

数字孪生可视化网页版 RayData Web 实践教学



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

实践教程

千人千面的项目效果

输入框搜索二维标记的使用

时间范围选择器筛选符合时间下的图表

表单筛选及下载数据的使用

下拉框查找并定位三维标记的使用

表格单元格给其他组件进行传值

搜索时动态展示卡片

表格二级联动

实践教程

千人千面的项目效果

最近更新时间：2025-05-20 15:51:02

操作场景

⚠ 注意：

基础版不提供该功能。

系统常量是产品开放出来的一些参数，主要是用户相关数据，可在接入数据或跳转外链时引用，以达到千人千面的项目数据查看效果。

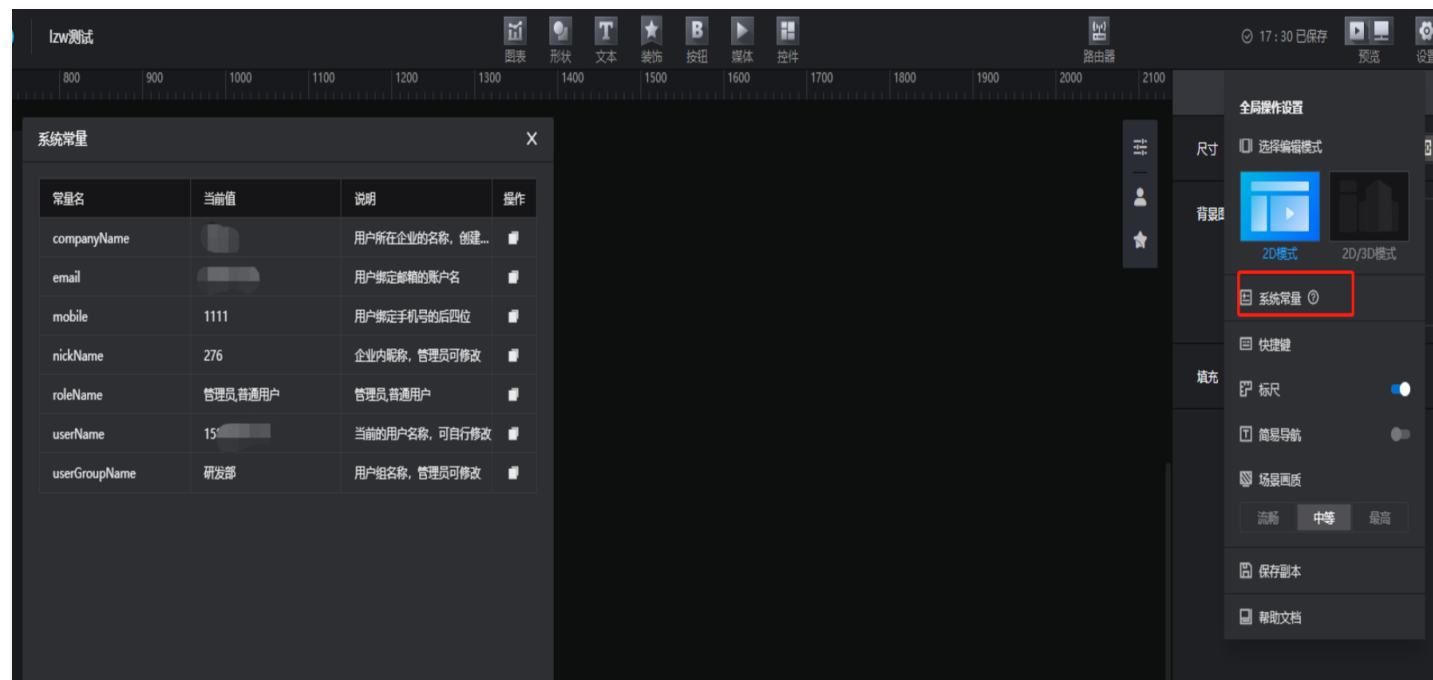
实现效果

不同用户登录后浏览项目时能看到不同数据效果。

[观看视频](#)

功能介绍

目前已开放的系统常量，您可在项目编辑页面 > 设置 > 系统常量进行查看。

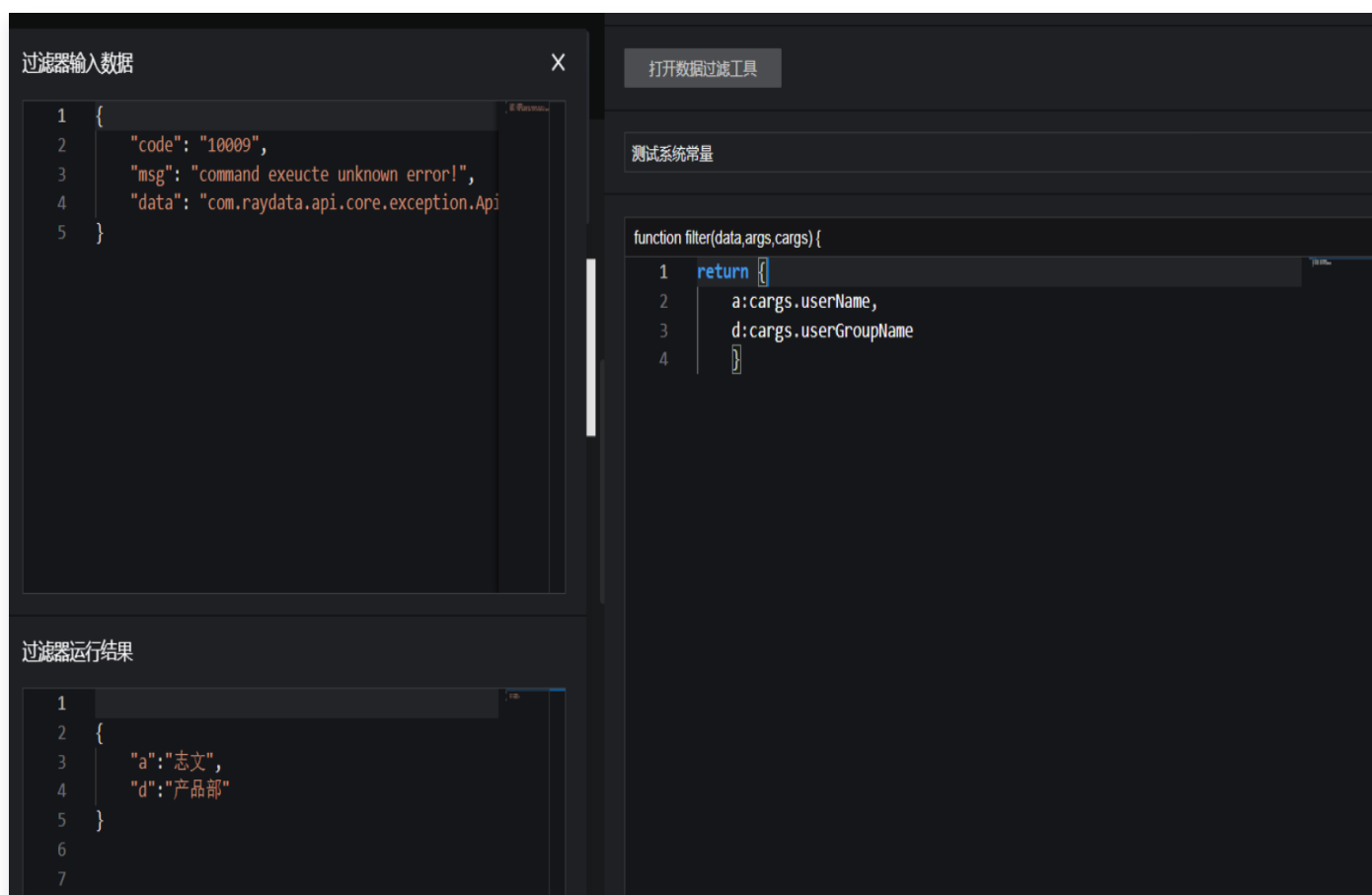


可引用系统常量的地方包括：

- **方式一：**SQL 语句、API/DAYU 数据服务的 URL 地址/请求头/body 参数、图片/视频/视频流（标记）和跳转外链时的地址。使用 “~{系统常量名}” 的格式引用，引用多个常量时中间用&连接。



