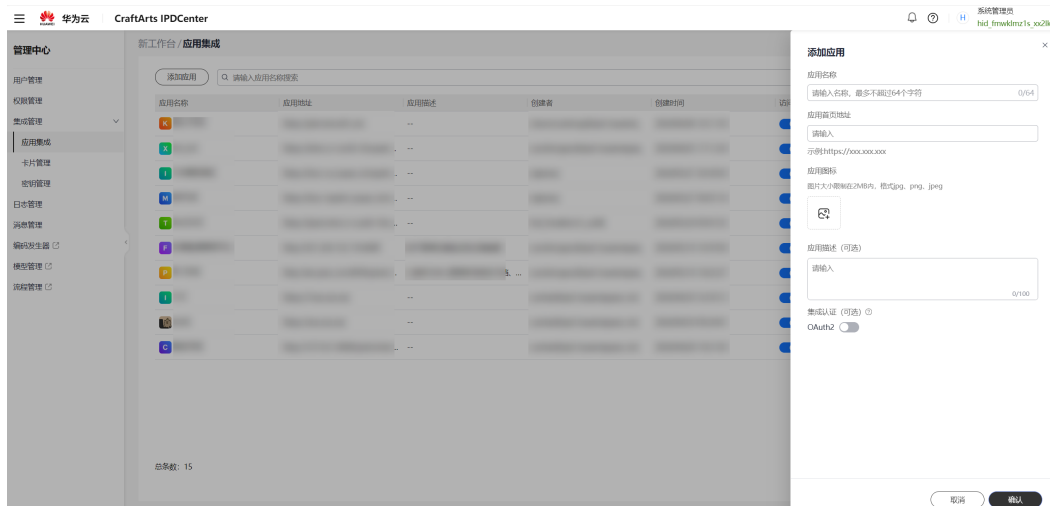
已有CraftArts IPDCenter登录账号，且具有集成管理员角色。

基本信息配置

步骤1 登录CraftArts IPDCenter 管理中心。

步骤2 选择左侧导航栏的“集成管理 > 应用集成”，单击“添加应用”。

图 3-7 添加自建应用



1. 输入应用名称：如“自建App”。
2. 输入应用首页地址：如“https://example.com”
3. 上传应用图标，图片大小限制在2MB内，格式jpg、png、jpeg。
4. 将集成认证OAuth2开关打开。
5. 退出地址，应用的退出地址，请以http或https开头，例如：https://xx.xx/logout。
6. Refresh Token有效期（秒），允许用户在多久时间内不用重新登录应用的时间。
7. Access Token有效期（秒），允许用户在多久时间内保持登录应用的时间。

步骤3 单击“确定”，并复制OAuth2集成认证的凭据。

图 3-8 认证凭据



----结束

📖 说明

配置“首页URL”，此参数配置为产品页面的URL，集成CraftArts IPDCenter单点登录后，在登录成功后用户会跳转到相应的地址，跳转地址的域名需要和此配置保持一致。

接口对接指导

应用需要完成的逻辑

1. 在用户访问应用时，应用需要判断是否已经登录，如果没有登录，需要重定向到 CraftArts IPDCenter登录页面，重定向的地址为下面接口中的**获取授权码接口**。
2. 用户在 CraftArts IPDCenter登录成功后，CraftArts IPDCenter会自动重定向到应用页面，同时会携带授权码，此时，应用在后端需要通过授权码调用 CraftArts IPDCenter的2个接口（getToken接口、getUserInfo接口），换取具体的用户信息。

接口信息地址

- [应用实例授权登录，获取授权码接口](#)
- [用户级Token获取](#)
- [Token刷新](#)
- [退出登录接口](#)

3.3 IPDCenter 外部集成开发

3.3.1 开发概述

关于认证

在系统间API交互时，需要对API做认证。CraftArts IPDCenter定义了API认证的规范与SDK，应用在与CraftArts IPDCenter的API集成过程中，需要按照此规范完成认证逻辑。

认证集成简介

CraftArts IPDCenter提供卡片市场功能，在卡片市场中，第三方应用可以上传自定义卡片，卡片会调用第三方应用的API，此时API需要检验请求的合法性，通过集成认证SDK，可实现API合法性校验。

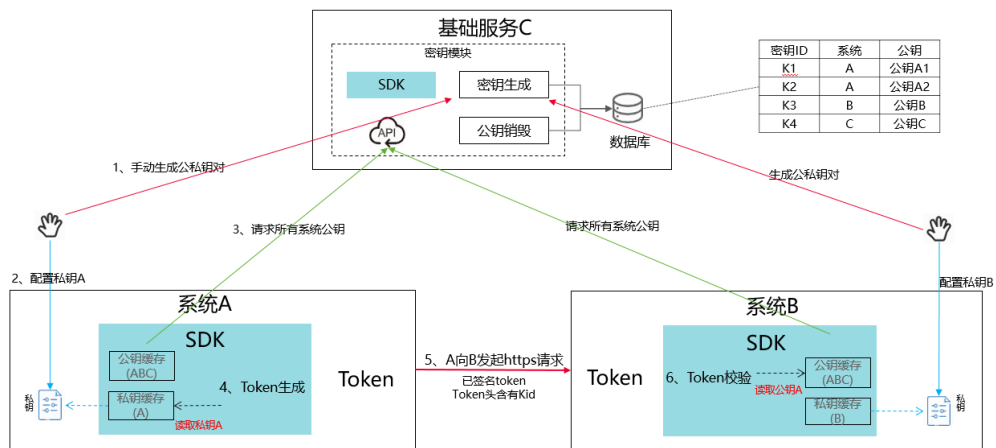
工作原理

统一认证SDK是基于JWT实现的一种认证机制，Json web token (JWT)是在网络应用环境间为了传递声明而执行的一种基于JSON的开放标准(RFC 7519)。该token的设计紧凑且安全，特别适用于分布式站点的单点登录（SSO）场景。JWT的声明一般被用来在身份提供者和服务提供者间传递被认证的用户身份信息，以便从资源服务器获取资源。它也可以被增加一些其它业务逻辑所必须的声明信息。该token可直接被用于认证，也可被加密。

API请求方将请求的用户信息利用私钥哈希生成签名，将用户信息与签名合并生成token，在调用API时将token传递给被调用方，API被调用方获取到此token后，利用公钥验证此token的正确性。

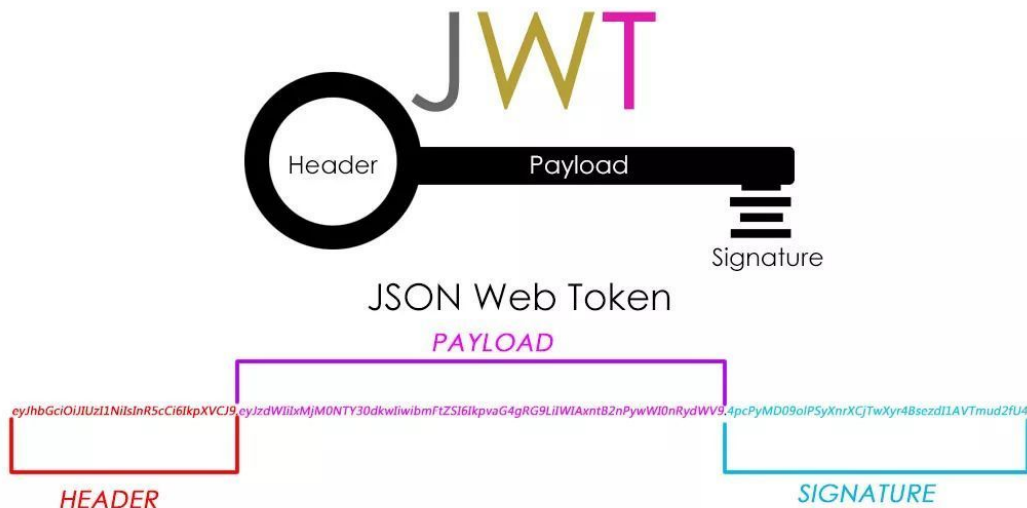
图 3-9 机机认证方案

机机认证方案—token签名模式（SDK）



JWT 的组成

图 3-10 Json web token



一个JWT实际上就是一个字符串，它由三部分组成，头部、载荷与签名。形式为：

A.B.C

样本数据如下：

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJhdWQiOiI0IiwiaWF0IjoiYXVsdCIsImV4cCI6IkpvaG4gRG9LJWIAxntB2nPywWl0nRydWV9.4pcPyMD09oiPSyXnrXCJTWYx48sezdI1AVTmud2FU4
```

头部 (Header)

JWT需要一个头部，头部用于描述关于该JWT的最基本的信息，例如其类型以及签名所用的算法等。

头部信息是经过BASE64编码的，包括以下几个属性：

表 3-2 属性

位置	字段	名称	是否必填	字段类型	字段说明
Header	typ	类型	是	String	固定值：JWT
Header	alg	算法	是	String	固定值：RSA
Header	kid	密钥ID	是	String	密钥ID，由 CraftArts IPDCenter颁发。

载荷 (Payload)

载荷信息是经过BASE64编码解码后的JSON对象，包括以下几个属性：

表 3-3 属性

位置	字段	名称	是否必填	字段类型	字段说明	使用方式
Payload	iss	发行人	否	String	该JWT的签发者	如果由机机自动触发，则放appid; 如果由人触发调用，可以放 userName:userId:tenantId
Payload	sub	主题	否	String	该JWT所面向的用户	暂时为空，建议为被调用方appid
Payload	aud	接收系统	否	String[]	接收该JWT的一方	目标系统的{域名:端口/上下文根}
Payload	exp	过期时间	是	Date	标识JWT的过期时间	这里是一个Unix时间戳，默认5分钟
Payload	nbf	生效时间	否	Date	在此之前JWT不可用	暂未启用
Payload	iat	签发时间	是	Date	标识JWT的签发时间	Unix时间戳
Payload	jti	JWTID	否	String	JWT唯一标识符，防重放攻击	暂未启用
Payload	claims	私有声明	否	Map<String, Object>	一个集合，可自定义字段	暂未启用

签名 (Signature)

将上面的两个Base64编码后的字符串都用英文句号“.”连接在一起（头部在前），就形成了：

```
eyJraWQwOilyZWQ3YWQ4My0zYzhlLTQzYTktOTE2ZC1jOGI2ZWMyYzExNjUilCJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiIiLCJzdWliOiIiLCJleHAiOiJlMTEwMjA0MTgsImhhdCI6MTcxMTAyMDEwOCwianRpIjpubWxsLCljbGFpbXMiOm51bGx9
```

我们将上面拼接完的字符串用RSA256算法进行加密。在加密的时候，我们还需要提供一个密钥（secret）。如果我们用mystar作为密钥的话，那么就可以得到我们加密后的内容：

```
eyJraWQwOilyZWQ3YWQ4My0zYzhlLTQzYTktOTE2ZC1jOGI2ZWMyYzExNjUilCJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiIiLCJzdWliOiIiLCJleHAiOiJlMTEwMjA0MTgsImhhdCI6MTcxMTAyMDEwOCwianRpIjpubWxsL
```

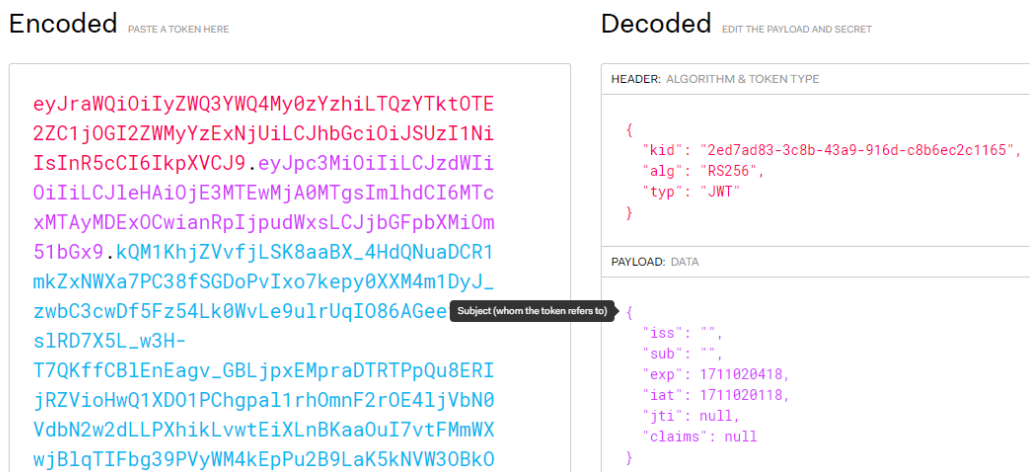
```
CjbbGFpbXMiOm51bGx9.  
kQM1KhjZVvfjLSK8aaBX_4HdQNuaDCR1mkZxNWxa7PC38fSGDoPvIxo7kepy0XXM4m1DyJ_zwbC3cwDf5Fz54  
Lk0WvLe9ulrUqIO86AGeetWfLsLRD7X5L_w3H-  
T7QKffCBLEnEagv_GBLjpxEMpraDTRTPpQu8ERljRZVioHwQ1XDO1PChgpal1rhOmnF2rOE4ljVbN0VdbN2w2dL  
LPXhikLwvtEiXlnBKaaOul7vtFmMwXwjBlqTIFbg39PVyWM4kEpPu2B9LaK5kNVW3OBkOLCP_trizMHINZIHdE  
H9gHh2_h8xLKHpAthHXUWvDprtSwWBdWH2rDZIMW03rhnwpXg3ABknRUPNfrtTNY8HDvEGUUVLmfkAdTs_D  
2y1RNt-  
MRn_fH50DiINOGi_Zf4uErrf5yHL1qln5cprID64ETctd0luzhCp7weK1TlxzIGqGXClHf6_yjwJ2kC7Xikdk38Zx946Z  
NB32jdfWbJDZoUnL2v3_T3KThW21huLgB
```

最后，我们在上面生成的token上加一个前缀“Internal:”，用于标识此token的类型，最后得到的token是：

```
Internal:eyJraWQiOiIyZWQ3YWQ4My0zYzhiLTQzYTktOTE2ZC1jOGI2ZWMyYzExNjUiLCJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiIiLCJzdWIiOiIiLCJleHAiOiJ3MTUwMjEwIiwiaWF0IjoiMTUwMjEwIiwiaXNjaWkiOiIiLCJzdmVudWxsLCJjbGFpbXMiOiOm51bGx9.kQM1KhjZVvfjLSK8aaBX_4HdQNuaDCR1mkZxNWxa7PC38fSGDoPvIxo7kepy0XXM4m1DyJ_zwbC3cwDf5Fz54Lk0WvLe9ulrUqIO86AGeetWfLsLRD7X5L_w3H-  
T7QKffCBLEnEagv_GBLjpxEMpraDTRTPpQu8ERljRZVioHwQ1XDO1PChgpal1rhOmnF2rOE4ljVbN0VdbN2w2dL  
LPXhikLwvtEiXlnBKaaOul7vtFmMwXwjBlqTIFbg39PVyWM4kEpPu2B9LaK5kNVW3OBkOLCP_trizMHINZIHdE  
H9gHh2_h8xLKHpAthHXUWvDprtSwWBdWH2rDZIMW03rhnwpXg3ABknRUPNfrtTNY8HDvEGUUVLmfkAdTs_D  
2y1RNt-  
MRn_fH50DiINOGi_Zf4uErrf5yHL1qln5cprID64ETctd0luzhCp7weK1TlxzIGqGXClHf6_yjwJ2kC7Xikdk38Zx946Z  
NB32jdfWbJDZoUnL2v3_T3KThW21huLgB
```

将上面的token在jwt网站（<https://jwt.io/>）上进行解析，可以看到具体的内容信息，如下：

图 3-11 解析内容



密钥

上面的token在签名时需要用到一个密钥，方案采用了RSA非对称密钥算法，由私钥进行签名，公钥进行验证签名。此密钥对的生命周期由IPDCenter基础服务管理。由IPDCenter基础服务颁发密钥对，每个系统拥有自己的一对密钥，在颁发完成密钥后，IPDCenter将记录所有的公钥信息、密钥ID信息、APPID等信息。私钥则由应用自己保存。

公钥获取

公钥是用来检验token的，而且校验token时需要使用与私钥对应的公钥，因此，应用需要定时调用IPDCenter提供的接口，查询所有的公钥信息，加载到内存，检验token时找到对应的token进行检验。检验接口的规格如下：

接口地址：

<https://域名/随机数文根/basic/mgmt/public/api/v1/keymanage/getPublicKeyList>

接口header:

Authorization: 上面生成的token
Content-Type: application/json

接口body:

```
{  
  "offset": 0,  
  "limit": 2000  
}
```

接口响应示例:

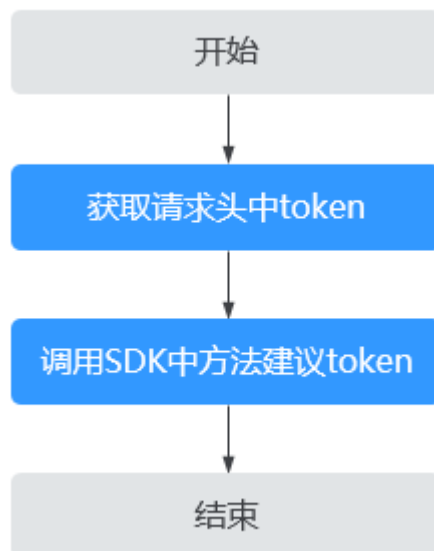
```
{  
  "code": "IPDC.01020000",  
  "datas": [  
    {  
      "appld": "121",  
      "appName": "应用名称",  
      "createTime": "2024-03-22T17:42:58.830872000",  
      "creator": "xx",  
      "isDeleteAble": "N",  
      "keyId": "密钥ID, token中解析出密钥ID,根据ID找到对应的公钥,检验token是否合法",  
      "publicKey": "公钥信息"  
    }  
  ],  
  "message": "success",  
  "totalCount": 3  
}
```

3.3.2 总体开发思路

工作流程

使用统一认证集成的工作流程如下:

图 3-12 工作流程



在第三方应用作为被调用方时，需要检验请求中的认证凭据的合法性:

1. 获取请求头中的认证凭据-Token。

获取请求头中名为**Authorization**的认证凭据。

2. 校验认证凭据。

调用SDK中的检验方法，将token传递给检验方法，如果token合法，可返回调用者的信息。

3.3.3 准备工作

开发技能要求

- 熟悉Java语言，能够编写Java语言代码。
- 了解基本的Http协议，熟悉Http的调用过程。

收集信息

表 3-4 收集信息

项目	说明
私钥/公钥获取URL/密钥ID/公钥/appld	访问私钥是在CraftArts IPDCenter API交互过程身份凭证。通过私钥生成认证凭据，用于确保请求者身份的正确性。公钥获取URL自动获取公钥的地址，通过此地址，可自动获取公钥信息，用于检验认证凭据的正确性。 公钥获取URL地址为： https://域名/文根/basic/mgmt/public/api/v1/keymanage/getPublicKeyList appld、公钥、私钥与密钥ID见下面 获取方式 。

获取方式

获取CraftArts IPDCenter系统管理员账号密码，通常为购买者注册的华为云账号。

步骤1 登录CraftArts IPDCenter系统首页。

步骤2 进入管理中心。

图 3-13 管理中心



步骤3 在应用集成页面添加一个自己产品的应用。

图 3-14 应用集成



步骤4 在密钥管理页面创建密钥，创建成功后将会弹出密钥编码，私钥信息、应用ID为appld。

图 3-15 密钥管理



----结束

环境要求

- 从[Oracle官网](#)下载并安装推荐使用的JDK版本。
推荐使用的JDK版本：JDK 8以上版本。
- 根据开发需要，下载并安装开发工具。
 - 从[Eclipse官网](#)下载并安装Eclipse IDE for Java Developers最新版本。
 - 从[IntelliJ IDEA官网](#)下载并安装IntelliJ IDEA开发工具。

安装 SDK

步骤1 解压缩下载的Java SDK。

步骤2 将解压缩后的JAR包引入到已有的Maven工程中。

以IntelliJ IDEA开发工具为例。

1. 在“resources”目录下创建一个“lib”目录，并将JAR包存放至该目录下。
2. 单击“File > Project Structure”。
3. 在弹出的窗口中，选择“Modules > Dependencies”，单击“+ > JARs or Directories...”。

4. 选中引入的JAR包，单击“OK”。
5. 单击“Apply”。

步骤3 在pom.xml文件中添加依赖。

在<dependencies>节点中加入如下配置：

```
<dependency>
  <groupId>com.auth0</groupId>
  <artifactId>java-jwt</artifactId>
  <version>4.4.0</version>
</dependency>
```

----结束

3.3.4 快速开始

本章节以一个被调用API为例，进行检验API中token的合法性。

操作步骤

在一个待集成的应用系统中，找到一个API接口，如/testApi。

步骤1 在工程中增加配置信息，具体配置获取方式见准备工作中收集信息章节。

```
jwt.service.publickey.url: 公钥获取地址
jwt.secret.id: 密钥ID
jwt.secret.appld: appld
jwt.secret.key: 私钥
jwt.secret.publickey: 公钥
```

步骤2 定时加载公钥代码。

```
public void autoRefreshPublicKey() {
    Calendar calendar = Calendar.getInstance();
    Date currentTime = calendar.getTime();
    Timer timer = new Timer();
    timer.schedule(new TimerTask() {
        @Override
        public void run() {
            try {
                //定时从IPDCenter获取所有公钥信息
                List<String> publicKeys = HttpClientUtil.post(url, token);
                for (String publicKeyStr : publicKeys) {
                    // 公钥加载到内存中
                    PublicKey publicKey = getPublicKey(publicKeyStr);
                }
            } catch (Exception e) {
                log.error("autoLoadPublicKey error", e);
            }
        }
    }, currentTime, refreshTime);
}

public PublicKey getPublicKey(String publicKeyStr) {
    byte[] keyBytes = Base64.getDecoder().decode(publicKeyStr);
    X509EncodedKeySpec x509EncodedKeySpec = new X509EncodedKeySpec(keyBytes);
    KeyFactory keyFactory;
    try {
        keyFactory = KeyFactory.getInstance("RSA");
        return keyFactory.generatePublic(x509EncodedKeySpec);
    } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
        log.error("Failed to load the public key to the JWT token. Exception: {}", e.getMessage());
    }
    return null;
}
```

步骤3 应用启动时加载私钥。

```
public PrivateKey getPrivateKey(String privateKeyStr) {
    try {
        byte[] privateKeyBytes = Base64.getDecoder().decode(privateKeyStr);
        PKCS8EncodedKeySpec keySpec = new PKCS8EncodedKeySpec(privateKeyBytes);
        KeyFactory keyFactory = KeyFactory.getInstance("RSA");
        return keyFactory.generatePrivate(keySpec);
    } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
        log.error("Failed to load the private key to the JWT token. Exception: {}", e.getMessage());
    }
    return null;
}
```

步骤4 检验token的方法。

```
private static boolean verify(HttpServletRequest httpServletRequest, String publicKeyId, String token) {
    // 从公钥缓存中获取对应的密钥编号的公钥
    PublicKey publicKey = PublicKeyCache.getInstance().publicKeysCache.get(publicKeyId);
    try {
        Algorithm algorithm = Algorithm.RSA256((RSAPublicKey) publicKey, null);
        DecodedJWT decodedJwt = JWT.require(algorithm).ignoreIssuedAt().build().verify(token);
        Date expiresAt = decodedJwt.getExpiresAt();
        // 验证token是否过期
        if (expiresAt.before(new Date())) {
            LOGGER.error("The token has expired.");
            return false;
        }
        // 验证自己是否为接收方
        List<String> audience = decodedJwt.getAudience();
        if (audience != null && !audience.isEmpty() && !validAudience(httpServletRequest, audience)) {
            return false;
        }
    } catch (JWTVerificationException | IllegalArgumentException | SdkException e) {
        LOGGER.error("Verification failed. Jwt Token verification exception: {}", e.getMessage());
        return false;
    }
    if (LOGGER.isDebugEnabled()) {
        LOGGER.debug("Token has verify successfully.");
    }
    return true;
}
```

步骤5 生成token的方法。

```
public static String generateToken(JwtToken jwtToken, boolean isUnique) {
    // 私钥从缓存中通过编号获取
    PrivateKey privateKey = PrivatekeyCache.getInstance().privateKeysCache.get(jwtToken.getKeyId());
    Algorithm algorithm = Algorithm.RSA256(null, (RSAPrivateKey) privateKey);
    Date issuedAt = new Date();
    // 未设置过期时间，则默认签发时间后5分钟内有效
    Date expiresAt =
        new Date(issuedAt.getTime() + (jwtToken.getExpirationTime() <= 0 ? 300000 :
jwtToken.getExpirationTime()));
    return "Internal:" + JWT.create()
        .withKeyId(jwtToken.getKeyId())
        .withIssuer(jwtToken.getIssuer())
        .withSubject(jwtToken.getSubject())
        .withAudience(jwtToken.getAudience())
        .withExpiresAt(expiresAt)
        .withIssuedAt(issuedAt)
        .withJWTId(jwtToken.getJwtId())
        .withClaim("claims", jwtToken.getClaimMap())
        .sign(algorithm);
}
```

步骤6 在“/testApi”接口中获取token。

```
String token = request.getHeader("Authorization");
```

步骤7 调用检验方法，检验token的合法性，并获取调用者信息。


```
String issuer = verify(token);
```

----结束

结果验证

步骤1 调用SDK中的方法生成一个临时的token。

```
String token = generateToken();
```

步骤2 调用API调试工具调用“/testApi”接口，调用时将步骤1中生成的token放入header中。

```
curl http://ip:port/xx/textApi -H "Authorization:token"
```

----结束

3.3.5 实现一个认证 API

3.3.5.1 开发指导

创建一个Spring的API，在API中获取token，并检验token的合法性。

开发过程

步骤1 在Spring boot项目工程中创建一个API。

```
@Controller
@RequestMapping("/api")
public class TestController {

    @GetMapping("/testApi")
    public String test(HttpServletRequest request) {
        return "success";
    }
}
```

步骤2 获取token并检验合法性。

```
@GetMapping("/testApi")
public String test(HttpServletRequest request) {
    String token = request.getHeader("Authorization");
    String issuer = verifyAndReturnIssuer(token);
    if (issuer.equals("user")) {
        return "success";
    }
    return "error";
}
```

----结束

3.3.5.2 注意事项

1. 当检验失败时返回的异常需要友好，用户能清楚知道错误原因。
2. 校验功能最好在一个filter中实现，避免所有接口都实现一次检验。

3.3.6 代码示例

代码示例

