



Repository layer testing using Junit

Step 1 make sure application is ready with all functionalities
copied 4.1 demo

Step 2 make sure spring boot test dependency added in pom

```
34 <dependency>  
35 <groupId>org.springframework.boot</groupId>  
36 <artifactId>spring-boot-starter-test</artifactId>  
37 <scope>test</scope>  
38 </dependency>
```

Step 3 Create a test file under test package
and inject dependency

```
19 @ExtendWith(SpringExtension.class) // integrates with Junit  
20 @DataMongoTest // integrated data class + db  
21 public class CustomerRepositoryTest {  
    8 usages  
22     @Autowired  
23     private CustomerRepository customerRepository;
```

Step 4 setup() / clean()
 @BeforeEach
 @AfterEach

make sure required objects are created with proper data before each test case

```
25       9 usages  
      private Customer customer;  
26       3 usages  
      private Address address;  
27  
28       @BeforeEach  
29       public void setup(){  
30           address = new Address( doorNo: "123", street: "s1", area: "areal", city: "city1");  
31           customer = new Customer( customerId: "C1", name: "krishna", email: "nomail.com", mobile: "1234",address);  
32       }
```

make sure objects are getting null after each test case

```
34       @AfterEach  
35       public void clean(){  
36           address=null;  
37           customer=null;  
38           customerRepository.deleteAll();  
39       }
```

5

test addrecord

customerRepo.insert(customer)

Step 5 Write test case methods

```
41       @Test  
42       public void getAllCustomers(){  
43           customerRepository.insert(customer); // C1  
44           customer.setCustomerId("C2");  
45           customerRepository.insert(customer);  
46           List<Customer> customers = customerRepository.findAll();  
47           assertEquals( expected: 2,customers.size());  
48       }
```

```
50      @Test
51      public void givenCustomerDetailsToSaveReturnCustomer(){
52          customerRepository.insert(customer);
53          Customer result=customerRepository.findById(customer.getCustomerId()).get();
54          System.out.println((result.equals(customer)));
55          assertEquals(result,customer);
56          //assertEquals(result.getName(), customer.getName());
57      }

59      @Test
60      public void givenCustomerToDelete(){
61          customerRepository.insert(customer);
62          customerRepository.deleteById(customer.getCustomerId());
63          assertEquals( expected: Optional.empty(), customerRepository.findById(customer.getCustomerId()));
64      }
```