- **方式二：**过滤器中的引用格式为 “cargs. 系统常量名”。

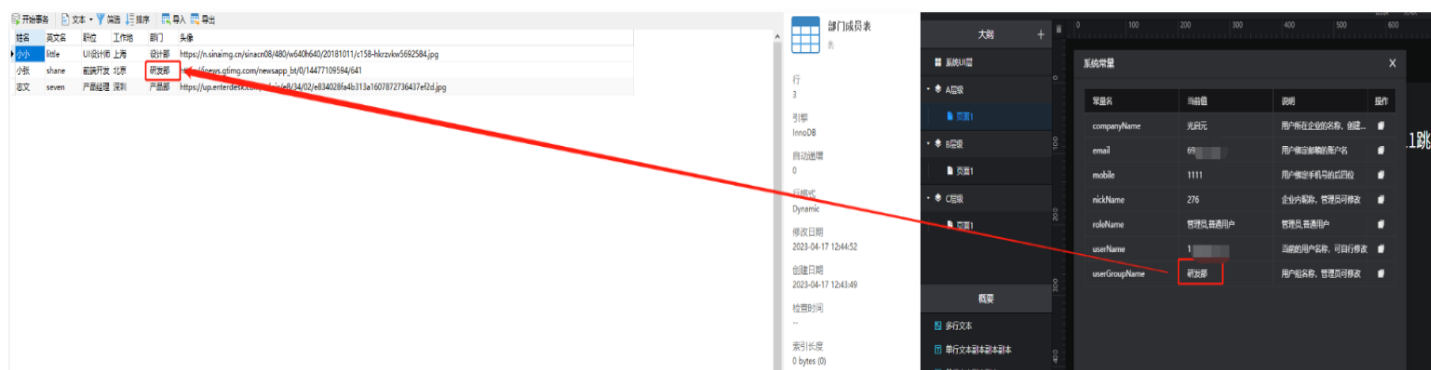


操作步骤

以 MySQL 数据库为例，演示如何进行配置逻辑，以能达到不同用户浏览项目时能看到不同数据效果。

❗ 说明：

数据源中的数据要和系统常量能够对应得上。



1. 配置数据路由器，将系统常量加到筛选条件中。

添加数据路由器

名称*

千人千面

备注

请输入内容（非必填）

数据源类型*

MySQL

选择数据源

选择已配置的数据源

数据配置

域名*

端口*

用户名*

密码*

.....