```
@Controller
@RequestMapping("/api")
```

```
public class TestController {
    @GetMapping("/testApi")
    public String test(HttpServletRequest request) {
        String token = request.getHeader("Authorization");
        String issuer = verifyAndReturnIssuer(token);
        if (issuer.equals("user")) {
            return "success";
        }
        return "error";
    }
}
```

3.3.7 常用调试方法

3.3.7.1 查看日志

在集成SDK后，在启动及运行过程中可能会出现错误，此时通过查看日志排查错误。

常见的错误有：

1. 密钥相关的配置不正确，获取不到公钥信息。
2. 密钥配置不正确或token失效，token检验失败。

4 统一数据模型指南

4.1 对外 API 规范

API 规范示例

1. 为了对外API在调用时有用户信息，API调用时需要在header中传递Modifier、Creator字段。
值格式为：*用户名+空格+用户id*，如图4-1所示。

图 4-1 示例

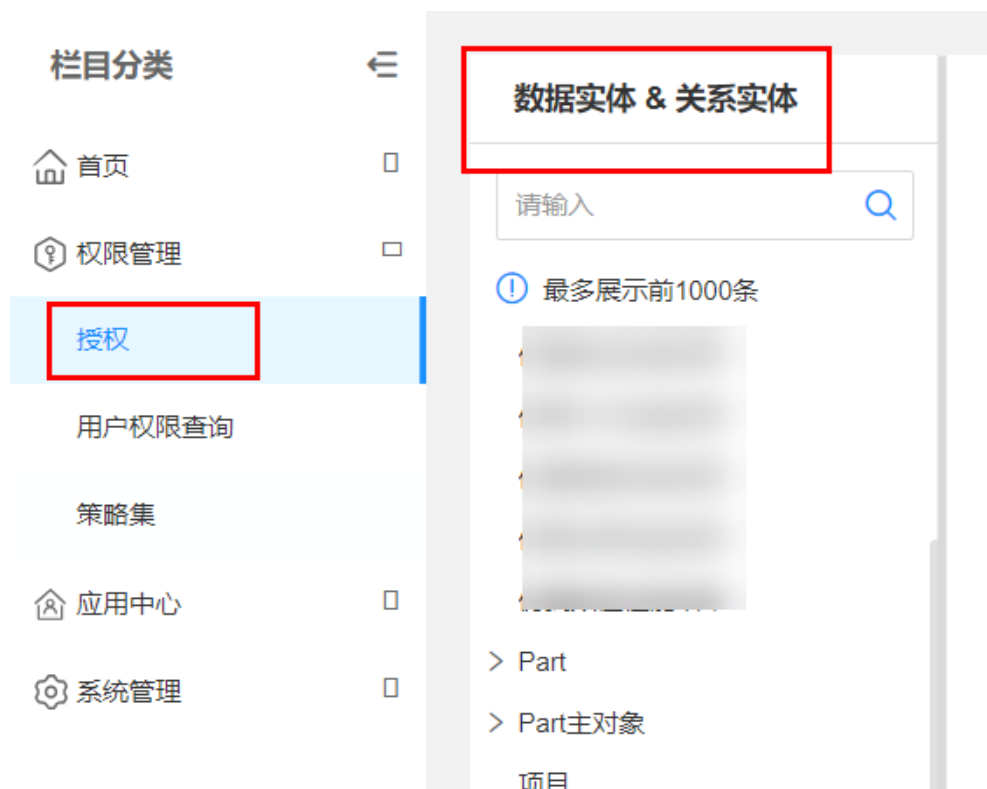
Modifier	lu [redacted]@reiapaas.com 100860000 [redacted]
Creator	lu [redacted]@reiapaas.com 100860000 [redacted]

2. 具有AccessControlled功能的实体，其API做了权限管控，权限管理依赖团队信息，因此在传递机机token时需要添加团队信息，即机机的jwt token中iss字段。
格式为：*用户名:用户id:租户id:团队名*

判断实体是否具有 AccessControlled 功能

进入生命周期数据模型服务，选择“权限管理”，在“授权”页面能看到的实体具有AccessControlled功能。

图 4-2 数据实体&关系实体



5 流程配置指南

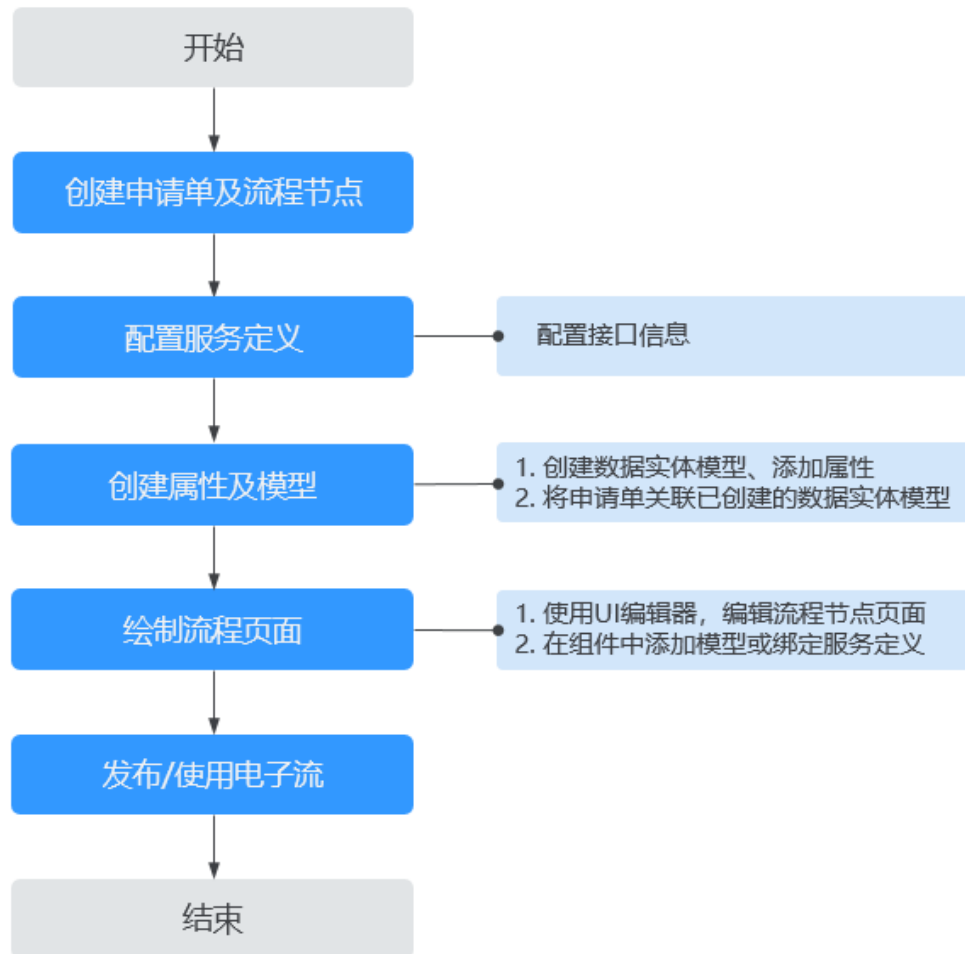
5.1 快速入门

5.1.1 入门指引

服务定义流程基于Camunda（开源版本Apache2.0协议，商业友好，明确专利授权）流程引擎拓展华为场景实现，支持BPMN2.0用于工作流和流程自动化。提供流程从设计、运行到管理的能力，提供Restful服务供业务端调用，是一个针对流程全生命周期的高性能一站式平台服务。

入门指引

图 5-1 操作流程



📖 说明

下文操作均已创建“自制件BOM审核发布流程”进行举例。

5.1.2 步骤一：创建申请单及流程节点

使用流程定义前需要创建流程申请单，并在申请单中创建流程节点、添加节点属性。

创建流程申请单

步骤1 在角色桌面首页，单击左上角的☰，选择“功能总览 > 流程管理 > 流程模板定义”，进入流程定义系统。

步骤2 在左侧导航中，选择“流程管理 > 流程申请单”，进入流程申请单列表页面。

步骤3 单击“新增”，输入中英文名称及描述。

步骤4 单击“保存并创建流程模板”。

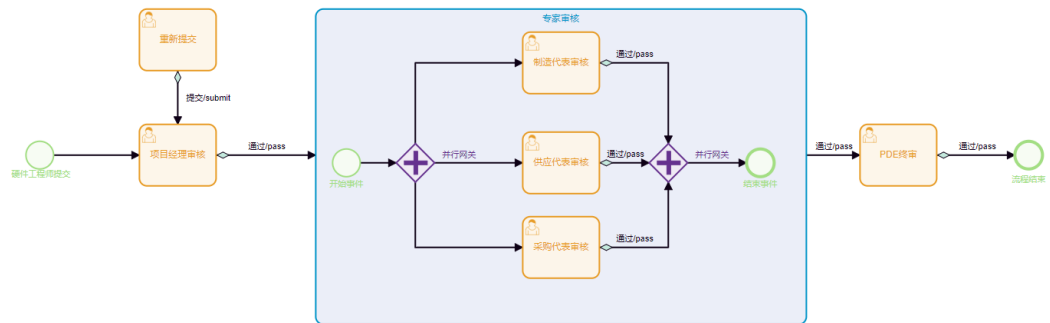
----结束

创建流程节点

流程申请单创建完成后，返回至申请单列表页面，单击操作列的“编辑”，进入流程编辑页面。

绘制如图5-2所示的流程图。

图 5-2 自制件 BOM 审核发布流程



步骤1 添加“开始事件”“结束事件”“用户任务”“并行网关”和“内嵌子流程”组件到画布中。

修改组件名称：双击组件修改名称，或单击组件，在右侧的属性面板中修改组件名称。

说明

- “重新提交”用户任务组件，支持提交人发起流程后，将流程撤回回到“重新提交”节点，重新修改内容后再次提交流程。
- 提交人在Start节点可关闭流程，在其他用户任务节点可撤回重填。

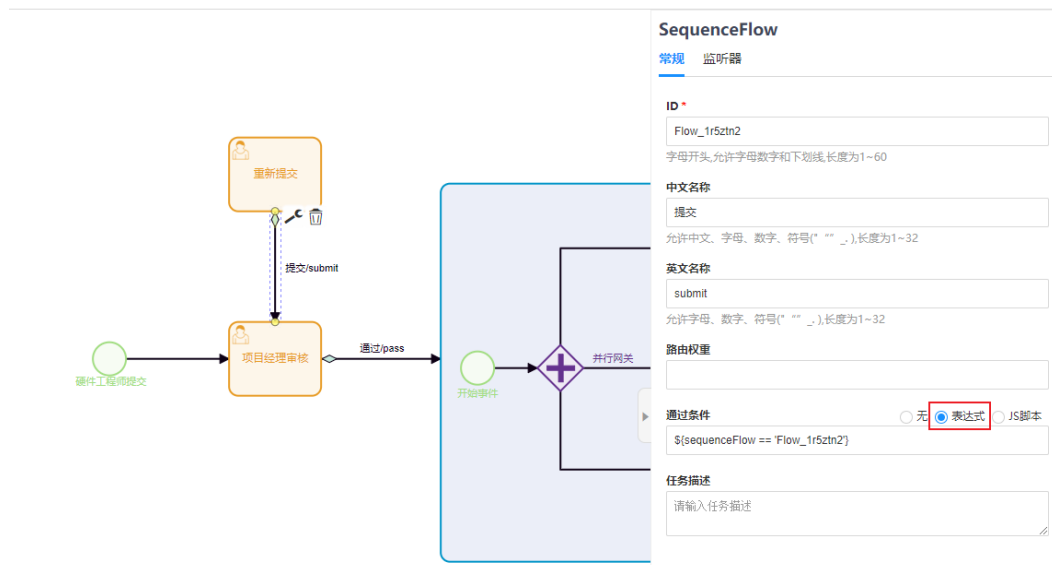
步骤2 单击组件，使用箭头连接线，连接组件。

连接时注意箭头的走向。

步骤3 选择连接线，在右侧的面板中设置“中文名称”及“英文名称”。

对于用户任务，如果有不止一个走向，还需要在“通过条件”中选择“表达式”来判断走向。

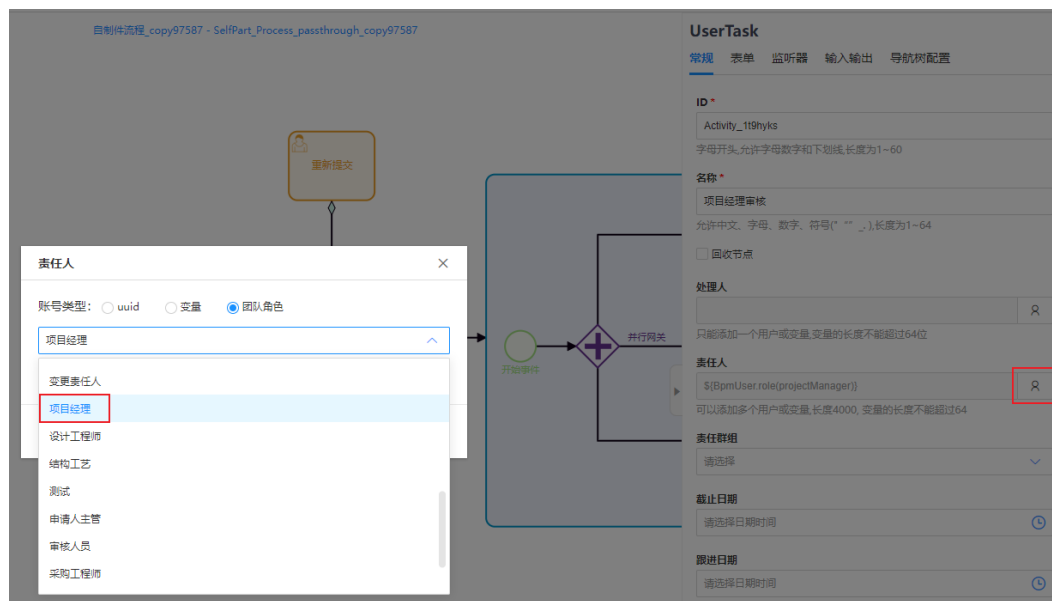
图 5-3 通过条件



步骤4 设置任务组件中的责任人信息。

单击“用户任务”组件，在“责任人”中选择已配置的**团队角色**。即任务由该角色处理，在提交页面也需要配置对应的参与者。

图 5-4 设置审批责任人



步骤5 修改完成后，单击右上角的“保存”。

----结束

5.1.3 步骤二：配置服务定义

服务定义即接口，当前调用BOM接口都需先调用获取机机认证token接口，然后再调用相应的业务接口。

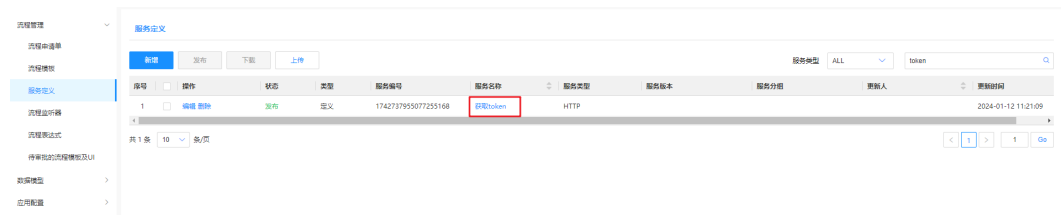
- 若不需要配置机机认证，则按照[配置BOM业务接口](#)方式进行配置。
- 若需要配置机机认证，则按照[配置组合接口](#)方式进行配置，需要提前根据实际情况配置一个机机接口。

获取机机认证 token 接口

在左侧导航中，选择“流程管理 > 服务定义”，进入服务定义列表页面，搜索token查看接口详情。

系统已配置好可以直接使用。

图 5-5 token 接口

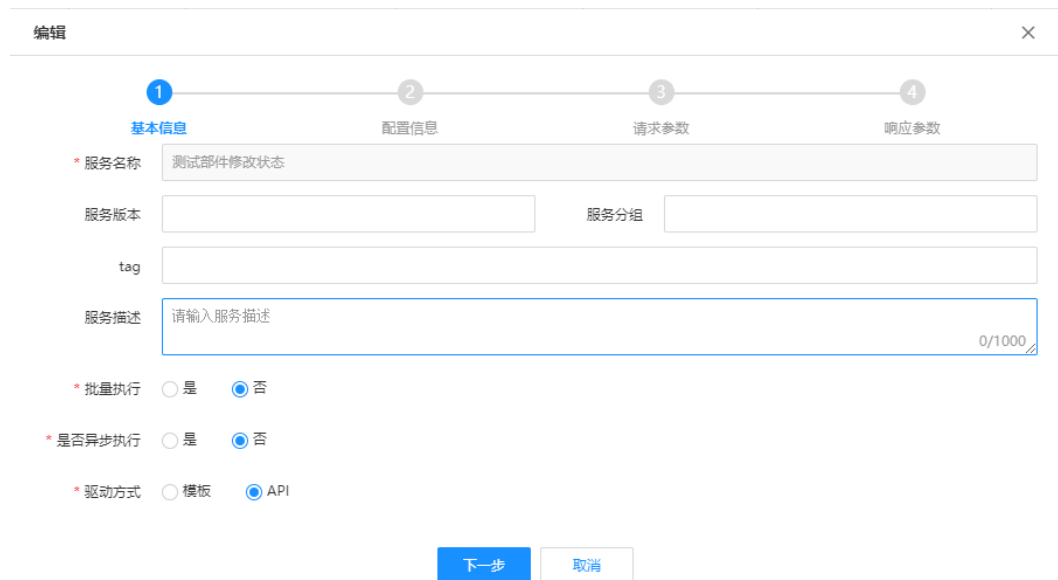


配置 BOM 业务接口

进入流程定义，选择“流程管理 > 服务定义”，进入服务定义列表页面，单击“新增”，进入新增页面。

步骤1 设置“基本信息”，如[图5-6](#)所示。

图 5-6 基本信息



步骤2 设置“配置信息”。

图 5-7 配置信息

- **服务地址：**接口地址。
- **方法类型：**选择接口对应的“GET”或“POST”，GET类型无需设置Body参数。
- **Header参数：**参照表5-1添加Header参数。

表 5-1 Header 参数

参数名称	参数值	加密
x-jwt-token	`\${x-jwt-token}`	否
Cookie	`\${Cookie}`	否
X-Pdm-lsc-Common-Token	`\${ipdctoken}`	否

- **Query参数（可选）：**根据实际情况填写参数信息。

图 5-8 Query 参数

	参数名称	参数值	加密	操作
1	<input type="text"/>	<input type="text"/>	<input type="checkbox"/> OFF	保存 删除
2	<input type="text"/>	<input type="text"/>	<input type="checkbox"/> OFF	保存 删除

- **Body参数**: 参数类型选择“JSON”，“Rest模板”中设置参数为接口入参。

步骤3 设置“请求参数”。基本信息中，当驱动方式选择API时，需要设置请求参数。

图 5-9 请求参数

请求参数(Query/Header/Body)

序号	操作	参数名	类型	是否为数组	是否必填	是否为模板	位置	默认值
1	编辑 删除	test1	java.lang.String	否	否	否	QUERY	

步骤4 设置“响应参数”。

响应参数按照实际情况填写，如果接口反参为json格式，按照图5-10所示填写即可。

图 5-10 响应参数

响应参数

序号	操作	参数名	类型	是否为数组	描述
1	编辑 删除	data	java.lang.Object	否	

步骤5 单击“保存并发布”。

----结束

配置组合接口

进入流程定义，选择“流程管理 > 服务定义”，进入服务定义列表页面，单击“新增”，进入新增页面。

步骤1 设置“基本信息”，如图5-11所示。

图 5-11 基本信息

编辑

1 基本信息 2 配置信息 3 请求参数 4 响应参数

* 服务名称 组合修改部件状态

服务版本 服务分组

tag

服务描述 请输入服务描述 0/1000

* 批量执行 是 否

* 是否异步执行 是 否

* 驱动方式 模板 API

下一步 取消

步骤2 设置“配置信息”。

服务类型选择“APIEDIT”，然后单击“编辑流程”，绘制如图5-12所示的接口流程图。

图 5-12 配置信息

编辑

1 基本信息 2 配置信息 3 请求参数 4 响应参数

* 服务类型 APIEDIT

* 执行流程 编辑流程

开始事件 服务任务 脚本任务 服务任务 结束事件

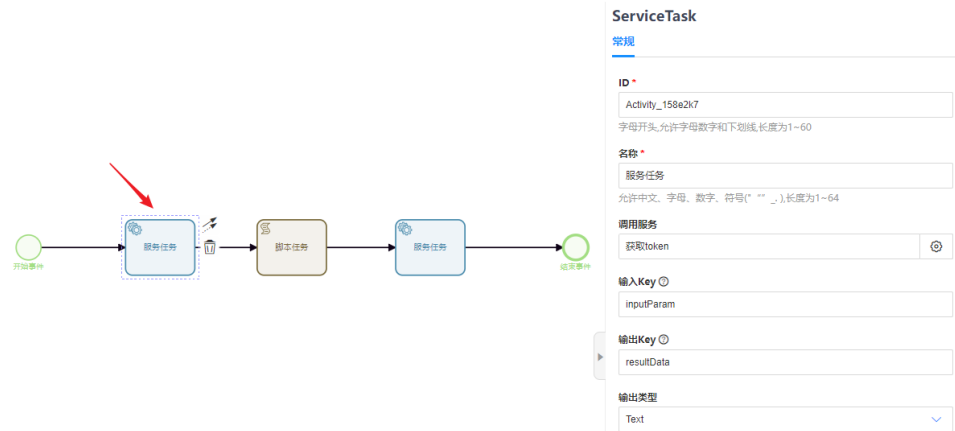
入参模板 请输入json格式 如: [{"key": "value"}] 0/1000

返参模板 请输入json格式 如: [{"key": "value"}] 0/1000

上一步 下一步 取消

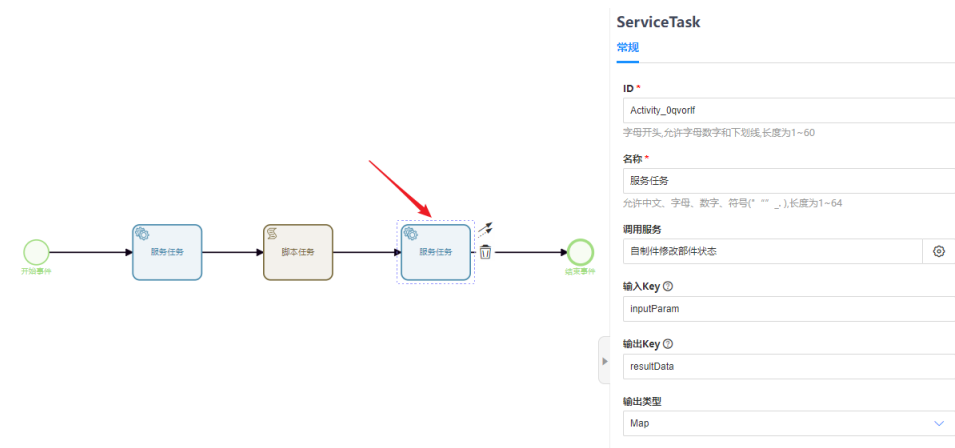
1. 选择第一个“服务任务”组件，设置属性信息，如图5-13所示。

图 5-13 服务任务 1



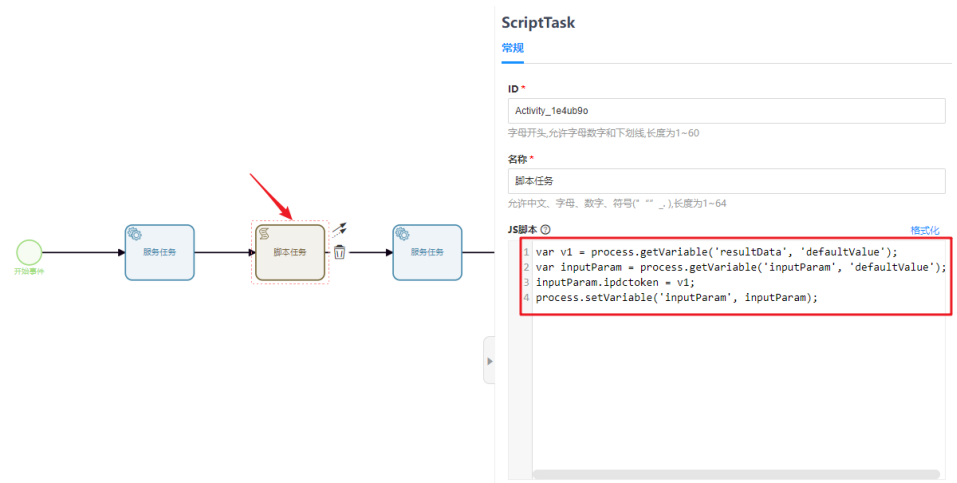
- 调用服务: 获取token
 - 输入Key: inputParam
 - 输出Key: resultData
 - 输出类型: Text
2. 选择第二个“服务任务”组件，设置属性信息，如图5-14所示。

图 5-14 服务任务 2



- 调用服务: 配置业务接口对应的服务定义，这里我们选择“自制件部件修改状态”。
 - 输入Key: inputParam
 - 输出Key: resultData
 - 输出类型: 输出类型根据实际接口来填写，通常选择Map。
3. 选择“脚本任务”组件，填写JS脚本信息，如图5-15所示。
表示将“服务任务1”获取的token传入“服务任务2”。

图 5-15 脚本任务

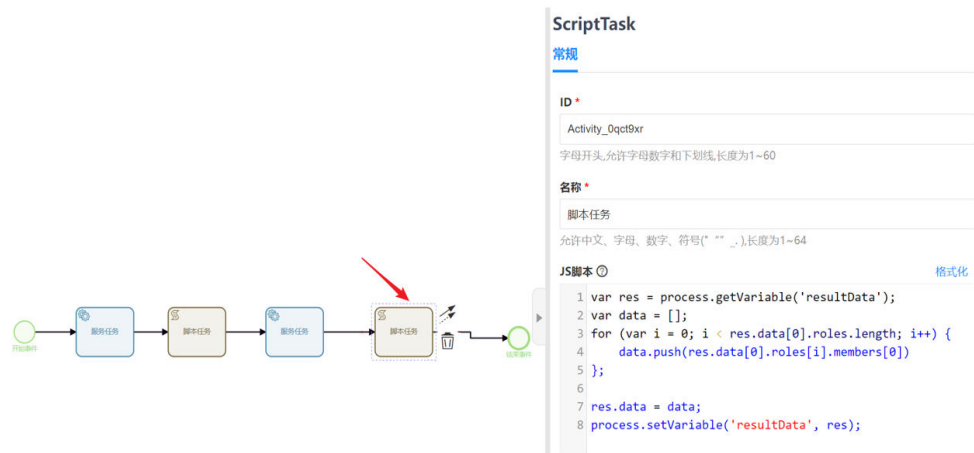


脚本信息如下:

```
var v1 = process.getVariable('resultData', 'defaultValue');
var inputParam = process.getVariable('inputParam', 'defaultValue');
inputParam.ipdctoken = v1;
process.setVariable('inputParam', inputParam);
```

- 4. **可选:** 对于需要对反参数据进行后处理的接口（如格式转换），可以在“服务任务 2”后面衔接一个脚本任务，JS脚本用于对反参的处理。

图 5-16 返回参数处理



示例脚本信息如下:

