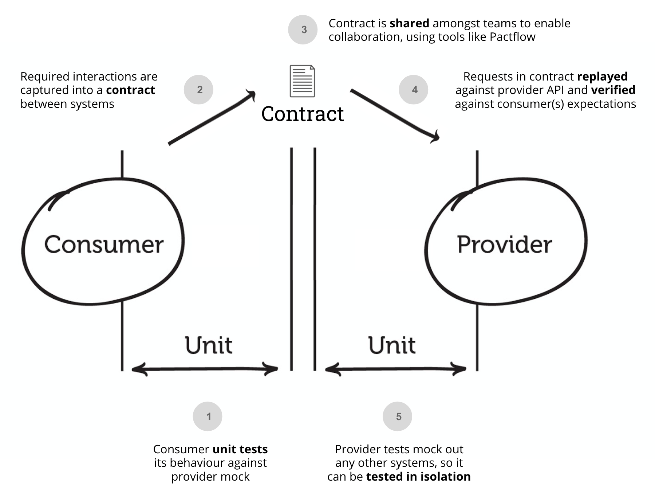
1. How to add request headers while using RestTemplate and Feign client?

  
2. Difference between rest template and feign client  
3. Micro services design patterns (CQRS, EVENT Sourcing, SAGA, Circuit Breaker, API Gateway)  
4. Core java design patterns  
5. Core spring (IOC, DI, Life cycle bean, how to create the bean) - http://dolszewski.com/spring/spring-bean/  
6. Contract Testing



***Contract testing*** is a technique for testing an integration point by checking each application in isolation to ensure the messages it sends or receives conform to a shared understanding that is documented in a "contract".

For applications that communicate via HTTP, these "messages" would be the HTTP request and response, and for an application that used queues, this would be the message that goes on the queue.

In practice, a common way of implementing contract tests (and the way Pact does it) is to check that all the calls to your test doubles [return the same results](https://martinfowler.com/bliki/ContractTest.html) as a call to the real application would.  
7. Usages of Gateway servers 🡪  
8. Repository testing (DB Testing), Unit Testing, Integration testing  
9. What are micro services and their advantages  
10. Rest concept  
11. Why spring boot and advantages of Spring boot   
12. TDD, BDD  
13. Spring validation (<https://reflectoring.io/bean-validation-with-spring-boot/>)   
14. Junit testing and Integration testing  
15. Bean wiring – Associating one object with another / Either by constructor or setters  
16. Auto wiring- Default is no, byName, byType, constructor, autodetect  
17. @Qualifier 🡪 It's worth noting that **if both the***@Qualifier***and***@Primary***annotations are present, then the***@Qualifier***annotation will have precedence.** Basically, *@Primary* defines a default, while *@Qualifier*is very specific.  
18. @Controller and @RestController ( controller + Response Body )   
19. Limitations of Mockito