数据库*

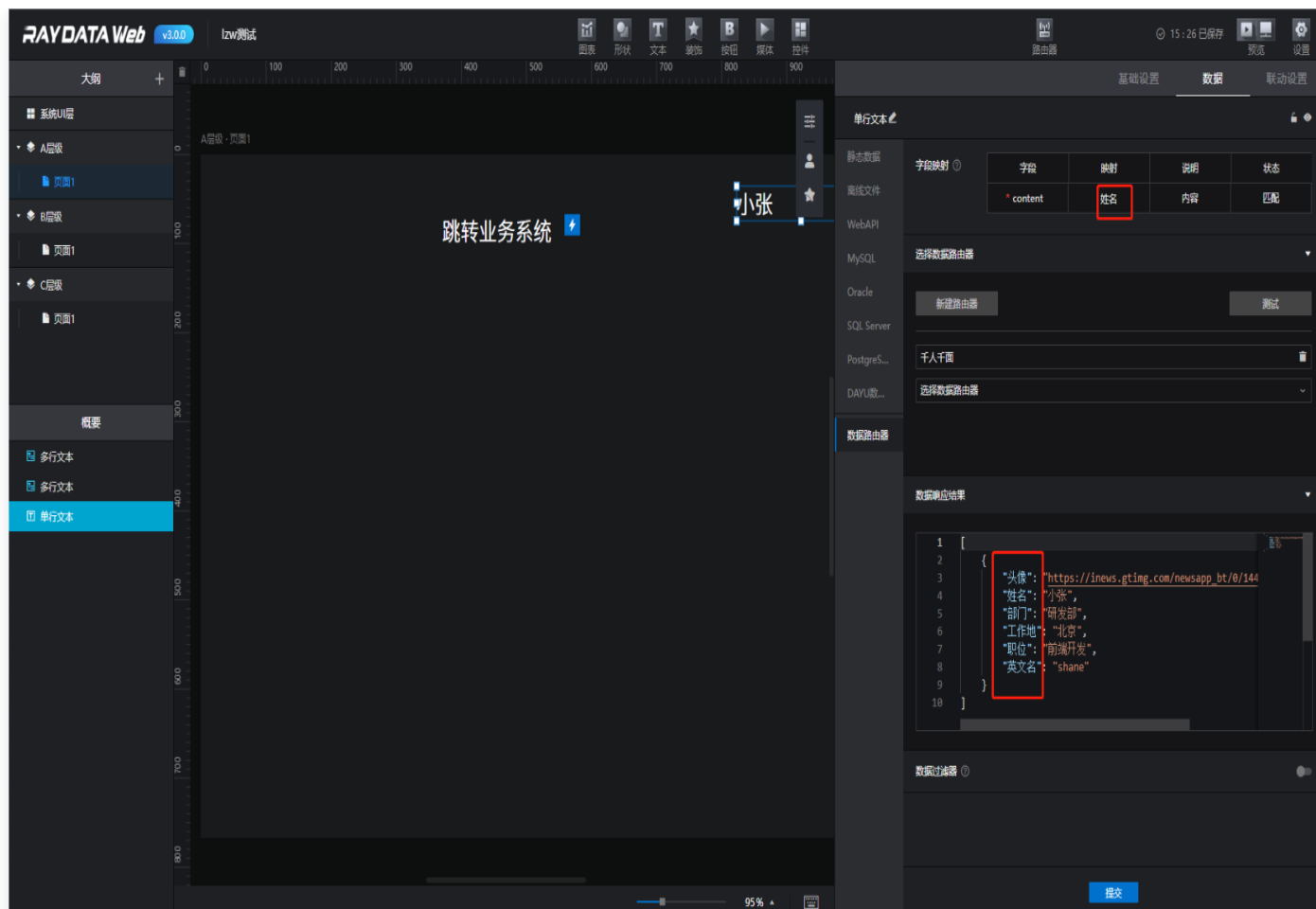
查询语句

1

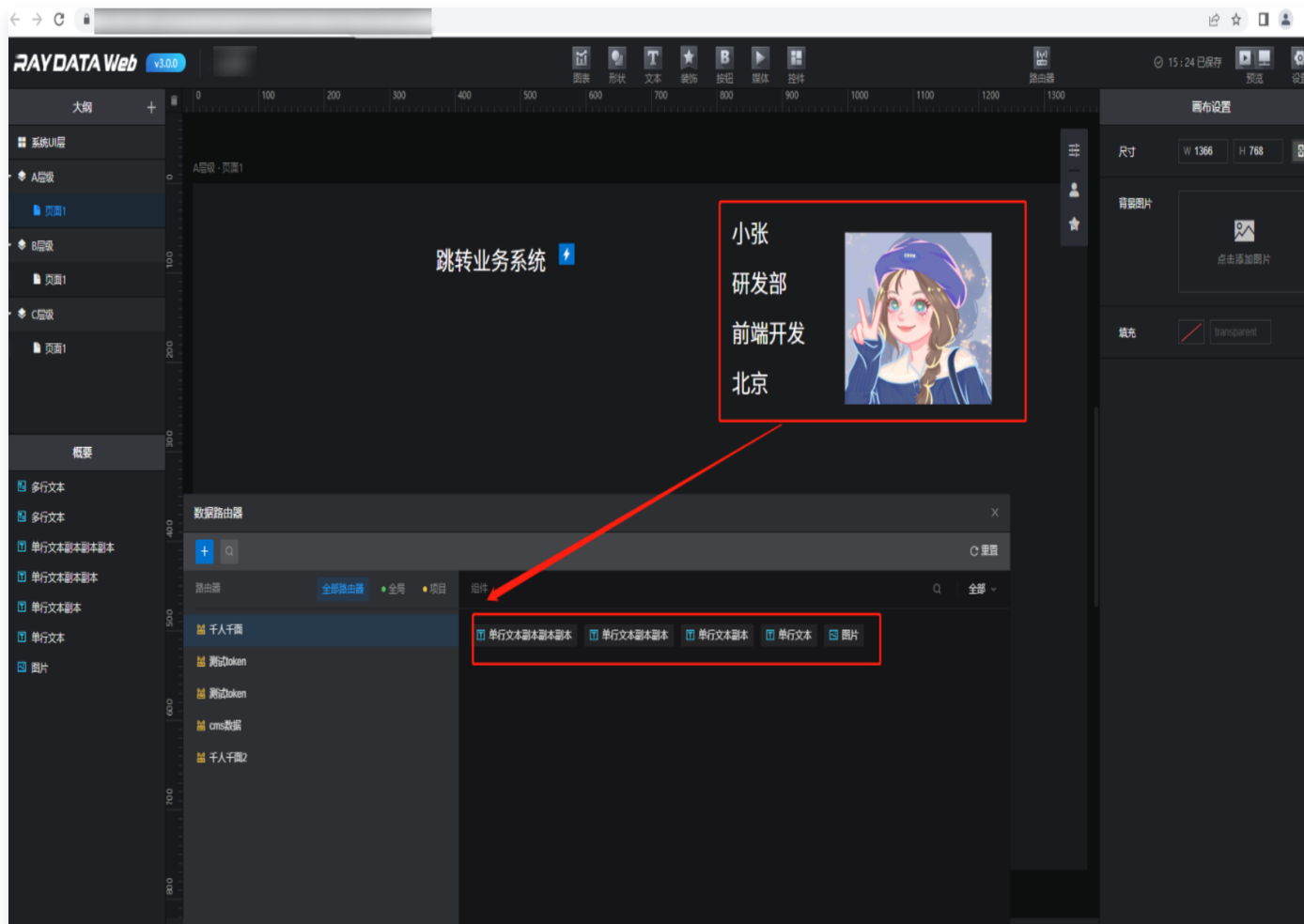
select * from 部门成员表 where 部门 = "{userGroupName}"