```
var res = process.getVariable('resultData');
var data = [];
for (var i = 0; i < res.data[0].roles.length; i++) {
    data.push(res.data[0].roles[i].members[0])
};
res.data = data;
process.setVariable('resultData', res);
```

----结束

服务定义验证

对于配置好的服务定义，可以通过下面接口来查看是否可用。

{环境域名（带文根）}/pdmcore/ipdcworkflowservice/console/servicetask/api/{服务定义编号}。

5.1.4 步骤三：创建属性及模型

1. 在属性库中创建属性。
2. 创建数据实体模型，并添加1中创建的属性。
3. 将数据实体模型关联[步骤一](#)中的流程申请单。

创建属性

步骤1 在左侧导航栏中，选择“数据模型 > 属性库”，进入属性库页面。

步骤2 单击“创建”。

图 5-17 创建属性

编码	英文名称	中文名称	英文描述	中文描述	类型	创建者
EA00000011	method_of_procedure	实施方案文档	method_of_procedure-...	method_of_procedure-...	文本	xdmAdmin
EA00000010	Gross_Apply	大规模海量应用	Gross_Apply-大规模海...	Gross_Apply-大规模海...	文本	xdmAdmin
EA00000009	middleware	中间件	middleware-中间件	middleware-中间件	文本	xdmAdmin
EA00000008	QQ	中文名称_aaS120	AS	AS	文本	xdmAdmin
EA00000006	TestParam01	TestParam01	示例属性01	示例属性01	文本	xdmAdmin
EA00000005	expertConclusion	评审结论	评审结论	评审结论	文本	xdmAdmin
EA00000004	expertOpinion	评审意见	评审意见	评审意见	文本	xdmAdmin
EA00000002	tree	树	tree_树	tree_树	文本	xdmAdmin
EA00000001	name	名称	name_名称	name_名称	文本	xdmAdmin

步骤3 根据页面提示，配置如下信息。

图 5-18 基本信息

The screenshot shows a web form titled '创建属性' (Create Attribute). The '基本信息' (Basic Information) section is expanded and contains the following fields:

- 英文名称** (English Name): Text input field, 0/128 characters.
- 中文名称** (Chinese Name): Text input field, 0/128 characters.
- 类型** (Type): Dropdown menu, currently set to '文本' (Text).
- 英文描述** (English Description): Text input field, 0/1024 characters.
- 中文描述** (Chinese Description): Text input field, 0/1024 characters.
- 扩展信息** (Extension Information): Text input field, 0/1024 characters.

The **约束信息** (Constraint Information) section includes:

- 文本长度** (Text Length): Text input field.
- 值存储方式** (Value Storage Method): Radio buttons for '原值' (Original), '大写' (Uppercase), and '小写' (Lowercase). '原值' is selected.
- 必填** (Required): Radio buttons for '是' (Yes) and '否' (No). '否' is selected.

表 5-2 属性基本信息

类型	参数	参数说明
基本信息	英文名称	属性的英文名称。
	中文名称	属性的中文名称。
	类型	类型固定为文本。
	英文描述	属性的英文描述。
	中文描述	属性的中文描述。
	扩展信息	与属性相关的描述。
约束信息	文本长度	超过4000记录为长文本，则数据输入长度不受此限制，如果属性已经基线，则修改此值后，不会改变文本类型。
	值存储方式	<ul style="list-style-type: none"> 原值：实例值按照实际输入的字符串存储。 大写：将实例值中小写字符串转换为大写字符串存储。 小写：将实例值中大写字符串转换为小写字符串存储。

步骤4 单击“保存”。

----结束

创建数据实体模型

步骤1 在左侧导航栏中，选择“数据模型 > 数据实体”，进入数据实体页面。

步骤2 单击“创建”，根据页面提示，配置如下信息。

图 5-19 创建数据实体

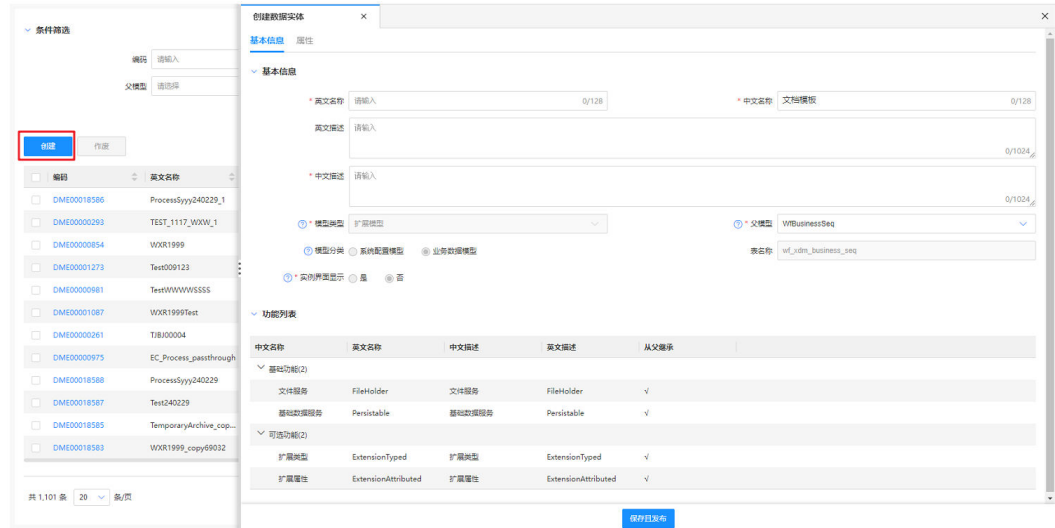


表 5-3 参数说明

参数	说明
英文名称	数据实体的英文名称。 需以大写字母开头，只能包含字母和数字，长度不能超过128个字符。
中文名称	数据实体的中文名称。 只能包含中文、英文字母、数字和“_”，长度不能超过128个字符。
英文描述	数据实体的英文描述。 不能包含中文字符，长度不能超过1024个字符。
中文描述	数据实体的中文描述。 支持各种字符，长度不能超过1024个字符。
模型类型	数据实体的模型类型，只支持“扩展模型”。
父模型	<ul style="list-style-type: none"> WfBusinessSeq：申请单对象 WfFormdata：表单对象 WfBusinessObject：评审对象
模型分类	提供系统配置模型和业务数据模型，当前只支持选择“业务数据模型”，且不可修改。
表名称	数据实体存储至数据库的表名称，默认格式为“应用名称_数据实体英文名称”，不可修改。

参数	说明
实例界面显示	用于标识“数据实例”界面展示/过滤此对象。 与“父模型”的“实例界面显示”一致，且不可修改。如需修改，需在“父模型”所选择的实体中修改。

步骤3 单击“保存且发布”。

----结束

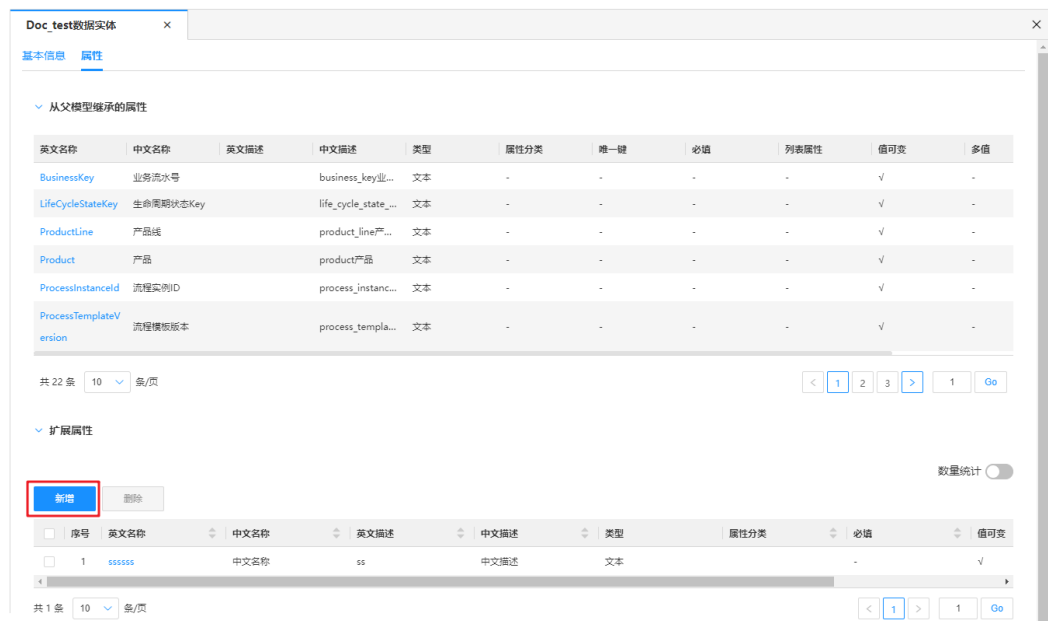
在模型中添加属性

创建数据实体完成后，将属性添加至数据实体中。

步骤1 选择“属性”页签，进入属性页面。

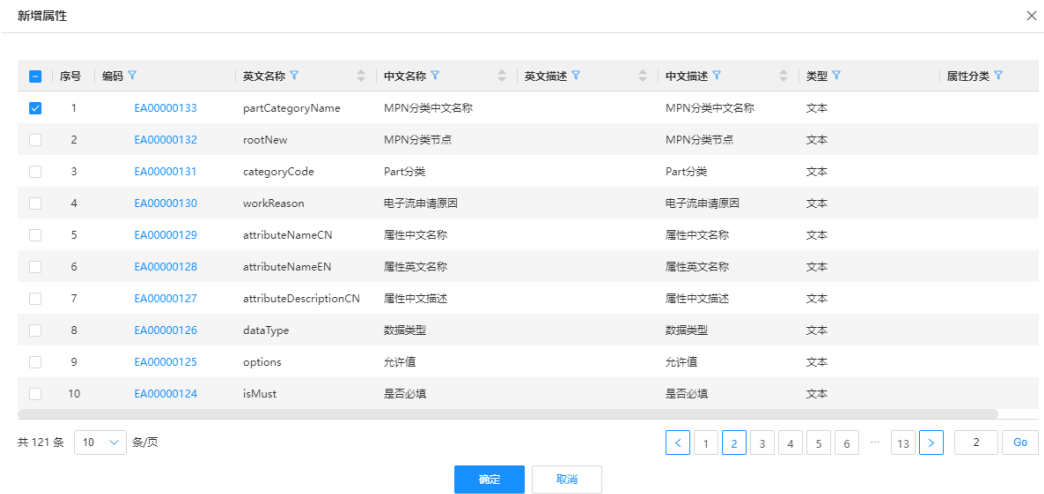
步骤2 在“扩展属性”栏中，单击“添加”。

图 5-20 添加扩展属性



步骤3 在弹出的窗口中，勾选需要添加的属性，单击“确定”。

图 5-21 新增属性



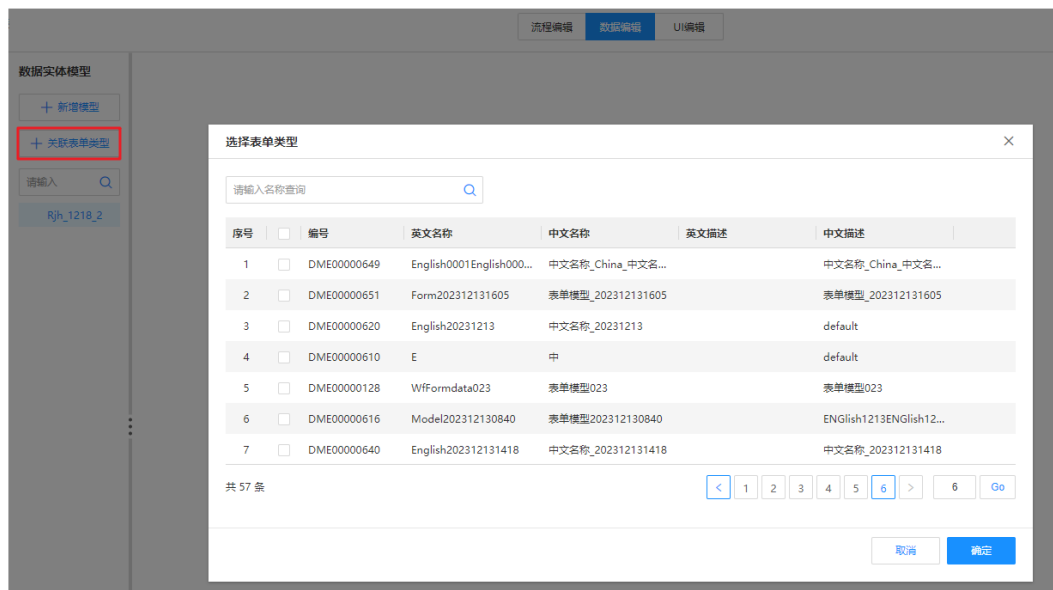
---结束

关联申请单

将建立好的模型关联到流程申请单中，这样在编辑流程页面时，就可以直接选用关联的模型了。

- 步骤1** 在左侧导航栏中，选择“流程管理 > 流程申请单”，进入申请单列表界面。
- 步骤2** 选择已创建的流程申请单，单击“编辑”，进入编辑页面。
- 步骤3** 选择“数据编辑 > 关联表单类型”，勾选已创建的数据实体模型。

图 5-22 添加表单类型



- 步骤4** 单击“确定”并“保存”。

---结束

5.1.5 步骤四：绘制流程页面

使用UI编辑器编辑流程节点的页面，在页面中添加实际流程所需的表格、表单、附件或设置流程审批责任人等操作。

进入 UI 编辑器

使用UI编辑器，编辑流程节点页面，您可以使用系统中预置的模板、组件进行编辑。


步骤1 在左侧导航栏中，选择“流程管理 > 流程申请单”，进入申请单列表界面。

步骤2 选择已创建的流程申请单，单击“编辑”，进入编辑页面。

步骤3 选择“UI编辑”页签。

----结束

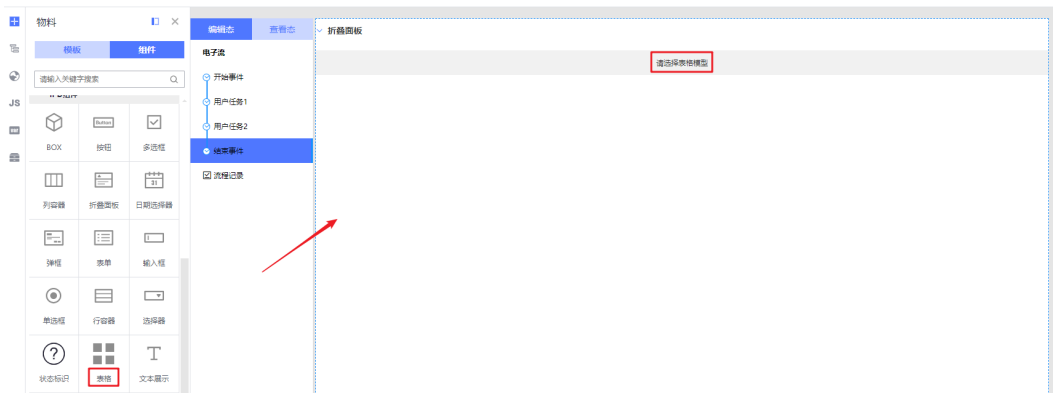
使用表格组件

步骤1 在UI编辑页面，单击左侧插件栏的  图标，选择“组件”页签。

选择表格或表单组件，将其拖拽至中心画布。也可以先拖入一个折叠面板，将表格或表单拖入折叠面板组件中。

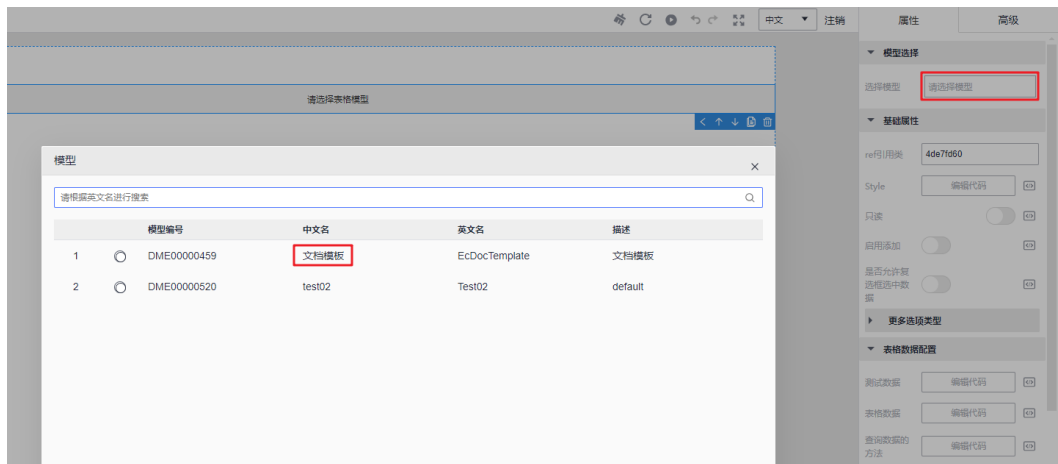
折叠面板组件可以折叠、隐藏高度受限的内容，优化页面结构。

图 5-23 添加表格组件



步骤2 选择画布中的表格组件，在右侧属性面板中，单击“选择模型”，添加**步骤三**中创建的数据实体模型（文档模板）。

图 5-24 添加数据实体模型



步骤3 在属性面板“表格列配置”中，设置表字段是否可编辑、是否必填、控件类型和是否只读等，也可以显示或隐藏字段，如图5-25所示。

若字段“控件类型”中选择“选择框”类型，需要在“选项数据”中填写相关代码，如图5-26所示。

图 5-25 设置表字段

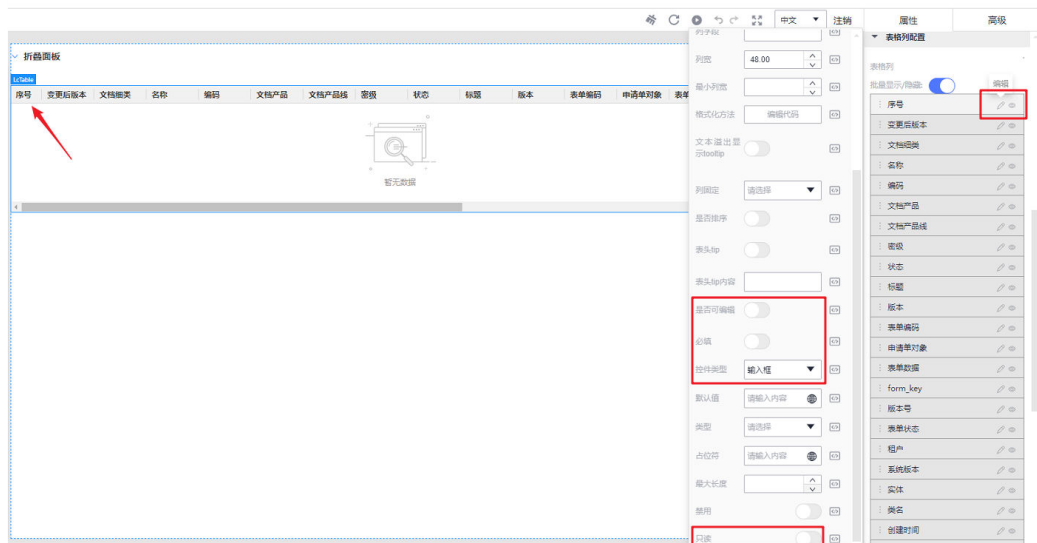
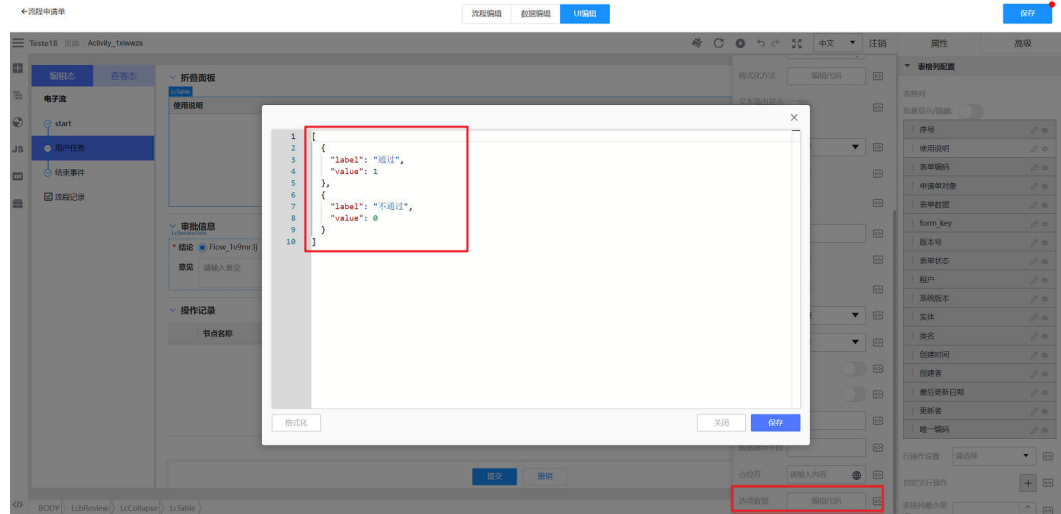


图 5-26 设置控件选项



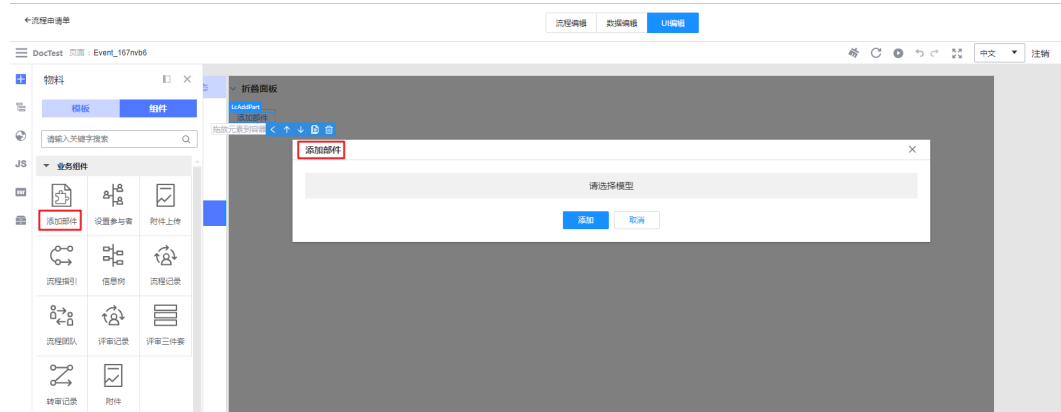
步骤4 单击“保存”。

----结束

使用部件组件

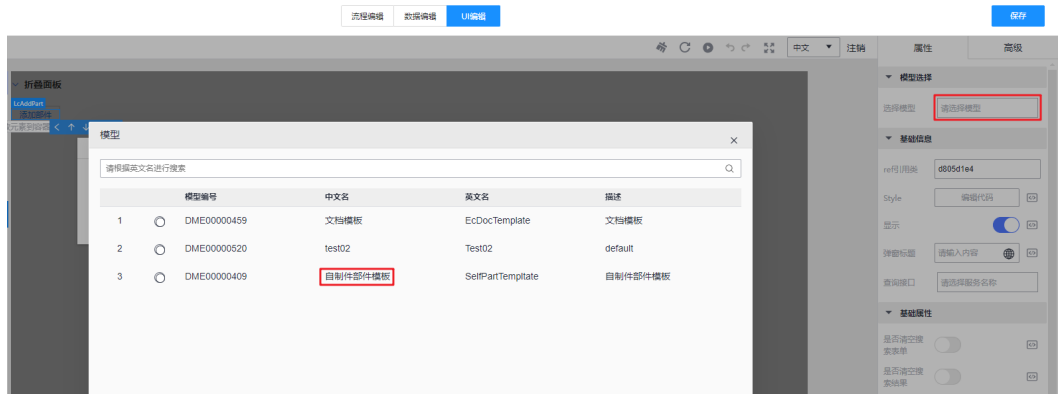
步骤1 选择部件组件，将其拖拽至中心画布。

图 5-27 添加部件



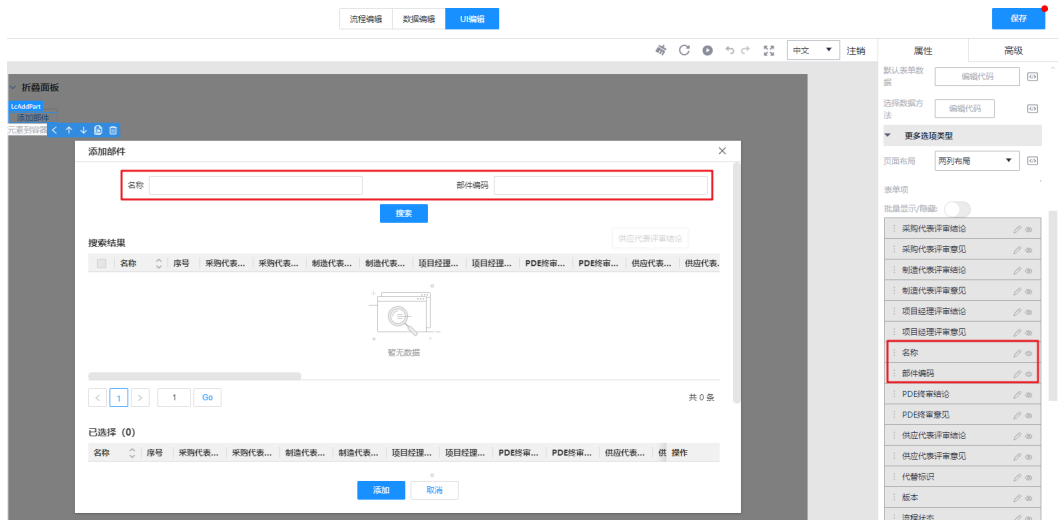
步骤2 选择画布中的部件组件，在右侧属性面板中，单击“选择模型”，添加步骤三中创建的数据实体模型（自制件部件模板）。

图 5-28 添加数据实体模型



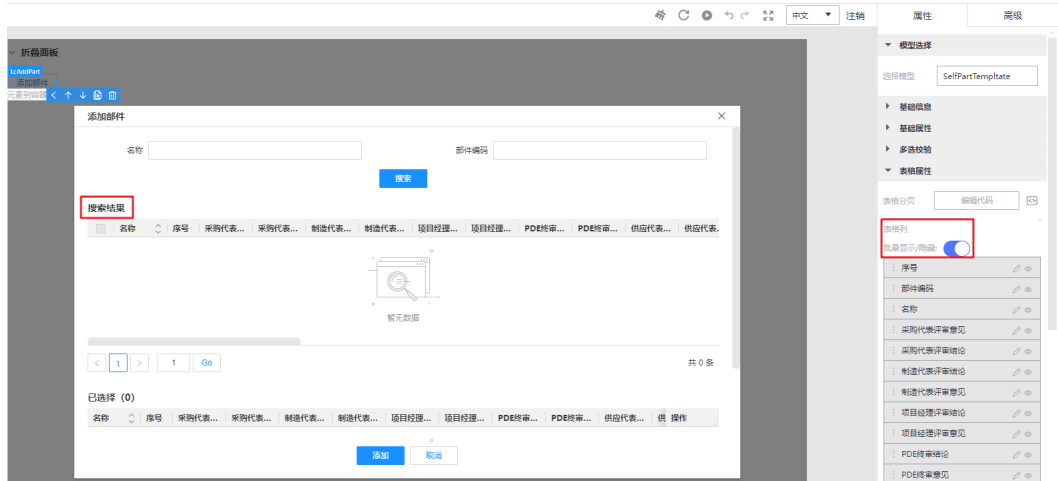
步骤3 在属性面板“基础属性 > 更多选项类型 > 表单项”中选择需要保留查询的字段。

图 5-29 查询字段



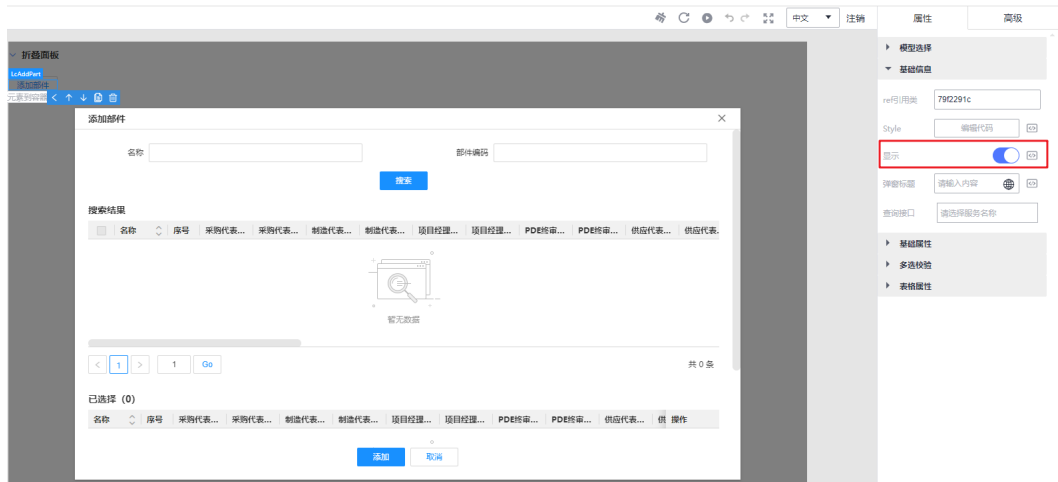
步骤4 在属性面板“表格属性 > 表格列”中选择搜索结果展示哪些字段。

图 5-30 搜索结果设置



步骤5 在“基础信息 > 显示”中设置显示或隐藏当前“添加部件”窗口。

图 5-31 显示/隐藏弹窗



----结束

在部件中绑定服务定义

选择画布中的部件组件，打开右侧的属性面板。

步骤1 绑定服务定义。

1. 选择“基础信息 > 查询接口”，在“查询接口”中绑定已配置好的“组合自制件部件查询”服务定义，用于按条件查询部件，如图5-32所示。
2. 在自制件查询部件服务定义中，搜索字段为提前设置的字段（可以更改，格式仿照图中的格式来写，即字段 = \${字段}），如图5-33所示。
服务定义的返回参数要和页面表格中的“列字段”一一对应。

图 5-32 绑定服务定义

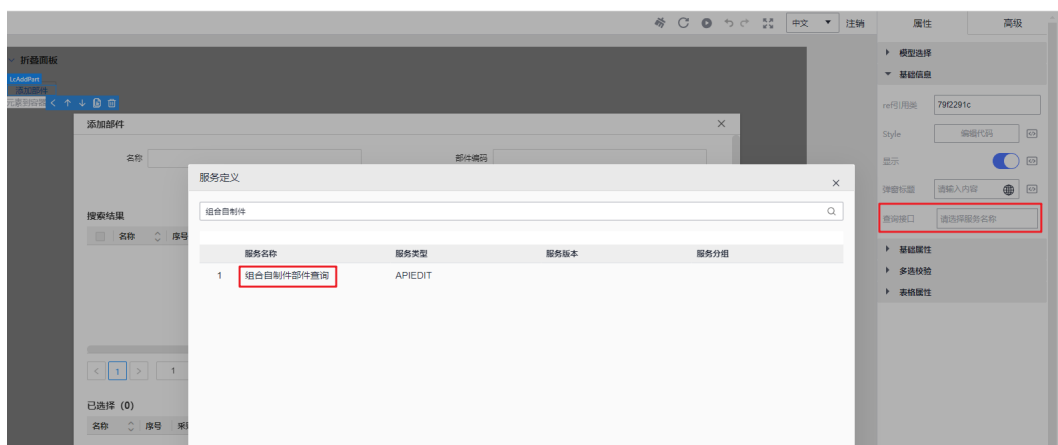
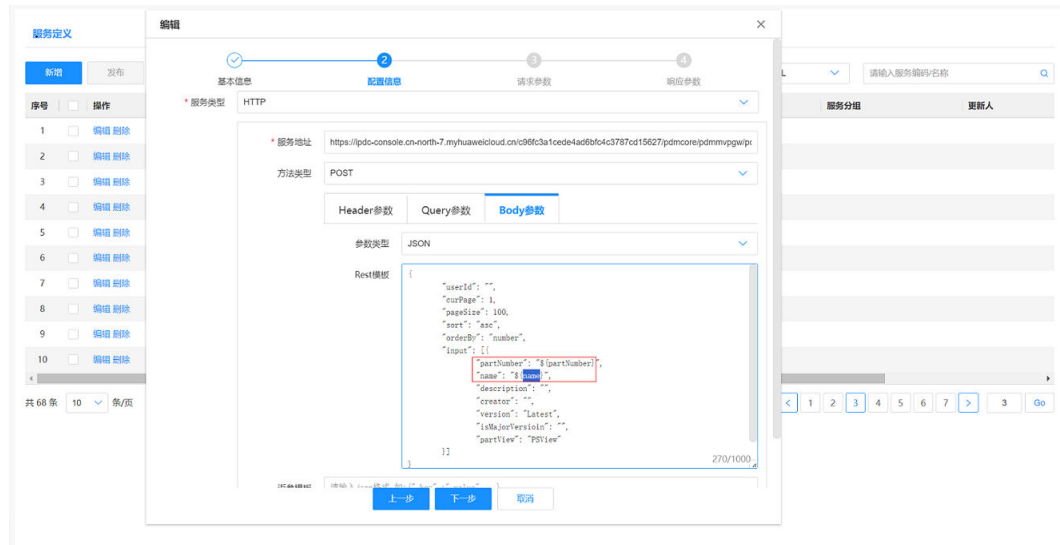


图 5-33 设置查询字段



步骤2 绑定JS函数。

- 选择“基础属性 > 选择数据方法”，“选择数据方法”一栏可以绑定一个用于添加部件的JS函数。

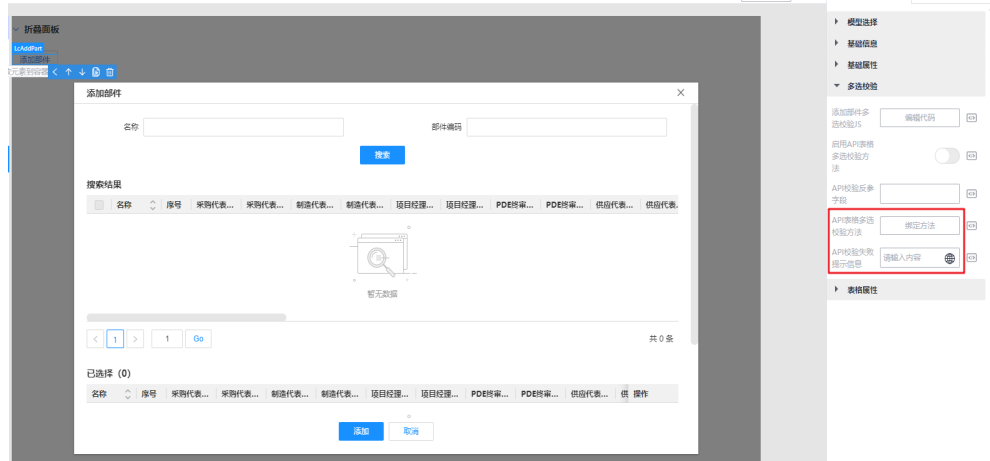
图5-34是校验JS示例，详见“自制件BOM审核发布流程”中的addPartClick()函数。

图 5-34 绑定 JS 函数

```
25 //添加部件逻辑
26 function addPartClick(data) {
27   const set = new Set(this.state.partList.map((item) => item.partNumber));
28   const arr = [].concat(this.state.partList);
29   //判断的是否校验成功标志
30   let result = true;
31   data.some((item) => {
32     //校验规则
33     if (!checkPart(item)) {
34       result = false;
35       return true;
36     }
37     if (!set.has(item.partNumber)) {
38       item.workflowStateCN1 = '正在审阅';
39       arr.push(item);
40     }
41     return false;
42   });
43   //校验不通过则结束
44   if (!result) {
45     return false;
46   }
47   const lineSet = new Set(arr.map((item) => item.productID));
48   const codeSet = new Set(arr.map((item) => item.categoryCode));
49   const productLineSet = new Set(arr.map((item) => item.productLineID));
50   if (lineSet.size > 1 || codeSet.size > 1 || productLineSet.size > 1) {
51     //弹窗提示
52     this.state.errorInfo = '请添加同意归属统一产品线/产品/分类的部件!';
53     showError();
54     return false;
55   } else {
56     console.log(arr);
57     if (arr.length > 0) {
58       this.state.productInfo.category = arr[0].categoryCode;
59       this.state.productInfo.productID = arr[0].productID;
60       this.state.productInfo.productLineChName = arr[0].productLineChName;
61       this.state.productInfo.productChName = arr[0].productChName;
62       this.state.productInfo.productLineID = arr[0].productLineID;
63     }
64   }
65 }
```

- 选择“多选校验”中的“API表格多选校验方法”或“API校验失败提示信息”，可以绑定一个JS校验函数，用于校验。如图5-35所示。

图 5-35 绑定校验 JS 函数



在“自制件BOM审核发布流程”中，绑定的是图5-36中checkPart()函数，用于校验选中的部件是否满足要求。

校验失败提示信息绑定一个变量，表示的当校验失败时提示对应信息。

图 5-36 JS 校验函数示例

```
69 //校验选中部件
70 function checkPart(part) {
71   const { currentAppUser } = this.utils.lcUtils().useBullitinAppState();
72   //校验是否自制件
73   if (part.source !== 'make') {
74     this.state.errorInfo = '请添加自制件!';
75     showError();
76     return false;
77   }
78   //校验是否A版本
79   if (part.majorVersion !== 'A') {
80     this.state.errorInfo = '请添加大版本是A版本的部件!';
81     showError();
82     return false;
83   }
84   //校验是否已检入
85   if (part.checkInOrCheckOut !== 'c/i') {
86     this.state.errorInfo = '请添加已检入的部件!';
87     showError();
88     return false;
89   }
90   //校验是否正在工作中的
91   if (part.workflowState !== 'INWORK') {
92     this.state.errorInfo = '请添加正在工作中的部件!';
93     showError();
94     return false;
95   }
96   //校验是否创建者
97   if (part.creator !== currentAppUser.value.userAccount) {
98     this.state.errorInfo = '请添加创建者是本人的部件!';
99     showError();
100    return false;
101  }
102  return true;
103 }
```

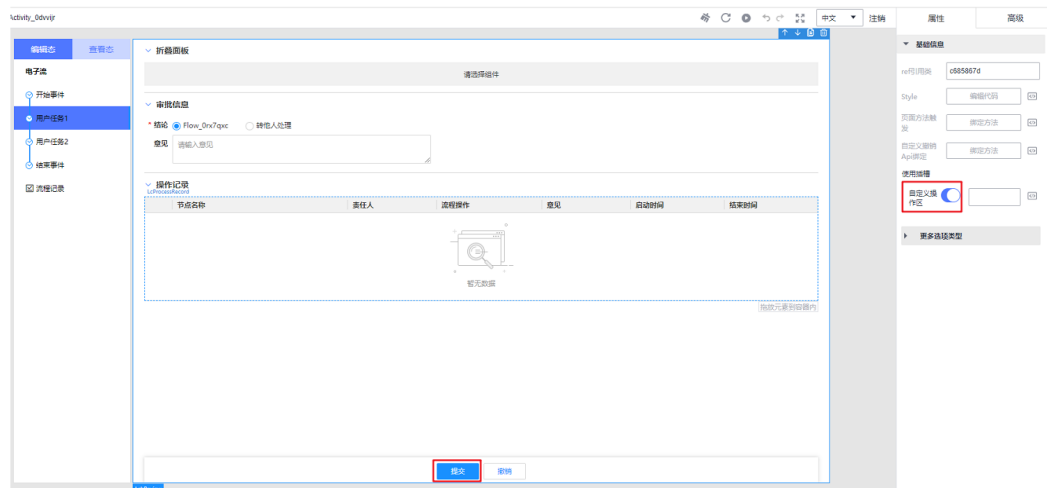
----结束

提交按钮绑定 JS 代码

如果对提交按钮不做任何修改，默认是正常的提交流程。当对其绑定JS代码后，可以实现对页面的数据进行一定的校验，提交流程，然后再做一些后置处理。

步骤1 选择画布中的提交按钮，打开右侧的属性面板，开启“自定义操作区”按钮。

图 5-37 提交按钮



步骤2 选择左侧插件栏的JS，进入JS页面，单击“展示使用帮助”。如图5-38所示。

模板中预置了常用的JS脚本模板，可以直接复制编辑使用。

- 提交节点可以粘贴提交模版事件，审批节点粘贴审批模版事件。before()中添加校验逻辑，例如在“自制件BOM审核发布流程”中添加校验是否所有部件的评审意见都为通过。
- 在after()中可以绑定一个服务定义用于后置处理，调用definitionRequest()方法。在“自制件BOM审核发布流程”该服务定义用于修改部件状态。如图5-39所示。

图 5-38 JS 使用帮助



图 5-39 示例代码

