ALIBABA CLOUD

Alibaba Cloud

DataV

Advanced Skills

Document Version: 20220608

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

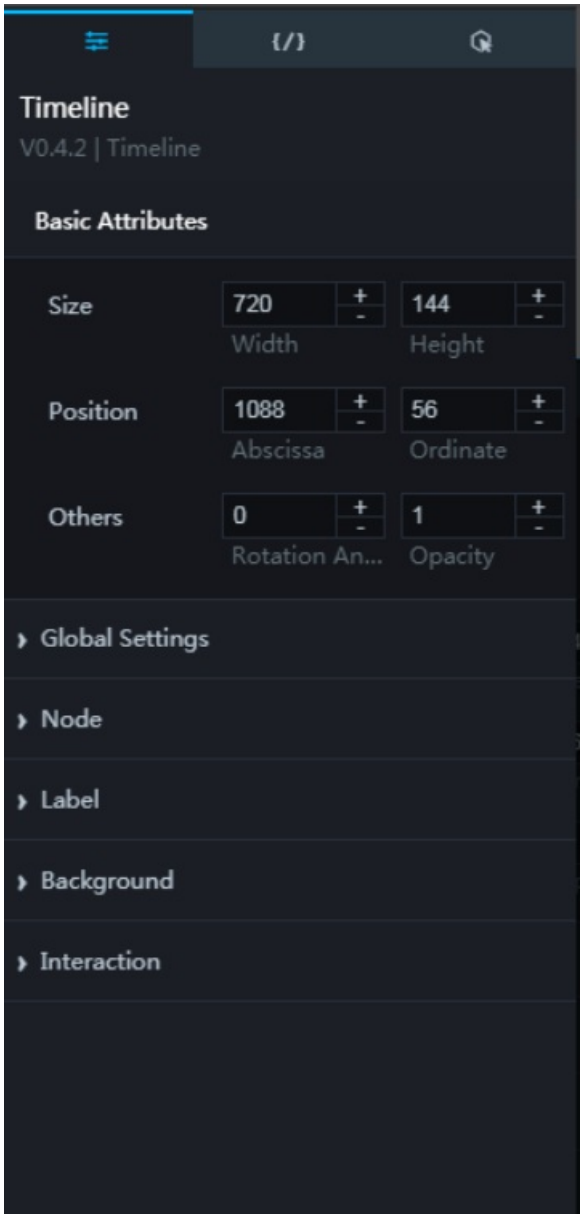
Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Widget interaction

Widgets can interact with each other through related callback IDs, see Configure widget interaction. By using callback IDs, if you click a widget, data of a widget with the corresponding callback ID will change accordingly.

Using a timeline widget as an example, with a callback ID, after you click a timestamp the map widget displays detailed information of this timestamp, see Configure widget interaction.

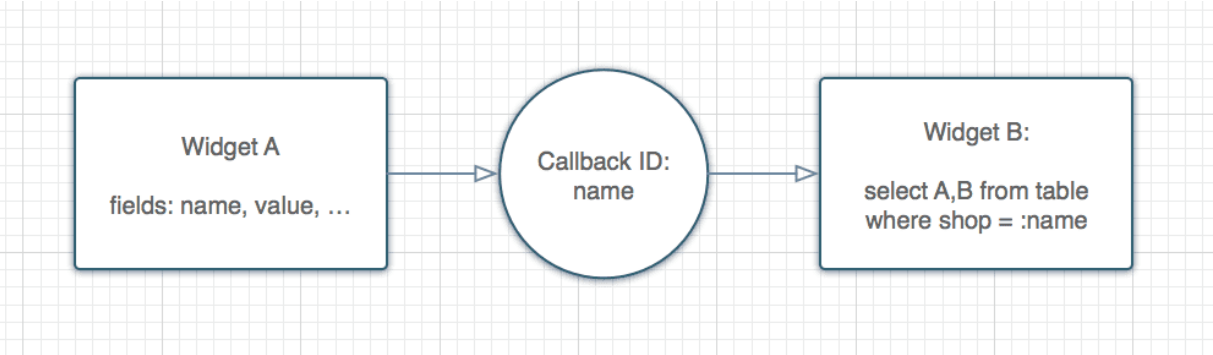


Set a callback ID

A callback ID can be considered as a parameter variable that is used to control the transmission of parameters between widgets, and realize interactions between them (that is, if data of one widget is updated, other widgets with the same callback ID are also updated).

> **Note** Callback ID is not supported if the Data Source Type is *Static Data* or a *CSV file*.

For instance, if you click a scattered point in a map, click a row in a list, or select a time point on a time table, the system temporarily stores the designated callback ID, for example, name, into the *name* variable.



Set variable input

When the Data Source Type is SQL or API, you can directly input the variable.

