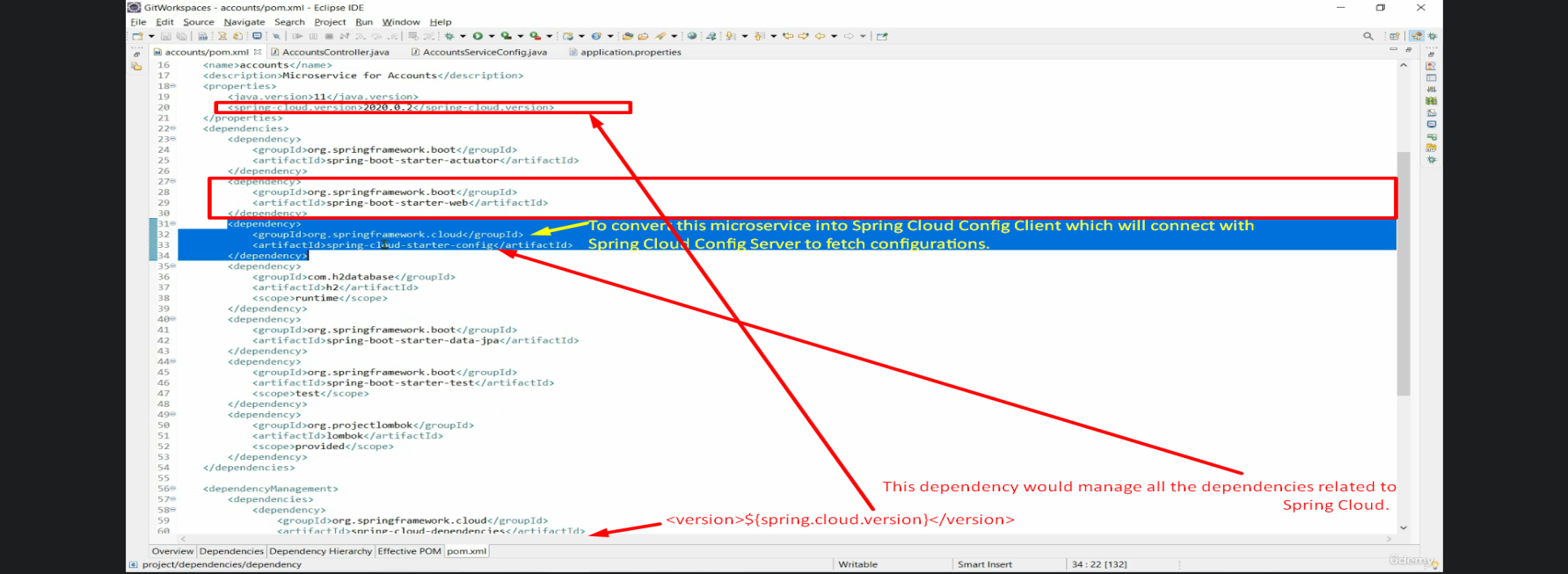
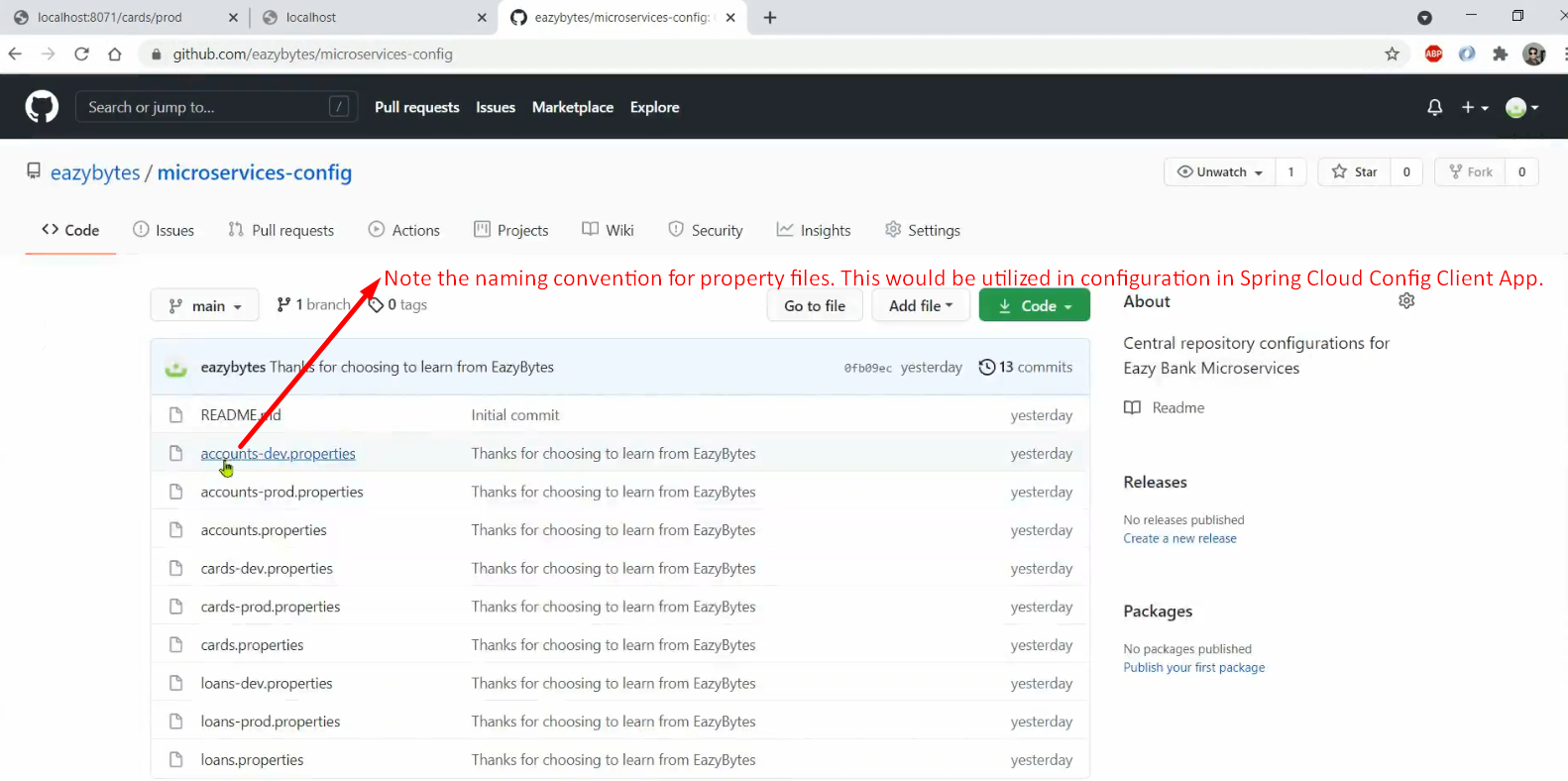
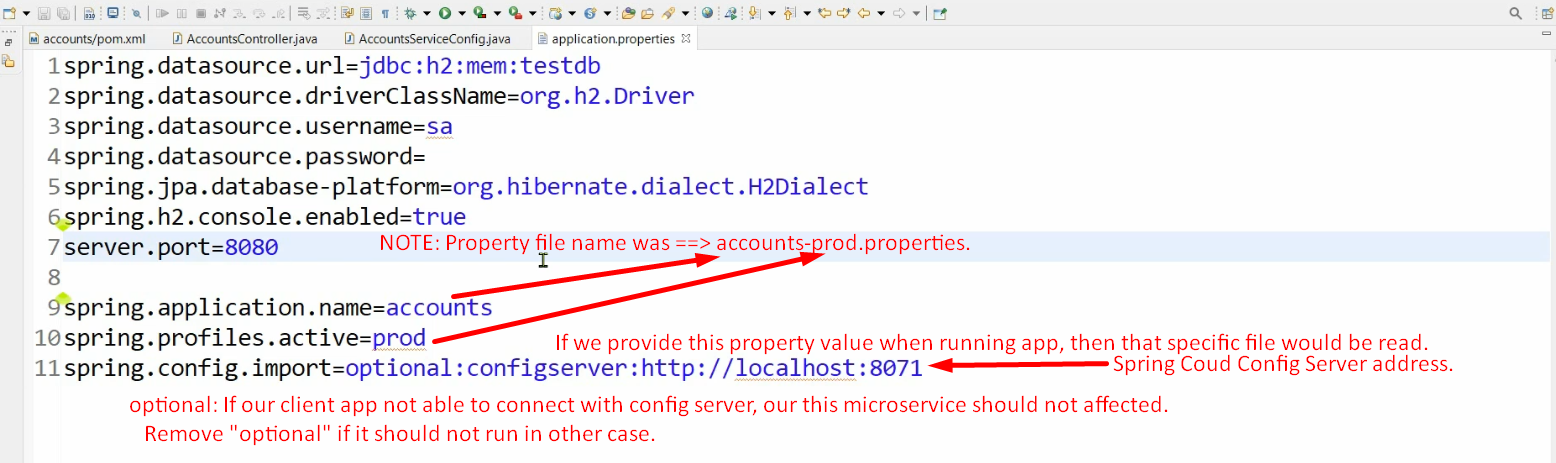
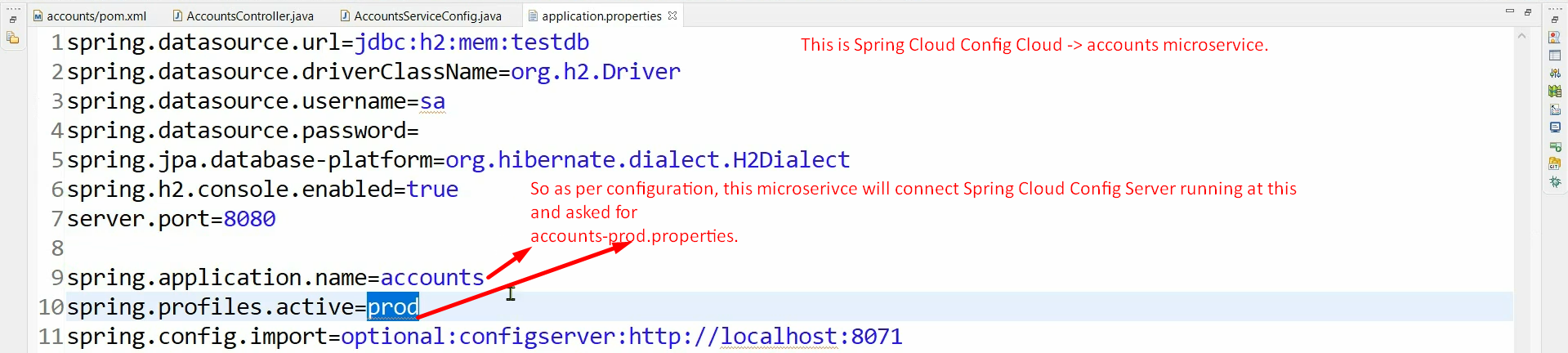
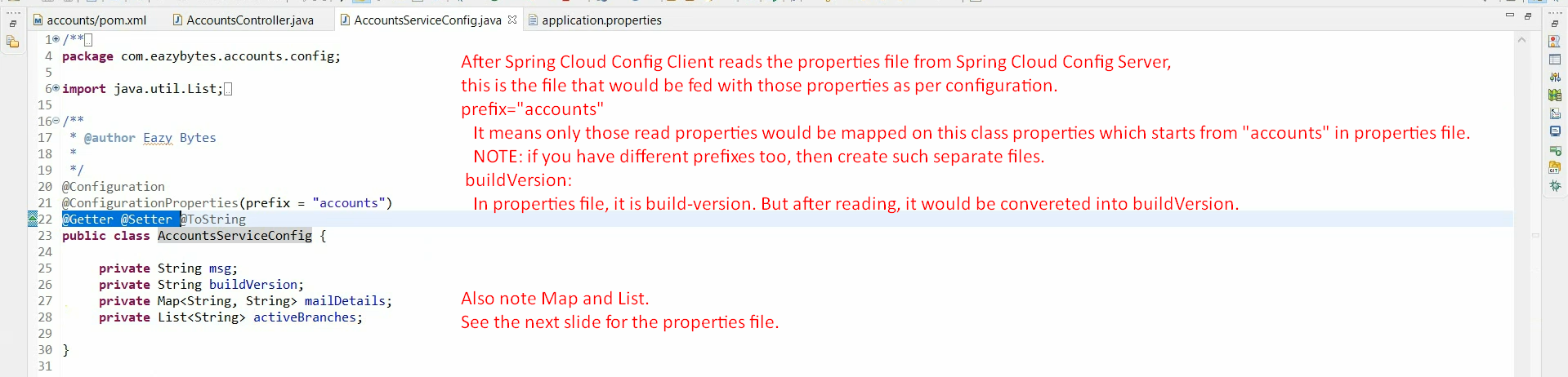
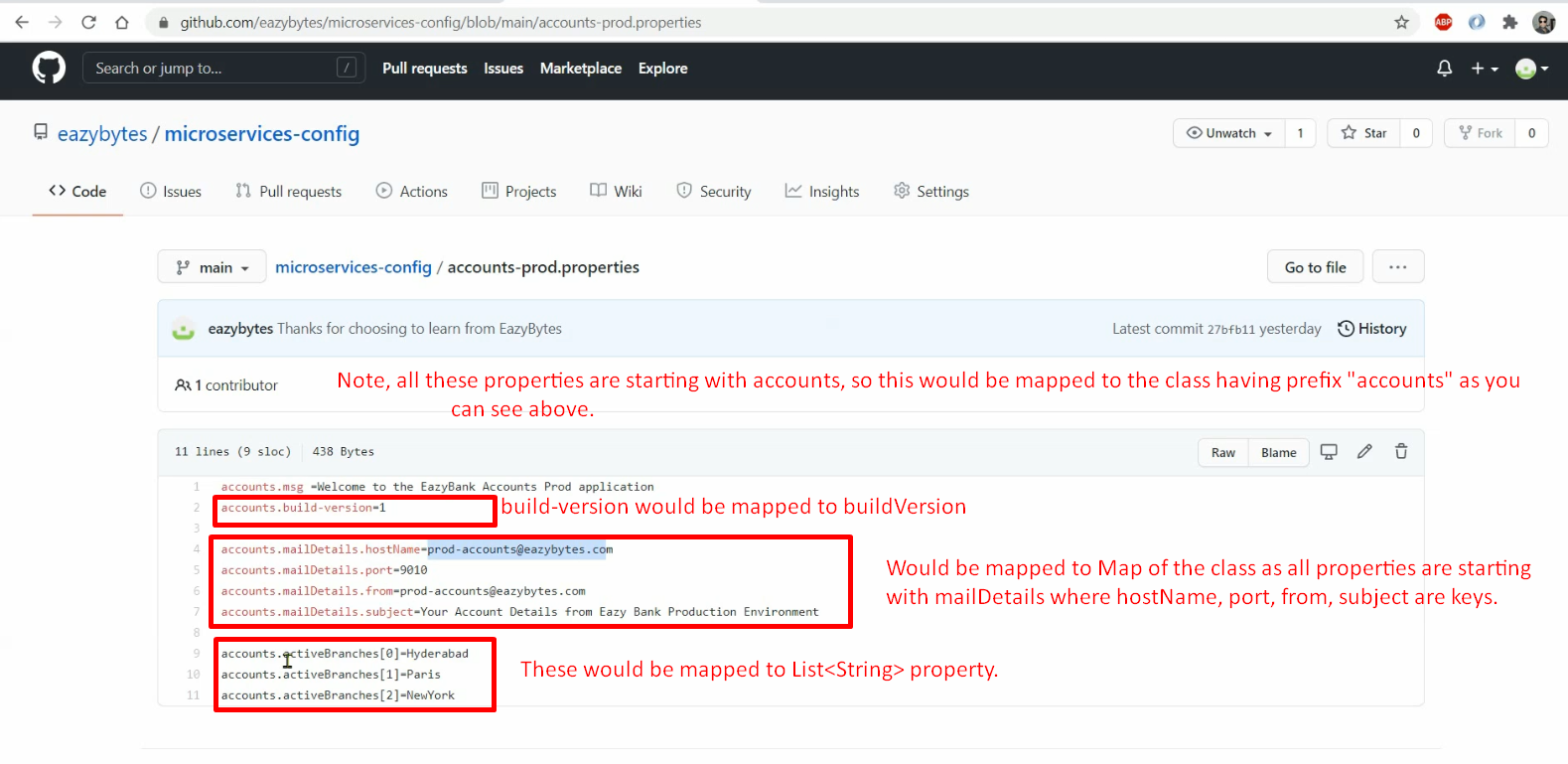
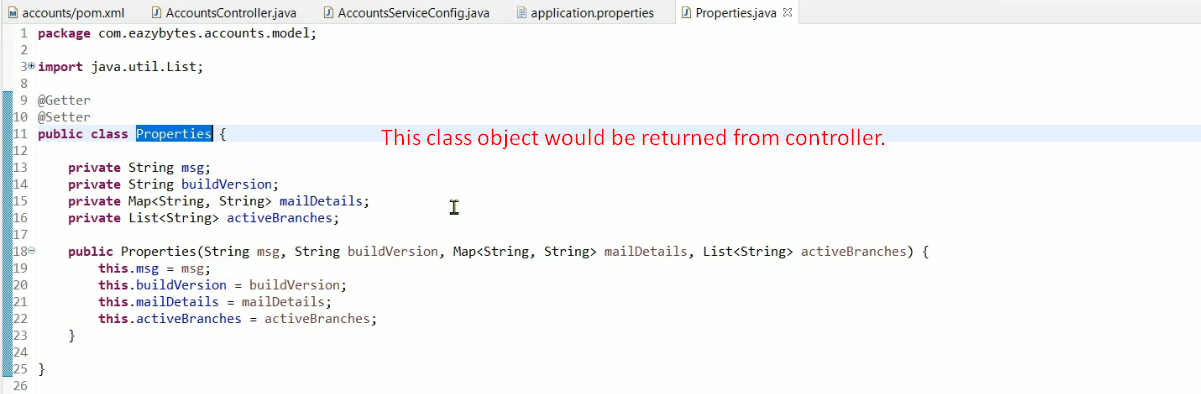
