
toc: true title: 《从0到1学习Flink》—— Data Sink 介绍 date: 2018-10-29 tags:

- Flink
 - 大数据
 - 流式计算
-



前言

再上一篇文章中 [《从0到1学习Flink》—— Data Source 介绍](#) 讲解了 Flink Data Source ，那么这里就来讲讲 Flink Data Sink 吧。

首先 Sink 的意思是：

yd sink



**vi. 下沉；消沉；渗透**
Search for 'yd sink' with default fallback

**vt. 使下沉；挖掘；使低落**
sɪŋk; [UK] sɪŋk; [US] sɪŋk

**n. 水槽；洗涤槽；污水坑**
sɪŋk; [UK] sɪŋk; [US] sɪŋk

**n. (Sink)人名；(英、瑞典)辛克**
sɪŋk; [UK] sɪŋk; [US] sɪŋk

**水槽**
翻译结果

**网络释义**
有道词典 for Alfred

**sink**
下沉；盥洗盆；洗碗池；洗涤槽

**sink in**
被了解；完全理解；被完全理解；被理解

**advise sink**
通知接收；告诉接受；表示已连接的通知接收

⌘2

⌘3

⌘4

⌘5

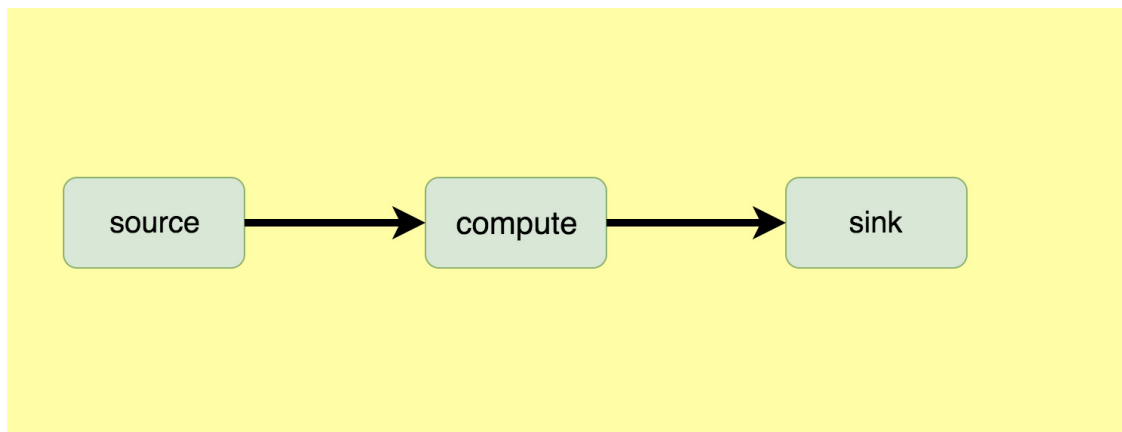
⌘6

⌘7

⌘8

⌘9

大概可以猜到了吧！Data sink 有点把数据存储下来（落库）的意思。



如上图，Source 就是数据的来源，中间的 Compute 其实就是 Flink 干的事情，可以做一系列的操作，操作完后就把计算后的数据结果 Sink 到某个地方。（可以是 MySQL、ElasticSearch、Kafka、Cassandra 等）。这里我说下自己目前做告警这块就是把 Compute 计算后的结果 Sink 直接告警出来了（发送告警消息到钉钉群、邮件、短信等），这个 sink 的意思也不一定非得说成要把数据存储到某个地方去。其实官网用的 Connector 来形容要去的地方更合适，这个 Connector 可以有 MySQL、ElasticSearch、Kafka、Cassandra RabbitMQ 等。

Flink Data Sink

前面文章 [《从0到1学习Flink》—— Data Source 介绍](#) 介绍了 Flink Data Source 有哪些，这里也看看 Flink Data Sink 支持的有哪些。

