http://www.egovframe.go.kr/wiki/doku.php?id=egovframework:rte2:brte:batch_example:db_jdbccursor

# JdbcCursor Examples

## Outline

This Section describes how DBs are read line-by-line. by paragraph You can check out a step processed by a multitude of files. In Spring you can use either JdbcCursorItemReader or HibernateCursorItemReader.

## Description

### Settings

#### Configuring Jobs

##### Check out jdbcCursorIoJob.xml, the job configuration file for JdbcCursor example.

The following configurations are available in JdbcCursorItemReader:

- dataSource : Database
- sql : Executable queries
- verifyCursorPosition : Verifiable cursor position
- rowMapper : Mapper of ResultSet, the result of SQL execution

```
<bean id="itemReader" class="org.springframework.batch.item.database.JdbcCursorItemReader">
        <property name="dataSource" ref="dataSource"/>
        <property name="sql" value="select ID, NAME, CREDIT from CUSTOMER"/>
        <property name="verifyCursorPosition" value="true"/>
        <property name="rowMapper">
                <bean class="egovframework.brte.sample.common.domain.trade.CustomerCreditRowMapper"/>
        </property>
</bean>
```

### Composition and Implementation of JunitTest

#### Composition of JunitTest

##### Implement the JdbcCursor example and verify the result of batch by comprising @Test as follows:

✔ See Junit Test Description for Batch Execution for more information.
✔ Pre-batch data and post-batch data are to be compared to each other in EgovAbstractIoSampleTests.
✔ assertEquals(BatchStatus.COMPLETED, jobExecution.getStatus()): Check out the batch implementation is COMPLETED.

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = "/egovframework/batch/jobs/jdbcCursorIoJob.xml")
public class EgovJdbcCursorFunctionalTests extends EgovAbstractIoSampleTests {

        /**
         * Pre-test DB works
         */
        @Before
        public void setUp() {
                simpleJdbcTemplate.update("DELETE from CUSTOMER");
```

```
                simpleJdbcTemplate.update("INSERT INTO CUSTOMER (ID, VERSION, NAME, CREDIT)
VALUES    (1, 0, 'customer1', 100000)");
                simpleJdbcTemplate.update("INSERT INTO CUSTOMER (ID, VERSION, NAME, CREDIT)
VALUES    (2, 0, 'customer2', 100000)");
                simpleJdbcTemplate.update("INSERT INTO CUSTOMER (ID, VERSION, NAME, CREDIT)
VALUES    (3, 0, 'customer3', 100000)");
                simpleJdbcTemplate.update("INSERT INTO CUSTOMER (ID, VERSION, NAME, CREDIT)
VALUES    (4, 0, 'customer4', 100000)");

        }
      /**
       * The method to configure the reader when the batch result are read over again.
            */
        @Override
        protected void pointReaderToOutput(ItemReader<CustomerCredit> reader) {
                // no-op
        }

}
@ContextConfiguration(locations = { "/egovframework/batch/simple-job-launcher-context.xml",
"/egovframework/batch/job-runner-context.xml"})
@TestExecutionListeners( { DependencyInjectionTestExecutionListener.class,
StepScopeTestExecutionListener.class })
public abstract class EgovAbstractIoSampleTests {

        // JobLauncherTestUtils to test the batches.
        @Autowired
        @Qualifier("jobLauncherTestUtils")
        private JobLauncherTestUtils jobLauncherTestUtils;

        // Reader for batches
        @Autowired
        private ItemReader<CustomerCredit> reader;

        /**
         * Batch Testing
         */
        @Test
        public void testUpdateCredit() throws Exception {

                open(reader);
                List<CustomerCredit> inputs = getCredits(reader);
                close(reader);