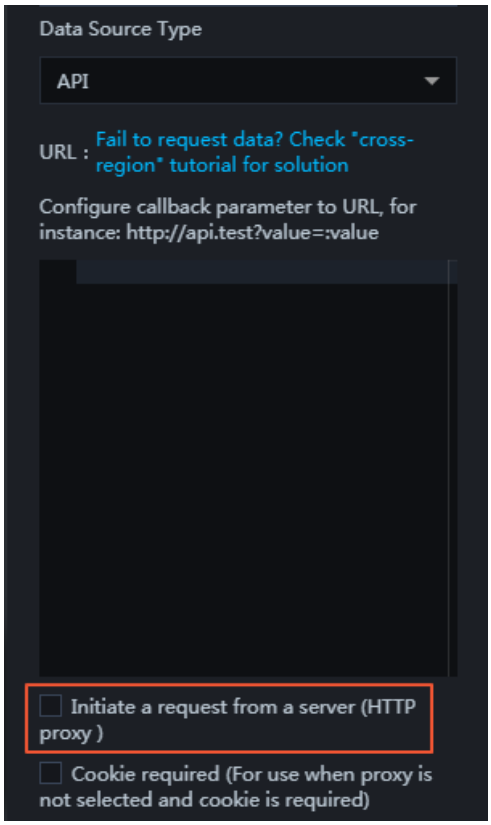
```
select :name as value
select A from table where year = :name
```

If you are using an API data source, the system directly assigns value to the variable parameters with the same name.

2.Cross-origin data configuration

Background

If user A creates a visualization project in DataV, and selects *API* from the Data Source Type drop-down list (shown in the following figure), depending on where the API is hosted, the following situations may occur.



- If the API is on a remote server and has access to a public network, select **Initiate a request from a server (HTTP proxy)**. The DataV backend server initiates a request to the API at the same time. The request has a 10-second timeout and cannot be changed.
- If the API is on a local server, deselect **Initiate a request from a server (HTTP proxy)**, and the interface needs to be configured across origins. The API is accessed by a local browser and the timeout duration is also determined by the local browser.

What is an issue related to cross-origin data configuration

Assume that user A's website needs to include data from their own website and from user B's website. Data on user A's website can be accessed by the website interface <http://userA.com/page1>. The website interface for user B is <http://userB.com/page2>. If user A uses a Javascript AJAX request to gain access to user B's website, user A cannot retrieve data from userB.com.

If, for example, a Chrome browser is opened during the preceding request, the following is displayed:

```
XMLHttpRequest cannot load http://userB.com/page2. Because no 'Access-Control-Allow-Origin'
header is present on the requested resource.Origin 'http://userA.com/' is not allowed acces
s.
```

This means we encountered a cross-domain issue.

Reason

Because each website contains various interfaces, such as user interface, order interface and article interface, each user can put the data returned from those interfaces on their own website. This means the browsers have a **same-origin policy**, which restricts a script from one origin to obtain resources from another origin.

> **Note** **Same origin**: If two pages have the same protocol (HTTP), port (80) and host (userA.com), these two pages are considered to be from the same origin.

Solution

- **Cross-origin between A.x.com and B.x.com**

  If the sub-origins for two sites are different, such as http://xyz.userA.com/ and http://123.userA.com/, the cross-origin issue still exists.

  To solve this issue, you must declare the page to a higher-level origin.

  ```
  <script>
    document.domain = "x.com";
  </script>
  ```

- **JSONP**

  Although JSONP is a commonly applied solution, it is vulnerable to Cross-Site Scripting (XSS) attacks. Therefore, DataV does not support this method.

- **Cross-Origin Resource Sharing (CORS)**

  > **Note** Cross-Origin Resource Sharing (CORS) was introduced to solve issues related to the sharing of restricted resources. Most new browsers are compatible with CORS.

  Concept

  Add the custom header information HTTP to user B's website (local API) so that its resources can be accessed by other websites.

  Example

  - Header information returned from a data server includes the following basic content:

    ```
    Access-Control-Allow-Origin: http://www.example.com
    ```

- If user B's website interface data are used by many specified websites, the header information must be generated dynamically. The following example uses PHP:

```
<? php
if (is_my_code($_SERVER['HTTP_ORIGIN'])) {
header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
}
? >
```

- If user B's website interface data are used by all the other websites, use "*".

```
Access-Control-Allow-Origin: *
```

Cookie

By default, CORS does not contain cookie information. If you want to add cookies, follow these steps:

i. Add the **withCredentials** parameter, such as jquery:

```
$.ajax({
 url: "http://userB.com/page2",
 xhrFields: {
   withCredentials: true
 }
});
```

ii. Configure the server to allow headers with credentials and disallow the wildcard character "*", as follows:

```
<? php
if (is_my_code($_SERVER['HTTP_ORIGIN'])) {
header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}"); // Disallow "*"
header("Access-Control-Allow-Credentials:true");
}
? >
```

