|  |  |
| --- | --- |
| **Topic** | Apache Kafka |
| **Document Name** | KAFKA-EX-03 |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Difficulty Level** | | | |
| **Beginner** | **Junior** | **Senior** | **Expert** |
| □ | ■ | □ | □ |

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Ver | Comments |
| 05.03.2025 | Mennan Tekbir | 1.0 | Initial Draft |
| 05.03.2025 | Mehmet Erdem Önal | 1.1 | Revisions |

# Apache Kafka

## Exercise KAFKA-EX-03:

**Definiton:** Create 2 sample java programs; one for Producer one for Consumer. Publish java objects ( you can use following class ) from Producer app. Read those objects from Consumer app and print them to the console.

**Java Classes & Objects:** <https://www.geeksforgeeks.org/classes-objects-java>

**Java Program For Class Example:**

class Student {

  // data member (also instance variable)

  int id;

  // data member (also instance variable)

  String name;

  public static void main(String args[])

  {

    // creating an object of

    // Student

    Student s1 = new Student();

    System.out.println(s1.id);

    System.out.println(s1.name);

 }

}

**Sample Kafka Producer code:** <https://github.com/chrissearle/kafka-java-to-scala/blob/master/java-starter/producer/src/main/java/net/chrissearle/kafka/BasicProducer.java>

**Sample Kafka Consumer code:** <https://github.com/chrissearle/kafka-java-to-scala/blob/master/java-starter/consumer/src/main/java/net/chrissearle/kafka/BasicConsumer.java>

Please provide screenshots to show your work.

## KAFKA-EX-03 Solution:

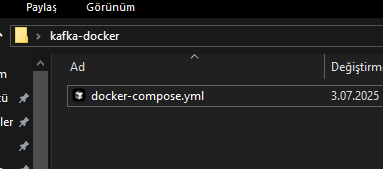
**Your Answer:**

**EGE DiREK**

**Steps**

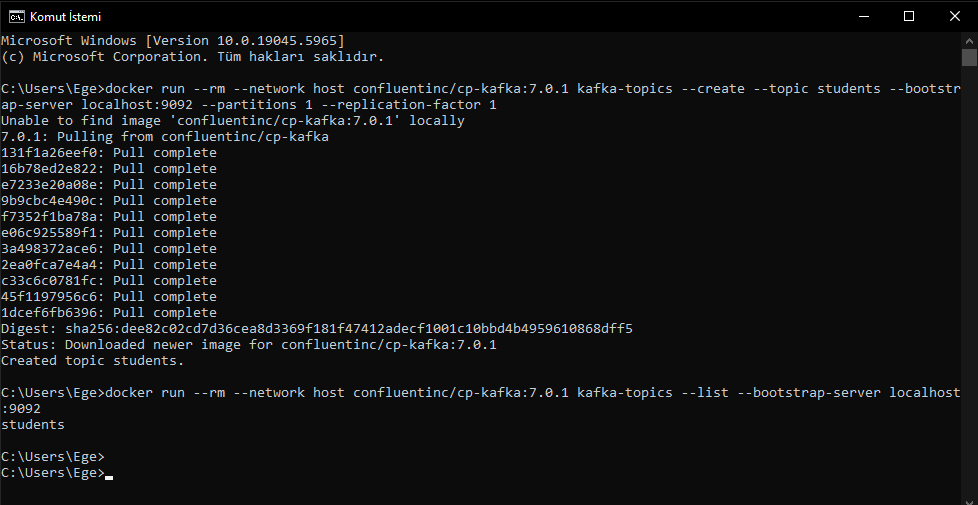
**1-First create a project folder in your IDE and use Maven for Kafka.**