Application Development

- Basic API Concepts
- Streaming (DataStream API)**
 - Overview
 - Event Time
 - State & Fault Tolerance
 - Operators
- Connectors**
 - Overview
 - Fault Tolerance Guarantees
 - Kafka
 - Cassandra
 - Kinesis
 - Elasticsearch
 - Rolling File Sink
 - Streaming File Sink
 - RabbitMQ
 - NiFi
 - Twitter
 - Side Outputs
 - Python API
 - Testing
 - Experimental Features
- Batch (DataSet API)
- Table API & SQL
- Data Types & Serialization
- Managing Execution
- Libraries
- Best Practices
- API Migration Guides
- Deployment & Operations

Predefined Sources and Sinks

A few basic data sources and sinks are built into Flink and are always available. The predefined data sources include reading from files, directories, and sockets, and ingesting data from collections and iterators. The predefined data sinks support writing to files, to stdout and stderr, and to sockets.

Bundled Connectors

Connectors provide code for interfacing with various third-party systems. Currently these systems are supported:

- Apache Kafka (source/sink)
- Apache Cassandra (sink)
- Amazon Kinesis Streams (source/sink)
- Elasticsearch (sink)
- Hadoop FileSystem (sink)
- RabbitMQ (source/sink)
- Apache NiFi (source/sink)
- Twitter Streaming API (source)

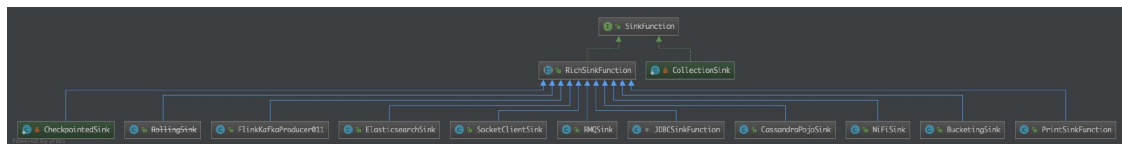
Keep in mind that to use one of these connectors in an application, additional third-party components are usually required, e.g. servers for the data stores or message queues. Note also that while the streaming connectors listed in this section are part of the Flink project and are included in source releases, they are not included in the binary distributions. Further instructions can be found in the corresponding subsections.

Connectors in Apache Bahir

Additional streaming connectors for Flink are being released through Apache Bahir, including:

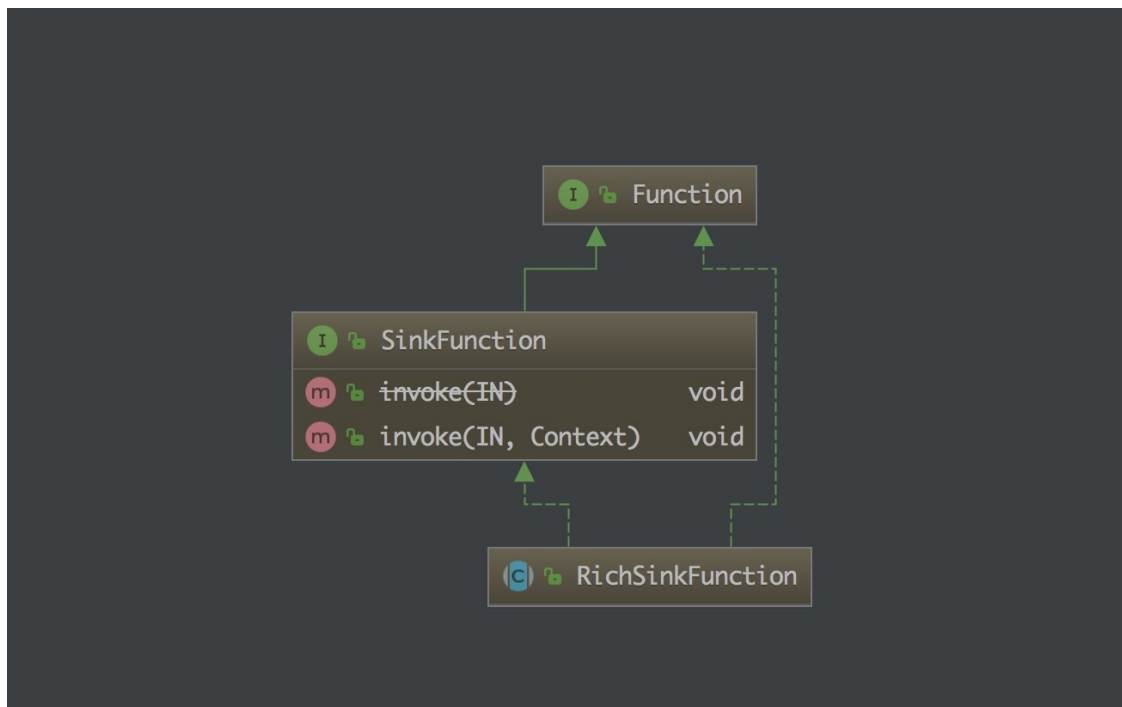
- Apache ActiveMQ (source/sink)
- Apache Flume (sink)
- Redis (sink)
- Akka (sink)
- Netty (source)