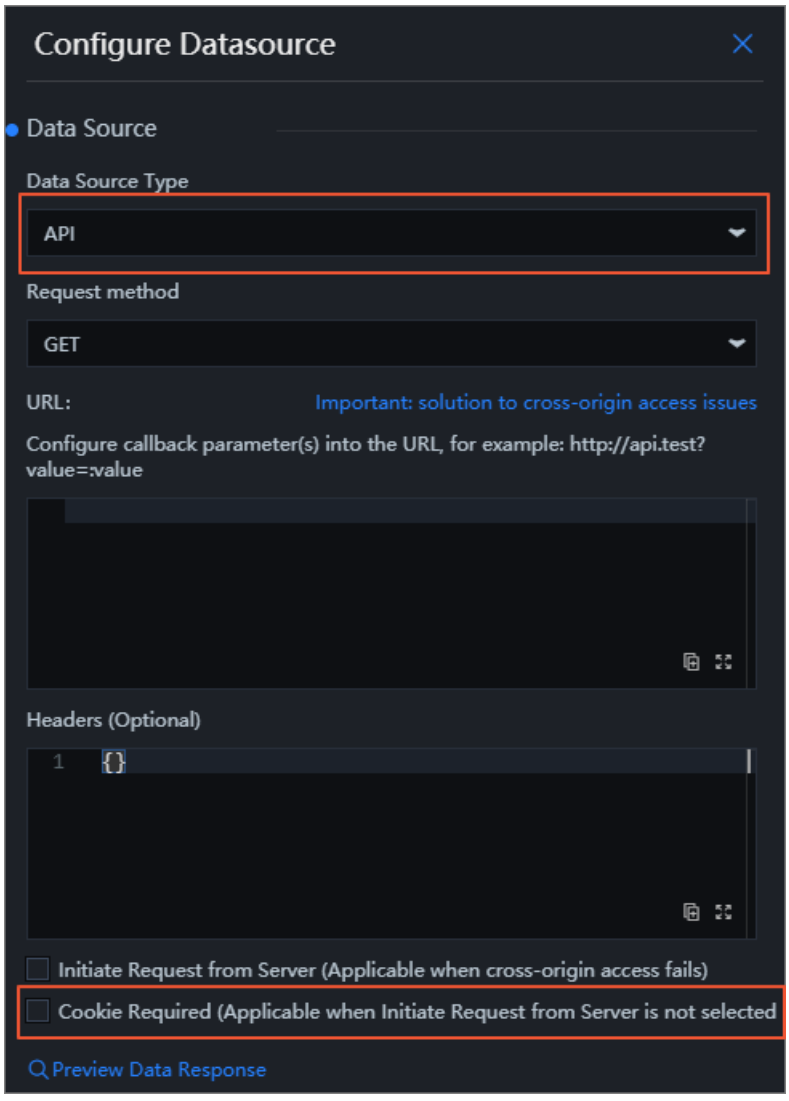
For more information, see HTTP access control (CORS).

3.Use cookies to isolate data of a DataV project

If you use an API data source in a DataV project, you can configure cookies to allow different users to view their own data after they log on to the system. This ensures data security in the project.

How it works

1. If you embed a DataV project into the web page of your business system by using an iframe, the web page requests contain user logon information, for example, Session_Id in cookies when a user logs on.
2. If you use an API data source and select **Cookie Required (Applicable when Initiate Request from Server is not selected and cookies are required)**, DataV adds cookies in HTTP requests. The cookies contain user logon information.



> **Note** If cross-origin requests are required, select Initiate Request from Server (Applicable when cross-origin access fails). For more information, see Cross-origin data configuration.

3. When users log on and request the DataV project, the server verifies user information in the requests and returns data to users based on the information. Individual users can view only their own data.

4.Use DataV Proxy

DataV Proxy provides a graphic user interface to configure DataV data proxies. You can use it to configure data queries without the need to access databases or compile APIs. This topic describes how to use DataV Proxy.

How it works

1. The DataV Proxy application obtains the encrypted SQL query strings and database IDs.
2. The application connects to the database to query data.
3. The application returns the results to the DataV console.

Start the DataV Proxy application

- Windows:
  i. Download the DataV Proxy software package.
  ii. Decompress the package and double-click the *datav_proxy_wins.exe* file.
  iii. Sign up and log on to the DataV Proxy application.
  iv. In the application, add a data source and query logs. For more information, see Configure the DataV Proxy application.
- Linux or macOS:
  i. Run the following command to download the DataV Proxy software package:

  ```
  wget https://sh-conf.oss-cn-shanghai.aliyuncs.com/doc_files/datav_proxy.zip
  ```

  ii. Decompress the package to the directory of the project.
  iii. Run the following commands to start DataV Proxy (port 8001 is used by default):

  ```
  chmod 777 ./*
  sh exec.sh start
  ```

  You can also run the **sh exec.sh start -p [Port number]** command to start DataV Proxy by using the specified port, for example, **sh exec.sh start -p 8080**.

  You can run the **sh exec.sh stop** command to stop DataV Proxy.

  iv. After you start DataV Proxy, access `http://Domain name or IP address:Port` in the browser.
    - Set **Domain name or IP address** to the public IP address of the server where DataV Proxy is installed.
    - Set **Port** to the port number used to start DataV Proxy.
  v. Sign up and log on to the DataV Proxy application.
  vi. In the application, add a data source and query logs. For more information, see Configure the DataV Proxy application.

Configure the DataV Proxy application

1. Access `http://{IP address or domain name of the DataV Proxy server}:Port number`, for example, `http://12.12.12.12:8001`. The DataV Proxy configuration page appears.

