jutf Documentation

Release 2.6.36

knightliao

Sep 27, 2021

Contents

Java Unit Test Framework (Warp H2/Mockito/jmockit tools to make java application better)

- Jutf: https://github.com/knightliao/jutf

:

CHAPTER 1

jutf

1.1 H2 Test

XML Configuration:

```
<configuration>
    <initialize-database>
        <schema>sql/schema/demo_schema.sql</schema>
    </initialize-database>
    <initialize-data>
        <data>sql/goldendata/demo_data.sql</data>
    </initialize-data>
</configuration>
```

H2 Test Code:

```
public class H2BaseTestCaseTest extends H2BaseTestCase {

    @Test
    public void foo() {

        try {

            String query = "select * from test.t_demo";
            List<Map<String, Object>> listOfMaps = executeSql(query);

            System.out.println(new Gson().toJson(listOfMaps));

            Assert.assertEquals(listOfMaps.size(), 2);

        } catch (SQLException se) {

            Assert.assertTrue(false);
            throw new RuntimeException("Couldn't query the database.", se);
```

(continues on next page)

(continued from previous page)

```
        }

    }
}
```

1.2 get/set/construct/tostring Test

```
@Test
public void test() {
    TestUtils.testAllClassUnderPackage("com.github.knightliao.test");
}
```

1.3 mockito Test

```
public class DemoServiceTest {

    @InjectMocks
    private DemoService demoService = new DemoService();

    @Spy
    private IUsedService usedService = new UsedServiceImpl();

    @Before
    public void setUp() {

        MockitoAnnotations.initMocks(this);
        Mockito.when(usedService.echo("hello")).thenReturn("world");

    }

    @Test
    public void testEcho() {

        String result = demoService.echo2("hello");
        Assert.assertEquals("world", result);

        result = demoService.echo2("hello world");
        Assert.assertEquals("hello world", result);
    }
}
```

1.4 logback message Test

```
public class LogbackCapturingAppenderTest {
    @After
    public void cleanUp() {
        LogbackCapturingAppender.Factory.cleanUp();
```

(continues on next page)

(continued from previous page)

```
    }

    @Test
    public void shouldCaptureAGivenLog() throws Exception {
        // Given
        LogbackCapturingAppender capturing = LogbackCapturingAppender.Factory.
↪weaveInto(OurDomainWithLogger.LOG);

        // when
        OurDomainWithLogger domainClass = new OurDomainWithLogger();
        domainClass.logThis("This should be logged");

        // then
        assertThat(capturing.getCapturedLogMessage(), is("This should be logged"));
    }

    @Test
    public void shouldNotCaptureAGiveLogAfterCleanUp() throws Exception {

        // Given
        LogbackCapturingAppender capturing = LogbackCapturingAppender.Factory.
↪weaveInto(OurDomainWithLogger.LOG);

        // when
        OurDomainWithLogger domainClass = new OurDomainWithLogger();
        domainClass.logThis("This should be logged at info");
        LogbackCapturingAppender.Factory.cleanUp();
        domainClass.logThis("This should not be logged");

        // then
        assertThat(capturing.getCapturedLogMessage(), is("This should be logged at␣
↪info"));
    }
}
```

CHAPTER 2

jutf-spring

2.1 H2 Test

```
@ContextConfiguration(locations = "classpath:spring-test.xml")
public class BaseTest extends AbstractTransactionalTest {
    @InjectMocks
    @Autowired
    DemoService demoService;

    @Mock
    UsedService usedService;

    @Autowired
    DemoDao demoDao;

    @Before
    public void setUp() {
        MockitoAnnotations.initMocks(this);
        Mockito.when(usedService.echo("hello")).thenReturn("world");
    }

    /**
     * databasemysql
     * demoSqlConfig
     * <p/>
     * demo.sql,
     */
    @Test
    @SqlConfig(database = Database.H2, sqlFiles = {"classpath:sql/goldendata/demo_
↪data.sql"})
    public void testEcho() {
        String result = demoService.echo("hello");
        Demo demo = demoDao.selectByPrimaryKey(1L);
        Assert.assertEquals("world", result);
```

(continues on next page)

(continued from previous page)

```
        Assert.assertEquals("demo", demo.getDemoValue());
    }
}
```