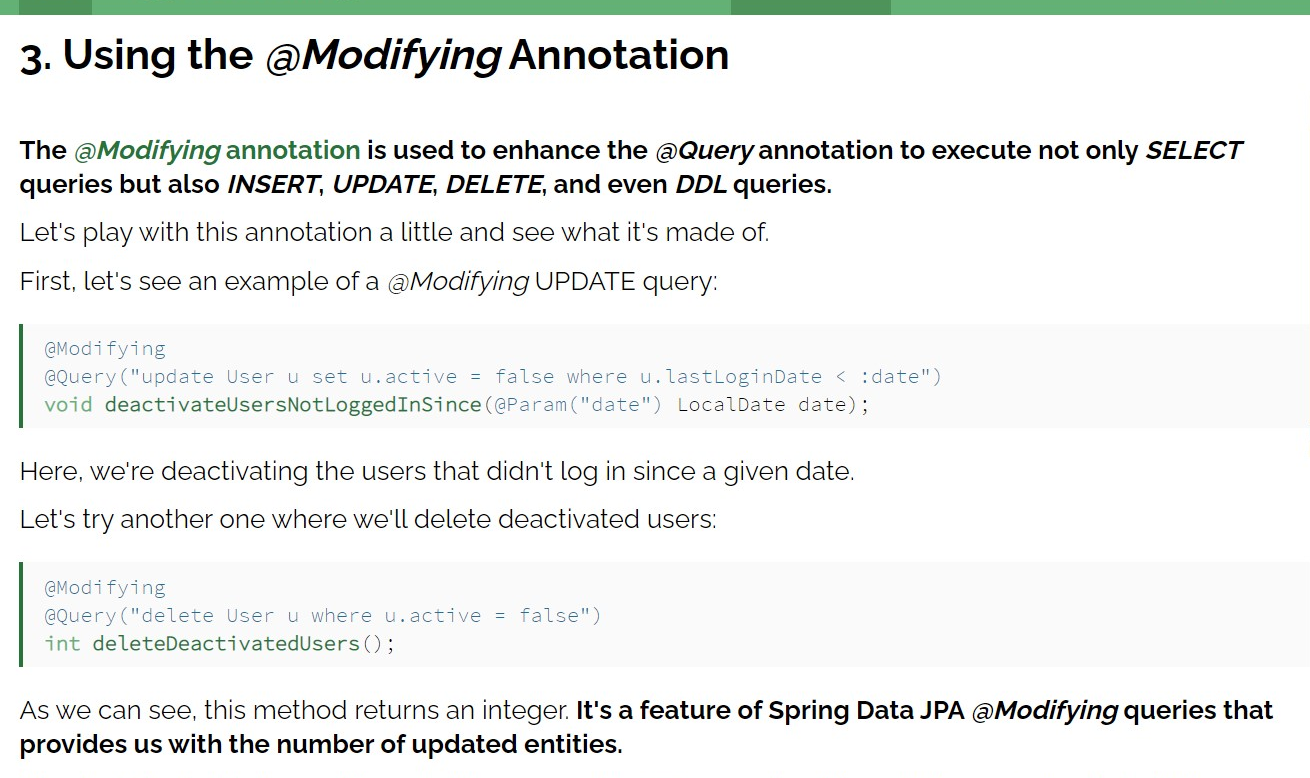
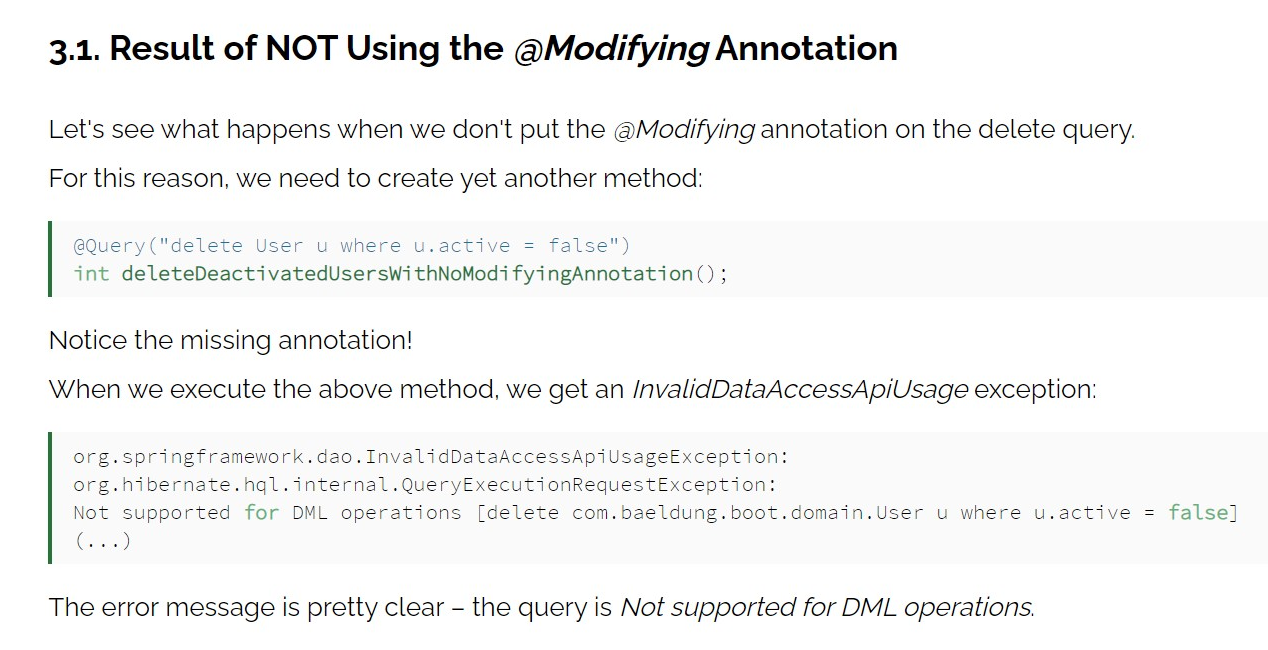
It cannot mock constructors or static methods.

