

# KYLIN-5417 实时自定义解析器使用手册

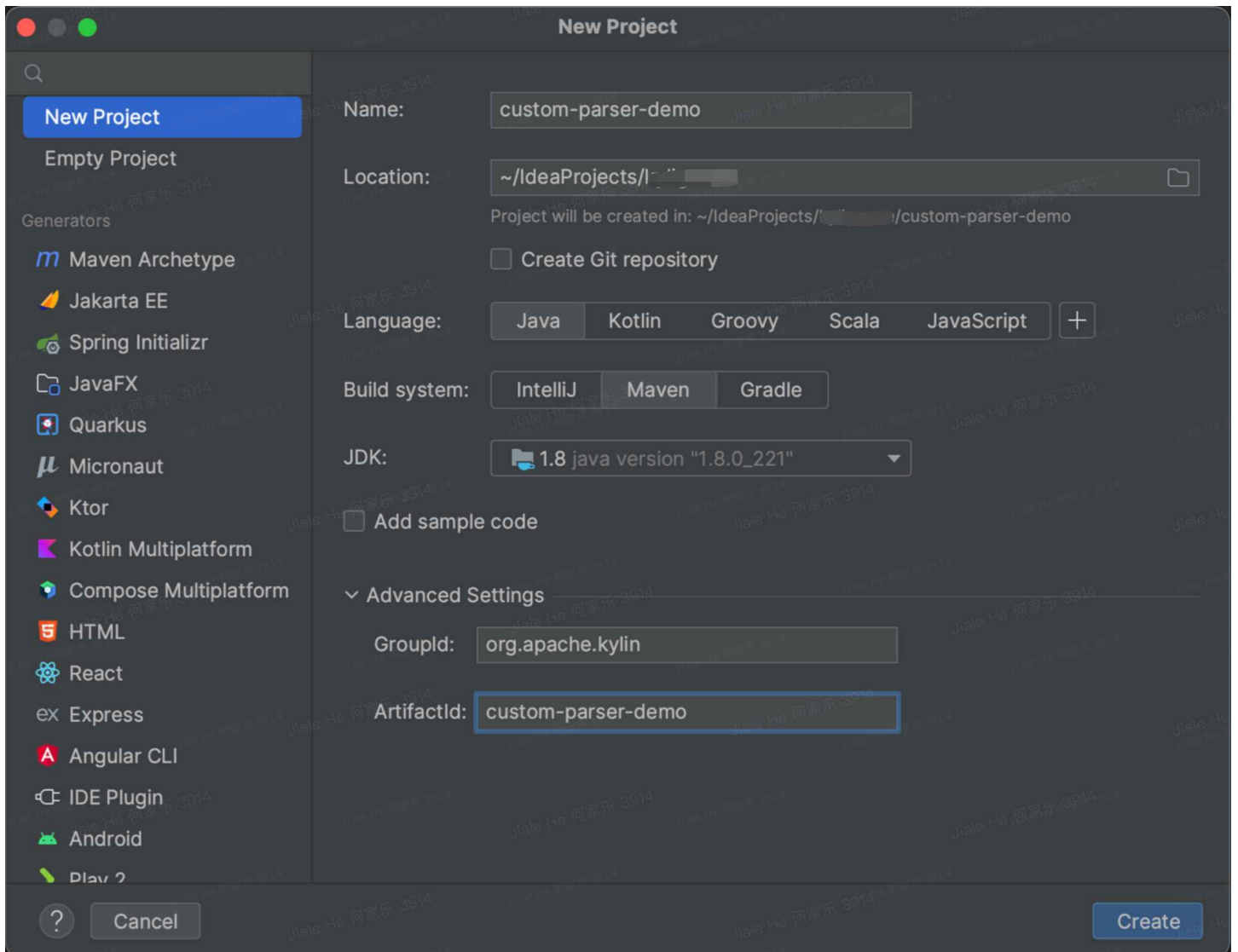
## SDK 的使用

### 1. 下载解析器 SDK

见附件 `kylin-streaming-sdk.jar`

### 2. 创建项目

- 请使用IDEA开发工具
- 请使用Maven管理依赖



### 3. 修改 pom.xml

pom.xml 中新增以下插件

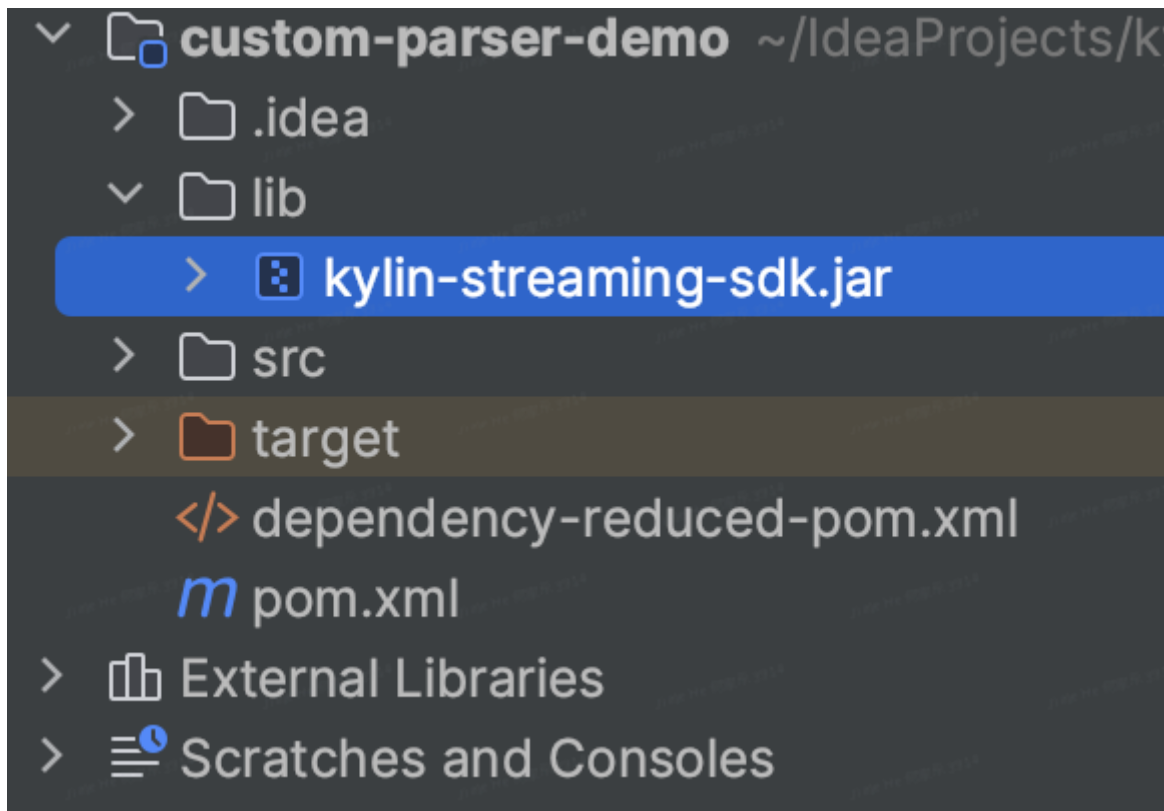
```
1 <build>
2   <plugins>
3     <plugin>
4       <groupId>org.apache.maven.plugins</groupId>
5       <artifactId>maven-compiler-plugin</artifactId>
6       <version>3.10.1</version>
7       <configuration>
8         <source>1.8</source>
9         <target>1.8</target>
10      </configuration>
11    </plugin>
12
13    <plugin>
14      <groupId>org.apache.maven.plugins</groupId>
15      <artifactId>maven-shade-plugin</artifactId>
16      <version>3.3.0</version>
17      <executions>
18        <execution>
19          <phase>package</phase>
20          <goals>
21            <goal>shade</goal>
22          </goals>
23          <configuration>
24            <finalName>${artifactId}</finalName>
25            <artifactSet>
26              <excludes>
27                <exclude>org.apache.kylin:kylin-streaming-sdk</e
28                <exclude>org.projectlombok:*</exclude>
29                <exclude>org.apache.commons:*</exclude>
30                <exclude>com.fasterxml.jackson.core:*</exclude>
31                <exclude>com.google.guava:*</exclude>
32                <exclude>org.slf4j:*</exclude>
33              </excludes>
34            </artifactSet>
35            <filters>
36              <filter>
37                <artifact>*:*</artifact>
38                <excludes>
39                  <exclude>META-INF/*.SF</exclude>
40                  <exclude>META-INF/*.DSA</exclude>
41                  <exclude>META-INF/*.RSA</exclude>
42                  <exclude>javax/annotation/**</exclude>
```

```
43         </excludes>
44     </filter>
45 </filters>
46 </configuration>
47 </execution>
48 </executions>
49 </plugin>
50 </plugins>
51 </build>
```

