

# Glue Framework 5

교육 교재 v1.1



**POSCO DX**

# 목차


I. 준비하기	.....	3
II. 개요	.....	7
III. 환경 구성	.....	13
IV. 제작 가이드	.....	31
V. Support 안내	.....	241
VI. 별첨	.....	247

# 1. 준비하기





- URLs

구분	URL	다운로드 대상 및 용도
솔루션포털	<a href="http://www.solutionpot.co.kr">http://www.solutionpot.co.kr</a>	 문의 접수 (Q&A) , FAQ, 온라인 문서 , sample, GlueSDK, Glue plug-in, License key,
JDK	<a href="http://www.oracle.com">http://www.oracle.com</a>	Java SE 6 이상 JDK 설치 파일
IDE	<a href="http://eclipse.org">http://eclipse.org</a>	Luna 이상 Eclipse 설치 파일
WAS	<a href="http://tomcat.apache.org">http://tomcat.apache.org</a>	Tomcat 7.x 이상 설치 파일
Maven	<a href="http://maven.apache.org">http://maven.apache.org</a>	Maven 3.2 이상 (권장) 설치 파일
Log library	<a href="http://logback.qos.ch">http://logback.qos.ch</a>	logback-classic-{version}.jar logback-core-{version}.jar

- Java SE
  - 최신버전 이외 다운로드 시 회원가입 필요
  - 이클립스 버전에 따라 JDK 버전 제약 있음
  - <https://wiki.eclipse.org/Eclipse/Installation>
- Tomcat
  - <http://tomcat.apache.org/whichversion.html>
  - tomcat 8.0.x 는 JDK 1.7 이상 필요
- Maven
  - Maven 3.3+ 부터는 JDK 1.7 이상 필요

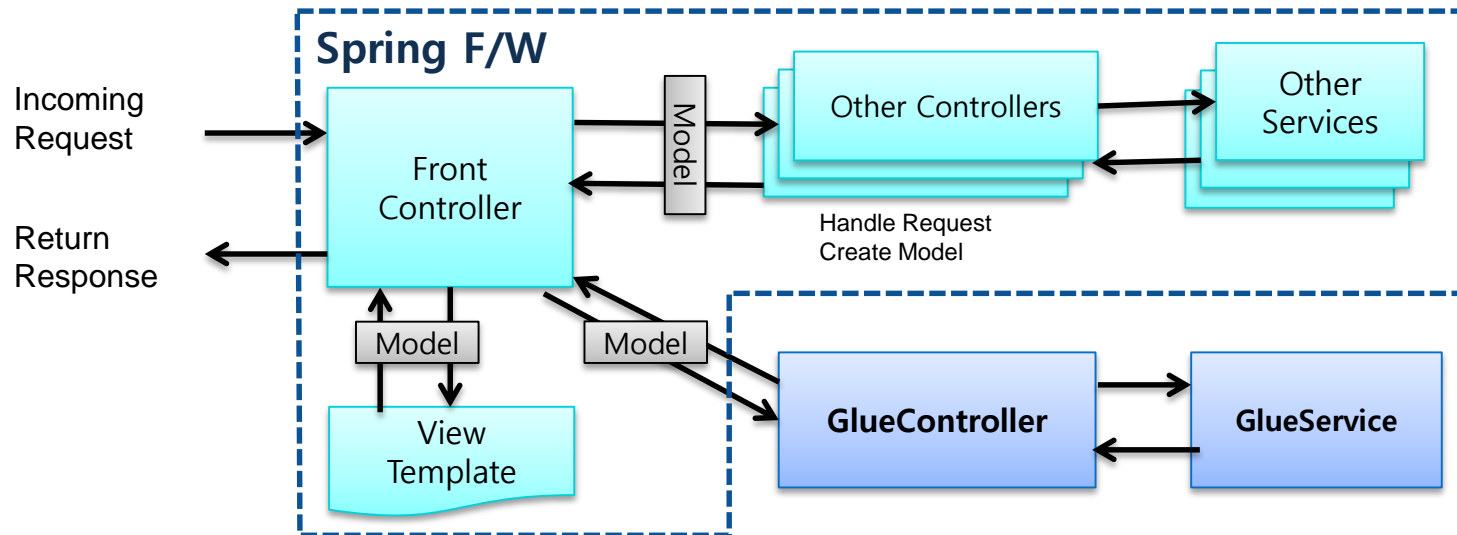
## II. 개요







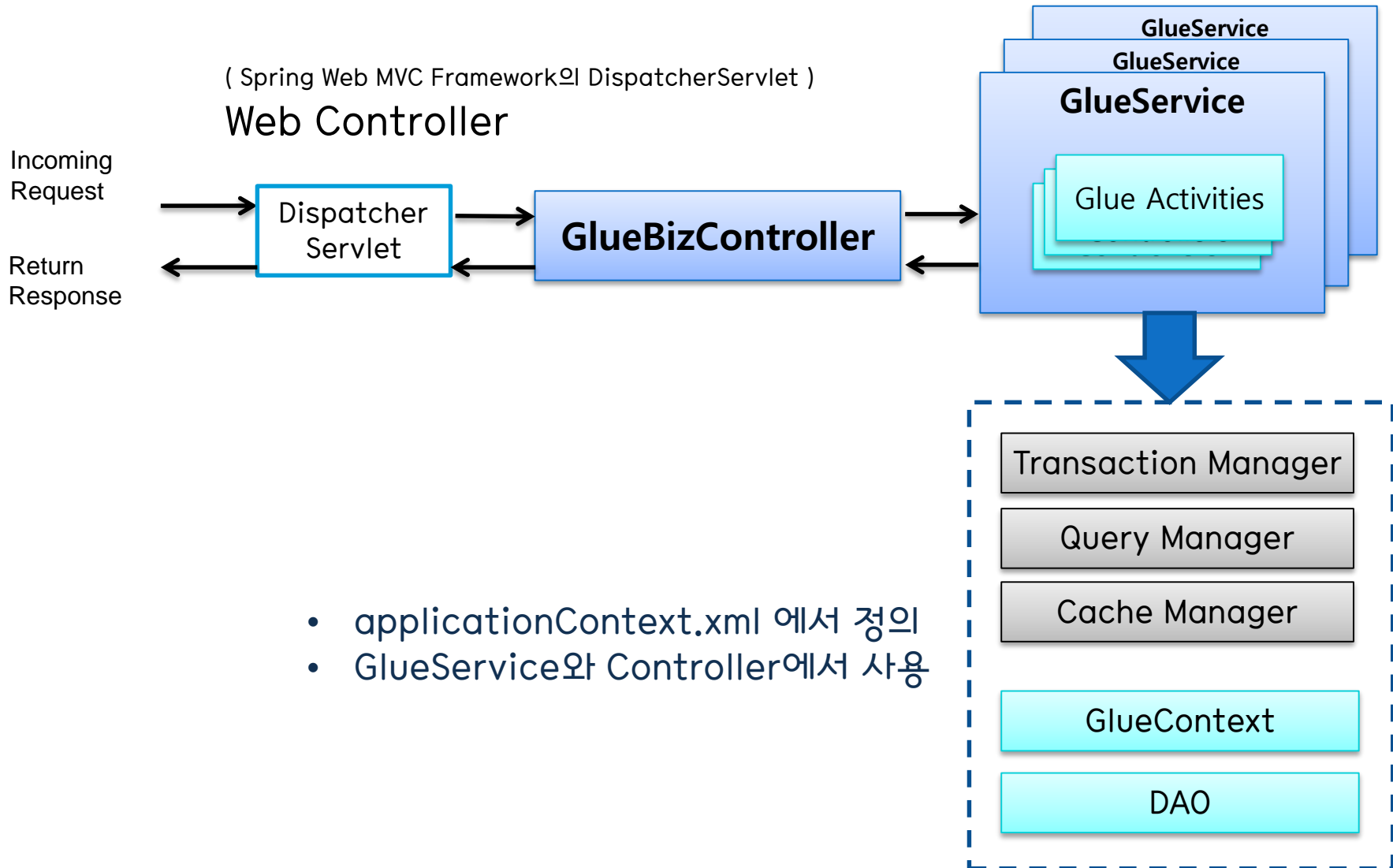
- Spring framework을 기반으로 **확장**한 MVC framework.
- Spring framework을 습득하지 않아도 쉽게 서비스를 만들 수 있는 솔루션.
- Any WAS, Any DB 에 적용 가능. (사전적 의미 : 접착제 )



- Spring F/W 위에 사용자가 만든 서비스(GlueService)를 기동시키는 모듈(GlueController)을 만들고, 서비스를 만들 수 있는 SDK(Software Development Kit) 및 자주 사용하는 기능(Glue Reusable Activity)를 미리 만들어 제공

구성 요소	포함 내용	용도
기본 개발 IDE	<ul style="list-style-type: none"><li>• Eclipse IDE for Java Developers<ul style="list-style-type: none"><li>• Juno(v4.2) 이후 버전 권장</li><li>• Neon 까지 지원</li></ul></li><li>• Eclipse IDE for Java EE Developers</li></ul>	<ul style="list-style-type: none"><li>• Java program 개발을 위한 기본 IDE로 eclipse 사용</li></ul>
Glue plug-in	<ul style="list-style-type: none"><li>• Glue Project generator</li><li>• Glue Activity Diagram Palette</li><li>• Glue Query Editor</li><li>• Glue Service generator</li></ul>	<ul style="list-style-type: none"><li>• Eclipse 상에서 Glue F/W을 쉽게 사용할 수 있는 plug-in</li></ul>
Glue SDK	<ul style="list-style-type: none"><li>• Glue Framework library</li><li>• Java Doc</li><li>• Sample Template</li><li>• Glue plug-in 에서 사용되는 설정 파일</li></ul>	<ul style="list-style-type: none"><li>• Application 개발에 필요한 기본 Kit.</li></ul>

- Glue Plug-in 2 는 LGPL license에 따라 소스 공개.
  - 별도 요청시 제공
- Glue Framework 4 이상부터 Glue plug-in 2 사용.
  - Glue Framework 3 이하는 Glue plug-in 1 사용.
  - IDE(Eclipse)별로 2개 major 버전의 Glue Plug-in 설치 불가.
- Glue plug-in 의 파일명 : name\_version.일자.jar
  - 예 : xxx\_2.0.0.v20130805.jar



### III. 환경 구성





- JAVA 버전 확인 (Java SE 6 이상 필요 )

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\yujin>java -version
java version "1.6.0_21"
Java(TM) SE Runtime Environment (build 1.6.0_21-b06)
Java HotSpot(TM) Client VM (build 17.0-b16, mixed mode, sharing)
```

- Glue Framework 5.x는 Java SE 6 ~ Java SE 8 까지 호환성 보장
- Eclipse 4.x 에서 Java SE 6 이상을 요구.
- Tomcat 7.x 에서 Java SE 6 이상을 요구.
- Eclipse 에서 사용할 JDK 설정 하기
  - eclipse.ini 에서 -vm 옵션으로 JVM 지정가능

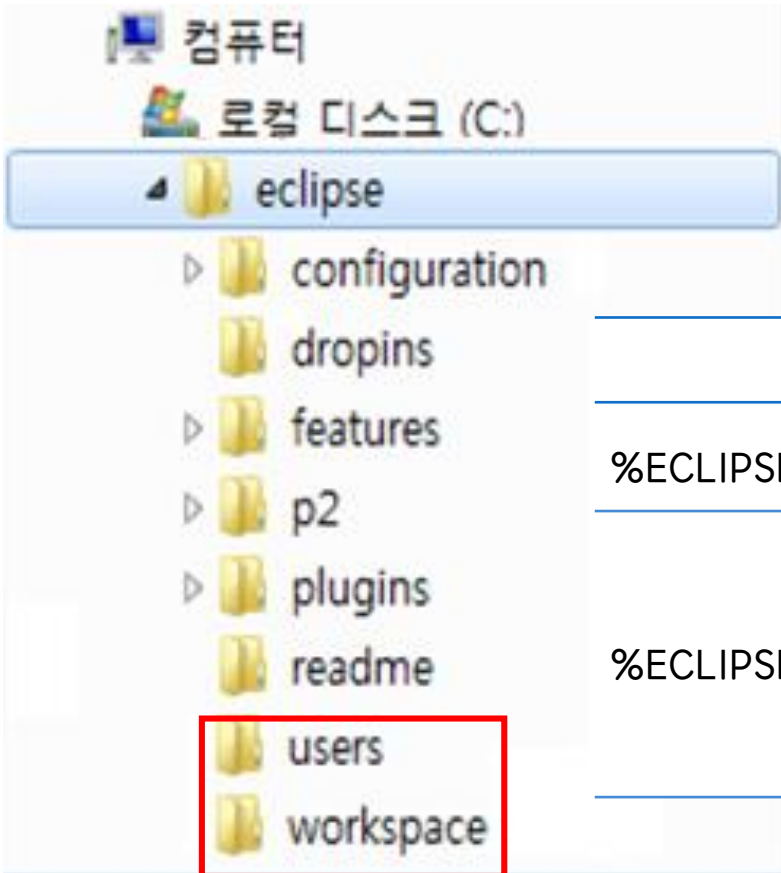
- Eclipse 설치
  - eclipse-java-xxxx-xxxxx.zip 압축해제
- 환경 파일 위치
  - ECLIPSE\_HOME : Eclipse 설치 위치라 가정
  - 파일 위치 : %ECLIPSE\_HOME% /eclipse.ini
- 환경 파일 수정
  - -vm 옵션으로 JVM 지정가능
  - -vmargs 위에 추가
  - 예)

**-vm**

