

전자정부 표준프레임워크 배치 실행환경 - 실습



실습 목차

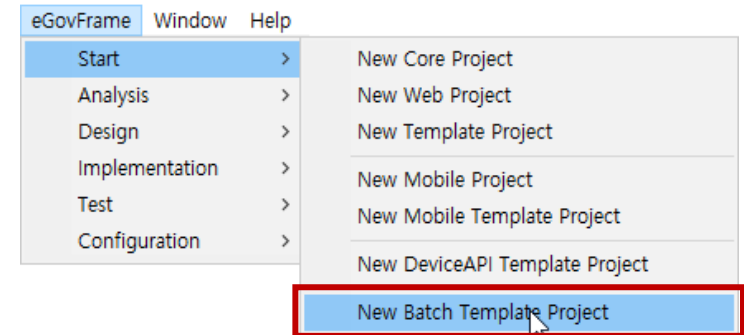
LAB 302-batch-job-test 배치템플릿 프로젝트 생성 및 테스트 실습

LAB 302-batch 1 Job 설정파일 구성

LAB 302-batch 2 File to DB 에서 File to File 으로 변경

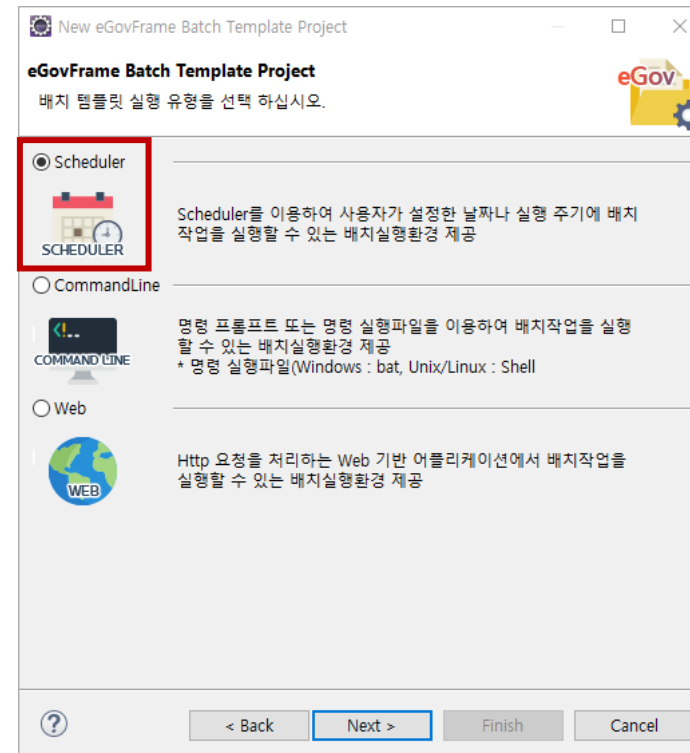
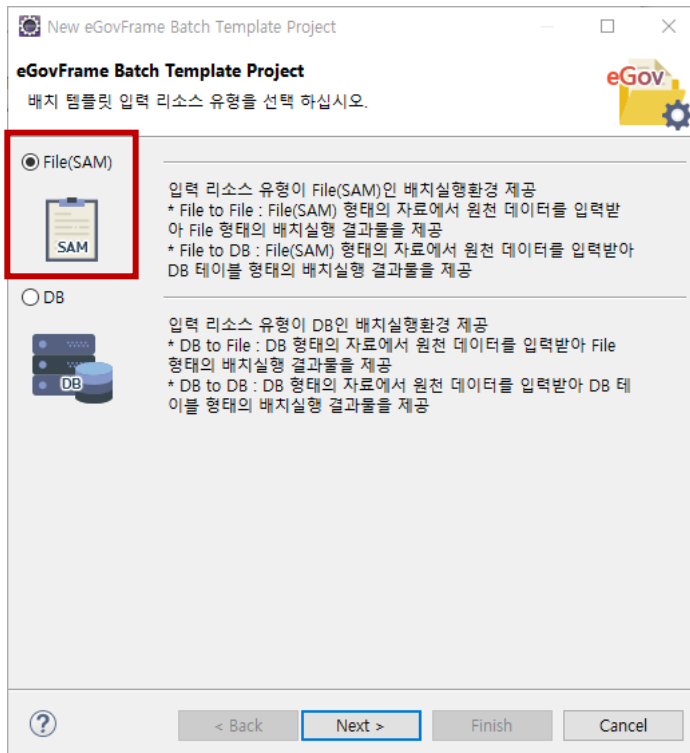
LAB 1-6 배치 템플릿 프로젝트 생성 및 테스트 실습(1/4)

