

# 脚本查询

脚本查询是指通过编写代码脚本获取数据结果。

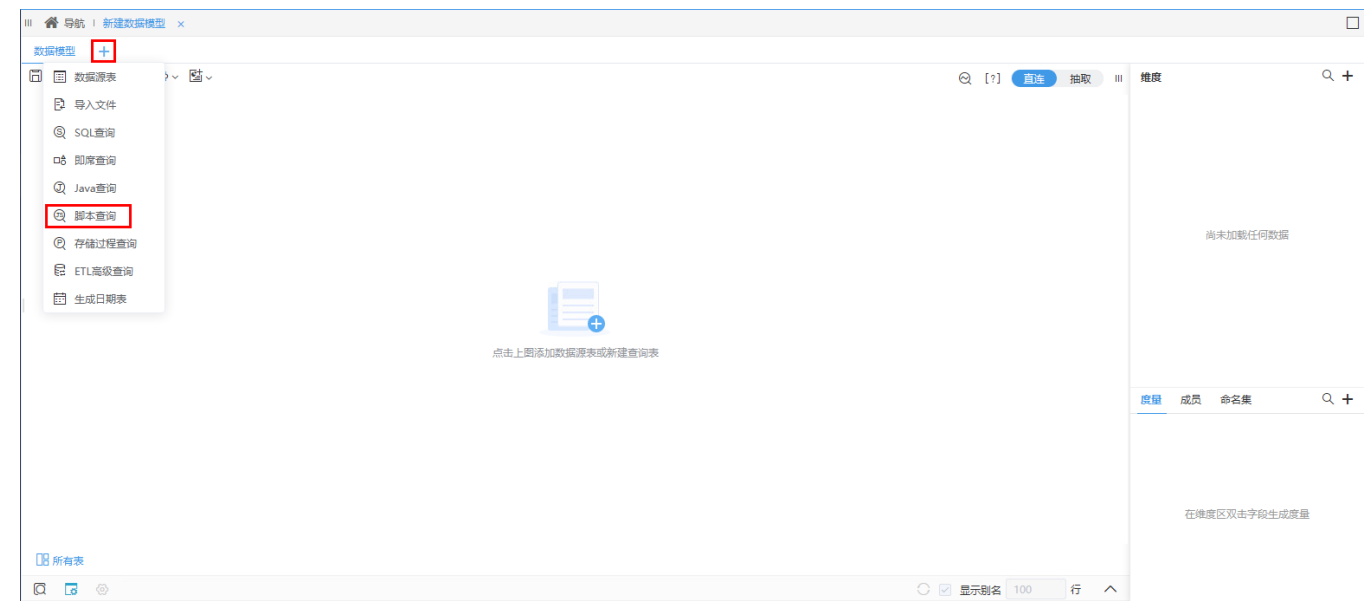
**适用情况：**各种非结构化数据、特殊数据。

目前“脚本查询”支持的语言只有JavaScript，因此适用于熟悉JavaScript的技术人员。相关接口文档可参考[JavaScriptData](#)。

- [操作入口](#)
- [界面介绍](#)
  - [功能按钮区](#)
  - [脚本编辑区](#)
  - [参数设置/数据预览区](#)
  - [获取资源节点ID](#)
  - [新建“脚本查询”节点](#)
  - [编写脚本](#)
  - [校验语法](#)
  - [执行脚本](#)
  - [保存节点到数据模型页面](#)

## 操作入口

在“数据模型”定制界面单击创建私有查询按钮，选择 **脚本查询**（如下图所示），进入脚本查询的定制界面。



## 界面介绍

脚本查询定制界面如下图所示：

功能按钮区

JavaScript脚本编辑区

```
1 /**
2  * JavaScript数据示例，在脚本中需要提供四个方法（其中getRowCount()在usePaging=false不需要提供）：
3  *
4  * function init() {
5  *     初始化方法，返回初始化结果。
6  *     @result 返回查询的参数与输出字段，结构为包含usePaging、params和fields属性的对象。
7  *     {
8  *         //声明getGridData是否使用分页返回，默认不使用分页
9  *         //当usePaging=false时，服务器会一次性加载所有数据到内存中
10        //因此在返回数据量可能很大的情况下必须usePaging=true
11        usePaging: true | false,
12        //参数数组，包含零个或多个参数。参数之间不能有相同的参数ID、参数名称、参数别名
13        params: [{
14            id: 'ParamId', //参数ID
15            name: 'ParamName', //参数名称
16            desc: 'ParamDesc', //参数描述
17            nullable: true, //参数是否允许是空
18        }]
19    }
20 }
```