```
C:\Java\jdk1.8.0_162\bin\javaw.exe
```

```
-vmargs
```





위치	용도
<code>%ECLIPSE_HOME%/Users</code>	GlueSDK 설치용
<code>%ECLIPSE_HOME%/workspace</code>	Glue Project 및 application 소스 저장용

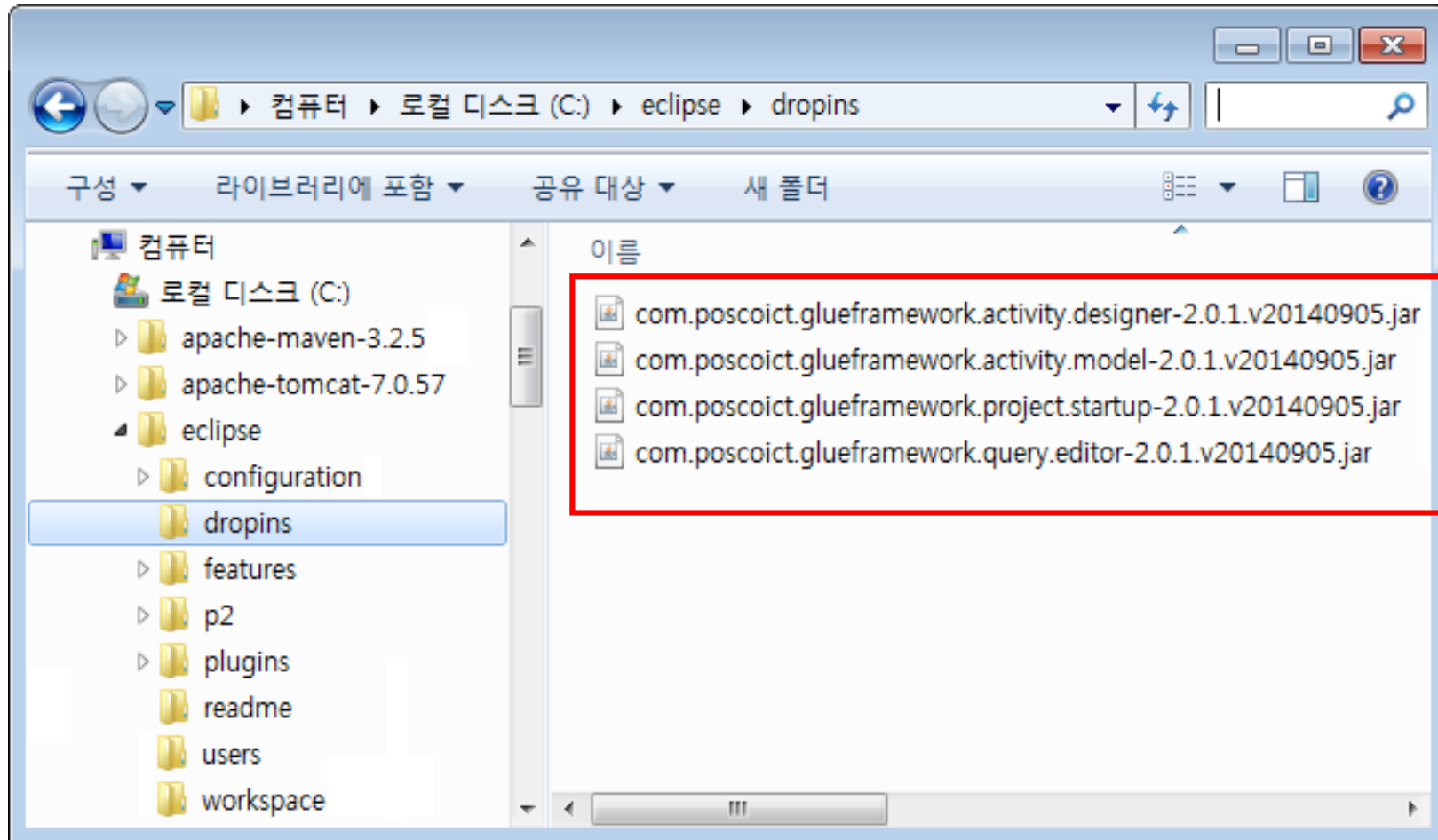
### C:\eclipse\eclipse.exe -clean

The image illustrates the steps to create a shortcut for Eclipse IDE:

- ① In the file explorer, right-click on `eclipse.exe` and select **바로 가기 만들기(S)** (Create Shortcut).
- ② The **Shortcut Properties** dialog box for `eclipse.exe` is shown. The **대상(T):** (Target) field is set to `C:\eclipse\eclipse.exe -clean`, which is highlighted with a red box and labeled ③.
- ④ The **확인** (OK) button is highlighted with a red circle, labeled ④.



- Glue plug-in 압축 풀기
- %ECLIPSE\_HOME%/dropins 에 plug-in jar 4개 복사



1. 실행 중인 Eclipse 종료

2. 기존 plugin 삭제

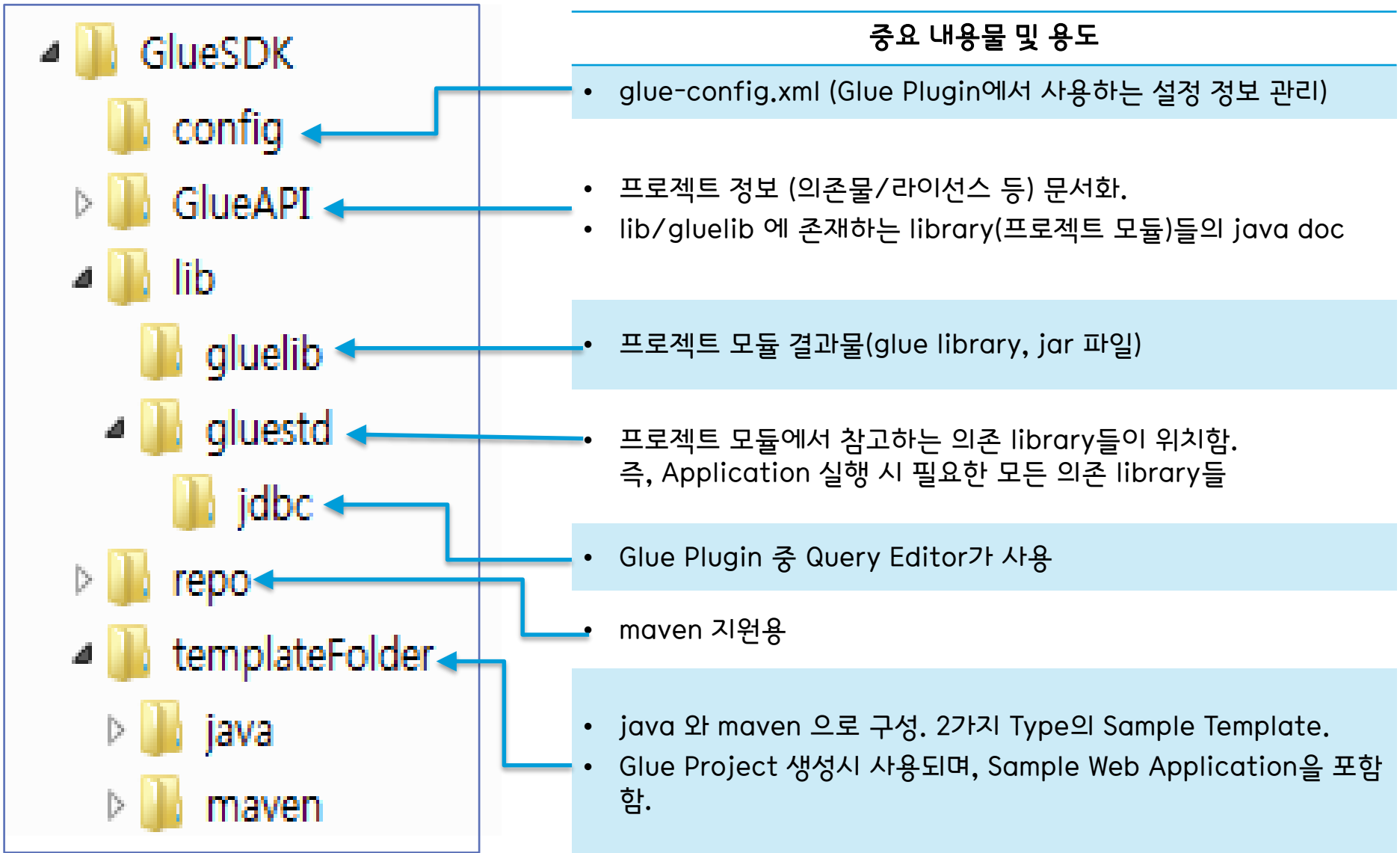
3. 새로운 plugin 복사

dropins 폴더에서

4. ‘**-clean**’이 포함된 바로 가기로 실행

5. **GlueSDK Location 재 설정**

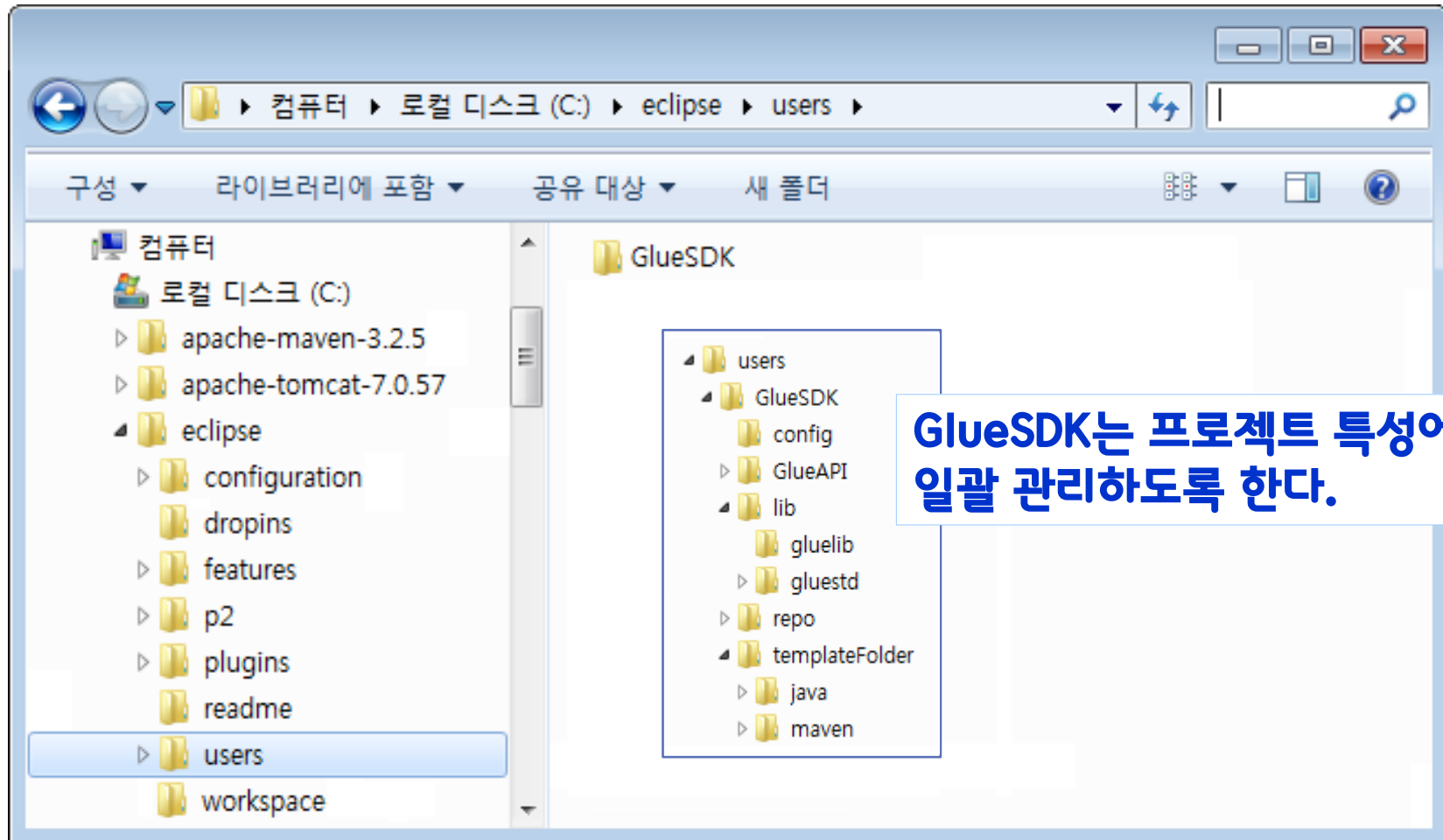
(Eclipse → Window → Preferences → Glue Framework)



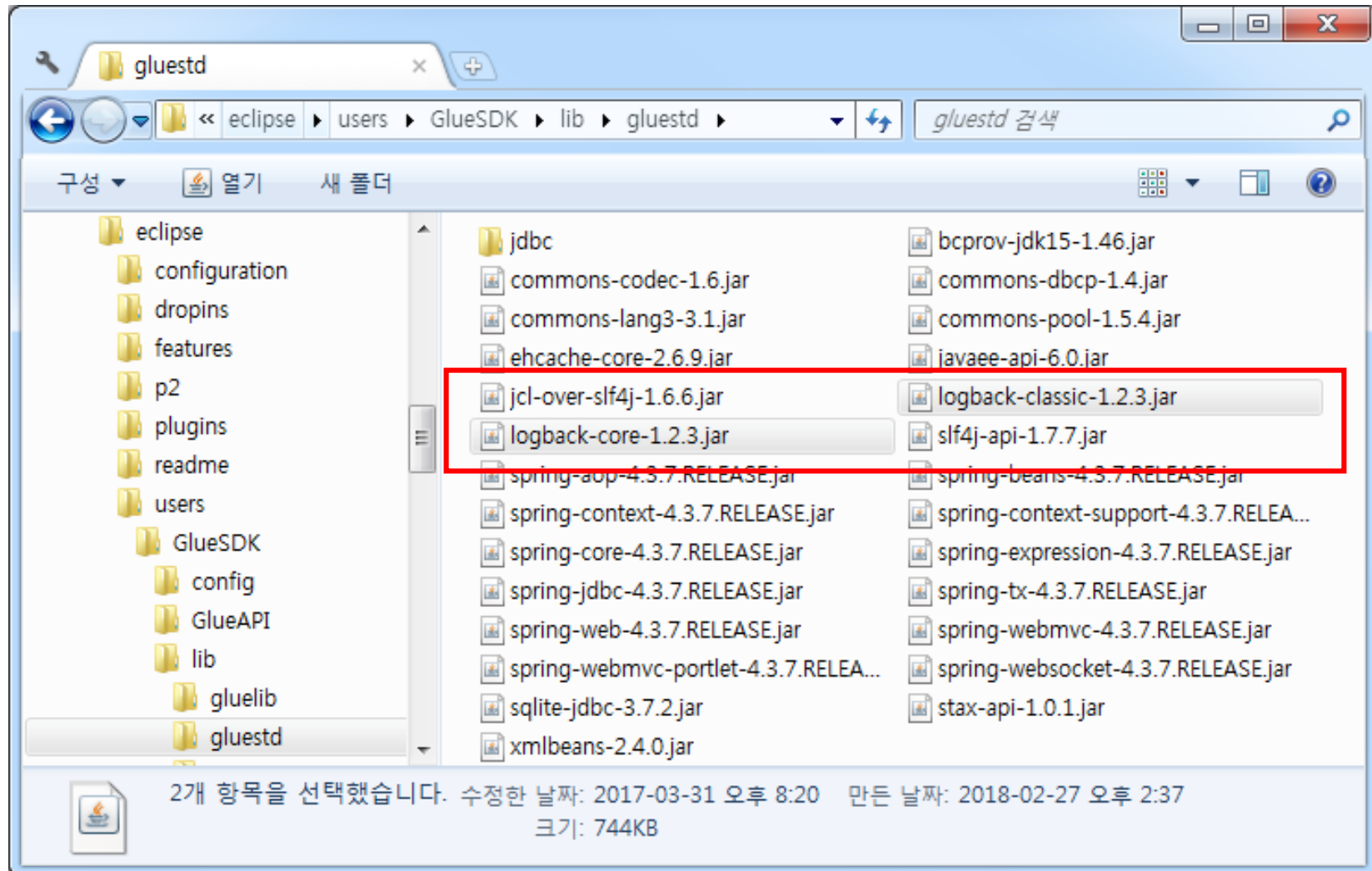


- GlueSDK의 구성물 중 상용 License, LGPL 를 따르는 것은 **직접 추가** 해야 함.  
(ex. ojdbc.jar , logback 등)
- GlueSDK 는 PL에 의해 **모든 개발자가 동일한 내용을 갖게** 하기를 권장.
- GlueSDK의 구성물 중 Glue Framework 5는 GPL 2를 따름.
  - <GlueSDK>/lib/gluelib
  - <GlueSDK>/repo
- 서버 환경 구성 시에는 **lib 폴더의 jar 파일을 활용함.**
- Glue Framework 패치/업그레이드 대상은 gluelib(repo)의 내용물임.
- Glue Plugin 은 Eclipse 에 설정된 GlueSDK의 위치를 사용함.
- Eclipse IDE의 workspace **별로 GlueSDK를 지정할 수 있음.**

- %ECLIPSE\_HOME%/Users 에 GlueSDK 의 압축을 풀고 복사

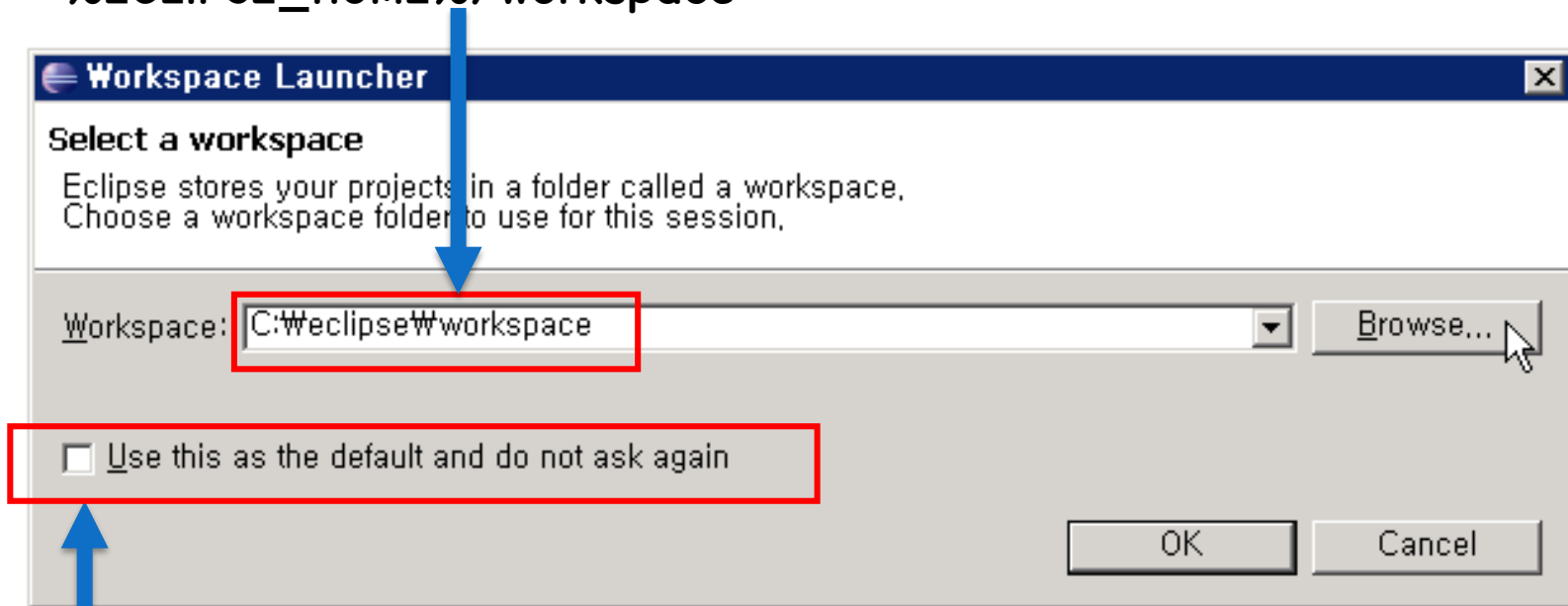


- Logback library를 GlueSDK/lib/gluestd 에 추가.



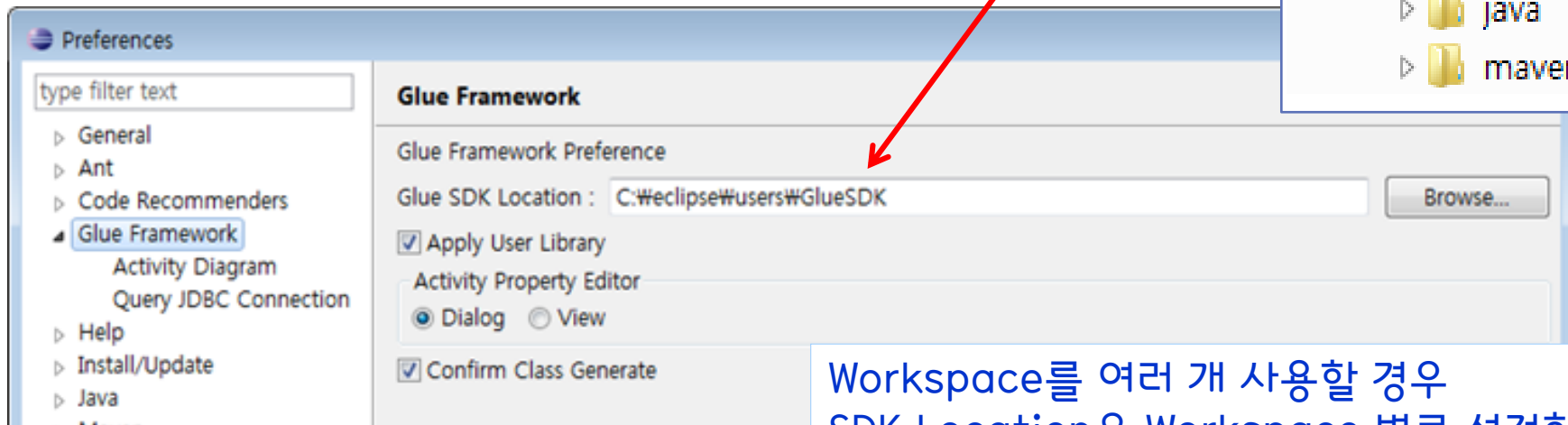
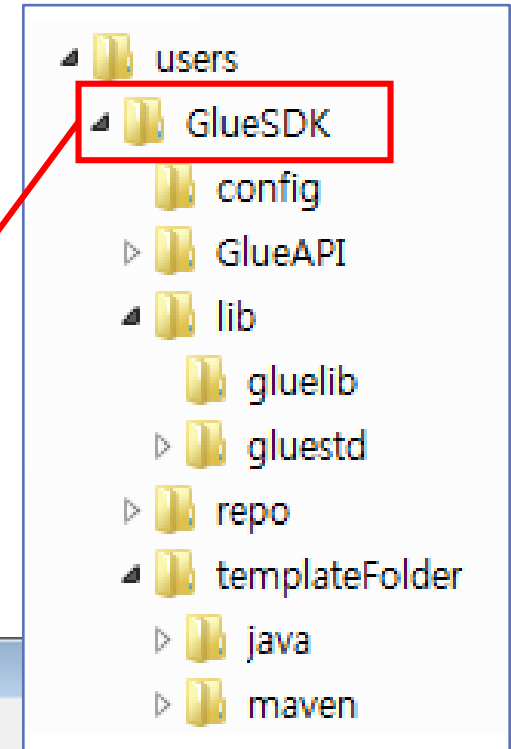


- Eclipse 실행 : -clean 추가한 바로가기 사용 (권장)
- Workspace 경로 지정
  - %ECLIPSE\_HOME%/workspace

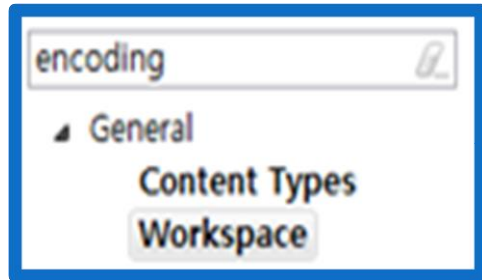


- 옵션 선택 시 최초 1회만 지정 필요 (권장)

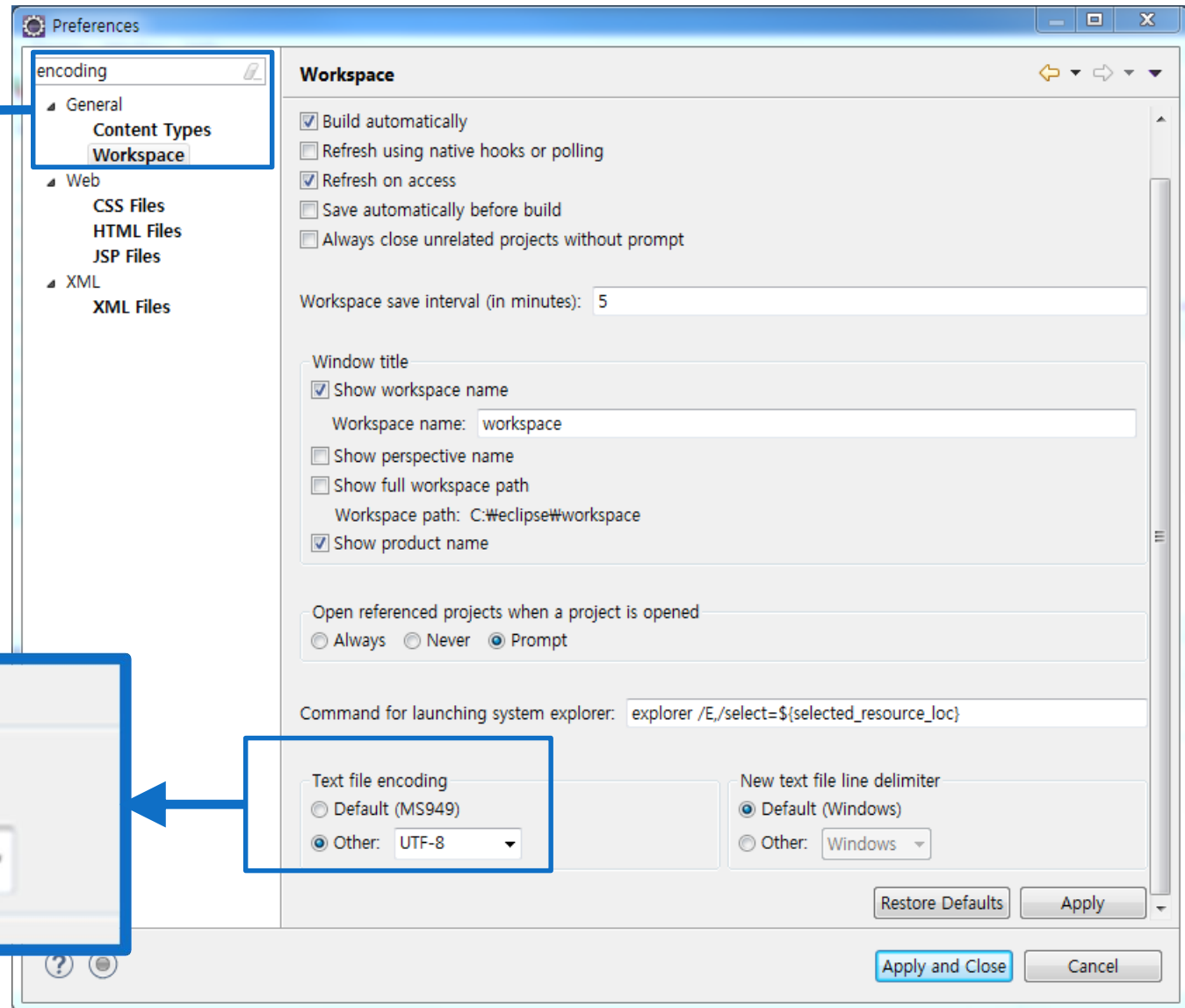
- Eclipse → Window → Preferences → Glue Framework
  - SDK Location 의 'Browse...' 클릭
  - `%ECLIPSE_HOME%\users\GlueSDK`



Workspace를 여러 개 사용할 경우  
SDK Location은 Workspace 별로 설정한다.

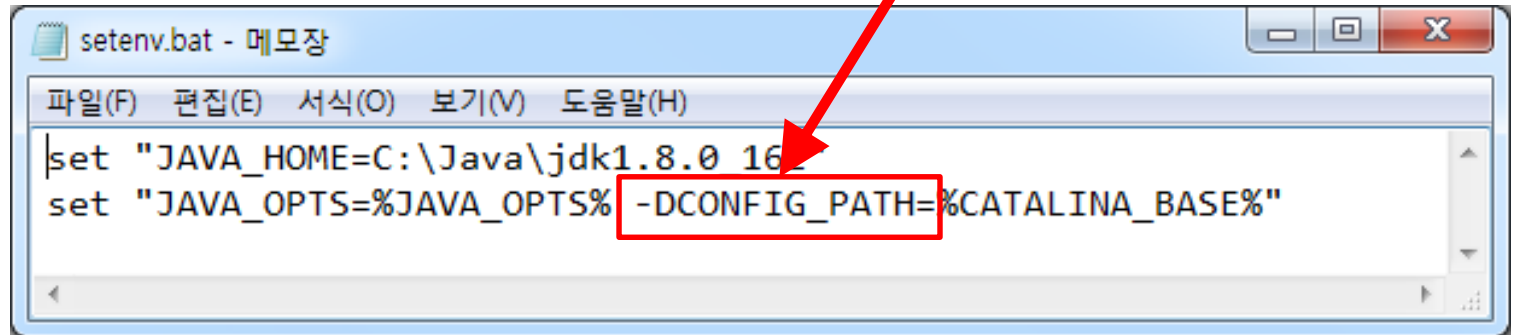


- Preference 에서 encoding으로 항목 검색
- UTF-8 으로 설정



- apache-tomcat-x.x.x.zip 압축 해제
- 환경변수 설정 파일 생성
  - CATALINA\_HOME : Tomcat 설치 위치
  - %CATALINA\_HOME%/bin/setenv.bat 생성
  - -D 옵션으로 CONFIG\_PATH 지정

Glue Framework 내부에서 사용하는 변수  
glue 라이선스 파일과  
glue.properties 파일의 저장 위치임.

A screenshot of a Windows Notepad window titled "setenv.bat - 메모장". The window contains two lines of text: "set "JAVA\_HOME=C:\Java\jdk1.8.0\_16..." and "set "JAVA\_OPTS=%JAVA\_OPTS% -DCONFIG\_PATH=%CATALINA\_BASE%". A red box highlights the "-DCONFIG\_PATH=%CATALINA\_BASE%" part, and a red arrow points from the "Point" callout above to this box.

```
setenv.bat - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
set "JAVA_HOME=C:\Java\jdk1.8.0_16..."
set "JAVA_OPTS=%JAVA_OPTS% -DCONFIG_PATH=%CATALINA_BASE%"
```

- 시스템 환경변수에 CATALINA\_HOME 변수 지정
- 여러 버전의 Tomcat 사용 시 startup.bat 에 CATALINA\_HOME 추가
  - set CATALINA\_HOME=C:\apache-tomcat-x.x.x

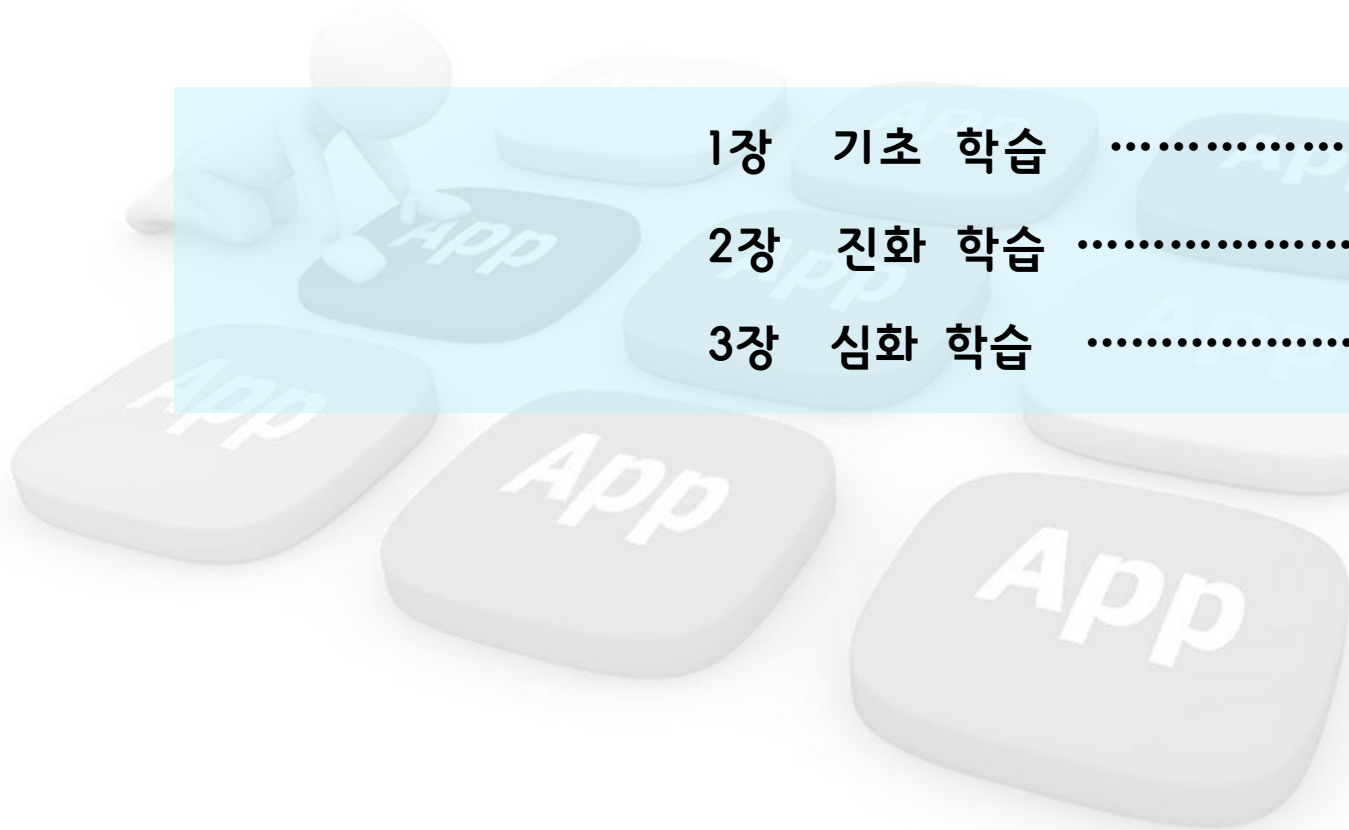
- Tomcat 실행 : %CATALINA\_HOME%/bin/startup.bat
- Browser 접속 확인 <http://127.0.0.1:8080>
- 확인 page 생성 : %CATALINA\_HOME%/webapps/ROOT/check.jsp
- Browser 에서 페이지 확인 <http://127.0.0.1:8080/check.jsp>

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
<form method=post>
<input type=text name="key" value="CONFIG_PATH">
<input type=submit value="check">
</form><%
    if(request.getParameter("key") !=null) {
        out.println(System.getProperty(request.getParameter("key")));
    }
%>
<hr>
</body>
</html>
```



## IV. Glue Framework 제작 가이드

1장	기초 학습	.....	32
2장	진화 학습	.....	63
3장	심화 학습	.....	121



# IV. Glue Framework 제작 가이드

## 1장 기초 학습





## 목표

- Glue Project의 기능을 이해한다.
- Glue Project 를 만들 수 있다.

## 실습

- WAS 없이 실행해 볼 수 있는 샘플 제작
- Glue Java Project 유형으로 간단한 Project 생성
- Project name : sample

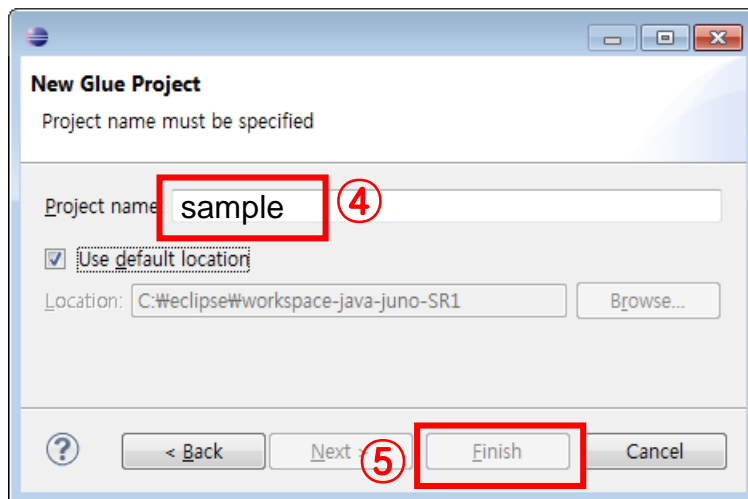
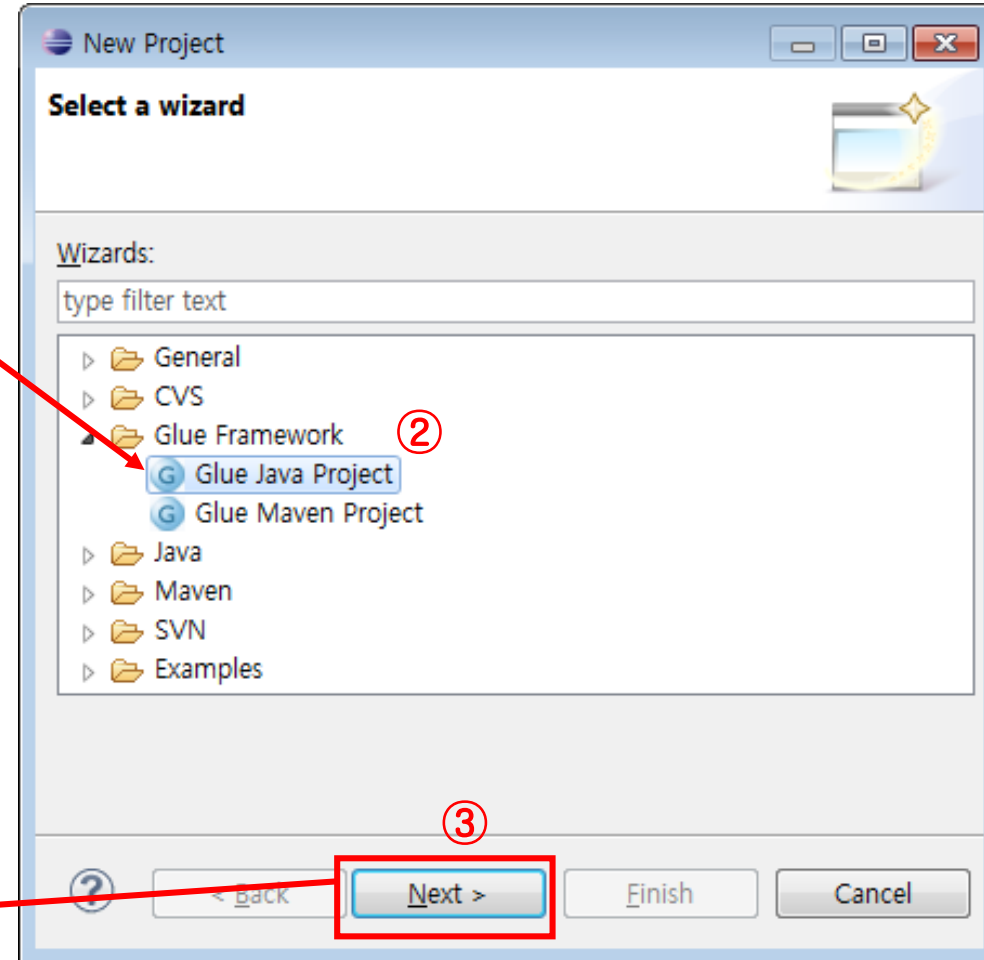
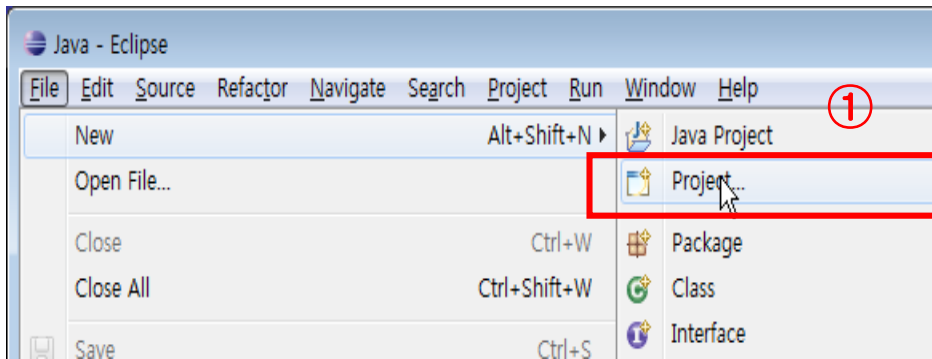
- AD(Activity Diagram)를 그릴 수 있다.
- GlueService를 만들 수 있다.
- Test를 위한 class를 만들 수 있다.