提交

2. 在组件上接入配置好的数据路由器，映射字段。



3. 在多个组件中使用该路由器。



4. 分享项目地址，不同的人登录后能查看对应效果。

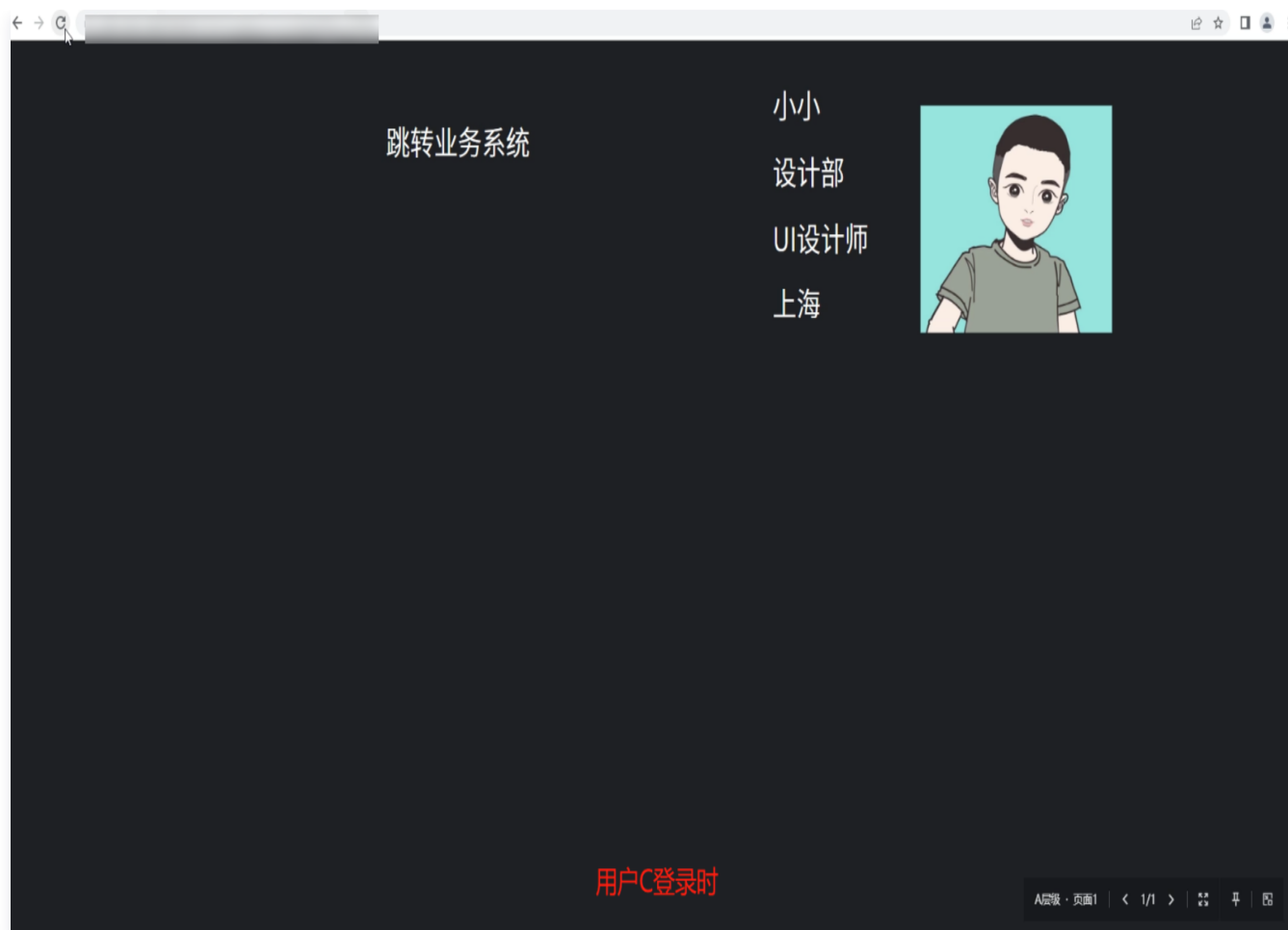
- 用户 A 登录时，界面效果如下：



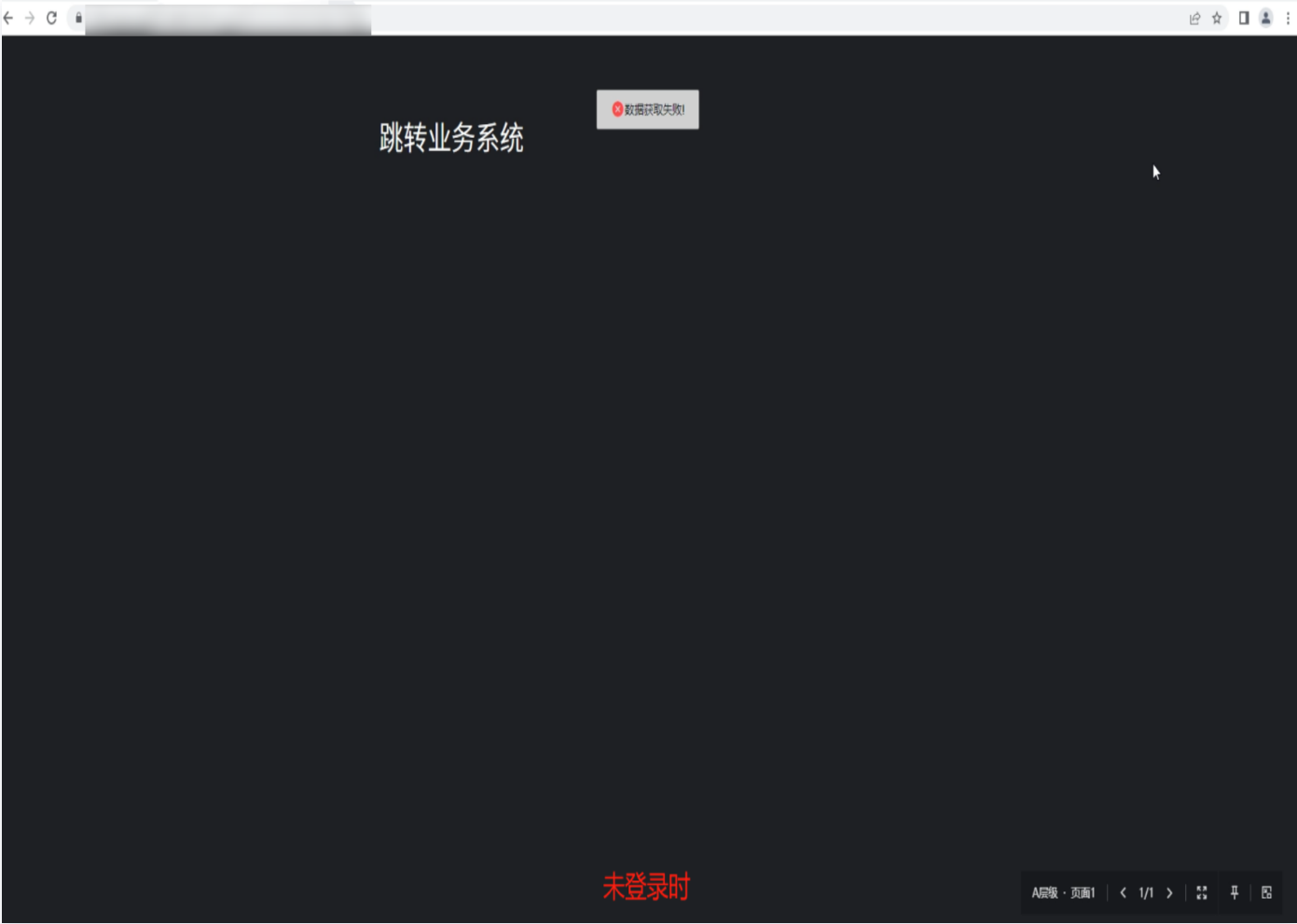
- 用户 B 登录时，界面效果如下：



- 用户 C 登录时，界面效果如下：



5. 当获取不到登录用户或信息或者无对应数据时，组件数据为空。



输入框搜索二维标记的使用

最近更新时间：2025-03-14 17:26:42

操作场景

输入框作为交互组件，可把输入的值通过变量传递给别的组件（下拉框、时间选择器均是类似用法，可套用）。

实现效果

输入框输入北京，则三维标记显示在北京，输入框输入成都，则三维标记显示在成都。

操作步骤

1. 管理中心添加数据源，格式选择离线文件，[单击下载示例数据](#) 后。将下载好的文件后上传至产品中。
2. 在编辑页面添加输入框组件，单击数据模块选择静态数据，无需填入任何值。
3. 场景选择中国版块，并添加三维标记图片标记，给图片标记添加刚上传的离线文件内容。

图片标记

静态数据

离线文件

模拟动态...

WebAPI

WebSock...

MySQL

Oracle

SQL Server

PostgreS...

达梦

InfluxDB

ClickHouse

TDSQL-C

DAYU数...

MQTT

数据路由器

数据类型

城市名

经纬度

字段映射

字段	映射	说明	状态
* cityName	cityName	城市	未匹配
title	title	标题	匹配
note	note	注释	未匹配
imageUrl	imageUrl	图片地址	未匹配

+

选择数据源

中国部分省份

解析

数据响应结果

表格视图

代码视图

	prov	provCapCity	acode	title	subTitle	area
1	北京市	北京	110000	北京市主标题...	北京市副标题...	1.64
2	天津市	天津	120000	天津市主标题...	天津市副标题...	1.19
3	上海市	上海	310000	上海市主标题...	上海市副标题...	0.63
4	重庆市	重庆	500000	重庆市主标题...	重庆市副标题...	8.2
5	河南省	郑州	410100	郑州市主标题...	郑州市副标题...	16.7
6	江西省	南昌	360100	南昌市主标题...	南昌市副标题...	16.6
7	浙江省	杭州	330100	杭州市主标题...	杭州市副标题...	10.5
8	四川省	成都	510100	成都市主标题...	成都市副标题...	10.5

变量挂载

数据过滤器

保存

4. 在变量挂载处，添加一个变量，变量名称设为 a，默认值可不填，假设不填，绑定该变量。

a

默认值

请输入默认值

数据源类型

无

路由器

WebSocket

描述

请输入描述

变量挂载

变量管理

▼

a、默认值:

添加变量

5. 在过滤器中写入简单语句，把变量 a 的值给到 city，只显示具体 city 下的数据内容。由于变量默认值为空，则无任何值给到 city，则经过过滤器后数据为空。
6. 给输入框组件，添加联动事件，设置完成后，单击**确定**。
 - 触发条件：选择**失去焦点时**。
 - 实现动作：选择**变量赋值**。让变量 a 与输入框字段 content 建立关系。
7. 单击**预览**按钮，在输入框中输入北京，鼠标失焦，则三维标记显示在北京，输入成都，则三维标记显示在成都。





❗ 说明:

上方为了方便用户体验完整流程，选择离线文件的数据类型来实现，也可使用其他类型的数据源。

时间范围选择器筛选符合时间下的图表

最近更新时间：2025-03-14 17:26:42

操作场景

时间范围选择器作为一个限定时间范围的交互组件，需要配合两个变量来实现对时间范围下内容的筛选。

实现效果

一个图表包含2008年-2016年的数据，通过时间范围选择器筛选，如开始时间选择2008年，结束时间选择2012年，能把图表2008到2012年的数据筛选出来并呈现出来。

操作步骤

1. 在管理中心添加数据源，格式选择离线文件，[单击下载示例数据](#)后，将下载好的文件后上传至产品中。
2. 在编辑页面添加时间范围选择器组件，单击数据模块选择静态数据，无需填入任何值。
3. 添加一个柱状图，给柱状图组件接入刚上传的离线文件。

基础柱状图

静态数据

离线文件

WebAPI

WebSock...

MySQL

Oracle

SQL Server

PostgreS...

DAYU数...

MQTT

数据路由器

字段映射 ?

字段	映射	说明	状态
* X	X	横坐标	匹配
* Y	Y	纵坐标	匹配

选择数据源

影院数据

解析

数据响应结果

表格视图

代码视图

```
{
  "cname": "类别总称",
  "series": [
    {
      "sid": 1,
      "name": "全国影院数量",
      "unit": "单位"
    }
  ],
  "data": [
    {
      "from": 1,
```

4. 在变量管理处，分别添加 a、b 两个变量，默认值可不填，由于本示例数据的年限范围为2008年–2016年，假设 a 的默认值填为2008年，b 的默认值填为2016，绑定该两变量。

☒ a、默认值: 2008年

☒ b、默认值: 2016年

5. 在过滤器中写入简单语句，让 $a \leq X \leq b$,只显示a与b时间范围下的数据内容。

```
data[0].data =data[0].data.filter(item => args.a <=
item.X&&args.b>=item.X)
return data
```

6. 给时间范围选择器组件，添加联动事件，设置成功后，单击确定。

- 触发条件：选择**时间范围变化时**。
- 实现动作：选择**变量赋值**。让变量 a 与时间范围选择器组件字段 startTime 建立关系，变量 b 与时间范围选择器组件字段 endTime 建立关系。

7. 单击**预览**按钮，在时间范围选择器开始时间选择2008年，结束时间选择2012年，则图表组件显示2008年-2012年的数据。





表单筛选及下载数据的使用

最近更新时间：2025-03-14 17:26:42

操作场景

表单筛选，常配合搜索框、下拉框或者时间选择等交互组件来对包含大量数据的表格或者图表内容进行筛选呈现，并且把筛选后的数据结果导出为本地文件。

实现效果

输入框查询姓名，下拉框选择工作地，时间选择器选择入职时间。

示例：可以实现输入框输入李四，下拉框工作地选择上海，入职时间选择2017年4月。

- 单击**查询**，则可以把满足这三个条件的李四的表格数据筛选出来。
- 单击**导出**，则可以把满足这几个条件下的数据，导出本地。

输入姓名

请选择工作地

请选择入职时间

查询

导出

员工编号	姓名	工作地	部门	岗位	入职时间
1000	成大	深圳	销售部	销售	2013年7月
1001	张三	北京	研发部	前端	2015年3月
1002	李四	成都	产品部	产品经理	2016年7月
1003	王五	北京	产品部	运营	2020年9月
1004	张一	上海	销售部	销售	2018年3月
1005	李四	上海	销售部	销售	2016年7月
1006	李四	上海	研发部	后端	2017年4月

李四

上海

2017年4月

查询

导出

员工编号	姓名	工作地	部门	岗位	入职时间
1006	李四	上海	研发部	后端	2017年4月

操作步骤

1. 在管理中心页面，单击**添加数据源**，格式选择离线文件，上传一个模拟的 Excel（例如xlsx 格式）的部门成员信息表。

员工编号	姓名	工作地	部门	岗位	入职时间
1000	成大	深圳	销售部	销售	2015年3月
1001	张三	北京	研发部	前端	2015年3月
1002	李四	程度	产品部	产品经理	2016年7月
1003	王五	北京	产品部	运营	2020年9月
1004	张一	上海	销售部	销售	2018年3月
1005	李四	上海	销售部	销售	2018年3月
1006	李四	上海	研发部	后端	2017年4月