```
// 自定义后置事件
async function after(reloadUrl) {
  const { definitionRequest } = this.utils.lcCommonJS();
  // definitionRequest(【服务定义编号】，【请求方法】，【入参】)
  // 【服务定义编号】：在流程定义的服务定义里面自行查看
  // 【请求方法】：在创建服务定义时的接口请求方法(get/post/put/delete)
  // 【入参】：请求的传参，如果是get会自行放在query上，其他类型会放在body上
  const res = await definitionRequest('1743118575232638976', 'post',
  { 'WorkflowState': this.state.WorkflowState });
  console.log(JSON.stringify(res));
  return reloadUrl;
}
```

步骤3 通过单击按钮展示弹框（“自制件BOM审核发布流程”中用于展示“查看不满足发布条件子项信息”按钮）。

- 将表单中的字段输入类型改成“按钮”，或者直接拖入一个按钮组件，将按钮名称改为自己需要的。如图5-40所示。
- 拖入一个弹窗组件，在展示弹窗处绑定一个变量，该变量为true可以展示弹窗，为false可以隐藏弹窗。该变量值可由JS代码控制。如图5-41所示。

图 5-40 添加按钮

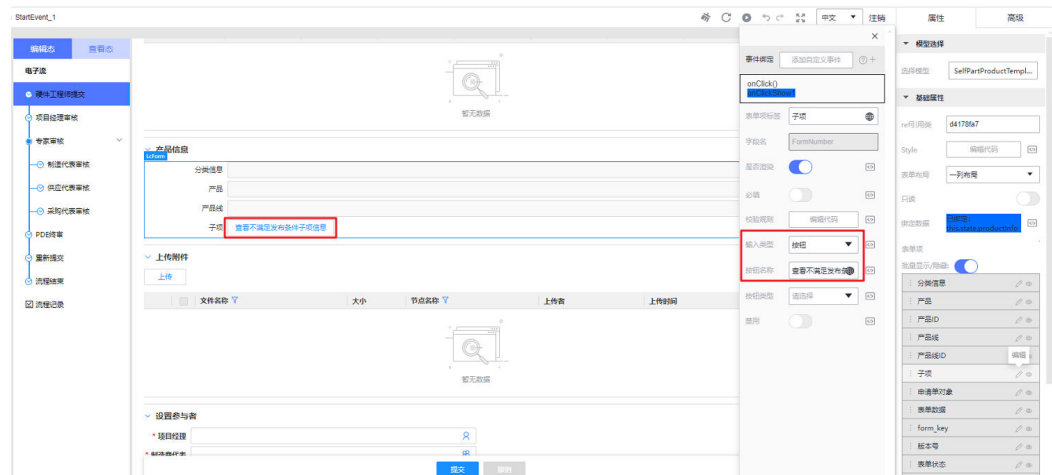
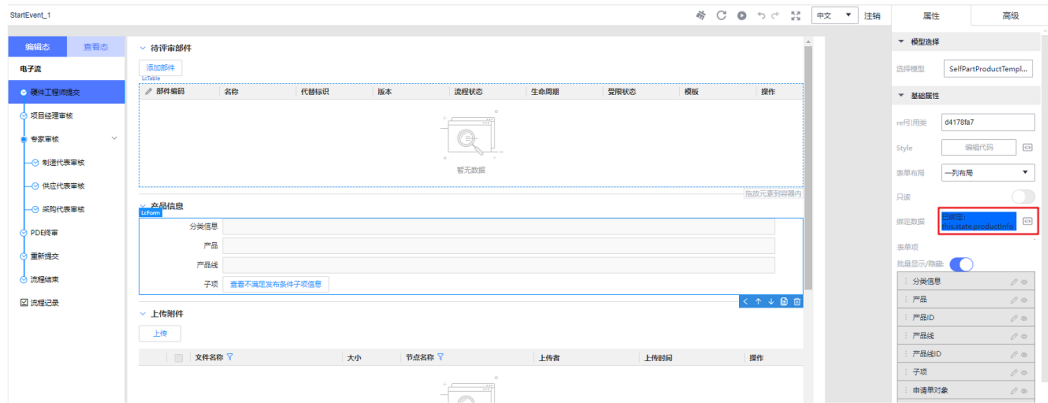


图 5-41 添加变量



----结束

将变量配置为超链接格式

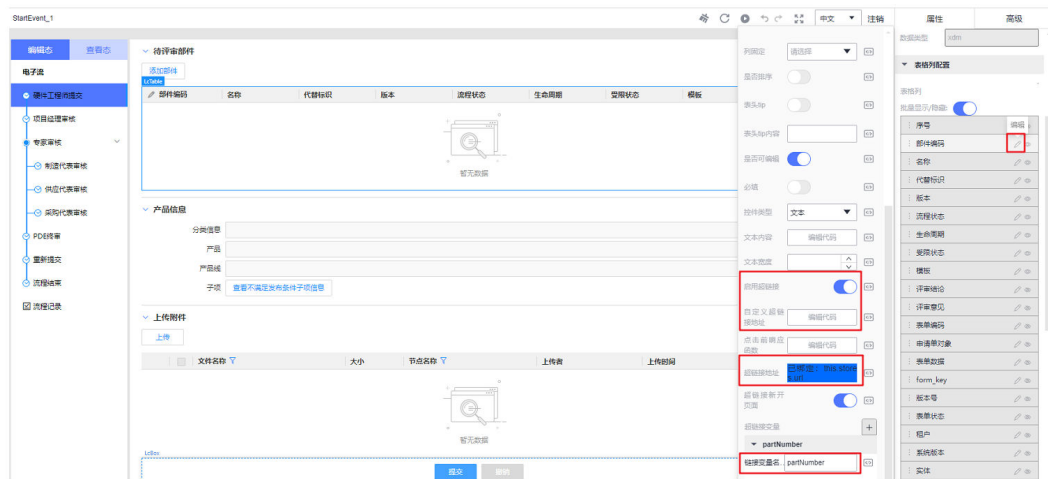
在“自制件BOM审核发布流程”中将部件编码制成超链接形式，可以跳转到对应的部件详情页。

图 5-42 超链接展示

序号	部件编码	名称	流程状态	生命周期状态	当前状态	EC类型	变更前版本	变更后版本	视图	EC在结构表	操作
1	03070128-017	配置件-OptiX Metro8100-SSC6LWMD01M01-多...	正在工作	EOP	Null	部件升级	B.3	C.1	PSView		查看详情
2	03070128-011	配置件-OptiX Metro8100-SSC6LWMD01M01-多...	正在工作	EOP	Null	部件升级	B.3	C.1	PSView		查看详情

打开启用超链接开关，超链接地址绑定一个变量，打开超链接新开页面开关，如有需要可以配置一个链接变量名。

图 5-43 添加超链接



其中超链接地址绑定的变量的值为：

[https://“实际域名”/pdmcore/pdmc/core/ui/part/info/?partNumber={partNumber}&partView=PSView&partVersion=A](https://实际域名/pdmc/core/pdmc/core/ui/part/info/?partNumber={partNumber}&partView=PSView&partVersion=A)

其中{}内容为链接变量名，表示把该部件编码传入url中。

5.1.6 步骤五：发布电子流

流程发布

步骤1 流程编辑好后，选择“流程管理 > 流程申请单”，进入流程申请单列表页面。

步骤2 勾选需要发布的申请单，单击“发布”。

发布后需要由安全管理员进行审核。

图 5-44 申请发布



----结束

流程应用

步骤1 打开浏览器访问如下地址。

`http://{域名}/pdmcore/ipdcworkflowservice/console/ui/definition_micro/{example1}/{example2}/index.html`

其中，**example1**为流程的英文名称，**example2**为版本号。

步骤2 按照定义好的流程顺序，填写流程信息及审批人。

步骤3 各节点审批通过后，流程结束。

📖 说明

流程申请人可以在过程中关闭流程，或撤回重新填写流程。

----结束

5.2 服务定义

操作场景

通过服务定义能够支持对第三方系统，如接口、数据库、消息的通用访问能力。

基本信息

步骤1 进入流程定义，选择“流程管理 > 服务定义”，进入服务定义列表页面。

步骤2 单击“新增”，进入新增页面。

步骤3 填写自定义服务的“基本信息”。

图 5-45 基本信息

新增

1 基本信息

2 配置信息

* 服务名称

服务版本

tag

服务描述 请输入服务描述 0/1000

* 批量执行 是 否

* 是否异步执行 是 否

* 驱动方式 模板 API MQS

下一步 取消

- 服务名称：服务定义的名称。
- 服务版本：服务定义的版本。
- 服务描述：关于服务定义的描述。
- 是否异步执行：设置服务定义是否立即返回，默认否；选择是时，显示查询方式。
- 驱动方式：**API**，对外提供一个接口，通过API方式调用。**模板**，该类型可以在流程模板中直接使用，即通过Camunda本地方法调用。

步骤4 单击“下一步”。

----结束

配置信息

步骤1 选择服务类型。目前支持HTTP和APIEDIT类型，默认为HTTP。

步骤2 输入服务地址，需要调用的外部URL地址。

步骤3 选择方法类型，GET/POST/PUT。

步骤4 根据类型的不同设置参数，包含Header参数、Query参数及Body参数。

图 5-46 参数设置



- 参数名称：目标接口地址需要的参数。
- 参数值：使用 $\${}$ 获取入参配置的参数。

📖 说明

- 参数中变量来自于当前请求上下文。服务定义用于流程模板时，会从流程变量、请求参数中获取；用于UI页面时，会从请求参数中获取。
- 如果需要调用电子流自身服务，需在header中增加 $x\text{-jwt-token}=\${x\text{-jwt-token}}$ 用于服务间认证。
- 请求类型：类型选择POST或PUT时，需要配置Body参数，其中：
 - 参数类型是FORM_DATA时，可以配置多行的参数。
 - 参数类型是JSON时，输入Rest模板。

步骤5 设置返参模板。

为页面设置返回参数格式。

----结束

请求参数（可选）

基本信息中，驱动方式选择API时，需要设置请求参数。设置完成后可以在UI编辑器中直接将参数绑定在API组件中。

步骤1 单击“新增”，进入“设置请求参数”页面。

步骤2 根据界面提示，设置请求参数。

图 5-47 请求参数



步骤3 单击“下一步”。

----结束

响应参数（可选）

基本信息中，驱动方式选择API时，需要设置响应参数。设置完成后可以在UI编辑器中直接将参数绑定在API组件中。

步骤1 单击“新增”，进入“设置响应参数”页面。

步骤2 根据界面提示，设置响应参数。

图 5-48 响应参数



步骤3 单击“保存”。

----结束

管理服务定义

进入流程定义，选择“流程管理 > 服务定义”，进入服务定义列表页面，参照表1管理服务定义。

表 5-4 管理服务定义

功能	操作
编辑	1. 在服务定义列表中，选择需要编辑的服务定义，单击操作列的“编辑”。 2. 编辑完成后，选择“保存”。
删除	1. 在服务定义列表中，选择需要删除的服务定义，单击操作列的“删除”。 2. 在确认弹窗中选择“删除”。
上传/下载	1. 在服务定义列表中，勾选需要下载的服务定义，单击“下载”，下载模板文件，按照模板填写配置信息。 2. 填写完成后，单击“上传”，选择“流程模板分类”并上传模板文件。
另存为	1. 在服务定义列表中，选择需要另存的服务定义，单击操作列的“另存为”。 2. 修改服务名称及描述，单击“确定”。另存为的服务定义为“草稿”状态。
发布	新增、编辑、删除服务定义，需要“发布”后才能够生效。


5.3 流程管理

5.3.1 创建流程申请单

操作场景

创建流程申请，在一个申请单中，定义其流程模板、UI编辑、数据模型等信息。

操作步骤

- 步骤1** 在角色桌面首页，单击左上角的，选择“功能总览 > 流程管理 > 流程模板定义”，进入流程定义系统。
- 步骤2** 在左侧导航中，选择“流程管理 > 流程申请单”，进入流程申请单列表页面。
- 步骤3** 单击“新增”，输入中英文名称及描述。
- 步骤4** 单击“保存并创建流程模板”。

----结束

5.3.2 流程设计

操作场景

使用设计组件，设计电子流的流程图，并为组件添加属性及监听器。

流程设计

流程申请单创建完成后，返回至申请单列表页面，单击操作列的“编辑”，进入流程编辑页面。

步骤1 添加组件。

- 拖拽左侧列表中的组件至工作布。
- 单击画布中的组件，会显示快捷操作按钮，通过快捷按钮，也可以快速的添加任务、网关、事件、注释或连接线等。

如图5-49所示，添加四个组件，开始事件、结束事件和“用户任务”，并使用箭头进行连接。更多组件说明请参考[组件介绍](#)。

图 5-49 创建流程节点

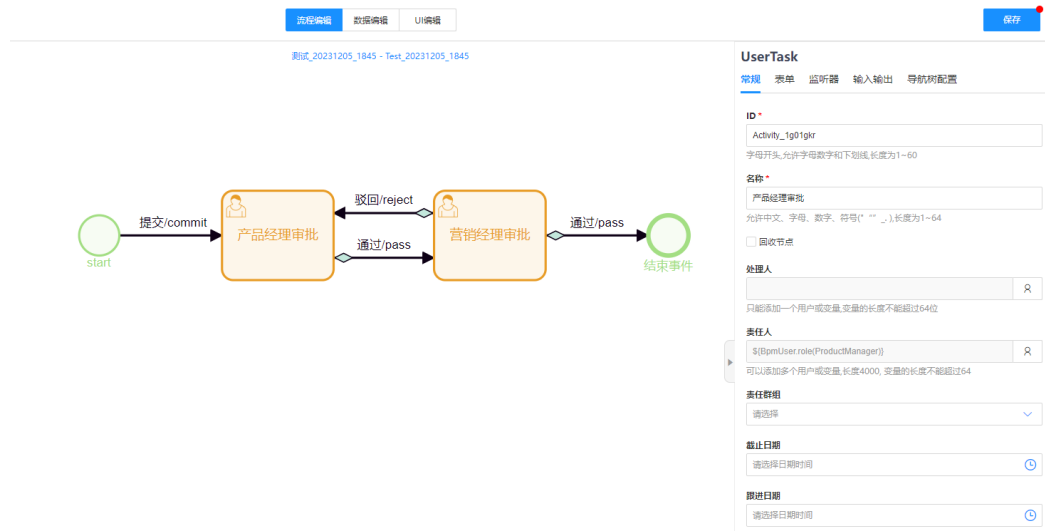


步骤2 为组件添加属性及监听器。

1. 选择“用户任务”组件，在右侧“常规”页签中设置“名称”“责任人”“责任群组”等信息。
2. 选择“连接箭头”组件，在右侧“常规”页签中设置“中文名称”“英文名称”，“通过条件”选择“表达式”。
3. 常规属性设置完成后，选择[监听器](#)页签，根据实际情况，在“监听器列表”中，为组件添加相应的监听动作。

如“超时自动转审”“超时自动审批”等。

图 5-50 属性设置



步骤3 设置完成后，单击右上角的“保存”。







选择“文件”可将当前界面保存为BPMN文件或SVG图。












----结束

组件介绍

系统预置多种工具及组件，详情如下表。

表 5-5 组件介绍

图标	名称	功能
	抓手工具	拖动画布。
	套索工具	选择多个图形，也可以使用快捷键Shift选取。
	创建/删除空间工具	创建/删除空间。
	开始事件	作为流程的开始。
	中间事件	作为流程组件的中间事件，是一个暂停事件，将流程事务提交，并启动一个新事务执行后续的流程。最常用的是作为消息捕获或定时任务。
	结束事件	代表整个流程的结束。
	排他网关	作为流程组件的路由网关，用于区分分支路由，出口走其中一条分支。

图标	名称	功能
	并行网关	作为流程组件的路由网关，并行网关允许将流程拆分为多个分支，也可以将多个分支汇集到一起。
	包容网关	包容网关是排他网关和并行网关的综合体。包容网关与并行网关类似都是配对使用。 与并行网关的区别是，并行网关出口分支都执行，包容网关是出口分支可以配置条件，执行或不执行。
	空白任务	空白任务节点作用与手动任务节点相同。流程引擎处理该节点时将简单通过，自动继续流程。
	用户任务	作为流程组件的用户处理任务，用户可以进行通过、驳回、转审等操作。
	服务任务	作为流程组件的服务任务节点，调用API服务。
	业务规则任务	作为流程组件的业务规则任务，执行一段表达式或java程序。
	发送任务	作为流程组件的发送消息组件。
	接收任务	作为流程组件的监听事件，等待一个特定的消息（用户配置的消息）到达才往下执行。
	手动任务	用于建模自定义任务内容的任务节点，流程引擎处理该节点时将简单通过，自动继续流程，常用于流程调试。
	内嵌子流程	内嵌子流程，它是一个可以包含其他活动、分支、事件等的活动。它表现为将一个流程（子流程）定义在另一个流程（父流程）的内部，子流程作为父流程的一部分。
	外接子流程	作为流程组件的内嵌流程，嵌入其他bpmn流程。
	导航分组	作为流程组件的分组节点，用户导航树自动生成时作为父节点展示。

快捷键介绍

在编排组件的过程中，还可以通过以下快捷键进行快捷操作。

表 5-6 快捷键命令

操作	快捷键	操作	快捷键
搜索元素	Ctrl + F	复制	Ctrl + C
全选	Ctrl + A	粘贴	Ctrl + V
撤销	Ctrl + Z	删除	Backspace / Delete / Del

操作	快捷键	操作	快捷键
恢复	Ctrl + Y	上下滚动	鼠标滚轮
手型工具	H	移动	←↑→↓
套索工具	L	左右滚动	Shift + 鼠标滚轮
空间工具	S	缩放	Ctrl + (鼠标滚轮 / +/-)
连接工具	C	恢复缩放	Ctrl + 0
编辑名称	E	选取	Shift + (鼠标左键 / 鼠标拖动)

监听器介绍

在流程编辑页面，可以为组件设置属性。其中，用户任务组件可选择“执行监听器”或“任务监听器”，非用户组件只可选择“执行监听器”。

步骤1 选择画布中的组件，在右侧属性设置中选择“监听器”。

步骤2 选择“执行监听器”或“任务监听器”，然后在列表中选择具体的监听内容。

图 5-51 监听器



步骤3 通过单击“表达式”“脚本”“java类”，对监听器添加执行内容。

---结束


5.3.3 导航树配置

操作场景

保存流程图时自动生成导航树，可对导航节点可进行状态设置，如节点点击时的状态、已审批节点是否重置等。

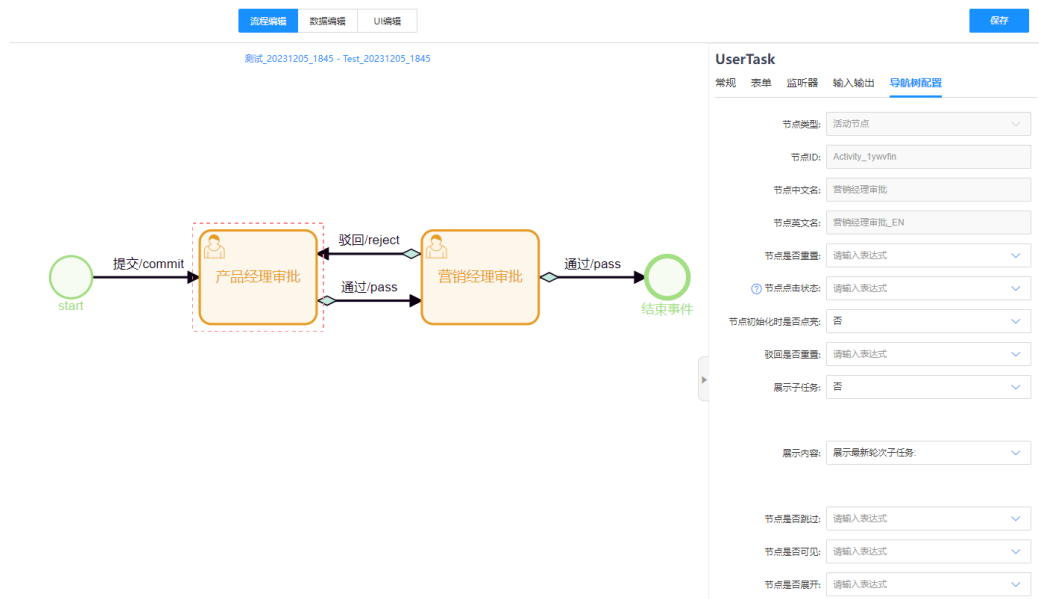
操作步骤

步骤1 进入流程编辑页面。

步骤2 选择左侧的 ，进入导航树配置页面，或选择组件，在右侧的页签中选择“导航树配置”。

步骤3 选中“事件”或“任务”设置节点状态。

图 5-52 导航树配置



- **节点是否重置**：从其他节点进入当前节点时，是否重置数据。
- **节点点击状态**：默认，未开始不可点击；可点击，永远可点击；不可点击，永不点击；表达式，根据条件满足可点击。
- **节点初始化时是否点亮**：若选择是，则节点亮起状态时为可选。
- **驳回是否重置**：可选是或否，若选择是，驳回时已完成的审批节点将重置，需再次审批。
- **展示子任务**：可选是或否，若选择是，在节点导航中可见子任务流程。
- **展示内容**：可选“展示新伦次子任务”或“展示所有子任务”。
- **节点是否跳过**：选择是后将跳过此节点。
- **节点是否可见**：设置节点可见状态，可选择是、否或自定义表达式。
- **节点是否展开**：设置节点展开状态，可选择是、否或自定义表达式。

步骤4 设置完成后，单击右上角的“保存”。

---结束

5.3.4 发布审批

流程发布

步骤1 流程编辑好后，选择“流程管理 > 流程申请单”，进入流程申请单列表页面。

步骤2 勾选需要发布的申请单，单击“发布”。

发布后需要由安全管理员进行审核。

图 5-53 申请发布



----结束

5.4 数据模型

5.4.1 创建数据实体模型

操作场景

创建一个自定义的表单模型，通过属性库添加属性的扩展字段。

创建模型

流程编辑完成后，选择“数据编辑”页签，进入数据编辑页面。

步骤1 单击“添加模型”，进入新增模型页面。

图 5-54 添加模型



步骤2 输入模型的中英文名称及描述，并选择一个父模型。

- **WfFormdata**: 表单模型，对应页面上一个表单或表格。
- **WfBusinessObject**: 基本业务对象，具有12个系统属性（Name、Description、ID、Creator、CreateTime、Modifier、LastUpdateTime、RdmVersion、RdmExtensionType、KIAGUID、SecurityLevel、Tenant）。

步骤3 单击“确定”。

----结束

添加属性

步骤1 选择左侧已创建的模型，在属性“页签”中，单击“新增”。

图 5-55 添加属性



步骤2 在新增页面，根据需要选择单个或多个属性。



说明

属性数据来源于属性库，如需创建新属性，请前往[属性库](#)创建。

步骤3 单击“确定”并“保存”。

----结束

5.4.2 申请单类型

操作场景

流程引擎中申请单类型、表单类型、评审对象类型由建模生成，添加关联后可在组件中快速添加模型。

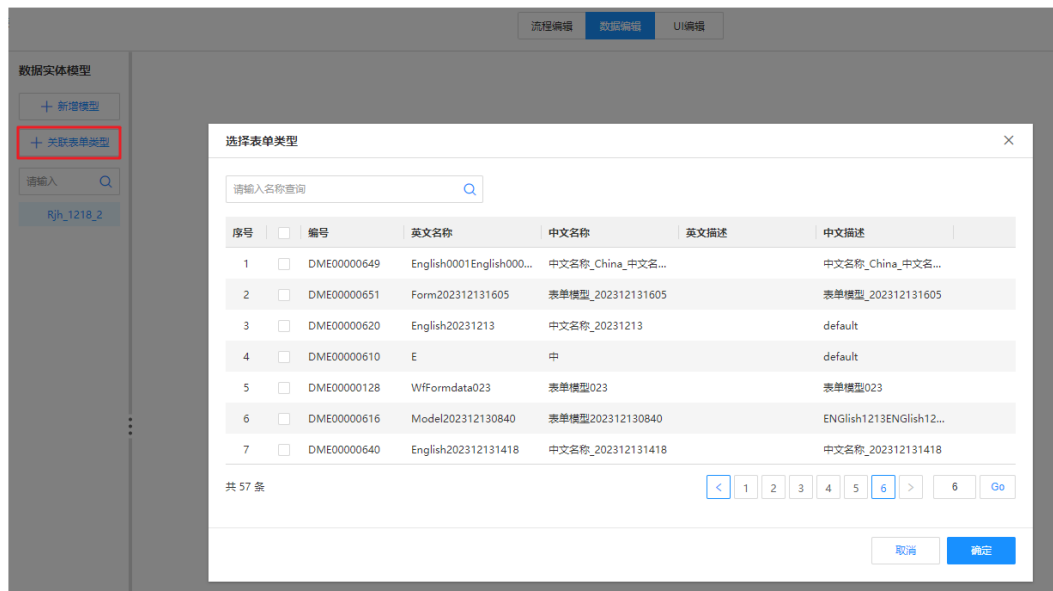
操作步骤

流程编辑完成后，选择“数据编辑”页签，进入数据编辑页面。

步骤1 选择“关联表单类型”。

步骤2 勾选需要的数据实体模型。

图 5-56 添加表单类型



步骤3 单击“确定”并“保存”。

----结束

5.4.3 属性库

操作场景

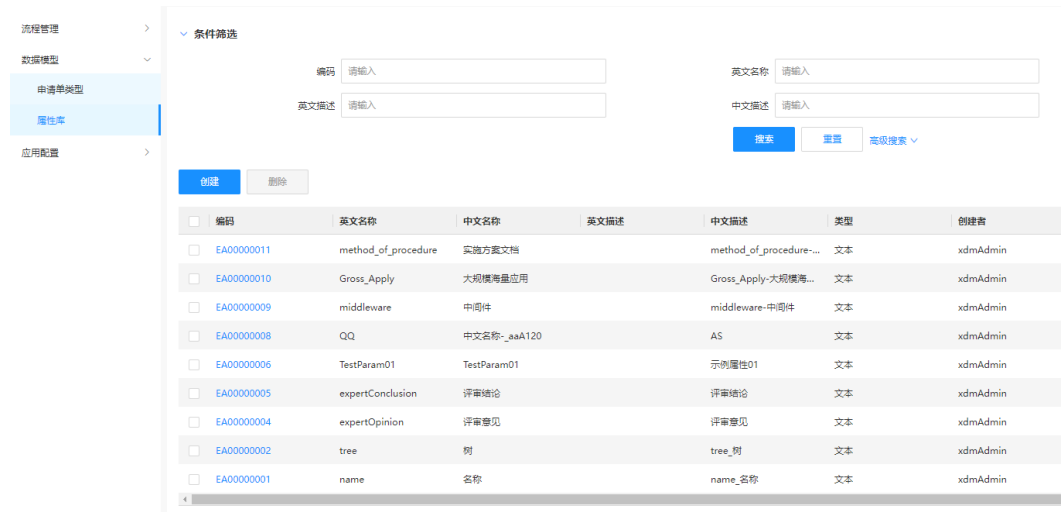
您可通过从属性库中添加、移除属性，实现对象内扩展属性的管理。

创建属性

步骤1 在左侧导航栏中，选择“数据模型 > 属性库”，进入属性库页面。

步骤2 单击“创建”。

图 5-57 创建属性



步骤3 根据页面提示，配置如下信息。

图 5-58 基本信息

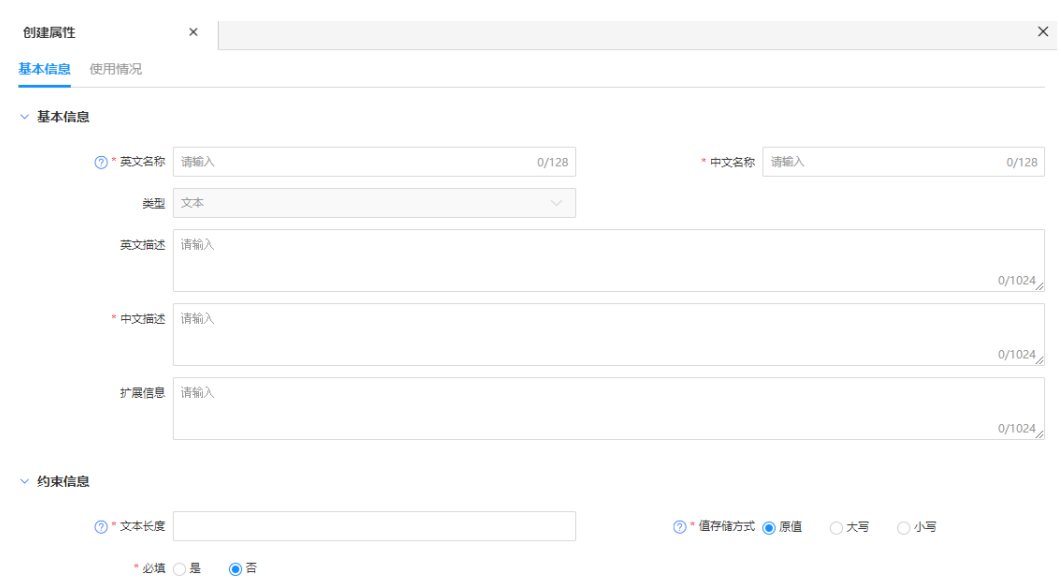


表 5-7 属性基本信息

类型	参数	参数说明
基本信息	英文名称	属性的英文名称。
	中文名称	属性的中文名称。
	类型	类型固定为文本。
	英文描述	属性的英文描述。
	中文描述	属性的中文描述。
	扩展信息	与属性相关的描述。

类型	参数	参数说明
约束信息	文本长度	超过4000记录为长文本，则数据输入长度不受此限制，如果属性已经基线，则修改此值后，不会改变文本类型。
	值存储方式	<ul style="list-style-type: none"> 原值：实例值按照实际输入的字符串存储。 大写：将实例值中小写字符串转换为大写字符串存储。 小写：将实例值中大写字符串转换为小写字符串存储。

步骤4 单击“保存”。

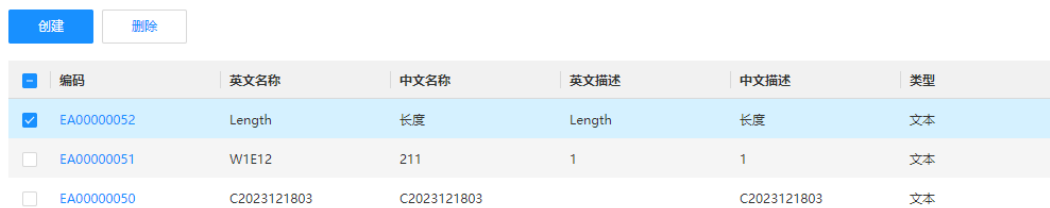
----结束

删除属性

步骤1 在左侧导航栏中，选择“数据模型 > 属性库”，进入属性库页面。

步骤2 勾选需要删除的属性，单击“删除”。

图 5-59 删除属性



说明

若属性被引用，不能删除该属性。如需删除，需要先断开该引用。

步骤3 在弹出的提示框中，单击“确定”。

----结束

编辑属性

步骤1 在左侧导航栏中，选择“数据模型 > 属性库”，进入属性库页面。


步骤2 找到需要编辑的属性，单击操作列的。

图 5-60 编辑属性



步骤3 在展开的属性详情页面，根据实际业务需求编辑属性，单击“保存”。

----结束

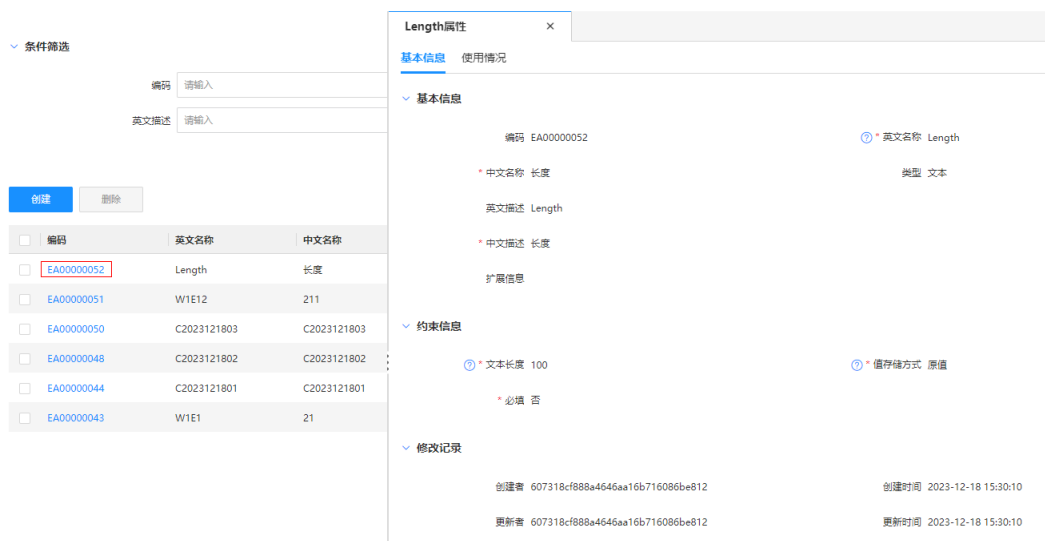
查询属性

步骤1 在左侧导航栏中，选择“数据模型 > 属性库”，进入属性库页面。

步骤2 根据实际需求，设置筛选条件（编码、英文名称、中文名称、类型、创建者等），单击“搜索”。

步骤3 单击属性编码，查看属性的“基本信息”和“使用情况”。

图 5-61 基本信息



----结束


5.4.4 数据实体

5.4.4.1 创建数据实体

操作场景

创建一个数据实体模型，管理所有模型的数据实体实例。

设置基本信息

步骤1 在角色桌面首页，单击左上角的 ，选择“功能总览 > 流程管理 > 流程模板定义”，进入流程定义系统。

步骤2 在左侧导航栏中，选择“数据模型 > 数据实体”，进入数据实体页面。

步骤3 单击“创建”，根据页面提示，配置如下信息。

图 5-62 创建数据实体

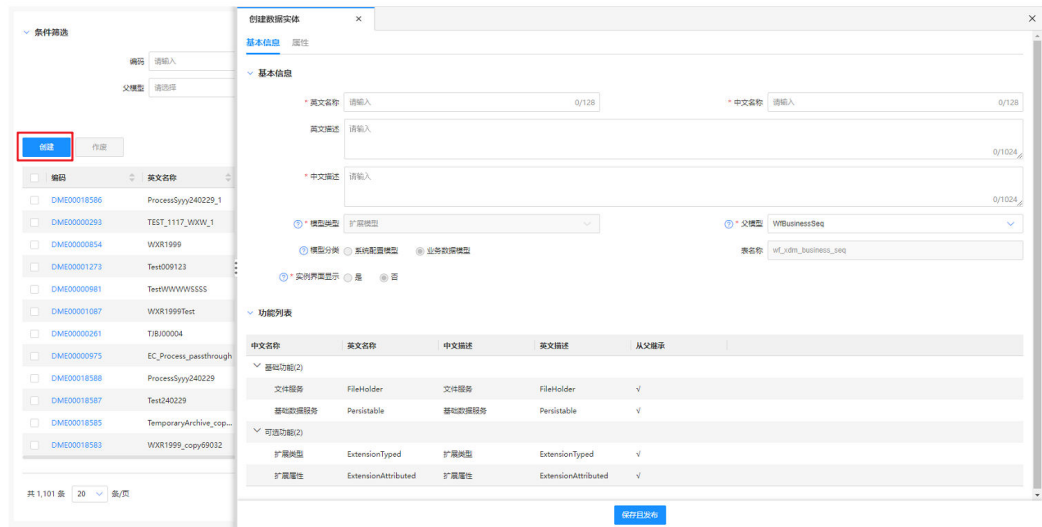


表 5-8 参数说明

参数	说明
英文名称	数据实体的英文名称。 需以大写字母开头，只能包含字母和数字，长度不能超过128个字符。
中文名称	数据实体的中文名称。 只能包含中文、英文字母、数字和“_”，长度不能超过128个字符。
英文描述	数据实体的英文描述。 不能包含中文字符，长度不能超过1024个字符。
中文描述	数据实体的中文描述。 支持各种字符，长度不能超过1024个字符。
模型类型	数据实体的模型类型，只支持“扩展模型”。
父模型	<ul style="list-style-type: none"> WfBusinessSeq: 申请单对象 WfFormdata: 表单对象 WfBusinessObject: 评审对象
模型分类	提供系统配置模型和业务数据模型，当前只支持选择“业务数据模型”，且不可修改。
表名称	数据实体存储至数据库的表名称，默认格式为“应用名称_数据实体英文名称”，不可修改。
实例界面显示	用于标识“数据实例”界面展示/过滤此对象。 与“父模型”的“实例界面显示”一致，且不可修改。如需修改，需在“父模型”所选择的实体中修改。

步骤4 单击“保存且发布”。

----结束

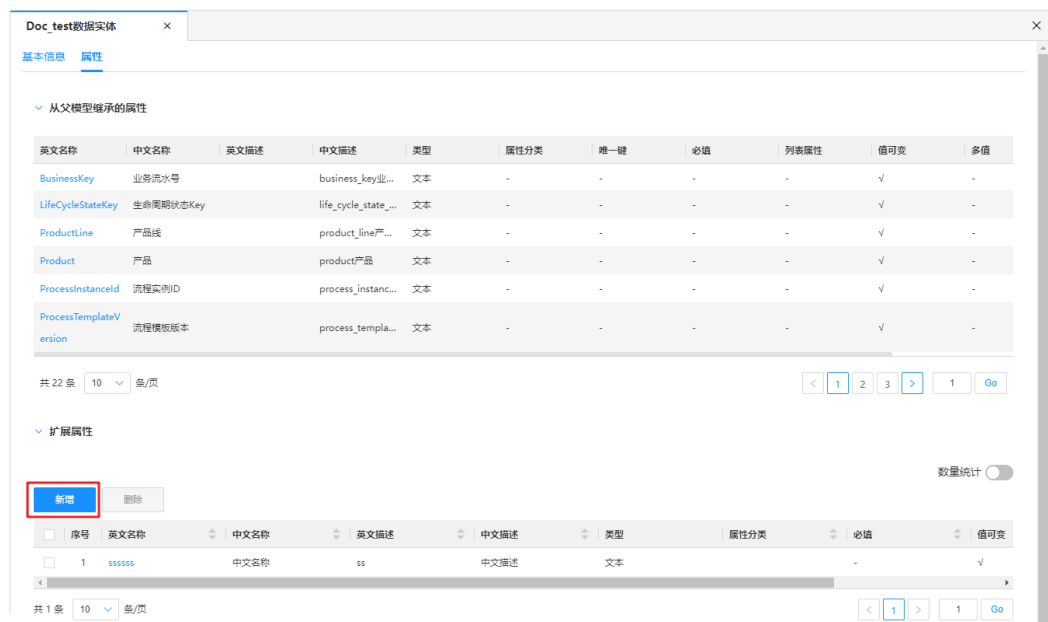
添加扩展属性

完成创建数据实体后，如果该数据实体的父模型具有“扩展属性”功能，可通过如下操作添加扩展属性。

步骤1 选择“属性”页签，进入属性页面。

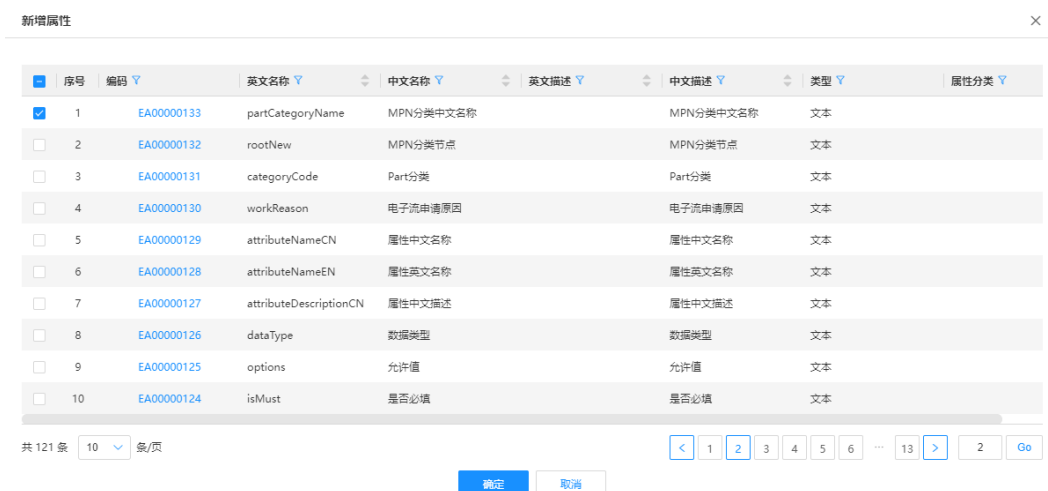
步骤2 在“扩展属性”栏中，单击“添加”。

图 5-63 添加扩展属性



步骤3 在弹出的窗口中，勾选需要添加的属性，单击“确定”。

图 5-64 新增属性



----结束

5.4.4.2 作废数据实体

操作场景

当“模型类型”为“扩展模型”的数据实体不想使用时，您可以作废该数据实体。作废的数据实体将无法使用，请谨慎操作！

操作步骤

步骤1 在左侧导航栏中，选择“数据模型 > 数据实体”，进入数据实体页面。

步骤2 勾选需要作废的数据实体，单击“作废”。

说明

- 如果待作废的数据实体被引用，需要先断开使用或者作废使用对象才能作废。
- 作废后，扩展模型实体引用的文件夹不可删除。

图 5-65 作废数据实体

	<input checked="" type="checkbox"/>	编码	英文名称	中文名称	英文描述	中文描述	父模型	模型类型
1	<input checked="" type="checkbox"/>	DME00001491					Test_WZR_1109	扩展模型
2	<input checked="" type="checkbox"/>	DME00001490	Wzr1114	Wzr1114	Wzr1114	Wzr1114	LifecycleOpera...	扩展模型
3	<input checked="" type="checkbox"/>	DME00001489	Test1114	中文名称在目...		中文名称在目...	LifecycleState	扩展模型
4	<input type="checkbox"/>	DME00001479	TestCyz1107	TestCyz1107	TestCyz1107	TestCyz110711	CyzSyncTestA	扩展模型

步骤3 在弹出的提示框中，单击“确定”。

----结束

5.4.4.3 查询数据实体

操作场景

当您需要查询某些数据实体时，您可以通过查询数据实体功能进行查询。

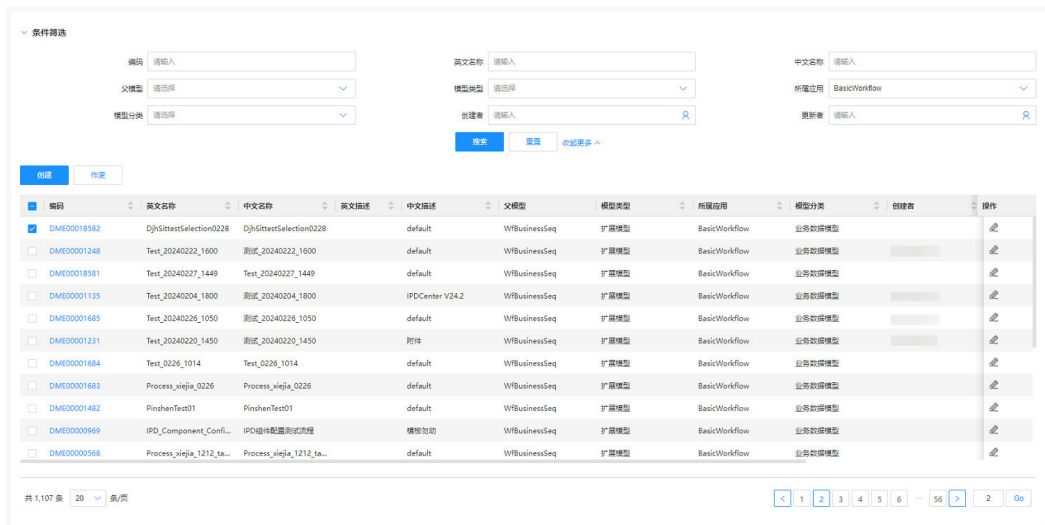
操作步骤

步骤1 在左侧导航栏中，选择“数据模型 > 数据实体”，进入数据实体页面。

步骤2 在“数据实体”页面，根据实际需求，设置筛选条件（编码、英文名称、中文名称、父模型、模型类型、所属应用、模型分类、创建者及更新者）。

步骤3 单击“搜索”。

图 5-66 条件筛选



在数据实体列表中显示的列表字段包含：编码、英文名称、中文名称、英文描述、中文描述、父模型、模型类型、所属应用、创建者及更新者等。

----结束

5.4.4.4 查看数据实体

操作场景

查看数据实体的详细信息，如基本信息、功能列表和修订记录。

操作步骤

- 步骤1** 在左侧导航栏中，选择“数据模型 > 数据实体”，进入数据实体页面。
- 步骤2** 找到需要查看的数据实体，单击该数据实体编码，进入数据实体详情页面。

图 5-67 数据实体详情

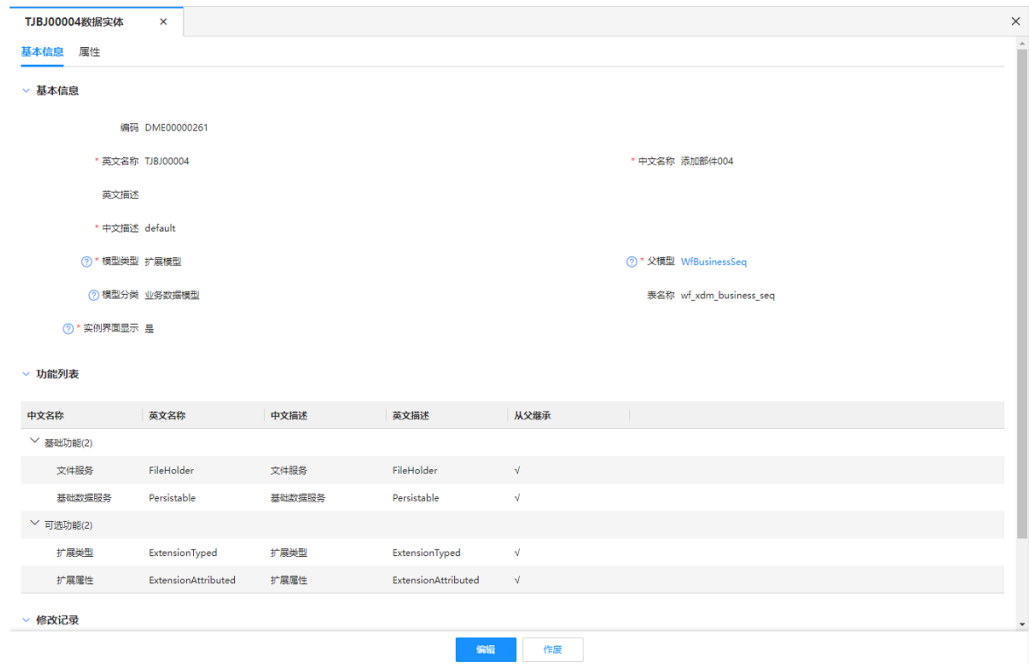


表 5-9 数据实体详情

类型	描述
基本信息	可查看当前数据实体的基本信息，包括：编码、英文名称、中文名称、英文描述、中文描述、父模型、模型类型、模型分类、表名称等。
属性	<p>可查看当前数据实体包含的属性列表，包括“从父模型继承的属性”和“扩展属性”。</p> <p>单击具体属性的英文名称，可以查看属性“基本信息”“约束信息”和“修改记录”。</p>

----结束


5.4.4.5 修改数据实体

操作场景

修改已创建的数据实体信息。

操作步骤

步骤1 在左侧导航栏中，选择“数据模型 > 数据实体”，进入数据实体页面。

步骤2 找到需要编辑的数据实体，单击 。

您也可以进入数据实体详情页面后，单击“编辑”。

步骤3 在展开的页面中，您可以根据实际业务需求进行如下操作。

- 基本信息：仅支持修改中文名称和中英文描述。
- 属性：仅支持新增或删除扩展属性。

步骤4 单击“保存”。

----结束

5.4.5 同步应用

5.4.5.1 创建同步应用

操作场景

应用同步主要用于在同一系统下不同应用中，快速同步应用间的属性和数据模型（申请单模型、表单模型、评审对象模型）。

操作步骤

步骤1 在左侧导航栏中，选择“数据模型 > 应用同步”，进入应用同步页面。

步骤2 单击“创建”。

步骤3 根据页面提示，配置如下信息。

图 5-68 创建应用同步

表 5-10 参数说明

参数	说明
任务名称	离线同步任务的名称，自定义。
同步模块	支持单模块数据同步，可同步的模块包括基础数据和数据模型。 <ul style="list-style-type: none">• 基础数据模块包含“属性库”。• 数据模型模块包含“数据实体”。 为减少关联数据报错，建议同步顺序为“基础数据 > 数据模型”。
同步内容方式	支持选择“全量”和“自定义”。 如果选择了“自定义”，创建离线同步任务后，还需在“同步内容”页签中添加相关待同步的内容。

参数	说明
处理方式	提供如下两种方式，请根据实际需求进行选择。 <ul style="list-style-type: none"> ● 版本覆盖：若目标应用版本正在工作中，则直接覆盖；若目标应用版本已发布，则视为修订更新。 ● 重复则跳过：根据英文名称校验，若目标应用已存在则跳过。

步骤4 单击“创建”。


----结束

5.4.5.2 管理同步应用

启动应用同步

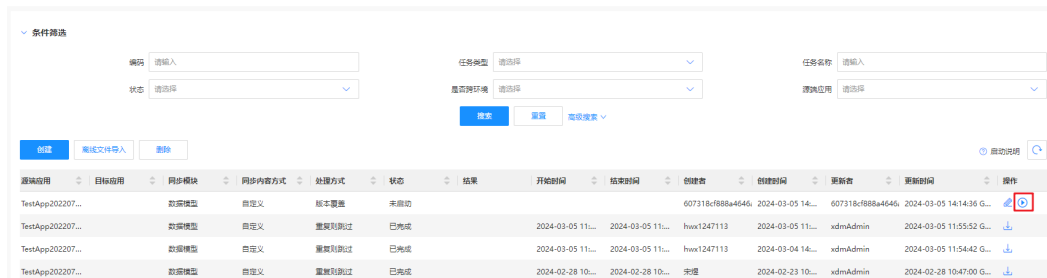
启动已创建的同步任务

步骤1 在左侧导航栏中，选择“数据模型 > 应用同步”，进入应用同步页面。

步骤2 找到需要启动的同步任务，在“操作”栏中单击。

您也可以进入该同步任务的详情页面，单击“启动”。

图 5-69 启动任务




步骤3 在弹出的提示框中，单击“确定”。

----结束

编辑同步任务

您可以编辑“状态”为“待提交”的在线同步任务或离线同步任务。

步骤1 在左侧导航栏中，选择“数据模型 > 应用同步”，进入应用同步页面。

步骤2 找到需要启动的同步任务，在“操作”栏中单击。

步骤3 在展开的页面，根据实际需求，修改基本信息和同步内容。

步骤4 单击“保存”。

----结束

删除同步任务

您可以删除不再使用的同步任务。

步骤1 在左侧导航栏中，选择“数据模型 > 应用同步”，进入应用同步页面。

步骤2 勾选需要删除的同步任务，单击“删除”。

说明

已启动的同步任务不支持删除。

步骤3 在弹出的提示框中，单击“确定”。

----结束

下载离线文件

步骤1 在左侧导航栏中，选择“数据模型 > 应用同步”，进入应用同步页面。


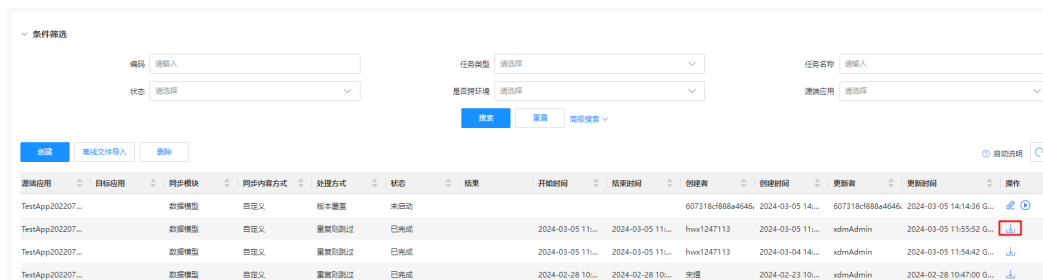
步骤2 找到需要下载离线文件的同步任务，单击，即可将离线文件下载至本地。

图 5-70 下载离线文件



----结束

导入离线文件

步骤1 在页面右上方，切换至目标租户。

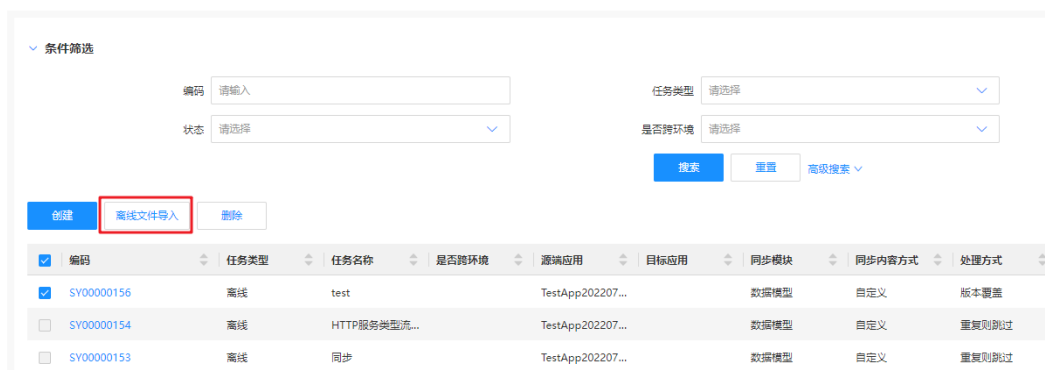
图 5-71 切换租户



步骤2 在左侧导航栏中，选择“数据模型 > 应用同步”，进入应用同步页面。

步骤3 单击“离线文件导入”。

图 5-72 离线文件导入



步骤4 在弹出的窗口中，选取需导入的离线文件，单击“确认导入”。


您可以单击 ，刷新应用同步的状态。待数据同步完成后，您可以在“结果”栏中查看同步结果。

图 5-73 同步结果



您也可以通过单击同步任务的编码，选择“同步内容”页签，查看具体的同步结果。

图 5-74 同步内容



----结束

5.4.5.3 查询同步应用

操作场景

当您需要查询某些应用同步任务时，您可以通过查询应用同步功能进行查询。

操作步骤

- 步骤1** 在左侧导航栏中，选择“数据模型 > 应用同步”，进入应用同步页面。
- 步骤2** 根据实际需求，设置筛选条件（如编码、任务类型、任务名称、状态、是否跨环境、源端应用等）。
- 步骤3** 单击“搜索”。

图 5-75 查询同步应用

----结束

5.4.5.4 查看同步应用

操作场景

查看应用同步的详细信息。

操作步骤

- 步骤1** 在左侧导航栏中，选择“数据模型 > 应用同步”，进入应用同步页面。
- 步骤2** 找到需要查看的应用同步，单击该数应用同步编码，进入应用同步详情页面。
 - **基本信息**：可查看同步任务的基本信息、修改记录。
 - **同步内容**：可查看应用同步具体的同步结果。

图 5-76 应用同步信息

图 5-76 应用同步信息展示了应用同步的基本信息和同步结果。基本信息包括：编码 SY00000143，状态 已完成，源文件 SYDATA_MODEL20240222_211034，任务名称 Test009123_copy01928，源应用 TestApp20220719Mip_dev，同步模块 数据模型，同步内容方式 自定义，目标租户 definition_micro_definition-system，处理方式 版本覆盖。修改记录显示创建者 xdmAdmin，创建时间 2024-02-22 21:11:02 GMT +8。

同步结果部分显示了同步成功的状态，并包含以下数据实体：

编码	英文名称	中文名称	修订号	同步结果	同步完成时间	异常说明
DME00001275	Test009123_copy01928	节copy01596_copy019...	A	{ "code": "SUCCESS", "c...	1708607462956	

----结束

5.5 应用配置

5.5.1 基本配置

操作场景

基本配置中提供管理流程定义所需的基本配置信息，包含运行所需的配置信息、推送消息信息、系统数字指点配置等。

- **基本信息、数据字典**：用于配置流程模板运行的配置项。
- **UI发布设置**：提供UI发布设置所需的配置项，如表2所示。

配置操作

进入流程定义，选择“应用配置 > 基本配置”，进入“基本配置”页面，选择功能页签，参照表1管理配置信息。

图 5-77 基本配置



表 5-11 配置管理

功能	操作
新增	单击“新增”，进入新增配置页面，按照界面提示配置信息，单击“保存”。
编辑	1. 在配置信息列表中，选择需要编辑的配置项，单击操作列的“编辑”。 2. 修改配置信息，编辑完成后，选择“保存”。
删除	1. 在配置信息列表中，选择需要删除的配置项，单击操作列的“删除”。 2. 在确认弹窗中选择“删除”。
上传/下载	1. 单击“下载”，下载模板文件，填写配置信息。 2. 填写完成后，单击“上传”，选择“流程模板分类”并上传模板文件。
发布配置	新增、编辑、删除配置信息，需要“发布”后才能生效。

表 5-12 配置项

参数	参数值	参数说明
LOCAL_API_BASE	/basicWorkflowDemo/console/servicetask/api/	基础配置路径
UI_ENGINE_API_XDM	/basicWorkflowDemo/console/servicetask/api/xdm	XDM请求地址
UI_ENGINE_API_LOCAL	/basicWorkflowDemo/console/servicetask/api/localMethod	本地请求路径
UI_ENGINE_BUILD_TYPE	NodeJs	UI编辑器部署方式，CICD或NodeJs
LOCAL_API_CURRENT	/basicWorkflowDemo/workflow/common/current	用户认证请求地址

5.5.2 团队角色管理

操作场景

您可以在团队角色管理中查看团队角色列表，添加、编辑、删除和发布团队角色。

添加角色

步骤1 进入流程定义，选择“应用配置 > 团队角色管理”，进入团队角色管理列表页面。

步骤2 单击“新增”，进入新增角色界面。

步骤3 输入角色的中英文名称及描述。

步骤4 选择“保存并发布”。

您也可以选择“保存”，然后在团队角色列表页面，单击“发布”。

----结束

删除角色

步骤1 在团队角色列表中，选择需要删除的角色。

步骤2 单击操作列下的“删除”。

步骤3 选择“删除并发布”。

您也可以选择“删除”，此时角色信息会标记删除线，勾选角色，单击“发布”。

单击“撤销”可取消操作。

----结束

编辑角色

步骤1 在团队角色列表中，选择需要编辑的角色。

步骤2 单击操作列下的“编辑”修改角色信息。

步骤3 单击“保存”。

修改后的信息为草稿状态，需要发布后生效。

----结束

发布角色

新增、删除、编辑角需要发布后才能够生效，设置完成后，单击团队角色列表页面中的“发布”。

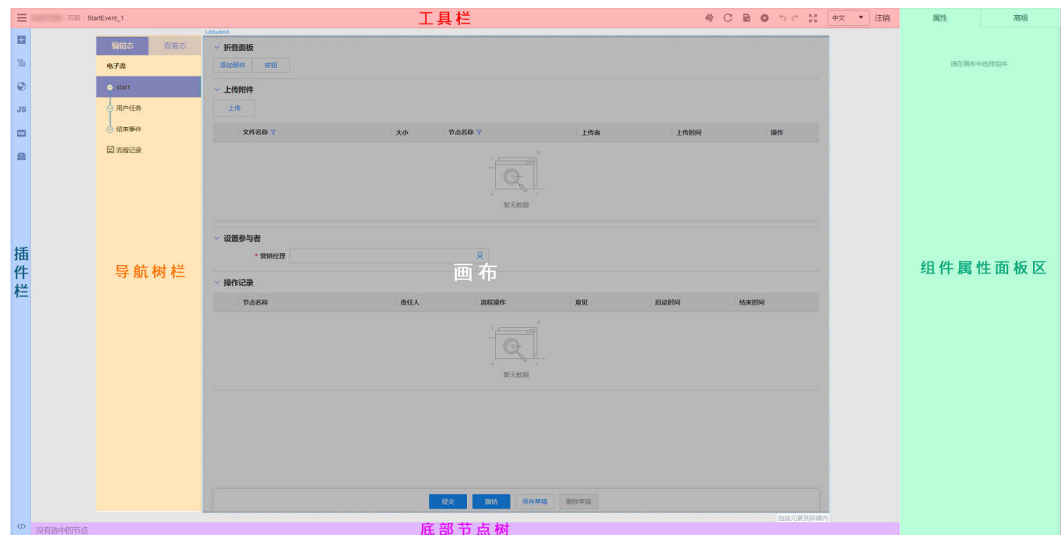
5.6 UI 编辑器使用帮助

5.6.1 设计器界面

设计器页面概览

设计器分为顶部工具栏、左侧插件栏、右侧属性面板区、底部节点树、中间的导航树、中间的导航树及画布，六个主要界面模块。

图 5-78 设计器页面



顶部工具栏




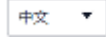
UI编辑器中的顶部工具栏从左至右包括如下功能按钮。

图 5-79 工具栏



表 5-13 工具栏

图标	名称	说明
	菜单按钮	编辑页面的菜单。
	页面名称	展示当前页面名称。
	画布清除工具	清空当前画布页面相关的所有信息（如页面schema、页面状态、页面JS等配置）。
	画布刷新工具	刷新当前页面信息，重新载入页面。
	保存工具	保存当前页面的所有配置信息。
	预览工具	在浏览器中打开新的标签页，预览当前编辑好的页面。

图标	名称	说明
	撤销工具	撤销当前操作并回退至上一步操作时的状态。
	重做工具	恢复上一步操作并回到上一次撤销前的状态。
	全屏工具	将当前页面全屏显示（和网页全屏功能一样）。
	中英文切换工具	单击可切换页面中英文显示。

左侧插件栏

图 5-80 插件栏



UI编辑器中的左侧插件栏从上至下包括如下功能，单击插件之后会向右展开对应插件的设置面板。

表 5-14 插件栏

图标	名称
	物料插件
	大纲树插件
	国际化插件
	页面JS插件
	状态管理插件
	资源管理插件

右侧属性面板

右侧组件设置面板分为“属性”设置和“高级”设置两个板块：

- 属性设置：设置组件的属性，包含固有属性（ref引用类、Style）及不同组件提供的配置属性。
- 高级设置：设置组件是否渲染、绑定DOM事件、设置组件是否循环等功能。

图 5-81 属性面板



底部节点树

底部节点树展示了从Body开始到当前选中节点的层级，单击可以切换选中层级。

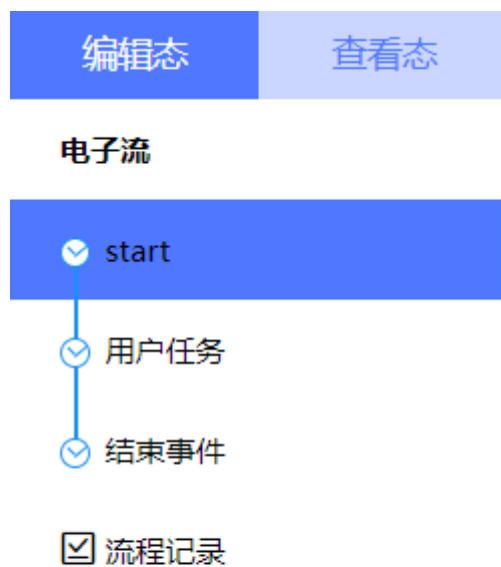
图 5-82 节点树

BODY > LcbSubmit > LcCollapse > LcUpload >

导航树栏

通过导航树，查看电子流的流程节点，单击进行切换。

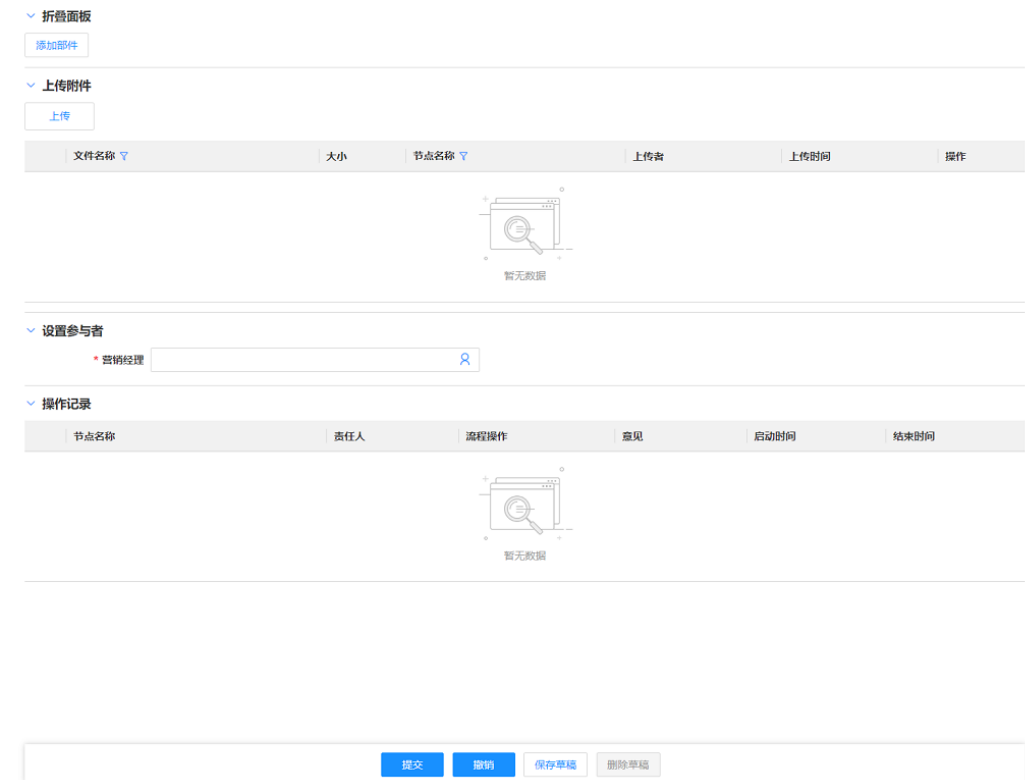
图 5-83 导航树



中心画布

中心画布位于整个设计器的中央位置，可以在左侧插件栏中的物料面板向中心画布拖入组件，也可以修改画布中已有的组件。

图 5-84 画布



5.6.2 物料及模板

功能介绍

物料是低代码开发的构建模块，它们是可复用的部件，可以轻松搭建并管理页面，分为[业务组件](#)和[IPD组件](#)。

模板是由许多基础物料组件构成的一个功能完善的可复用页面模板，对于一些特定的场景提供了[预置模板](#)。

添加组件

从物料面板将组件拖入画布。

单击左侧插件栏的图标，展开插件栏，选择模板或者组件，将其拖拽至中心画布或画布中的某一容器中，实现页面的快速构建。

图 5-85 添加组件



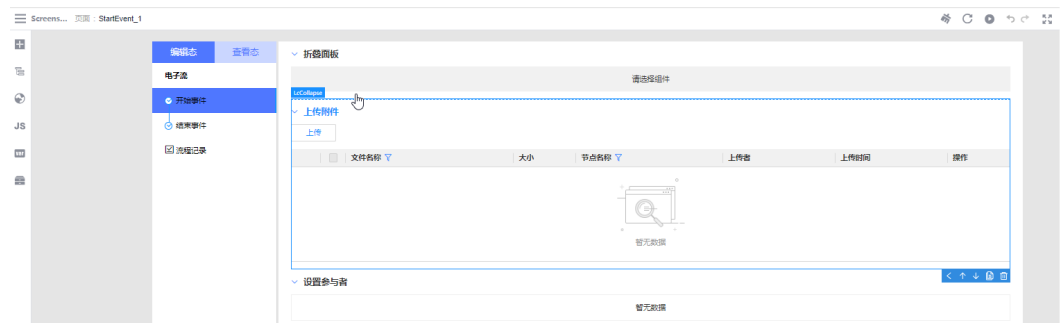
选中组件

在画布中选中组件

在画布中可以直观地看见已拖入的组件，鼠标单击即可选中组件。

- 优点：能在画布中直观地看见组件。
- 缺点：当组件层级嵌套过多时，直接选中父组件将变得困难。

图 5-86 在画布中选中组件

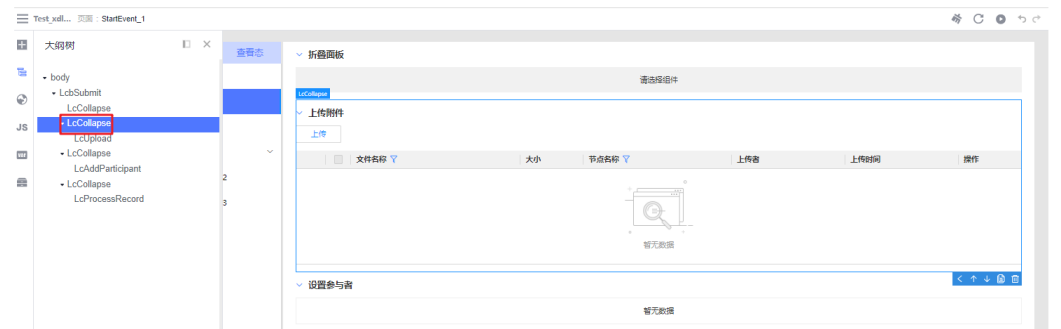


在大纲树中选中组件

在左侧大纲树插件中，以树的形式展示了当前页面中所有的组件，单击树节点即可选中画布中的组件。

优点：能够轻松选中画布中的所有组件。

图 5-87 在大纲树选中组件

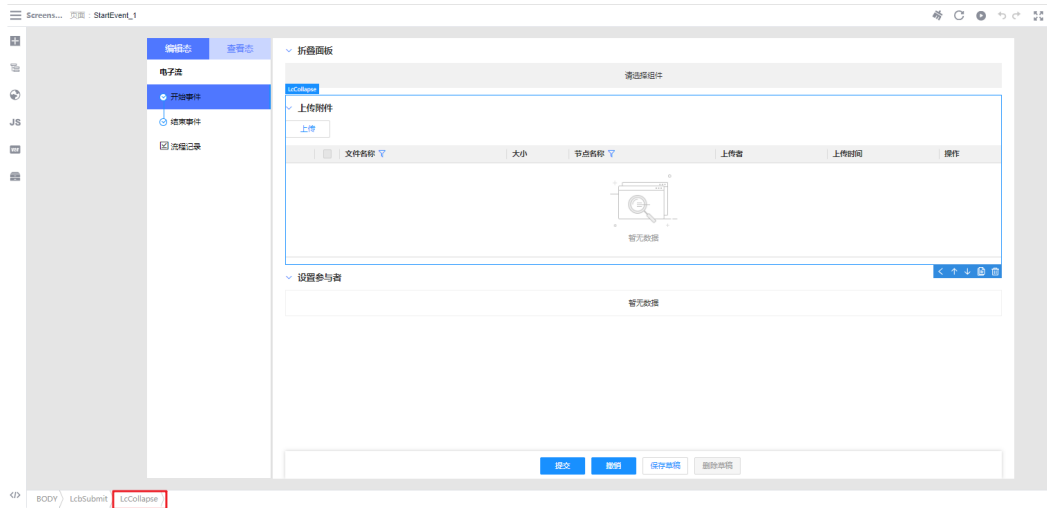


在底部节点树选中组件

底部节点树展示了从根节点到当前选中组件的层级节点，可以直接单击底部的节点树的某一个节点，选中父组件。

- 优点：能够直观看见当前节点相关的父层级，以便快速选中。
- 缺点：需提前选中某一组件，且只能选中当前组件的相关父组件，无法选中所有组件。

图 5-88 在底部节点树选中组件



编辑组件

编辑组件在画布中的位置

- 选中组件后，可以继续通过拖拽的方式移动组件所处的位置，进而改变整个页面的布局。
- 选中组件后，在画布中会在组件周围用一个方框，框中当前所选的组件。通过快捷按钮调整组件位置。

图 5-89 移动组件

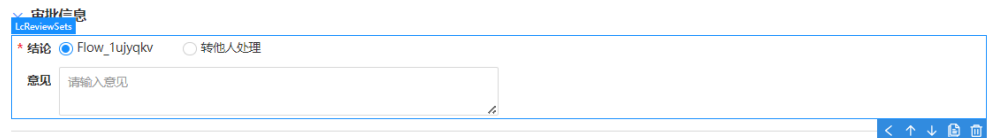







表 5-15 快捷按钮

图标	名称	说明
	选择父级	选择当前选中组件的父级组件。
	移动至上一级	当前选中组件在父组件中向上一层移动。

图标	名称	说明
	移动至下一级	当前选中组件在父组件中向下一层移动。
	复制	复制当前选中组件至同一层级。
	删除	删除当前选中组件。

编辑组件的属性、高级属性等


选中组件后，可以在右侧组件属性面板区设置所选组件的相关属性及高级事件等。

5.6.3 业务组件

5.6.3.1 添加部件

使用场景

添加一个部件库，通过搜索、勾选，添加需要的部件。

组件图标：

基本属性

表 5-16 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	选择模型	用于选择数据模型	模型选择器	-	-	√
2	显示	是否显示	switch开关	开/关	开	√
3	弹窗标题	弹窗标题	input输入框	-	-	√
4	查询接口	查询API，弹出服务定义选择框	服务定义	-	-	√
5	是否为分页接口	切换分页或跳页时，接口是否发起请求重新加载数据	switch开关	开/关	关	√
6	是否清空搜索表单	弹框展开，是否清空搜索表单	switch开关	开/关	关	√

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
7	是否清空搜索结果	弹框展开，是否清空搜索结果	switch开关	开/关	关	√
8	是否清空已选择	弹框展开，是否清空已选择	switch开关	开/关	关	√
9	是否显示高级搜索	是否显示高级搜索，打开开关后显示高级搜索按钮	switch开关	开/关	关	√
10	默认表单数据	表单数预设置默认值，JSON格式	代码编辑器	-	-	√
11	选择数据方法	设置选择数据的方法	代码编辑器	-	-	√
12	页面布局	设置表单布局	select选择器	一列/ 两列/ 三列	三列	√
13	表单项	弹框展开表单项设置	metarray list	-	-	√
14	启用表格多选校验方法	表格多选会做校验，是否可以多选	switch开关	-	-	√
15	校验返参字段	检验返回参数设置字段	input输入框	-	-	√
16	表格多选校验方法	通过绑定api的方式提供表格的校验请求	绑定API	-	-	√
17	表格分页	设置表格分页	代码编辑器	-	-	√
18	表格列	对标表格组件操作表格列	metarray list	-	-	√

属性详解

是否清空搜索表单

弹框展开后，清空表单中的历史数据值。

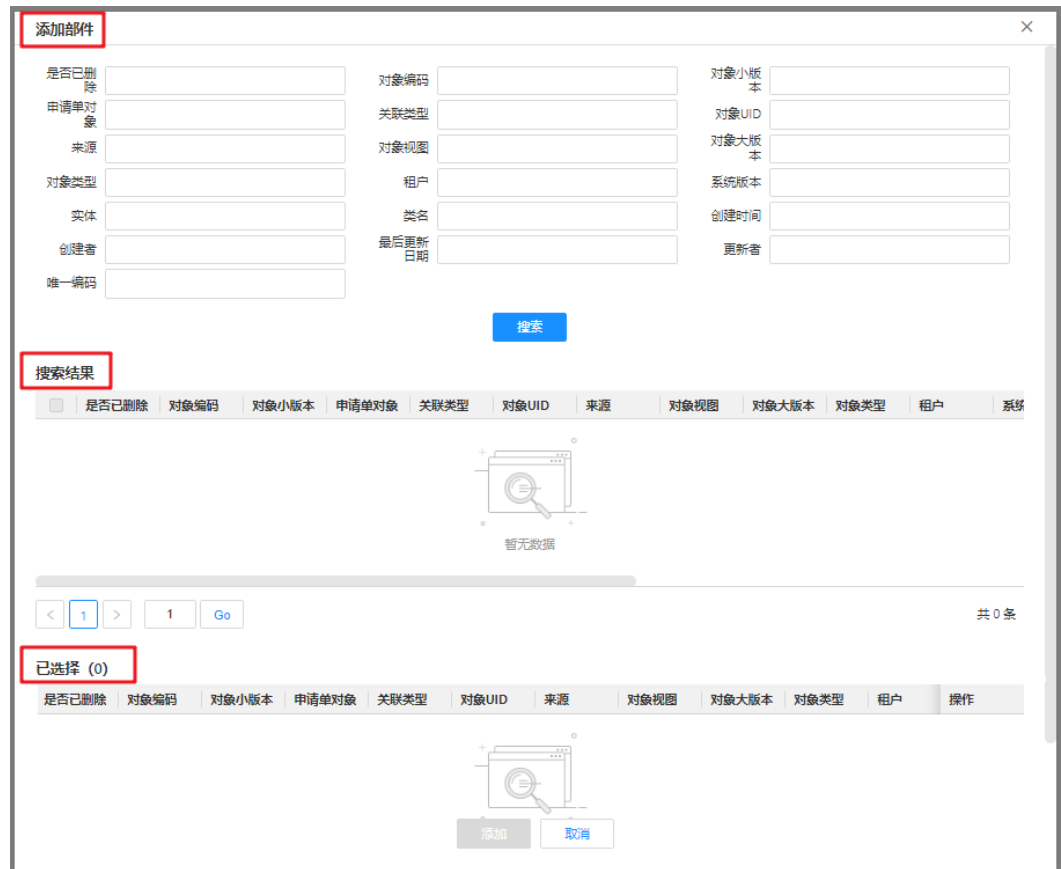
是否清空搜索结果

弹框打开后，清空“搜索结果”数据。

是否清空已选择

清空“已选择”中的数据。


图 5-90 清空数据



5.6.3.2 流程团队

使用场景

可以选择多个“设置参与者”，组成一整个流程团队审批流程。

组件图标：

基本属性

表 5-17 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	团队名称	设置流程团队的名称	input输入框	-	-	√
2	查询接口	查询流程团队的服务	查询定义服务	-	-	√

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
3	弹框标题	设置单击流程团队的标题	input输入框	-	流程团队	√
4	启用聊天框跳转	设置是否启用聊天框单击人员跳转	switch开关	开/关	关	√
5	显示弹框	是否显示弹出框	switch开关	开/关	关	√

属性详解

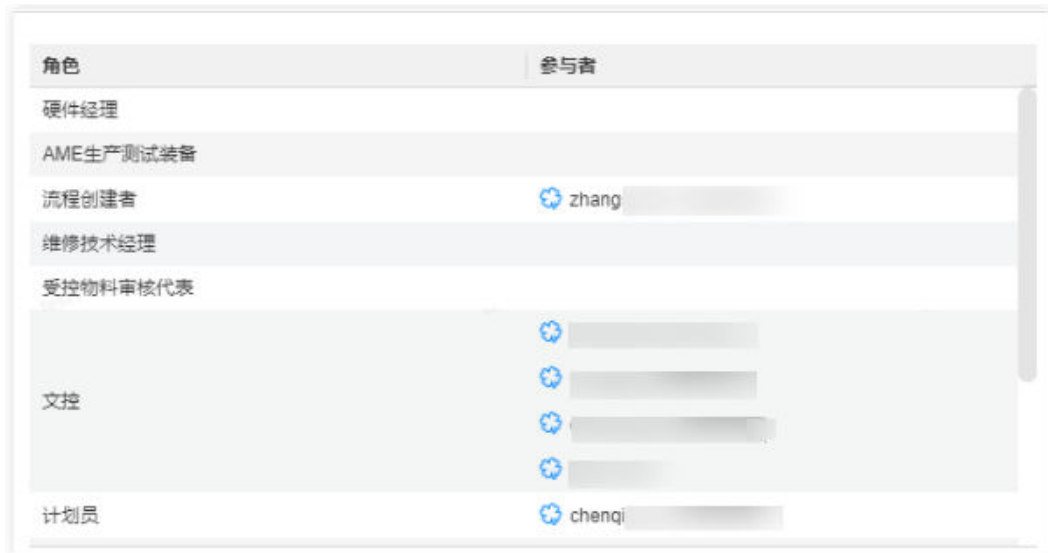
启用聊天框跳转

“参与者”列若只有一名成员，允许一键开启超链接。开启后，单击人员名称可直接打开聊天窗口。

显示弹框

单击团队名称，弹出表格框，展示角色名称、参与者。

图 5-91 团队角色



5.6.3.3 流程指引

使用场景

在界面中需要的位置，添加流程指引信息。

组件图标：

基本属性

表 5-18 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	展示弹框	在编辑态可以设置是否展示弹框	switch开关	开/关	关	√
2	弹框标题	弹框左上角的标题	input输入框	-	-	√
3	指引内容	流程指引的具体内容	代码编辑器	-	-	√
4	初始位置	手动填写css代码设置指引的初始位置	代码编辑器	-	-	√

属性详解

初始位置


编辑css代码，设置指引弹框的初始位置。