- hello.glue\_uml\_ad 그리기
- sample.HelloActivity class 만들기
- hello-service.xml 만들기
- test.HelloTest class 만들기

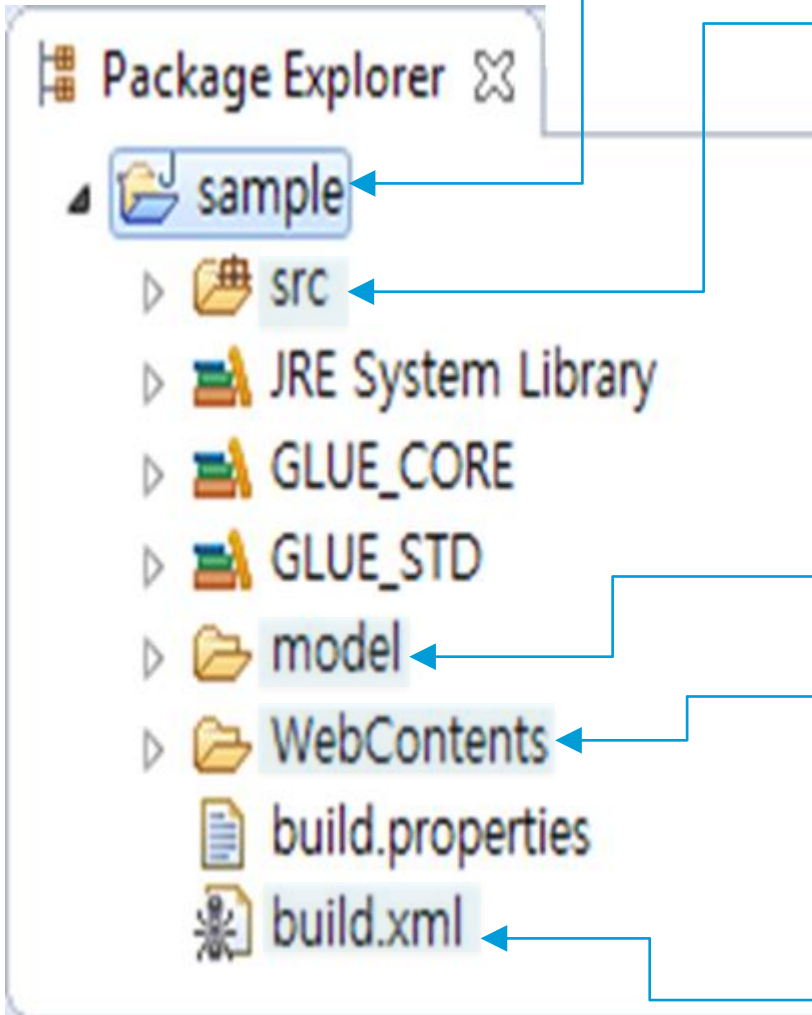
- GlueContext 의 용도를 이해한다.
- Glue Framework의 동작 원리를 이해할 수 있다.

- Glue Project란
  - Glue Framework으로 개발하는 환경을 만들어 주는 Eclipse project
- 주요 기능
  - Glue F/W으로 개발에 필요한 Lib Path 설정
    - Glue F/W의 환경 파일 자동 생성
      - applicationContext.xml, logback-test.xml 등
  - 자동 Build 환경 구성
    - build.xml으로 자동 Build 환경 구성
    - pom.xml 이용 maven 환경 구성
  - Sample Application 예제 제공

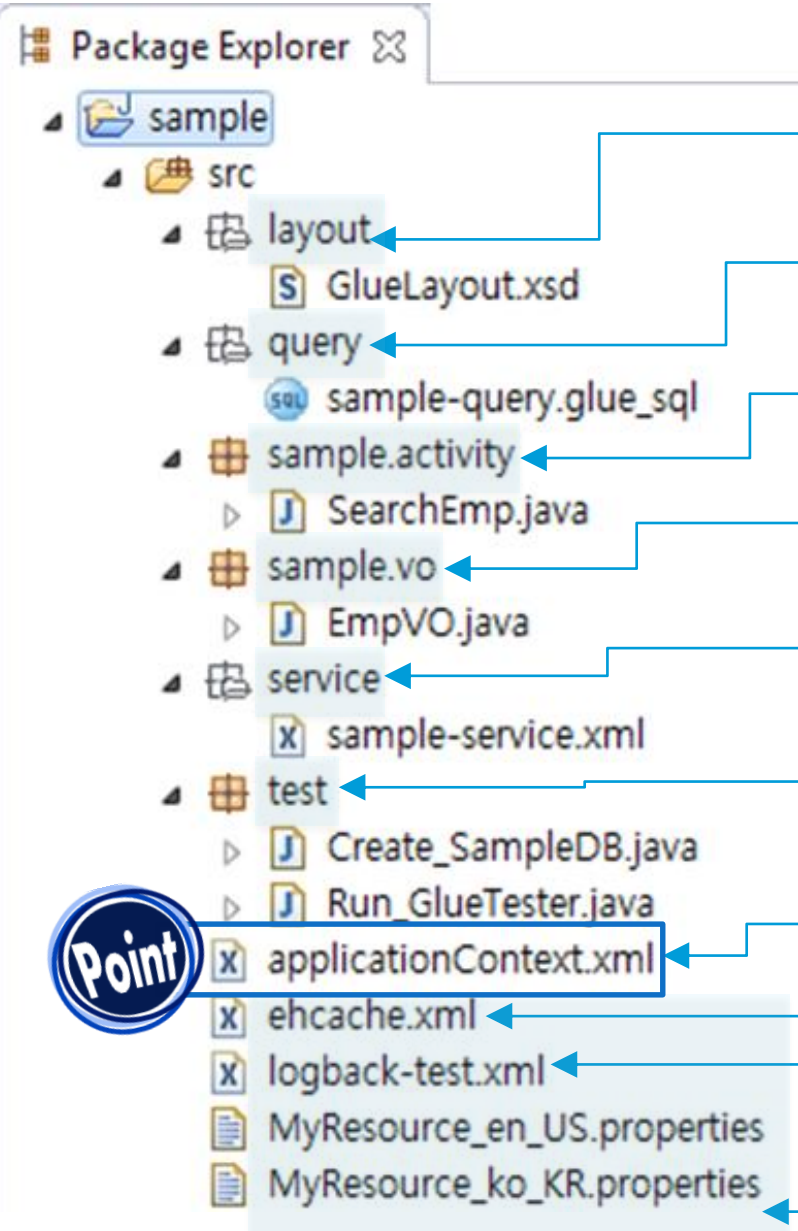
- Eclipse에서 File > New > Project 선택
- Wizard에서 Glue Framework > Glue Project 선택 > Next 버튼 클릭
- Project name – sample 입력
- Finish



## 중요 내용물 및 용도



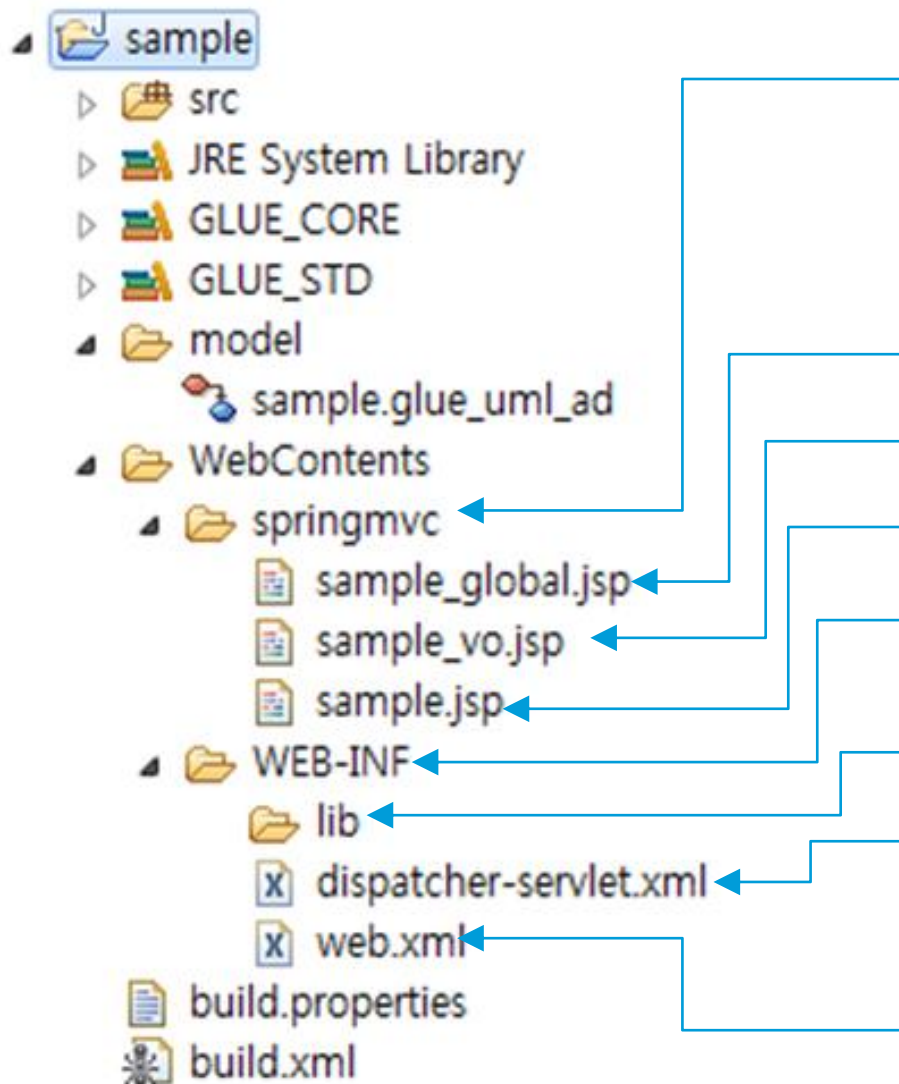
- Java Type 의 Project 명
- 소스 폴더
  - Bean 정의 파일 : applicationContext.xml
  - 캐시설정파일 : ehcache.xml
  - 로그 설정파일 : logback-test.xml
  - property 파일 : \*.properties
  - Glue Query 파일 : query 폴더의 \*-query.glue\_sql
  - Glue Service 파일 : service 폴더의 \*-service.xml
  - Java 파일(Activity, VO등)
  - GlueTester : test package 를 사용함.
- 설계 산출물 관리
  - Glue Activity Diagram 파일 : \*.glue\_uml\_ad
- Web 소스 폴더로 다음과 같은 파일이 위치한다.
  - web.xml
  - dispatcher-servlet.xml
  - JSP 파일 : \*.jsp
  - js, css, html, gif, jpeg 등의 파일
- Java 기반 Build Tool. War 생성 task 포함.



## 중요 내용물 및 용도

- msg.xml 파일의 위치.
- Layout 설계 산출물 XML
- query.glue\_sql(또는 -hquery.glue\_sql) 파일의 위치.
- SQL Query 설계 산출물 XML
- Custom Activity 예제.
- 개발 시 참조하여 개발할 Sample
- VO 예제
- 개발 시 참조하여 개발할 Sample
- service.xml 파일의 위치.
- BusinessLogic 설계산출물 XML
- GlueTester 실행 소스.
- Local에서 개발 내용 Test 지원 Tool
- 사용 컴포넌트(bean) 정의 .
- DB URL, Transaction Manager, DAO, ServiceManager, CacheManager, LayoutManager 등 정의
- Memory Cache 정의
- Logging 정의, Path 변경 필요
- Global 지원 또는 자체 Property 파일

## Package Explorer



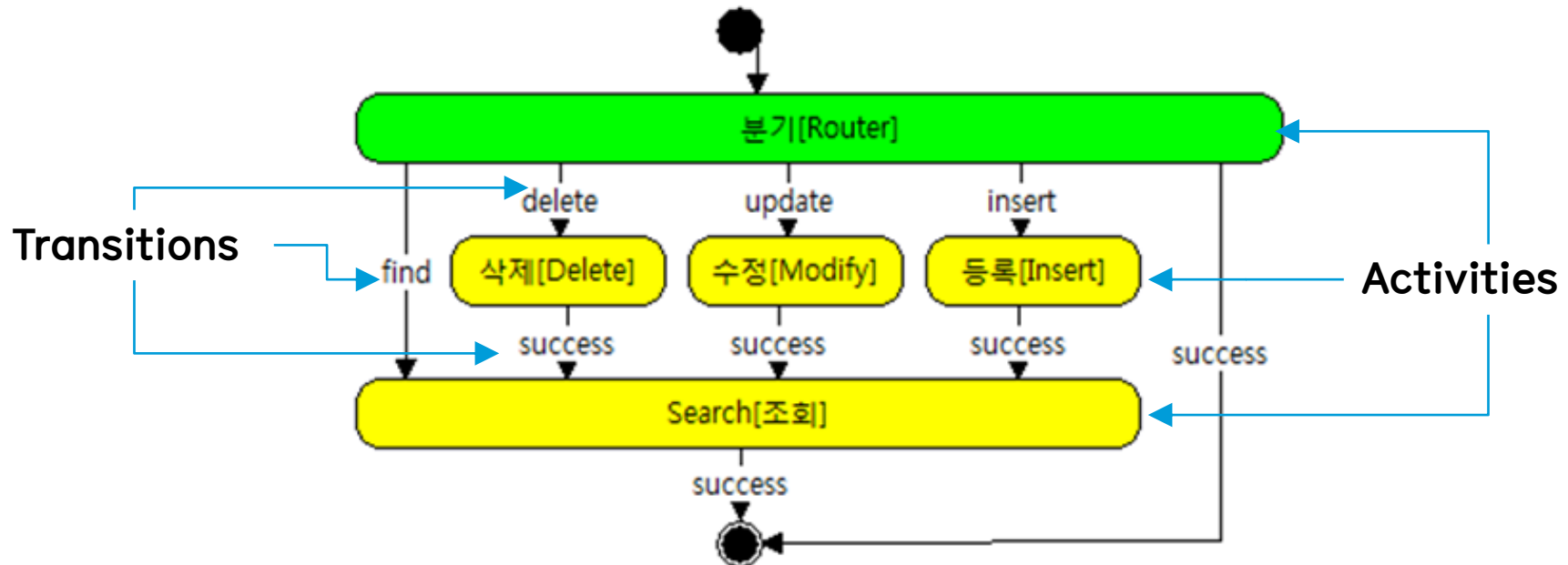
## 중요 내용물 및 용도

- JSP 파일의 위치.
- JSP 파일(HTML, JS, CSS 등 포함)은 WEB-INF를 제외한 WebContents 나 WebContents 하위 디렉토리에 위치함.
- 다국어 적용 예제
- VO 사용 화면 예제
- 기본 화면 예제
- Web Application의 필수 폴더.
- Web Application을 위한 설정 파일이 위치함.
- Library 폴더
- SpringMVC 의 DispatcherServlet이 필요로 하는 파일로 Spring Web MVC에 특화된 Component(bean)들을 포함함.
- Web Application의 필수 파일.
- servlet, filter, listener 등 정의

# Activity Diagram

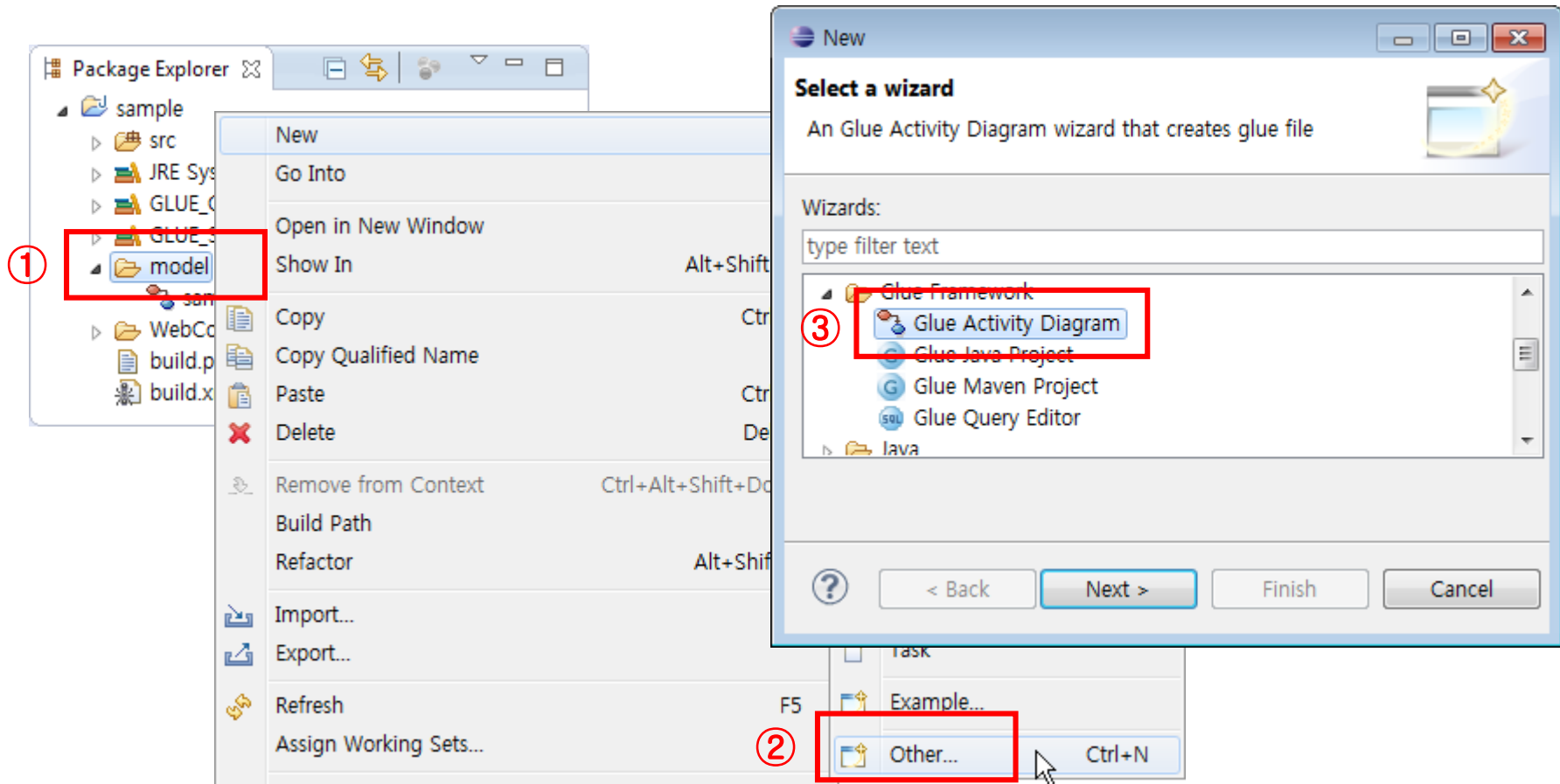


- Activity Diagram (AD)
  - 업무에 대한 흐름을 알기 쉽도록 표현하기 위한 Diagram.
  - 각 단위 업무(Activity) 와 흐름(transition)의 모음.



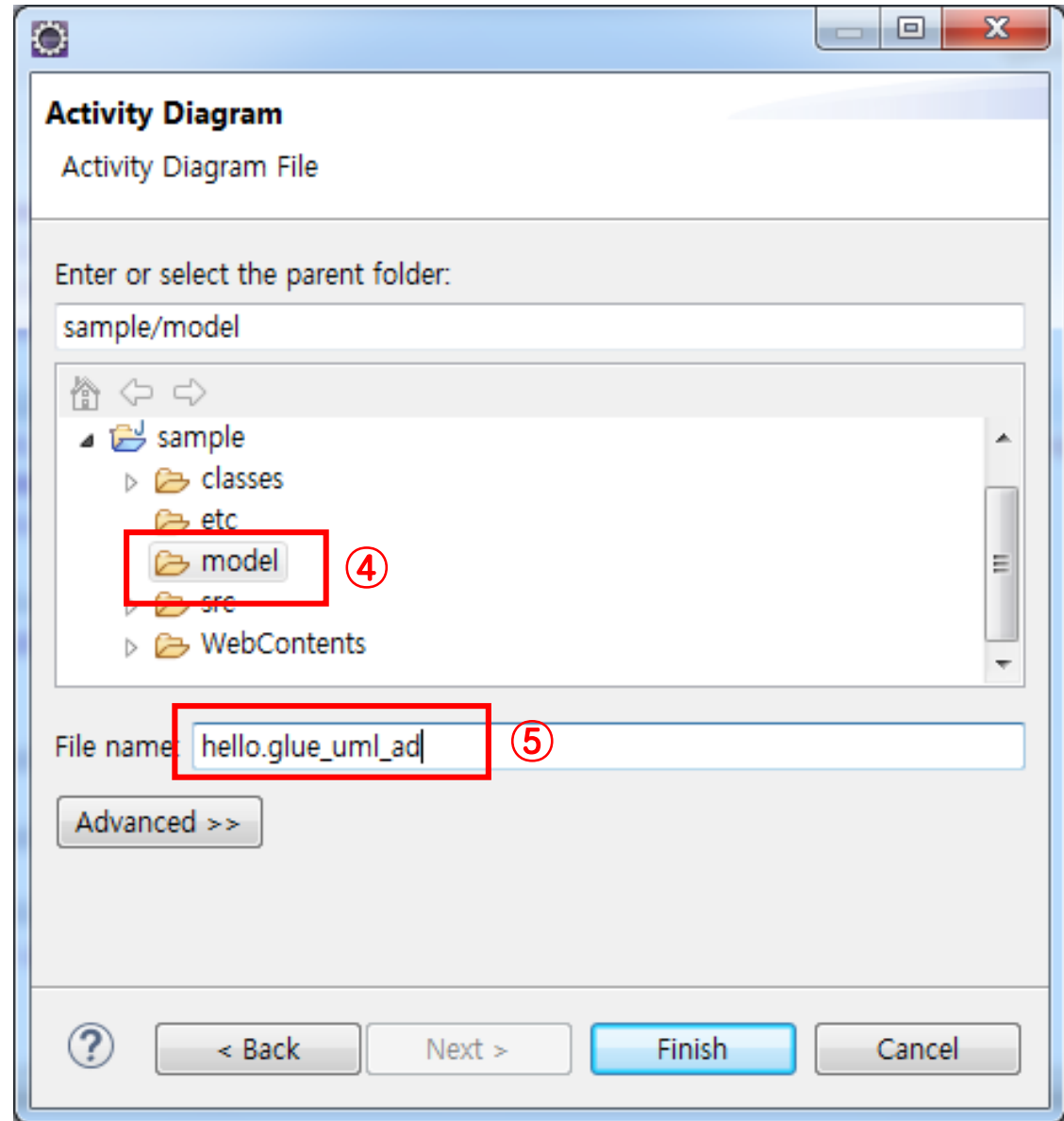
- 설계에 해당하며 , AD로부터 Service를 생성할 수 있다.
- 확장자는 glue\_uml\_ad. [sample01.glue\\_uml\\_ad](#)

- wizard 실행 (Project > model(우클릭) > New > Other > Glue Activity Diagram )

