SECTION 03 - TESTING CUSTOMER REGISTRATION FEATURE

Index

- Intro

- Once you test a unit in isolation then you guaratee it works, so you don't have to test it again in other classes. All you have to do is simply mock the unit already tested.

- In case of a service I want to test the unit in isolation, so everything within the outside world is mocked.

- If I want to make sure that the mock receives the correct arguments I should use an Argument Captor.

- Imports

- import **static** org.assertj.core.api.Assertions.**assertThat**;
- import **static** org.mockito.BDDMockito.**given**;
- import **static** org.mockito.BDDMockito.**then**;

- Mocking

- Add on top of the test class: @ExtendWith(MockitoExtension.class)

- For mocking a class:

```
@Mock
private CustomerRepository customerRepository;
```

- To instantiate and inject the mock in the actual class:

```
@InjectMocks
private CustomerRegistrationService underTest;
```

- Working with mocks

→ Example:

```
given(customerRepository.selectCustomerByPhoneNumber(customer.getPhoneNumber()))
            .willReturn(Optional.empty());

then(customerRepository).should().save(customerArgumentCaptor.capture());
```

- Argument Captor example

```
@Captor
private ArgumentCaptor<Customer> customerCaptor;

then(customerRepository).should().save(customerCaptor.capture());

assertThat(customerCaptor.getValue()).isEqualTo(customer);
```

- Should Never

- It’s used to check that a certain method never will be invoked.

- import **static** org.mockito.Mockito.**never**;
- import **static** org.mockito.ArgumentMatchers.**any**;

→ Example:

```
then(customerRepository).should(never()).save(any());
```

- Handling exceptions

- Import: import static org.assertj.core.api.Assertions.assertThatThrownBy;

```
assertThatThrownBy(() -> underTest.registerNewCustomer(request))
    .hasMessageContaining(String.format("phone number [%s] is taken", phoneNumber))
    .isInstanceOf(IllegalStateException.class);
```

- Allow scanning of @Column annotations

```
@DataJpaTest(
        properties = { "spring.jpa.properties.javax.persistence.validation.mode=none" }
        )
```