Step 1-6-01. 구현도구에서 eGovFrame>Start>New Batch Template Project 메뉴를 선택합니다.



Step 1-6-02. eGovFrame Batch Template Project 위저드에서 File(SAM)을 선택하고 Next를 클릭합니다.

Step 1-6-03. eGovFrame Batch Template Project 위저드에서 Scheduler를 선택하고 Next를 클릭합니다.



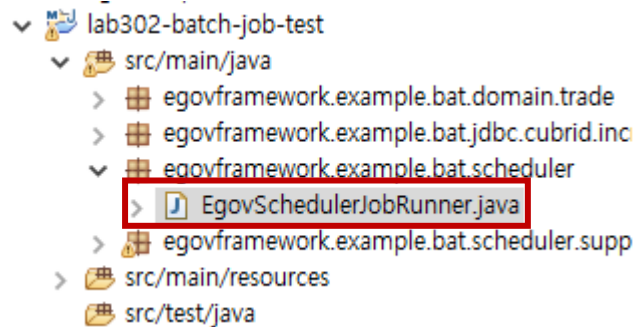
LAB 1-6 배치 템플릿 프로젝트 생성 및 테스트 실습(2/4)

Step 1-6-04. Project name과 Group Id, Artifact Id, Version을 입력하고 Finish를 클릭합니다.

항목	입력내용	비고
Project name	lab302-batch-job-test	수동입력
Group Id	egov.lab	수동입력
Artifact Id	lab302-batch-job-test	자동입력
Version	3.6.0	자동입력

Step 1-6-05. 생성된 lab302-batch-job-test 프로젝트의 생성 내역을 확인한다.

Step 1-6-06. 생성된 lab302 프로젝트의 EgovSchedulerJobRunner.java 클래스를 우클릭하여 Run As > Java Application을 실행한 후 배치 실행 결과를 콘솔에서 확인합니다