                JobExecution jobExecution = jobLauncherTestUtils.launchJob(getUniqueJobParameters());
                assertEquals(BatchStatus.COMPLETED, jobExecution.getStatus());

                pointReaderToOutput(reader);
                open(reader);
                List<CustomerCredit> outputs = getCredits(reader);
                close(reader);

                assertEquals(inputs.size(), outputs.size());
                int itemCount = inputs.size();
                assertTrue(itemCount > 0);

                for (int i = 0; i < itemCount; i++) {

        assertEquals(inputs.get(i).getCredit().add(CustomerCreditIncreaseProcessor.FIXED_AMOUNT).intValue(),
                                        outputs.get(i).getCredit().intValue());
```

```
                }
            }
    ...
}
```

### Implementation of JunitTest

See [Implementation of JunitTest](#) for more information.

### Verification of Result

1. You can check out the cursors for DB in the query of console log. A singular data is read out of an attempt to read DB.

```
Tasks | Properties | Servers | Coverage | Bookmarks | Console | Javadoc | SVN Reposit | SQL Results | Execution Pl | DBIO Search | Query Result | JUnit | Search
<terminated> EgovJdbcCursorFunctionalTests [JUnit] C:₩eGovFrame-2.0₩bin₩jdk1.5.0_22₩bin₩javaw.exe (2012. 10. 4. 오후 3:26:33)
15:26:39,599  INFO main audit:156 - 2. Connection.getAutoCommit() returned true
15:26:39,600  INFO main audit:156 - 2. Connection.prepareStatement(select ID, NAME, CREDIT from CUSTOMER, 1003, 1007) returned net.sf.log4jdbc.
15:26:39,600  INFO main sqlonly:191 - select ID, NAME, CREDIT from CUSTOMER
15:26:39,601  INFO main sqltiming:235 - select ID, NAME, CREDIT from CUSTOMER  {executed in 0 msec}
15:26:39,602  INFO main audit:156 - 2. PreparedStatement.executeQuery() returned net.sf.log4jdbc.ResultSetSpy@10bc995
15:26:39,602  INFO main audit:156 - 2. PreparedStatement.getWarnings() returned null
15:26:39,604  INFO main audit:156 - 3. Connection.getAutoCommit() returned true
15:26:39,605  INFO main audit:156 - 3. Connection.getAutoCommit() returned true
15:26:39,605  INFO main audit:156 - 3. Connection.setAutoCommit(false) returned
15:26:39,607  INFO main audit:156 - 3. Connection.prepareStatement(UPDATE BATCH_STEP_EXECUTION_CONTEXT SET SHORT_CONTEXT = ?, SERIALIZED_CONTEX
15:26:39,607  INFO main audit:156 - 3. PreparedStatement.setString(1, "{"map":{"entry":{"string":"JdbcCursorItemReader.read.count","int":0}}}")
15:26:39,608  INFO main audit:156 - 3. PreparedStatement.setNull(2, 2005) returned
15:26:39,608  INFO main audit:156 - 3. PreparedStatement.setLong(3, 23) returned
15:26:39,608  INFO main sqlonly:191 - UPDATE BATCH_STEP_EXECUTION_CONTEXT SET SHORT_CONTEXT = '{"map":{"entry":{"string":"JdbcCursorItemReader.
SERIALIZED_CONTEXT = null WHERE STEP_EXECUTION_ID = 23
15:26:39,609  INFO main sqltiming:235 - UPDATE BATCH_STEP_EXECUTION_CONTEXT SET SHORT_CONTEXT = '{"map":{"entry":{"string":"JdbcCursorItemReade
```

2. You can check out the job is implemented and modified in the value Credit in the table Customer of DB.

Status | Result1

| | ID | VERSION | NAME | CREDIT |
|---|---|---|---|---|
| 1 | 1 | 0 | customer1 | 100005 |
| 2 | 2 | 0 | customer2 | 100005 |
| 3 | 3 | 0 | customer3 | 100005 |
| 4 | 4 | 0 | customer4 | 100005 |

# References

- [JdbcCursorItemReader](#)