

SECTION 04 - PAYMENT FEATURE AND TESTS

Index

- Intro.....	2
- Mockito.mock.....	2
- shouldHaveNoInteractions: another way to .should(never()).....	2

- Intro

- In this unit I can see the power of using interfaces because still I don't have neither implementation of Stripe (for registering a payment) nor twilio (for sending a text message). So I can simply pass or inject the interface and then simply mock the invocations of that interface.
- This is because when we test, we should do it in isolation so although I would have the implementation of Stripe, I should use a mock of it when testing Payment service.

- Mockito.mock

- By using Mockito.mock I don't need to instantiate an object to pass it into a mocked method.

Example:

```
given(customerRepository.findById(customerId))  
  .willReturn(Optional.of(Mockito.mock(Customer.class)));
```

Instead of:

```
given(customerRepository.findById(customerId))  
  .willReturn(Optional.of(new Customer(...)));
```

- shouldHaveNoInteractions: another way to .should(never())

- Instead of:
 then(paymentRepository).should(never()).save(any(Payment.class));
I can use:
 then(cardPaymentCharger).**shouldHaveNoInteractions**();