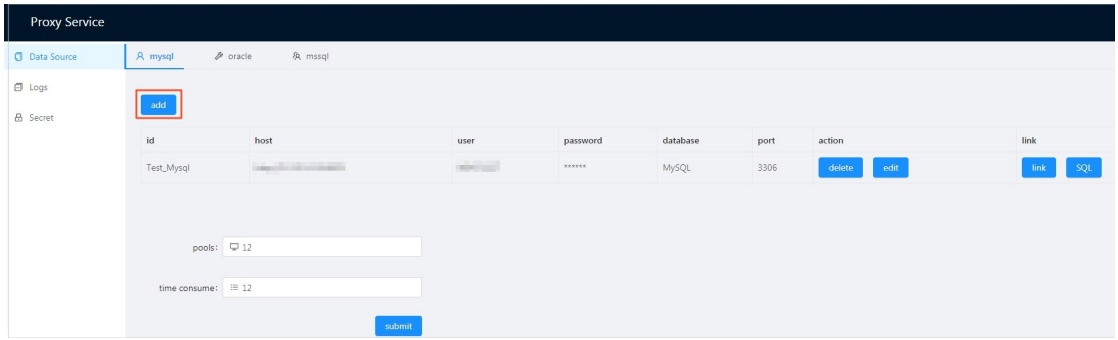
> **Note** You need to sign up and log on to the DataV Proxy application for the first access.

2. Configure a data source.

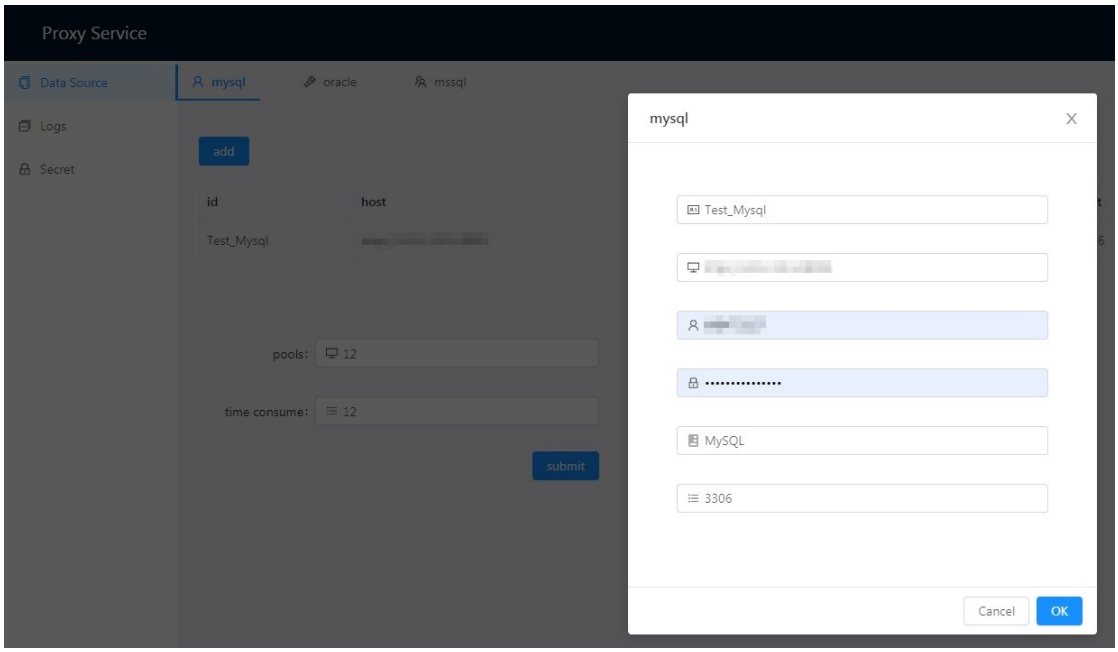
   Configure the basic information of the database. After the configuration is complete, you can **test the database connectivity** and **SQL query operations** to ensure reliability of the data source.

   > **Notice** Before you add a data source, add the `IP address or domain name of the DataV Proxy server` in step 1 to the whitelist of your database, for example, an ApsaraDB RDS for MySQL instance. For more information, see Use a database client or the CLI to connect to an ApsaraDB RDS for MySQL instance.

   i. Click **Data Source**, select a data source type, and click **add**.



ii. In the dialog box that appears, enter information of the data source.

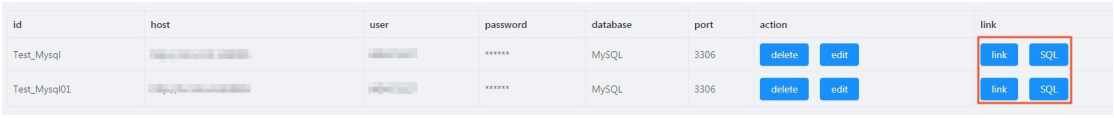


Parameter	Description
id	The ID of the data source. This parameter can be customized and must be unique.
host	The IP address or domain name of the server where the database resides. For example, if the data source is an ApsaraDB RDS for MySQL instance, set this parameter to the **external endpoint** of the instance. You can obtain the endpoint on the **Basic Information** page of the ApsaraDB for RDS console, for example, **rm-bp1xxxxxxxxxxxxhmo.mysql.rds.aliyuncs.com**.
user	The username used to log on to the database.
password	The password used to log on to the database.
database	The name of the database.
port	The port number used to connect to the database. For example, the port number used to connect to ApsaraDB RDS MySQL instances is **3306** by default.

iii. Test the configured data source.