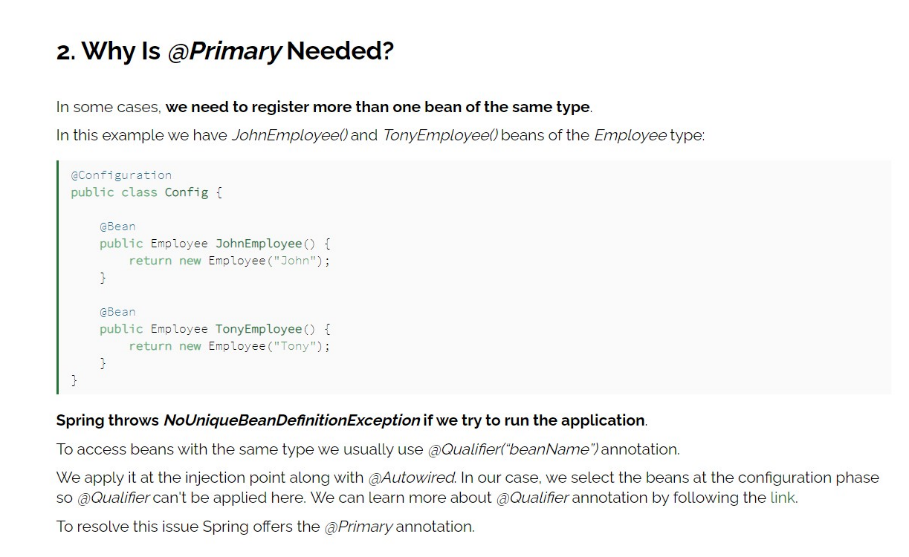
It requires Java version 6 plus to run.

It also cannot mock equals(), hashCode() methods.

VM mocking is only possible on VMs that are supported by Objenesis.

20. Version Controller (Git merge vs rebase ament, cherrypic)  
21. Different ways to handle null pointer exception  
22. 12 factor app (IMP) – <https://www.baeldung.com/spring-boot-12-factor> Steps to improve SB MS services : codebase ,logs, concurrency , dev and prod parity( diff should be min), stateless processes, build-release-run model , configurations, dependencies.   
23. What is Spring cloud( Service registration and discovery,Routing,Service-to-service calls,Load balancing,Circuit Breakers)   
24. Spring cloud bus  
25. Spring cache  
26. Lombok (@Builder)  
27. Hexagonal architecture (IMP)  
28. @Inject - <https://www.baeldung.com/spring-annotations-resource-inject-autowire>   
29. Completable feature  
30. Monitoring Mechanism  
31. @Modifying annotation

   
32. What is the use of Spring Data JPA  
33. Circuit breaker (fall back per exception mechanism)  
34. @Primary - https://www.baeldung.com/spring-primary

  
35. Patch  
36.HTTP status codes   
37. CI/CD

38. Richardson Maturity Model

<https://restfulapi.net/richardson-maturity-model/>

1. Solid Principles  
2. Java 8 concepts (Functional interface, stream API, FlatMap, StringUtils, Why stream are lazy?, Parallel stream, List iterator)  
3. Design patterns (where you have used design patterns and each design pattern usage in project)  
4. What is Stream API  
5. What is terminal operation  
6. What if you don’t use the terminal operations   
1. Java memory model   
2. Garbage collection   
3. JVM, JDK, JRE  
4. double check locking  
5. transient, volatile  
6. multi thread programing (notify, notifyall)  
7. concurent map  
8. Why Enums are type safe in java  
9. interface vs abstract  
10. What is the need of overriding hashcode and equals method  
11. fail safe and fail fast iterator  
12. comparator vs comparable   
13. how many ways we can handle exception   
14. checked and unchecked exception