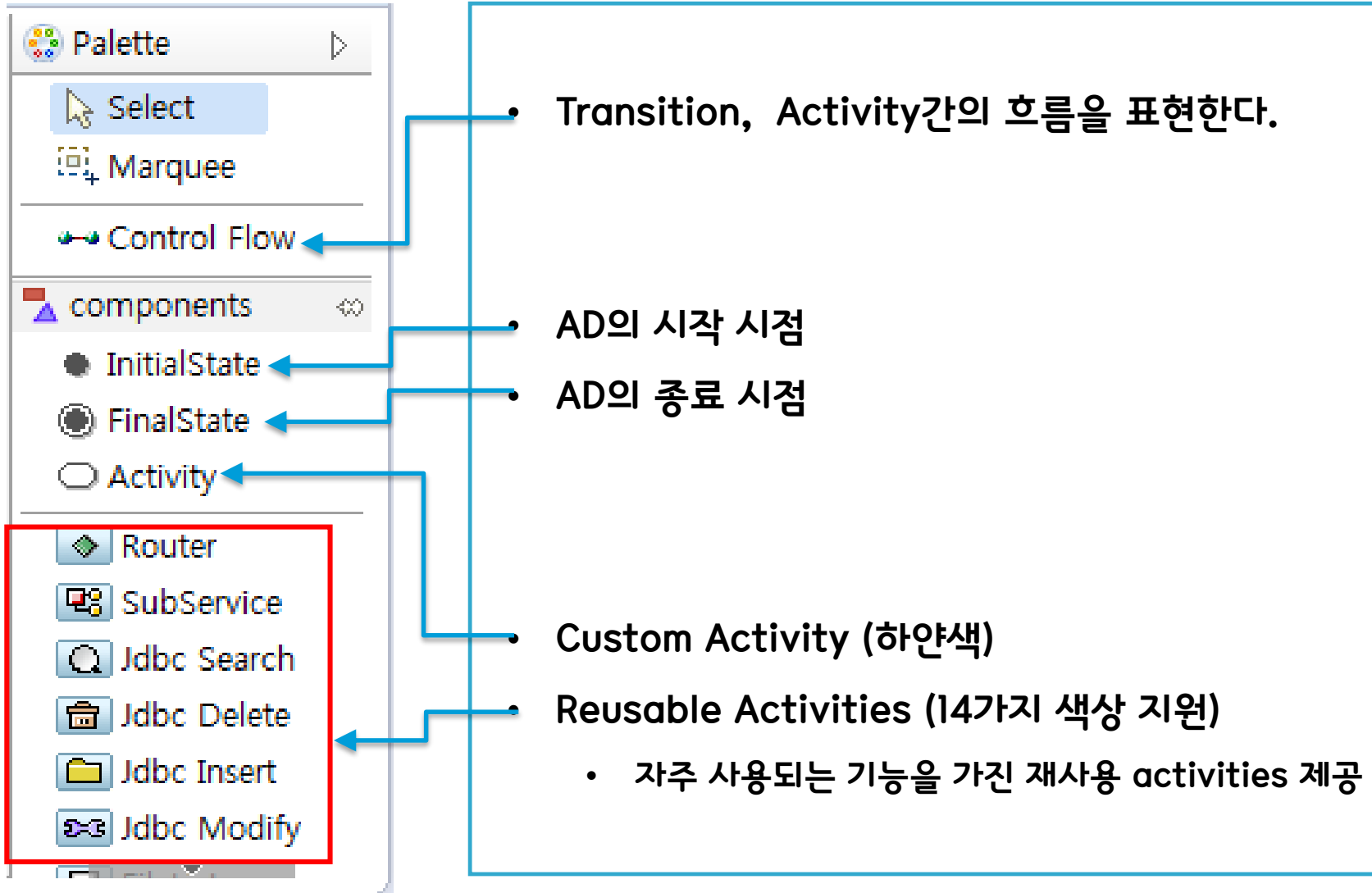


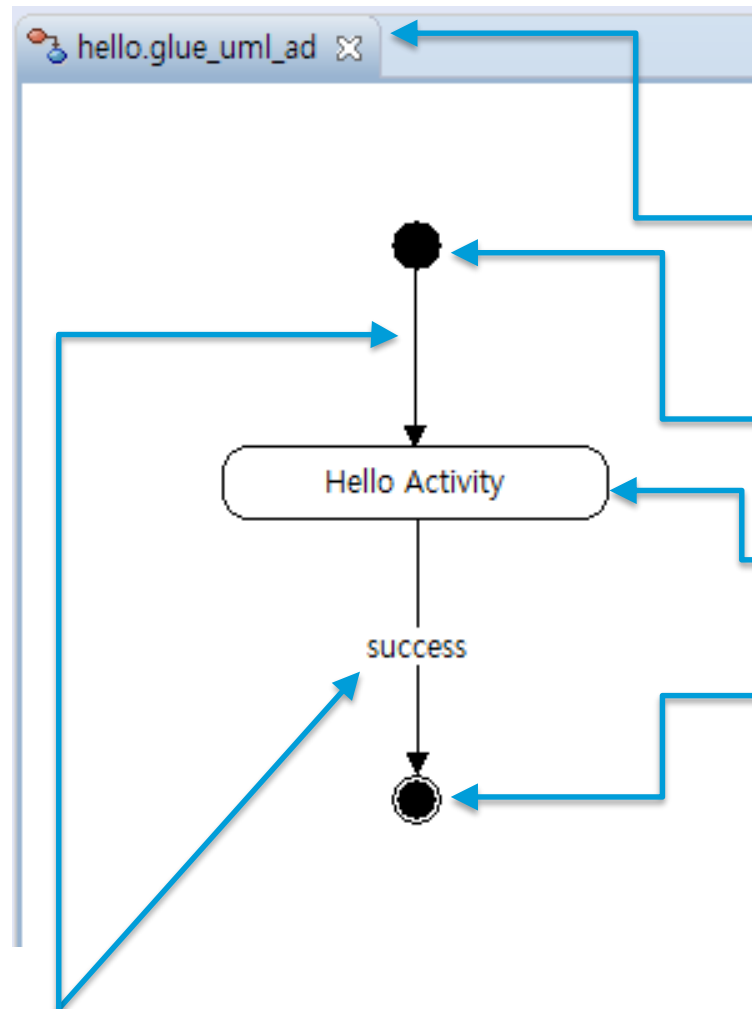


- ④ 파일 위치 지정.
- ⑤ name.glue\_uml\_ad 입력
  - name은 프로그램ID 권장.
  - 서비스명을 파일이름으로 기재
  - Glue Service 파일은 name-service.xml 으로 생성됨.



- Glue Framework에서 AD 작성시에는 다음 컴포넌트를 사용한다.





• Double click 하여 AD 파일 열기 ( hello.glue\_uml\_ad )

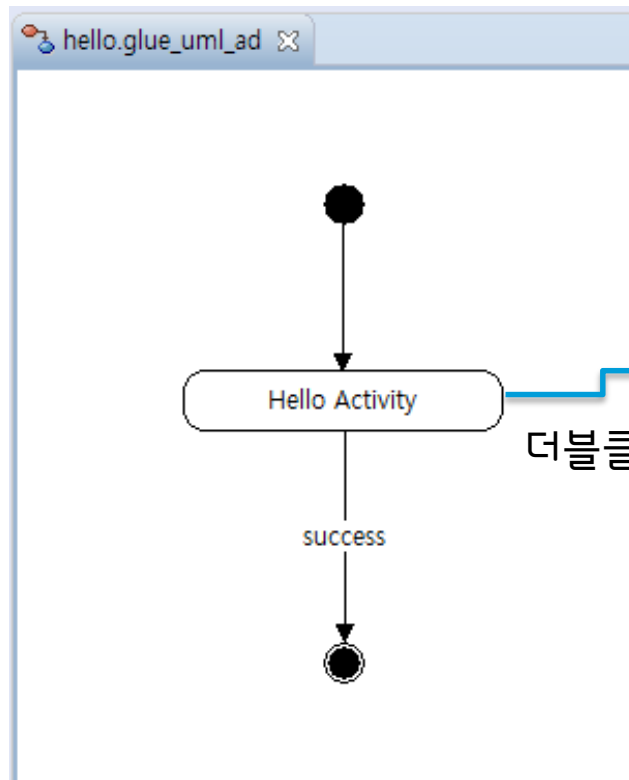
• InitialState 추가

• Custom activity 추가

• FinalState 추가

• Control flow 로 각 객체 연결

- property 편집 버튼 (Add, Delete) 사용가능
- 소스 편집 버튼 (Create Class, Open Class) 사용가능



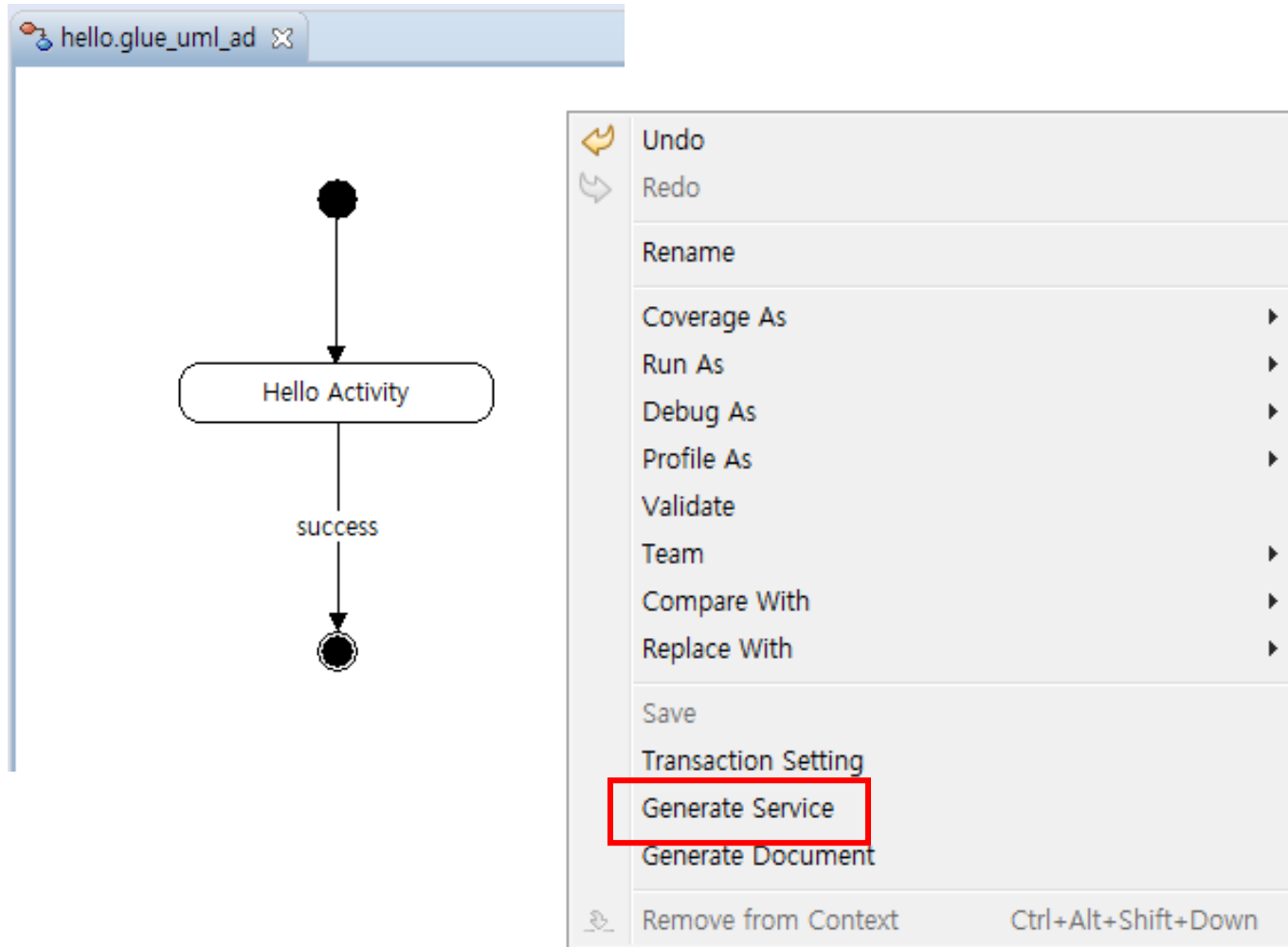
더블클릭

Name	Value
<input checked="" type="checkbox"/> class	sample.HelloActivity

Package를 포함한 class 이름 입력

Buttons: Add, Delete, Edit, Create Class, Open Class, Open Service, OK, Cancel

- AD 바탕에 우클릭 > Generate Service



## Package Explorer

### sample

#### src

▷ (default package)

▷ layout

▷ query

▷ sample

▷ HelloActivity.java

▷ sample.activity

▷ sample.vo

▷ service

hello-service.xml

sample-service.xml

applicationContext.xml

ehcache.xml

logback-test.xml

MyResource\_en\_US.properties

MyResource\_ko\_KR.properties

- 입력한 경로대로 만들어진 java 파일

- service 폴더에 만들어진 service 파일

- hello-service.xml 내용

```
<?xml version="1.0" encoding="UTF-8"?>
<service name="hello-service" initial="HelloActivity"
xmlns="http://www.poscoict.com/glueframework/service">
  <activity name="HelloActivity" class="sample.HelloActivity">
    <transition name="success" value="end"/>
  </activity>
</service>
```

- Activity 의 내용을 아래와 같이 구현한다.
- `sample.HelloActivity`

```
package sample;

import com.poscoict.glueframework.biz.activity.Glu
import com.poscoict.glueframework.biz.control.GlueBizControlConstants;
import com.poscoict.glueframework.context.GlueContext;

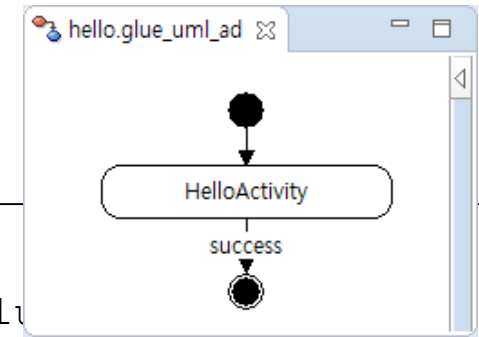
public class HelloActivity extends GlueActivity<GlueContext> {

    public String runActivity( GlueContext ctx ) {
        System.out.println( "ServiceName : " +
ctx.get( GlueBizControlConstants.SERVICE_NAME ) );
        System.out.println( "This is '" + this.getName() + "'
activity");

        Object input = ctx.get( "input" );
        System.out.println( "data : " + input );

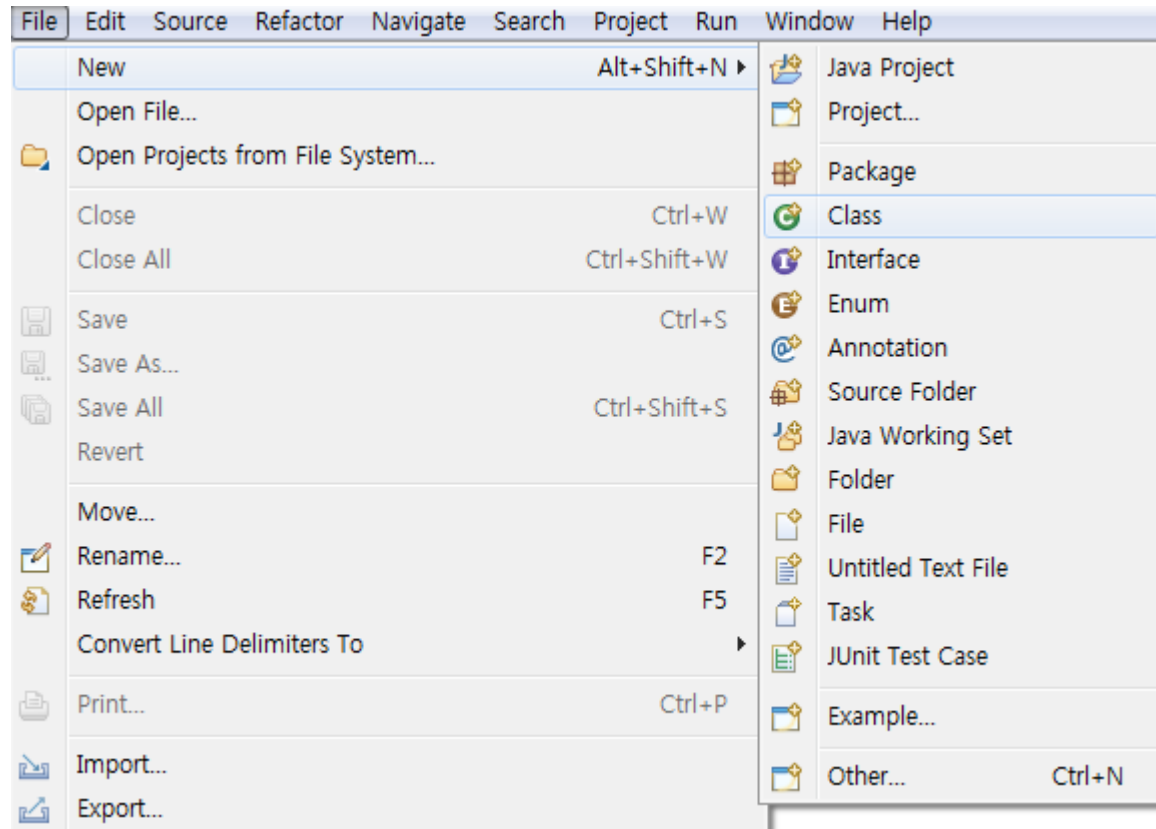
        Object output = "Hello " + input + "!!!";
        ctx.put( "result", output );

        return GlueBizControlConstants.SUCCESS;
    }
}
```

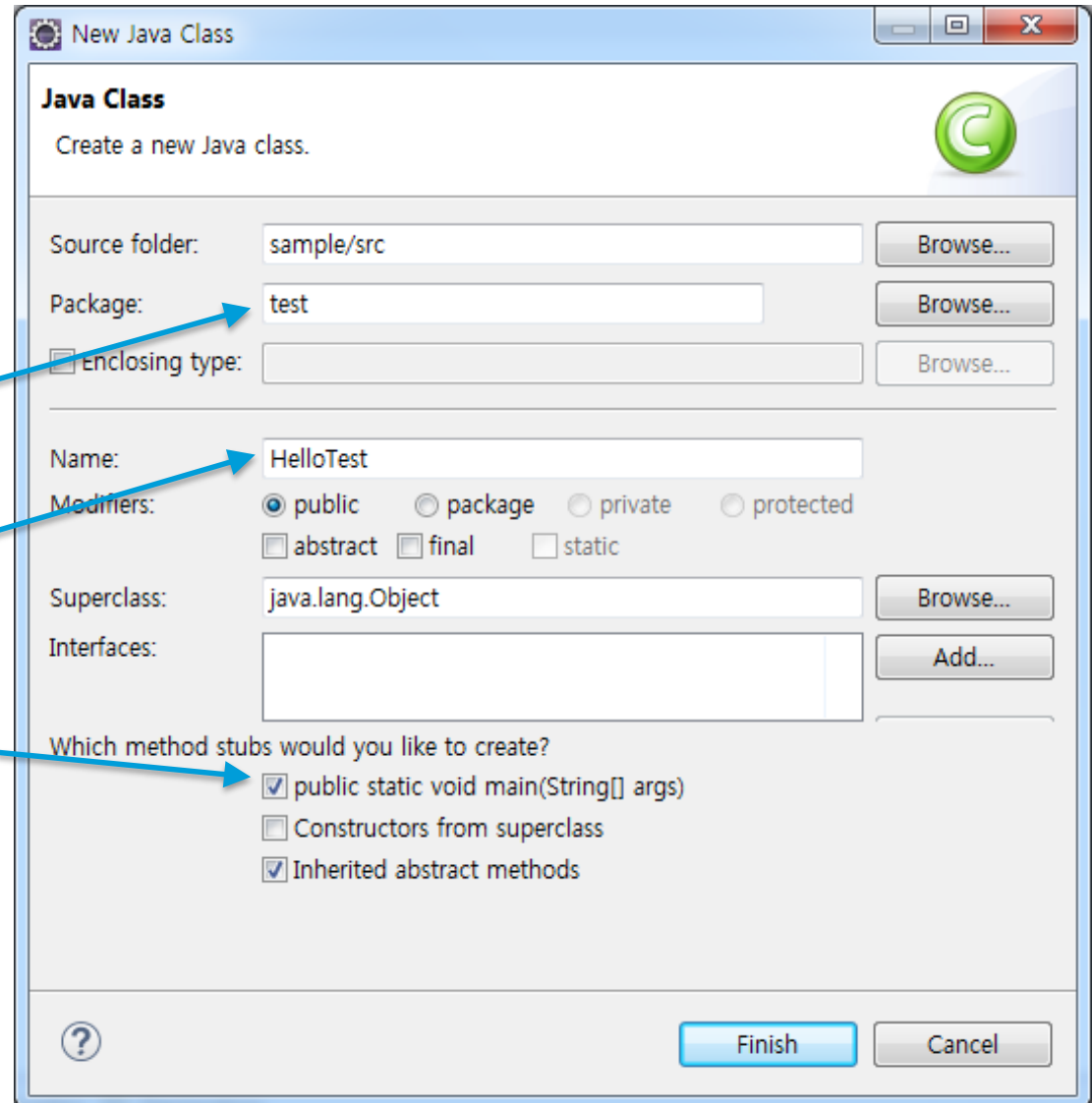




- File > New > Class



- Package : test
- Name : HelloTest
- public static void main 체크



- **test.HelloTest** 를 아래와 같이 구현한다.

```
package test;

import com.poscoict.glueframework.biz.control.GlueBizControlConstants;
import com.poscoict.glueframework.biz.control.GlueBizController;
import com.poscoict.glueframework.biz.control.GlueBizProvider;
import com.poscoict.glueframework.context.GlueContext;
import com.poscoict.glueframework.context.GlueDefaultContext;

public class HelloTest
{
    public static void main( String[] args )
    {
        GlueContext ctx = new GlueDefaultContext();
        ctx.put( GlueBizControlConstants.SERVICE_NAME, "hello-
service" );
        ctx.put( "input", "Glue" );

        GlueBizController controller =
        GlueBizProvider.getController();
        controller.doAction( ctx );

        System.out.println( ctx.get( "result" ) );
    }
}
```

- **Ctrl + Shift + F** : Source Format
- **Ctrl + Shift + O** : Organize Imports

```
build.properties  logback-test.xml  applicationContext.xml X
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/beans
5         http://www.springframework.org/schema/beans/spring-beans-4.0.xsd">
6
7 <bean id="serviceManager" class="com.poscoict.glueframework.biz.control.GlueServiceManagerImpl" lazy-init="true">
8     <property name="cacheManager" ref="cacheManager" />
9     <property name="serviceLoader" ref="serviceLoader" />
10 </bean>
11 <bean id="cacheManager" class="com.poscoict.glueframework.cache.ehcache.GlueEhCacheManager"/>
12 <bean id="serviceLoader" class="com.poscoict.glueframework.biz.control.GlueServiceLoader"/>
13
14 <!--
15 <bean id="test-dao" class="com.poscoict.glueframework.dao.jdbc.GlueJdbcDao">
16     <property name="dataSource" ref="test-datasource"/>
17     <property name="queryManager" ref="queryManager"/>
18 </bean>
19 <bean id="test-tx" class="com.poscoict.glueframework.transaction.GlueDataSourceTransactionM
20     <property name="dataSource" ref="test-datasource"/>
21 </bean>
22 <bean id="test-datasource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="
23     <property name="driverClassName" value="org.sqlite.JDBC"/>
24     <property name="url" value="jdbc:sqlite:C://apache-tomcat-8.5.16/sample.db"/>
25     <property name="defaultAutoCommit" value="false"/>
26     <property name="minIdle" value="0"/>
27     <property name="maxActive" value="-1"/>
28     <property name="maxIdle" value="1000"/>
29 </bean>
30 <bean id="queryManager" class="com.poscoict.glueframework.dao.manager.GlueQueryManagerImpl"
31     <property name="cacheManager" ref="cacheManager"/>
32     <property name="queryLoader" ref="queryLoader"/>
```

- 간단한 샘플 제작을 위해  
기본 빈 정보만 남기고 나머지는 주석처리

Package Explorer

- sample
  - src
    - layout
    - query
    - sample.activity
    - sample.vo
    - service
    - test
      - applicationContext.xml
      - ehcache.xml
      - logback-test.xml
      - MyResource\_en\_US.properties
      - MyResource\_ko\_KR.properties
  - JRE System Library [jdk1.6]
  - GLUE\_CORE
  - GLUE\_STD
  - model
  - WebContents
  - build.properties
  - build.xml

- Run As – Java Application 실행

```

Problems @ Javadoc Declaration Console
<terminated> HelloTest [Java Application] C:\Java\jdk1.8.0_162\bin\javaw.exe (2018. 2. 27. 오후 4:39:10)
16:39:11,960 |-INFO in ch.qos.logback.core.joran.action.NestedComplexPropertyIA - Assuming default type [ch.qos.logback.classic.encoder.PatternLayoutEncoder] f
16:39:11,961 |-INFO in ch.qos.logback.core.FileAppender[FILE] - File property is set to [glue_app_debug_log.log]
16:39:11,965 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - com.poscoict.glueframework level set to DEBUG
16:39:11,966 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - com.poscoict.glueframework.util.license.GlueLicenseManager level set to TRACE
16:39:11,966 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - org.springframework.jdbc.core.StatementCreatorUtils level set to TRACE
16:39:11,966 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - com.poscoict.glueframework.dao.manager.GlueQueryLoader level set to TRACE
16:39:11,967 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - org.springframework level set to INFO
16:39:11,967 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - net.sf.ehcache level set to INFO
16:39:11,967 |-INFO in ch.qos.logback.classic.joran.action.RootLoggerAction - Setting level of ROOT logger to WARN
16:39:11,967 |-INFO in ch.qos.logback.core.joran.action.AppenderRefAction - Attaching appender named [CONSOLE] to Logger[ROOT]
16:39:11,969 |-INFO in ch.qos.logback.core.joran.action.AppenderRefAction - Attaching appender named [FILE] to Logger[ROOT]
16:39:11,969 |-INFO in ch.qos.logback.classic.joran.action.ConfigurationAction - End of configuration.
16:39:11,970 |-INFO in ch.qos.logback.classic.joran.JoranConfigurator@3bf400 - Registering current configuration as safe fallback point
16:39:11,987 [main] INFO com.poscoict.glueframework.context.GlueStaticContext - glueframework version : 5.0.0-RELEASE
16:39:11,997 [main] INFO com.poscoict.glueframework.context.GlueStaticContext - EDUCATION
16:39:12,000 [main] WARN com.poscoict.glueframework.context.GlueStaticContext - Check Java Option 'CONFIG_PATH'
16:39:12,113 [main] INFO org.springframework.context.support.ClassPathXmlApplicationContext - Refreshing org.springframework.context.support.ClassPathXmlAppli
16:39:12,248 [main] INFO org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from class path resource [applicationCon
16:39:12,616 [main] INFO com.poscoict.glueframework.cache.ehcache.GlueEhCacheManager - create EhCache CacheManger
16:39:13,374 [main] INFO com.poscoict.glueframework.cache.ehcache.GlueEhCacheManager - start EhCache CacheManger : __DEFAULT__
16:39:13,375 [main] DEBUG com.poscoict.glueframework.cache.ehcache.GlueEhCacheManager - [_sql_region, _default_region, _message_definition_region, _service-reg
16:39:13,378 [main] INFO com.poscoict.glueframework.biz.control.GlueServiceLoader - 'service' url : file:/C:/eclipse/workspace/sample/classes/service
16:39:13,467 [main] WARN com.poscoict.glueframework.biz.control.GlueBizControllerImpl - running in EDUCATION MODE : 1
16:39:13,467 [main] INFO com.poscoict.glueframework.biz.control.GlueBizControllerImpl - ServiceName:[hello-service] StartTime[Tue Feb 27 16:39:13 KST 2018]
16:39:13,469 [main] DEBUG com.poscoict.glueframework.biz.control.GlueServiceLoader - Loading service file : file:/C:/eclipse/workspace/sample/classes/service%5
16:39:13,853 [main] INFO com.poscoict.glueframework.biz.control.GlueService$GlueActivityHandler - ActivityName:[hello-service][HelloActivity] StartTime[Tue Fe
ServiceName : hello-service
This is 'HelloActivity' activity
data : Glue
16:39:13,855 [main] INFO com.poscoict.glueframework.biz.control.GlueService$GlueActivityHandler - ActivityName:[hello-service][HelloActivity] EndTime[Tue Feb
16:39:13,855 [main] INFO com.poscoict.glueframework.biz.control.GlueBizControllerImpl - ServiceName:[hello-service] EndTime[Tue Feb 27 16:39:13 KST 2018] RunT
Hello Glue!!!
    
```

- Activity란?
  - Activity는 단위 기능을 수행하며 GlueActivity를 상속 받은 Class 이다.
  - Activity는 Service 뿐만이 아니라 다른 Service에서도 공유 할 수 있으므로 재사용 가능하게 설계 하는 것이 좋다.