2. 在编辑页面，添加表格组件，数据类型选择离线文件，选择部门成员信息表，新建 a、b、c 三个变量，默认值设置为空，由于上传的 Excel 表格的数据结构与 Web 内表格数据结构的不匹配，添加一个包含 a、b、c 变量引用的过滤器，过滤器内容如下：

```
function findTableDataResultFinal(option, dataSource) {

    // 变量设置

    var tablePanel = [{}], currentDataList = dataSource;

    var r = getCurrentLengthFinal(option, currentDataList);

    var hederIndex = "";

    if (option.hPosition === "top" || option.hPosition === "left" ||
option.hPosition === "" || option.hPosition === "none" ||
!option.hPosition) {

        hederIndex = "0";

    } else if (option.hPosition === "bottom") {

        hederIndex = "" + r.h - 1;

    } else if (option.hPosition === "right") {

        hederIndex = "" + r.w - 1;

    }

    // 数据为空时候

    if (currentDataList.length === 0) {

        tablePanel = drawTablePanelFinal(r.w, r.h);

    }

}
```

```
if (currentDataList && Array.isArray(currentDataList)) {

    // 绘制表格格式

    tablePanel = drawTablePanelFinal(r.w, r.h);

    // 表头

    var hederList = option.headerType === "static" ?
option.headerNameList : [];

    // 表头数据设置

    for (var i in hederList) {

        if (option.hPosition === "left" || option.hPosition ===
"right") {

            tablePanel[i]["cells"][hederIndex]["text"] = hederList[i];

        } else if (option.hPosition === "top" || option.hPosition ===
"bottom" || option.hPosition === "" || option.hPosition === "none" ||
!option.hPosition) {

            tablePanel[hederIndex]["cells"][i]["text"] = hederList[i];

        }

    }

    if ((option.hPosition === "top" || option.hPosition === "left"
|| option.hPosition === "" || option.hPosition === "none" ||
!option.hPosition) && option.headerType !== "none" &&
option.headerType) {

        n = 1;

    } else {

        n = 0;

    }

}
```

```
// 源数据循环

currentDataList.forEach(function (item, index) {

    for (var i in option.cellsFieldNameList) {

        var key = option.cellsFieldNameList[i], v = item[key] !=
null ? item[key] : '';

        if (option.hPosition === "left" || option.hPosition ===
"right") {

            tablePanel[i]["cells"][n]["text"] = v;

        } else {

            tablePanel[n]["cells"][i]["text"] = v;

        }

    }

    n++;

})

} else {

    tablePanel = drawTablePanelFinal(r.w, 1);

    var hederList = option.headerType === "static" ?
option.headerNameList : [];

    for (var i in hederList) {

        if (option.hPosition === "left" || option.hPosition ===
"right") {

            tablePanel[i]["cells"][hederIndex]["text"] = hederList[i];

        } else {

            tablePanel[hederIndex]["cells"][i]["text"] = hederList[i];

        }

    }

}
```



```
    }

    }

    for (var i in option.cellsFieldNameList) {

        var key = option.cellsFieldNameList[i], v =
currentDataList[key] != null ? currentDataList[key] : 0;

        if (option.hPosition === "left" || option.hPosition ===
"right") {

            tablePanel[i]["cells"][hederIndex]["text"] = hederList[i];

        } else {

            tablePanel[0]["cells"][i]["text"] = v;

        }

    }

}

return tablePanel;

}

function getCurrentLengthFinal(option, dataSource) {

    var r = { w: 1, h: 1 };

    var obj = [];

    if (dataSource && Array.isArray(dataSource)) {

        if (option.fn) { dataSource.forEach(function (item) { if
(option.fn(item)) obj.push(item); }) }

        else { obj = dataSource; }

    } else {
```

```
    obj = dataSource

}

if (option.hPosition === "top" || option.hPosition === "bottom" ||
!option.hPosition) {

    r.w = option.headerType === "static" ?
option.headerNameList.length : option.cellsFieldNameList.length;

    r.h = option.headerType === "static" ? obj.length + 1 :
obj.length;

}

if (option.hPosition === "left" || option.hPosition === "right") {

    var headerType = option.headerType

    r.w = headerType === "static" ? obj.length + 1 : obj.length

    r.h = headerType === "static" || headerType === "none" ||
!headerType ? option.headerNameList.length : r.h

}

return r

}

function drawTablePanelFinal(width, height) {

    var row, cell, row = { "cells": {} }, cell = {};

    for (var i = 0; i < width; i++) row['cells'][i] = { "text": null
};

    for (var i = 0; i < height; i++) cell[i] =
JSON.parse(JSON.stringify(row));

    return cell;

}
```

```
}

function calculate(arrValue,data){if(!Array.isArray(data)){let e=
[];e.push(data),data=e}return arrValue.forEach(handleValue=>
{switch(handleValue.defaultType){case 0:switch(handleValue.dealType)
{case 2:data.forEach((e,t)=>{data[t]
[handleValue.fieldDesc.filed]=e[handleValue.fieldDesc.filed].toString(
).replaceAll(handleValue.config.field1,handleValue.config.field2)});br
eak;case 3:data.forEach((e,t)=>{data[t]
[handleValue.fieldDesc.filed]=e[handleValue.fieldDesc.filed].toString(
).replaceAll(handleValue.config.field1,"")});break}break;case 1:let
arr=[0,1,2,3,4,5,"+","-","*","/",10,"%"];10==handleValue.dealType?
data.forEach((e,t)=>{let
a=Math.round(parseFloat(e[handleValue.fieldDesc.filed])*Math.pow(10,ha
ndleValue.config.field1))/Math.pow(10,handleValue.config.field1);data[
t]
[handleValue.fieldDesc.filed]=a.toFixed(handleValue.config.field1)}):h
andleValue.dealType&&data.forEach((item,index)=>{const d=
(""+item[handleValue.fieldDesc.filed]).split(".").length>1?
(""+item[handleValue.fieldDesc.filed]).split(".")[1].length:0,d1=
(""+handleValue.config.field1).split(".").length>1?
(""+handleValue.config.field1).split(".")[1].length:0;let
pow=7==handleValue.dealType||6==handleValue.dealType?d>d1?
d:d1:8==handleValue.dealType?d+d1:d1-
d;if(7==handleValue.dealType||6==handleValue.dealType)data[index]
[handleValue.fieldDesc.filed]=eval(`((${parseFloat(item[handleValue.f
ieldDesc.filed])} * ${Math.pow(10,pow)}) ${arr[handleValue.dealType]}
(${parseFloat(handleValue.config.field1)} * ${Math.pow(10,pow)}))/
${Math.pow(10,pow)}).toFixed(${pow})`);else
if(8==handleValue.dealType||9==handleValue.dealType){const filed=
(""+item[handleValue.fieldDesc.filed]).replace(".", ""),filed1=
(""+handleValue.config.field1).replace(".", "");data[index]
[handleValue.fieldDesc.filed]=eval(8==handleValue.dealType?
`${Number(filed)} ${arr[handleValue.dealType]} ${Number(filed1)}`
`${arr[9]}
${Math.pow(10,pow)} `:numMulti(Number(filed)/Number(filed1),Math.pow(10
,pow)))}else data[index]
[handleValue.fieldDesc.filed]=eval(`parseInt(`${parseFloat(item[handleV
alue.fieldDesc.filed])} ${arr[handleValue.dealType]}
${parseFloat(handleValue.config.field1)})`);break;case 2:let arrD=
["yyyy","yyyy年","yyyy年MM月","yyyy-MM","yyyy/MM","yyyy年MM月dd
日","yyyy-MM-dd","yyyy/MM/dd","yyyy.MM.dd","MM","MM月","MM月dd
日","MM/dd","MM-dd","MM.dd","dd日","dd"];data.forEach((e,t)=>
```

```
{handleValue.dealType?data[t]
[handleValue.fieldDesc.filed]=dateformatFunc(e[handleValue.fieldDesc.f
iled],arrD[Number(handleValue.dealType)-12]):data[t]
[handleValue.fieldDesc.filed]=e[handleValue.fieldDesc.filed]}};break;d
efault:break}}),data;function numMulti(e,t){var
a=0;try{a+=e.toString().split(".")[1].length}catch(s)
{}try{a+=t.toString().split(".")[1].length}catch(s){}return
Number(e.toString().replace(".", ""))*Number(t.toString().replace(".", ""
))/Math.pow(10,a)}}
```



```
function dateformatFunc(e,t){var a=new
Date(e),s=a.getFullYear(),i=a.getMonth()+1,o=a.getDate();return
t=t.replace("yyyy",s).replace("MM", (""+i).length>1?
i:"0"+i).replace("dd", (""+o).length>1?o:"0"+o),t}
```



```
function dataSourceProcess(data, args) {

if (args.a) {

    data = data.filter(o => o["姓名"] == args.a);

}

if (args.b) {

    data = data.filter(o => o["工作地"] == args.b);

}

if (args.c) {

    data = data.filter(o => o["入职时间"] == args.c);

}

return data

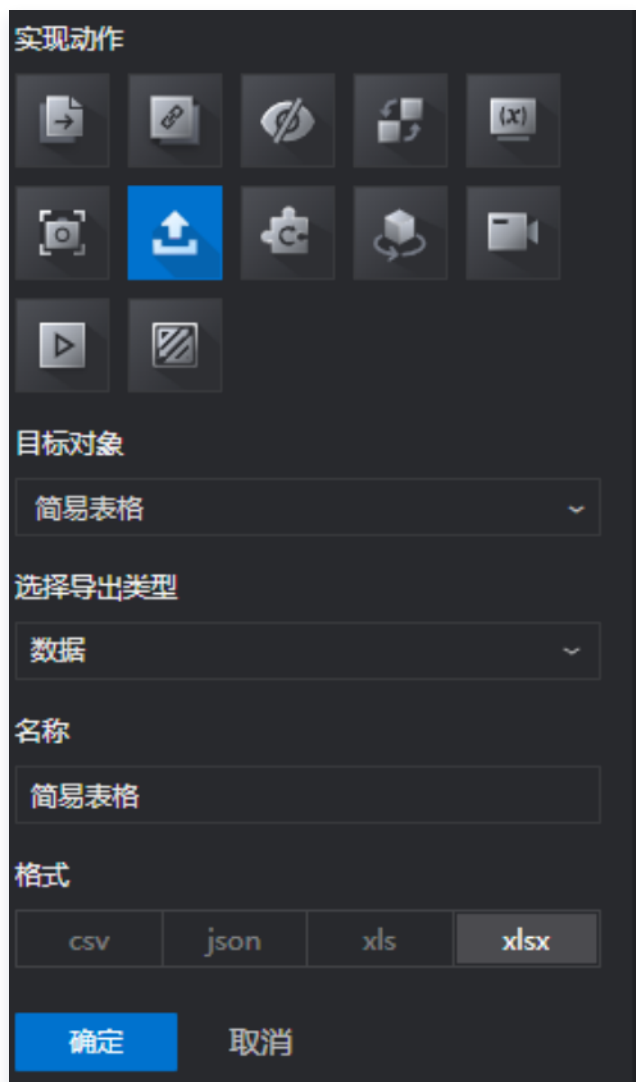
}
```

```
var logicItem = [{"name": "员工编号", "defaultType": 0, "config":  
{"field1": "", "field2": "", "isEdit": false}, "source": 13348, "fieldDesc":  
{"filedId": 13348, "dataItemConfigId": 0, "filed": "id", "mapName": "员工编  
号", "content": ""}}, {"name": "姓名", "defaultType": 0, "config":  
{"field1": "", "field2": "", "isEdit": false}, "source": 13349, "fieldDesc":  
{"filedId": 13349, "dataItemConfigId": 0, "filed": "姓名", "mapName": "姓  
名", "content": ""}}, {"name": "工作地", "defaultType": 0, "config":  
{"field1": "", "field2": "", "isEdit": false}, "source": 13351, "fieldDesc":  
{"filedId": 13351, "dataItemConfigId": 0, "filed": "工作地", "mapName": "工作  
地", "content": ""}}, {"name": "部门", "defaultType": 0, "config":  
{"field1": "", "field2": "", "isEdit": false}, "source": 13350, "fieldDesc":  
{"filedId": 13350, "dataItemConfigId": 0, "filed": "部门", "mapName": "部  
门", "content": ""}}, {"name": "岗位", "defaultType": 0, "config":  
{"field1": "", "field2": "", "isEdit": false}, "source": 13352, "fieldDesc":  
{"filedId": 13352, "dataItemConfigId": 0, "filed": "岗位", "mapName": "岗  
位", "content": ""}}, {"name": "入职时间", "defaultType": 0, "config":  
{"field1": "", "field2": "", "isEdit": false}, "source": 13353, "fieldDesc":  
{"filedId": 13353, "dataItemConfigId": 0, "filed": "入职时间", "mapName": "入职  
时间", "content": ""}}]  
  
data = dataSourceProcess(data, args)  
  
data = calculate(logicItem, data)  
  
var option = {"path": "", "cellsFieldNameList": ["id", "姓名", "工作地", "部  
门", "岗位", "入职时间"], "headerNameList": ["员工编号", "姓名", "工作地", "部  
门", "岗位", "入职时间"], "headerType": "static", "hPosition": "top"}  
  
return findTableDataResultFinal(option, data)
```