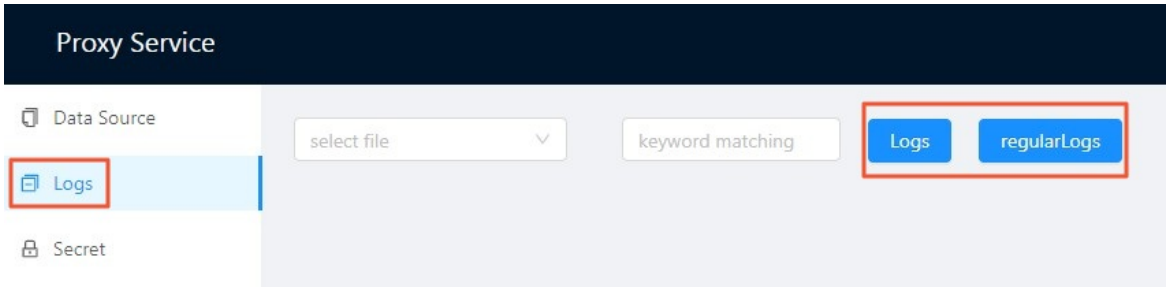
Click **link** to test the connectivity of the database.

Click **SQL**. In the dialog box that appears, enter an SQL statement to test data in the database.



3. Query logs.

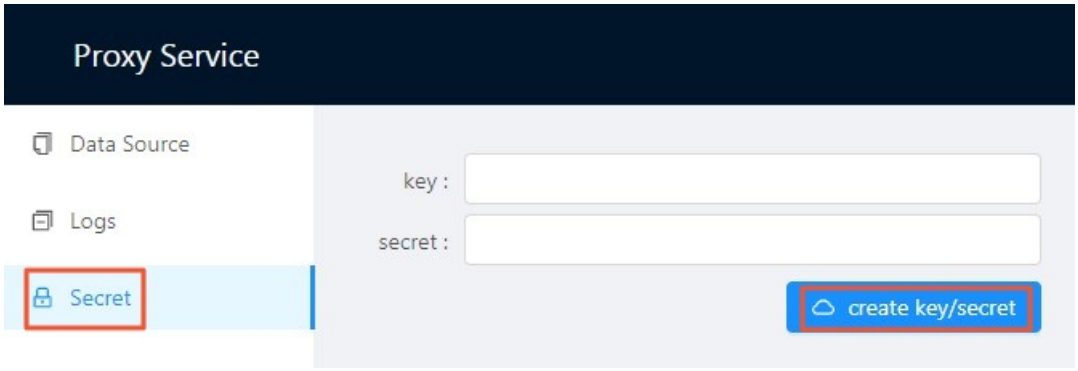
Specify keywords and a range of log rows to query the logs.



Notice In Windows, you can only view the logs but cannot query them based on keywords or the specified range of log rows.

4. Generate a key.

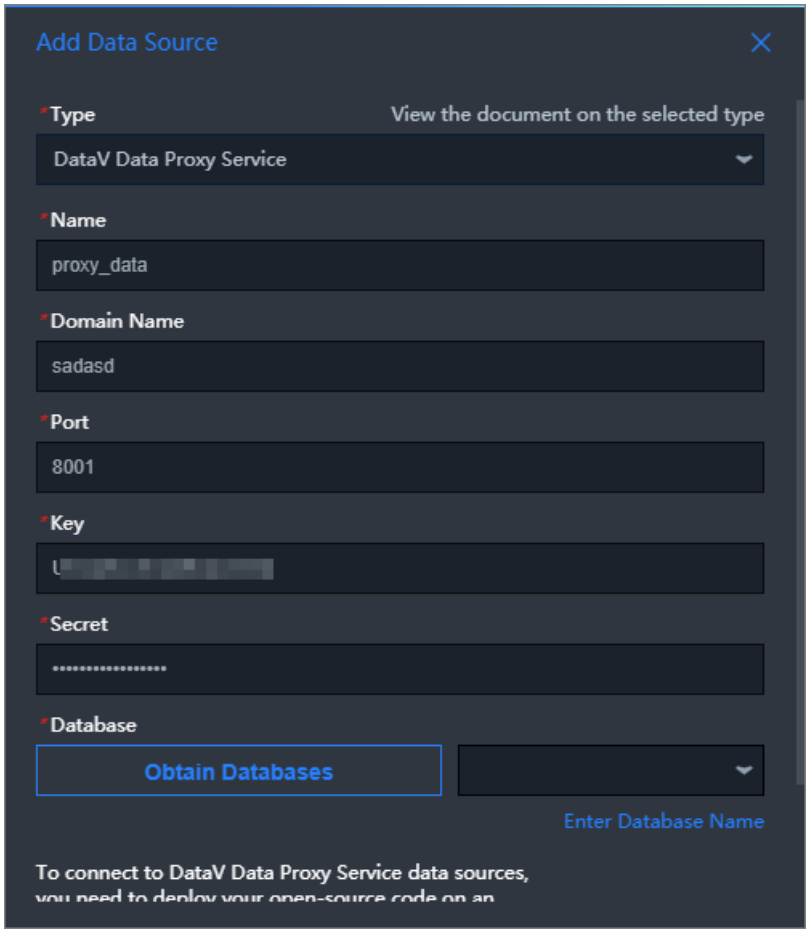
Click **create key/secret**. A **key** and a **secret** are generated, and the previous key becomes invalid.



Notice The key is empty by default.