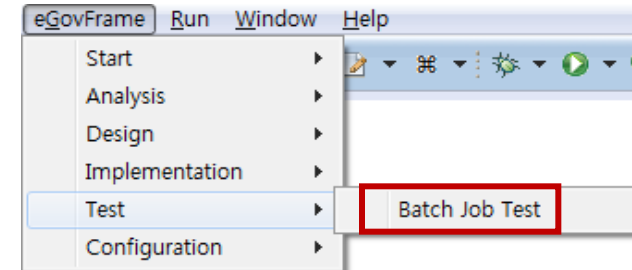


```

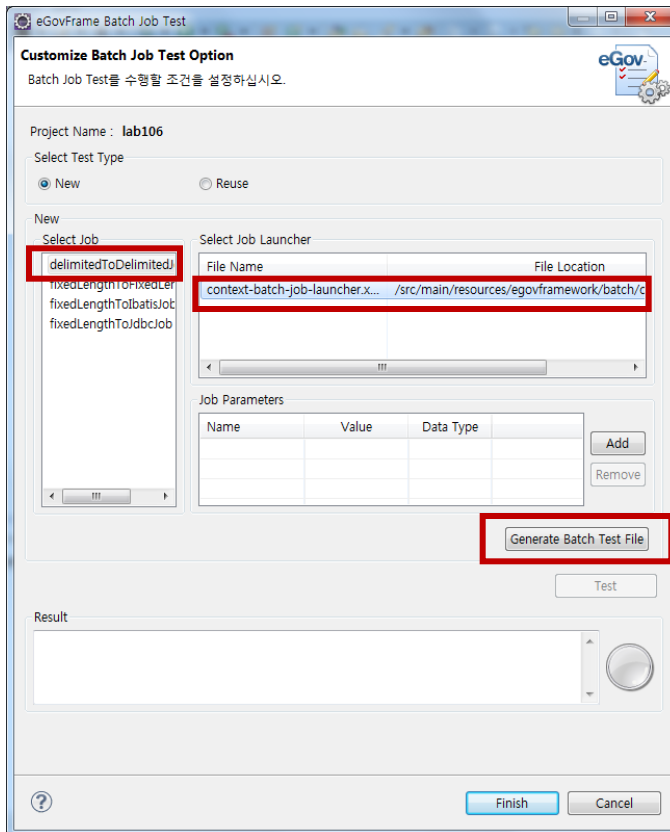
<terminated> EgovSchedulerJobRunner [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (
19:26:50,094 DEBUG [egovframework.example.bat.scheduler.support.EgovJobLauncherDetail]
19:26:50,150 INFO [org.springframework.batch.core.launch.support.SimpleJobLauncher]
19:26:50,162 INFO [org.springframework.batch.core.job.SimpleStepHandler] Executing s
19:26:50,223 INFO [org.springframework.batch.core.launch.support.SimpleJobLauncher]
19:27:00,002 WARN [egovframework.example.bat.scheduler.support.EgovJobLauncherDetail]
19:27:00,003 DEBUG [egovframework.example.bat.scheduler.support.EgovJobLauncherDetail]
19:27:00,003 DEBUG [egovframework.example.bat.scheduler.support.EgovJobLauncherDetail]
19:27:00,003 DEBUG [egovframework.example.bat.scheduler.support.EgovJobLauncherDetail]
    
```

LAB 1-6 배치 템플릿 프로젝트 생성 및 테스트 실습(3/4)

Step 1-6-07. 구현도구에서 eGovFrame>Test>Batch Job Test메뉴를 선택합니다.

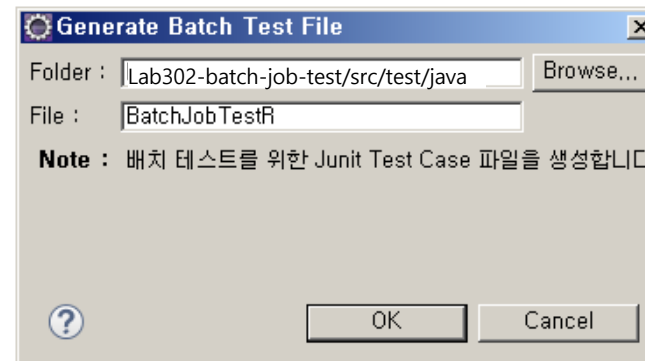


Step 1-6-08. eGovFrame Batch Job Test 위저드에서 Job 정보 (DelimitedToDelimitedJob)와 Job Launcher 정보를 선택한 후 Generate Batch Test File 버튼을 클릭합니다.



