Raviteja Kommalapati
raviteja.kommalapati@sjsu.edu
Junit Demo Document (03/06/2019)

Null Check

Possible Scenarios

1)we get a coplex transaction with more parameters into our APIS, so there is a chase that dev team may not map them correctly to the inner classes.

2)More the parameters more concentration should be taken to validate each.

Business impact

1)May be a parameter variable is not important in our service (microservice) But it will for sure a important attribute for other services.

Equals Check

Possible Scenarios

1)When parameters are more there is quite possible for human errors.

Example: res.setStartTime(input.getStartTime());

res.setEndTime(input.getStartTime());→Human error.

Business Impact:

Just imagine in business displaying card expiry date same as card activation date.

In Junit we use assertEquals to validate this kind of issues

```
@Test
public void repalceCardNumberSuccess() {
String expected_CardNumber= "XXXXXXXXXXX2049";
assertEquals(expected_CardNumber,
                    exmp.repalceCardNumber("149402940492049"));
     }
```

//Original java method. That replace the card Number

```
final String REPLACER="XXXXXXXXXXX";
public String repalceCardNumber(String incomingCardNumber) {
//Null check and length check
if(incomingCardNumber ==null||
incomingCardNumber.length()!=15)return null;

     String res = null;
          res = REPLACER+ incomingCardNumber.substring(11,15);

     return res;
     }
```

Exception Check (Start checking with Null Pointer Exception)

Possible Scenarios:

It is common when developing enterprise API Application class will have sub class ,sub class will have sub class in it.So, all together form as a single transaction from the API. While building super class there is chance that sub class have set to parameters but its object is not created

Example:

Employee employee = new Employee();

employee.**getAddress().**setPinCode("1222"); →getAddress is null if we did not created object and set into it.

employee.setAddress(**new Address());**

Business Impact:

Not only Null check all different kinds of exception (Format Exceptions,SQl Exceptions,Service Layer Exceptions ,(All **Runtime Exceptions**)should be identified in Junit testing).

API Will crash during production if Exceptions are not handled properly.

```
@Test(expected=NullPointerException.class)
	public void exception_NullPointerTest() {
		exmp.moveZeroesToEnd(null);
	}
```

DateTime Check

Possible Scenarios

1)In java we may use different APIS (dependencies) to convert the string to datetime but it is important to make conversations correct.

Business impact:

Uneven date display to the user and also errors in conditions validations (completion of subscription will give error).

Performance Testing(timeout Junit Test Case)

Possible Scenarios:

It is important to write a method which have good Algorithm to solve the given problem. If unnecessary creation of connections unnecessary creation of objects increase the execution time and memory of the API.

Business Impact:

API crash during load test and also crash during high volume of incoming data.

```
@Test(timeout=1000)
	public void repalceCardNumber_performanceTest() {
		for(int i=0;i<1000000;i++)
```

```
        assertEquals(expected_CardNumber,
    exmp.repalceCardNumber("149402940492049"));
}
```

Mockito and PowerMockito

While testing I faced many in creating stubs for dependent classes and testing private methods
So Mockito, PowerMock is used for mocking dependent classes
Mainly :
When(dependent call).**thenReturn**(some object);
When(dependent call).**thenThrow**(new SomeException());
When then of Mockito is one main method we used in junit testing.
Power Mockito is used to test private methods.