

DolphinScheduler Seatunnel plugin form configuration scheme

General idea

In DolphinScheduler, there are two configuration options for data integration plugins such as DATAX and SQOOP, one is to customize the template, and the other is to generate the configuration file in the back end by clicking the form. But at present, SEATUNNEL plugin only supports custom configuration. Therefore, this pr tries to support SEATUNNEL plugin to generate configuration files through form mode, so as to lower the threshold of using SEATUNNEL plugin. You only need to click on the form to achieve data synchronization.

Achieve effect

The front-end form page contains the basic data item configuration, including the source and destination of the data, as well as other common configurable parameters (such as Parallelism). It also supports custom Transform content, which is spliced by the back-end program to generate the final Seatunnel configuration.

The screenshot shows a configuration form for the Seatunnel plugin. It includes the following elements:

- Parallelism:** A numeric input field with the value '1' and minus/plus buttons.
- Job Mode:** A dropdown menu currently set to 'BATCH'.
- Custom Config:** A toggle switch that is currently turned off.
- Data Source:** A section header with a blue vertical bar.
- Source Datasource Type *:** A dropdown menu set to 'MYSQL'.
- Source Datasource Instance *:** A dropdown menu set to 'Please Select'.
- Source Table Name *:** A dropdown menu set to 'Please Select'.
- Add Params:** A blue circular button with a white plus sign.

Data Target

Target Datasource Type *

Target Datasource Instance *

Target Table Name *

Add Params



Custom Transform



Custom Parameters



The data source instance filters the data source configured in the data source center module according to the data source type, and requests the back-end to automatically obtain all data tables under the current data source instance. You only need to select the source data table and the destination data table from the drop-down list.

Data Source

Source Datasource Type *

Source Datasource Instance *

Source Table Name *

- TablePartitionInfo
- active_user_info
- dashi_uid_active_dates
- data_connection

The form contains only fixed seatunnel configurations. If you need to add additional parameter configurations during the configuration process, you can click “Add Params” Configuration to add more configuration parameters supported by the Seatunnel Connector.

The following figure shows the effect of adding a custom attribute of "partition_column" under the MYSQL data type and generating the configuration file

Data Source


Source Datasource Type * Source Datasource Instance *

MYSQL mysql@mail_dw

Source Table Name *

mnt_flume

Add Params

partition_cc IN VARCH... id 

```
2025-02-04 18:58:56.149 INFO [exclusive-task-executor-container-worker-0] - Generate Seatunnel Config =>
env {
  job.mode = "BATCH"
  parallelism = 1
}
source {
  jdbc {
    url = "jdbc:mysql://10.224.192.136:3306/mail_dw?serverTimezone=GMT%2b8&useUnicode=true&characterEncoding=UTF-8"
    driver = "com.mysql.cj.jdbc.Driver"
    user = "mail_dev"
    password = "123"
    query = "select * from mnt_flume"
    partition_column = "id"
  }
}
transform {
  sql {
    query = "select id, module, ip from mnt_flume"
  }
}
sink {
  jdbc {
    url = "jdbc:mysql://10.224.192.136:3306/test?serverTimezone=GMT%2b8&useUnicode=true&characterEncoding=UTF-8"
    driver = "com.mysql.cj.jdbc.Driver"
    user = "mail_dev"
    password = "123456"
    database = "test"
    table = "mnt_flume"
    generate_sink_sql = "true"
  }
}
```

Back-end module

The back-end module mainly adds the function code of generating Seatunnel

configuration file according to the form parameters. The “Dolphinscheduler-task-seatunnel” module's “generator” folder contains templates for generating configuration files for different Connector type. The “parameter” folder contains the parameters of the Connector form.

Modify the 'buildOptions' method of the “SeatunnelTask” class to support configuration **in three different cases**:

1. Supports quick configuration based on form items. The back-end generates configuration content based on form parameters.
2. If the Customize Configuration button is displayed, the user-defined configuration content is used.
3. If the user-defined configuration button is displayed and a user selects a specified resource, the system uses the specified resource.

In the first case, the generated configuration is displayed in the log when the task is executed, so that users can check whether the generated configuration is different from the expected configuration and adjust the task easily.

The results generated during the test are shown below, using MYSQL data types as an example:

```
.....
2025-02-01 23:37:42.745 INFO [exclusive-task-executor-container-worker-0] - Generate Seatunnel Config =>
env {
  job.mode = "BATCH"
  parallelism = 1
}
source {
  Jdbc {
    url = "jdbc:mysql://192.168.1.10:3306/mail_dw?serverTimezone=GMT%2b8&useUnicode=true&characterEncoding=UTF-8"
    driver = "com.mysql.cj.jdbc.Driver"
    user = "root"
    password = "123"
    query = "select * from mail_dw_data"
  }
}
sink {
  jdbc {
    url = "jdbc:mysql://192.168.1.10:3306/test?serverTimezone=GMT%2b8&useUnicode=true&characterEncoding=UTF-8"
    driver = "com.mysql.cj.jdbc.Driver"
    user = "root"
    password = "123"
    database = "test"
    table = "mail_dw_data"
    generate_sink_sql = "true"
  }
}
.....
```

The configuration generated after adding a custom Transform is as follows:

```

*****
2025-02-02 11:04:42.420 INFO [exclusive-task-executor-container-worker-0] - Generate Seatunnel Config =>
env {
  job.mode = "BATCH"
  parallelism = 1
}
source {
  Jdbc {
    url = "jdbc:mysql://192.168.1.100:3306/mail_dw?serverTimezone=GMT%2b8&useUnicode=true&characterEncoding=UTF-8"
    driver = "com.mysql.cj.jdbc.Driver"
    user = "root"
    password = "123"
    query = "select * from mnt_flume"
  }
}
transform {
  sql {
    query = "select id, module, ip from mnt_flume"
  }
}
sink {
  jdbc {
    url = "jdbc:mysql://192.168.1.100:3306/test?serverTimezone=GMT%2b8&useUnicode=true&characterEncoding=UTF-8"
    driver = "com.mysql.cj.jdbc.Driver"
    user = "root"
    password = "123456"
    database = "test"
    table = "mnt_flume"
    generate_sink_sql = "true"
  }
}

```

The above configuration was successfully executed in the local test and written to the target MYSQL data table.

Conclusion

The code submitted by pr this time belongs to the functional code that I initially conceived and developed. If there are any inadequacies or other defects, please communicate and solve them together. Thanks for review.