- 자주 사용 될만한 기능을 미리 정의 해 둔 Activity
  - Reusable Activity 에서 사용되는 다양한 property 는 javadoc 문서에서 확인
  - 사용자가 별도의 Reusable Activity를 만들어 사용할 수도 있다
- GlueSDK 내 config 폴더의 glue-config.xml 에 정의
  - 사용 여부 , 지정된 색상 정의 가능
- 대표적인 Reusable Activity
  - Router Activity
  - GlueJdbc CRUD 관련 Activity
  - 전문관련 Message parsing and creating Activity
  - SubService Activity
    - GlueService를 재사용 할 수 있는 기능을 제공하는 activity
    - <https://www.solutionpot.co.kr/doc/framework/guide/activity-subservice.html>

- **Name** : Activity의 Name은 Service 에서 유일하여야 한다.
- **Class** : 반드시 GlueActivity Class를 상속 받은 Class이어야 한다.



- **transition**

- Activity는 반드시 하나 이상의 transition을 가지고 있어야 하며 Router가 아닌 Activity는 반드시 "success" transition을 가지고 있어야 한다.
- 만일 해당 Activity가 다음에 수행할 Activity가 존재하지 않는다면 <transition name="success" value="end"/> 로 지정하여야 한다.
- success transition은 Activity Class가 Error 없이 정상 수행했을 경우이고 Error가 발생하면 failure transition이 수행 된다.
- 만일 failure 가 정의 되지 않았다면 Service는 바로 종료 되고 Service를 호출한 Caller에게 Exception을 발생 하게 된다.

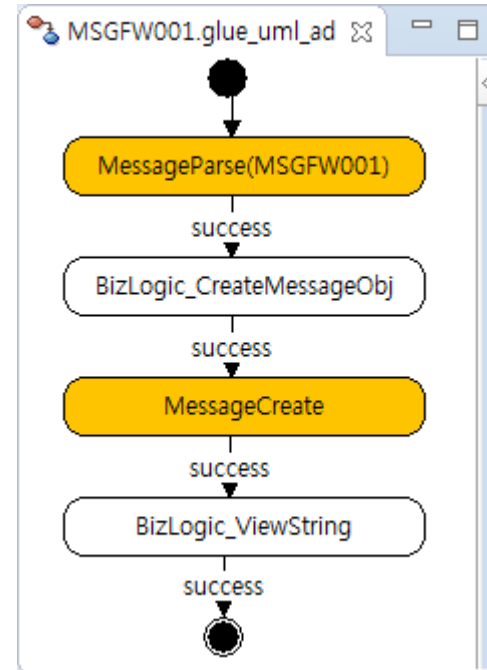
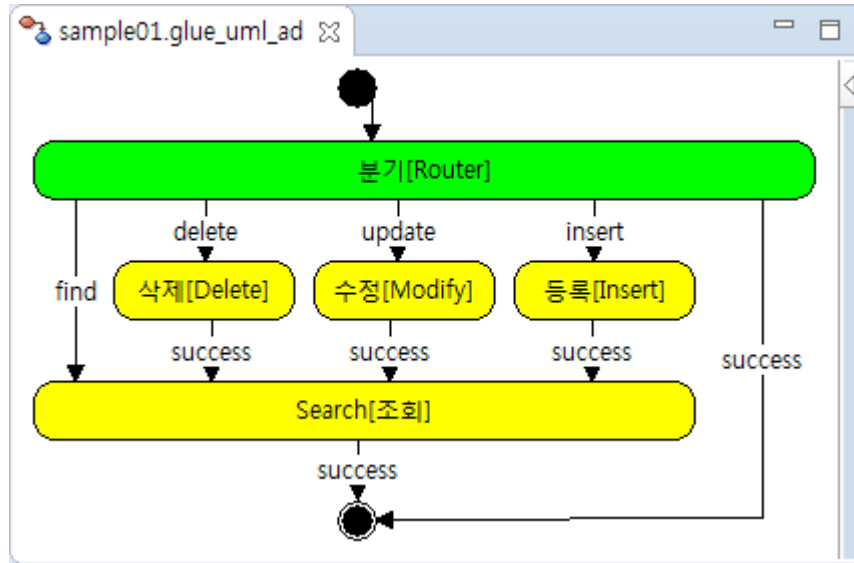
- **property**

- property는 해당 Activity에서 사용할 수 있는 사용자 정의 Static Data이다.
- 해당 Property에 개수 제한은 없고 필요한 만큼 선언 하여 사용할 수 있다.
- 하나의 Activity내에서 같은 이름의 Property는 존재 할 수 없다.



- **System Level의 Business 설계**
  - Activity Diagram을 통해 기능(Activity)과 기능들의 Flow를 정의
- **Service 자동 생성**
  - Glue F/W의 Business Control 에서 제어 되는 Service.xml이 생성
- **Service 명세서 자동 생성**
  - 설계된 Service 명세서를 자동으로 생성함.
  - System Level의 Use Case Diagram과 명세서, Class Diagram, Sequence Diagram, 기능 상세 정의서를 대체 할 수 있다.
- **Service에 정의된 Custom Class 자동 생성**
  - Service에 정의된 Custom Class를 Service 생성시 자동 생성함.
- **Activity 정의 기능**
  - Activity를 정의하고 개발자와 설계자간의 Communication 역할을 함.
- **Reusable Activity 자동 Setting 기능**
  - Preference에 등록하여 반복적인 작업을 줄여줌.
- **Site 공통 Activity 정의 및 등록 기능**
  - Glue 배포 Activity 뿐만이 아니라 Custom으로 작성된 Activity도 팔레트에 등록하여 사용 할 수 있음.

- Service 란?

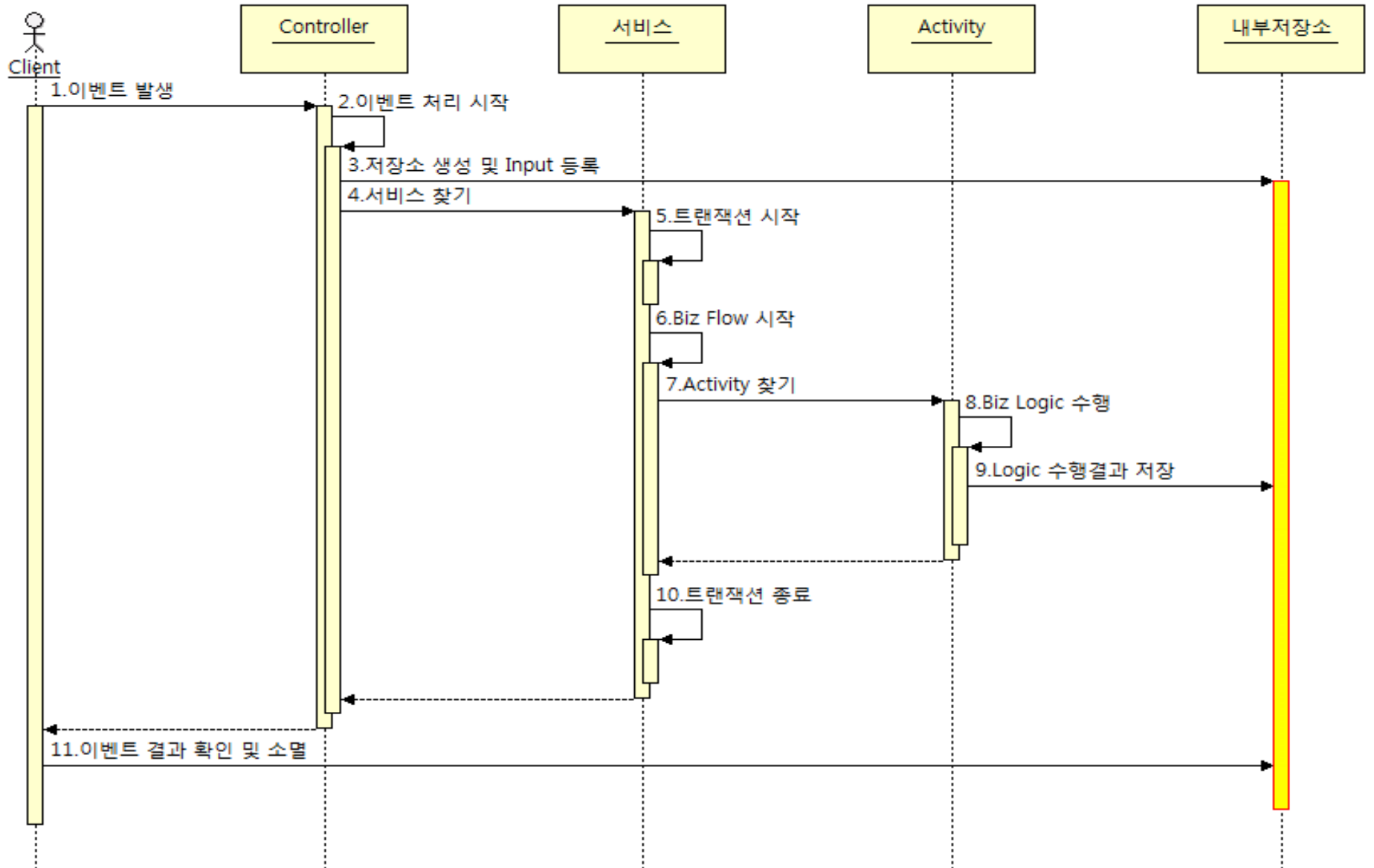


- Service는 독립적인 Business의 한 기능을 수행하는 단위로 보통 재사용이 가능한 단위로 작성하게 된다. 즉 Service는 아주 작은 하나의 기능단위로 할 수도 있고 반대로 모든 Business를 하나의 Service로도 처리할 수 있지만 이것은 설계자의 능력에 따라 정의 될 수 있다.

- Glue Activity Diagram 으로부터 생성.
- transaction-manager와 activity들로 구성되어 있음.
- Service 파일 이름 규칙
  - ServiceName.glue\_uml\_ad → ServiceName-service.xml
- 애플리케이션 상의 위치
  - service\
    - GlueSample.war! \WEB-INF \ classes \ service
    - user-application.jar! \ service
- Glue Project 상의 위치
  - src\service
    - <sample> \ src \ service
    - <sample-maven> \ src \ main \ resource \ service



- GlueContext는 Request 마다 생성되는 Data Transfer Object
- GlueContext는 JSP 부터 Activity 까지 모든 Data의 공유 통로
- Web에서 GlueContext는 Request, Session, Attribute Data까지 포함합니다.
- 다양한 EventType, ContentType 이 존재함. 아래 링크 참조
  - <https://www.solutionpot.co.kr/doc/framework/docs/core-context.html>





# IV. Glue Framework 제작 가이드

## 2장 진화 학습

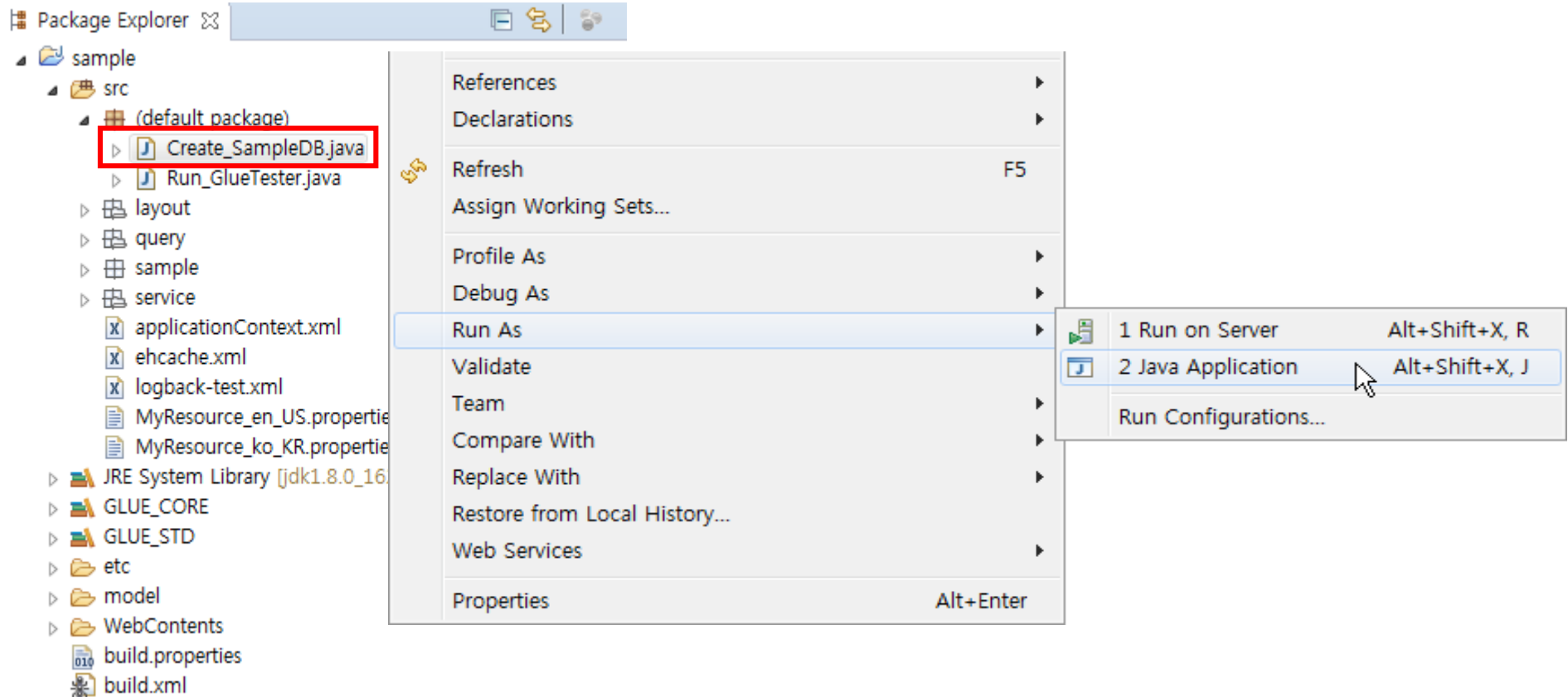






목표	실습
<ul style="list-style-type: none"> <li>• applicationContext.xml을 설정할 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>• applicationContext.xml 변경</li> </ul>
<ul style="list-style-type: none"> <li>• DB 연동 service를 만들 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>• Activity Diagram 그리기                             <ul style="list-style-type: none"> <li>• sample01.glue_uml_ad</li> <li>• dept.glue_uml_ad</li> </ul> </li> <li>• Service 만들기                             <ul style="list-style-type: none"> <li>• sample01-service.xml</li> <li>• dept-service</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• DB 데이터를 화면에 표시할 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>• JSP 만들기                             <ul style="list-style-type: none"> <li>• sample01.jsp</li> <li>• dept.jsp</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• Query Editor를 사용할 수 있다.</li> <li>• Named query를 사용할 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>• sample용 db 만들기 (sqlite)</li> <li>• Query 파일 만들기                             <ul style="list-style-type: none"> <li>• sample01-query.glue_sql</li> <li>• dept-query.glue_sql</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• Web application을 만들고 배포할 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>• Build 하기                             <ul style="list-style-type: none"> <li>• build.properties</li> </ul> </li> </ul>

- Create\_SampleDB class 실행
- sample.db(파일DB)가 생성됨 → %CATALINA\_HOME% 으로 복사



```
applicationContext.xml
3  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4  xsi:schemaLocation="http://www.springframework.org/schema/beans
5                        http://www.springframework.org/schema/beans/spring-beans-4.0.xsd">
6
7  <bean id="serviceManager" class="com.poscoict.glueframework.biz.control.GlueServiceManagerImpl" lazy-init="true">
8      <property name="cacheManager" ref="cacheManager" />
9      <property name="serviceLoader" ref="serviceLoader" />
10 </bean>
11 <bean id="cacheManager" class="com.poscoict.glueframework.cache.ehcache.GlueEhCacheManager"/>
12 <bean id="serviceLoader" class="com.poscoict.glueframework.biz.control.GlueServiceLoader"/>
13
14 <bean id="test-dao" class="com.poscoict.glueframework.dao.jdbc.GlueJdbcDao">
15     <property name="dataSource" ref="test-datasource"/>
16     <property name="queryManager" ref="queryManager"/>
17 </bean>
18 <bean id="test-tx" class="com.poscoict.glueframework.transaction.GlueDataSourceTransactionManager">
19     <property name="dataSource" ref="test-datasource"/>
20 </bean>
21 <bean id="test-datasource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
22     <property name="driverClassName" value="org.sqlite.JDBC"/>
23     <property name="url" value="jdbc:sqlite:/C://apache-tomcat-8.5.16/sample.db"/>
24     <property name="defaultAutoCommit" value="false"/>
25     <property name="minIdle" value="0"/>
26     <property name="maxActive" value="-1"/>
27     <property name="maxIdle" value="1000"/>
28 </bean>
29 <bean id="queryManager" class="com.poscoict.glueframework.dao.manager.GlueQueryManagerImpl">
30     <property name="cacheManager" ref="cacheManager"/>
31     <property name="queryLoader" ref="queryLoader"/>
32 </bean>
33 <bean id="queryLoader" class="com.poscoict.glueframework.dao.manager.GlueQueryLoader"/>
34
35 </beans>
```

- GlueServiceManager
  - GlueServiceLoader 를 이용해 저장한 xml 을 parsing 해 ServiceInfo 리턴

```
<bean id="serviceManager"
      class="com.poscoict.glueframework.biz.control.GlueServiceManagerImpl"
      lazy-init="true">
  <property name="cacheManager" ref="cacheManager" />
  <property name="serviceLoader" ref="serviceLoader" />
</bean>
```

- GlueServiceLoader
  - service.xml 파일을 cache 에 저장 (ehcache 이용)

```
<bean id="serviceLoader"
      class="com.poscoict.glueframework.biz.control.GlueServiceLoader"/>
```

- GlueJdbcDao
  - DAO : Data Access Object
  - org.springframework.jdbc.core.namedparam.NamedParameterJdbcDaoSupport  
를 상속

```
<bean id="test-dao" class="com.poscoict.glueframework.dao.jdbc.GlueJdbcDao">  
  <property name="dataSource" ref="test-datasource"/>  
  <property name="queryManager" ref="queryManager"/>  
</bean>
```

- **GlueQueryManager**

- GlueQueryLoader 를 이용해 저장한 xml 을 parsing 해 GlueQueryDefinition 리턴

```
<bean id="queryManager"
      class="com.poscoict.glueframework.dao.manager.GlueQueryManagerImpl">
  <property name="cacheManager" ref="cacheManager"/>
  <property name="queryLoader" ref="queryLoader"/>
</bean>
```

- **GlueQueryLoader**

- glue\_sql 파일을 cache 에 저장 (ehcache 사용)

```
<bean id="queryLoader"
      class="com.poscoict.glueframework.dao.manager.GlueQueryLoader"/>
```

- GlueDataSourceTransactionManager

- org.springframework.jdbc.datasource.DataSourceTransactionManager 를 상속

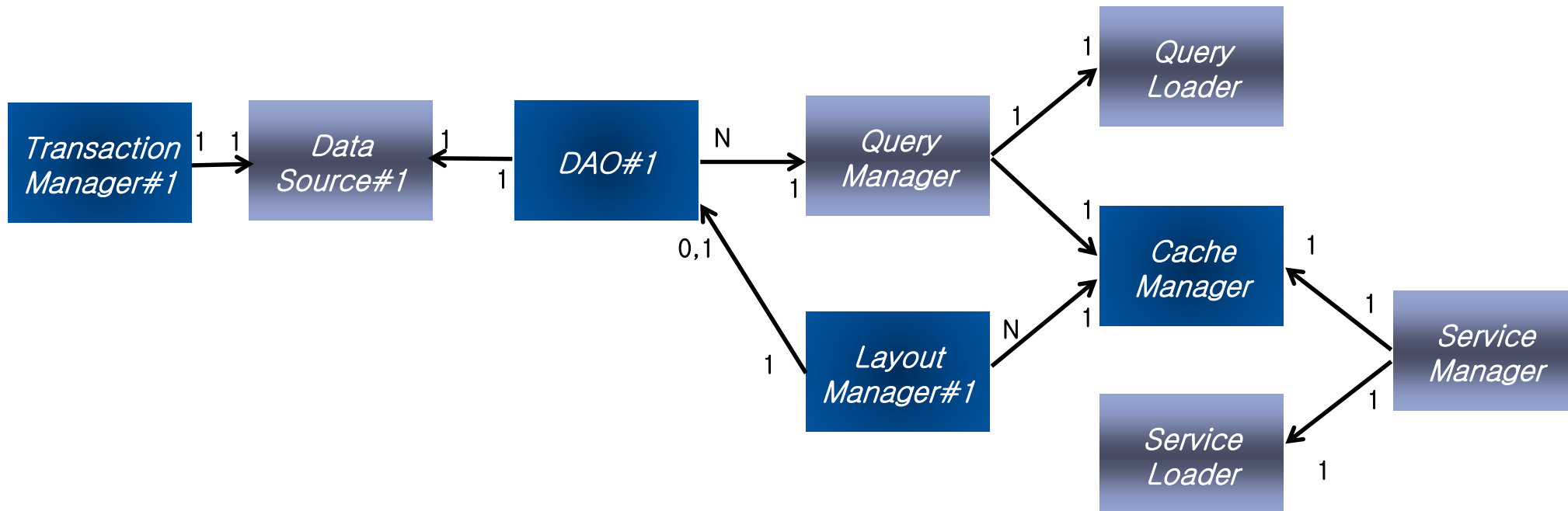
```
<bean id="test-tx"  
      class="com.poscoict.glueframework.transaction.GlueDataSourceTransactionManager">  
  <property name="dataSource" ref="test-datasource"/>  
</bean>
```

- DataSource

```
<bean id="test-datasource"  
      class="org.apache.commons.dbcp.BasicDataSource"  
      destroy-method="close">  
  <property name="driverClassName" value="org.sqlite.JDBC"/>  
  <property name="url" value="jdbc:sqlite:/C:// WAS HOME /sample.db"/>  
  <property name="defaultAutoCommit" value="false"/>  
  <property name="minIdle" value="0"/>  
  <property name="maxActive" value="-1"/>  
  <property name="maxIdle" value="1000"/>  
</bean>
```



