

Spring Boot @MockBean



MockAnnotationTest.java

```
import org.mockito.Mock;
import org.mockito.InjectMocks;
...

@SpringBootTest(classes=MvcTestingExampleApplication.class)
public class MockAnnotationTest {

    @Mock
    private ApplicationDao applicationDao;

    @InjectMocks
    private ApplicationService applicationService;

    ...

}
```

Inject mock dependencies

Note: Will only inject dependencies annotated with @Mock or @Spy

Spring Boot @MockBean

- Instead of using Mockito: @Mock and @InjectMocks
- Use Spring Boot support: @MockBean and @Autowired
- @MockBean
 - includes Mockito @Mock functionality
 - also adds mock bean to Spring ApplicationContext
 - if existing bean is there, the mock bean will replace it
 - thus making the mock bean available for injection with @Autowired

**Use Spring Boot @MockBean
when you need to inject mocks
AND
inject regular beans from app context**

BEFORE

MockAnnotationTest.java

```
import org.mockito.Mock;
import org.mockito.InjectMocks;
...

@SpringBootTest(classes=MvcTestingExampleApplication.class)
public class MockAnnotationTest {

    @Mock
    private ApplicationDao applicationDao;

    @InjectMocks
    private ApplicationService applicationService;

    ...
}
```

AFTER

MockAnnotationTest.java

```
import org.springframework.boot.test.mock.mockito.MockBean;
import org.springframework.beans.factory.annotation.Autowired;
...

@SpringBootTest(classes=MvcTestingExampleApplication.class)
public class MockAnnotationTest {

    @MockBean
    private ApplicationDao applicationDao;

    @Autowired
    private ApplicationService applicationService;

    ...
}
```

Create Mock for the DAO

Inject dependencies

No longer a limitation ...
can inject any dependency ...
mock or regular bean