参数设置/数据预览区

| 名称 | 数据类型 | 默认值 |
|----|------|-----|
|    |      |     |

- “脚本查询”界面主要分为以下几个区域：
- 功能按钮区：列出了可操作脚本查询的工具按钮；
  - 脚本编辑区：编辑脚本代码；
  - 参数设置/预览数据区：设置脚本的参数，查看输出的数据；

功能按钮区

功能按钮区的按钮具体说明如下表：

| 按钮                                                                                           | 说明                                                                |
|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| 执行<br>    | 执行脚本。若脚本的结果集中包含参数且未设置默认值，则会提示用户输入参数默认值。                           |
| 保存<br>    | 将当前脚本查询保存到“数据模型”定制界面的表关系区内，如果是第一次创建，则会在表关系区创建对应的节点，节点名称默认为“脚本查询”。 |
| 格式化脚本<br> | 对脚本编辑区内的代码进行格式化操作，使其符合代码规范                                        |

功能按钮区

JavaScript脚本编辑区

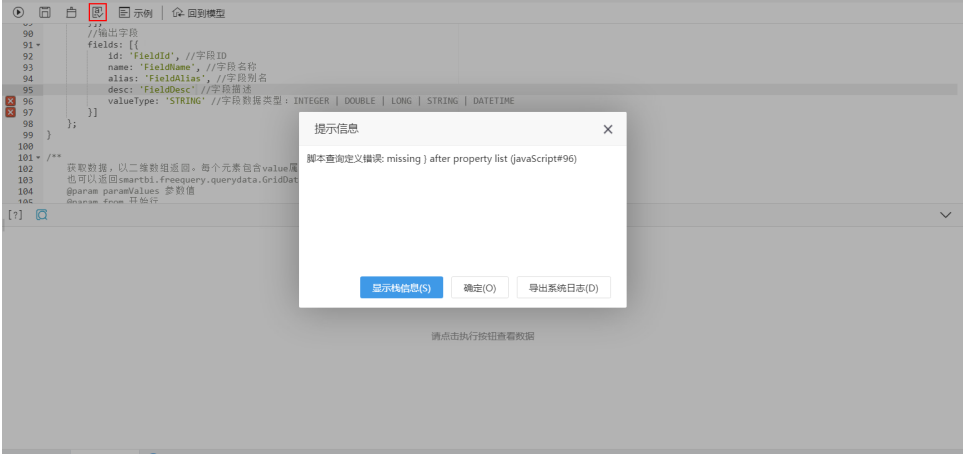
```
79 */
80 function init() {
81     return {
82         //参数列表
83         params: [{
84             id: 'ParamId', //参数ID
85             name: 'ParamName', //参数名称
86             desc: 'ParamDesc', //参数描述
87             nullable: true, //参数是否允许是空
88             valueType: 'STRING' //参数数据类型: INTEGER | DOUBLE | LONG | STRING | DATETIME
89         }],
90         //输出字段
91         fields: [{id: 'FieldId', name: 'FieldName', alias: 'FieldAlias', desc: 'FieldDesc', valueType: 'STRING' }]
92     };
93 }
```

参数设置/数据预览区

ParamName

共 1 行 每页 1000 行 << 1 / 1 >> | 刷新

| FieldAlias |
|------------|
| Bad        |

|                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div>校验</div> <div></div>                      | <div><ul style="list-style-type: none"><li>校验脚本编辑区中的代码，如果有语法错误则提示具体的错误</li></ul></div> <div></div> <div><ul style="list-style-type: none"><li>如果正确则提示校验成功。</li></ul></div> <div></div> |
| <div>示例</div> <div></div>                     | <div>在脚本编辑区生成内置的JavaScript示例代码</div>                                                                                                                                                                                                                                                                                                                       |
| <div>回到数据集</div> <div></div> <div>回到模型</div> | <div>返回到数据模型主页面</div>                                                                                                                                                                                                                                                                                                                                      |