Use the DataV Proxy application

1. Log on to the DataV console.
2. Navigate to **Data Sources > Add Source**.
3. In the **Add Data Source** dialog box that appears, specify the required information.



Parameter	Description
Type	Select DataV Data Proxy Service.
Name	The name of the data source. You can enter a name as needed.
Domain Name	The IP address or domain name of the DataV Proxy server. You can obtain the value from the Configure the DataV Proxy application section.
Port	The port used to start DataV Proxy. You can obtain the value from the Configure the DataV Proxy application section. By default, the port is **8001**.
Key	Enter the **key** generated in step 4 in the Configure the DataV Proxy application section.
Secret	Enter the **secret** generated in step 4 in the Configure the DataV Proxy application section.
Database	The ID of the data source that is added to the DataV Proxy application.

After the preceding information is specified, DataV automatically checks the connectivity of the data source.

4. Click **Obtain Databases** and select a database as the data source from the **database list**.

   > **Note** If the ID of the data source that is added to the DataV Proxy application is displayed in the list after you click **Obtain Databases**, the connection between DataV and the data source is established.

5. Click **OK**.

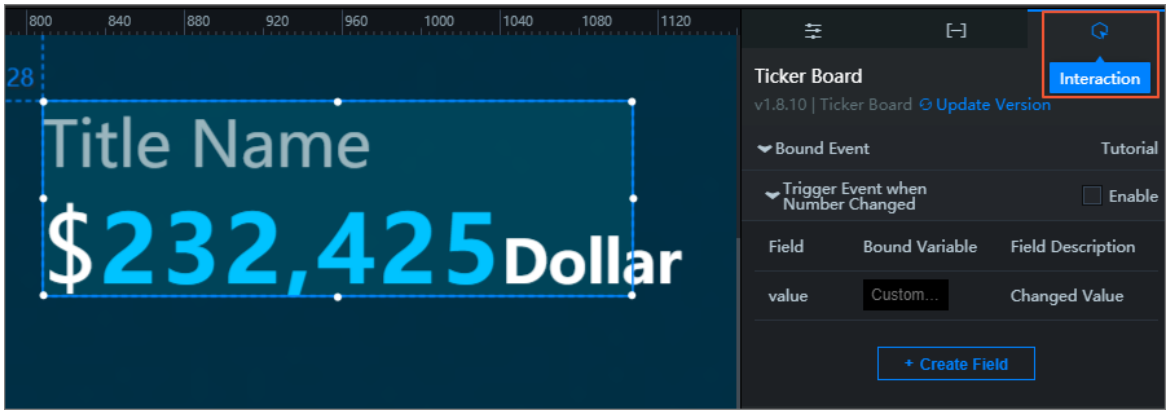
   The data source is added. You can configure it in your widgets. For more information, see Configure widget data.

5.Configure a callback ID for a ticker board

A callback ID in DataV is a parameter that a widget sends to another widget when the first widget responds to operations or when an update is automatically triggered. The parameter is a variable used to query data of one widget in another. This topic uses a **ticker board** widget as an example to demonstrate the use of callback IDs.

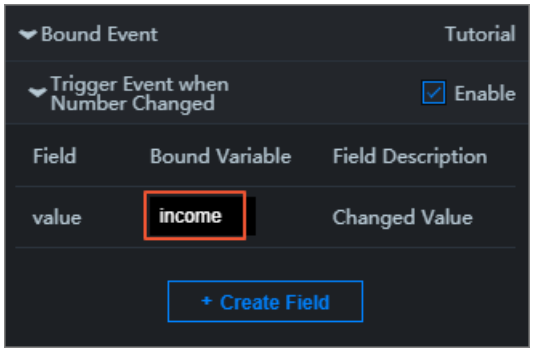
Procedure

1. Click the ticker board widget, and then click the **Interaction** tab on the right-side panel.



> **Note** You can configure callback IDs on the **Interaction** tab on the right-side panel.

2. Select the **Enable** check box to the right of **Trigger Event when Number Changed**.
3. In the **Bound Variable** column, change the variable name from *value* to *income*.



> **Note**
>
> - After you change the variable name to *income*, you can use *income* to obtain the parameter value in another widget that calls the callback ID of this widget.
> - You can configure different variable names in different widgets to differentiate parameters.

4. Click the ticker board widget. On the Data tab on the right-side panel, click Modify next to Data Source and use the callback ID that you have configured in the format of `:Variable name`, for

example, `:income`.

Examples:

- SQL (Use one of the following two methods.)

```
select :income as value
```

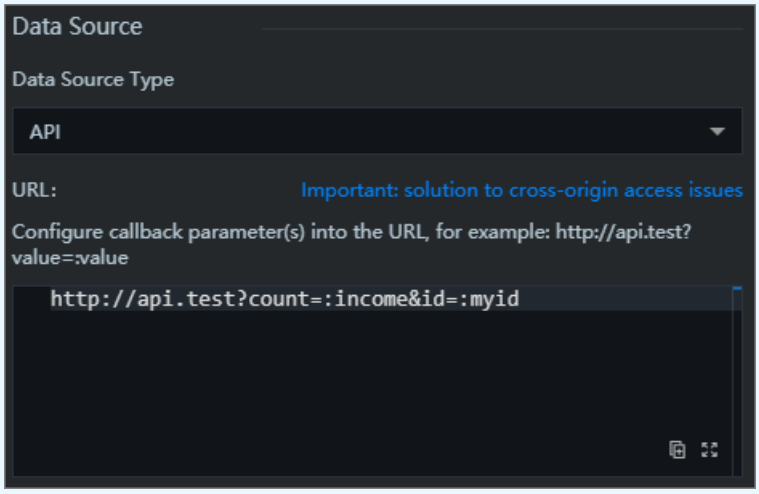
```
select A from table where count = :income
```

income is the callback ID, value is the field you specified to receive the value of the callback ID, and count and A are the fields in your database.