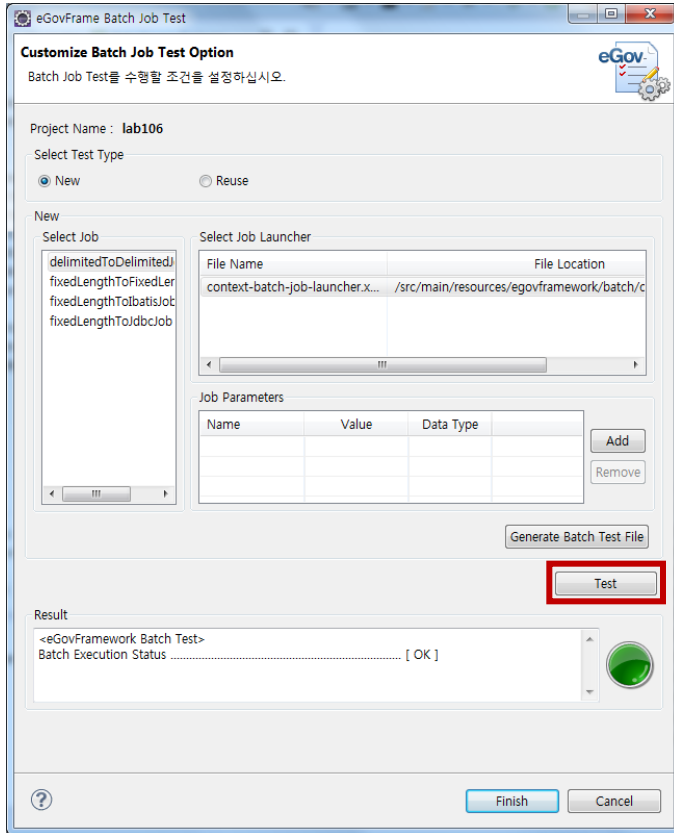
Step 1-6-09. TestCase 클래스 파일 생성을 위해 파일명을 입력하고 OK를 클릭합니다.

항목	입력내용
파일명	BatchJobTest



LAB 1-6 배치 템플릿 프로젝트 생성 및 테스트 실습(4/4)

Step 1-6-10. Test 버튼을 클릭하여 테스트케이스를 실행하여 실행 결과를 확인합니다.



Step 1-6-11. 테스트케이스 파일을 확인합니다.

```
BatchJobTest.java
1 *import static org.junit.Assert.assertEquals;
12 @RunWith(SpringJUnit4ClassRunner.class)
13
14 /**
15  * Test File Information
16  * Job:: /egovframework/batch/job/delimitedToDelimitedJob.xml
17  * Job Launcher:: /egovframework/batch/context-batch-job-launcher.xml
18  * job Parameters:: Date_Default Timestamp
19  */
20
21 @ContextConfiguration(locations = { "/egovframework/batch/context-batch-job-launcher.xml", "/egovframework/batch/j
22 public class BatchJobTest{
23
24     @Autowired
25     @Qualifier("eGovBatchRunner")
26     private EgovBatchRunner egovBatchRunner;
27
28     @Test
29     public void testJobRun() throws Exception {
30
31         String jobName = "delimitedToDelimitedJob";
32
33         JobParametersBuilder jobParametersBuilder = new JobParametersBuilder();
34         jobParametersBuilder.addLong("timestamp", new Date().getTime());
35     }
36 }
```