**2-Create a folder include compose.yml file for attach our code to Kafka and Kafka to Docker Dektop with ZooKeeper.**

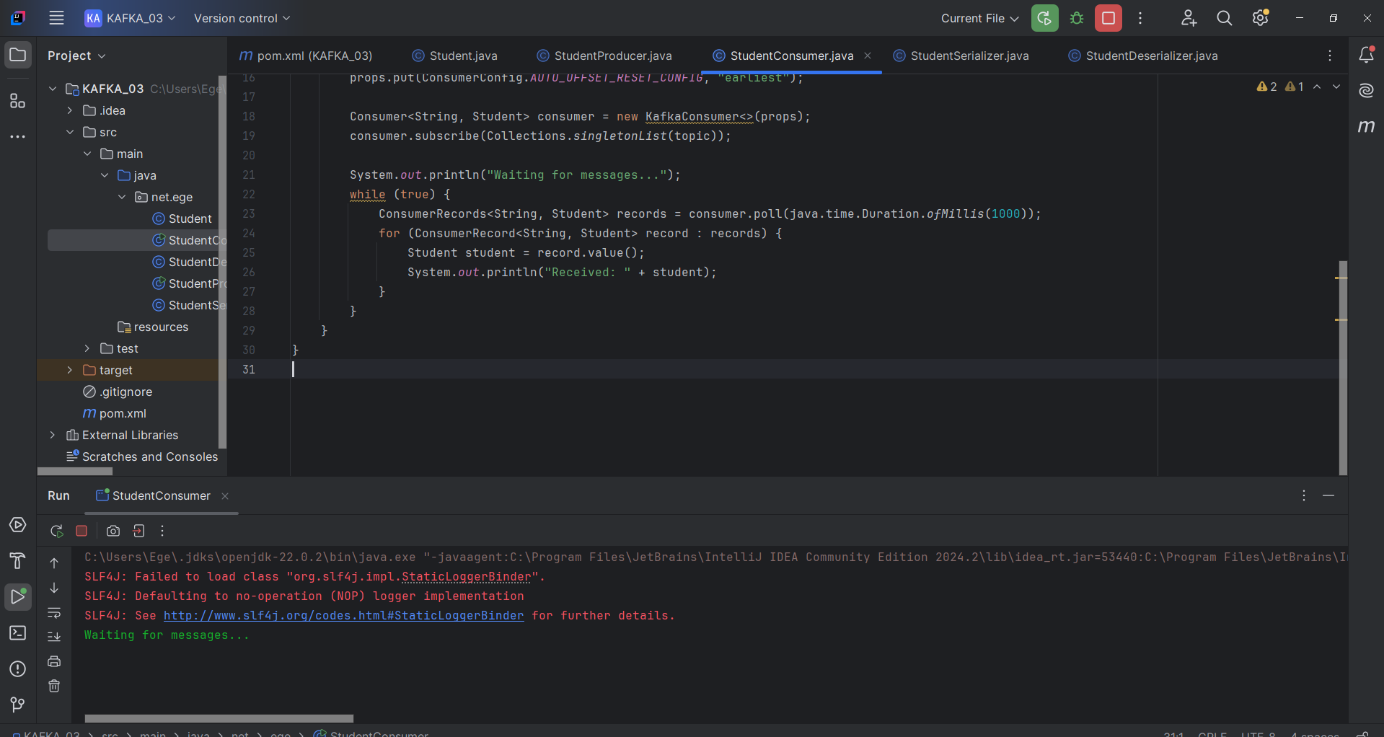
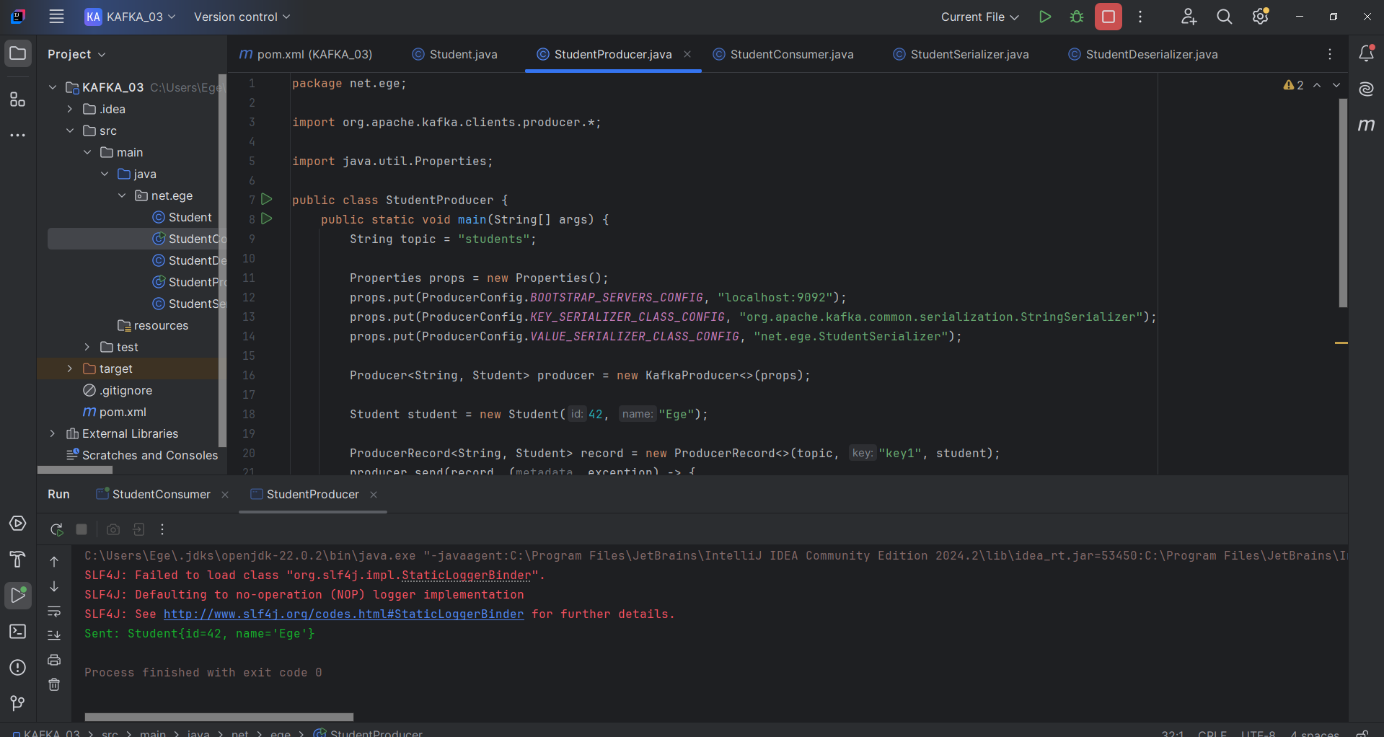
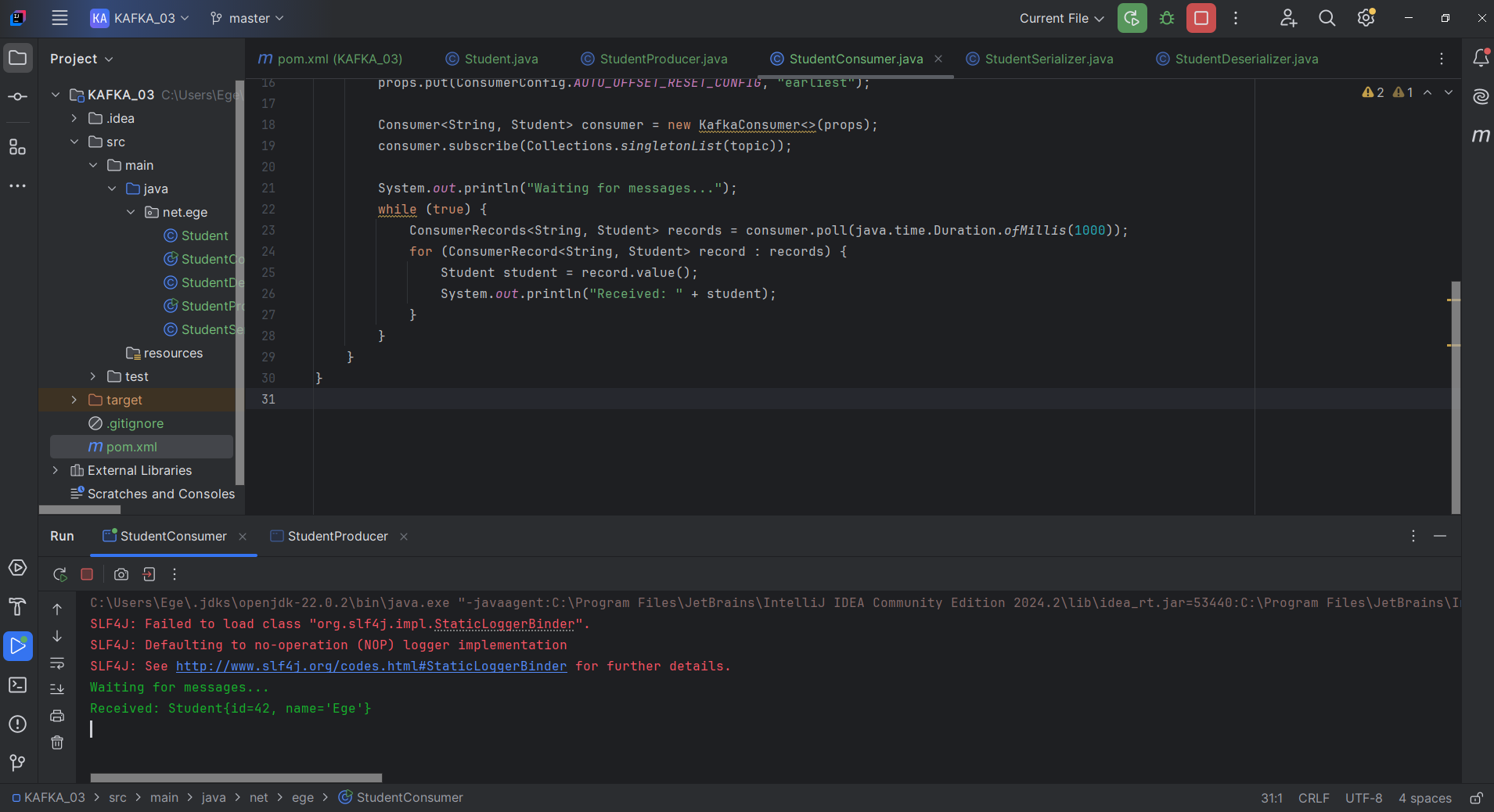
**[](https://private-user-images.githubusercontent.com/209494583/461984473-274aa744-84ad-4507-bbd9-4f3d0dcd1ade.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3NTE1NDE2MjgsIm5iZiI6MTc1MTU0MTMyOCwicGF0aCI6Ii8yMDk0OTQ1ODMvNDYxOTg0NDczLTI3NGFhNzQ0LTg0YWQtNDUwNy1iYmQ5LTRmM2QwZGNkMWFkZS5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBVkNPRFlMU0E1M1BRSzRaQSUyRjIwMjUwNzAzJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDI1MDcwM1QxMTE1MjhaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT1iMDEyOTgyZThiNTQ0MDdlZjdiY2ZlNmYyZmMxYWE0NGJhMTg5OTBlMjY1NjAzODZjOWQ0ODQ2NTU3OTNhODIzJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCJ9.Zy4VjWHbxdRfftrKNWAVZ9LbFN7n2djcq_4CWW_55kQ)**

