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

Pre/Post Process Examples

Outline

When implementing the batch, the eGovFramework using the Class Processor of Core to implement how Pre/Post Process Examples are processed before and after Step and Chunk.

Description

Refer to the following examples EgovPreProcessorFunctionalTests and EgovPostProcessorFunctionalTests for Pre/Post Process examples. Note that the description assumes EgovPreProcessorFunctionalTests as Pre/Post Process examples involve the same configuration and description.

Settings

Configuring Jobs

Check preProcessorJob.xml, the Job configuration file for the pre-processing example.

You may wisely use <listener> to work Pre/Post Processing. Each SampleXXXProcessor class is registered as Beans for implementation under eGovFramework Processors. See Pre/Post Process Examples for more details.

✔ It is advised that <listeners> is configured within the tag (e.g. Job-related Listeners are to be used within the tag Job and the Listeners related to Step within the tag Step.)

```
<job id="preProcessorJob" xmlns="http://www.springframework.org/schema/batch">
	<listeners>
		<listener ref="jobPreListener" />
	</listeners>
	<step id="preProcessorStep1">
		<tasklet>
			<chunk reader="itemReader" writer="itemWriter" commit-interval="2">
				<listeners>
					<listener ref="chunkPreListener" />
				</listeners>
			</chunk>
		</tasklet>
		<listeners>
			<listener ref="stepPreListener" />
		</listeners>
	</step>
</job>

<bean id="jobPreListener" class="egovframework.brte.sample.example.listener.EgovSampleJobPreProcessor" />
<bean id="stepPreListener" class="egovframework.brte.sample.example.listener.EgovSampleStepPreProcessor" />
<bean id="chunkPreListener" class="egovframework.brte.sample.example.listener.EgovSampleChunkPreProcessor"
/>
```

Configuring Classes

In eGovFramework, you can use SampleProcessor inheriting Job, Step and Chunk to make use of the simple log. You can check out the point of listener implementation by checking the log output.

✔ For eGovFramework Processor, refer to the core's Pre Processor / Post Processor Management for more information.

- Method to be called before Job

```
public class EgovSampleJobPreProcessor extends EgovJobPreProcessor {
	protected Log log = LogFactory.getLog(this.getClass());
	public void beforeJob(JobExecution jobExecution) {
		log.info(">>>>>>>>> beforeJob :::    Start " + jobExecution.getJobInstance().getJobName());
	}
}
```

- Method to be called after Job

```
public class EgovSampleJobPostProcessor extends EgovJobPostProcessor {
	protected Log log = LogFactory.getLog(this.getClass());
	public void afterJob(JobExecution jobExecution) {
		log.info(">>>>>>>>> afterJob :::    Finish " + jobExecution.getJobInstance().getJobName());
	}
}
```

- Method to be called before Step

```
public class EgovSampleStepPreProcessor<T, S> extends EgovStepPreProcessor<T, S> {
	protected Log log = LogFactory.getLog(this.getClass());
	public void beforeStep(StepExecution stepExecution) {
		log.info(">>>>>>>>> beforeStep ::: start " + stepExecution.getStepName());
	}
}
```

- Method to be called after Step

```
public class EgovSampleStepPostProcessor<T, S> extends EgovStepPreProcessor<T, S> {
	protected Log log = LogFactory.getLog(this.getClass());
	public ExitStatus afterStep(StepExecution stepExecution) {
		log.info(">>>>>>>>> afterStep ::: finish " + stepExecution.getStepName());
		return null;
	}
}
```

- Method to be called before Chunk

```
public class EgovSampleChunkPreProcessor extends EgovChunkPreProcessor {
	protected Log log = LogFactory.getLog(this.getClass());
	public void beforeChunk() {
		log.info(">>>>>>>>> beforeChunk :::");
	}
}
```

- Method to be called after Chunk

```
public class EgovSampleChunkPostProcessor extends EgovChunkPostProcessor {
	protected Log log = LogFactory.getLog(this.getClass());
	public void afterChunk() {
		log.info(">>>>>>>>> afterChunk :::");
	}
}
```

Composition and Implementation of JunitTest

Composition of JunitTest

Work Junit Test that comprises preProcessorJob 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("COMPLETED", jobExecution.getExitStatus().getExitCode()) : Make sure you check the batch execution result is COMPLETED.
✔ In getUniqueJobParameters, you can submit the input and outpur resources required by batch of JobParameter.

```
@ContextConfiguration(locations = { "/egovframework/batch/jobs/preProcessorJob.xml" })
public class EgovPreProcessorFunctionalTests extends EgovAbstractIoSampleTests {

	@Test
	public void testUpdateCredit() throws Exception {
		JobExecution jobExecution = jobLauncherTestUtils.launchJob(getUniqueJobParameters());

		// Define success or failure based on the event output in Job, Step and Chunk of the console.
		assertEquals("COMPLETED", jobExecution.getExitStatus().getExitCode());
	}
	/*
	* Method to configure JobParameter
	*/
	@Override

	protected JobParameters getUniqueJobParameters() {
		return new JobParametersBuilder(super.getUniqueJobParameters())
			.addString("inputFile","/egovframework/data/input/delimited.csv")
			.addString("outputFile","file:./target/test-outputs/delimitedOutput.csv")
			.toJobParameters();
	}
}
```

Implementation of JunitTest

See Implementation of JunitTest for more information.

Verify Result

The test result reflects the result of batch implementation only. Refer to the log available in the console for Pre/Post Processing Success.

```
Console
<terminated> EgovPreProcessorFunctionalTests [JUnit] C:₩eGovFrame-2.0₩bin₩jdk1.5.0_22₩bin₩javaw.exe (2012. 9. 24. 오후 2:49:34)
14:49:40,676  INFO main audit:156 - 2. PreparedStatement.close() returned
14:49:40,676  INFO main audit:156 - 2. Connection.commit() returned
14:49:40,677  INFO main audit:156 - 2. Connection.setAutoCommit(true) returned
14:49:40,678  INFO main audit:156 - 2. Connection.isReadOnly() returned false
14:49:40,678  INFO main audit:156 - 2. Connection.isClosed() returned false
14:49:40,679  INFO main audit:156 - 2. Connection.getAutoCommit() returned true
14:49:40,679  INFO main audit:156 - 2. Connection.clearWarnings() returned
14:49:40,680  INFO main audit:156 - 2. Connection.getAutoCommit() returned true
14:49:40,680  INFO main EgovSampleJobPreProcessor:45 - >>>>>>>> beforeJob :::  Start preProcessorJob
14:49:40,682 DEBUG main SimpleFlow:135 - Resuming state=preProcessorJob.preProcessorStep1 with status=UNKNOWN
14:49:40,682 DEBUG main SimpleFlow:143 - Handling state=preProcessorJob.preProcessorStep1
14:49:40,683  INFO main audit:156 - 2. Connection.getAutoCommit() returned true
14:49:40,684  INFO main audit:156 - 2. Connection.getAutoCommit() returned true
```

References

- Listener