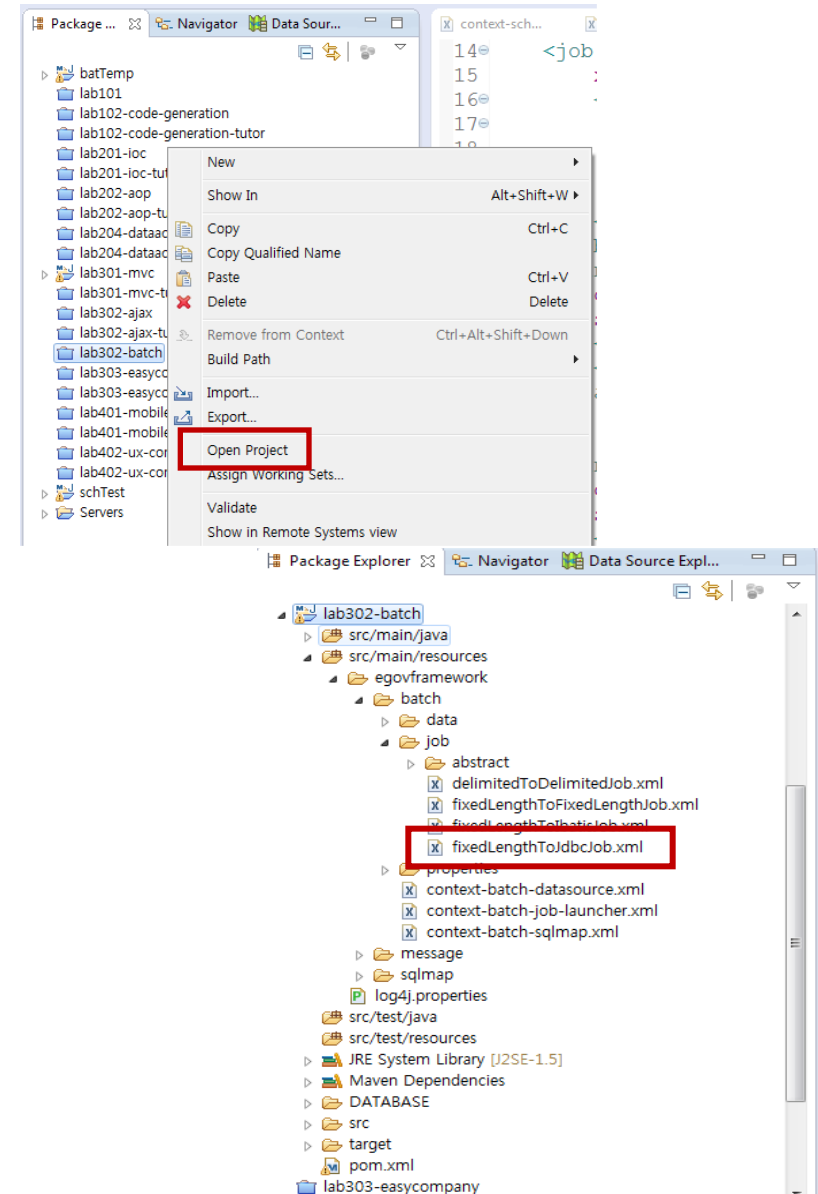
LAB 302-1 Job설정파일 구성 (1/3)

Step 302-1-01. 교육프로젝트에서 lab302-batch 마우스 오른쪽 버튼을 이용하여 프로젝트를 선택 후 메뉴에서 Open Project를 클릭한다.

Step 302-1-02. src/main/resources/egovframework/batch/job/ 하위의 fixedLengthToJdbcJob.xml을 연다.

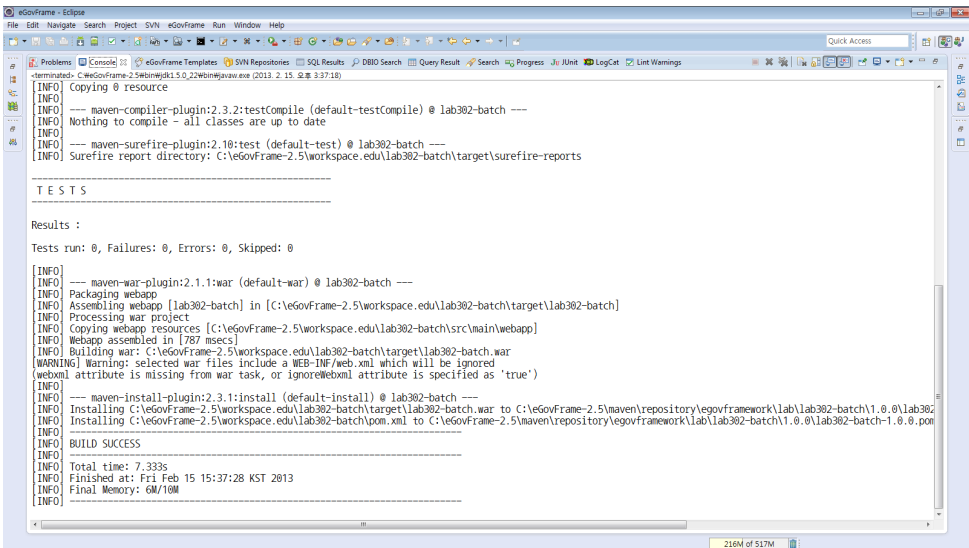
Step 302-1-03. job설정 파일 작성시 필요한 reader/process/writer를 fixedLengthToJdbcJob.xml파일에서 정의된 id를 찾아 입력한다.

```
<!--  
// TODO [Step 3-2-01]  
fixedLengthToJdbcJob 에서 사용하게되는 itemReader, itemProcessor, itemWriter들을  
찾아서 정의해 준다.  
-->  
<job id="fixedLengthToJdbcJob" parent="eGovBaseJob"  
xmlns="http://www.springframework.org/schema/batch">  
  <step id="fixedLengthToJdbcStep" parent="eGovBaseStep">  
  
    <tasklet>  
      <chunk reader=""  
        processor=""  
        writer=""  
        commit-interval="2" />  
    </tasklet>  
  </step>  
</job>
```



