

Testing - final

Will start
at 9:10 PM

Agenda

① Mocking

② Types of Doubles

Mocks
Fakes
Stubs

③ WebMvc Mock + Mockito

↳ Test our product
and category APIs

Mocking

Unit → testing a
block of
code in
isolation
Integration
Test

```
→ class ProductService {  
    ProductRepository pr;  
  
    → findProducts ByTitle (String title) {  
        → pr. ( ) ;  
    }  
}
```

→ I create a test case for ps. find Product ByTitle.

→ ① Create 4 products

↳ 2 of them had Delhi in title

② Called ps. find PBT (Selhi)

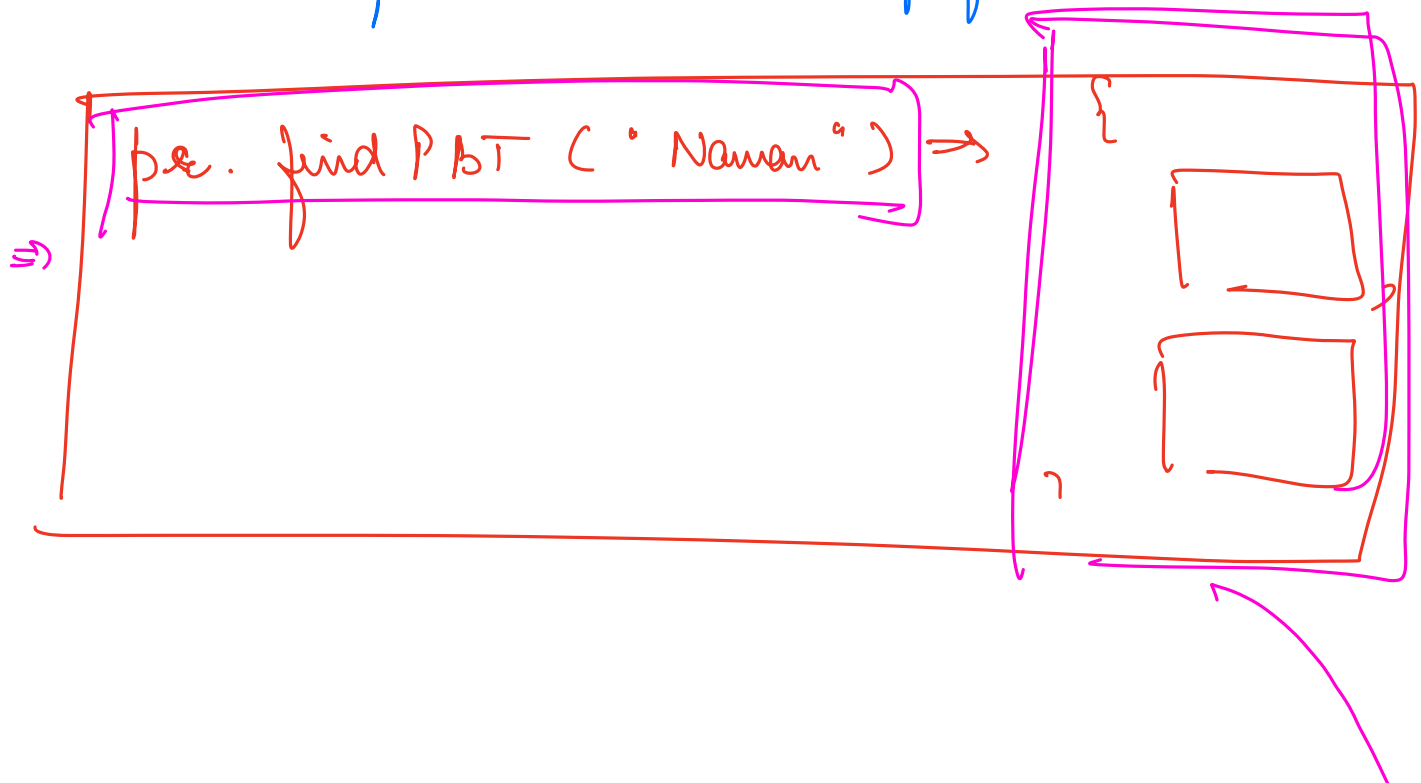
③ Checked y I got 2 products.

Oh!!
↓
The products that I got were diff.
⇒ I can't deterministically say if error is in product service. Because error can also be in product repository or any other dependency.

⇒ We are not testing our code in isolation.

