

Batch Execution Log Example

Outline

In execution of batch, you need to make sure the logs of batch-related information are stored in the registry and DB and officially verified.

Description

Settings

Configuring Jobs

Check out **logManagementJob.xml**, the job configuration file for Batch Execution Log example.

Configuring jobs for Batch Execution Log Example is nothing special. Refer to [Skip Example](#) for more information.

```
<job id="logManagementJob" incrementer="incrementer" xmlns="http://www.springframework.org/schema/batch">
    <step id="step1" parent="baseStep">
        <tasklet>
            <chunk reader="fileItemReader" processor="tradeProcessor" writer="tradeWriter"
commit-interval="3" skip-limit="10">
                <skippable-exception-classes>
                    <include
class="org.springframework.batch.item.file.FlatFileParseException" />
                    <include
class="org.springframework.batch.item.WriteFailedException" />
                </skippable-exception-classes>
            </chunk>
        </tasklet>
        <next on="COMPLETED WITH SKIPS" to="errorPrint" />
        <fail on="FAILED" exit-code="FAILED" />
    </step>
    <step id="errorPrint">
        <tasklet ref="errorLogTasklet" />
    </step>
</job>
```

Composition and Implementation of JunitTest

Composition of JunitTest

Work Junit Test that comprises logManagementJob configuration and relevant classes, where batch implementation is involved and the associated contents are viewed.

- ✓ See [Junit Test Description for Batch Execution Environment](#) for structure of the Class JunitTest.
- ✓ assertEquals(BatchStatus.COMPLETED, jobExecution.getStatus()) : Make sure you check the batch execution result is COMPLETED.
- ✓ assertEquals("[logManagementJob]", jobRegistry.getJobNames().toString()) : Check out JobName in Registry.
- ✓ assertEquals("3 records were skipped!", simpleJdbcTemplate.queryForObject("SELECT MESSAGE from ERROR_LOG where JOB_NAME = ?", String.class, "logManagementJob")) : Check out the ERROR_LOG table information stored in DB.

```
@ContextConfiguration(locations = {"/egovframework/batch/simple-job-launcher-context.xml",
"/egovframework/batch/jobs/logManagementJob.xml",
```

```

        "/egovframework/batch/job-runner-context.xml" })
public class EgovLogManagementFunctionalTests {
    ...
    @Test
    public void testUpdateCredit() throws Exception {
        JobExecution jobExecution = jobLauncherTestUtils.launchJob();
        assertEquals(BatchStatus.COMPLETED, jobExecution.getStatus());

        // Registry : Job information logs
        assertEquals("[logManagementJob]", jobRegistry.getJobNames().toString());

        // DB : Number of table logs in ERROR_LOG
        assertEquals(1, SimpleJdbcTestUtils.countRowsInTable(      simpleJdbcTemplate,
        "ERROR_LOG"));
        // DB : ERROR_LOG Log Data
        assertEquals("3 records were skipped!",simpleJdbcTemplate.queryForObject(
            "SELECT MESSAGE from ERROR_LOG where JOB_NAME =?",String.class,
        "logManagementJob"));
    }
}

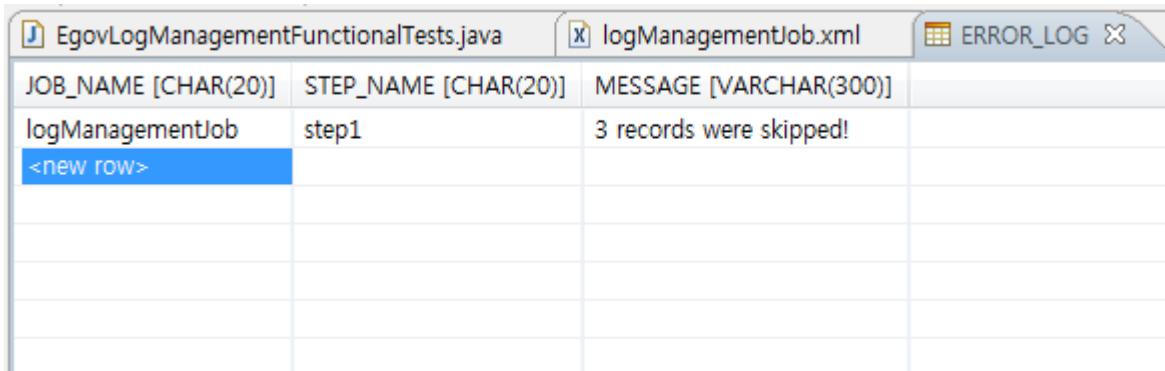
```

Implementation of JunitTest

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

Verify Result

You can check out data in the table ERROR_LOG of DB.



The screenshot shows a database interface with three tabs at the top: 'EgovLogManagementFunctionalTests.java', 'logManagementJob.xml', and 'ERROR_LOG'. The 'ERROR_LOG' tab is selected and displays a table with three columns: 'JOB_NAME [CHAR(20)]', 'STEP_NAME [CHAR(20)]', and 'MESSAGE [VARCHAR(300)]'. There is one row in the table:

JOB_NAME [CHAR(20)]	STEP_NAME [CHAR(20)]	MESSAGE [VARCHAR(300)]
logManagementJob	step1	3 records were skipped!

A blue button labeled '<new row>' is visible at the bottom left of the table area.

References

- [JobRegistry](#)
- [JobRepository](#)