## 脚本编辑区

用户可在此区域进行脚本代码的编写。

- 在方法init()内可以自定义结果集，自定义私有参数和输出字段。私有参数，用于过滤数据。

```
function init(){
  return {
    //参数列表
    params: [{
      id: 'ParamId', //参数ID
      name: 'ParamName', //参数名称
      alias: 'ParamAlias', //参数别名
      desc: 'ParamDesc', //参数描述
      nullable: true, //参数是否允许是空
      valueType: 'STRING' //参数数据类型: INTEGER | DOUBLE | LONG | STRING | DATETIME
    }],
    //输出字段
    fields: [{
      id: 'FieldId', //字段ID
      name: 'FieldName', //字段名称
      alias: 'FieldAlias', //字段别名
      desc: 'FieldDesc', //字段描述
      valueType: 'STRING' //字段数据类型: INTEGER | DOUBLE | LONG | STRING | DATETIME
    }]
  };
}
```

- 代码编辑完成后，鼠标点击编辑区域以外的区域，则会校验脚本语法，并识别所编写脚本的结果集中的参数，在“参数设置”区域生成对应的参数。

▶ 回 白 目 示例 | 回到数据集

104 获取数据，以二维数组返回。每个元素包含value属性

105 也可以返回smartbi.freequery.querydata.GridData对象

106 @param paramValues 参数值

107 @param from 开始行

108 @param count 返回的行数

109 @result 返回数据

110 \*/

111 function getGridData(paramValues, from, count) {

112 //通过 paramValues.get("参数ID") 获取参数值对象（包含value与displayValue属性）

113 var val = paramValues.get("ParamId");

114 if (val && val.value == "ParamValueA") {

115 return [

116 [{value:"OK"}]

[?] 🔍

▼

| 名称 | 数据类型 | 默认值 |
|----|------|-----|
|    |      |     |

## 参数设置/数据预览区

用于显示执行脚本查询后的数据结果，主要分为参数设置界面和数据预览区。

### 1) 参数设置界面

若在脚本代码的结果集中定义了参数，点击按钮 **参数设置** [?] 切换至“参数设置”页面，“参数设置”页面会显示对应的参数，设置参数的默认值；

[?] 🔍

▼

| 名称        | 数据类型 | 默认值 |
|-----------|------|-----|
| ParamName | 字符串  |     |

### 2) 数据预览区

单击 **数据预览** 按钮可切换至数据预览区，查看输出数据：

[?] 🔍

▼

共 未知 行 每页 10 行 << 1 / 未知 >> | 刷新

| CategoryID | Discontinued | ProductID | ProductName | QuantityPerUnit | ReorderLevel | SupplierID | UnitPrice | UnitsInStock | UnitsOnOrder |
|------------|--------------|-----------|-------------|-----------------|--------------|------------|-----------|--------------|--------------|
| 1          | -1           | 1         | 苹果汁         | 每箱24瓶           | 10           | 1          | 18.00     | 39           | 0            |
| 1          | 0            | 2         | 牛奶          | 每箱24瓶           | 25           | 1          | 19.00     | 17           | 40           |
| 2          | 0            | 3         | 蕃茄酱         | 每箱12瓶           | 25           | 1          | 10.00     | 13           | 70           |
| 2          | 0            | 4         | 盐           | 每箱12瓶           | 0            | 2          | 22.00     | 53           | 0            |
| 2          | -1           | 5         | 麻油          | 每箱12瓶           | 0            | 2          | 21.35     | 0            | 0            |