```
{
  "top": 100,
  "right": 20
}
```

5.6.3.4 评审三件套

使用场景

设置评审路由及意见，组件包含结论、意见及转他人处理。

组件图标：

基本属性

表 5-19 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	评审三件套标题	设置评审三件套标题	input输入框	-	-	√

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
2	意见标题	设置意见标题	input输入框	-	-	√
3	转审标题	设置转审标题	input输入框	-	-	√
4	禁用	是否禁用	switch开关	开/关	关	√
5	结论必填	是否结论必填	switch开关	开/关	关	√
6	意见必填	是否意见必填	switch开关	开/关	关	√
7	是否展示评审操作	是否展示评审操作	switch开关	开/关	开	√
8	是否自定义意见	是否自定义意见	switch开关	开/关	关	√
9	是否允许多选意见	是否允许多选意见	switch开关	开/关	关	√
10	自定义选项	新增自定义选项给自定义意见	metacoll apase	-	-	√

属性详解

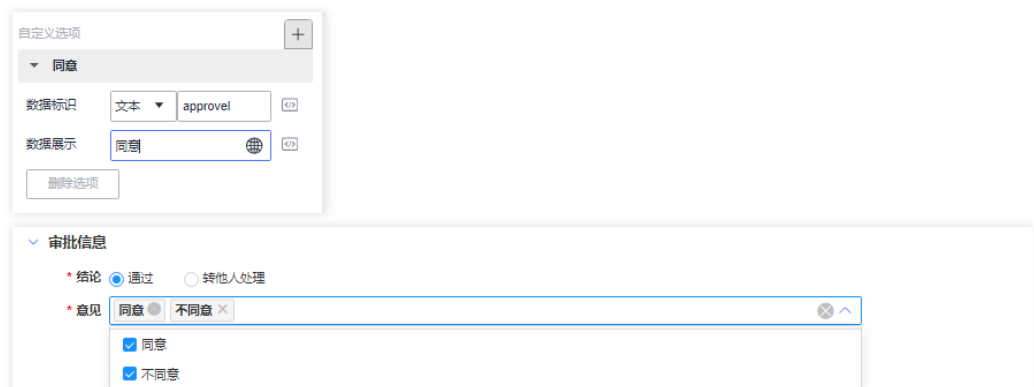
是否展示评审操作

如果不展示评审操作，则在审批信息一栏展示空白。

是否自定义意见

开关打开后，页面意见展示下拉列表，可自定义意见；自定义意见选项可配置下拉选项，且可设置多种意见。

图 5-92 自定义意见



转他人处理

选择转他人处理，底部出现转审人，用户可以选择将流程任务转给其他人处理。

图 5-93 转他人处理



5.6.3.5 流程记录

使用场景

通过表格展示流程记录，只做记录展示，不可修改。


组件图标：

图 5-94 流程记录



- 表字段：节点名称、责任人、流程操作、意见、启动时间、结束时间。（不允许用户修改）
- 排序：默认所有列均支持排序，数据按处理时间倒序。（不允许用户修改）

5.6.3.6 评审记录

使用场景

展示当前节点处理任务及当前任务前的评审记录（仅查看）。


组件图标：

图 5-95 评审记录



基本属性

表 5-20 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	待评审对象内容	手动输入待评审对象内容	input输入框	-	-	√
2	待评审对象字段映射	映射后台的评审对象字段	input输入框	-	-	√
3	待评审对象数量字段映射	映射后台评审对象数量字段映射	input输入框	-	-	√

5.6.3.7 转审记录

使用场景

通过表格展示转审记录，只做记录展示，不可修改。


组件图标：

图 5-96 转审记录




图 5-96 展示了转审记录的表格界面。表格包含以下列：节点名称、原责任人、新责任人、意见、处理时间。表格下方显示“暂无数据”。

- 表字段：节点名称、原责任人、新责任人、意见、处理时间。（不允许用户修改）
- 排序：默认所有列均支持排序，数据按处理时间倒序。（不允许用户修改）

5.6.3.8 附件

使用场景

文件上传与下载管理，单击上传后可弹出文件夹。

组件图标：

基本属性

表 5-21 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	获取附件列表类型	附件列表类型设置	select选择数据	所有/当前用户/他人	-	√
2	上传按钮文本	上传按钮文本设置	input输入框	-	-	√
3	只读	只读情况下，上传按钮消失，只能看表格数据	switch开关	开/关	关	√
4	文件最大限制 (M)	默认大小为100MB，可配置的最大限度为500MB	数字输入框	0-500MB	100	-


5.6.4 IPD 组件

5.6.4.1 容器组件

5.6.4.1.1 BOX (盒子)

使用场景

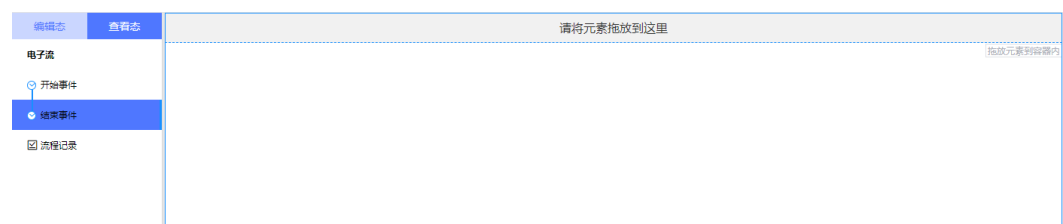
布局容器，内置插槽，可以拖入所有的组件元素，方便对页面进行布局、管理。

组件图标：

BOX 组件

组件在画布中的展示。


图 5-97 BOX 组件



5.6.4.1.2 栅格布局（列容器）

使用场景

通过基础的24分栏结构，迅速简便地创建栅格布局。

组件图标：

基本属性

表 5-22 基础信息

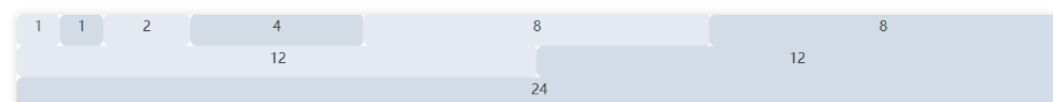
序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	占据的列数	栅格占据的列数	数字输入框	1-24	-	√
2	左侧间隔格数	栅格左侧的间隔格数	数字输入框	0-23	-	√
3	向(左 右)移动格数	栅格向(左/右)移动格数	数字输入框	0-23	-	√
4	≥(1920 1200 992 768 <768)px 响应式栅格数	当前页面的宽度在(超出/少于)对应像素大小时栅格占据的列数	数字输入框	1-24	-	√

属性详解

占据的列数

一行占据的总列数为24，可选值为1-24，实际展示的大小为行宽 × (占据的列数 / 24) * 100%，若配置了页面对应宽度的响应式栅格数时，优先使用响应式栅格数。

图 5-98 占据的列数



左侧间隔格数

栅格距离左侧的间隔格数，计算方式同占据的列数。

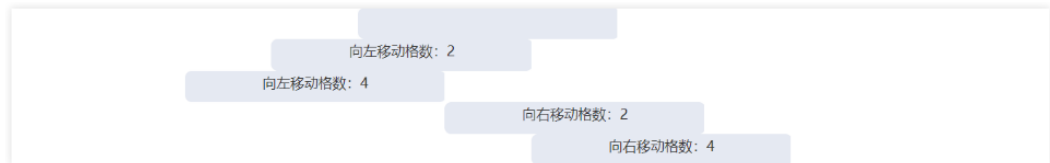
图 5-99 左右间隔格数



向(左 | 右)移动格数

栅格向左或向右移动格数，计算方式同占据的列数。

图 5-100 移动格数



响应式栅格数

当前页面的宽度在(超出/少于)对应像素大小时栅格占据的列数。

- 当前页面的宽度 $\geq 1920\text{px}$ 时，列容器宽度为 $(4/24)*100\% = 16.6666666667\%$
- 当前页面的宽度 $\geq 1200\text{px}$ 时，列容器宽度为 $(4/24)*100\% = 16.6666666667\%$
- 当前页面的宽度 $\geq 992\text{px}$ 时，列容器宽度为 $(8/24)*100\% = 33.3333333333\%$
- 当前页面的宽度 $\geq 768\text{px}$ 时，列容器宽度为 $(12/24)*100\% = 50\%$
- 当前页面的宽度 $< 768\text{px}$ 时，列容器宽度为 $(24/24)*100\% = 100\%$


图 5-101 响应式栅格数

$\geq 1920\text{px}$ 响应式栅格数	<input type="text" value="4"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>	<input type="button" value="↔"/>
$\geq 1200\text{px}$ 响应式栅格数	<input type="text" value="4"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>	<input type="button" value="↔"/>
$\geq 992\text{px}$ 响应式栅格数	<input type="text" value="8"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>	<input type="button" value="↔"/>
$\geq 768\text{px}$ 响应式栅格数	<input type="text" value="12"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>	<input type="button" value="↔"/>
$< 768\text{px}$ 响应式栅格数	<input type="text" value="24"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>	<input type="button" value="↔"/>

5.6.4.1.3 栅格布局（行容器）

使用场景

通过基础的24分栏结构，迅速简便地创建栅格布局。

组件图标：

基本属性

表 5-23 基础信息

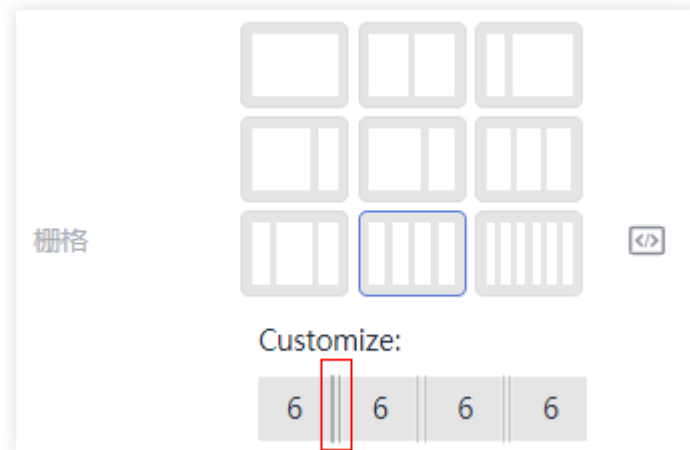
序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	栅格	一行中栅格的分栏数	布局选择器	24, 12:12, 6:18, 18:6, 16:8, 8:8:8, 6:12:6, 6:6:6:6, 4:4:4:4:4:4	12:12	√
2	栅格间隔	栅格的间隔格数	数字输入框	≥0	-	√
3	Flex布局模式	是否开启Flex布局	开关	关/开	关	√
4	水平排列方式	开启Flex布局后的水平排列方式(需开启Flex布局模式)	下拉选择框	从行首开始排列/从行尾开始排列/居中/两端对齐/均匀分布	-	√
5	垂直排列方式	开启Flex布局后的垂直排列方式(需开启Flex布局模式)	下拉选择框	居下, 居中, 居上	-	√

属性详解

栅格

控制行容器中列容器占据的栅格数，提供了9个可用的预设，并且可以通过拖动下方的自定义滑块，控制每一个列容器的宽度占比。

图 5-102 栅格



栅格间隔

列容器两两之间相距的距离（单位：像素）。

图 5-103 栅格间隔

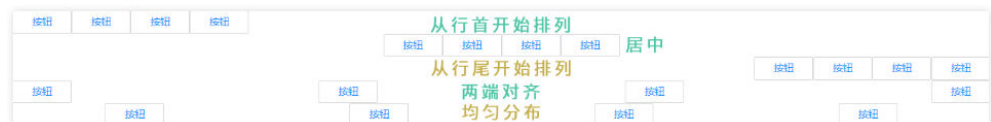


Flex布局模式

是否开启Flex布局模式，开启后展示“水平排列模式”和“垂直排列模式”两个选项。

- 水平排列方式如图5-104所示。

图 5-104 水平排列方式



- 垂直排列方式如图5-105所示。


图 5-105 垂直排列方式



5.6.4.1.4 折叠面板

使用场景

用于需要折叠、隐藏高度受限的内容。

组件图标: 

基本属性

表 5-24 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	标题	折叠面板显示的标题	代码编辑框	-	{ "en_US": "Collapse Panel", "zh_CN": "折叠面板" }	√
2	启用提示	是否启用提示	开关	关/开	关	√
3	提示内容	提示的内容文本	输入框	-	-	√

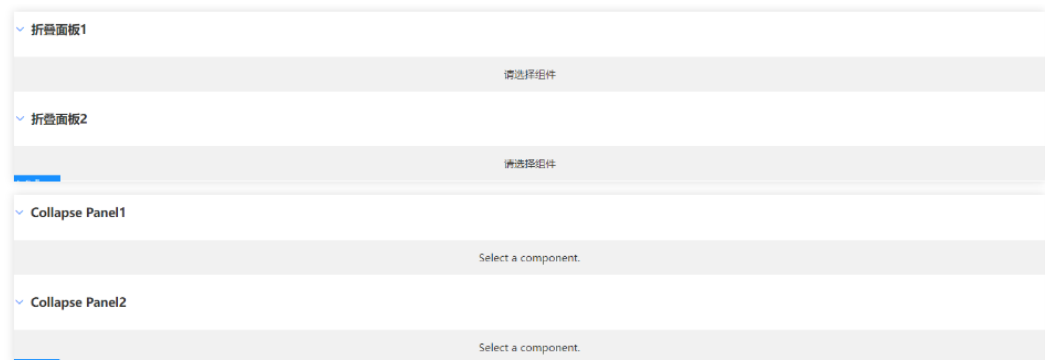
属性详解

标题


使用JSON格式的代码配置，“en_US”为英文状态下显示的标题文本，“zh_CN”为中文。

