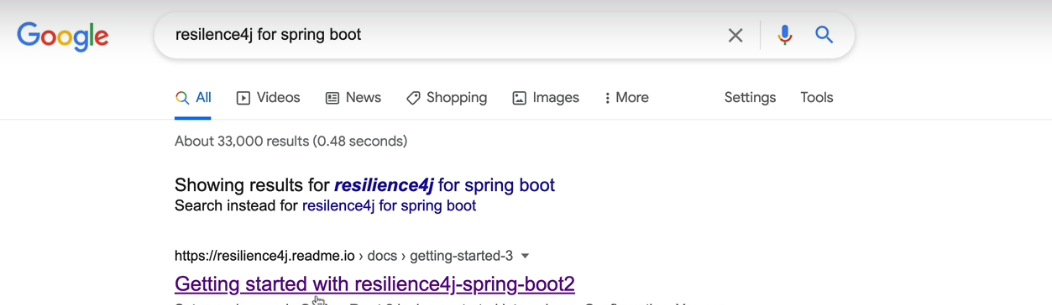
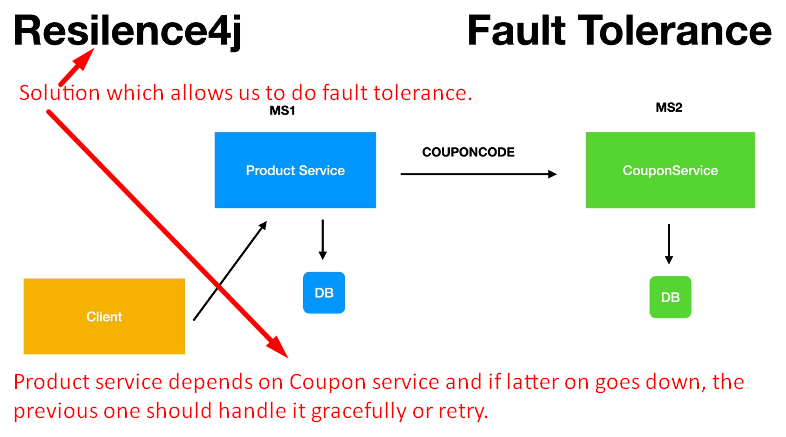
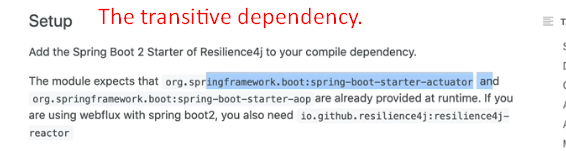
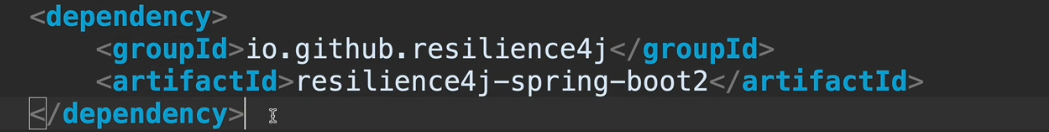
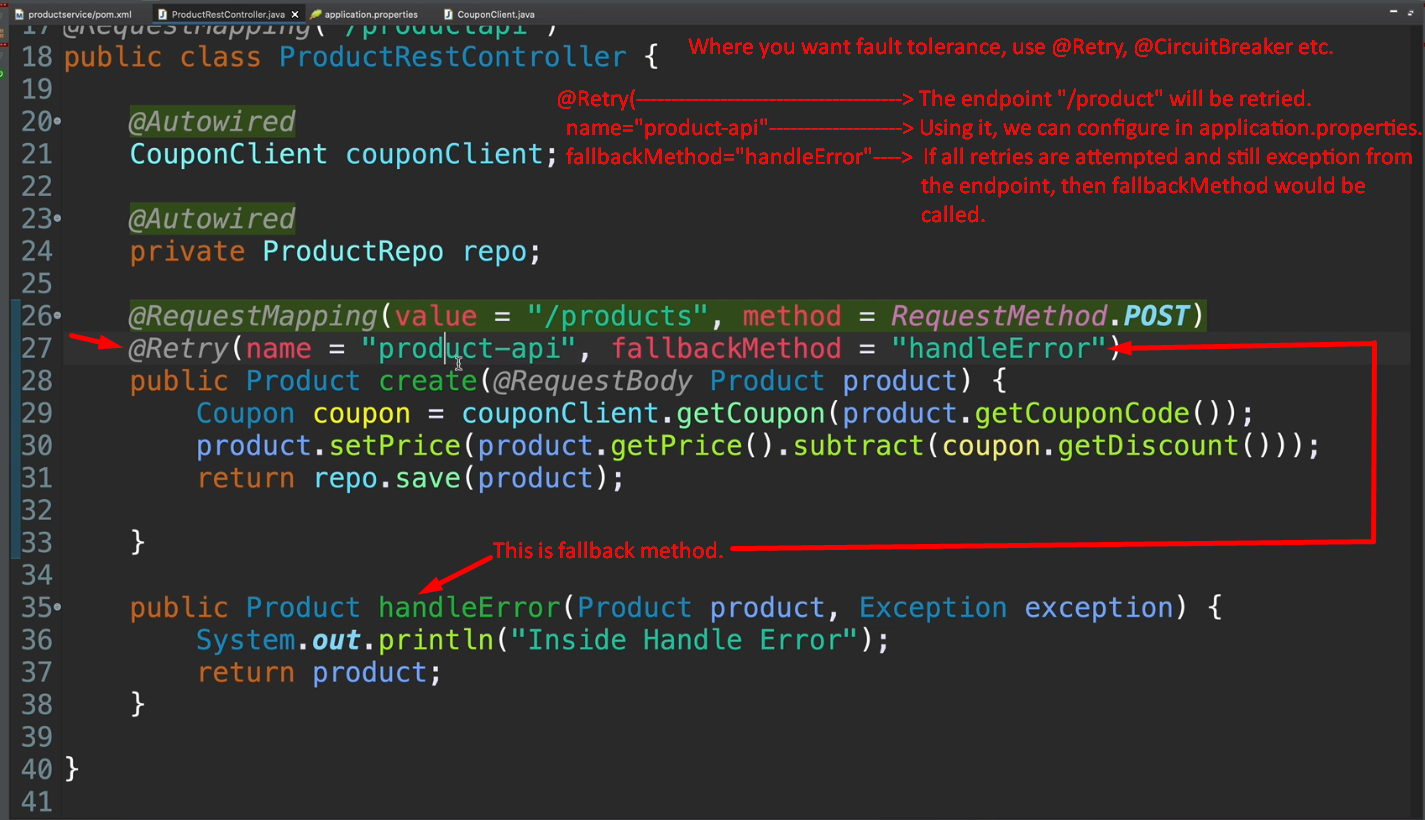
1. We don’t want if one microservice goes down, the entire system goes down.
2. Actually, in that case it should retry or handle the errors gracefully.  
   For example, If X’s endpoint microservice is using Y’s endpoint and it is down, then X’s endpoint must be invoked again (retry) or it should handle the error/exception gracefully.
3. That is where Resilience4j comes into picture.
4. It allows our application to do **fault tolerance**.
5. 
6. 
7. 
8. 
9. 