数据预览前需在“参数设置”页面设置默认值后，点击功能按钮区中的“执行”按钮，数据预览页面才会输出相关数据。

### (3) 展开或收缩 数据结果展示区

- 点击符号“^”可将该区域展开，点击符号“v”可将开区域收起。

79 \*/  
80 \* function init() {  
81 \* return {  
82 \* //参数列表  
83 \* params: [{  
84 \* id: 'ParamId', //参数ID  
85 \* name: 'ParamName', //参数名称  
86 \* desc: 'ParamDesc', //参数描述  
87 \* nullable: true, //参数是否允许是空  
88 \* valueType: 'STRING' //参数数据类型: INTEGER | DOUBLE | LONG | STRING | DATETIME  
89 \* }],  
90 \* //输出字段  
91 \* fields: [{  
92 \* id: 'FieldId', //字段ID  
93 \* name: 'FieldName', //字段名称  
94 \* alias: 'FieldAlias', //字段别名

点击按钮展开/收起面板

[?] [X]

| 名称 | 数据类型 | 默认值 |
|----|------|-----|
|    |      |     |

## 示例演示

本示例以数据源“northwind”中的“产品表”为目标资源，展示了如何通过编写JavaScript脚本获取指定节点ID的数据源表数据，具体操作如下：

### 获取资源节点ID

左侧“系统导航栏”中的“数据连接”，选择“数据连接” > “northwind” > “产品表”，右键打开扩展菜单，点击“属性”。

The screenshot shows a system navigation pane on the left with a dark blue sidebar. The sidebar contains icons for a monitor, a database, a cube, a bar chart, a line chart, and a wrench. The main pane displays a tree structure of data sources. The 'Data Connection' (数据连接) icon is highlighted with a red box. A red arrow points from this icon to the 'Product Table' (产品表) entry under the 'northwind' data source. Another red arrow points from the 'Product Table' entry to the 'Properties' (属性) option in the right-click context menu.

记录下该资源的节点ID：

资源属性

节点ID:

TAB.northwind.null.products

名称:

products

别名:

产品表

路径:

数据连接\northwind\DEFAULT\产品表

类型名:

基础表

描述:

创建人:

管理员

创建时间:

2021-04-23 13:52:55

图标:

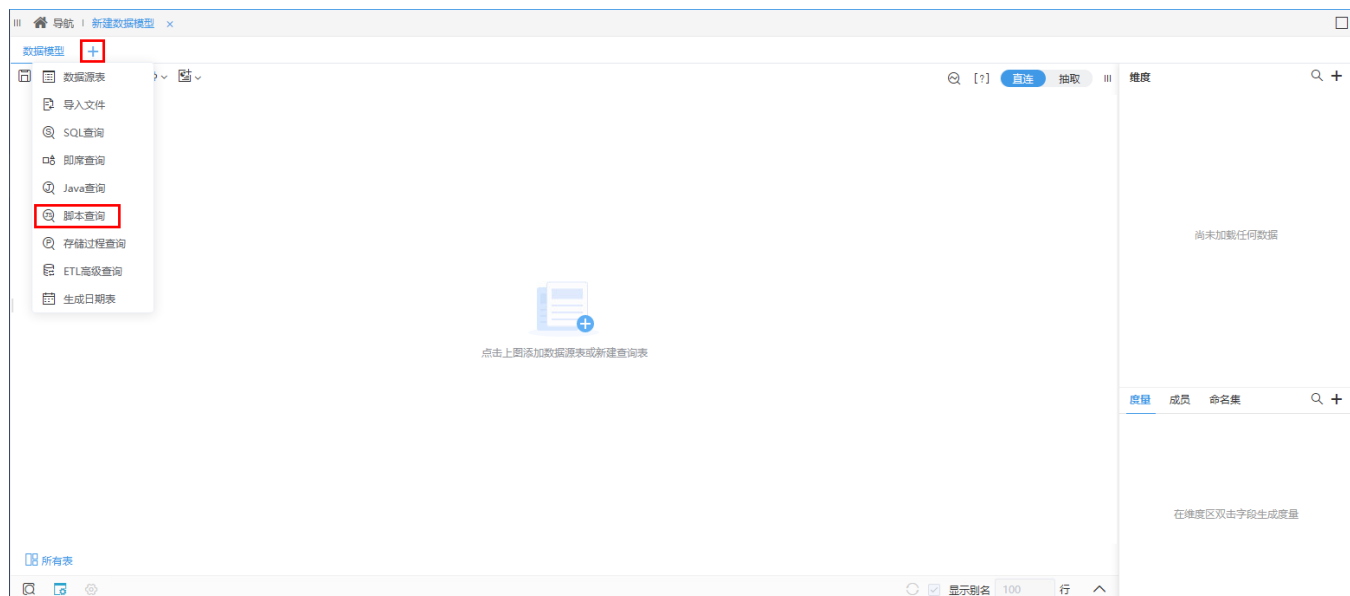
选择

确定(O)

取消(C)

## 新建“脚本查询”节点

单击创建私有查询按钮，选择 脚本查询：



## 编写脚本

复制代码到“脚本编辑区”，并修改目标资源的节点id。

🔍 📄 🏠 📌 示例 | 🏠 回到模型

```
1 importPackage(Packages.smartbi.freequery.client.datasource);
2
3 var datasourceService = DataSourceService.getInstance();
4 var tableId = "TAB.northwind.null.products"; //目标资源的节点ID
5 var table = datasourceService.getTable(tableId);
6 var sql = "select";
7 var data;
```

#### 示例代码 • 读取指定节点资源的表数据

```
importPackage(Packages.smartbi.freequery.client.datasource);

var datasourceService = DataSourceService.getInstance();
var tableId = "TAB.northwind.null.products"; //ID
var table = datasourceService.getTable(tableId);
var sql = "select";
var data;

function init() {
    var fieldList = table.getFields();
    var fields = [];
    for (var i = 0; i < fieldList.size(); i++) {
        var field = fieldList.get(i);
        sql = sql + " " + field.getName();
        fields[i] = {
            id: field.getId(), //ID
            name: field.getName(), //
            alias: field.getAlias(), //
            desc: field.getDesc(), //
            valueType: parseValueType(field.getDataType().getSQLType()) //INTEGER | DOUBLE | LONG | STRING |
DATETIME
        }
        if (i < fieldList.size() - 1) {
            sql = sql + ",";
        }
    }
    sql = sql + " from " + table.getName();
    return {
        // //
        // params: [{
            // id: 'Param.northwind.null.products', //ID
            // name: 'ParamName', //
            // alias: 'ParamAlias', //
            // desc: 'ParamDesc', //
            // nullable: true, //
            // valueType: 'STRING' //INTEGER | DOUBLE | LONG | STRING | DATETIME
        // }],
        //
        fields: fields
    };
}

function getGridData(paramValues, from, count) {
    data = datasourceService.executeNoCacheable(table.getDataSource().getId(), sql);
    return data;
}

function getRowCount() {
    return data.getRowsCount();
}
```

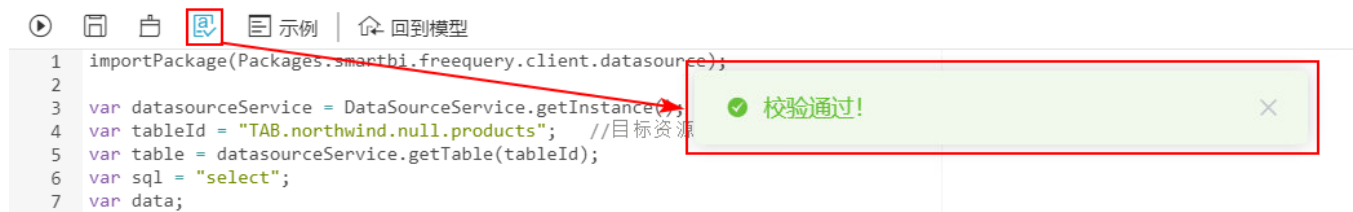
```
/**
 */
function close() {}

function parseValueType(type) {
  switch (" " + type) {
    case "LONG":
      return "LONG";
    case "BIGINT":
    case "INTEGER":
      return "INTEGER";
    case "BIGDECIMAL":
    case "DOUBLE":
      return "DOUBLE";
    case "TIME":
    case "DATE":
    case "DATETIME":
      return "DATETIME";
    default:
      return "STRING";
  }
}
```

