

Spring Testing

Bledar Aga

<http://scg.unibe.ch/wiki/students/bledaraga>



^b
**UNIVERSITÄT
BERN**

Main scenarios:

- normal unit test —> no dep. with classes in the framework
- with mocking —> dep. with framework

```
public interface UserDao extends CrudRepository<User, Long> {  
    User findByEmail(String emailAddress);  
}
```

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"file:src/.../springMVC.xml", "file:src/.../springData.xml"})
@Transactional
@TransactionConfiguration(defaultRollback = true)
public class UserDaoIntegrationTest {
```

normal unit test

```
@Autowired
UserDao userDao;
```

```
@Test
```

```
public void testTeamReference() {
    String TEAMNAME = "testTeam";
```

```
    Team team = new Team();
    team.setName(TEAMNAME);
    User user = new User();
    user.setTeam(team);
```

```
    assertNull(user.getId());
```

```
    user = userDao.save(user);
```

```
    assertNotNull(user.getId());
    assertTrue(user.getId() > 0);
    assertEquals(user.getTeam().getName(), TEAMNAME);
```

```
}
```

```
@Test
```

```
public void testFindUserByEmail() {
    String EMAIL = "test@test.test";
```

```
    User user = new User();
    user.setEmail(EMAIL);
    user = userDao.save(user);
    User findUser = userDao.findByEmail(EMAIL);
```

```
    assertEquals(findUser.getEmail(), EMAIL);
```

```
}
```

```
}
```

mock testing framework

Mockito

- allows to mock a class or interface

```
//You can mock concrete classes, not only interfaces  
LinkedList mockedList = mock(LinkedList.class);
```

```
//stubbing - before execution  
when(mockedList.get(0)).thenReturn("first");
```

```
//following prints "first"  
System.out.println(mockedList.get(0));
```

```
//following prints "null" because get(999) was not stubbed  
System.out.println(mockedList.get(999));
```

@Service

```
public class SampleServiceImpl implements SampleService {
```

```
    @Autowired    UserDao userDao;
```

```
    public SampleServiceImpl() {  
    }
```

```
    @Autowired
```

```
    public SampleServiceImpl(UserDao userDao) {  
        this.userDao = userDao;  
    }
```

```
    @Transactional
```

```
    public SignupForm saveFrom(SignupForm signupForm) throws InvalidUserException{  
        String firstName = signupForm.getFirstName();  
        if(!StringUtils.isEmpty(firstName) && "ESE".equalsIgnoreCase(firstName)) {  
            throw new InvalidUserException("Sorry, ESE is not a valid name");  
        }
```

```
        User user = new User();  
        user.setFirstName(signupForm.getFirstName());  
        user.setEmail(signupForm.getEmail());  
        user.setLastName(signupForm.getLastName());
```

```
        user = userDao.save(user);    // save object to DB  
        signupForm.setId(user.getId());  
        return signupForm;
```

```
    }
```

```
}
```

define mock object

```
public class SampleServiceTest {  
    private UserDao userDao;  
    private SampleService sampleService;  
  
    @Before  
    public void doSetup() {  
        userDao = mock(UserDao.class);  
        sampleService = new SampleServiceImpl(userDao);  
    }  
  
    @Test  
    public void testSaveForm() {  
        SignupForm signupForm = new SignupForm();  
        signupForm.setLastName("formLast");  
        signupForm.setFirstName("formFirst");  
        signupForm.setEmail("form@test.com");  
  
        when(userDao.save(any(User.class)))  
            .thenReturn(new Answer<User>() {  
                public User answer(InvocationOnMock invocation) throws Throwable {  
                    User user = (User) invocation.getArguments()[0];  
                    user.setId(1L);  
                    return user;  
                }  
            });  
  
        assertNull(signupForm.getId());  
        signupForm = sampleService.saveFrom(signupForm);  
        assertNotNull(signupForm.getId());  
        assertTrue(signupForm.getId() > 0);  
    }  
}
```

Project configuration:

<https://github.com/ese-unibe-ch/ese2014-wiki/tree/master/Skeleton-Test>

References:

- mockito: <https://code.google.com/p/mockito/>
- tutorials: <http://www.luckyryan.com/2013/06/25/testing-spring-web-apps-series-index/>
- spring testing: <http://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/testing.html>