**3-Create classes for this project. And config pom.xml (Maven) file for Kafka install and run.**

**4- Create Kafka topic on your terminal and list for check.**

**[](https://private-user-images.githubusercontent.com/209494583/461988237-ed36953e-e8e4-49c4-83af-b82a5720f9e8.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3NTE1NDE2MjgsIm5iZiI6MTc1MTU0MTMyOCwicGF0aCI6Ii8yMDk0OTQ1ODMvNDYxOTg4MjM3LWVkMzY5NTNlLWU4ZTQtNDljNC04M2FmLWI4MmE1NzIwZjllOC5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBVkNPRFlMU0E1M1BRSzRaQSUyRjIwMjUwNzAzJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDI1MDcwM1QxMTE1MjhaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT02YzE3ZjFhMjkxOTMyNjdmODFmYWYzNTA1MTRmNDFlYWYxYzQ0YWYxMDRjYzhhMmE0ZWZlNDAwZWQyMGNiZmY5JlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCJ9.i3xEiEPQnsMi69-0O-Qa0kH-TGU86OOBf7J5zLqlb7c)**

**5- Lastly, for execute our project first run StudentConsumer class for connection to Kafka and listen new incoming messages; then run StudentProducer class for sent 'Student' object to Kafka.**

**[](https://private-user-images.githubusercontent.com/209494583/461993645-2d0f1e6b-4336-4401-a5ab-42af5c1b04e5.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3NTE1NDE2MjgsIm5iZiI6MTc1MTU0MTMyOCwicGF0aCI6Ii8yMDk0OTQ1ODMvNDYxOTkzNjQ1LTJkMGYxZTZiLTQzMzYtNDQwMS1hNWFiLTQyYWY1YzFiMDRlNS5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBVkNPRFlMU0E1M1BRSzRaQSUyRjIwMjUwNzAzJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDI1MDcwM1QxMTE1MjhaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT1iNTk2NTkwYTc1YWMxYTlmYTQzMDEzZWQ4ZTVhMGJlMTIxYmRiZTkwODdmNWNjYjhmMGM2YTliZWIxMjhmODZjJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCJ9.5lRsc2-NFTq27yBHG-uKzoAEqQ5ZzithlT8CBuddKpk) [](https://private-user-images.githubusercontent.com/209494583/461993658-29bca6aa-679a-4f7c-8e5e-c75de5a09a1a.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3NTE1NDE2MjgsIm5iZiI6MTc1MTU0MTMyOCwicGF0aCI6Ii8yMDk0OTQ1ODMvNDYxOTkzNjU4LTI5YmNhNmFhLTY3OWEtNGY3Yy04ZTVlLWM3NWRlNWEwOWExYS5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBVkNPRFlMU0E1M1BRSzRaQSUyRjIwMjUwNzAzJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDI1MDcwM1QxMTE1MjhaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT01MDMwOGRlMDU0NWQxYjUxMGVkODg3ZjEwYjg1NmZkMjk2OWI2YmYzNDdiODc4YjRhMWIyZThjMDgyNGI0N2FiJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCJ9.R79chDhEOT_12-C25V1DFmfJ868RQ-L0568L4cdvZq0) [](https://private-user-images.githubusercontent.com/209494583/461993666-344112ec-6fa6-4a79-9552-95d687ebf02d.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3NTE1NDE2MjgsIm5iZiI6MTc1MTU0MTMyOCwicGF0aCI6Ii8yMDk0OTQ1ODMvNDYxOTkzNjY2LTM0NDExMmVjLTZmYTYtNGE3OS05NTUyLTk1ZDY4N2ViZjAyZC5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBVkNPRFlMU0E1M1BRSzRaQSUyRjIwMjUwNzAzJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDI1MDcwM1QxMTE1MjhaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT0xMTFmOWVmN2E4ODJiY2ZmNjcwMzdjM2I4M2M0NjA3Nzc1MDk1ZjNiYTE3MjQ1ZjYyODI1ZGJmODBiNzFjYmFjJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCJ9.kN8tCGuCSPFzYSTNBdEpbeiIu8CYYpZwsvhHDTo5iiA)**

**6- Conclusion: You should see these messages on order in your terminals. Our producer app successfully read and print our object.🎉🎉**

* **Waiting for messages.. // message in your StudentConsumer terminal**
* **Sent: Student{id=42, name='Ege'} // message in your StudentProducer terminal**
* **Received: Student{id=42, name='Ege'} // message in your StudentConsumer terminal**