

# اختبارات في Spring

التفاعلات: وتأكيد طرق، استبدال وهمية، كائنات لإنشاء الخطوات هذه اتبع، اختبارات في Spring لاستخدام

## 1. الاعتماديات إضافة

أضف `Mockito` و `JUnit` إلى ملف `pom.xml`:

```
<dependency>
  <groupId>org.mockito</groupId>
  <artifactId>mockito-core</artifactId>
  <version>5.3.1</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter</artifactId>
  <version>5.9.2</version>
  <scope>test</scope>
</dependency>
```

## 2. الاختبار فئة إعداد

امتداد مع 5 `Test` استخدام:

```
import org.junit.jupiter.api.extension.ExtendWith;
import org.mockito.junit.jupiter.MockitoExtension;

@ExtendWith(MockitoExtension.class)
public class UserServiceTest {
  // ...
}
```

## 3. الاعتماديات وإدخال وهمية كائنات إنشاء

- استبدالها. تريد التي للاعتماديات `@Mock` استخدام
- الاختبار. فئة إلى الهمية الكائنات لإدخال `@InjectMocks` استخدام

```
@Mock
private UserRepository userRepository;
```

```
@InjectMocks
private UserService userService; // UserService      UserRepository
```

#### 4. الطرق استدعاءات استبدال

الوهمي: الكائن سلوك لتحديد when().thenReturn() استخدم

```
@Test
void getUserById_ReturnsUser_WhenUserExists() {
    User mockUser = new User(1L, "Alice");
    //
    when(userRepository.findById(1L)).thenReturn(mockUser);

    User result = userService.getUserById(1L);

    assertEquals("Alice", result.getName());
}
```

#### 5. التفاعلات تأكيد

متوقع: هو كما استدعاؤها تم الوهمي الكائن طريقة أن من تحقق

```
@Test
void getUserById_CallsRepository() {
    userService.getUserById(1L);
    //
    verify(userRepository, times(1)).findById(1L);
}
```

---

شائعة استخدام حالات

الاستثناءات استبدال

```
@Test
void getUserById_ThrowsException_WhenRepositoryFails() {
    when(userRepository.findById(anyLong())).thenThrow(new RuntimeException("DB Error"));

    assertThrows(RuntimeException.class, () -> userService.getUserById(1L));
}
```

مزن: بشكل ال عمل لمطابقة إلخ, any(), eq() استخدم ال عمل لمطابقة

```
when(userRepository.findById(anyLong())).thenReturn(new User(1L, "Bob"));
```

ال قراءة: من لمزيد given().willReturn() استخدم

```
import static org.mockito.BDDMockito.*;
```

```
@Test
```

```
void bddStyleExample() {  
    given(userRepository.findById(1L)).willReturn(new User(1L, "Alice"));  
  
    User result = userService.getUserById(1L);  
  
    then(userRepository).should().findById(1L);  
    assertEquals("Alice", result.getName());  
}
```

تفصيلية: تأكيدات لإجراء ال عمل لمات أسد ال عمل لمات مقبض

```
@Test
```

```
void saveUser_CapturesArgument() {  
    ArgumentCaptor<User> userCaptor = ArgumentCaptor.forClass(User.class);  
  
    userService.saveUser("Charlie");  
  
    verify(userRepository).save(userCaptor.capture());  
    assertEquals("Charlie", userCaptor.getValue().getName());  
}
```

رئىسية ملاحظات

[] MockitoAnnotations.openMocks(this) أو [] 5 [] @ExtendWith(MockitoExtension.class) استخدم: ال تنشيط []  
[] @BeforeEach في

[] ال ثابتة [] ال مبانى. ال طرق ل استبدال mockito-inline اعتمادية استخدم: ال ثابتة ال وهمية ال كائنات

[] جزئى. [] استبدال ال حقيقي ال كائنات لتغلي [] @Spy استخدم: ال جواسيس []

## الأخطاء إصلاح

- `openMocks()` أو `@ExtendWith` □ استخدم تنشيدها تم الوهمية الكائنات أن من تأكد: ال فارغة الوهمية الكائنات □
- الاختبار. في استدعائها قبل الطرق استبدل: الاستبدال أخطاء □

□□□□□□□□. توثيق راجع المتقدم، للمستخدم