- API

```
http://api.test?count=:income&id=:myid
```

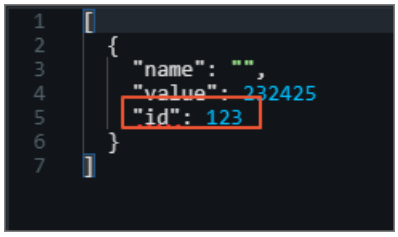
income and myid are the callback IDs, and count and id are the fields in your database.

> **Note**
>
> - If the **data source type** of your widget is **Static Data** or **CSV File**, callback IDs are not supported.
> - DataV automatically completes the callback IDs. When you configure the data source, enter `:`, and the editor lists callback IDs configured for the widget. You can use the up and down arrow keys to select a callback ID and then press **Enter**. This helps you quickly find a callback ID when there are a lot of interactive widgets in the project.
>
> 
>

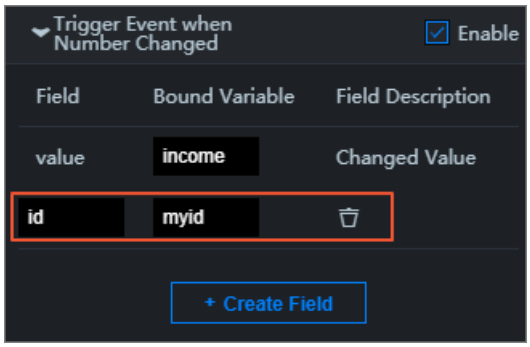
Advanced features

Configure a custom field

1. Click the ticker board widget, and then click the **Data** tab on the right-side panel.
2. Add an id field and set its value to *123*.



3. Click the **Interaction** tab.
4. Click **Create Field**.
5. In the **Field** column, enter id. In the **Bound Variable** column, enter a variable name.



> **Note** You must specify values in both the **Field** and **Bound Variable** columns so that the variable can take effect.

Set a default value for the callback ID

Specify a request parameter in the URL to set a default value for the callback ID. Example:

```
http://datav.aliyun.com/screen/000000?myid=123
```

`000000` is the project ID.

When you access the project by using this URL, the callback ID *myid* is automatically set to *123*.

Multiple callback IDs are separated with ampersands (&). The following example sets the default values for both the callback IDs *myid* and *income*:

```
http://datav.aliyun.com/screen/000000?myid=123&income=1000
```

`000000` is the project ID.

FAQ

I have configured a callback ID in the project editor page. What do I do if data query fails because the system cannot obtain the callback value based on the callback ID?

This error occurs because the editor cannot trigger an event to obtain the callback value. You can view the callback value by using one of the following methods:

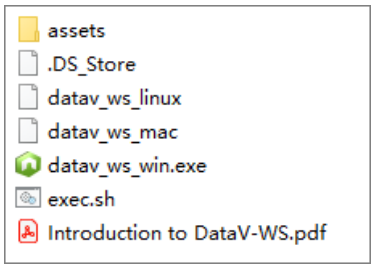
- Preview the project. View the callback value in the URL, or click the Network tab in the browser console and obtain the value in the response.
- Edit the project. Add the callback ID and value to the end of the URL:

```
https://datav.aliyun.com/admin/screen/99999?Callback ID=Callback value
```

6.Use the DataV-WS service

DataV-WS integrates the static file service and WebSocket. The static file service is used to load local model addresses. WebSocket can be used in the blueprint editor to connect multiple projects in the same network range without additional development. This topic describes how to use DataV-WS.

DataV-WS document directories



Notice Do not delete the `assets` directory.

Start and stop DataV-WS

- Windows:
  i. Download the DataV-WS installation package.
  ii. Decompress the package and double-click the *datav_ws_win.exe* file.
  iii. View the returned message. If the following information appears, DataV-WS is started:

  ```
  Server listen on IP address:8181 (for example, 127.0.0.1:8181)
  ```

  Notice Do not close the command prompt after the start.

  iv. Develop and demonstrate WebSocket and the static file service after you start DataV-WS. For more information, see Configure DataV-WS.
- Linux or macOS:
  i. Run the following command to download the DataV-WS installation package:

  ```
  wget http://sh-conf.oss-cn-shanghai.aliyuncs.com/doc_files/datav_ws.zip
  ```

  ii. Decompress the package, start the terminal, and enter the service directory, for example, datav-ws.
  iii. Run the following commands to start DataV-WS (port 8001 is used by default):

  ```
  chmod 777 ./
  sh exec.sh start
  ```

  You can run the **sh exec.sh stop** command to stop DataV-WS.

  You can run the **sh exec.sh restart** command to restart DataV-WS.

