

Flink

Apache Flink Component

Available as of Camel 2.18

This documentation page covers the <https://flink.apache.org> component for the Apache Camel.

The **camel-flink** component provides a bridge between Camel connectors and Flink tasks. This Camel Flink connector provides a way to route message from various transports, dynamically choosing a flink task to execute, use incoming message as input data for the task and finally deliver the results back to the Camel pipeline.

Maven users will need to add the following dependency to their pom.xml for this component:

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-flink</artifactId>
  <version>x.x.x</version>
  <!-- use the same version as your Camel core version -->
</dependency>
```

Currently, the Flink Component supports only Producers. One can create DataSet, DataStream jobs.

URI format

```
flink:dataset?dataset=#myDataSet&dataSetCallback=#dataSetCallback
flink:datastream?datastream=#myDataStream&dataStreamCallback=#dataStreamCallback
```

You can append query options to the URI in the following format, ?option=value&option=value&...

FlinkEndpoint Options

Name	Default Value	Description
endpointType		Required Type of the endpoint (dataset datastream).
collect	true	Indicates if results should be collected or counted.
dataSet		DataSet to compute against.
dataSetCallback		Function performing action against a DataSet.
dataStream		DataStream to compute against.
dataStreamCallback		Function performing action against a DataStream.
exchangePattern	InOnly	Sets the default exchange pattern when creating an exchange
synchronous		Sets whether synchronous processing should be strictly used or Camel is allowed to use asynchronous processing (if supported).

FlinkComponent Options

Name	Default Value	Description
dataSet		DataSet to compute against.
dataStream		DataStream to compute against.
dataSetCallback		Function performing action against a DataSet.
dataStreamCallback		Function performing action against a DataStream.

Examples

Flink DataSet Callback

```
@Bean
public DataSetCallback<Long> dataSetCallback() {
    return new DataSetCallback<Long>() {
        public Long onDataSet(DataSet dataSet, Object... objects) {
            try {
                dataSet.print();
                return new Long(0);
            } catch (Exception e) {
                return new Long(-1);
            }
        }
    };
}
```

Flink DataStream Callback

```
@Bean
public VoidDataStreamCallback dataStreamCallback() {
    return new VoidDataStreamCallback() {
        @Override
        public void doOnDataStream(DataStream dataStream, Object... objects) throws Exception {
            dataStream.flatMap(new Splitter()).print();
            environment.execute("data stream test");
        }
    };
}
```

Flink Producer

```
CamelContext camelContext = new SpringCamelContext(context);
String pattern = "foo";
try {
    ProducerTemplate template = camelContext.createProducerTemplate();
    camelContext.start();
    Long count = template.requestBody("flink:dataSet?dataSet=#myDataSet&dataSetCallback=#countLinesContaining",
    pattern, Long.class);
} finally {
    camelContext.stop();
}
```

See Also

- [Configuring Camel](#)
- [Component](#)
- [Endpoint](#)
- [Getting Started](#)