⇒ We use concept of Mocking

⇒ when we will create our test case, we will fix the behaviour that dependencies should perform.



Product Service {

→ find Product By Title (String title) {

ps find PBT (name);

→ Now also if some unexpected Behaviour is coming, that is because of issue in PS.
Why? → Because I have hard coded what dependencies should return.

Unit Testing → code to be tested in isolation

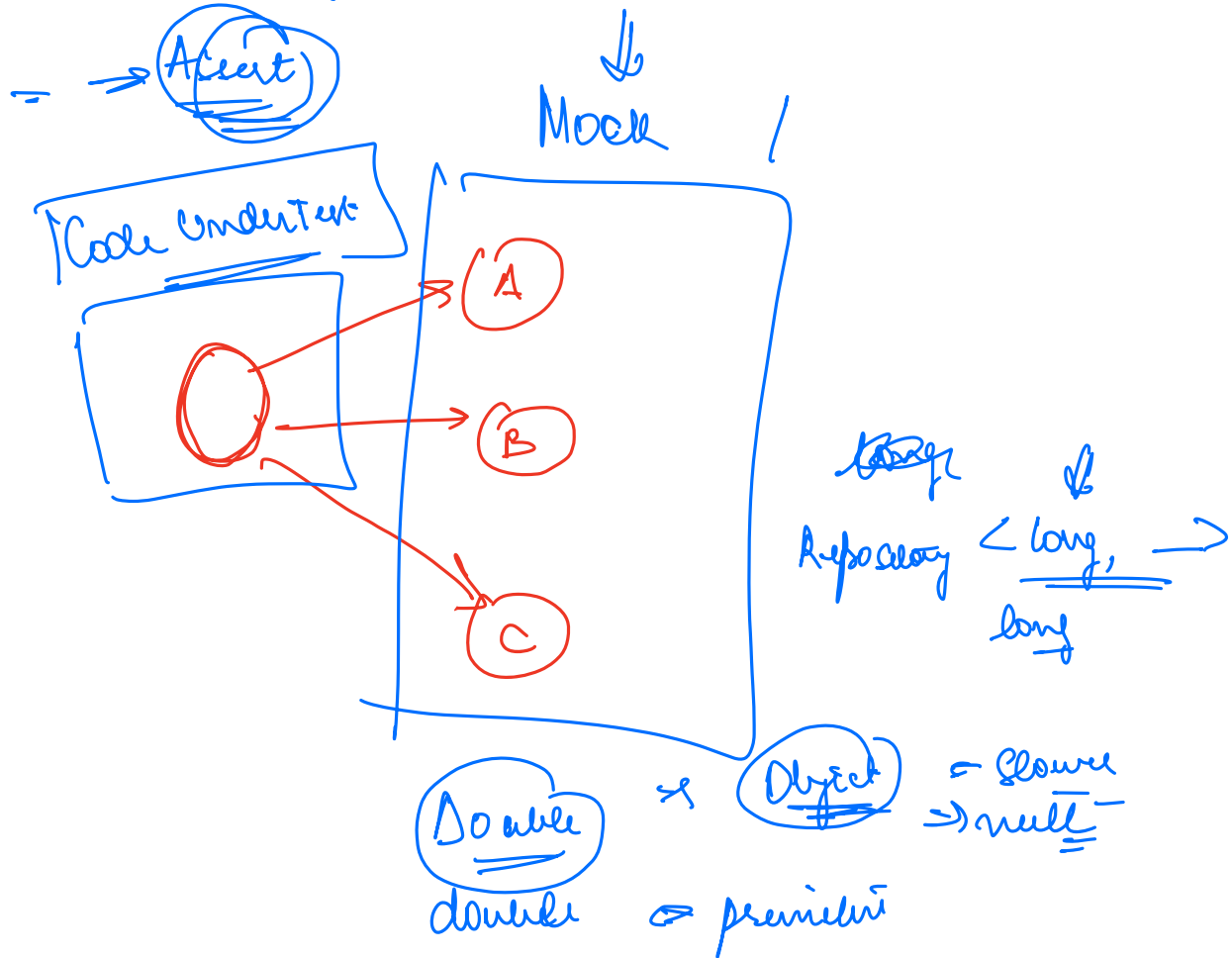


Mock how each dependencies should behave.



Call the code that you want to test.





long id = 0
 long id = null → clear

Back at 10:25 PM

H/W → Write Mock MVC tests for all of those
 7 APIs.

→ for each API ensure you have written

both | good, bad, edgy scenarios