TransactionManager : DataSource : DAO  
1 : 1 : 1





- logback-test.xml 로 설정
  - log 파일 위치 지정 : FILE appender의 <file> 요소

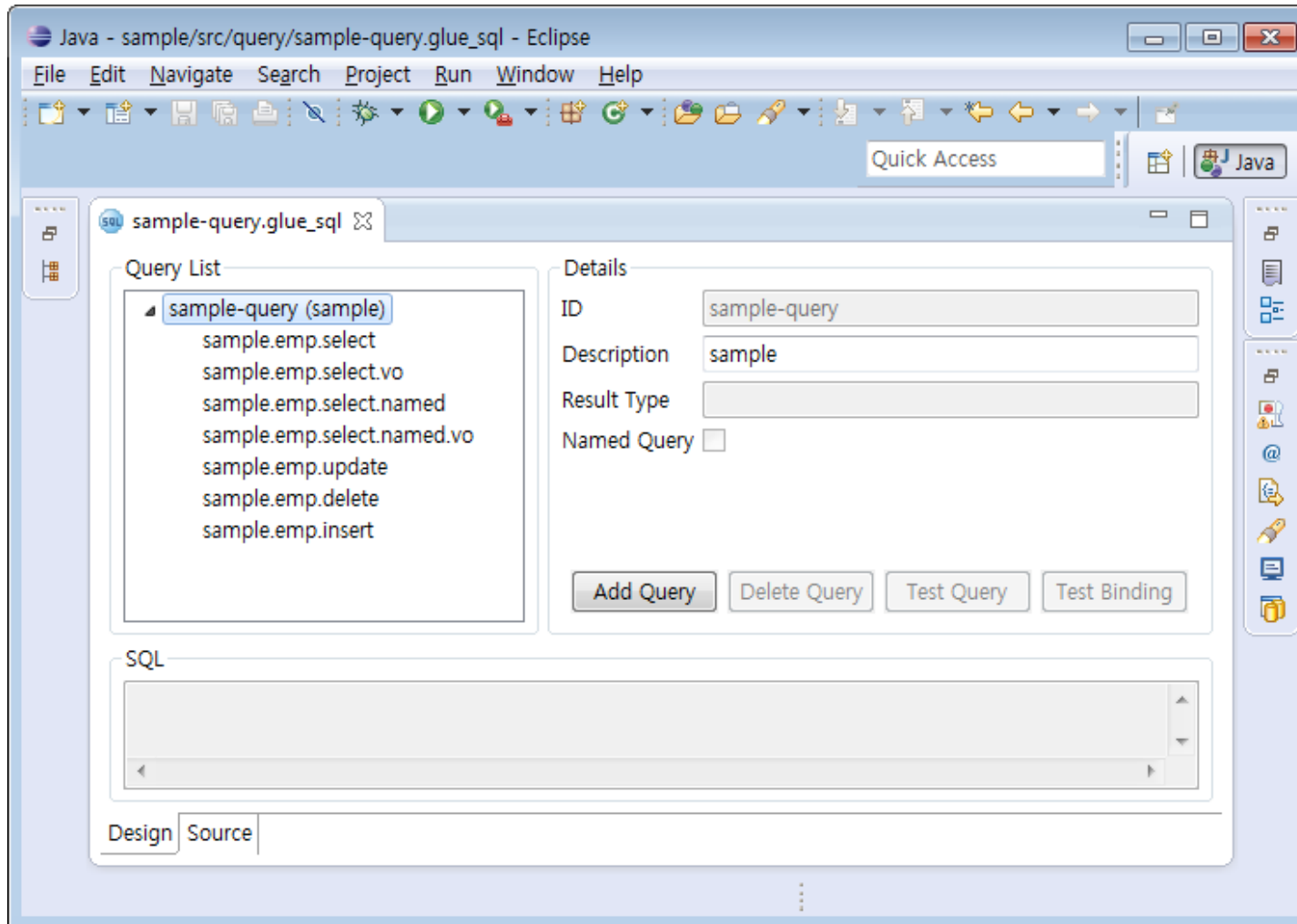
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration debug="true">
3   <appender name="CONSOLE" class="ch.qos.logback.core.ConsoleAppender">
4     <encoder>
5       <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger - %msg%n</pattern>
6     </encoder>
7   </appender>
8   <appender name="FILE" class="ch.qos.logback.core.FileAppender">
9     <file>glue_app_debug_log.log</file>
10    <append>true</append>
11    <encoder>
12      <pattern>%d [%thread] %-5level %logger - %msg%n</pattern>
13    </encoder>
14  </appender>
15
16  <logger name="sample">
17    <level value="debug" />
18  </logger>
19  <logger name="com.poscoict.glueframework">
20    <level value="debug" />
21  </logger>
22  <logger name="com.poscoict.glueframework.util.license.GlueLicenseManager">
23    <level value="trace" />
24  </logger>
25  <logger name="org.springframework.jdbc.core.StatementCreatorUtils">
26    <level value="trace" />
27  </logger>
28  <logger name="com.poscoict.glueframework.dao.manager.GlueQueryLoader">
29    <level value="trace" />
30  </logger>
31  <logger name="org.springframework">
32    <level value="info" />
```

sample 패키지 debug level 로 설정

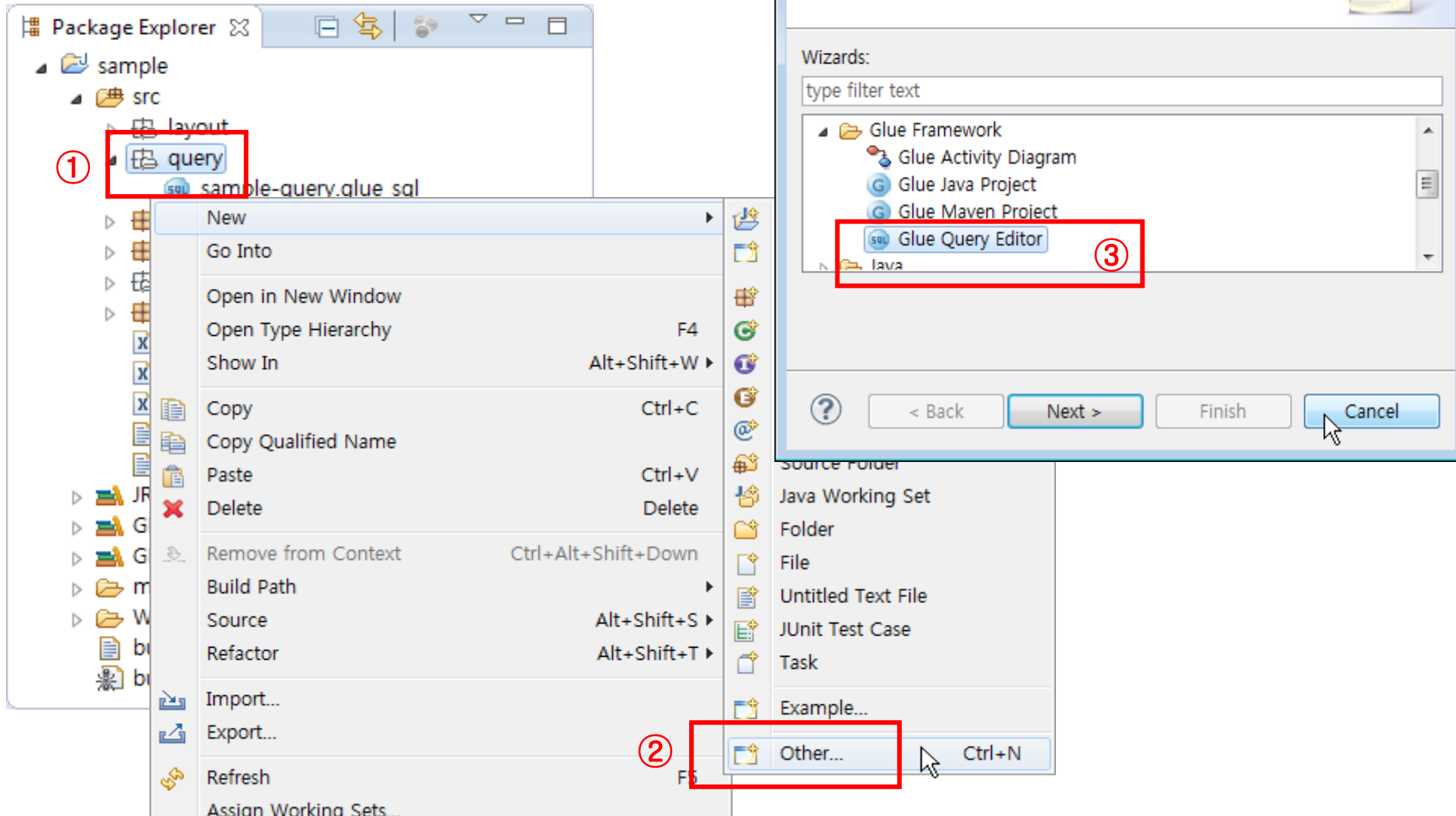
절대경로 사용시 전체 경로를 지정한다.  
directory가 없다면 생성한다.

- 자바 소스 코드 내 하드 코딩된 SQL 을 지양하고 별도 파일로 관리.
- 쿼리의 가독성과 관리 편리성 향상
- 파일명 규칙
  - xxx-query.glue\_sql ( JdbcDao 이용시)
  - xxx-hquery.glue\_sql (HibernateDao 이용시)
- 애플리케이션 상의 위치
  - query \
    - jar:file:GlueSample.war! \WEB-INF \ classes \ query
    - jar:file:user-application.jar! \ query
- Glue Project 상의 위치
  - src \ query
    - <sample> \ src \ query
    - <sample-maven> \ src \ main \ resource \ query

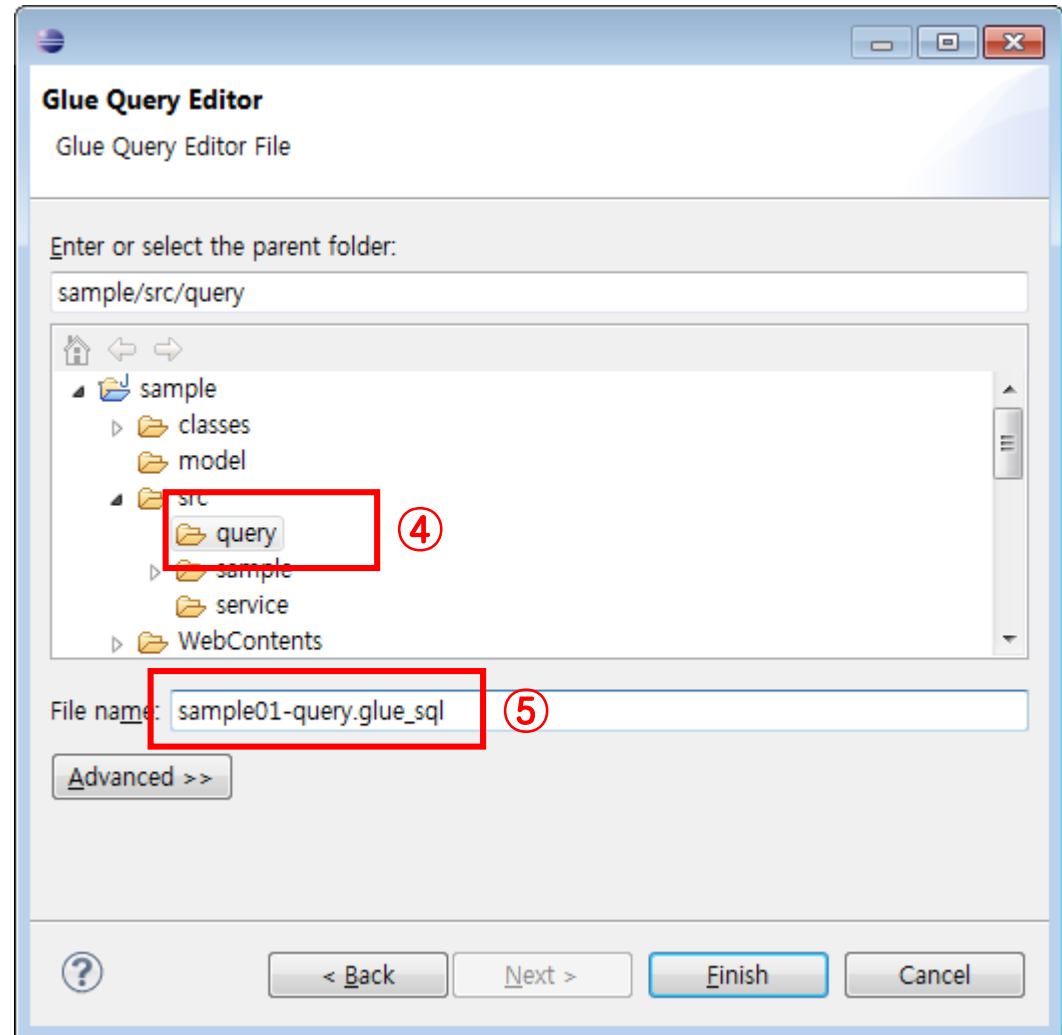
- Query 파일 생성 및 관리하는 도구.
- MyBatis DAO는 Glue Query Editor 불필요.



- wizard 실행.



- wizard 실행
  - ④ query 폴더 선택.
- File name 입력
  - ⑤ name-query.glue\_sql



The screenshot shows the Eclipse IDE interface with the 'Add Query' dialog box open. The dialog box has the following fields:

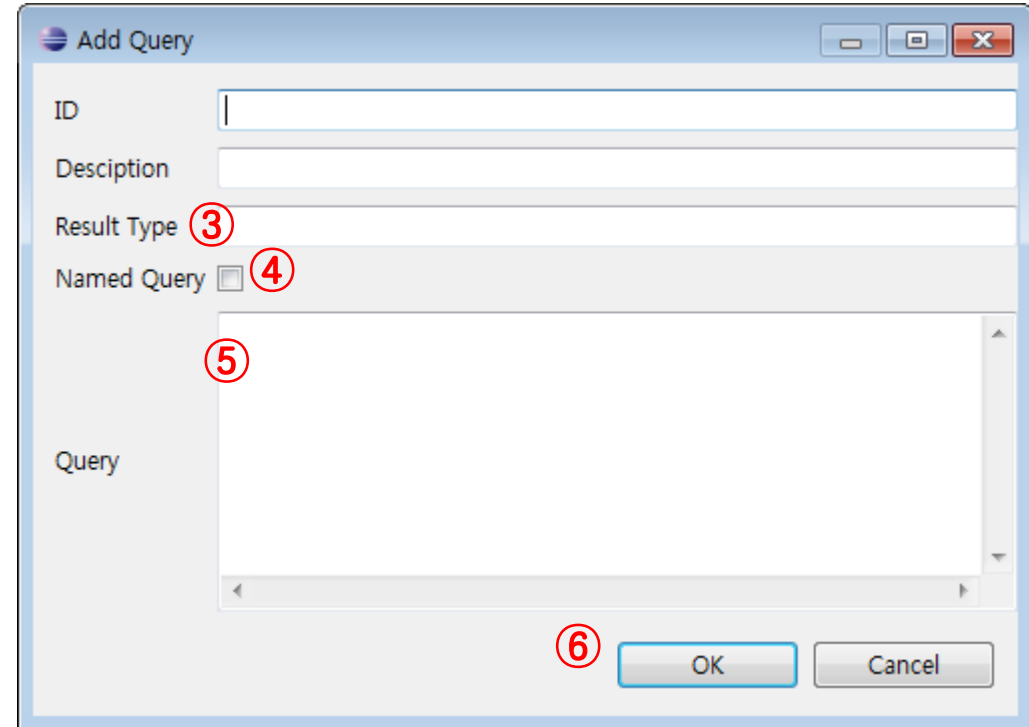
- ID: A text input field with a red circle '1' next to it.
- Description: A text input field with a red circle '2' next to it.
- Result Type: A dropdown menu.
- Named Query: A checkbox.
- Query: A large text area for entering the SQL query.

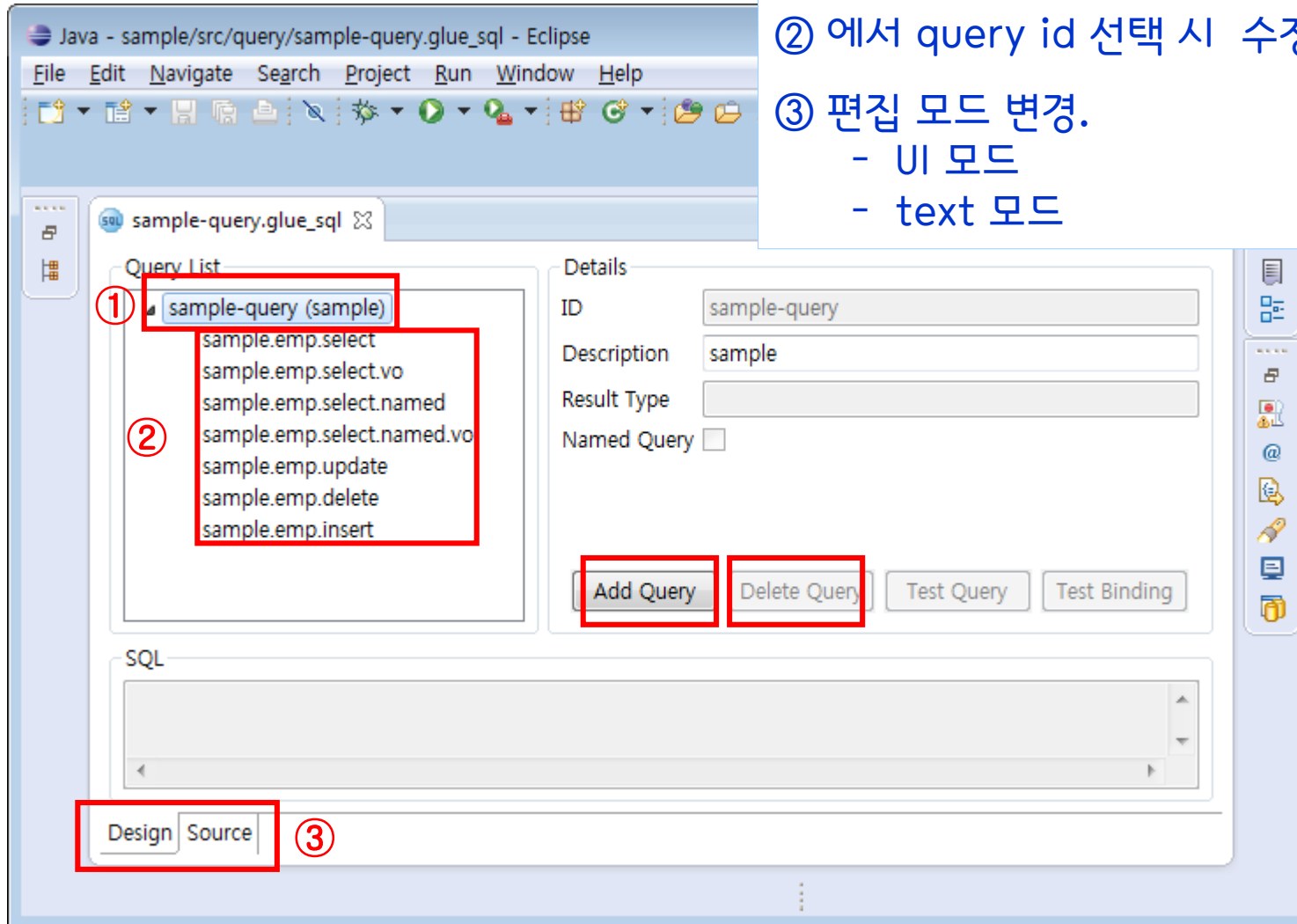
The 'Query List' panel in the background shows a list of queries, with the 'Add Query' button highlighted in red. A blue arrow points from the 'Add Query' button to the 'Add Query' dialog box.

A callout box contains the following text:

- ① ID (필수)
  - 전체 Query 파일에서 유일
- ② description
  - 해당 SQL 설명 작성

- ③ Result Type
  - select query 의 경우  
1개 row를 표현하는 객체 type
  - package 포함 class 명
- ④ Named Query (필수)
  - 아래 형태의 Named query 일 경우 체크
  - `select * from emp where deptno=?`
  - `select * from emp where deptno=:key`
- ⑤ query (필수)
- ⑥ 추가 완료





- ① 클릭 해야 Add 버튼 활성화.
- ② 에서 query id 선택 시 수정/삭제 가능
- ③ 편집 모드 변경.
  - UI 모드
  - text 모드



- `sample01-query.glue_sql` : named 쿼리

```
<queryMap desc="sample01" ...>
  <query id="sample01.emp.select" desc="" resultType="" isNamed="true">
    <![CDATA[
select EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, DEPTNO
from EMP
where DEPTNO=:deptno
    ]]>
  </query>
  <query id="sample01.emp.update" desc="" resultType="" isNamed="true">
    <![CDATA[
update emp
set sal=:sal, ename=:ename
where empno=:empno
    ]]>
  </query>
```

- `sample01-query.glue_sql` : named 쿼리

```
<query id="sample01.emp.delete" desc="" resultType="" isNamed="true">
  <![CDATA[
delete from emp
where empno=:empno
  ]]>
</query>
<query id="sample01.emp.insert" desc="" resultType="" isNamed="true">
  <![CDATA[
insert into emp(empno, ename, sal, deptno)
values (:empno, :ename, :sal, :deptno)
  ]]>
</query>
</queryMap>
```

- dept-query.glue\_sql

```
<queryMap desc="dept" ...>
  <query id="dept.select" desc="" resultType="" isNamed="false" >
    <![CDATA[
select *
from DEPT

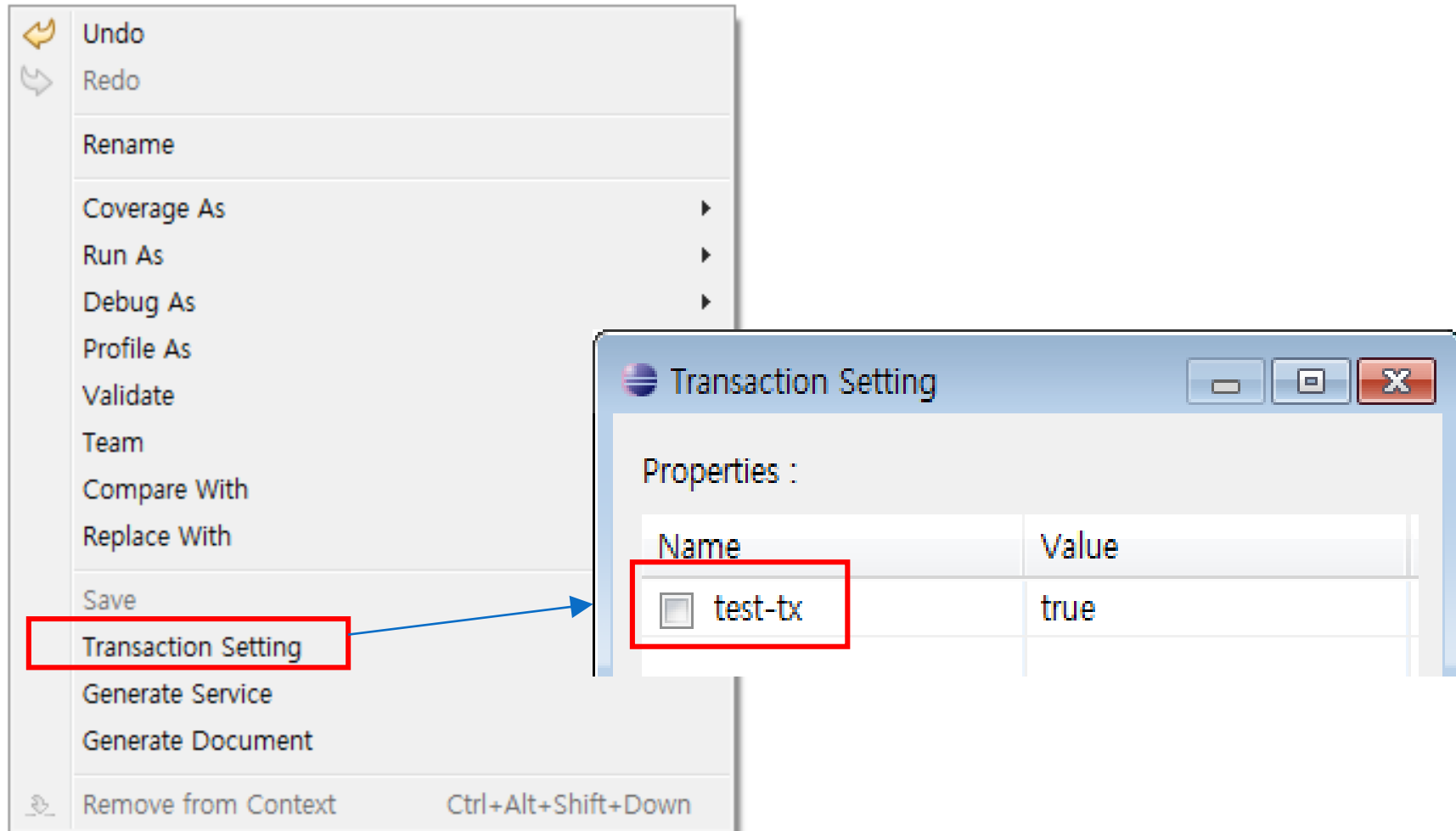
        ]]>
  </query>
  <query id="dept.update" desc="" resultType="" isNamed="false">
    <![CDATA[
update DEPT
set DNAME=?, LOC=?
where DEPTNO=?