```
{
  "en_US": "Collapse Panel",
  "zh_CN": "折叠面板"
}
```

图 5-106 折叠面板



启用提示

开启后，鼠标放在上后会显示提示内容。

提示内容

单击后在弹出编辑框中输入提示内容。

高级事件


表 5-25 高级事件

事件名	触发条件	说明
onChange	折叠面板状态改变	单击折叠面板使折叠状态改变时触发

5.6.4.1.5 页签

使用场景

将各类物料组件拖入进标签页组件中，分隔内容上有关联但属于不同类别的数据。

组件图标：

基本属性

表 5-26 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	默认页签	第一个展示选项卡的名称	下拉选择框	选项卡中非禁用、非隐藏的页签项	-	√
2	隐藏页签	隐藏哪些页签	下拉选择框	选项卡中的页签项	-	√
3	风格类型	页签展示风格类型	下拉选择框	默认、扁平卡片、带边框卡片	默认	-
4	选项卡所在位置	选项卡所在位置	下拉选择框	上、左、右	上	-
5	Tab选中触发事件	单击标签页项时触发事件	代码编辑器	-	-	√

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
6	页签初始化事件	页签初始化加载时触发事件	代码编辑器	-	-	√

表 5-27 选项卡

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	选项卡ID	当前选项卡ID，标签页项的唯一标识符	输入框	-	-	-
2	选项卡标题	当前选项卡标题	国际化输入框	-	-	-
3	是否禁用	当前选项卡是否禁用，被禁用的标签页项会被置灰，且无法单击、无法选为默认页签	开关	开、关	关	-

属性详解

默认页签

默认页签以下拉选择框方式显示选项，选项名称为“选项卡标题”，选择的默认页签为标签页首次加载激活的页签。

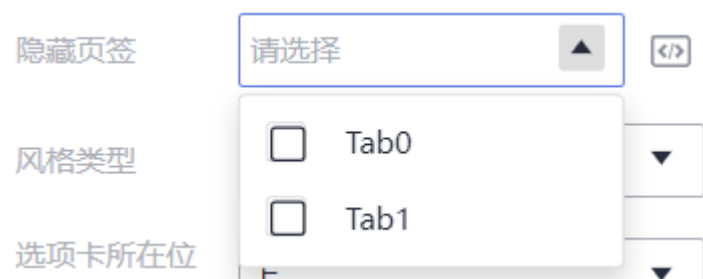
图 5-107 默认页签



隐藏页签

隐藏页签的下拉选择框（多选）显示所有选项卡。隐藏的页签将不再显示，且无法被选中为默认页签。

图 5-108 隐藏页签



风格类型

图 5-109 风格类型



选项卡所在位置

图 5-110 选项卡位置



Tab选中触发事件

```
/*  
 * @function tabClick - 单击选中Tab标签页项触发事件  
 * @param {object} tabItem - 被选中的标签 tab 实例  
 */  
function tabClick(tabItem) {  
    //自定义操作...  
}
```

页签初始化事件

```
/*  
 * @function tabInitial - 首次加载标签页触发事件  
 */  
function tabInitial(tabItem) {  
    //自定义操作...  
}
```

5.6.4.2 信息组件

5.6.4.2.1 文本展示

使用场景

展示文本或者展示超链接进行页面跳转。

组件图标: 

基本属性

表 5-28 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	文本内容	文本展示中的内容	代码编辑器	-	-	√

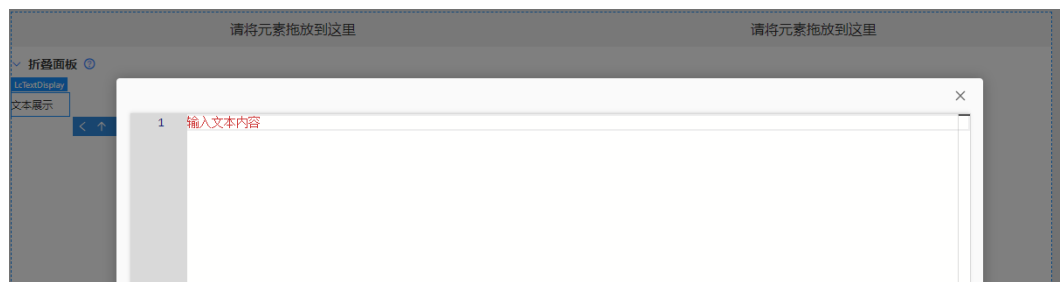
序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
2	启用超链接	文本展示是否为超链接	开关	关/开	关	√
3	文本宽度	文本展示所占的宽度	数字输入框	≥0	-	√

属性详解

文本内容

单击文本内容，在弹出框内部填入想展示的文本，编辑完成后单击“保存”。

图 5-111 编辑文本




启用超链接

单击启用超链接开关，显示如下属性：

表 5-29 启用超链接属性

属性名	说明	设置器类型	可选值	默认值	支持变量绑定
超链接地址	填写超链接地址，即后续想跳转的地址。	输入框	-	例如：https://www.example.com	√
超链接新开页面	打开开关，预览态单击超链接会跳转到全新的页面(安全考虑，不允许在本页面跳转，非本项目请求头的地址，只能新换一个页面打开，不然会被拦截)。	switch开关	开/关	-	√

属性名	说明	设置器类型	可选值	默认值	支持变量绑定
超链接变量	-	collapse新增输入框	-	属性：文本内容填写的值	√
<p>1. 超链接地址的填写如果包含变量，变量的形式是\${ }，例如： https://www.\${abc}example.com，即abc是变量，其值可以任意修改。</p> <p>2. 链接变量名用于映射地址里面的变量值。</p> <p>3. 链接变量值用于最终展示的值，即变量名映射匹配上后，最终会将变量值填入超链接地址里做最终的地址。 最终链接地址为：https://www.defgexample.com</p> <p>图 5-112 超链接变量</p> 					

文本宽度

文本展示所占的宽度，单位px（像素）。


图 5-113 文本宽度



5.6.4.2.2 状态标识

使用场景

为功能添加状态时使用，共可以选择三种状态，成功、失败和告警。

组件图标：

基本属性

表 5-30 基础信息

序号	属性	说明	设置器类型	可选值	默认值	支持变量绑定
1	类型	选择不同类型的标识	select选择框	-	成功	√

属性详解

类型

共有三种类型标识可选，不同状态标识展示不同颜色按钮。

图 5-114 状态标识



5.6.4.2.3 帮助文档

使用场景

以角标的形式，给当前功能添加提示，具有帮助指引的功能。当鼠标悬停在图标上时，会显示提示信息。

组件图标： ?

基本属性

表 5-31 基础信息

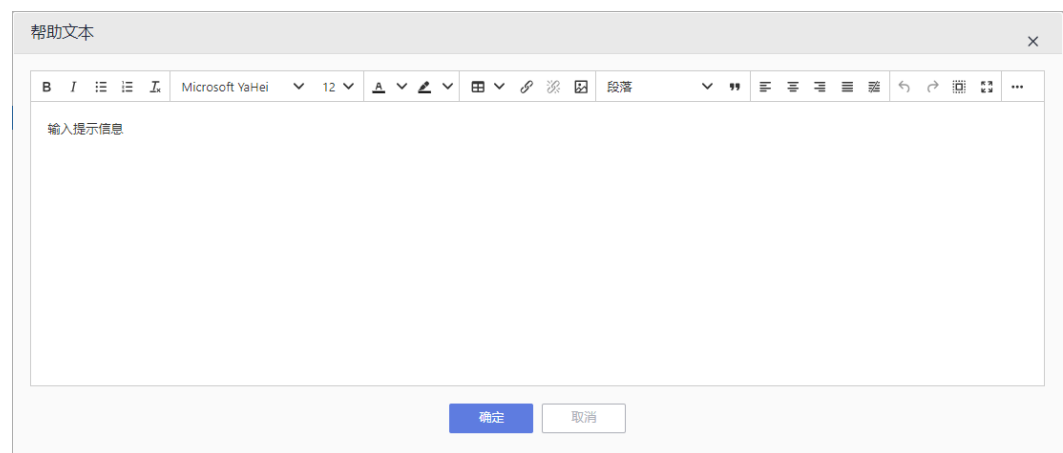
序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	帮助文档提示	填写需要提示用户的文字	文本编辑器	-	-	√
2	提示框大小	通过设置数字大小控制提示框的大小	数字输入框	-	-	√
3	提示位置	选择弹出框展示位置	select选择器	-	-	√

属性详解

帮助文档提示

单击输入框后会弹出帮助文本的编辑器，这里可以设置提示文档文本格式。

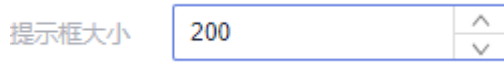
图 5-115 帮助文档提示



提示框大小

可手动输入数字调整大小，也可通过右侧的上下按钮进行调整。

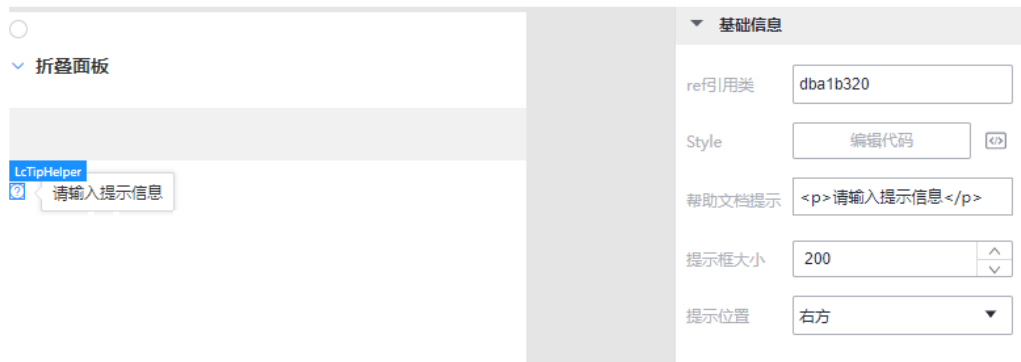
图 5-116 提示框大小



提示位置

用来控制提示框的展示方位。

图 5-117 提示位置



5.6.4.2.4 弹框

使用场景

提供弹框功能，里面插槽可拖入任意组件，用来做遮罩层弹框展示。

组件图标：

基本属性

表 5-32 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	弹窗标题	弹窗左上角文字编辑	input 输入框	-	弹窗	√
2	展示弹窗	是否让弹窗展示	switch 开关	开/关	开	√
3	弹窗宽度	设置弹框的宽度	数字输入框	-	-	√

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
4	底部操作	底部操作的两个按钮区域是否展示	switch 开关	开/关	开	√
5	全屏展示弹框	弹框变大到全屏展示	switch 开关	开/关	关	√
6	单击遮罩层关闭弹框	单击灰色遮罩层区域弹窗关闭	switch 开关	开/关	关	√
7	展示关闭图标	右上角关闭图标展示控制	switch 开关	开/关	关	√
8	首尾内容居中	弹框内容区的内容是否水平居中	switch 开关	开/关	关	√
9	关闭弹框销毁元素	关闭弹框后清空弹框内容区的内容	switch 开关	开/关	开	√
10	ESC键关闭弹窗	按键盘的ESC键弹窗会关闭	switch 开关	开/关	关	√

属性详解

底部操作

底部操作区两个按钮，可同时控制显示与隐藏，使用的是Button组件，具体属性可参考[按钮组件](#)属性。

图 5-118 底部操作



高级事件


表 5-33 高级事件

序号	事件名	触发条件	说明
1	onClose	弹框关闭后触发	单击确认、取消或关闭触发close回调函数
2	onOpen	弹框打开后触发	弹框打开后触发事件可用于获取弹框里的数据

5.6.4.2.5 消息提示框

使用场景

提供消息提示功能，常用于主动操作后的反馈提示。

组件图标：

基本属性

表 5-34 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	显示	通过JS方法或事件，控制弹框显示或隐藏	switch开关	开/关	开	√
2	类型	选择消息提示框类型	select选择框	信息，成功，告警，错误，弱信息	信息	√
3	消息内容	自定义消息内容	input输入框	-	-	√
4	展示时间	提示框展示时长，单位为秒，若设置为0则不会关闭	数字输入框	-	5	√
5	宽度	提示框宽度	数字输入框	-	400	√
6	是否显示关闭按钮	提示框中的关闭按钮，可以手动关闭提示框	switch开关	开/关	关	√

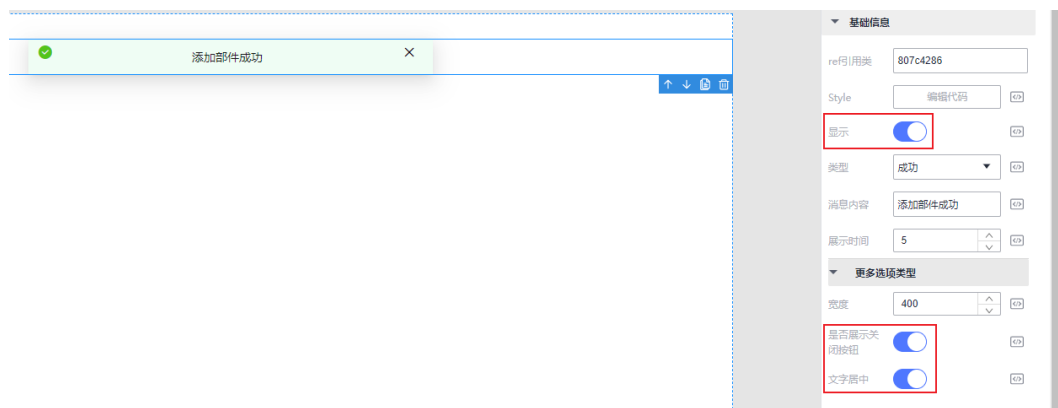
序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
7	文字居中	提示消息居中显示	switch开关	开/关	关	√

属性详解

显示

消息提示框的显示设置。

图 5-119 消息提示框

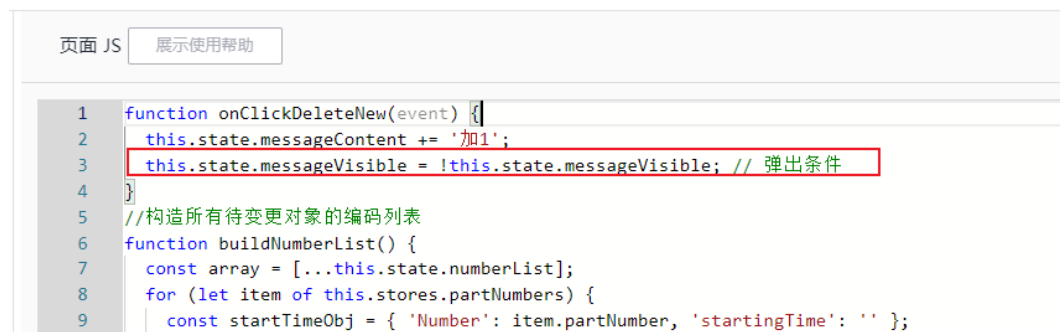


若需要在指定条件下弹出提示框，可以将绑定的显示变量值取反。

例如：自定义变量为“this.state.messageVisible”，打开提示框需要在对应地方加入如下函数：

