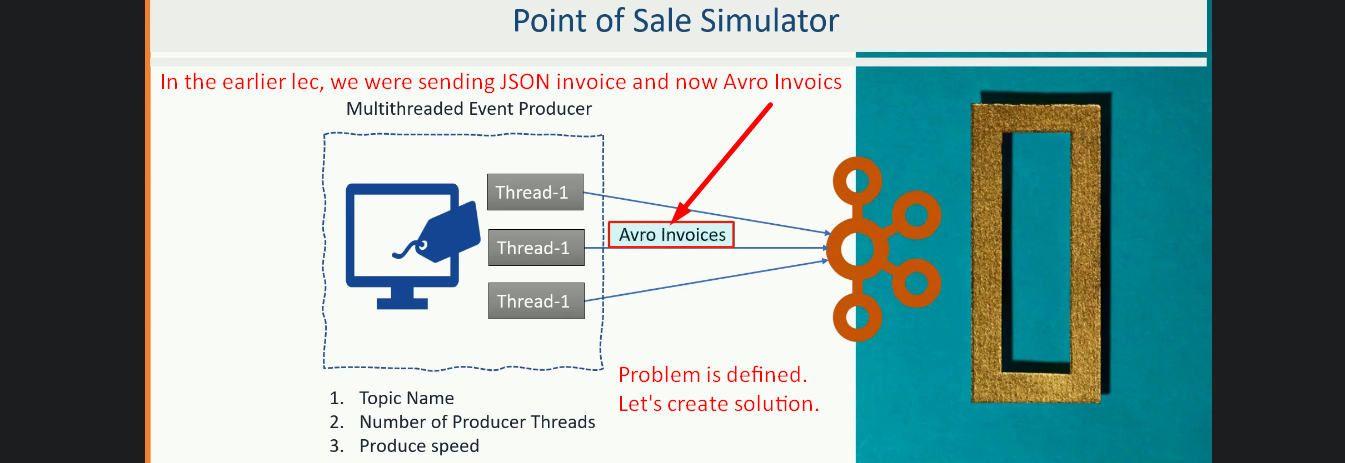
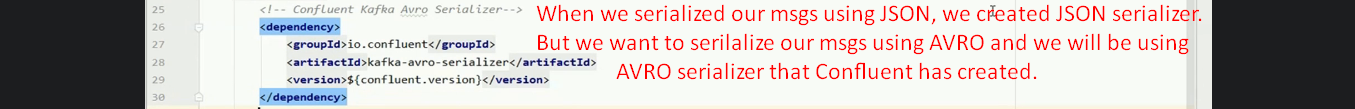
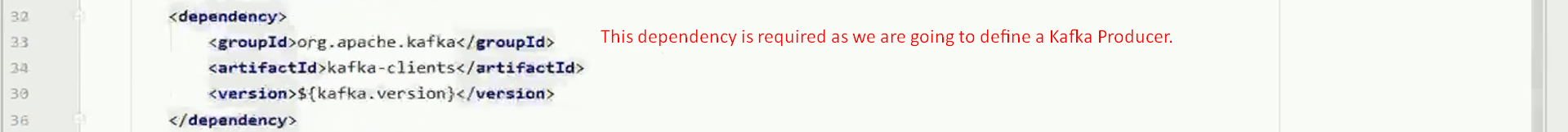
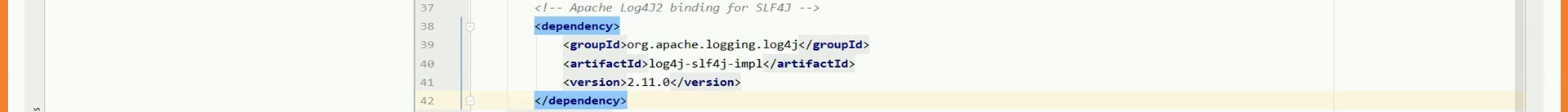
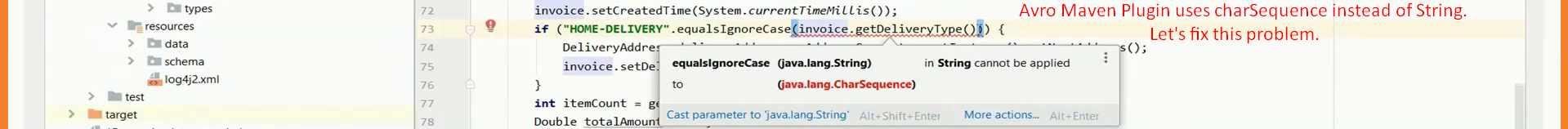
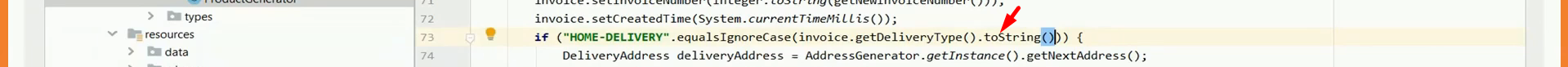
