

Testing Microservicios con Mock Frameworks

Dummies

Dummies are dependencies required by the test only when it creates or invokes the unit under test. Common examples include objects passed to constructor methods that are mandatory but never used during the test.

Fakes

Fakes replace dependencies with a different implementation. The implementation is simpler and easier to manage during the test phase, but not suitable for production usage. Examples of fakes are in-memory databases, or allow-all authenticators.

Stubs

Stubs provide the code unit under test with canned responses. Use stubs when testing for expected behavior and responses of the unit against well-known dependency responses. Common examples include methods that always return the same known result.

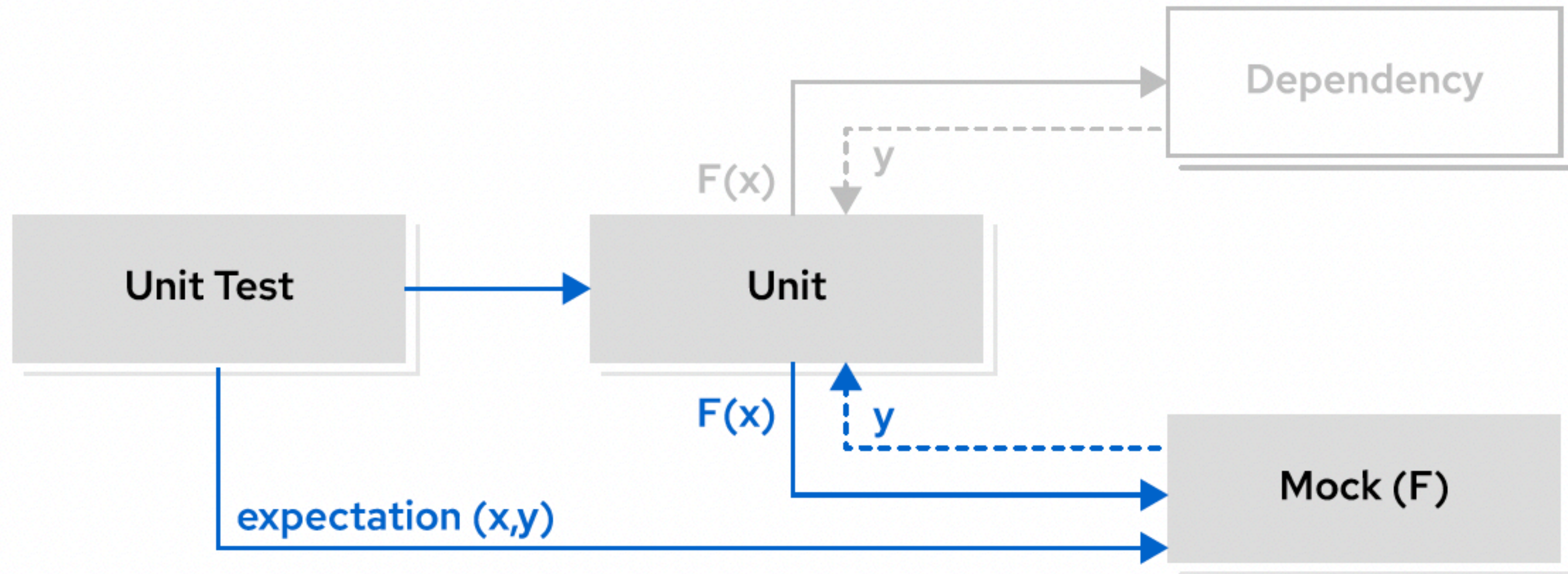
Mocks

Mocks are dependencies pre-programmed to respond to a limited set of requests. Use mocks when the original dependencies are too complex to use a stub. Examples of mocks include objects that mimic REST clients, or services.

Spies

Spies are stubs or mocks that record interactions with the unit under test. Unit tests can then make assertions over the interactions.

Tipo de Mock



Testing indirect interactions

`Mockito.verify(objeto mockeado , # calls de un mock method)`

- `Mockito.times(int wantedNumberOfInvocations)`: verifies that the code calls the method a specified number of times.
- `Mockito.atMost(int maxNumberOfInvocations)`: verifies that the code does not call the method more than a specified number of times.
- `Mockito.never()`: verifies that the code never calls the method.
- `Mockito.atLeastOnce()`: verifies that the code calls the method one or more times.

No puedes usar Mockito para mocking
Panache entities, si los entities usan el patrón
active record. Hay que usar PanacheMock

Spies