## 4. 导入 SDK

项目中导入SDK请使用 Maven 管理

在项目根目录下新建 **lib** 目录



### 1. 加载Jar到项目依赖

```
2 <properties>
3     <maven.compiler.source>8</maven.compiler.source>
4     <maven.compiler.target>8</maven.compiler.target>
5     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
6     <noop.version>1</noop.version>
7 </properties>
```

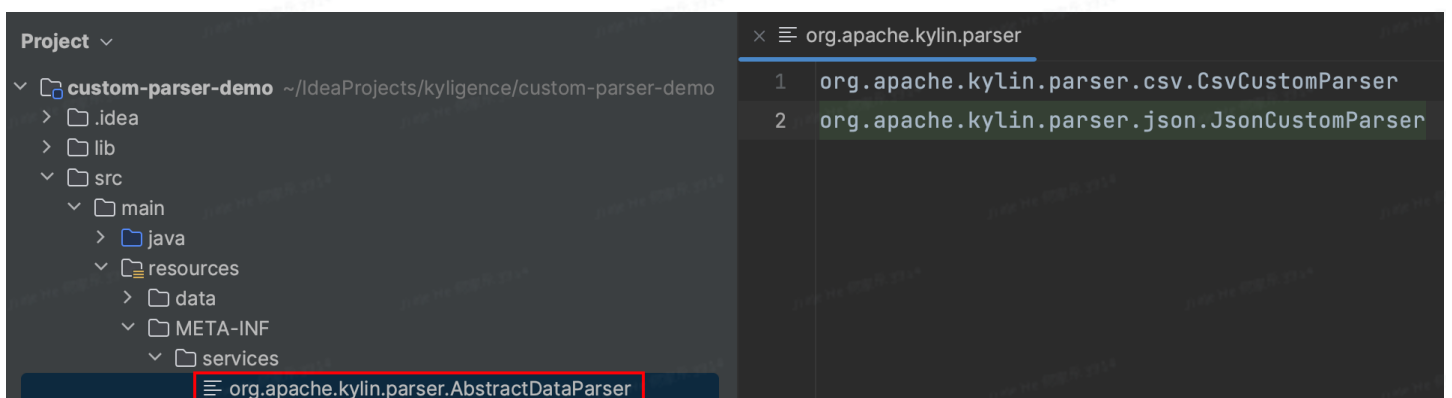
```

8
9 <dependency>
10     <groupId>org.apache.kylin</groupId>
11     <artifactId>kylin-streaming-sdk</artifactId>
12     <version>${noop.version}</version>
13     <scope>system</scope>
14     <systemPath>${project.basedir}/lib/kylin-streaming-sdk.jar</systemPath>
15 </dependency>
16
17 2. 解析器SDK自带以下依赖。如项目中也需用到此依赖，可以不用重复导入
18 <dependency>
19     <groupId>org.apache.commons</groupId>
20     <artifactId>commons-lang3</artifactId>
21     <version>3.10</version>
22 </dependency>

```

## 5. 自定义解析器

- 新建解析器类，`XXXParser extends AbstractDataParser<ByteBuffer>`
- `Override parse(ByteBuffer input)` 方法，在方法内解析单条数据，返回Map<字段名, 字段值>
- 如需有初始化动作需要在实例化解析器类时完成，请在**无参构造中完成**
- 如需在每条数据解析前，有初始化动作，请 `Override before()`
- **解析单条数据时抛出异常**，会在构建时被认定脏数据从而跳过该条数据的构建。
- 如需在每条数据处理后对数据做检查，请 `Override after()`
- 在项目 `${project.basedir}/src/resources/` 目录下新建 **META-INF/services/org.apache.kylin.parser.AbstractDataParser** 文件，并将每个解析器类的类路径填入其中。



# Demo 项目

见附件 `custom-parser-demo.zip`

可以尝试运行

- `org.apache.kylin.parser.json.JsonCustomParser#main`
- `org.apache.kylin.parser.csv.CsvCustomParser#main`

## 解析器性能测试

请使用 `org.apache.kylin.parser.utils.ParserBenchmark`

```
1 // parser BenchMark
2 System.out.printf("parser 20k data, cost: %s ms \n", ParserBenchmark.test20K(byte
3 System.out.printf("parser 40k data, cost: %s ms \n", ParserBenchmark.test40K(byte
4 System.out.printf("parser 60k data, cost: %s ms \n", ParserBenchmark.test60K(byte
5 System.out.printf("parser 999999 data, cost: %s ms \n", ParserBenchmark.testWith
```

