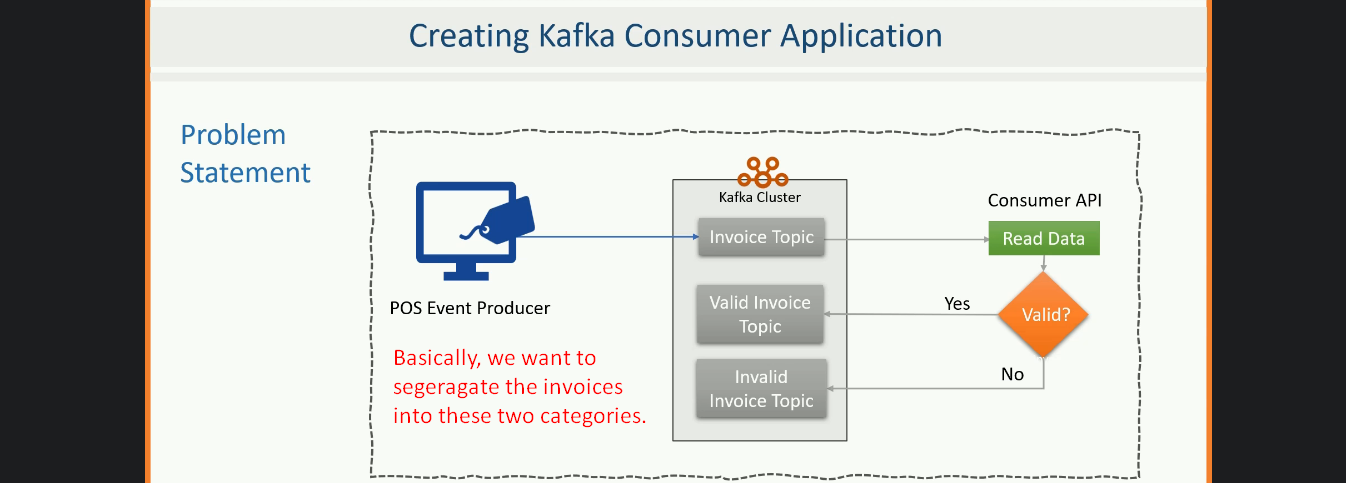
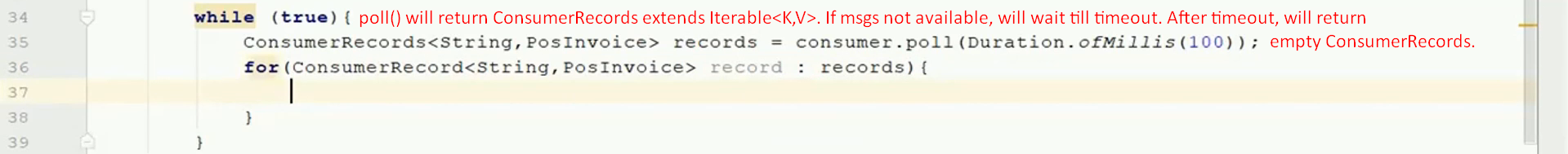
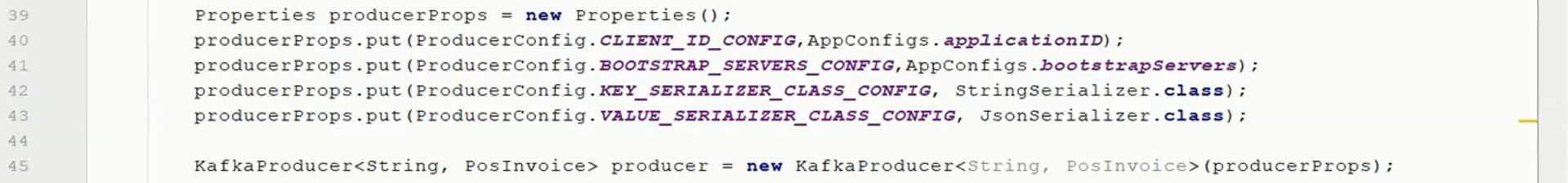
1. Text

   Description automatically generated
2. 
3. We already learnt about Kafka Producer API.
4. **Agenda**:
   1. In this lecture, I will give you a practical introduction about Kafka Consumer API.
5. 
6. Same as earlier. Let’s take a problem solution approach to learn kafka Consumer.
7. Here is my problem statement.
   1. In the earlier lecture, we created POS (Points-Of-Sale) Simulator that generates a series of invoices and sends them to a Kafka Topic.
   2. In this example, we want to implement miniature form of realtime data validation service for invoices.
   3. So, we want to read all the invoices in real-time and want to apply business rules to validate the invoices.  
      If valid, then send the invoices to topic 🡺 “**Valid Invoice**” otherwise to “**Invalid Invoice**”.  
      As a result, we will be able to segregate valid and invalid invoices into two different topics.
   4.   
      We will end the example there.   
        
      However, the overall system might look something like this (below)  
      In the following snapshot, **res = responsible**  
       Chart, diagram

      Description automatically generated  
      Let’s build real-time **Data Validation Service**.
8. The first thing we need is business rules to define a valid invoice.
9. Diagram

   Description automatically generated with medium confidence
10. Same as earlier, we have a starter project attached with lecture and note.
11. Text

    Description automatically generated with medium confidence
12. Let’s create a main() method having code for Kafka Consumer Creation.  
    kafka Consumer creation is 4-step Process.  
    1. **Step 01**: **Setting Config Parameters**:  
       Graphical user interface, text, application

       Description automatically generated
    2. **Step 02: Creating Kafka Consumer**.  
       
    3. **Step 03**: **Subscribing to the Topic:**
    4. **Step 04:** **Read the msgs in a loop:**Basically, this loop run infinitely as we’re expected to keep reading the msgs from the topic and process them in realtime for the life of the app.  
       This thing should never stop.  
         
       We completed our 4 steps to consume records from a topic.  
       The rest of the things are related to business logic.
13. Let’s study the further part of our problem.  
    We want to send the valid and invalid invoices to their respective topics.  
    The complete code is written in main() -> reading invoices, segregating them and sending them to valid and invalid topic.  
    So, let’s identify them and create a producer.  
    **Creating Producer**:  
      
      
    **Checking valid and invalid records and sending to corresponding topic.**Text

    Description automatically generated with low confidence
14. The app implements consume-transform-produce pipeline.
15. How to run the complete app to check, please see the last few minutes of the lecture.
16. There are a lot of things that can go wrong with consumer app along with scalability & fault-tolerance.   
    Let’s discuss about the scalability & fault-tolerance in the next lecture.
17. The complete code you can see in the attached downloadable.