LAB 302-1 Job설정파일 구성 (2/3)

Step 302-1-04. 이클립스에서 Run As > Maven Install 을 클릭하여 Maven 을 실행한다.



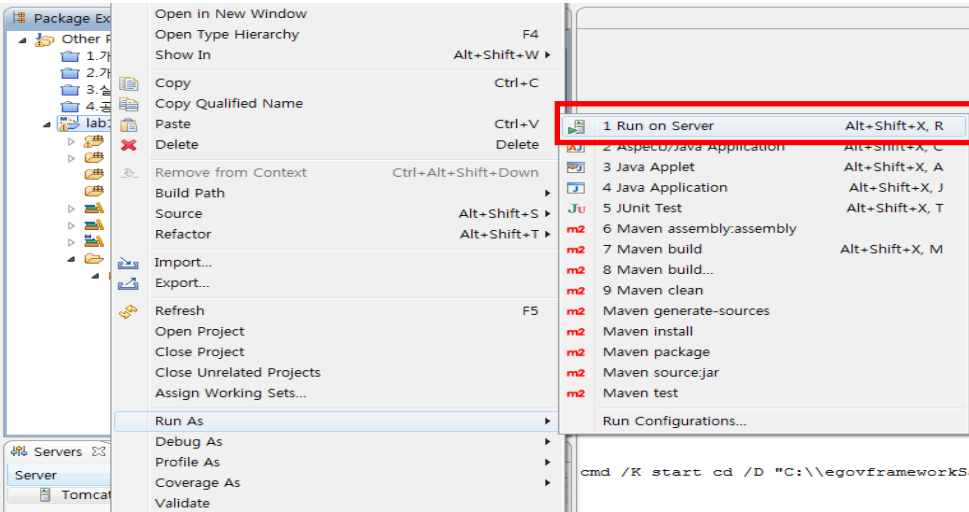
Step 302-1-05. 생성한 프로젝트를 실행하여 결과를 확인한다.

[Eclipse 이용]

lab302-batch > Run As > Run on Server 클릭 Finish 버튼 클릭

[오류발생시]

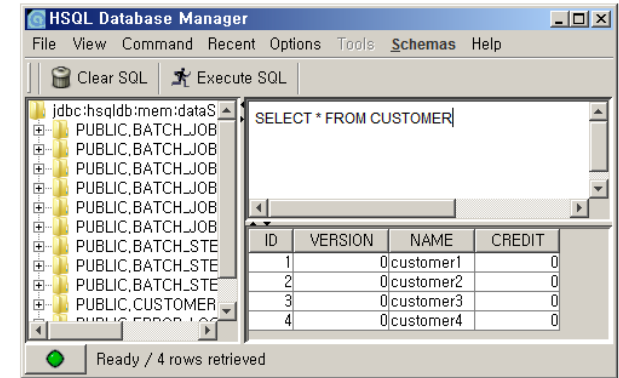
구현도구(eclipse) 재기동, Maven Clean , Maven Install
실행 후 다시 Run On Server 실행



LAB 302-1 Job설정파일 구성 (3/3)

Step 302-1-06. Step 302-1-05. 프로젝트를 톰캣 서버에 실행한 후, 배치를 실행하여 본다. (<http://localhost:8080/lab302-batch/>)

Step 302-1-07. HSql Database Manager에서 DML SQL(SELECT * FROM CUSTOMER)을 통해서 CUSTOMER 테이블의 초기 데이터를 확인한다.

