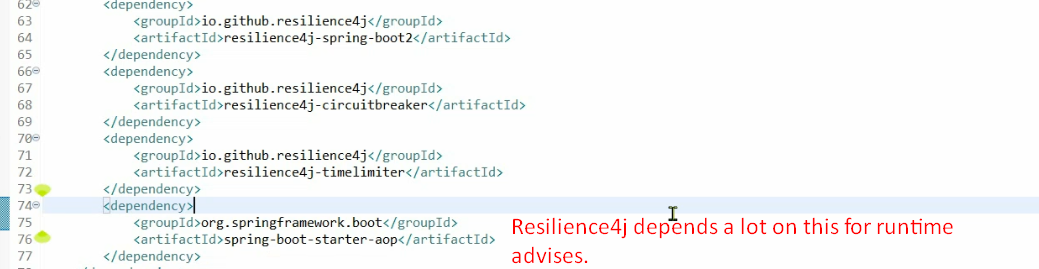
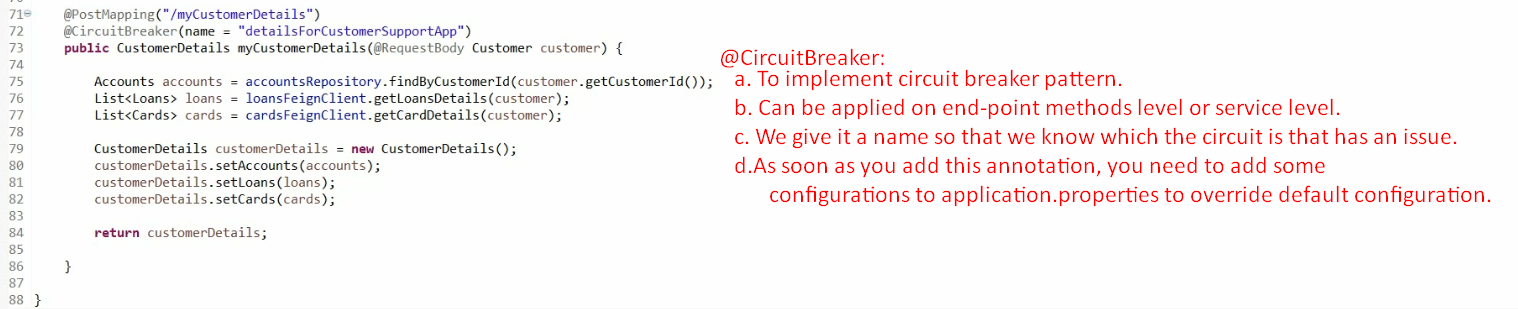
1. 
2. @**CircuitBreaker**:
   1. To implement circuit breaker pattern.
   2. Can be applied on end-point methods level or service level.
   3. We give it a name so that we know which the circuit is that has an issue and can be configured using this name.
   4. As soon as you add this annotation, you need to add some configurations to application.properties to override default configuration.
3. 
4. After adding this annotation @CircuitBreaker, we can configure the default behaviour of circuit breaker pattern.
5. **Properties**:
   1. 
      1. The purpose of this property is to inform Resiliency4j to register all circuit related health information to actuator so that I can see inside my actuator what kinds of circuit breakers that I have, what are their health, their state etc.
      2. By default, this is disable.
   2. 
      1. Circuit Breaker Pattern would monitor 5 calls to take decision whether it should go into OPEN state or not.
   3. 
      1. If 50% of calls which are being monitored (in our case, it is 5) are failed, the circuit would go into OPEN state.  
         So, 50% of 5 calls = 3 calls.  
         Means, it will fail fast when will give default configured response back.
      2. Default value is 90%.
   4. 
      1. After 30 seconds (30000miliseconds), it will come out of OPEN state into HALF\_OPEN state.
      2. Default is 90seconds.
   5. 
      1. When the circuit is in HALF\_OPEN state, only 2 calls are allowed.
      2. If 50% of them gets failed, the circuit will go into OPEN state.