

# Spring Boot Testing with Mockito

Spring boot supports the concept of test splices. Test Splices bring up a targeted segment of auto-configured spring boot environment.

- i.e. just the database components; or just the web components.
- User defined Spring beans typically are not initialised

@WebMvcTest -> is a spring boot test splice which creates a MockMvc environment for the controller(or controllers) under test.

- Dependencies controllers are not included

We do this using @MockBean given by spring boot starter test

Types of Mocks:

1. Dummy: Object used to get the code to compile
2. Fake: An object that has an implementation but not production ready
3. Stub: An object with pre-defined answers to method calls
4. Mocks: An object with pre-defined answers to method calls, and has expectations of executions. Can throw an exception if an unexpected invocation is detected.
5. Spy: In Mockito Spies are Mock like wrappers around the actual object.

Argument Captors can be used to verify request data is properly being parsed and passed to service layer.

Verify interactions can be used Mocked objects was called

Mock return values supply data back to controller

I.e. - object returned when getByld is called on service

Mocks can be instructed to throw expectations to test exceptional handling.

Junit Test -> <- MockMvc -> <- Controller Under Test -> <- Mockito

Mockito handles all the dependencies of the controller by mocking them.

MockMvc will raise request handled by controller and return values which we will be testing.