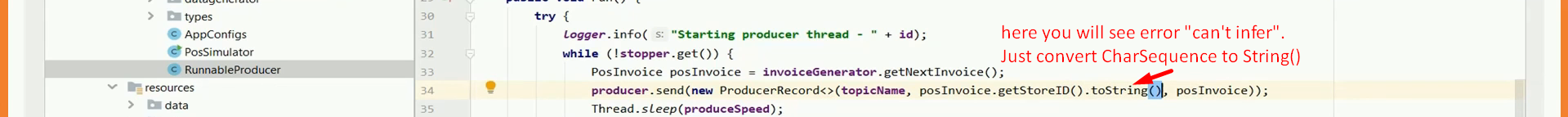
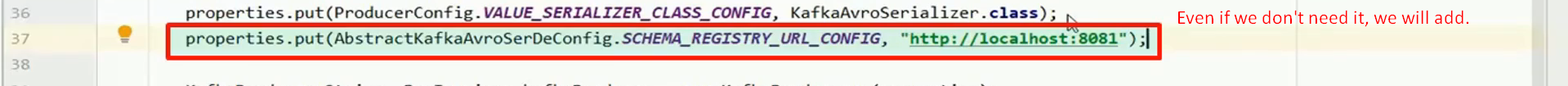
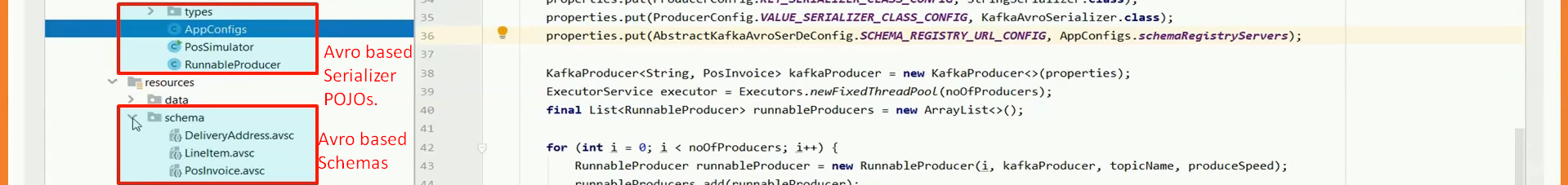
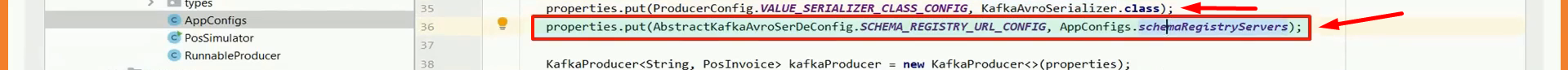
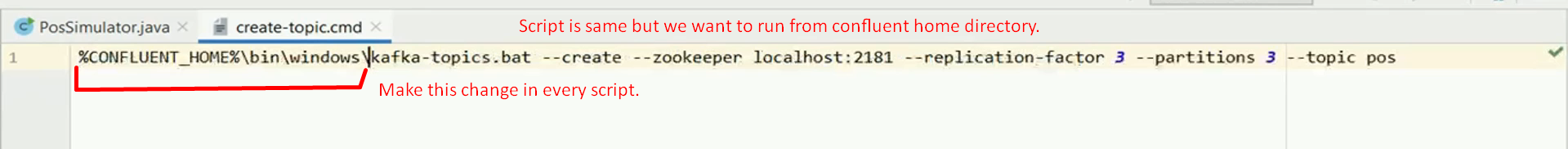
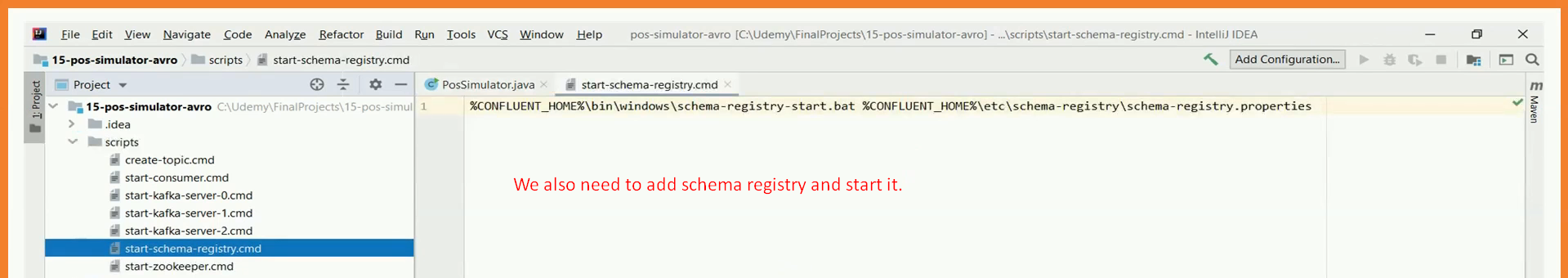
1.   
   