## 1. 解析JSON

输入:

```
1 {
2   "name": "Li",
3   "sex": "man",
4   "age": 24,
5   "addr": {
6     "country": "China",
7     "city": "Shanghai",
8     "region": "YangPu"
9   },
10  "works": [
11    "work_1",
12    "work_2",
13    "work_3"
14  ],
15  "create_time": "2022-11-01 08:00:00",
16  "update_time": "2022-11-20 12:00:00"
17 }
```

输出:

```

1 {
2   "name": "Li",
3   "sex": "man",
4   "age": 24,
5   "addr_country": "China",
6   "addr_city": "Shanghai",
7   "addr_region": "YangPu",
8   // first_works 解析输入JSON中works数组的第一个元素
9   "first_works": "work_1",
10  "create_time": "2022-11-01 08:00:00",
11  "update_time": "2022-11-20 12:00:00",
12  // 该字段为解析器新增字段，代表解析的时间
13  "process_time": "2022-11-20 13:00:00"
14 }

```

## 2. 解析CSV ('|' 分割)

输入:

```

1 1|Li|"deve|loper"|Table tennis|2022-11-01 08:00:00|2022-11-02 08:00:00
2 2|He|developer|"Table|tennis"|2022-11-01 08:00:00|

```

输出:

```

1 {
2   "id" : 1,
3   "name" : "Li",
4   "job" : "Table tennis",
5   "sport" : "deve|loper",
6   "create_time" : "2022-11-01 08:00:00",
7   "delete_time" : "2022-11-02 08:00:00",
8   "process_time" : "2022-11-21 16:28:05"
9 }
10
11 {
12   "id" : 2,
13   "name" : "He",
14   "job" : "Table|tennis",
15   "sport" : "developer",

```

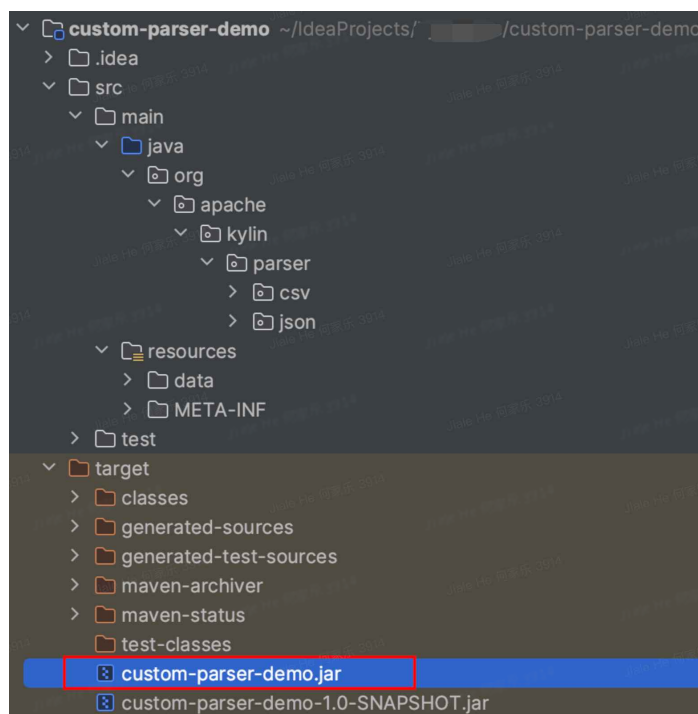
```
16 "create_time" : "2022-11-01 08:00:00",
17 "delete_time" : "",
18 "process_time" : "2022-11-21 16:29:22"
19 }
```

## 在KYLIN5中使用解析器

### 1. 解析器打包

```
1 mvn clean package -DskipTests
```

在项目 **target** 目录下生成 **custom-parser-demo.jar**



### 2. 上传解析器Jar包

调用上传Jar包Open Api，上传Jar文件，并获取解析器。

POST http://[redacted]/kylin/api/custom/jar

Params Authorization Headers (14) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> file	custom-parser-demo.jar ×
<input checked="" type="checkbox"/> project	parserdemo
<input checked="" type="checkbox"/> jar_type	STREAMING_CUSTOM_PARSER
Key	Value