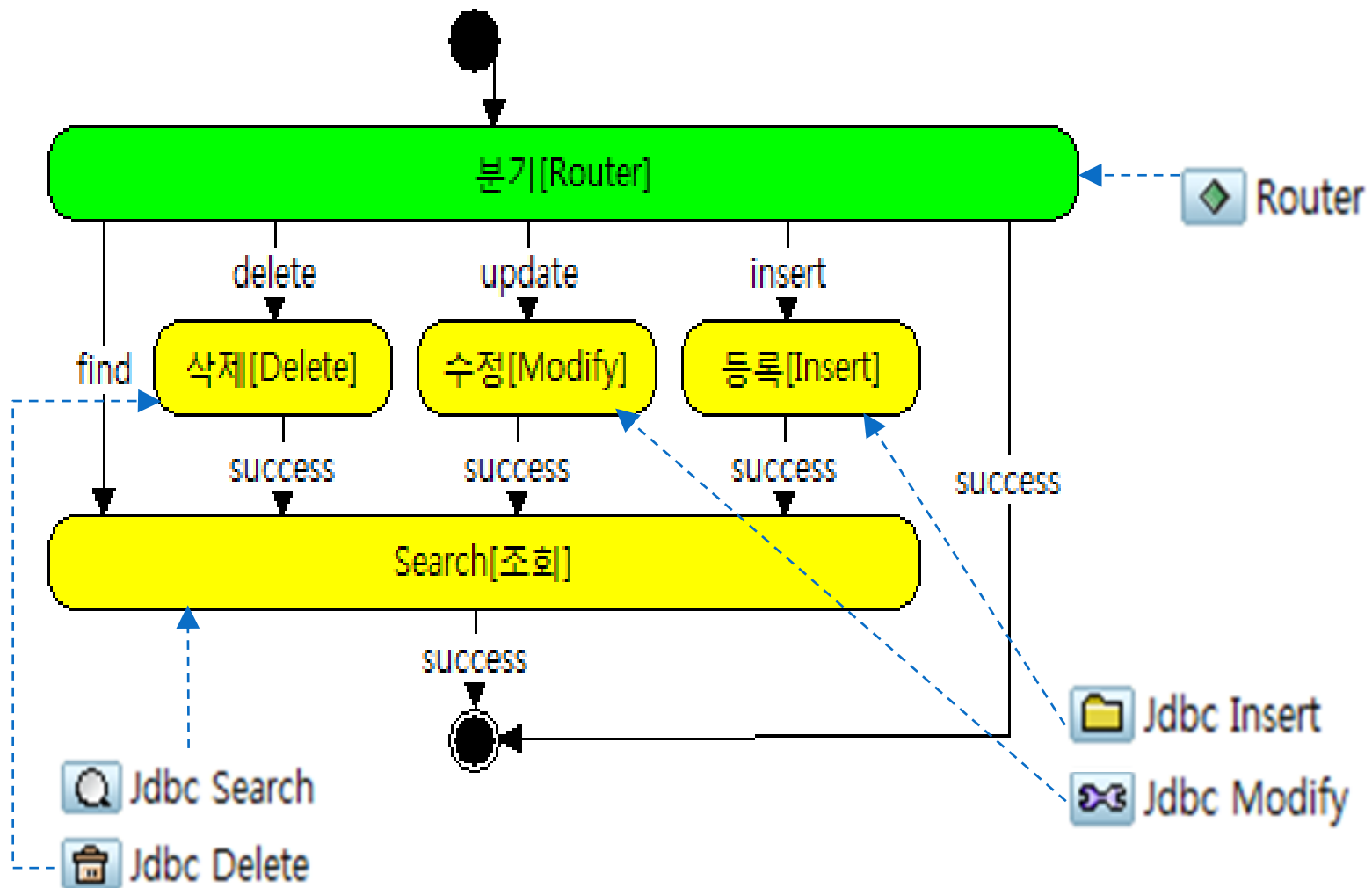
        ]]>
  </query>
```

- dept-query.glue\_sql

```
<query id="dept.delete" desc="" resultType="" isNamed="true">
  <![CDATA[
delete from DEPT
where DEPTNO=:deptno
  ]]>
</query>
<query id="dept.insert" desc="" resultType="" isNamed="false">
  <![CDATA[
insert into DEPT(DEPTNO, DNAME, LOC)
values (?, ?, ?)
  ]]>
</query>
</queryMap>
```

- AD파일 만들기 : Project > model > sample01.glue\_uml\_ad
- Transaction Manager 설정하기 : AD에서 우클릭 > Transaction Setting





Search[조회]

조회[Search]

Properties :

Name	Value
<input type="checkbox"/> class	com.poscoict.glueframework.biz.activity.GlueJdbcSearch
<input type="checkbox"/> dao	test-dao
<input type="checkbox"/> result-key	EmpList
<input type="checkbox"/> sql-key	sample01.emp.select

SQL

```
select EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, DEPTNO
from EMP
where DEPTNO=:deptno
```

- param-count 삭제
- param0, param1, .. 삭제
- param-bindings 삭제
- cache-key 삭제
- cache-region 삭제
- cache-manager 삭제

삭제[Delete]

삭제[Delete]

Properties :

Name	Value
<input type="checkbox"/> class	com.poscoict.glueframework.biz.activity.GlueJdbcDelete
<input type="checkbox"/> chk-name	chk
<input type="checkbox"/> dao	test-dao
<input type="checkbox"/> result-key	deleteCnt
<input type="checkbox"/> sql-key	sample01.emp.delete
<input type="checkbox"/> param-bindings	empno=EMPNO

- param-count 삭제
- param0, param1, .. 삭제
- list-key 삭제

SQL

```
delete from emp
where empno=:empno
```



## 등록[Insert]

등록[Insert]

Properties :

Name	Value
<input type="checkbox"/> class	com.poscoict.glueframework.biz.activity.GlueJdbcInsert
<input type="checkbox"/> chk-name	chk_insert
<input type="checkbox"/> dao	test-dao
<input type="checkbox"/> result-key	insertCnt
<input type="checkbox"/> sql-key	sample01.emp.insert
<input type="checkbox"/> param-bindings	empno=empno_insert ename=ename_insert sal=sal_insert deptno=deptno_insert

- param-count 삭제
- param0, param1, .. 삭제
- list-key 삭제
- is-audit 삭제

SQL

```
insert into emp(empno, ename, sal, deptno)
values(:empno, :ename, :sal, :deptno)
```

수정[Modify]

수정[Modify]

Properties :

Name	Value
<input type="checkbox"/> class	com.poscoict.glueframework.biz.activity.GlueJdbcModify
<input type="checkbox"/> chk-name	chk
<input type="checkbox"/> dao	test-dao
<input type="checkbox"/> result-key	updateCnt
<input type="checkbox"/> sql-key	sample01.emp.update
<input type="checkbox"/> param-bindings	empno=EMPNO sal=SAL ename=ENAME

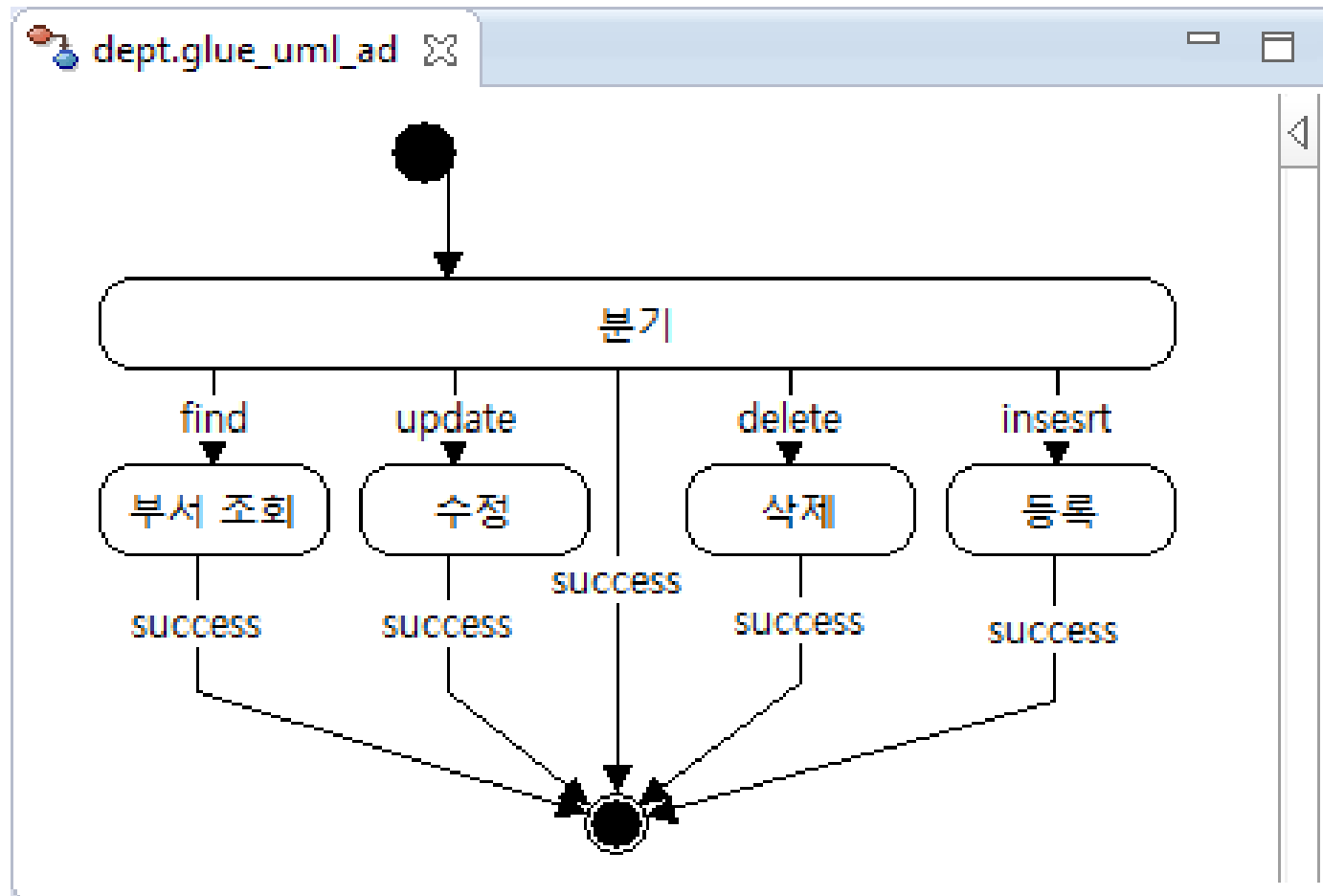
- param-count 삭제
- param0, param1, .. 삭제
- list-key 삭제
- is-audit 삭제

SQL

```
update emp
set sal=:sal, ename=:ename
where empno=:empno
```

# Activity Diagram 그리기 - dept.glue\_uml\_ad

- AD파일 만들기 : Project > model > dept.glue\_uml\_ad
- AD 그리기
  - 모두 custom activity 로 그린다.



Activity name	Properties	
	name	value
분기	class ctx-keys	sample.activity.DeptRouter DeptList cnt
조회	class	sample.activity.DeptSearch
수정	class dao dml-type update-loop update-sql update-params result-key	sample.activity.DeptUpsert test-dao update chk dept.update DNAME,LOC,DEPTNO cnt

Activity name	Properties	
	name	value
등록	class	sample.activity.DeptUpsert
	dao	test-dao
	dml-type	insert
	insert-sql	dept.insert
	insert-params	DEPTNO,DNAME,LOC
	result-key	cnt
삭제	class	sample.activity.DeptDelete
	dao	test-dao
	select-sql	sample.emp.select
	delete-sql	dept.delete
	param-key	DEPTNO

- 파일위치 : GlueSDK/config/glue-config.xml
- Generate Service 실행 시 자동 생성 제외되는 class
  - Reusable Activity의 class들
  - **exclude-package** 에 지정된 package의 하위 class들

The screenshot shows the XML configuration for the Glue Framework. The `<class-generator>` element is expanded to show two `<exclude-package>` entries: `com.poscoict.glueframework` and `sample.activity`. A blue box highlights the text: "Sample.activity 하위 class들을 자동으로 생성되게 하기 위해 exclude-package 를 주석처리 한다." (To prevent automatic generation of sub-classes of sample.activity, the exclude-package is commented out). A red box highlights the two `<exclude-package>` lines, with a red arrow pointing to the text: "변경된 내용은 Eclipse 재기동해야 반영됨" (Changes are reflected after restarting Eclipse). The IDE interface includes a "Design" and "Source" tab at the bottom.

```
<?xml version="1.0" encoding="UTF-8"?>
<glue-eclipse-config
  xmlns="http://www.poscoict.com/glueframework/plugin/config"
  xmlns:xsi="http://www...
  xsi:schemaLocation="h...
  <activities>
  <common>
    <document>
    <service>
    <query>
    <class-generator>
      <exclude-package>com.poscoict.glueframework</exclude-package>
      <exclude-package>sample.activity</exclude-package>
    </class-generator>
    <activity-template>
  </common>
</glue-eclipse-config>
```

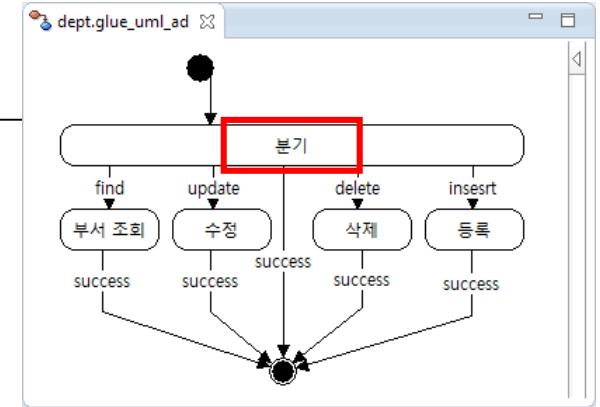
- sample.activity.DeptRouter (1/2)

```
package sample.activity;
```

```
import java.util.*;
```

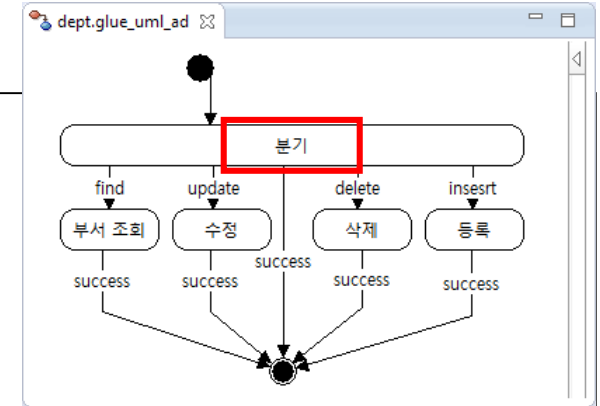
```
import com.poscoict.glueframework.biz.activity.*;  
import com.poscoict.glueframework.biz.control.*;  
import com.poscoict.glueframework.context.*;
```

```
public class DeptRouter extends GlueActivity<GlueContext> {  
    public String runActivity( GlueContext ctx ) {  
        /* ResultKey List (reusable activity) */  
        String resultKeyList  
            = this.getProperty( GlueActivityConstants.CTX_KEY_LIST );  
        if ( resultKeyList != null ) {  
            String[] keys = resultKeyList.split( "\\|" );  
            List<String> keyList = new ArrayList<String>( keys.length );  
            for ( int i = 0, iz = keys.length; i < iz; i++ ) {  
                keyList.add( keys[i] );  
            }  
            // GlueJsonController 에서 result-keys 에 지정된 Object 만 리턴한다  
            ctx.put( GlueBizControlConstants.RESULT_KEY_LIST, keyList );  
        }  
    }  
}
```



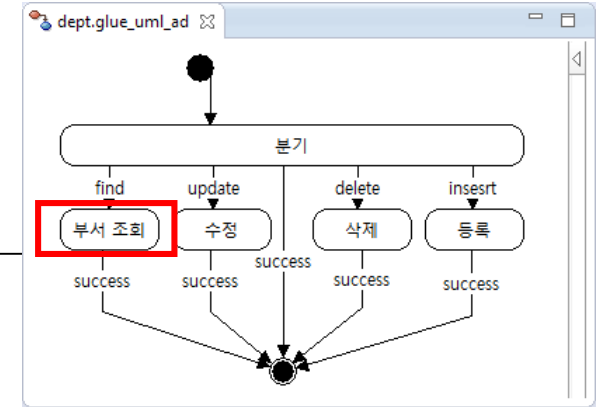
- sample.activity.DeptRouter (2/2)

```
/* Default Router : Key 유무에 의한 분기 */  
if ( ctx.get( "find" )!=null )  
    return "find";  
if ( ctx.get( "update" )!=null )  
    return "update";  
if ( ctx.get( "delete" )!=null )  
    return "delete";  
if ( ctx.get( "insert" )!=null )  
    return "insert";  
/* Value Router : Key 값에 따라 분기 */  
if ( "U".equals(ctx.get( "CRUD" )) )  
    return "update";  
if ( "D".equals(ctx.get( "CRUD" )) )  
    return "delete";  
if ( "C".equals(ctx.get( "CRUD" )) )  
    return "insert";  
return GlueBizControlConstants.SUCCESS;  
}  
}
```





- sample.activity.DeptSearch



```
package sample.activity;
```

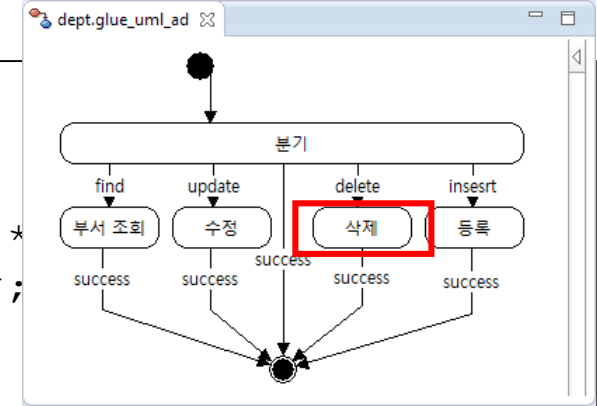
```
import java.util.*;
```

```
import com.poscoict.glueframework.biz.activity.*;  
import com.poscoict.glueframework.biz.control.*;  
import com.poscoict.glueframework.context.*;  
import com.poscoict.glueframework.dao.*;
```

```
public class DeptSearch extends GlueActivity<GlueContext> {  
    public String runActivity(GlueContext ctx) {  
        GlueGenericDao dao = this.getDao("test-dao");  
  
        List<?> result = dao.find("dept.select");  
        ctx.put("DeptList", result);  
  
        return GlueBizControlConstants.SUCCESS;  
    }  
}
```

## • sample.activity.DeptDelete (1/2)

```
package sample.activity;
import java.util.*;
import com.poscoict.glueframework.*;
import com.poscoict.glueframework.biz.activity.*;
import com.poscoict.glueframework.biz.control.*;
import com.poscoict.glueframework.context.*;
import com.poscoict.glueframework.dao.*;
import com.poscoict.glueframework.dao.vo.*;
public class DeptDelete extends GlueActivity<GlueContext> {
    public String runActivity(GlueContext ctx) {
        GlueGenericDao dao = this.getDao( this.getProperty( "dao" ) );
        Object data = ctx.get( this.getProperty( "param-key" ) );
        Object value = data;
        if ( data instanceof Object[] ){
            value = ( (Object[])data ) [0];
        }
        /* select EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, DEPTNO
           from EMP
           where DEPTNO=? */
        List<Object> paramList = new ArrayList<Object>();
        paramList.add( value );
        String selectKey = this.getProperty( "select-sql" );
        List<?> results
            = dao.find( selectKey, new
                GlueParameter<List<Object>>( paramList ) );
```



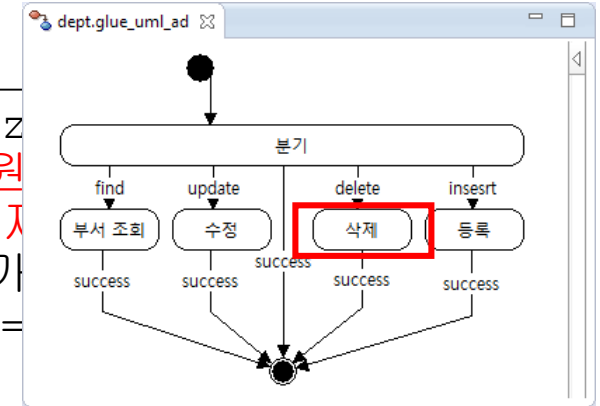
- sample.activity.DeptDelete (2/2)

```

if ( results != null && results.size() > 0 )
    throw new GlueException( "부서원
                                " 부서정보를 삭제
                                /* 또는 cascading 삭제로직 수행 가능
                                delete from emp where deptno=
                                */
}
System.out.println( " 부서를 삭제합니다. " );
/* delete from DEPT
   where DEPTNO=:deptno */
Map<String, Object> paramMap
    = new HashMap<String, Object>();
paramMap.put( "deptno", value );
String deleteKey = this.getProperty( "delete-sql" );
int cnt
    = dao.delete( deleteKey, new
GlueParameter<Map<String, Object>>( paramMap ) );
ctx.put( "cnt", cnt );
System.out.println( " delete count : " + cnt );

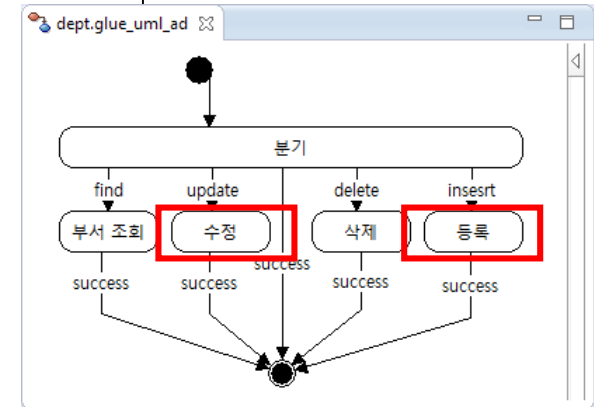
return GlueBizControlConstants.SUCCESS;
}
}

```



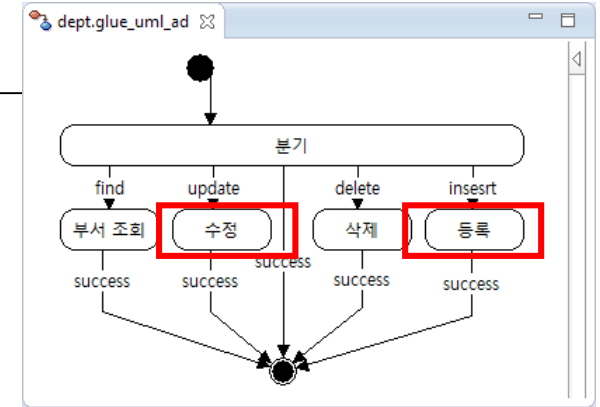
```
package sample.activity;
import java.util.*;
import com.poscoict.glueframework.biz.activity.*;
import com.poscoict.glueframework.biz.control.*;
import com.poscoict.glueframework.context.*;
import com.poscoict.glueframework.dao.*;
import com.poscoict.glueframework.dao.vo.*;
public class DeptUpsert extends GlueActivity<GlueContext> {
    public String runActivity( GlueContext ctx ) {
        GlueGenericDao dao = this.getDao( this.getProperty( "dao" ) );
        String type = this.getProperty( "dml-type" );
        int cnt = 0;
        if ( "insert".equals( type ) ) {
            String[] paramNames = this.getProperty( "insert-params" ).split( "," );
            List<Object> paramList = new ArrayList<Object>();
            for ( String name : paramNames ){
                Object value = ctx.get( name );
                if ( value instanceof Object[] )
                    paramList.add( ( (Object[])value )[0] );
                else
                    paramList.add( value );
            }
            cnt += dao.insert( this.getProperty( "insert-sql" )
                , new GlueParameter<List<Object>>( paramList ) );
            this.logger.info( "{}건이 등록되었습니다", cnt );
        }
    }
}
```

- **sample.activity.DeptUpsert (1/3)**



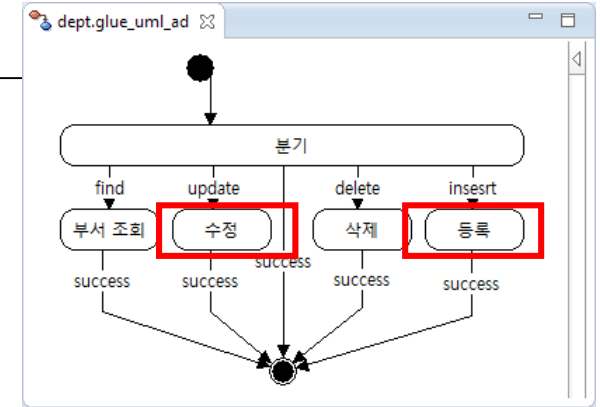
- **sample.activity.DeptUpsert (2/3)**

```
} else if ( "update".equals( type ) ) {  
    String paramInfo  
        = this.getProperty( "update-params" );  
    String[] paramNames = paramInfo.split( "," );  
    String loop = this.getProperty( "update-loop" );  
    if ( ctx.get( loop ) != null ) {  
        this.logger.debug( "수정" );  
        String[] checked = (String[]) ctx.get( loop );  
        for ( String chk : checked ) {  
            int idx = Integer.parseInt( chk );  
            this.logger.debug( "수정 {}", idx+1 );  
            List<Object> paramList = new ArrayList<Object>();  
            for ( String name : paramNames ) {  
                Object value = ctx.get( name );  
                if ( value instanceof Object[] )  
                    paramList.add( ((Object[])value)[idx] );  
                else  
                    paramList.add( value );  
            }  
            cnt += dao.update( this.getProperty( "update-sql" )  
                , new GlueParameter<List<Object>>( paramList ) );  
        }  
        this.logger.info( "{}건이 수정되었습니다", cnt );  
    }  
}
```



- sample.activity.DeptUpsert (3/3)

```
}else {  
    List<Object> paramList  
        = new ArrayList<Object> ();  
    for ( String name : paramNames ) {  
        Object value = ctx.get( name );  
        if ( value instanceof Object[] )  
            paramList.add( ((Object[])value)[0] );  
        else  
            paramList.add( value );  
    }  
    cnt += dao.update( this.getProperty( "update-sql" )  
        , new GlueParameter<List<Object>>( paramList ) );  
    this.logger.info( "{}건이 수정되었습니다", cnt );  
}  
ctx.put( this.getProperty( "result-key" ), cnt );  
}else {  
    this.logger.warn( "{}은 허용되지 않습니다.", type );  
    ctx.put( this.getProperty( "result-key" ), "not supported" );  
}  
ctx.put( this.getProperty( "result-key" ), cnt );  
return GlueBizControlConstants.SUCCESS;  
}  
}
```



- sample.jsp 을 복사하여 sample01.jsp 를 만든다.
- sample01.jsp 파일에서 아래처럼 form action, ServiceName를 수정한다.

```
...
```

```
<form name="form1" method="post" action="sample01.mvc">
```

```
...
```

```
Glue Service Name : <input type="text" name="ServiceName"
```

```
value="sample01-service">
```

```
</form>
```

```
...
```

- dept.jsp (1/2)

```
<%@ page contentType="text/html; charset=UTF-8"%>
<%@ page import="com.poscoict.glueframework.context.GlueContext" %>
<%@ page import="com.poscoict.glueframework.web.control.GlueWebConstants" %>
<%@ page import="java.util.*" %>
<html>
<head>
<title>실습</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<body bgcolor="#FFFFFF" text="#000000" topmargin=10 leftmargin=10>
<form name="form1" method="post" action="dept.mvc">
  <input type="hidden" name="ServiceName" value="dept-service">
  <input type="submit" name="find" value="조회"><hr>
</form>
<form name="form2" method="post" action="dept.mvc">
  <input type="hidden" name="ServiceName" value="dept-service">
  <input type="text" name='DEPTNO' value='' size=2>
  | <input type="text" name='DNAME' value='' size=14>
  | <input type="text" name='LOC' value='' size=13>
  <input type="submit" name="insert" value="등록"><hr>
</form><%
  GlueContext ctx = (GlueContext) request.getAttribute(GlueWebConstants.CONTEXT) ;
  Throwable t = ctx!=null ? ctx.getException() : null;
%><%= t==null ? "" : "[ERROR]" + t.getMessage() %><%
  List<Map> rowSet = ctx!=null ? (List<Map>)ctx.get("DeptList") : null;
%>
```



- dept.jsp (2/2)

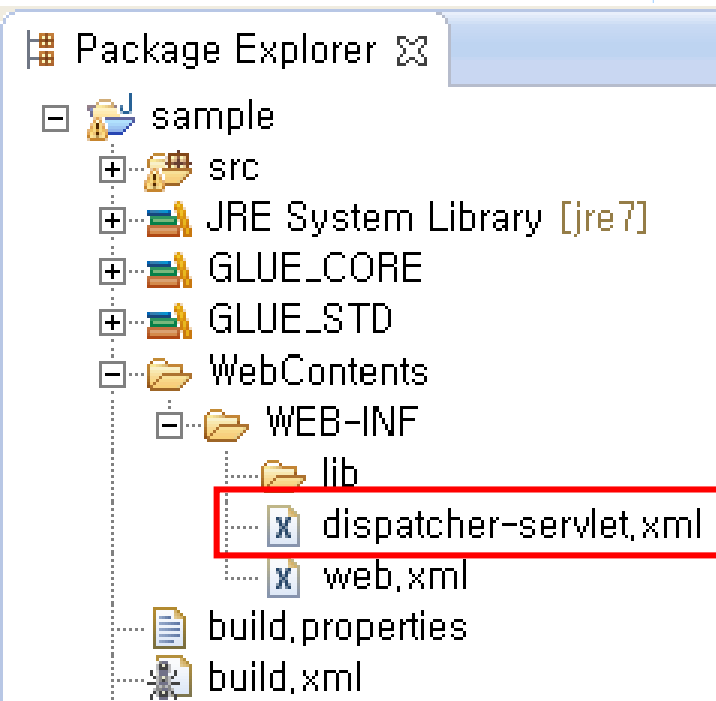
```
<form name="form3" method="post" action="dept.mvc">
  <input type="hidden" name="ServiceName" value="dept-service"><%
    if(rowSet!=null){
      int i=0;
      for ( Map row : rowSet ) {
%>
  <input type="text" name='DEPTNO' value='<%=row.get("DEPTNO") %>' size=2 readonly>
| <input type="text" name='DNAME' value='<%=row.get("DNAME") %>' size=14>
| <input type="text" name='LOC' value='<%=row.get("LOC") %>' size=13>
| <input type="checkbox" name='chk' value='<%=i++%>'><%
      if(i==rowSet.size()){ %>
  <input type="submit" name="update" value="수정"><hr><%
      }else{ %><br><%}
    }
  }
%>
</form>
<form name="form4" method="post" action="dept.mvc">
  <input type="hidden" name="ServiceName" value="dept-service">
  <input type="text" name='DEPTNO' value='' size=2>
  <input type="submit" name="delete" value="삭제">
</form>
</body>
</html>
```

- web.xml 구성 (spring mvc)
  - spring Web MVC Framework 사용시 Dispatcher Servlet 추가.
    - org.springframework.web.servlet.DispatcherServlet

```
<servlet>
  <servlet-name>dispatcher</servlet-name>
  <servlet-class>
    org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>dispatcher</servlet-name>
  <url-pattern>*.mvc</url-pattern>
</servlet-mapping>
```

- web.xml 구성 (spring mvc)
  - Dispatcher Servlet 의 configuration 파일 추가.
    - 파일명 : <servlet-name>-servlet.xml
    - Controller, HandlerMapping, ViewResolver 를 포함함.