3. 添加一个输入框、一个下拉框、一个时间选择器，及两个文本框，两个矩形（用来模拟查询按钮，及导出按钮）。
4. 下拉框组件，数据类型选择离线文件，选择部门成员信息表，字段映射 content，映射工作地，单击提交，下拉框显示工作地。
5. 给查询按钮绑定交互事件，触发条件选择单击，实现动作选择变量赋值：
 - 让 a 关联输入框的 content 字段。
 - b 关联下拉框的 content 字段。
 - c 关联时间选择器的 time 字段。



6. 给导出按钮绑定交互事件，触发条件选择单击，实现动作选择导出，导出格式选择 xlsx，设置完成后，单击确定。



- 单击**预览**，输入框输入李四，下拉框工作地选择上海，时间选择器的入职时间选择2017年4月。单击**查询**，可筛选出符合该三个限定条件的数据，单击**导出**，可把符合该三个限定条件的数据下载到本地。

李四

上海

2017年4月

查询

导出

员工编号	姓名	工作地	部门	岗位	入职时间
1006	李四	上海	研发部	后端	2017年4月

说明：
上方为了方便用户体验完整流程，选择离线文件的数据类型来实现，也可使用其他类型的数据源。

下拉框查找并定位三维标记的使用

最近更新时间：2025-03-14 17:26:42

操作场景

下拉框选择北京，则三维标记显示在北京并且摄像机注视到北京，下拉框选择成都，则三维标记显示在成都并且摄像机注视到成都。

操作步骤

1. 在管理中心页面，单击**添加数据源**，格式选择**离线文件**，[单击下载示例数据](#)后，将下载好的文件后上传至产品中。
2. 在编辑页面添加下拉框组件，单击数据模块选择**离线文件**，添加该数据源。



3. 场景选择中国版块，并添加三维标记图片标记，将图片标记添加刚上传的离线文件内容。

基础设置

数据

联动设置

图片标记

静态数据

数据类型

城市名

经纬度

离线文件

WebAPI

WebSock...

MySQL

Oracle

SQL Server

PostgreS...

DAYU数...

MQTT

字段映射

字段

映射

说明

状态

cityName

city

城市

匹配

title

title

标题

匹配

note

note

注释

未匹配

imageUrl

pic

图片地址

匹配

选择数据源

标记-城市

解析

数据路由器

数据响应结果

表格视图

代码视图

acode

area

city

pic

prov

subTi

20

210100

14.8

沈阳

https://n.sinai...

辽宁省

沈阳

21

230100

47.3

哈尔滨

https://bking....

黑龙江省

哈尔

22

220100

18.74

长春

https://pics4.b...

吉林省

长春

23

150100

118.3

呼和浩特

https://pics5.b...

内蒙古自治区

呼和

24

140100

15.67

太原

https://bking....

山西省

太原

25

330100

10.55

杭州

https://5b098...

浙江省

杭州

26

130100

18.88

石家庄

https://pics4.b...