看下源码有哪些呢？



可以看到有 Kafka、ElasticSearch、Socket、RabbitMQ、JDBC、Cassandra POJO、File、Print 等 Sink 的方式。

SinkFunction



从上图可以看到 SinkFunction 接口有 invoke 方法，它有一个 RichSinkFunction 抽象类。

上面的那些自带的 Sink 可以看到都是继承了 RichSinkFunction 抽象类，实现了其中的方法，那么我们要自己定义自己的 Sink 的话其实也是要按照这个套路来做的。

这里就拿个较为简单的 PrintSinkFunction 源码来讲下：

```
@PublicEvolving
public class PrintSinkFunction<IN> extends RichSinkFunction<IN> {
    private static final long serialVersionUID = 1L;

    private static final boolean STD_OUT = false;
    private static final boolean STD_ERR = true;

    private boolean target;
    private transient PrintStream stream;
    private transient String prefix;

    /**
     * Instantiates a print sink function that prints to standard
     out.
     */
    public PrintSinkFunction() {}
```

```

    /**
     * Instantiates a print sink function that prints to standard
    out.
     *
     * @param stderr True, if the format should print to standard
    error instead of standard out.
     */
    public PrintSinkFunction(boolean stderr) {
        target = stderr;
    }

    public void setTargetToStandardOut() {
        target = STD_OUT;
    }

    public void setTargetToStandardErr() {
        target = STD_ERR;
    }

    @Override
    public void open(Configuration parameters) throws Exception {
        super.open(parameters);
        StreamingRuntimeContext context = (StreamingRuntimeContext)
    getRuntimeContext();
        // get the target stream
        stream = target == STD_OUT ? System.out : System.err;

        // set the prefix if we have a >1 parallelism
        prefix = (context.getNumberOfParallelSubtasks() > 1) ?
            ((context.getIndexOfThisSubtask() + 1) + "> ") :
    null;
    }

    @Override
    public void invoke(IN record) {
        if (prefix != null) {
            stream.println(prefix + record.toString());
        }
        else {
            stream.println(record.toString());
        }
    }

    @Override
    public void close() {
        this.stream = null;
        this.prefix = null;
    }

```

```

    }

    @Override
    public String toString() {
        return "Print to " + (target == STD_OUT ? "System.out" :
"System.err");
    }
}