```
this.state.messageVisible = !this.state.messageVisible
```

图 5-120 按条件提示




5.6.4.3 表单组件

5.6.4.3.1 按钮

使用场景

页面中承载操作的控件，如确认、提交、取消或下载等操作，也可用于表单表格中的操作。

组件图标：

基本属性

表 5-35 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	按钮名称	按钮中间显示的文本内容	多语言输入框	-	按钮	√
2	按钮类型	按钮样式类型	下拉选择框	主要、文本、告警、提示、成功、危险	-	√
3	禁用	按钮是否为禁用状态	开关	关/开	关	√

属性详解

按钮名称

可以自定义按钮名称。



按钮类型

按钮样式类型。



禁用

- 正常状态：可单击。
- 禁用状态：不可单击。



高级事件


表 5-36 高级事件

序号	事件名	触发条件	说明
1	onClick	单击按钮	单击按钮后触发JS方法或绑定的API接口

5.6.4.3.2 单选框

使用场景

用于选项选择，可以让用户在必选一中做选择操作。

组件图标：

基本属性

表 5-37 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	默认值	多个选项时，控制默认值是什么	select选择框	选项属性添加选项后，默认值从中选择	-	√
2	禁用	禁用所有的单选框，按钮置灰无法选择	switch开关	开/关	关	√
3	选项	手动配置需要做选择的选项	collapse新增选项框	-	-	√

属性详解

默认值

初始化没有数据，默认值选项为空；下面“选项”属性新增选项后，可在这里配置哪个为初始默认值，确定默认值后页面初始化默认选中该值。

图 5-121 单选框



选项

1. 单击右上角的新增按钮，可新增选项。
2. “数据标识”，下拉选择“选项类型”（文本、数值、布尔）。
若选文本，则联动展示输入框；若选数值，联动展示计数器；若选布尔，则联动展示下拉选择，可选项为true/false。
3. “数据展示”用来展示选择的数据，填写后返显到单选框后面展示。

5.6.4.3.3 多选框

使用场景

提供多选场景，多个数据可选择，且可自动配置多选数据。

组件图标：

基本属性

表 5-38 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	默认值	多个选项时，控制默认值是什么，且下拉列表的值可以多选	select选择框	选项属性添加选项后，默认值从中选择	-	√

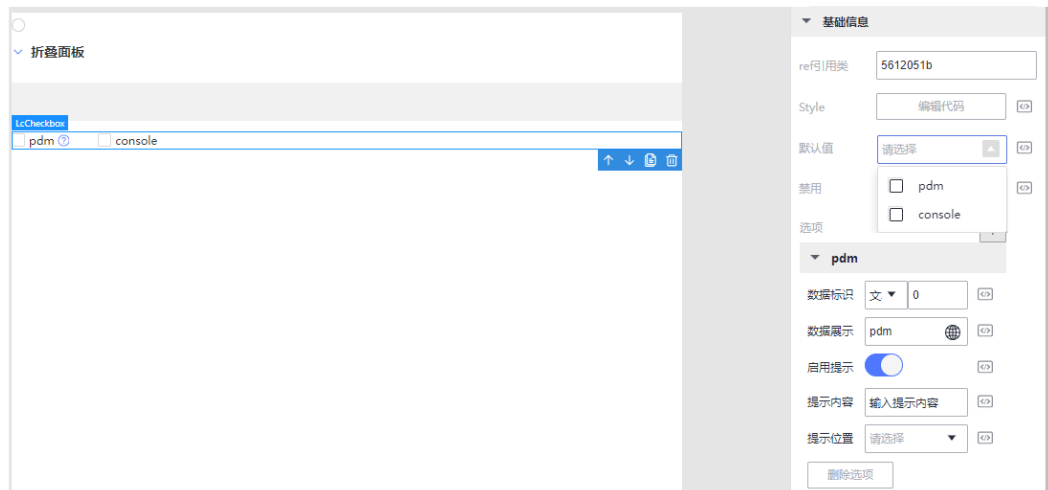
序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
2	禁用	禁用所有的多选框，按钮置灰无法选择	开/关	开/关	-	√
3	选项	手动配置需要做选择的选项	collapse新增选项框	-	-	√

属性详解

默认值

默认值选项为“选项”属性配的值，且默认值可多选。

图 5-122 多选框



选项

1. 单击右上角的新增按钮，可新增选项。
2. “数据标识”，下拉选择“选项类型”（文本、数值、布尔）。
若选文本，则联动展示输入框；若选数值，联动展示计数器；若选布尔，则联动展示下拉选择，可选项为true/false。
3. “数据展示”用来展示选择的数据，填写后返显到多选框后面展示。
4. 启用提示按钮打开，多选框后会出现提示图标，鼠标悬停在图标会弹出提示框。
5. 提示内容可自定义提示框里的文本。
6. 提示位置用来控制提示框的展示方位。

5.6.4.3.4 输入框

使用场景

输入框承载文本内容，多用于填写内容提交、确认场景。

组件图标:

基本属性

表 5-39 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	默认值	输入框展示的默认值	input输入框	-	-	√
2	类型	文本框的类型	select选择框	文本框/文本域	-	√
3	占位符	placeholder文字内容，即输入框提示语言	input输入框	-	-	√
4	最大长度	控制输入框的长度大小	数字输入框	-	-	√
5	禁用	控制输入框不可输入	switch按钮	开/关	关	√
6	只读	文本框消失，只展示填写的内容	switch开关	开/关	关	√

属性详解

类型

文本框类型分为“文本框”和“文本域”。文本框，输入文本相对较少；文本域，可以输入一整个区域的文字。

占位符

用作输入框的提示文字。

只读

只读按钮打开后，input输入框消失，只展示输入框的文字。

高级事件

表 5-40 高级事件


序号	事件名	触发条件	说明
1	onBlur	光标离开	鼠标光标从输入框离开触发回调

序号	事件名	触发条件	说明
2	onFocus	光标聚焦	鼠标光标点击输入框触发回调
3	onChange	文本变化	输入框文本发生变化触发回调
4	onInput	输入文字	输入框键入文本的时候触发回调

5.6.4.3.5 选择器

使用场景

在页面中供用户选择多个数据，展示形式是下拉列表。

组件图标：

基本属性

表 5-41 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	默认值	下拉框默认选中值	select选择框	-	-	√
2	多选	设置下拉框是否可多选	switch开关	开/关	关	√
3	禁用	设置选择框是否禁用	switch开关	开/关	关	√
4	数据标识字段	作为value唯一标识的键名，绑定值为对象类型时必填，同时需要el-select添加属性value-key="id"，el-option绑定属性key="item.label"	input输入框	-	-	√
5	数据展示字段	作为tag展示对象属性的键名，绑定值为对象类型时必填	input输入框	-	-	√
6	占位符	提示用户的作用	input输入框	-	-	√
7	选项数据	填入数组，数据形式以key，value的形式展示	metaco代码编辑器	-	-	√

属性详解

数据标识字段

设置数据标识字段，即key字段的绑定值，设置后选中之后值为当前设置的标识字段。

数据展示字段

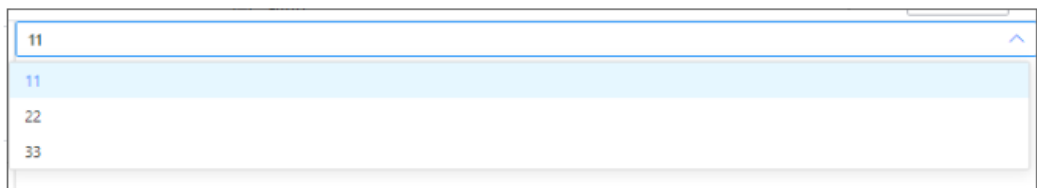
设置数据展示字段，即label字段的绑定值，设置后下拉列表展示的值是设置的字段。

选项数据

选择“编辑代码”，通过填写JSON数据，创建选项数据。

```
[  
  {  
    "value": 1,  
    "label": 11  
  },  
  {  
    "value": 2,  
    "label": 22  
  },  
  {  
    "value": 3,  
    "label": 33  
  }  
]
```


图 5-123 选项展示



5.6.4.3.6 时间选择器

使用场景

展示时间，可以精确到时分秒，也可设置一个时间段供用户选择。

组件图标：

基本属性

表 5-42 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	默认值	设置默认时间	时间选择器	-	-	√
2	只读	是否设置时间只读	switch开关	开/关	关	√

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
3	禁用	是否让时间不可选择	switch开关	开/关	关	√
4	可编辑	可手动在输入框手动修改日期	switch开关	开/关	开	√
5	输入框尺寸	可以选择输入框大小	select选择框	中/小/迷你	-	√
6	时间显示格式	设置最终日期显示格式	select选择框	-	-	√
7	数据传递格式	选中日期后传递的格式	select选择框	-	-	√
8	面板对齐方式	日期选择的面板弹出位置	select选择框	-	-	√
9	默认时间	下拉面板打开时默认显示的时间	时间选择器	-	-	√
10	启用范围选择	是否可以设置时间范围	switch开关	开/关	关	√
11	开始占位符	开始日期的placeholder提示	input输入框	-	-	√
12	结束占位符	结束日期的placeholder提示	input输入框	-	-	√
13	分隔符	两个日期选择器之间的分隔符	input输入框	-	-	√
14	可选时间段	设置日期在哪个时间段可以选择	input输入框	-	-	√

属性详解

输入框尺寸

控制输入框的大小，共有三种尺寸可以选择。

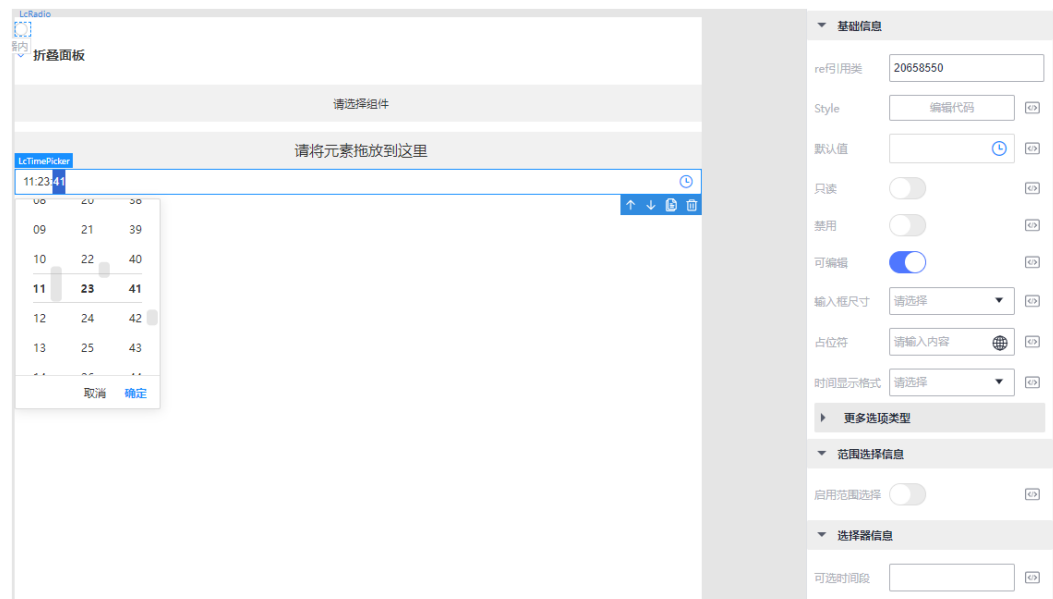
时间显示格式

设置日期格式。

启用范围选择

打开开关后，会显示“开始占位符”“结束占位符”和“分隔符”，同时页面出现两个日期选择器，为开始和结束时间，可选时间段，此时输入框会隐藏。

图 5-124 时间选择器



5.6.4.3.7 日期选择器

使用场景

展示日期及时间。

组件图标:

基本属性

表 5-43 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	默认值	设置默认时间	时间选择器	-	-	√
2	只读	是否设置时间只读	switch开关	开/关	关	√
3	禁用	是否让时间不可选择	switch开关	开/关	关	√
4	可编辑	可手动在输入框手动修改日期	switch开关	开/关	开	√
5	占位符	用户提示信息	input输入框	-	-	√
6	日期类型	设置日期选择器的展示类型	select选择框	-	日期时间	√

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
7	日期显示格式	选中后回显的日期格式	select选择框	-	年月日 小时分 分钟秒	√
8	面板对齐方式	日期选择的面板弹出位置	select选择框	-	-	√
9	默认日期	下拉面板打开时默认显示的日期	日期选择器	-	-	√
10	数据传递格式	选中日期后传递的格式	select选择框	-	-	√
11	开始占位符	开始日期的提示	input输入框	-	-	√
12	结束占位符	结束日期的提示	input输入框	-	-	√
13	分隔符	两个日期选择器之间的分隔符	input输入框	-	-	√
14	日期范围独立展示	日期选择器选择范围时是否支持选单个日期	switch开关	-	-	√
15	禁用日期	日期选择器设置禁用状态	代码编辑器	-	-	√
16	快捷项	快捷操作，给日期选择器创建快捷事件	collapse新增按钮	-	-	√

属性详解

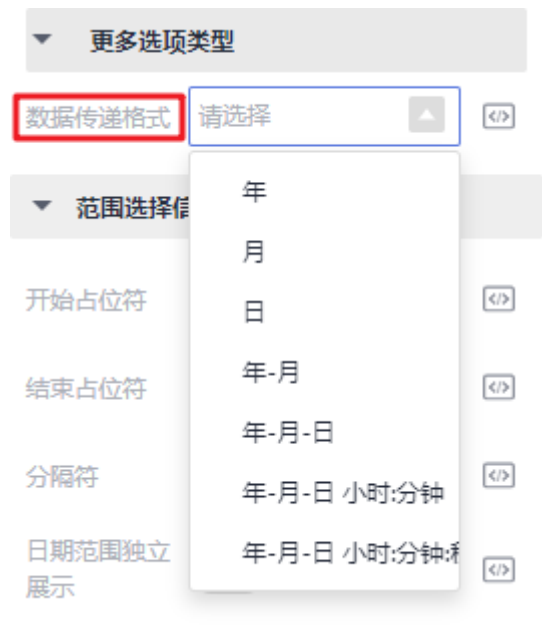
日期类型

设置日期选择器的类型，需特别注意的是，选择日期、月份范围等后，底下才会显示“范围选择信息”等属性。

数据传递格式

设置数据传递格式，有如下格式：

图 5-125 数据传递格式



禁用日期

通过编写JS代码，设置禁用的日期。

```
function disabledDate(time)
{
  // return time.getTime() > Date.now();
}
```

高级事件

表 5-44 高级事件

序号	事件名	触发条件	说明
1	onChange	用户确认选中的值时触发	当用户选择某一日日期触发的回调
2	onBlur	当input失去焦点时触发	当光标离开日期输入框触发的回调
3	onFocus	当input获取焦点时触发	当光标选择日期输入框触发的回调

5.6.4.3.8 选人组件

使用场景

选人组件可以配合后台，搜索、添加、删除单个成员或多个成员。

组件图标:

基本属性

表 5-45 基础信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	默认值	默认选中人员	input输入框	-	-	√
2	占位符	输入框占位文本	input输入框	-	-	√
3	多选	是否多选	switch按钮	开/关	关	√
4	禁用	是否禁用	switch按钮	开/关	关	√
5	选人必填	选人是否必填	switch按钮	开/关	关	√
6	校验提示内容	手动输入校验规则来校验	input输入框	-	-	√

属性详解

多选

打开多选按钮后，选中人数可以多选。选择多人时，支持删除不需要的选项。

高级事件

表 5-46 高级事件


序号	事件名	触发条件	说明
1	onBlur	失去焦点时触发	光标离开输入框时触发
2	onFocus	获取焦点时触发	光标点击输入框触发
3	onClear	点击清空按钮时触发	点击清空选中的人后触发回调
4	onSelectData	选中人员变化时触发	选中人发生变化可以触发回调

5.6.4.4 数据组件

5.6.4.4.1 表单

使用场景

以表单的形式展现多种类型数据。

组件图标: 

基本属性

表 5-47 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	选择模型	用于选择数据模型	模型选择器	-	-	√
2	表单布局	设置表单布局	select选择器	具体见下	-	√
3	只读	控制表单内部只显示文字，不显示输入框	switch开关	开/关	关	√
4	绑定数据	给表单绑定数据	代码编辑器	-	-	√
5	表单项	选择模型后触发的表单项	metalist表单项	-	-	√

属性详解

选择模型

单击选择模型，弹出模型选择框，选择需要的[表单模型](#)。

表单项

表单项选择后，渲染出所有表单项，表单项包含批量显示与隐藏、编辑、隐藏单项等功能。单击编辑后，可以手动编辑单个表单项。

图 5-126 表单项

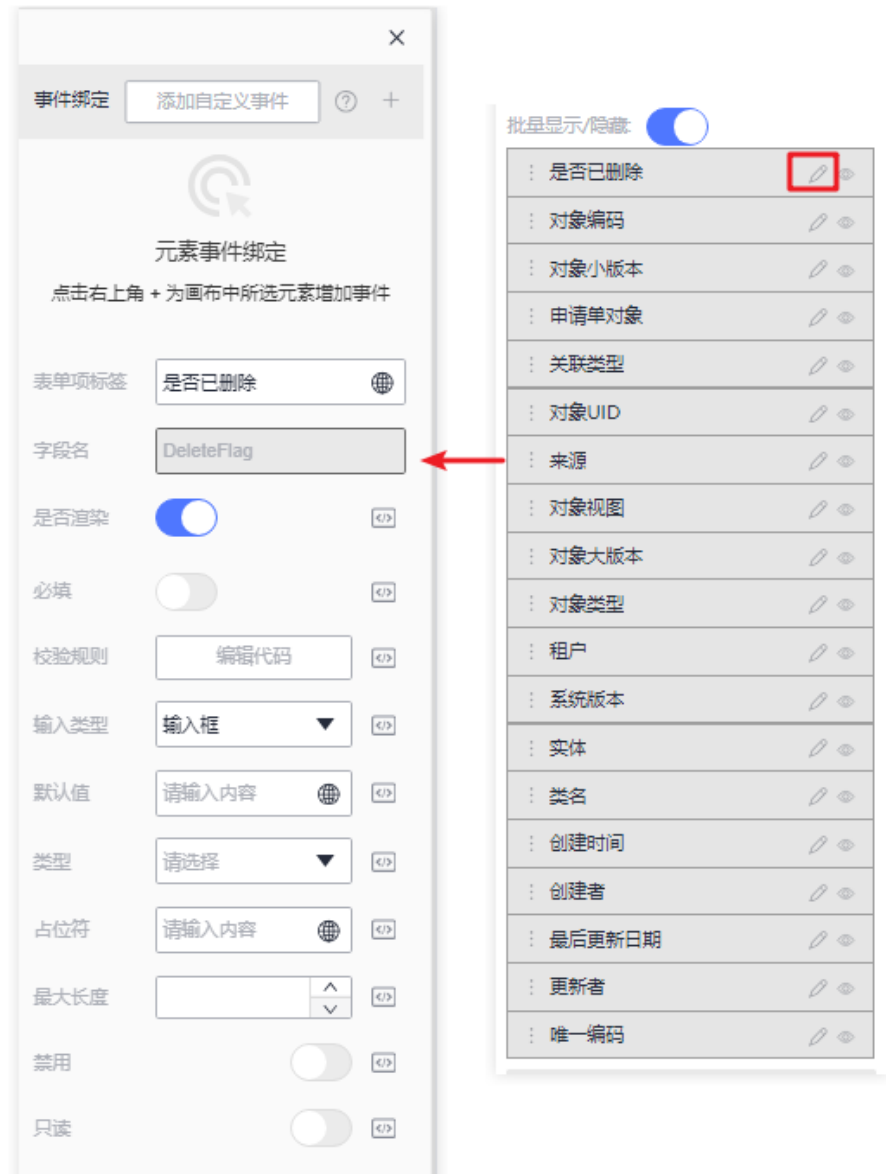


表 5-48 表单项属性


序号	属性名	说明	设计器类型	可选值	默认值	支持变量绑定
1	是否渲染	是否渲染该表单项	switch开关	开/关	开	√
2	必填	是否必填	switch开关	开/关	关	√
3	校验规则	表单项校验	代码编辑器	-	-	√

序号	属性名	说明	设计器类型	可选值	默认值	支持变量绑定
4	输入类型	控制该条表单项的类型	select输入框	详细见下	-	√
5	默认值	绑定默认值	input输入框	-	-	√
6	类型	与上级输入类型联动	select输入框	根据输入框类型确定	-	√
7	占位符	用于提示信息	input输入框	-	-	√
8	最大长度	限制表单项的输入长度	number输入框	0以下不可填	-	√
9	禁用	是否禁用	switch开关	开/关	关	√
10	只读	是否只读	switch开关	开/关	关	√

5.6.4.4.2 表格

使用场景

表格用来规整数据，将数据以表格的形式合理化展示。

组件图标：

基本属性

表 5-49 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	选择模型	用于选择数据模型	模型选择组件	-	-	√
2	只读	只读只显示数据，不可编辑操作表格	switch开关	开/关	关	√
3	启用添加	新增添加按钮，可以添加新一行	switch开关	开/关	关	√

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
4	是否允许复选框选中数据	可通过勾选复选框，选择多条数据	switch开关	开/关	关	√
5	表格高度	手动设置表格高度	数字选择器	-	-	√
6	表格最大高度	限制表格的最大高度	数字选择器	-	-	√
7	默认排序	设置字段，通过写一个Object，根据对象定位表格	input输入框	-	-	√
8	测试数据	提供表格的测试数据	代码编辑器	-	-	√
9	表格数据	提供表格的默认数据	代码编辑器	-	-	√
10	查询数据的方法	获取表格数据的方法	代码编辑器	-	-	√
11	分页配置	配置表格的分页功能	代码编辑器	-	-	√
12	表格列	选中模型后展示全部的表格列	metaArrayList	-	-	√
13	行操作设置	设置对单个行的操作	select选择器	-	-	√
14	自定义行操作	新增操作列，对行可新增自定义操作列	metaCollapse	-	-	√
15	表格列最小宽度	设置表格列的宽度最小值	数字输入框	-	-	√
16	表格列宽度	设置表格列的宽度	数字输入框	-	-	√
17	启用折叠	新增折叠列	switch开关	开/关	关	√
18	树形表字段	从表格数据中选择	select选择器	-	-	√
19	折叠表格映射	选择需要折叠的某一行或某一列	代码编辑器	-	-	√
20	默认展开所有行	默认展开表格的每一行设置	switch开关	开/关	关	√
21	展开的行id数组	设置想要折叠某一行的数据，通过id设置	代码编辑器	-	-	√
22	渲染优化	优化表格数据渲染	代码编辑器	-	-	√

属性详解

表格数据

表格数据写成JSON格式，key对应每一行的字段value。

表格列

选择表格模型后，显示表格列数据，支持批量显示/隐藏功能、编辑表格某一列、显示/隐藏某一列。

表 5-50 表格列属性

序号	属性名	说明	设置器类型	可选值	默认值
1	列标题	列标题列文字	input输入框	-	-
2	列字段	列的标识字段	input输入框	-	-
3	列宽	设置每一列宽度	数字编辑器	大于0的整数	48
4	最小列宽	设置列最小宽度	数字编辑器	大于0的整数	-
5	格式化方法	格式化表格列的方法	代码编辑器	-	-
6	文本溢出显示 tooltip	文本溢出是否显示tooltip	switch开关	关	-
7	列固定	列固定的方位	select选择器	-	-
8	是否排序	是否自动对表格进行排序	switch开关	关	-
9	表头tip	表头提示按钮显示	switch开关	关	-
10	表头tip内容	内容的文本设置	input输入框	-	-
11	是否可编辑	设置这一行数据是否可编辑	switch开关	关	-
12	必填	是否必填	switch开关	关	-
13	控件类型	设置表格某一列是什么类型	select选择器	-	输入框
14	默认值	设置表格列的默认值	select选择器	-	-
15	类型	根据控件类型变换展示类型	select选择器	根据控件类型变换	-

高级事件


表 5-51 高级事件

序号	事件名	触发条件	说明
1	onSortChange	排序条件变化事件	表格顺序发生变化，也可在删除行时触发
2	onExpandChange	表格行展开事件	展开一行表格时触发回调的事件
3	onUpdate:modelValue	绑定值改变时触发	表格绑定的表格数据发生变化时触发

5.6.4.4.3 树形控件

使用场景

将JSON数据以树的形式进行展示，可做单选多选的操作。

组件图标：

基本属性

表 5-52 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	数据唯一标识	树节点唯一标识	input输入框	-	-	√
2	父子关联选中	是否启用关联选中	switch开关	开/关	关	√
3	多选	是否启用多选	switch开关	开/关	关	√
4	显示虚线	是否显示虚线	switch开关	开/关	关	√
5	默认全部展开	是否默认全部展开	switch开关	开/关	关	√
6	最大高度	树的最大高度	数字输入框	-	大于0的整数	√

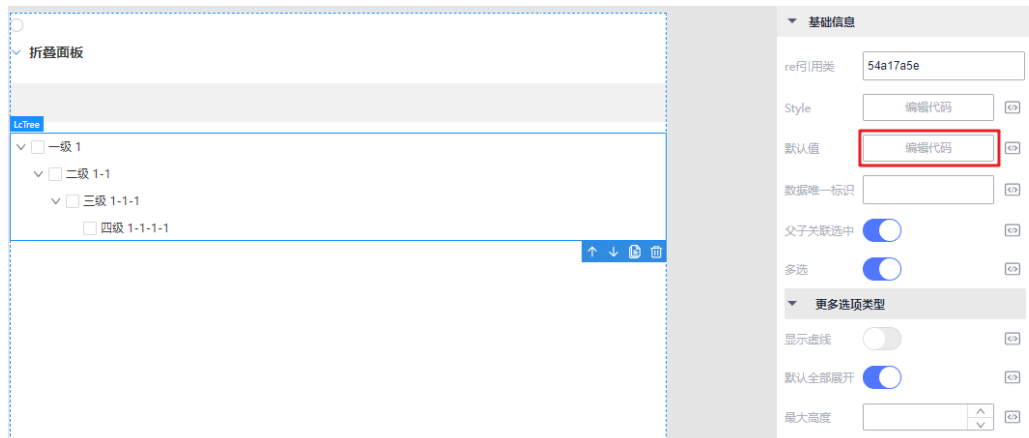
属性详解

默认值

以JSON格式设置树形控件的值，也可通过绑定变量的方式设置。

```
[
  {
    "label": "一级 1",
    "children": [
      {
        "label": "二级 1-1",
        "children": [
          {
            "label": "三级 1-1-1",
            "children": [
              {
                "label": "四级 1-1-1-1"
              }
            ]
          }
        ]
      }
    ]
  }
]
```

图 5-127 树形展示



高级事件

表 5-53 高级事件

序号	事件名	触发条件	说明
1	onCheck	多选节点选中变化触发	多选节点选中变化触发
2	onNodeClick	节点被单击时触发	节点被单击时触发

5.6.5 流程模板

5.6.5.1 提交记录模板

使用场景

用于提交页面，且可自定义删除或创建拖拽组件，自定义创建提交模板。

内置组件包含插槽（用户可拖拽任意组件）、上传附件、设置参与者、评审记录、底部操作按钮（提交、保存草稿、删除草稿）。

基本属性

表 5-54 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	页面方法触发	发送提交api请求的同时，还会发送“绑定API”里的api请求	绑定API编辑器	-	-	√
2	自定义撤销Api绑定	撤销不想要的API请求	绑定API编辑器	-	-	√
3	使用插槽	使用插槽，底部按钮操作区隐藏，转换成展示插槽	switch开关	开/关	关	√
4	绑定全局变量	默认绑定store全局变量	-	-	默认绑定stores	-

属性详解

使用插槽

打开“自定义操作区”，下方按钮全部隐藏，画布底部变成可拖拽组件的插槽。

图 5-128 使用插槽



5.6.5.2 评审页面模板

使用场景

提取电子流评审页面常用组件，封装为模板，便于用户快速搭建评审页面。

内置组件包含插槽（用户可拖拽任意组件）、评审三件套、评审记录、底部操作按钮（提交）。

基本属性

表 5-55 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	页面方法触发	发送提交api请求的同时，还会发送“绑定API”里的api请求	绑定API编辑器	-	-	√
2	自定义撤销Api绑定	撤销不想要的api请求	绑定API编辑器	-	-	√
3	使用插槽	使用插槽，底部按钮操作区隐藏，转换成展示插槽	switch开关	开/关	关	√
4	绑定全局变量	默认绑定store全局变量	-	-	默认绑定stores	-

属性详解

使用插槽

打开“自定义操作区”，下方按钮全部隐藏，画布底部变成可拖拽组件的插槽。

图 5-129 使用插槽



评审结论获取方法

获取评审结论的方法已写到JS帮助文档中，使用时直接调用此函数就可以获取。

