
Kafka Connect Datagen Documentation

Release 1.0

Xu Shiyan

Apr 30, 2020

Contents

1 Quick Start	1
2 Table of Contents	3
2.1 Development	3
2.2 Configurations	4
3 Indices and tables	5

- Go to `example/quickstart/` and start all services

```
docker-compose up -d
```

- Run `docker-compose ps` to see all services' states

Name	Command	State	Ports
↪-----			
quickstart_broker_1	/etc/confluent/docker/run	Up	0.0.0.0:9092->
↪9092/tcp			
quickstart_connect_1	/etc/confluent/docker/run	Up	0.0.0.0:8083->
↪8083/tcp, 9092/tcp			
quickstart_kafka-connect-ui_1	/run.sh	Up	0.0.0.0:8001->
↪8000/tcp			
quickstart_kafka-rest-proxy_1	/etc/confluent/docker/run	Up	0.0.0.0:8082->
↪8082/tcp			
quickstart_kafka-topics-ui_1	/run.sh	Up	0.0.0.0:8000->
↪8000/tcp			
quickstart_zookeeper_1	/etc/confluent/docker/run	Up	0.0.0.0:2181->
↪2181/tcp, 2888/tcp, 3888/tcp			

Wait for Kafka Broker and Kafka Connect cluster to be fully started.

- Check <http://localhost:8000> to see the Broker UI
- Check <http://localhost:8001> to see the Connect UI

- Create data generation task

```
curl -X POST http://localhost:8083/connectors \
-H 'Content-Type:application/json' \
-H 'Accept:application/json' \
-d @connect.source.datagen.json | jq
```

- Based on the configurations, you should observe from Broker UI that

- messages are being published to topic `generated.events` at rate of 10 every 5 seconds
- every message is randomized over `status` and `direction` fields
- every message contains a timestamp field `event_ts`
- Go to Connect UI, select the “datagen” connector and click “PAUSE” or “DELETE”.

2.1 Development

2.1.1 Install

Build from source

- Import as Maven project
- Generate the jar file

```
mvn package
```

- Copy the jar file `target/kafka-connect-datagen-$version.jar` to a Kafka Connect worker's classpath

2.1.2 Docs

Update connector configs

Connector configurations are defined as `ConfigDef` objects. To convert the code-level definitions to documentation files, run

```
mvn clean compile test -Pgenerate-config-docs
```

Update user guide

We write user guide in `ReStructuredText` and use `Sphinx` to generate static HTML pages.

- Install `Sphinx` in a Python virtualenv
- Activate the virtualenv and run

```
sphinx-build -b html docs/ docs/_build
```

- Open `docs/_build/index.html` in browser to view the updated version.

2.2 Configurations

2.2.1 Performance Connector

topic.name Name of the Kafka topic to publish data to.

- Type: string
- Importance: high

poll.size Number of messages to be sent in one poll.

- Type: int
- Default: 1
- Importance: medium

poll.interval.ms Time interval (ms) between two polls.

- Type: int
- Default: 10000
- Importance: medium

message.template Message template to be used for each message.

- Type: string
- Importance: medium

random.fields List of fields to be randomized.

- Type: list
- Importance: medium

event.timestamp.field Name of the field to store event timestamp.

- Type: string
- Default: ts
- Importance: low

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`