# **WORKING WITH KAFKA**

v2.11 - 0.10.0.1

This section focuses on the core of the project. It connects two entities that have been tested independent from each other, i.e. Kafka and Ignite.

It lays the foundation for the streaming of data from Kafka to Ignite, followed by subsequent querying.

## **Getting Started**

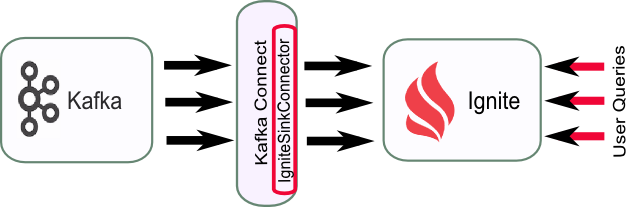
Apache Ignite Kafka Streamer module provides streaming from Kafka to Ignite cache.There are two ways this can be achieved-

* Importing Kafka Streamer module and instantiate KafkaStreamer for data streaming.
* Using Kafka Connect functionality.

Out of the two, the following manual explains the **Kafka Connect** method.

## **Kafka Connect (IgniteSinkConnector)**

From the Apache Ignite release 1.6, another way to integrate data processing was introduced. It is based on [Kafka Connect](http://kafka.apache.org/documentation.html#connect), a new feature introduced in Apache Kafka 0.9 that enables scalable and reliable streaming data between Apache Kafka and other data systems.  
Such integration enables continuous and safe streaming data from Kafka to Ignite for computing and transacting on large-scale data sets in memory.

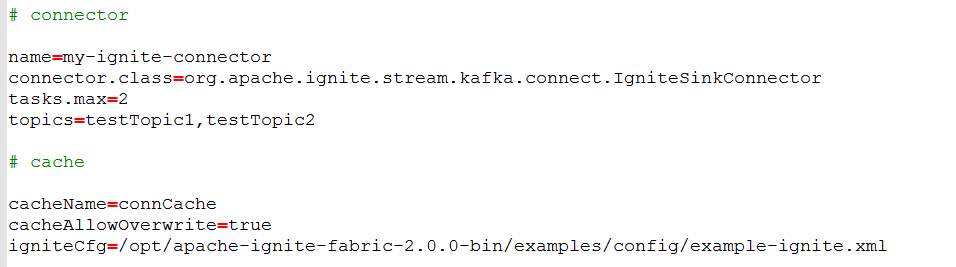


## **Configurating the Connector**

The module to implement Kafka Connect is present in **apache-ignite-fabric-2.0.0-bin/libs/optional/ignite-kafka**.

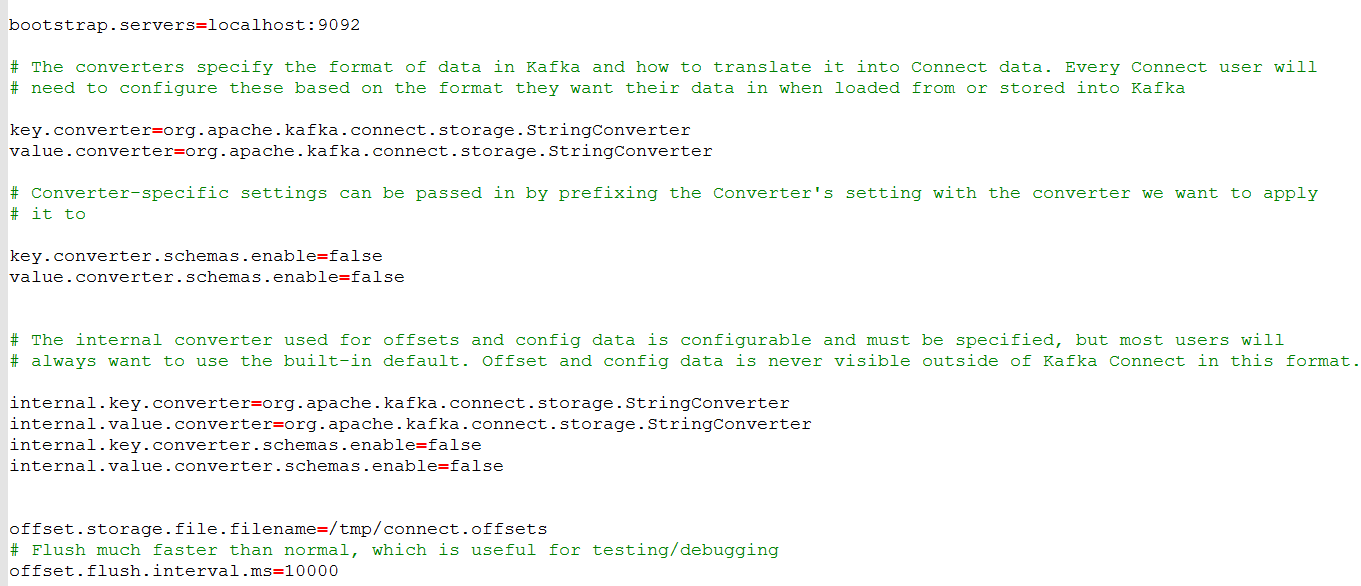
To configure the connector, we create two files, put them in a directory called myconfig in the Kafka Home directory. The contents of the files are as follows-

In **kafka\_2.11-0.10.0.1/myconfig/ignite-connector.properties**,



* name specifies the connector name.
* topics specifies the Kafka topics that the data is extracted from, to be transferred to Ignite.
* cacheName specifies the name of Ignite cache where data is pushed into.
* igniteCfg specifies the file that contains the cache configurations. The default example can be used here, or a custom one, depending on the requirements. It has to be taken care that the cache mentioned in cacheName is created (either in the igniteCfg file, or programmatically), or else the connect will fail.

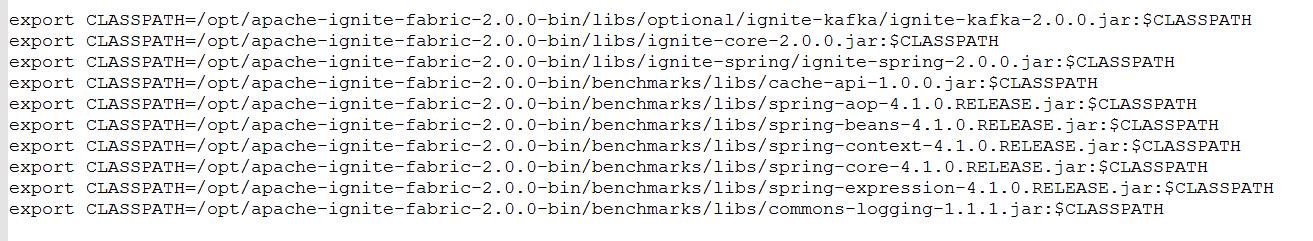
In **kafka\_2.11-0.10.0.1/myconfig/connect-standalone.properties**,



The properties specified in the above file are mostly defaults, except that it has been configured to handle String type data. The default is JSON. This configures the connector to run in standalone mode.

Once all of this has been setup, the next and most important step is to set the **classpath**. Without this, any of the connector and Ignite-related files will not be found, and numerous errors will be thrown.

The required JAR files to be included in the classpath are-



## **Running the Connector**

The first thing to do here is to make sure Apache ZooKeeper and Apache Kafka are running as processes. Once this is done, the final step is to run the connector, which can be achieved by simply running the following command from the Kafka Home directory-

**bin/connect-standalone.sh myconfig/connect-standalone.properties myconfig/ignite-connector.properties**