

iBatis Examples

Outline

This Section describes how iBatis reads and writes DB. In Spring, You can check out a step processed by a multitude of files. In Spring IbatisPagingItemReader and IbatisBatchItemWriter are available.

Description

Settings

Configuring Jobs

Check out **ibatisIoJob.xml**, the job configuration file for iBatis example.

Interlocking with iBatis requires configuration of dataSource to be provided to the configuration file for sql-map-config via SqlMapClientFactoryBean. Both IbatisPagingItemReader and IbatisBatchItemWriter come into play to execute the query that are identical to queryId and statementId.

```
<bean id="itemReader" class="org.springframework.batch.item.database.IbatisPagingItemReader">
    <property name="queryId" value="getAllCustomerCredits" />
    <property name="sqlMapClient" ref="sqlMapClient" />
</bean>

<bean id="itemWriter" class="org.springframework.batch.item.database.IbatisBatchItemWriter">
    <property name="statementId" value="updateCredit" />
    <property name="sqlMapClient" ref="sqlMapClient" />
</bean>

<bean id="sqlMapClient" class="org.springframework.orm.ibatis.SqlMapClientFactoryBean">
    <property name="dataSource" ref="dataSource" />
    <property name="configLocation" value="ibatis-config.xml" />
</bean>
```

iBatis Configurations

Check out the iBatis configuration as follows:

sql-map-config

ibatis-config.xml

```
<sqlMapConfig>
    <sqlMap resource="ibatis-customer-credit.xml"/>
</sqlMapConfig>
```

sql-map

ibatis-customer-credit.xml

```
<resultMap id="result" class="egovframework.brte.sample.common.domain.trade.CustomerCredit">
    <result property="name" column="NAME" />
    <result property="credit" column="CREDIT" />
```

```

</resultMap>

<select id="getAllCustomerCredits" resultMap="result">
    select ID, NAME, CREDIT from CUSTOMER
</select>
...

```

Composition and Implementation of JunitTest

Composition of JunitTest

Implement the iBatisIo 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/ibatisIoJob.xml")
public class EgovIbatisFunctionalTests extends EgovAbstractIoSampleTests {

    @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 = getOutputs(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

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

- [iBATIS Configuration](#)
- [Data Access](#)
- <http://static.springsource.org/spring-batch/reference/html/readersAndWriters.html#IbatisPagingItemReader>