## 校验语法

单击 **校验脚本** 按钮，校验脚本语法。

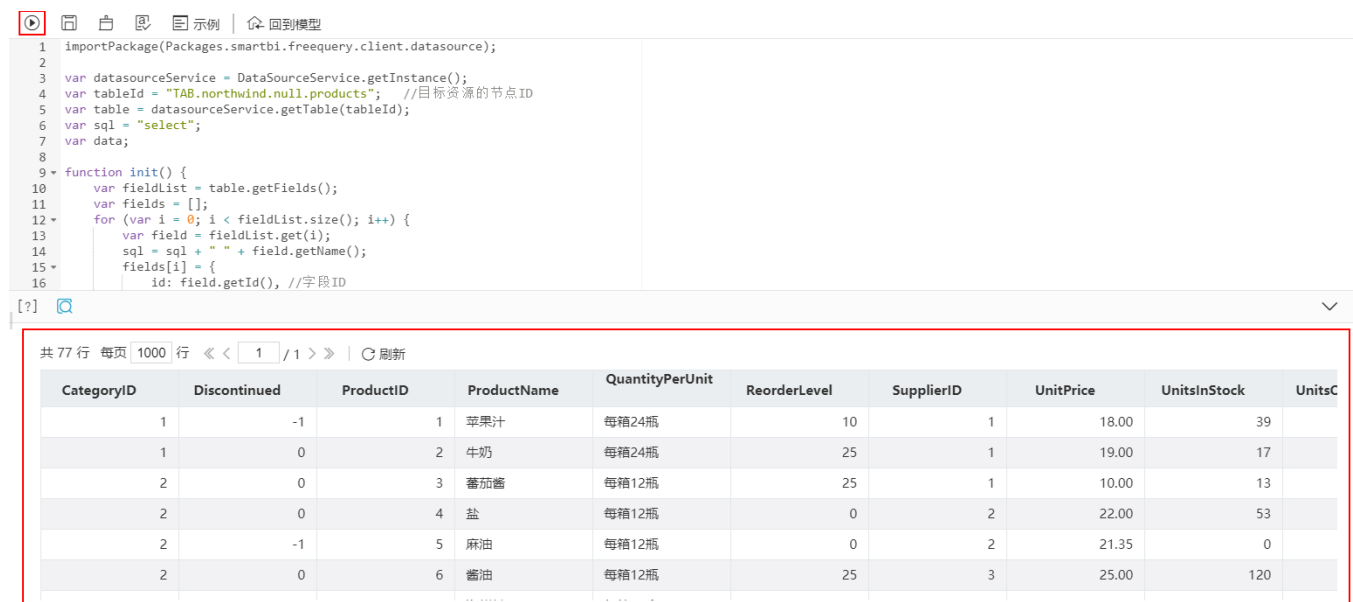
单击 **校验脚本** 按钮，校验脚本语法。



## 执行脚本

单击 **执行** 按钮，执行脚本，执行完成后数据预览区便会显示输出的数据。

单击 **执行** 按钮，执行脚本，执行完成后数据预览区便会显示输出的数据。



## 保存节点到数据模型页面

单击 **保存** 按钮，系统提示保存成功后，当前脚本查询会以表的输出节点方式保存在数据模型的”表关系区“中。

单击 **保存** 按钮，系统提示保存成功后，当前脚本查询会以表的输出节点方式保存在数据模型的“表关系区”中。



```
1 importPackage(Packages.smartbi.freequery.client.datasource);
2
3 var datasourceService = DataSourceService.getInstance();
4 var tableId = "TAB.northwind.null.products"; //目标资源
5 var table = datasourceService.getTable(tableId);
6 var sql = "select";
7 var data;
```

✔ 保存成功!



数据模型

脚本查询

+

🔍 脚本查询

直连

抽取

维度

成员

命名集

在维度区双击字段生成度量

所有表

脚本查询

显示别名

100

行