Body Cookies (1) Headers (12) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "code": "000",
3   "data": [
4     "org.apache.kylin.parser.json.JsonCustomParser",
5     "org.apache.kylin.parser.csv.CsvCustomParser"
6   ],
7   "msg": ""
8 }
```

### 3. 创建 Kafka Topic

创建两个Topic

- custom\_parser\_json
- custom\_parser\_csv

并分别向两个Topic生产样例数据。

### 4. 创建表

Json



## 加载源表



获取集群信息

选择 Kafka Topic

样例数据

换一条

custom

▼ kafka-cluster-1

custom\_parser\_csv

custom\_parser\_json

```
1  {
2    "name": "Li",
3    "sex": "man",
4    "age": 24,
5    "addr": {
6      "country": "China",
7      "city": "Shanghai",
8      "region": "YangPu"
9    },
10   "works": [
11     "work_1",
12     "work_2",
13     "work_3"
14   ],
15   "create_time": "2022-11-01 08:00:00",
16   "update_time": "2022-11-20 12:00:00"
17 }
```

已选: custom\_parser\_json

解析器名称

org.apache.kylin.parser.json.JsonCustomParser

io.kyligence.kap.parser.TimedJsonStreamParser

org.apache.kylin.parser.csv.CsvCustomParser

org.apache.kylin.parser.json.JsonCustomParser



加载源表

×

\* 填写数据库和表名

SSB

CUSTOM\_JSON

关联 Hive 表

列信息

搜索列名

列名	列类型	样例值	注释
FIRST_WORK	varchar(256)	work_1	
CREATE_TIME	date	2022-11-01 08:00:00	
UPDATE_TIME	date	2022-11-20 12:00:00	
PROCESS_TIME	timestamp	2022-11-21 16:40:19	

上一步

加载

parserdemo

+

存储配额

服务状态

数据源表

+

搜索数据库名或表名

数据源: Kafka

SSB

CUSTOM\_JSON

SSB.CUSTOM\_JSON

最后更新时间: 2022-11-21 16:40:51 GMT+8

所有列

Kafka 源信息

ID	列	数据类型
1	NAME	varchar(256)
2	SEX	varchar(256)
3	AGE	integer
4	ADDR_COUNTRY	varchar(256)
5	ADDR_CITY	varchar(256)
6	ADDR_REGION	varchar(256)
7	FIRST_WORK	varchar(256)
8	CREATE_TIME	date
9	UPDATE_TIME	date
10	PROCESS_TIME	timestamp

共 10 条

10条/页

<

1

>

前往

1

页

CSV

## 加载源表



获取集群信息

选择 Kafka Topic

样例数据

换一条

cus

▼ kafka-cluster-1

custom\_parser\_csv

custom\_parser\_json

```
1 1|Li|"developer"|Table tennis|2022-11-01 08:00:00|2022-11-02 08:00:00
```

已选: custom\_parser\_csv

✖ 样例数据非 JSON 字符串, 请预先自定义解析器, 再在下方选择并解析。

解析器名称

org.apache.kylin.parser.csv.CsvCustomParser

io.kyligence.kap.parser.TimedJsonStreamParser

org.apache.kylin.parser.csv.CsvCustomParser

org.apache.kylin.parser.json.JsonCustomParser

页

加载源表



\* 填写数据库和表名

SSB

CUSTOM\_CSV

关联 Hive 表 ☐

列信息

搜索列名

列名	列类型	样例值	注释
SPORT	varchar(256)	developer	
CREATE_TIME	date	2022-11-01 08:00:00	
DELETE_TIME	date	2022-11-02 08:00:00	
PROCESS_TIME	timestamp	2022-11-21 16:44:31	

上一步

加载

parserdemo

数据源表

搜索数据库名或表名

数据源: Kafka

SSB

CUSTOM\_CSV

CUSTOM\_JSON










SSB.CUSTOM\_CSV

最后更新时间: 2022-11-21 16:45:19 GMT+8

所有列 Kafka 源信息

ID	列	数据类型
1	ID	integer
2	NAME	varchar(256)
3	JOB	varchar(256)
4	SPORT	varchar(256)
5	CREATE_TIME	date
6	DELETE_TIME	date
7	PROCESS_TIME	timestamp

共 7 条 10条/页 < 1 > 前往 1 页

- ✓  **custom-parser-demo** ~/IdeaProjects/ [redacted]/custom-parser-
- >  .idea
- ✓  lib
  - >  **kylin-streaming-sdk-5.0.0-SNAPSHOT.jar**
- >  src
- >  target
- </> dependency-reduced-pom.xml
-  pom.xml
- >  External Libraries
- >  Scratches and Consoles