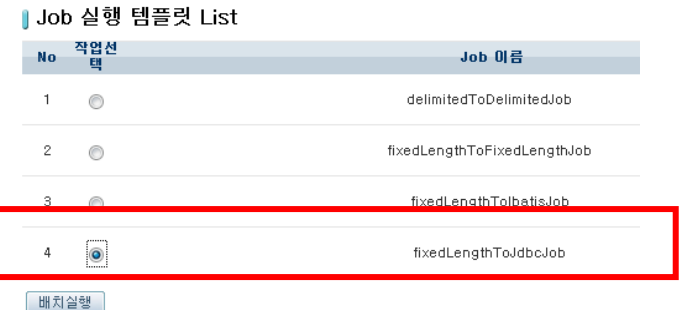


The screenshot shows the HSQL Database Manager interface. The left pane displays a tree view of the database structure, including 'PUBLIC.BATCH_JOB' and 'PUBLIC.CUSTOMER'. The right pane shows the SQL command 'SELECT * FROM CUSTOMER' and the resulting data table.

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

Ready / 4 rows retrieved

Step 302-1-07. Job 실행 템플릿 List 화면에서 fixedLengthToJdbcJob 작업을 선택 후 [배치실행]버튼을 클릭하여 실행한다.

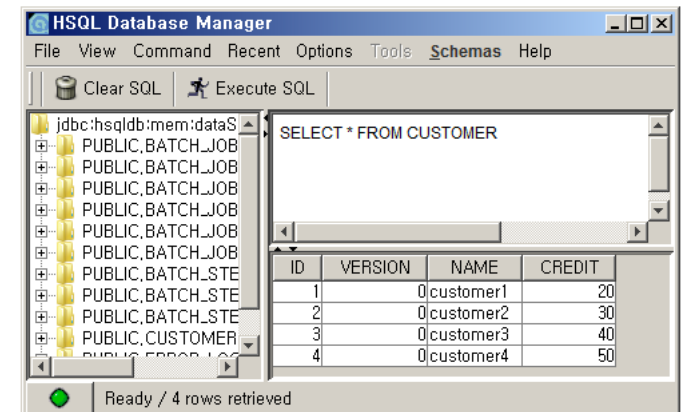


The screenshot shows the 'Job 실행 템플릿 List' screen. It contains a table with columns 'No', '작업선택' (selected), and 'Job 이름'. The row for 'fixedLengthToJdbcJob' is highlighted with a red box. Below the table is a '배치실행' (Batch Execute) button.

No	작업선택	Job 이름
1	<input type="radio"/>	delimitedToDelimitedJob
2	<input type="radio"/>	fixedLengthToFixedLengthJob
3	<input type="radio"/>	fixedLengthToFlatFileJob
4	<input checked="" type="radio"/>	fixedLengthToJdbcJob

배치실행

Step 302-1-08. HSql Database Manager에서 DML SQL(SELECT * FROM CUSTOMER)을 통해서 CUSTOMER 테이블의 변경 데이터를 확인한다.



The screenshot shows the HSQL Database Manager interface. The left pane displays the same tree view as before. The right pane shows the SQL command 'SELECT * FROM CUSTOMER' and the resulting data table, which now shows updated credit values.

ID	VERSION	NAME	CREDIT
1	0	customer1	20
2	0	customer2	30
3	0	customer3	40
4	0	customer4	50

Ready / 4 rows retrieved

LAB 302-2 File to DB방식에서 File to File으로 변경 (1/2)

Step 302-2-01. src/main/resources/egovframework/batch/job/ 하위의 fixedLengthToJdbcJob.xml을 연다.

