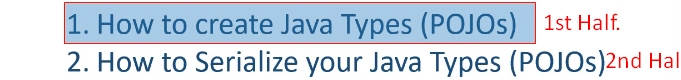
1. Text

   Description automatically generated
2. 
3. In the earlier lectures, we learnt how to create Kafka Producer and send the data to the Kafka Cluster.
4. However, we were sending simple text msgs (String).
5. But the real-life records are not that plain text or simple msgs like String but they are complex java object.
6. 
7. Kafka Programming is mainly dealing with data records in a variety of formats and that brings two critical questions.
   1. How to create Java Types (POJOs):
   2. How to serialize your Java Types (POJOs):  
      To send msgs to Kafka Cluster.
8. A simple example can manage with 4-5 types.  
   However, in real-life scenario, a complex Data Processing requirement can quickly scale up to hundreds of unique record formats.  
   Creating POJO for the message type is a tedious mechanical activity.  
   **Jatin**: Like in our app API Banking, we send msgs to downstream and now downstream needs to deserialize the msgs into appropriate Java Object. So downstream needs to create POJO class for each msg type which is a tedious job.
9. Can we automate that?
   1. Yes we can.
10. Basically, what we want to achieve is straightforward.
    1. **Use A Schema Definition Language**: We want to be able to define a msg schema using some simple schema Definition Language(Even this can be automated as we did in Banking app).
    2. **Auto-Generate Java Class Definition**: Then we want our IDE or some java build tool to generate Java Class definition from the Schema definition automatically.
11. There are many ways and several tools to achieve this.
12. However, there are two altivenatives those can be suggested for your Kafka App.
13. 
14. We have enough open source support for generating POJO for both options above.  
    And I will cover each of these formats.
15. Generating POJO is the 1st half of the problem. 2nd half is to serialize and deserialize them.  
    
16. Every Java app will use a bunch of Java types as msg type and you must provide a serializer and a deserializer for all of them.  
    And creating a serializer of each type is a big headache.
17. Can we develop **reusable serializer and deserializer** and can apply that to all the Java Types.  
    Yes, we can.
18. There are many object serialization formats but for Kafka App, I will suggest you two alternatives which are most commonly used formats.
    1. JSON Serialization.
    2. Avro Serialization.
19. In this section we will learn the following things:
    1. We will learn to define the schema of your events using JSON and
    2. How you can auto-generate serializable POJO definition from the schema definition.
    3. Then we will learn to define the schema definition of your events (msgs) using AVRO Schema Definition Language and how you can auto-generate a serializable POJO definition from the AVRO Schema Definition.
    4. Finally, in the next section, I’ll help you to serialize these objects using JSON as well as AVRO serialization.