Testing with Scout Neon
Scout UGM 2016, Ludwigsburg

Judith Gull, BSI Business Systems Integration AG
Agenda

- Simple unit test
- Replacing Beans withMocks
- TestRunners
Test Runners

BlockJUnit4ClassRunner

PlatformTestRunner

ClientTestRunner

ServerTestRunner

JUnit

Scout
Custom Scout Test Runners

- Allows setting contextual info (session, subject)
- Custom ExceptionHandler (JUnitExceptionHandler)
- Convenience for mocking beans
**PlatformTestRunner**

*Use to run tests which require the IPlatform!*

```java
@RunWithNewPlatform
@RunWithSubject("testSubject")
@RunWith(PlatformTestRunner.class)
public class MyTest {

    @Times(100)
    @Test
    public void testXY() {
        ...
    }
}
```

- **New instead of globally shared platform**
- **Subject with Principle**
- **Execute multiple times to find problems**
ClientTestRunner
Runs inside ModelJob with ClientRunContext

```java
@RunWithClientSession(MySession.class)
@RunWith(ClientTestRunner.class)
public class MyTest {

    @Test
    public void testXY() {
        ...
    }
```
ServerTestRunner
Each Test Method is Executed in Separate Transaction
ServerTestRunner

Each Test Method is Executed in Separate Transaction

```java
@RunWith(ClientNotifications)
@RunWith(ServerSession(MySession.class))
@RunWith(ServerTestRunner.class)
public class MyTest {

    @Test
    public void testXY() {
        ...
    }
```

Sets clientId, Collector, ThreadLocals

Cached per subject (by default)
Manipulating BeanRegistry for Testing

```java
private List<IBean<?>> m_beans;

@Before
public void before() {
    SqlServiceMock sqlService = new SqlServiceMock();
    m_beans = TestingUtility.registerBeans(new BeanMetaData(ISqlService.class)
        .withInitialInstance(sqlService)
        .withApplicationScoped(true));
}

@After
public void after() {
    TestingUtility.unregisterBeans(m_beans);
}
```
Thanks!
@EclipseScout

judith.gull@gmail.com