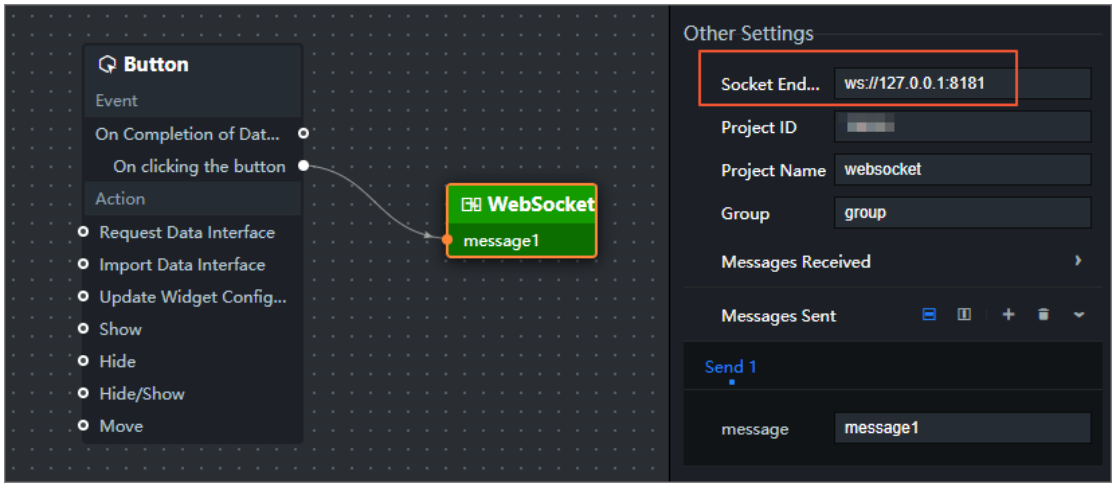
  iv. Develop and demonstrate WebSocket and static file service after you start DataV-WS. For more information, see Configure DataV-WS.

Configure DataV-WS

You can configure WebSocket and the static file service.

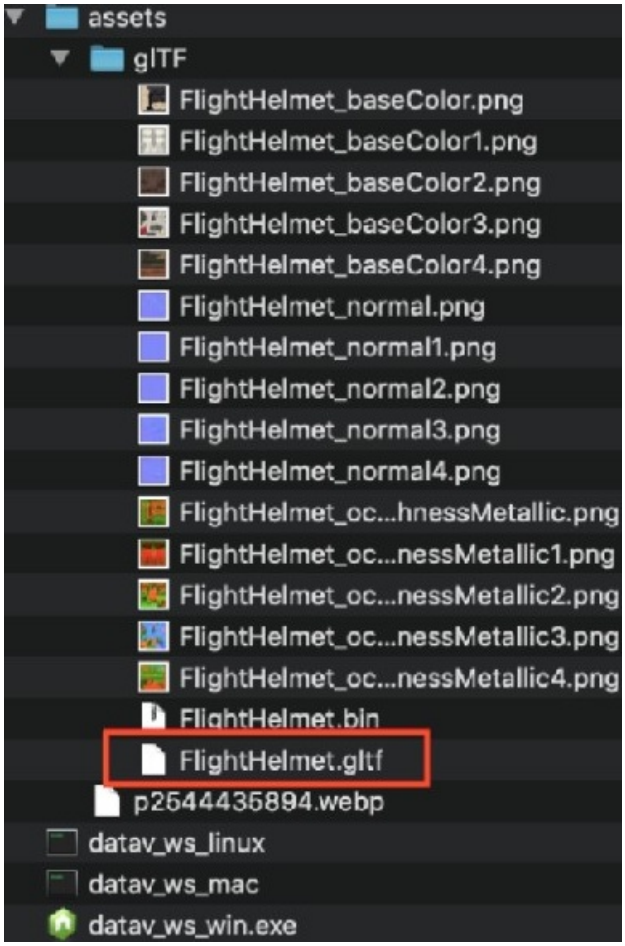
- Configure WebSocket.
  - Development:
    a. Start DataV-WS on your local computer. For more information, see Start and stop DataV-WS.
    b.
    c. Create an empty DataV project.
    d. In the upper-left corner of the **canvas editor**, click the **Blueprint Editor** icon. The **blueprint editor** appears.

    e. In the **blueprint editor**, drag and drop the **WebSocket** node to the canvas.
    f. Click the **WebSocket** node in the blueprint editor and set **Socket Endpoint** to `ws://127.0.0.1:8181`.



  - Demonstration:
    a. Start DataV-WS on a demonstration server that is in the same network range as the development server. For more information, see Start and stop DataV-WS.
    b. Add a **WebSocket** node. Set **Socket Endpoint** of the **WebSocket** node to `ws://IP address of the demonstration server:8181`. The projects that use the two WebSocket nodes can communicate with each other.

- Configure the static file service.
  i. Start DataV-WS on your local computer. For more information, see Start and stop DataV-WS.
  ii. Add static files (such as images) to the *assets* directory.



iii.

iv. Create an empty DataV project.

v. In the canvas editor, add an **image** widget. For more information, see Add a widget.

vi. Set **Background Image** to `http://IP address:8181/p2544435894.webp` so that the project can obtain the image from DataV-WS.