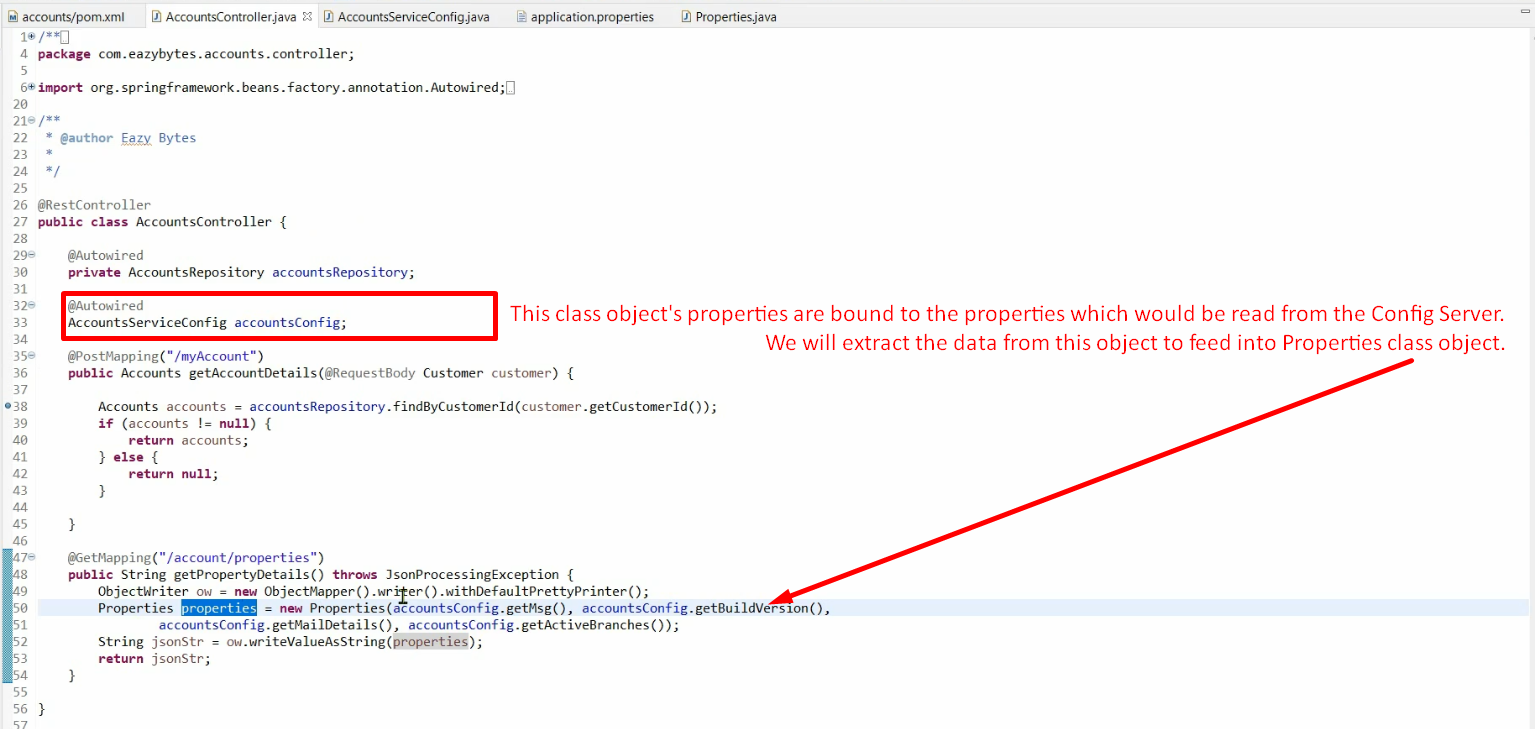
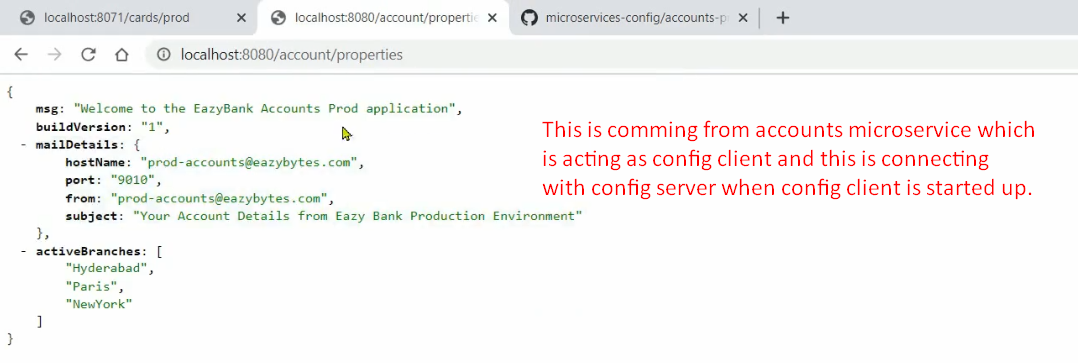
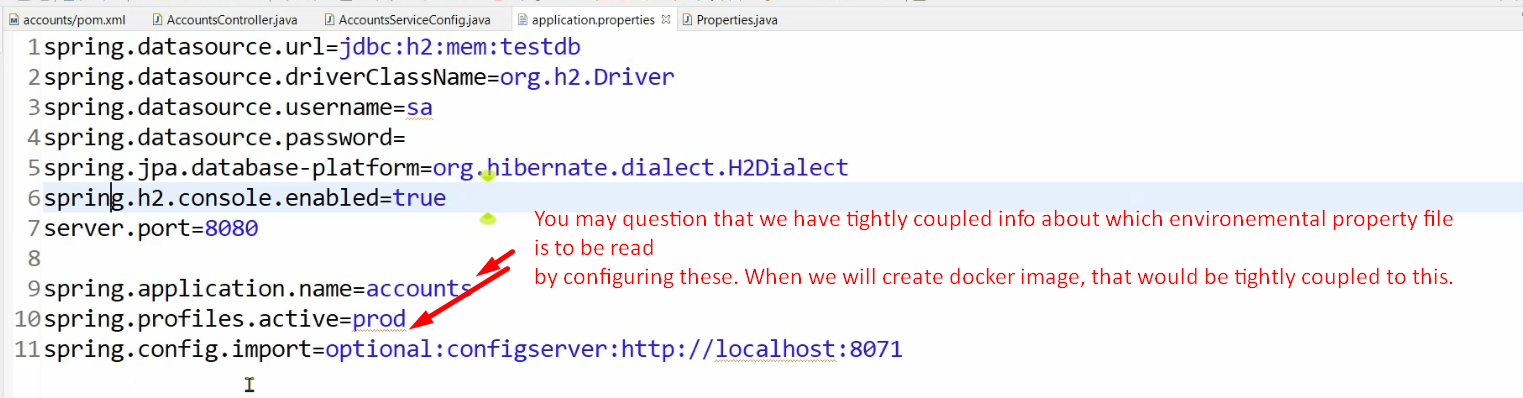
1. So far we studied
   1. How to create a Config server.
   2. how to read property files from
      1. File System.
      2. Classpath.
      3. Github (Central Repo)
2. **Agenda**:
   1. What changes are to be made in microservice to read properties from Spring Cloud Config Server?
3. 
4. In above snapshot, add the above dependency at line# 32 when you want your microservice to act as a Spring Cloud Config Client
5. 
6. 
7. Next, we should write code to read those properties into a class object so that we can use them in our business logic.
8. First properties file from the Config Server would be fetched then its properties would be added to spring context.   
   Then a bean of the AccountSerivceConfig would be created and all the properties staring with accounts would be mapped to properties.  
     
   
9. 
10. Let’s now run the microservice and return those read configurations from Spring Cloud Config Client by exposing some end-point.
11. Before that let’s add a model class whose object would be returned from the controller.
12. 
13. 
14. Let’s run first the Config server and then “Account Microservice”.
15. 
16. 
17. 
18. Don’t worry. We will see how to decouple this environmental configuration from the microservice.