```
function getConclusion(event) {  
  // 所有节点信息  
  const nodeList = this.utils.lcUtils().usePageRootInject().pageRoot.resolveDataNodes();  
  // 评审信息节点  
  const reviewNodeData = nodeList.find(({ dataNodeType }) => dataNodeType ===  
'assessment_trilogy').getInstance().getRuntimeData()?.[0];  
  // 评审意见  
  const opinion = reviewNodeData.opinion;  
  // 评审结论  
  const conclusion = reviewNodeData.formValue.routerIdAry.filter(({ id }) => id ===  
reviewNodeData.formValue.conclusion)?.[0]?.routerNameEn;  
  console.log(conclusion);  
  console.log(opinion);  
  return conclusion;  
}
```

5.6.5.3 流程页面模板

使用场景

提取电子流，流程记录页面常用组件，封装为模板，便于用户快速搭建流程记录页面。

流程信息、当前任务节点、历史转他人处理记录、评审记录，用户不能修改或删除。

基本属性

表 5-56 基本信息

序号	属性名	说明	设置器类型	可选值	默认值	支持变量绑定
1	页面方法触发	发送提交api请求的同时，还会发送“绑定API”里的api请求	绑定API编辑器	-	-	√
2	查询接口	流程记录页面信息的查询流程定义	流程定义服务	-	-	√
3	绑定全局变量	默认绑定store全局变量	-	-	默认绑定stores	-

5.6.6 属性面板设置

5.6.6.1 属性面板

属性面板

当用户选中某一组件时，右侧组件属性设置面板，会显示当前所选组件相关的可配置属性。若修改组件属性配置，将在保存当前页面后生效。

一些样式相关的修改，如修改组件的css style、组件显示的文本等，将会在修改后实时的在画布中体现。

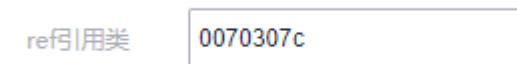
通用属性

通用属性，所有的组件都会暴露出的属性。

ref引用类

组件拖入画布时会自动生成，类似于组件的ID值（唯一编码）。

图 5-130 ref 引用类

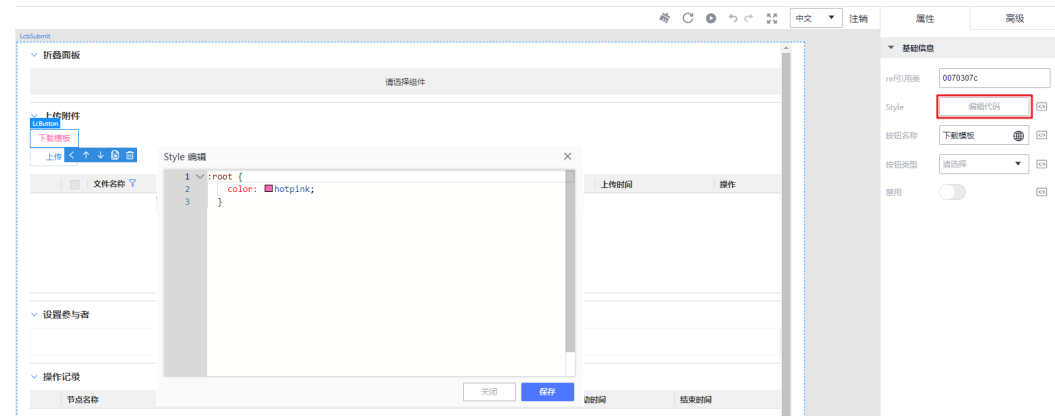


Style（元素CSS样式）

单击“编辑代码”弹出代码编辑框，编辑好代码后保存，即可给当前选中的组件元素添加行内样式（需要用户具备前端相关知识）。

高代码功能，当前组件元素的CSS行内样式，只作用于当前组件的最外层根元素，对内层元素无效。

图 5-131 编辑代码



5.6.6.2 设置器

设置器介绍

当选中组件时，右侧属性设置面板会显示出当前所选组件暴露出的全部属性，每条属性包括三个部分：属性标题、属性设置器及绑定变量。

属性设置器提供设置属性值的功能，根据属性值类型的不同，提供的属性设置器也会不同。

图 5-132 设置器



输入框

输入框一般出现在给组件配置文本标题的场景中，输入类型为文本。



数字输入框

数字输入框为输入框的变形，限定输入框只能输入数字，输入类型为数字，单击右侧的上下按钮，可以使数字+1或-1。



下拉选择框

下拉选择框会提供一些可选项，只能从提供的选项中选择值。

图 5-133 下拉框



开关

开关提供一个二值选项，即布尔值True和False，True为开、False为关。

- 开状态:
- 关状态:

三合一输入框

三合一输入框提供三种不同类型值（文本、数字、布尔值）的输入方式，根据下拉选择框选择需要的输入方式。

图 5-134 三合一输入框



多语言绑定

直接在输入框内输入内容，作用同输入框。


单击右侧  按钮后，弹出多语言绑定面板，支持绑定现有多语言项或新增多语言项，也可打开左侧多语言插件面板对所有多语言项进行管理。

图 5-135 多语言绑定

请输入内容

请根据显示值搜索

+ 创建新的多语言文案

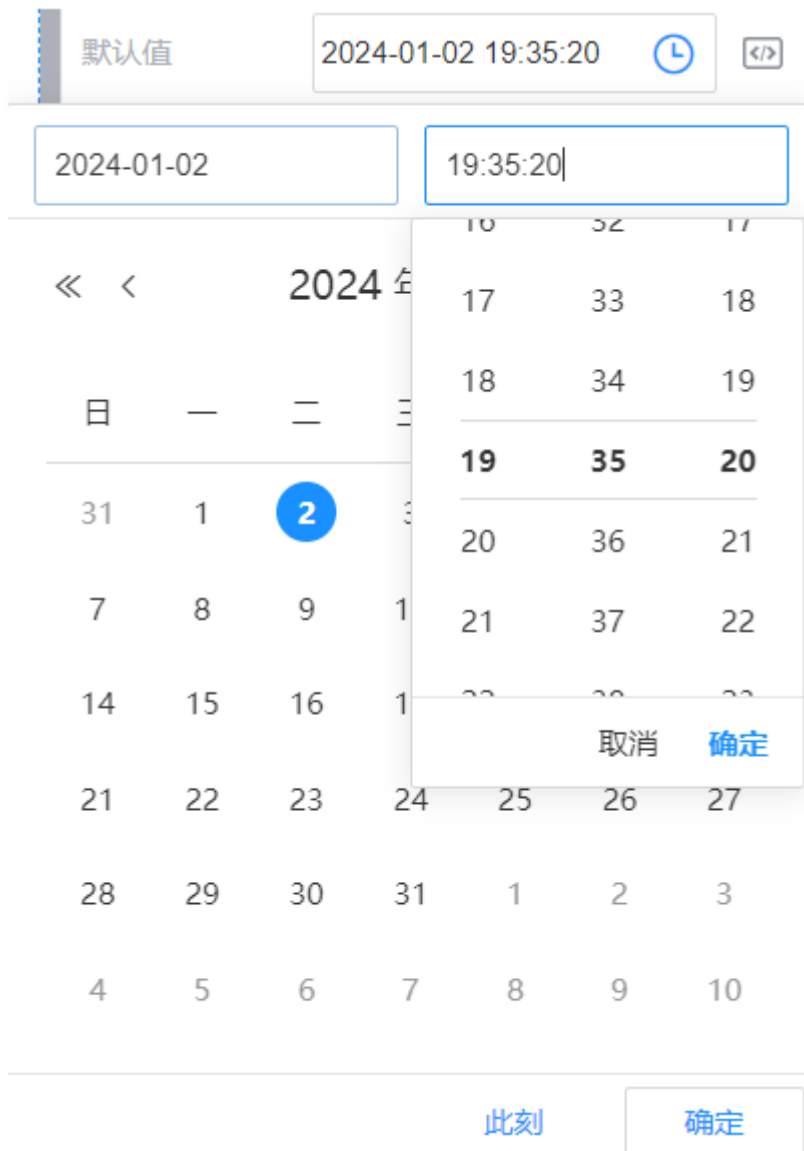
唯一键	lowcode.dc5564c8	
zh_CN		
en_US		

国际化管理 添加并关联

日期时间选择器

日期时间选择器提供选择日期和时间的能力。

图 5-136 时间选择器



折叠面板项

单击右上角+号按钮，新增一项折叠项，单击左下方“删除选项”按钮，删除当前折叠项。

用于多选、单选、信息树、设置参与者、评审三件套等组件属性配置中。

图 5-137 折叠面板



折叠子面板


用于表单、表格、添加部件等需要配置xdm模型的组件中。

最左侧为拖拽按钮，按住即可改变子项的顺序，右侧为编辑按钮（弹出子面板），显示隐藏按钮（控制模型项的显示和隐藏）。

图 5-138 折叠子面板



绑定变量

当一条属性支持配置变量绑定时，设置器的右侧会多出一个  按钮，单击按钮会弹出“变量绑定”窗口，支持多种类型的绑定。


绑定成功后，设置器会被带有变量名的蓝色背景所覆盖，若想取消绑定，需再次单击  按钮，单击“移除绑定”移除对变量的绑定。

图 5-139 绑定变量



5.6.7 高级面板设置

条件渲染

在实际的页面业务开发中，可能需要根据某些条件动态显示或隐藏页面中的内容，如：当用户已经登录的时候，显示欢迎登录的文本，未登录的时候，显示请登录的文本。那么就可以在组件上面绑定条件渲染，单击“绑定变量”，选择变量（如：state.isUserLogin），组件就会根据变量（state.isUserLogin）的布尔值来进行渲染。

图 5-140 条件渲染



循环渲染

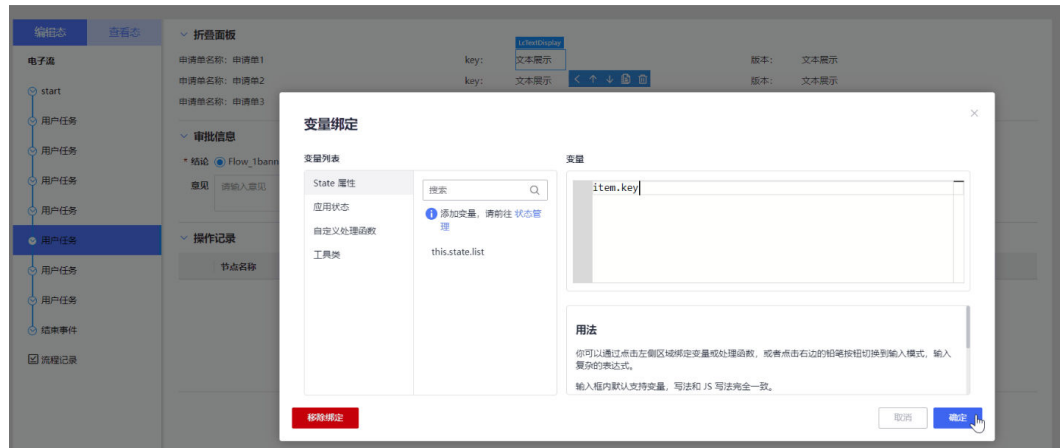
页面可能有许多重复的、动态生成的内容，如：列表页、表格数据等，这时候，我们就需要用到循环渲染。

可以在高级面板中指定循环数据绑定的变量、迭代的变量名、索引变量名、以及唯一的key。

举例：假如状态变量中有一个申请单列表list，我们希望渲染出申请单的名称、申请单Key以及版本，那么我们可以使用循环渲染来实现。

```
// 申请单list 实例数据
const list = [
  {
    "name": "申请单1",
    "key": "Process1",
    "version": "1"
  },
  {
    "name": "申请单2",
    "key": "Process2",
    "version": "2"
  },
  {
    "name": "申请单3",
    "key": "Process3",
    "version": "3"
  }
]
```

图 5-141 循环渲染



相关属性


- 循环数据，即需要循环渲染的数组，在这里是state.list。
- 迭代变量名，在循环渲染子项对应的变量名，默认为item。
- 索引变量名，循环渲染的索引变量名，默认为index。
- key，标识唯一的key，默认为index。

5.6.8 国际化词条

5.6.8.1 添加国际化词条

在国际化插件中添加

添加单个词条

步骤1 选择编辑器左侧插件面板中的国际化插件.

步骤2 单击“新增词条”，输入“zh_CN”（中文显示值）和“en_US”（英文显示值）。


图 5-142 手动添加词条



步骤3 单击“确定”。

----结束

批量添加词条

步骤1 选择编辑器左侧插件面板中的国际化插件.

步骤2 单击“下载导入模板”，修改压缩包中的json文件。

图 5-143 批量添加词条



步骤3 单击“批量上传”，上传修改后的文件。

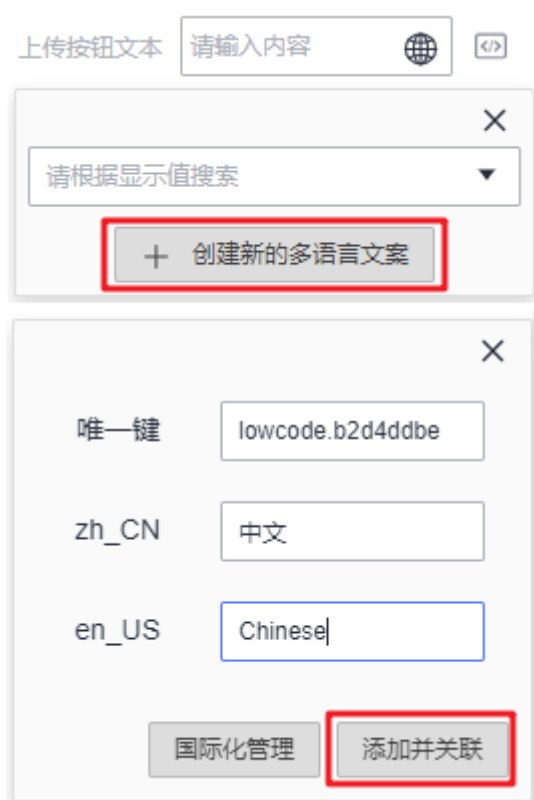
----结束

在多语言绑定器中添加

步骤1 选中一个支持多语言属性绑定的组件（这里用按钮举例）。

步骤2 单击右侧的, 打开多语言绑定面板，单击“创建新的多语言文案”。

图 5-144 添加国际化词条



步骤3 输入中英文值，单击“添加并关联”。

----结束

5.6.8.2 绑定国际化词条

在多语言绑定器中绑定

步骤1 选中一个支持多语言属性绑定的组件（这里用按钮举例）。


步骤2 单击右侧的，打开多语言绑定面板，在下拉框中搜索并选择在国际化插件中，已经新增好的国际化词条。

图 5-145 绑定国际化词条



----结束

在变量绑定中绑定

国际化词条不仅可以在支持多语言绑定器中实现绑定，也支持在“变量绑定”中绑定国际化词条。

如文本展示组件的文本内容属性，单击变量绑定按钮弹出变量绑定面板，在“state”属性中填写 `'lowcode.5dfa1b4e'`，其中 `lowcode.5dfa1b4e` 为国际化词条的键值（key）。

图 5-146 绑定词条-1



图 5-147 绑定词条-2



5.6.8.3 管理国际化词条

编辑词条

单击词条项右侧的编辑按钮即可编辑已有词条的相关属性。

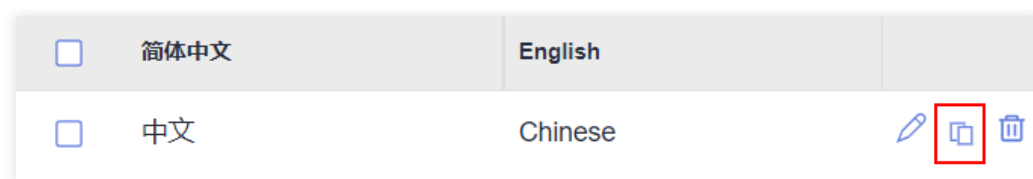
图 5-148 编辑词条



复制词条键值

单击复制按钮, 即可将选中词条的键值复制到剪切板。

图 5-149 复制键值



删除词条

图 5-150 删除词条



- 单个删除：单击词条项中的删除按钮，在确认框中单击“确认”，即可单个删除。
- 批量删除：勾选需要删除的词条，单击“批量删除”，在确认框中单击“确认”，即可批量删除。

5.6.9 页面 JS 及事件绑定

5.6.9.1 页面 JS 基本使用与管理

编写 JS 脚本



单击左侧插件栏中页面JS按钮  即可打开页面JS面板。在代码编辑区中编写好代码后单击右上角保存按钮 。

图 5-151 代码编辑区



系统预先编写好了一些范例脚本，单击“展示使用帮助”可以查看提供的范例脚本。可以将需要使用的相关范例复制，然后单击“关闭使用帮助”回到JS代码编辑页面，粘贴到代码编辑区进行使用。

给组件绑定 JS 方法

步骤1 在页面JS中添加脚本并保存。

```
function exampleFunction() {  
  // JS逻辑  
}
```

步骤2 选中画布中已有的组件元素。

步骤3 在右侧“高级”设置面板中，鼠标移动至添加按钮上，弹出可添加事件的菜单栏（若没有可配置的事件，则菜单栏为空），选择需要绑定的事件。

图 5-152 绑定事件



步骤4 在事件栏中鼠标移至添加按钮上，弹出菜单栏，选择“自定义JS”。

步骤5 事件绑定中选择在页面JS中新增的**exampleFunction**方法（或添加新方法）。

图 5-153 自定义 JS




步骤6 单击“确定”

----结束

给属性绑定 JS 方法

步骤1 选中画布中已有的组件元素。

步骤2 在右侧“属性”设置面板中，选择一个支持变量绑定的属性，单击变量绑定按钮 ，弹出“变量绑定”页面。

步骤3 在“变量列表”选择“自定义处理函数”，并选择在页面JS中新增的方法。

若需要执行JS方法，需要在变量中给方法名后添加括号，否则为字符串格式。

图 5-154 自定义处理函数



步骤4 单击“确定”。

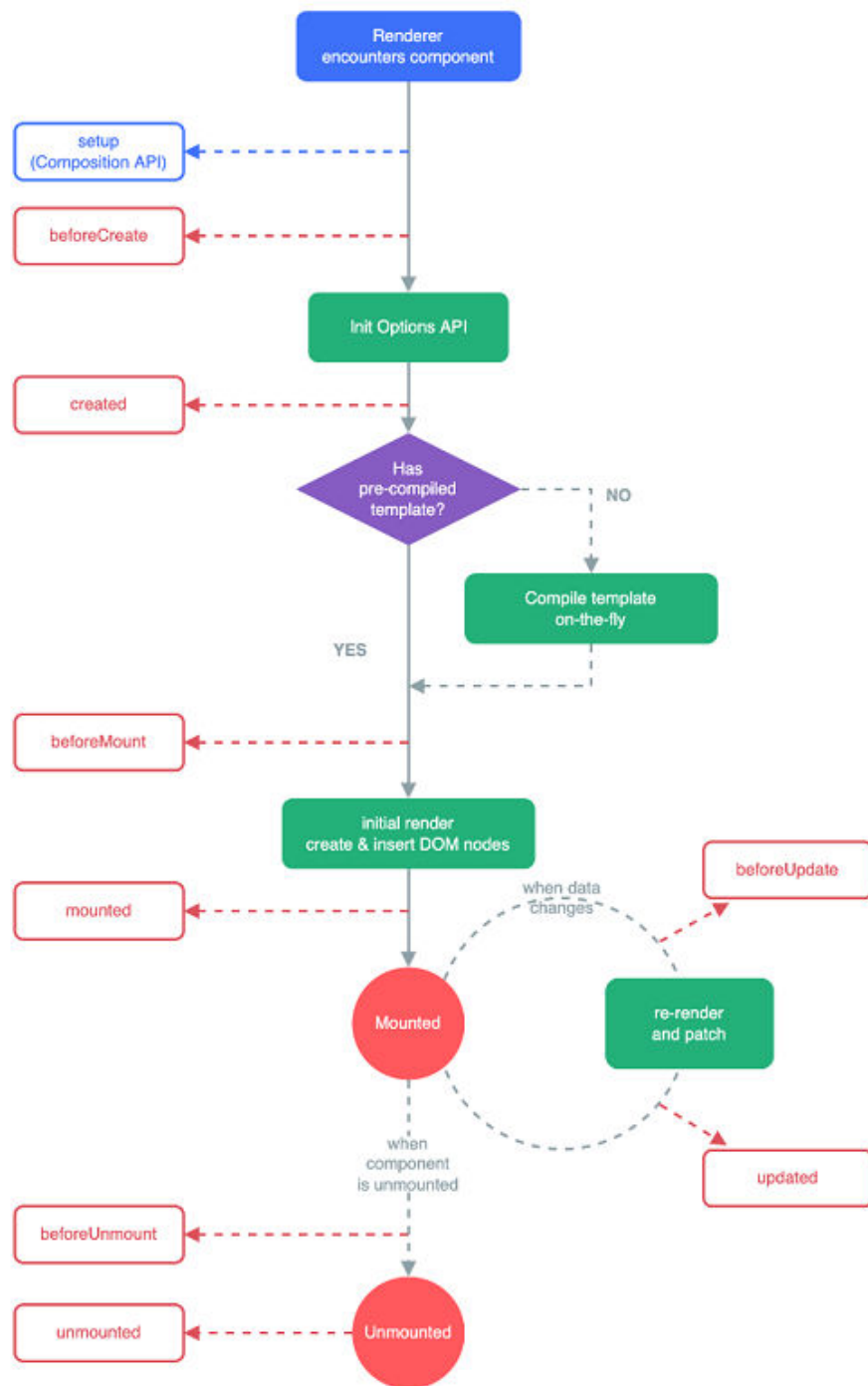
----结束

在生命周期中绑定 JS 方法

生命周期介绍

这里的生命周期特指Vue的生命周期，Vue的生命周期可以简单分为四个阶段：创建阶段，挂载阶段，运行阶段，销毁阶段。

图 5-155 生命周期介绍



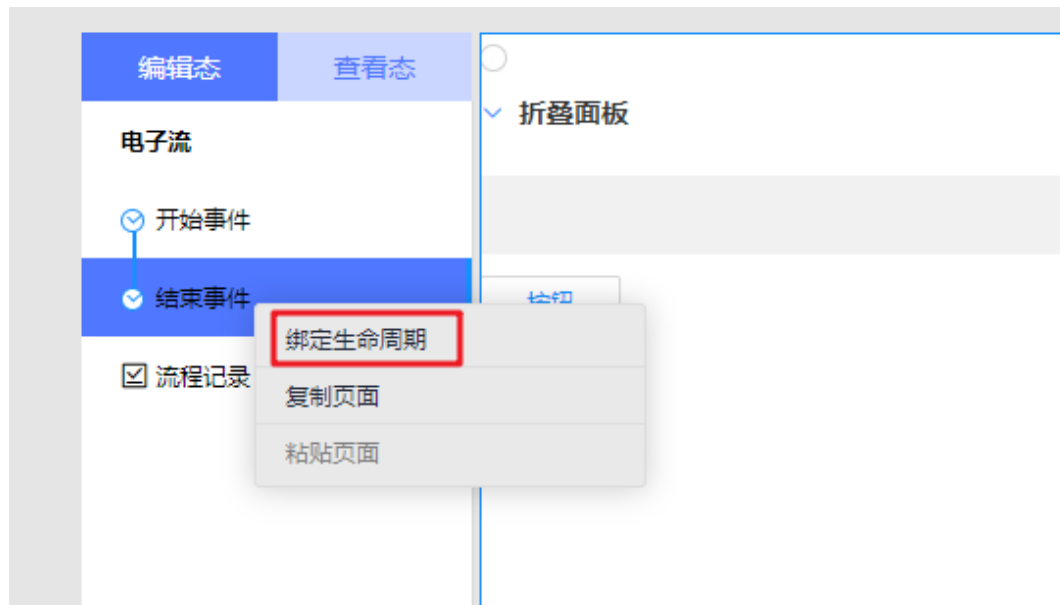
- setup: 在组件创建时执行。

- onBeforeMount: 在组件被挂载之前执行。
- onMounted: 在组件挂载完成后执行。
- onBeforeUpdate: 在组件更新之前执行。
- onUpdated: 在组件更新完成后执行。
- onBeforeUnmount: 在组件被卸载之前执行。
- onUnmounted: 在组件卸载完成后执行。
- onErrorCaptured: 在捕获了组件传递的错误时执行。
- onActivated: 当组件被插入到DOM中激活时执行。
- onDeactivated: 当组件从DOM中被移除时执行。

生命周期绑定JS脚本

步骤1 在导航树栏中，右键单击当前页面节点弹出菜单栏，单击“绑定生命周期”，弹出绑定生命周期面板。

图 5-156 绑定生命周期



步骤2 根据实际需要，编辑JS脚本，单击“确定”保存。

----结束

5.6.9.2 绑定 API

在服务定义中新增 API

请参考[服务定义](#)进行操作，“基本信息”中驱动方式选择“API”。

为事件绑定 API

在Workflow中的服务定义中创建好API后，即可回到UI编辑器页面完成对API的绑定。

步骤1 选中画布中已有的组件元素。

- 步骤2** 在右侧“高级”设置面板中，鼠标移至添加按钮上，弹出可添加事件的菜单栏（若没有可配置的事件，则菜单栏为空），选择需要绑定的事件。
- 步骤3** 在事件栏中鼠标移至添加按钮上，弹出菜单栏，选择“绑定API”弹出绑定API面板。

图 5-157 绑定 API



- 步骤4** 填写基本配置。

“接口地址”选择服务定义中需要绑定的API，并设置“请求类型”和“触发条件”。

图 5-158 绑定 API

字段	描述	是否必填	绑定类型	绑定字段
1	ObjUid	否	模型	请选择

字段	描述	是否必填	绑定类型	绑定字段
1	error	否	模型	请选择

步骤5 设置“入参配置”和“出参配置”。入参和出参支持绑定模型和变量。

- 模型：页面所有组件暴露出的配置项。
- 变量：包括变量(页面状态)、全局变量(应用状态)、流程变量。

步骤6 单击“确定”绑定API。

----结束

5.6.10 状态管理及变量绑定

5.6.10.1 状态管理

插件介绍

状态管理插件用于管理页面的状态（变量），状态可以在多处地方使用。

页面状态state和应用状态global都是响应式的数据，状态管理面板对响应式变量进行管理，包含添加、删除、搜索和编辑。

页面状态和应用状态的区别在于：

- 页面状态针对当前的页面（局部变量）。
- 应用状态针对当前应用的所有页面（全局变量）。

图 5-159 状态管理



“页面状态”和“应用状态”的基础操作（包括添加、删除、搜索、编辑）都是一致的，唯一的区别是，创建应用状态时初始值是必须赋值的不能为空（包括空字符串）。





状态管理

- 查看变量：单击左侧插件栏的 `var`，查看变量。当数据较多时，可在搜索框中输入关键字搜索查找。
- 添加变量：单击“添加变量”，填写“变量名”及“初始值”，单击“保存”。
- 修改变量：单击 修改变量。
- 复制变量：单击 复制变量。
- 删除变量：单击 删除变量。

图 5-160 变量管理



初始值上方操作区，功能按钮说明：

- 插入词条：单击，在初始值中引入国际化的词条（格式为JSON）。
- 插入JS函数：单击，给变量插入js方法（显示为object对象），可以在JS方法中通过`this.state.xxx()`或`this.global.xxx()`调用。
- 格式化：单击，格式化代码，格式化初始值，使其代码整洁美观。
- 全屏：单击，在页面中心区域内显示面板。

5.6.10.2 变量绑定及使用

将变量与组件绑定


步骤1 在画布中选中组件，然后在右侧属性设置面板中，选择需要绑定变量的配置项，单击变量绑定按钮，进入“变量绑定”页面。

图 5-161 绑定变量



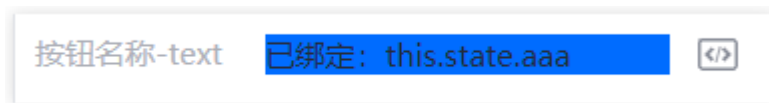
步骤2 选择左侧“State属性”（页面状态）或“应用状态”，在中间选择框中选择已添加好的变量，单击“确定”。

图 5-162 变量绑定



步骤3 绑定成功配置项会显示“已绑定”。

图 5-163 绑定成功



----结束

修改变量

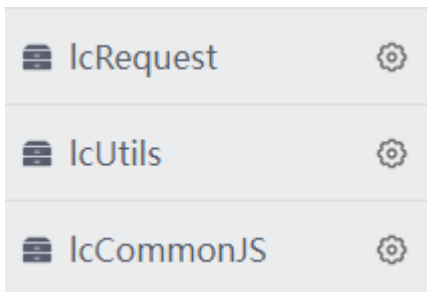
添加的状态通常是一个可以随不同状态有不同值的变量，如果想要改变它，可以在JS面板中通过`this.state.xxx`或`this.global.xxx`获取到页面，应用状态并且改变它。

5.6.11 资源管理

在常规代码开发中，通常会将一些高频率用到的代码片段抽离出来成为公共函数，减少重复的代码，从而达到代码复用的目的。同样，在低代码开发中，我们不可避免的需要编写一些高代码进行组合开发，这里同样存在存储一些公共函数以供各处调用需求。

工具类方法utils可以将一些可复用的公共函数编写到工具类方法中，供后续调用。如图5-164所示，UI编辑器内部已经提供了三个内置工具类供用户使用，下文以lcRequest举例说明。

图 5-164 内置工具



lcRequest

内置网络请求工具Axios，因为已添加拦截器和部分参数验证，所以在编辑器内发起请求时，可直接使用此工具。

详细参数类型及使用方法，可参考Axios使用文档，这里给出基础使用方法。

获取请求实例

```
const request = this.utils.lcRequest();
```

配置请求参数

```
// 请求接口Url地址
const url = 'http://www.example.com/api';
// 请求方法 ( get | post | put | delete )
const method = 'get';
// query参数, 用于get请求
const params = {
```

```
    param1: 'xxx',
    param2: 'xxx',
  }
  // body请求体, 用于其他类型的请求, 可包含数组, 对象等复杂类型
  const body = {
    param1: 'xxx',
    param2: 'xxx',
    arrayParam: ['arrayItem1', 'arrayItem2'],
    objectItem: {
      objectItem1: 'xxx',
      objectItem2: 'xxx'
    }
  }
  // headers请求头, 用于自定义请求的请求头参数
  const headers = {
    // 'Accept': 'application/json'
  }
```

发起get请求

```
// get http://www.example.com/api?param1=xxx&param2=xxx
request.get(url, { params, headers }).then(res => {
  // 调用成功后的逻辑, res为调用成功后的反参
  // ...
}).catch(err => {
  // 调用失败后的逻辑, err为调用失败后的错误信息
  // ...
})
```

发起post请求

```
// post http://www.example.com/api
request.post(url, body, { headers }).then(res => {
  // 调用成功后的逻辑, res为调用成功后的反参
  // ...
}).catch(err => {
  // 调用失败后的逻辑, err为调用失败后的错误信息
  // ...
})
```

5.7 电子流应用

流程应用

步骤1 打开浏览器访问如下地址。

`http://{域名}/pdmcore/ipdcworkflowservice/console/ui/definition_micro/
{example1}/{example2}/index.html`

其中, **example1**为流程的英文名称, **example2**为版本号。

步骤2 按照定义好的流程顺序, 填写流程信息及审批人。

步骤3 各节点审批通过后, 流程结束。

📖 说明

流程申请人可以在过程中关闭流程, 或撤回重新填写流程。

----**结束**

A 修订记录

发布日期	修订记录
2024-05-24	新增《模板卡片配置指南》。
2024-03-29	《统一认证配置指南》中增加 基于OrgID的单点登录配置 ，更新了外部集成开发的操作。 《流程配置指南》快速入门中，增加了流程定义的操作示例。
2024-02-26	《流程配置指南》 UI编辑器使用帮助 中，为插件、组件、设置面板等功能，增加了详细的使用介绍。
2023-12-29	第一次正式发布。