```

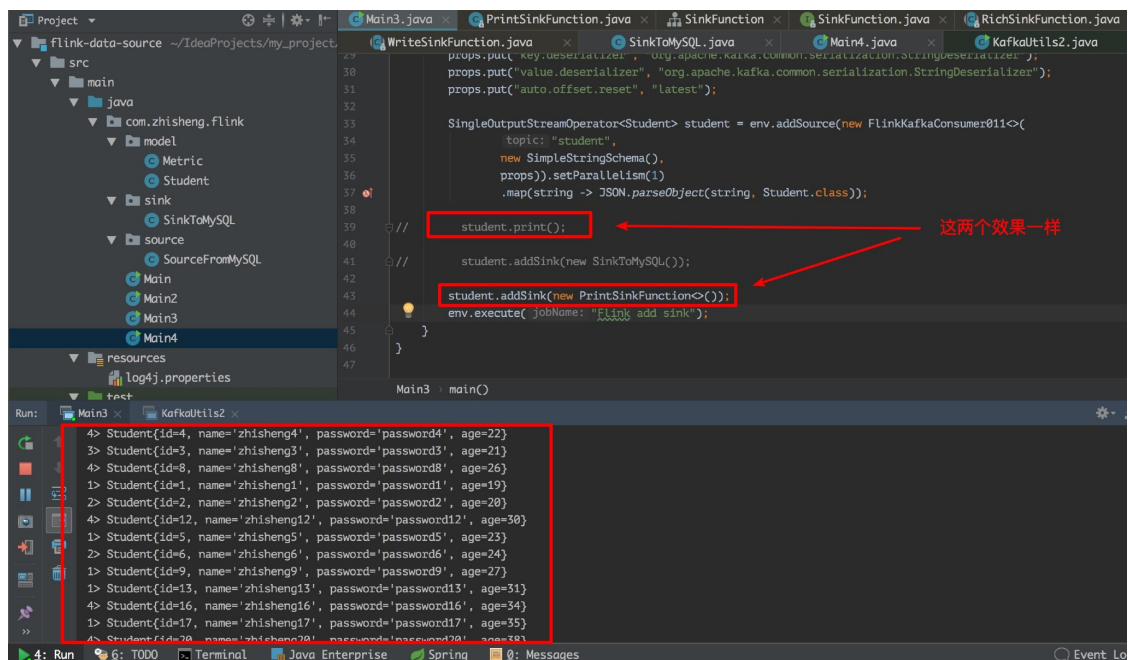
可以看到它就是实现了 RichSinkFunction 抽象类，然后实现了 invoke 方法，这里 invoke 方法就是把记录打印出来了就是，没做其他的额外操作。

如何使用？

```
SingleOutputStreamOperator.addSink(new PrintSinkFunction<>());
```

这样就可以了，如果是其他的 Sink Function 的话需要换成对应的。

使用这个 Function 其效果就是打印从 Source 过来的数据，和直接 Source.print() 效果一样。



下篇文章我们将讲解下如何自定义自己的 Sink Function，并使用一个 demo 来教大家，让大家知道这个套路，且能够在自己工作中自定义自己需要的 Sink Function，来完成自己的工作需求。

最后

本文主要讲了下 Flink 的 Data Sink，并介绍了常见的 Data Sink，也看了下源码的 SinkFunction，介绍了一个简单的 Function 使用，告诉了大家自定义 Sink Function 的套路，下篇文章带大家写个。

关注我

转载请务必注明原创地址为：<http://www.54tianzhisheng.cn/2018/10/29/flink-sink/>

另外我自己整理了些 Flink 的学习资料，目前已经全部放到微信公众号了。你可以加我的微信：zhisheng_tian，然后回复关键字：Flink 即可无条件获取到。



Github 代码仓库

<https://github.com/zhisheng17/flink-learning/>

以后这个项目的所有代码都将放在这个仓库里，包含了自己学习 flink 的一些 demo 和博客

相关文章

- 1、《从0到1学习Flink》—— Apache Flink 介绍
- 2、《从0到1学习Flink》—— Mac 上搭建 Flink 1.6.0 环境并构建运行简单程序入门
- 3、《从0到1学习Flink》—— Flink 配置文件详解
- 4、《从0到1学习Flink》—— Data Source 介绍
- 5、《从0到1学习Flink》—— 如何自定义 Data Source ?
- 6、《从0到1学习Flink》—— Data Sink 介绍
- 7、《从0到1学习Flink》—— 如何自定义 Data Sink ?
- 8、《从0到1学习Flink》—— Flink Data transformation(转换)
- 9、《从0到1学习Flink》—— 介绍Flink中的Stream Windows
- 10、《从0到1学习Flink》—— Flink 中的几种 Time 详解
- 11、《从0到1学习Flink》—— Flink 写入数据到 ElasticSearch