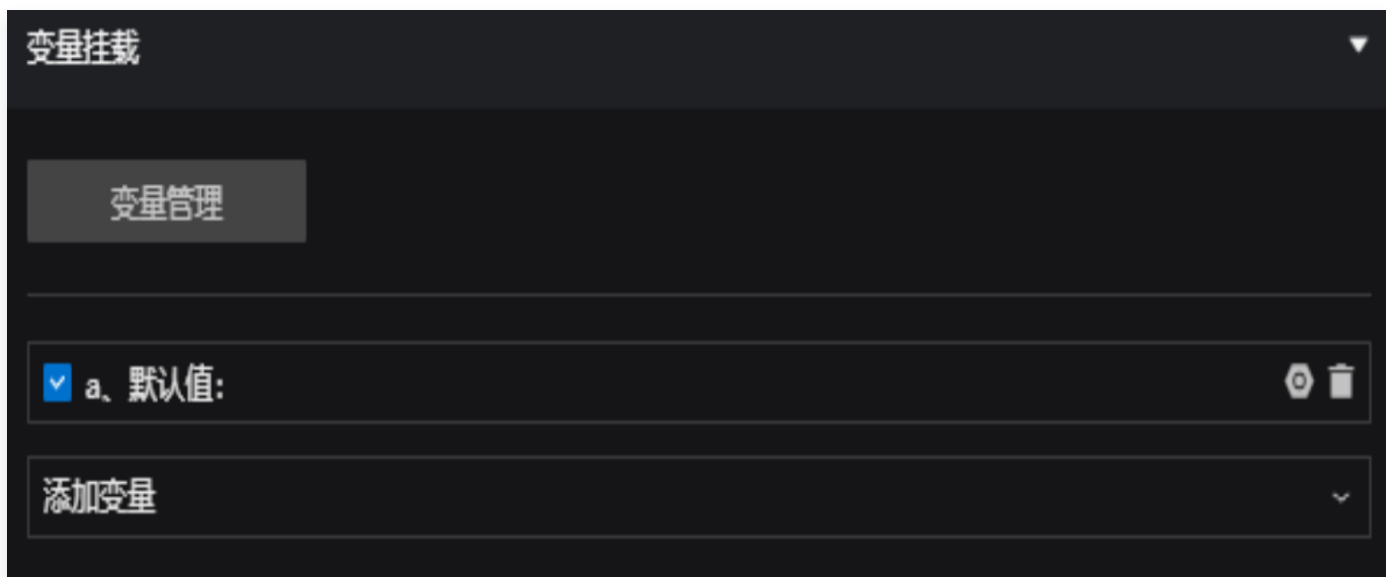
河北省

石家

变量挂载

提交

4. 在变量挂载处，添加一个变量，变量名称设为 **a**，默认值可不填，假设不填，绑定该变量。

5. 在过滤器中写入简单语句，把变量 **a** 的值给到 **city**，只显示具体 **city** 下的数据内容。由于变量默认值为空，则无任何值给到 **city**，则经过过滤器后数据为空。

```
return data.filter(item => args.a === item.city);
```

6. 为下拉框组件，添加联动事件，设置完成后，单击**确定**。

- 触发条件：选择**选项变化时**。
- 实现动作：选择**变量赋值**。让变量 **a** 与输入框字段 **content** 建立关系，由于让同一个选项变化时的触发条件同时生效几个实现动作时，会有时间先后顺序上的依赖关系，此时为了实现摄像机可以准确及时注视到通过下拉框选项变化时传递的变量影响到的三维标记，需要再添加另外一个延时时间相关的联动事件。

- 触发条件：选择**选项变化时**。
- 延迟时间：选择**1s**或者其他的时间。
- 实现动作：选择**摄像机注视到图片标记**。

7. 单击**预览**，在下拉框中选择北京，则三维标记显示在北京并摄像机注视到北京，选择成都，则三维标记显示在成都并注视到成都。



表格单元格给其他组件进行传值

最近更新时间：2025-03-14 17:26:42

操作场景

鼠标单击表格单元格组件，单元格的内容传递给文本组件。

操作步骤

1. 在编辑页面添加表格组件，单击数据模块 API。

○ 在API 中：添加 `https://v1-api.raykite.com/demo/api-data/table` 链接。

○ 在过滤器中：写入 `return data.data;` 。

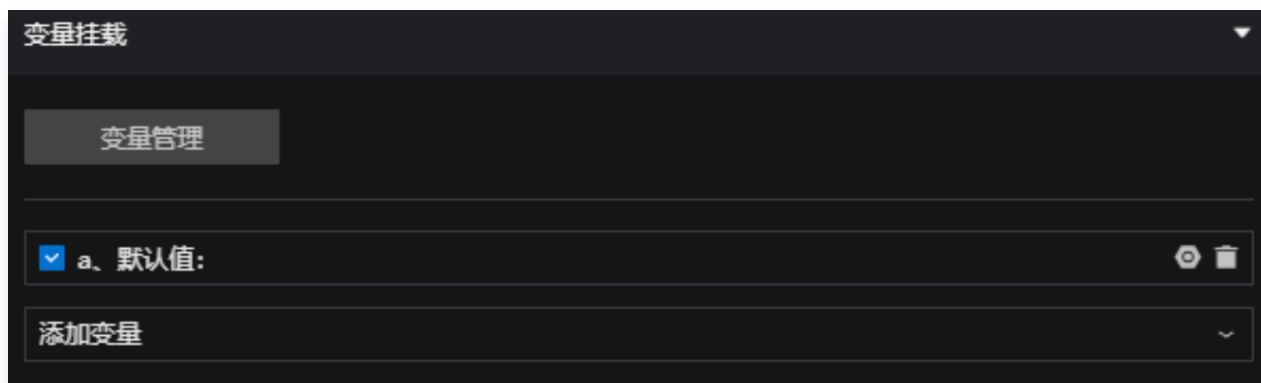
2. 单击测试和提交，表格接入 API 数据。

序号	标志	内容	数量	内容
01	1	文字内容	985	0.1
02	2	文字内容	666	0.2
03	3	文字内容	302	-0.3
04	4	文字内容	202	-0.4

3. 添加一个多行文本组件，单击数据模块 API，在 API中 添加

`https://v1-api.raykite.com/demo/api-data/table` 链接。

3.1 在变量挂载处，添加一个变量，变量名称设为 a，默认值可不填，假设不填，绑定该变量，并让 content 映射 a。



字段映射 ②	字段	映射	说明	状态
	* content	a	内容	未匹配

3.2 在过滤器中写入简单语句：`return args;`

4. 给表格组件，添加联动事件，设置完成后，单击确定。

- 触发条件：选择鼠标单击。
- 触发主体：选择单元格。
- 实现动作：选择变量赋值。让变量 a 与表格字段 text 建立关系。



5. 单击预览按钮，单击任意表格单元格，通过传值，下方文本内容也变成单元格内容。



搜索时动态展示卡片

最近更新时间：2025-03-14 17:26:42

操作场景

可以结合输入框、变量合联动事件实现查询时动态展示对应卡片且自动调整位置。

操作步骤

制作流程及效果见视频：

[观看视频](#)

表格二级联动

最近更新时间：2025-03-14 17:26:42

操作场景

适用于进行二级联动的效果，如点击组件1，组件2的数据会跟随变化，点击组件2，组件3的数据会跟随变化。

操作步骤

制作流程及效果见视频：

[观看视频](#)