**Source code setup (Python)**

* **Introduction**
* **Environment setup** 
  + **beta,prod**
* **Repo setup & Version control**
  + **repo**
* **Code standards & Dependency** 
  + **Class reference diagram**
* **Class Level process flow from source code**
* **Environment config & Build process**
* **Database setup**
* **Common issues and solution**

# Python Kafka Client

Contents

[Kafka Consumer 1](#_Toc143037671)

[Introduction: 2](#_Toc143037672)

[Environment setup: 2](#_Toc143037673)

[Repo setup & Version control: 2](#_Toc143037674)

[Code standards & Dependency 2](#_Toc143037675)

[Class Level process flow from source code: 2](#_Toc143037676)

[Environment config & Build process: 3](#_Toc143037677)

[Docker File: 3](#_Toc143037678)

[Database setup 3](#_Toc143037679)

[Common issues and solution: 4](#_Toc143037680)

## Introduction:

**This project entails the consumption of live streaming data from IBM via Kafka and the subsequent storage of this data into a MySQL database.**

## Environment setup:

**To run this Python project, ensure the following components are in place:**

1. **Python Version 3.9**
2. **Preferred Python Editor (e.g., Visual Studio Code)**

## Repo setup & Version control:

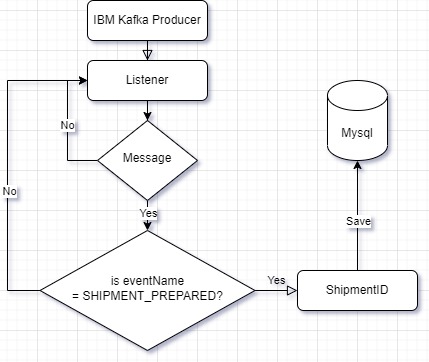
1. **Repo Manger: Github**
2. **Repo Name: Python\_kafka\_client**
3. **Repo Remote:** 
   * 1. **Fetch:** [**https://github.com/yeasinmahi/Python\_kafka\_client.git**](https://github.com/yeasinmahi/Python_kafka_client.git)
     2. **Push:** [**https://github.com/yeasinmahi/Python\_kafka\_client.git**](https://github.com/yeasinmahi/Python_kafka_client.git)
4. **Version Control: Git**

Code standards & Dependency**:**

**This project employs a straightforward Python console application as a consumer without the use of any frameworks. The following package dependencies are utilized:**

1. confluent\_kafka==2.2.0
2. mysql-connector-python==8.1.0
3. python-dotenv==1.0.0
4. cryptography==41.0.3

## Class Level process flow from source code:



Environment config & Build process:

**To deploy the project, create a Docker image containing the required dependencies and deploy it to your server. Ensure you have a MySQL server in place for data storage.**

### Docker File:

**Create a Dockerfile with the necessary package dependencies and deploy it to your server.**

Database setup**:**

**Below is the MySQL schema designed for storing the data:**

CREATE TABLE dt\_ibm\_shipment (

Id INT AUTO\_INCREMENT PRIMARY KEY,

shipment\_id VARCHAR(50),

createddate DATETIME DEFAULT CURRENT\_TIMESTAMP,

COMINV\_status BIT DEFAULT 0,

COMINV\_proces\_date DATETIME,

COMINV\_retry\_count INT DEFAULT 0,

invoice\_number VARCHAR(255),

file\_names VARCHAR(255),

COMINV\_DOC\_status BIT DEFAULT 0,

COMINV\_DOC\_proces\_date DATETIME,

COMINV\_DOC\_retry\_count INT DEFAULT 0

);

ALTER TABLE dt\_ibm\_shipment

ADD CONSTRAINT shipment\_id\_unique UNIQUE (shipment\_id);

Common issues and solution:

**Currently, the project is still in the UAT phase, and specific common issues and their corresponding solutions are yet to be documented.**