Step 302-2-02. 파일을 읽어 DB에 입력하는 로직을 가지는 XML파일을 DB입력대신 File에 쓰는 기능으로 변경하기 위하여 bean에 매핑하는 클래스를 EgovJdbcBatchItemWriter 대신 FlatFileItemWriter로 변경한다.

Step 302-2-03. 파일을 만들 위치를 절대경로로 지정한다.

Step 302-2-04. FieldExtractor를 통해 VO의 field값에서 값을 꺼내와서 지정한 Format으로 변환하여 한 라인으로 합친 후에 File에 Write를 수행.

Step 302-2-05. Item의 필드 값들을 1 Line의 String으로 만들 때 필드값의 범위(고정길이) 지정.

```
<bean id="fixedLengthToJdbcJob.fixedLengthToJdbcStep.jdbcItemWriter"
      class="org.springframework.batch.item.file.FlatFileItemWriter" scope="step">
  <property name="resource" value="file:C:/eGovFrame-3.8.0/workspace.edu/lab302-
batch/src/main/resources/egovframework/batch/data/outputs/txtOutput.txt" />
  <property name="lineAggregator">
    <bean
      class="egovframework.brte.core.item.file.transform.EgovFixedLengthLineAggregator">
      <property name="fieldExtractor">
        <bean
          class="egovframework.brte.core.item.file.transform.EgovFieldExtractor">
          <property name="names" value="name,credit" />
        </bean>
      </property>
      <property name="fieldRanges" value="9,2" />
    </bean>
  </property>
</bean>
```

LAB 302-2 File to DB방식에서 File to File으로 변경 (2/2)

Step 302-2-06. 서버를 구동시켜 서버가 정상적으로 구동되는지 확인한다.

[오류발생시]

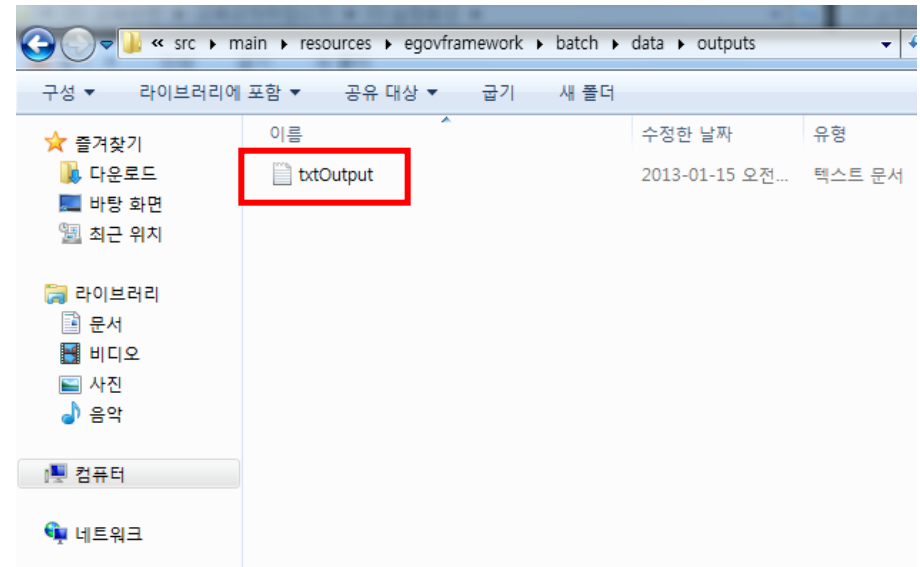
구현도구(eclipse) 재기동, Maven Clean , Maven Install

실행 후 다시 Run On Server 실행

Job 실행 템플릿 List

No	작업선택	Job 이름
1	<input type="radio"/>	delimitedToDelimitedJob
2	<input type="radio"/>	fixedLengthToFixedLengthJob
3	<input type="radio"/>	fixedLengthToIbatisJob
4	<input checked="" type="radio"/>	fixedLengthToJdbcJob

배치실행



수고 하셨습니다. ^^*