2. In the previous section, we learnt to generate serializable POJO classes from Avro Schema definition.
3. In this lecture, we’re going to use them in Kafka Producer Program.
4. 
5. 
6. Avro serializer was offered by Confluent Platform.  
   So we’re going to use Confluent Community Edition.
7. Let’s create a brand new project.
8. Versions:
   1. confluent.version: 5.3.0 🡺 We will be using this confluent version.
   2. kafka.version: 5.3.0 🡺 Instead of using Apache Kafka, we will be using Kafka version that comes with confluent 5.3.0.
      1. Using Confluent Kafka version is necessary to avoid version conflicts because we’re going to execute our program on Confluent Platform and hence we need the same version and this how confluent defines Kafka Version in their platform.
9. Confluent artifacts are not hosted on standard maven repository.  
   Hence we need to define a custom repository here.  
   
10. Let’s define the dependencies now:
    1. We need Kafka Avro Serializer offered by Confluent.  
       
    2. 
    3. 
    4. Timeline

       Description automatically generated
    5. Graphical user interface, text, application

       Description automatically generated
11. 
12. Graphical user interface, text, application

    Description automatically generated
13.   
    Solution:  
    
14. 
15. 
16. Let’s come to the POS Simulator.
17. So far, we fixed very simple errors like instead of String, we’re getting CharSequence and we converted it to String. So simple errors.   
    At a high level, till now everything is same as we had in JSON based Producer.  
    Now, we’re going to make changes that are fundamental changes for making it work with the Avro.
18.   
    **Solution**:  
    
19. Confluent Avro Serializer doesn’t work without Confluent Schema Registry.  
      
    We will put this constant into the constant class.  
    Graphical user interface, text, application, email

    Description automatically generated
20. So, how is the JSON Based Producer (discussed in the previous lecture) is different from Avro Based Producer (in this lecture).  
    **Summary**.
    1. **Change 01**:  
       
    2. **Change 02 & 03**:  
       
    3. **Change 04**:  
       Text, application

       Description automatically generated
21. To test it, we need Kafka Scripts. Let’s copy and paste from previous lecture.  
    After copying and pasting, change the directory to confluent directory in all the script.  
      
    NOTE: Also change the configuration file location in each script means where we’re passing properties file.
22. 
23. Run all the scripts.
24. Run the main()  
    Graphical user interface, text, application

    Description automatically generated
25. Graphical user interface, text

    Description automatically generated
26. Project file is attached to the lec and note itself. Look for “**08-pos-simulator-avro.zip**”