```
<bean id="beanNameUrlMapping"  
      class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">  
  <property name="mappings">  
    <value>  
      /*.mvc=controller  
    </value>  
  </property>  
</bean>  
<bean name="controller"  
      class="com.poscoict.glueframework.web.control.spr  
<bean id="viewResolver"  
      class="org.springframework.web.servlet.view.Interr  
  <property name="prefix" value="/springmvc/" />  
  <property name="suffix" value=".jsp" />  
</bean>
```



- build.properties 수정
  - Deploy 위치 지정
    - target.WebContent.dir=%CATALINA\_HOME%/webapps
  - GlueSDK 위치 지정
    - GlueSDK.dir= %ECLIPSE\_HOME%/users/GlueSDK
  - Web Application 이름 지정
    - WEB\_NAME=GlueSample

The screenshot shows the Eclipse IDE interface. On the left, the 'build.properties' file is open, displaying the following content:

```
#-----  
# Directories  
#-----  
WebContent.dir=./WebContents  
build.dir=./build  
class.dir=./classes  
  
target.WebContent.dir=C:/apache-tomcat-7.0.57/webapps  
GlueSDK.dir=C:/eclipse/users/GlueSDK  
WEB_NAME=GlueSample  
  
project_version=1.0.0
```

On the right, the 'Package Explorer' shows the project structure:

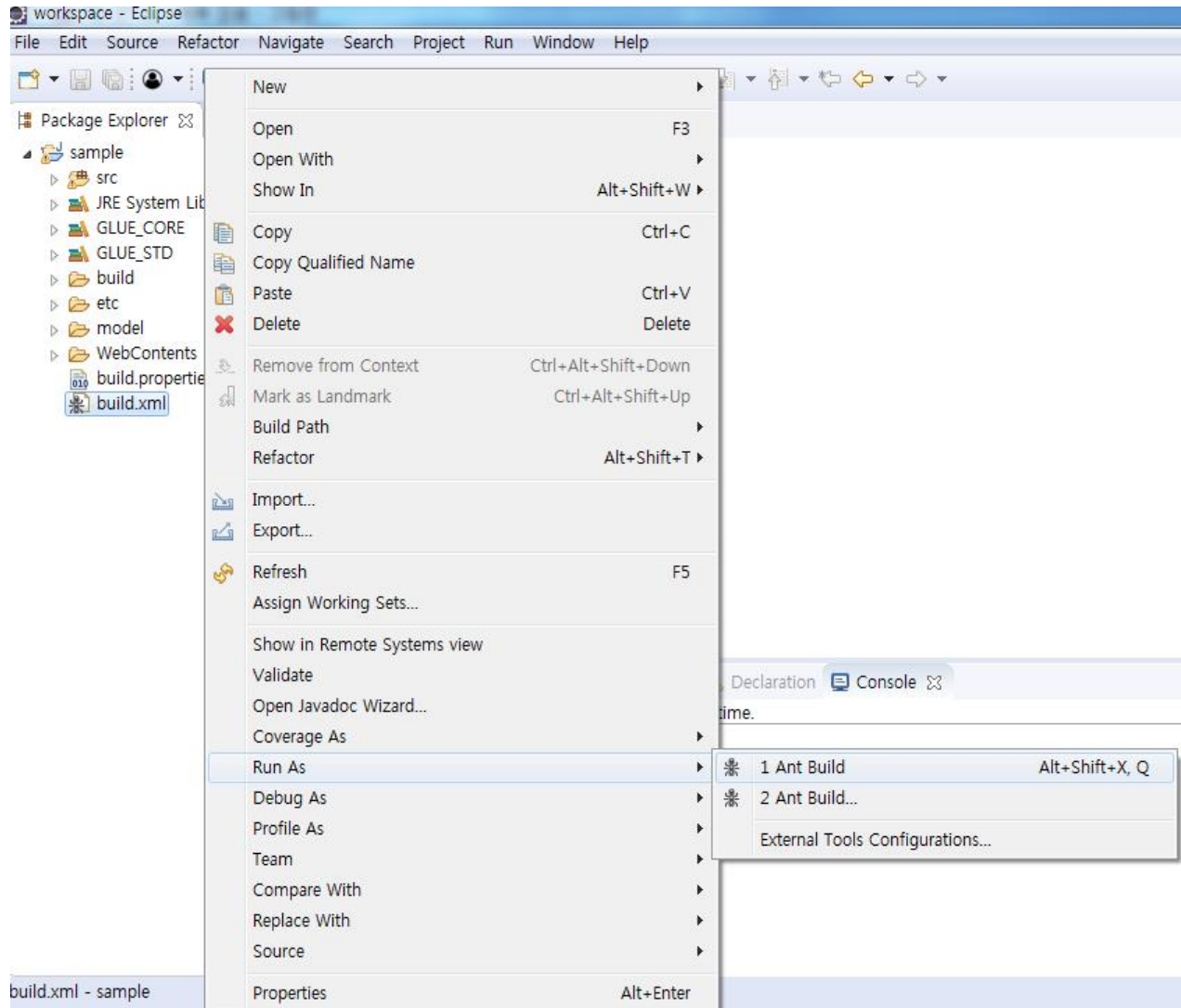
- sample
  - src
  - JRE System Library [JavaSE-1.6]
  - GLUE\_CORE
  - GLUE\_STD
  - model
  - WebContents
    - build.properties
    - build.xml

Red arrows point from the 'build.properties' file to the corresponding paths in the code: 'C:/apache-tomcat-7.0.57/webapps', 'C:/eclipse/users/GlueSDK', and 'GlueSample'. A callout box contains the text: 'Directory 구분자는 슬래시(/) 1개 또는 역슬래시(\) 2개를 사용해야 합니다.'

- build.xml 수정

```
...
<target name="war" description="Creates the WAR file" depends="clean">
...
</target>
<target name="clean">
  <echo>Deleting Build Directory WAR File..</echo>
  <delete failonerror="false">
    <fileset dir="${build.dir}">
      <include name="*.*" />
    </fileset>
    <fileset dir="${target.WebContent.dir}">
      <include name="${WEB_NAME}.war" />
    </fileset>
  </delete>
  <delete dir="${target.WebContent.dir}/${WEB_NAME}" failonerror="false" />
</target>
...
```

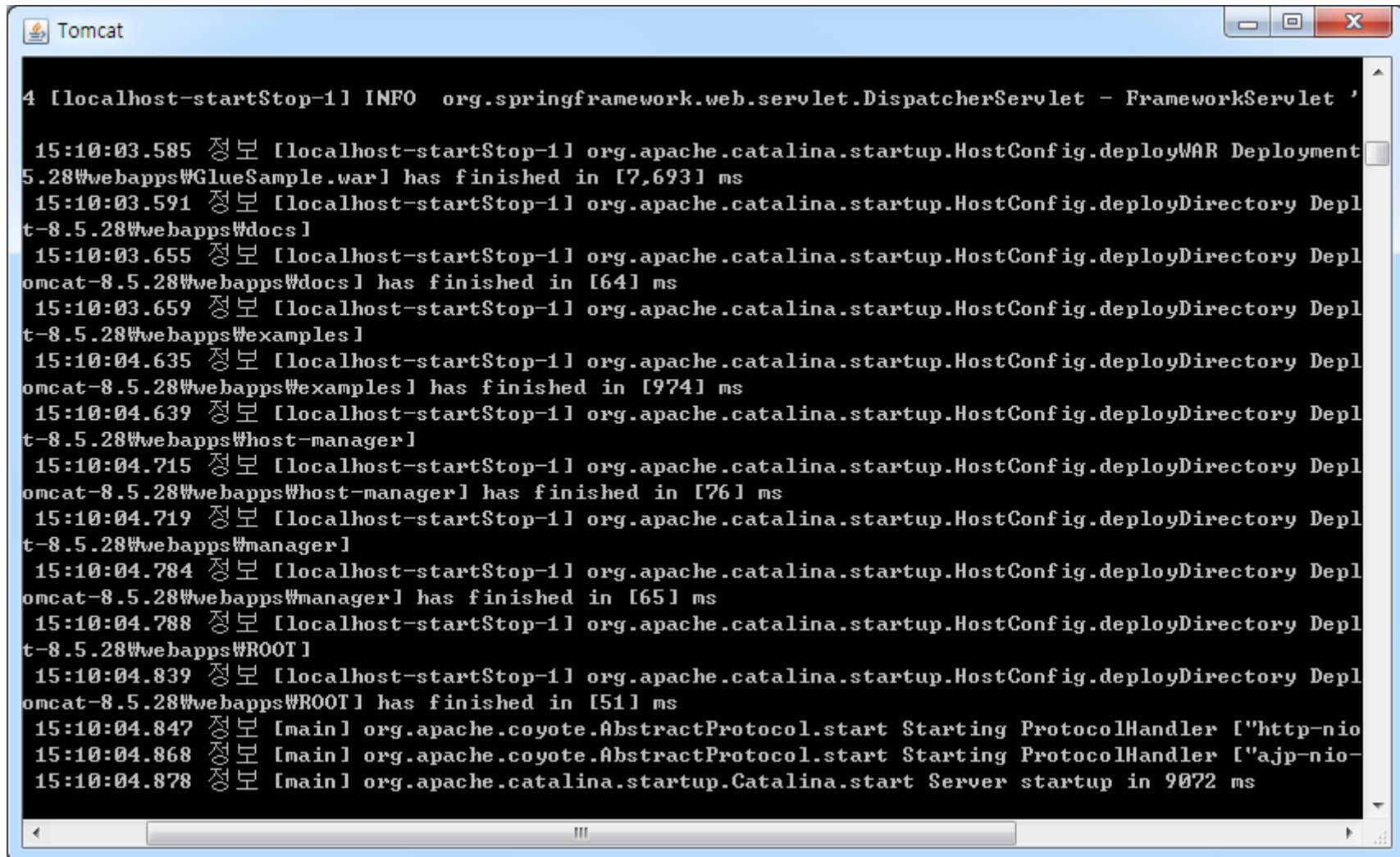
- build.xml 우측마우스 - Run As - 1. Ant Build 실행



- Ant Build 실행 결과
  - packaged war name : GlueSample.war
  - 복사된 위치 : %TOMCAT\_HOME%\webapps

```
Problems @ Javadoc Declaration Console
<terminated> sample build.xml [Ant Build] C:\Java\jdk1.8.0_162\bin\javaw.exe (2018. 3. 19. 오후 2:59:28)
Buildfile: C:\eclipse\workspace\sample\build.xml
clean:
    [echo] Deleting Build Directory WAR File..
    [delete] Deleting directory C:\apache-tomcat-8.5.28\webapps\GlueSample
war:
    [war] Building war: C:\eclipse\workspace\sample\build\GlueSample.war
Copy WAR:
    [copy] Copying 1 file to C:\apache-tomcat-8.5.28\webapps
BUILD SUCCESSFUL
Total time: 3 seconds
```

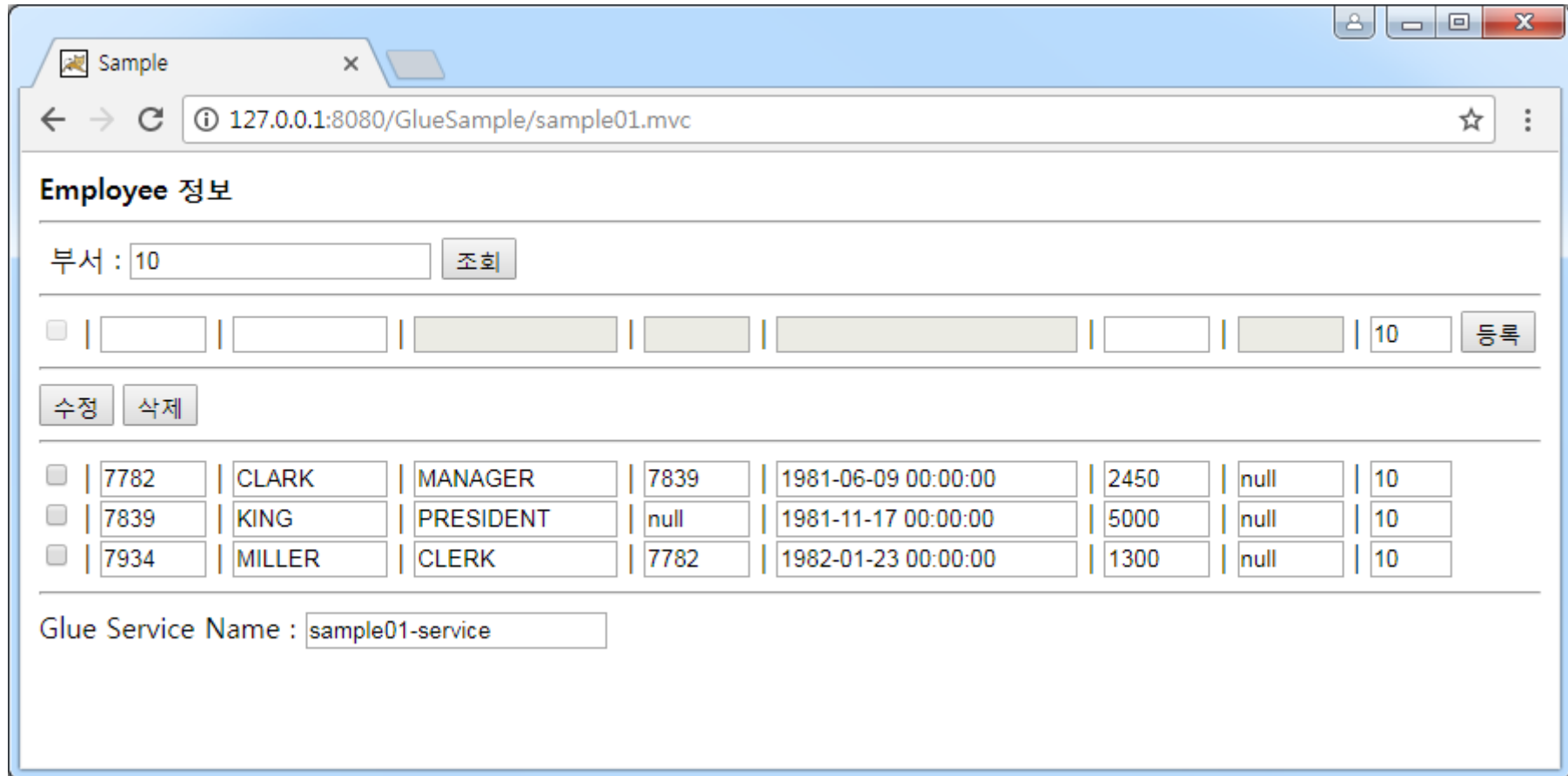
- Tomcat WAS(Web Application Server) Start
  - %TOMCAT\_HOME%\bin\startup.bat 실행



```
Tomcat
4 [localhost-startStop-1] INFO org.springframework.web.servlet.DispatcherServlet - FrameworkServlet '
15:10:03.585 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployWAR Deployment
5.28[webapps\GlueSample.war] has finished in [7,693] ms
15:10:03.591 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deplo
t-8.5.28[webapps\docs]
15:10:03.655 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deplo
omcat-8.5.28[webapps\docs] has finished in [64] ms
15:10:03.659 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deplo
t-8.5.28[webapps\examples]
15:10:04.635 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deplo
omcat-8.5.28[webapps\examples] has finished in [974] ms
15:10:04.639 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deplo
t-8.5.28[webapps\host-manager]
15:10:04.715 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deplo
omcat-8.5.28[webapps\host-manager] has finished in [76] ms
15:10:04.719 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deplo
t-8.5.28[webapps\manager]
15:10:04.784 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deplo
omcat-8.5.28[webapps\manager] has finished in [65] ms
15:10:04.788 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deplo
t-8.5.28[webapps\ROOT]
15:10:04.839 정보 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deplo
omcat-8.5.28[webapps\ROOT] has finished in [51] ms
15:10:04.847 정보 [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ['http-nio
15:10:04.868 정보 [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ['ajp-nio-
15:10:04.878 정보 [main] org.apache.catalina.startup.Catalina.start Server startup in 9072 ms
```



- Browser 로 화면 접속
  - <http://127.0.0.1:8080/GlueSample/sample01.mvc>



- JSON 예제
  - 라이브러리 추가( build.xml )
    - GlueSDK\lib\gluestd\_json2\jackson-annotations-2.6.0.jar
    - GlueSDK\lib\gluestd\_json2\jackson-core-2.6.7.jar
    - GlueSDK\lib\gluestd\_json2\jackson-databind-2.6.7.jar
  - web.xml
  - dispatcher-json-servlet.xml

- **build.xml**

... 생략

```
<lib dir="${GlueSDK.dir}/lib/gluestd">
  <include name="bcprov*.jar"/>
  <include name="xmlbeans*.jar"/>
  <include name="jaxp-api*.jar"/>
  <include name="spring*.jar"/>
  <include name="ehcache*.jar"/>
  <include name="slf4j*.jar"/>
  <include name="logback*.jar"/>
  <include name="jcl*.jar"/>
  <include name="commons*.jar"/>
  <include name="sqlite*.jar"/>
</lib>
<lib dir="${GlueSDK.dir}/lib/gluestd_json2">
  <include name="jackson*.jar"/>
</lib>
```

... 생략

- **web.xml**

```
<web-app . . .>

  <servlet>
    <servlet-name>dispatcher-json</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>dispatcher-json</servlet-name>
    <url-pattern>*.json</url-pattern>
  </servlet-mapping>

</web-app>
```

- **dispatcher-json-servlet.xml (1/2)**

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:dwr="http://www.directwebremoting.org/schema/spring-dwr"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.0.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd
    http://www.directwebremoting.org/schema/spring-dwr
    http://www.directwebremoting.org/schema/spring-dwr-3.0.xsd">

  <bean id="urlMapping"
    class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
    <property name="mappings">
      <value>
        /*.json=controller
      </value>
    </property>
  </bean>
```

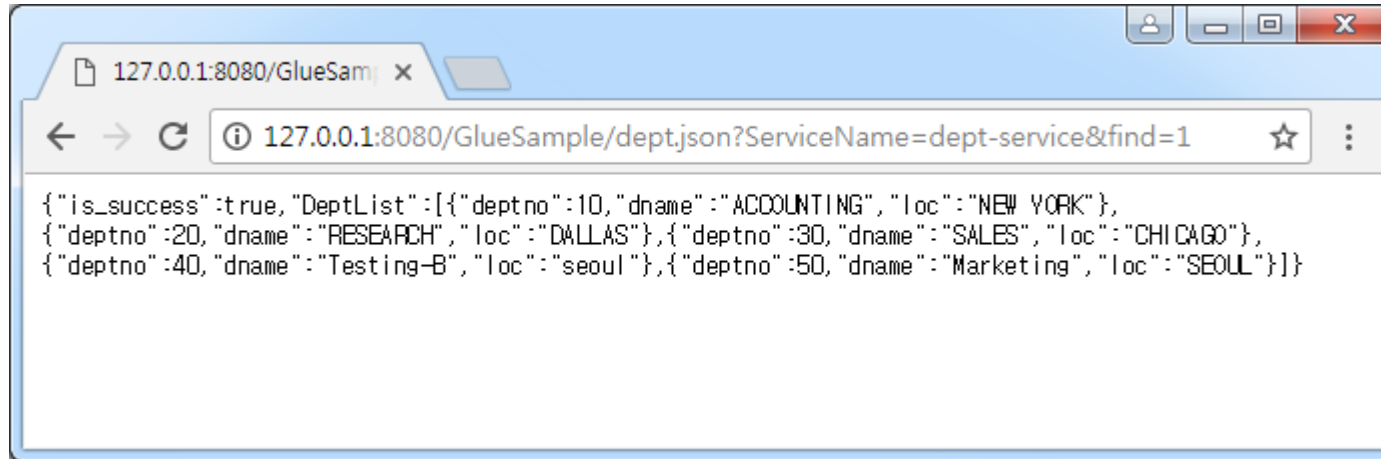
- **dispatcher-json-servlet.xml (2/2)**

```
<bean id="controller"  
      class="com.poscoict.glueframework.web.control.spring.GlueJsonController">  
</bean>  
<bean id="viewResolver"  
      class="org.springframework.web.servlet.view.BeanNameViewResolver"/>  
<bean name="jsonView"  
      class="org.springframework.web.servlet.view.json.MappingJackson2JsonView</beans>
```

- **spring 4.x**에서는 **MappingJacksonJsonView** -> **MappingJackson2JsonView** 변경
  - **spring-webmvc-4.x.x.jar** 에 포함되어 있음

- **Browser로 응답 확인 하기**

- <http://127.0.0.1:8080/GlueSample/dept.mvc>
- <http://127.0.0.1:8080/GlueSample/dept.json?ServiceName=dept-service&find=1>







# IV. Glue Framework 제작 가이드

## 3장 심화 학습

1. EAI interface 하기	.....	122
2. RESTful API 만들기	.....	163
3. WebSocket 사용하기	.....	193
4. Batch Job 만들기	.....	217
5. Glue Job Scheduler 용 User Job 만들기	.....	231

# IV. Glue Framework 제작 가이드

## 3장 심화 학습

### 1. EAI interface 하기



## 목표

- Message Layout을 사용하도록 어플리케이션을 설정할 수 있다.
- Message Layout 사용을 설정할 수 있다
- Message Layout 을 이해할 수 있다.
- 정의된 Layout의 message를 만들 수 있다.
- EAI로 message를 송신할 수 있다.
- 정의된 Layout의 message를 EAI (u-CUBE)로부터 수신할 수 있다.
- 수신한 message를 parsing 하여 사용할 수 있다.
- Sub service를 호출할 수 있다.

## 실습

- applicationContext.xml 설정하기
- Message Layout 파일 만들기
  - MSGFW001-msg.xml
  - MSGFW002-msg.xml
- TC 송신 Service 만들기
  - MSGFW001.glue\_uml\_ad
  - MSGFW001-service.xml
- TC 수신 Service 만들기
  - MSGFW003.glue\_uml\_ad
  - MSGFW004.glue\_uml\_ad
  - MSGFW003-service.xml
  - MSGFW004-service.xml

```
applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/beans
5         http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
6     <bean id="serviceManager" class="com.poscoict.glueframework.biz.control.GlueServiceImpl" lazy-init="true">
10    <bean id="cacheManager" class="com.poscoict.glueframework.cache.ehcache.GlueEhCacheManager"/>
11    <bean id="serviceLoader" class="com.poscoict.glueframework.biz.control.GlueServiceLoader"/>
12    <bean id="test-dao" class="com.poscoict.glueframework.dao.jdbc.GlueJdbcDao">
16    <bean id="test-tx" class="com.poscoict.glueframework.transaction.GlueDataSourceTransactionManager">
19    <bean id="test-datasource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
27    <bean id="queryManager" class="com.poscoict.glueframework.dao.manager.GlueQueryManagerImpl">
31    <bean id="queryLoader" class="com.poscoict.glueframework.dao.manager.GlueQueryLoader"/>
32
33    <bean id="layoutManager" class="com.poscoict.glueframework.message.layout.GlueXmlMessageLayout">
34        <property name="cacheManager" ref="cacheManager"/>
35    </bean>
36 </beans>
```

- 요건정의

송신	수신	인터페이스	... ..
시스템#1	시스템#2	TC : 부서정보	...
시스템#1	시스템#3	FILE : 사원정보	...
시스템#2	시스템#4	TC : 부서관리자정보	...
...	...	...	...

- Message Layout 정의

부서정보 : MSGFW001				
항목명	항목ID	Data Type	길이	비고
CRUD구분	CRUD	String	1	처리타입 정의, - C:추가, S:송신, U:수정,D:삭제
부서번호	DEPTNO	Number	2	...
부서이름	DNAME	String	14	...
부서위치	LOC	String	13	...

- TC(전문)의 Data 구성을 정의한 파일, Interface Layout
- TC를 parsing, 생성할 때 기준 정보로 사용 됨

- TC별로 작성

- 작명 규칙

- TCID-msg.xml

• TC : Transaction Code 의 약자지만, 약속된 layout을 가진 text message 를 뜻하기도 한다.

- 애플리케이션 상의 위치

- layout \

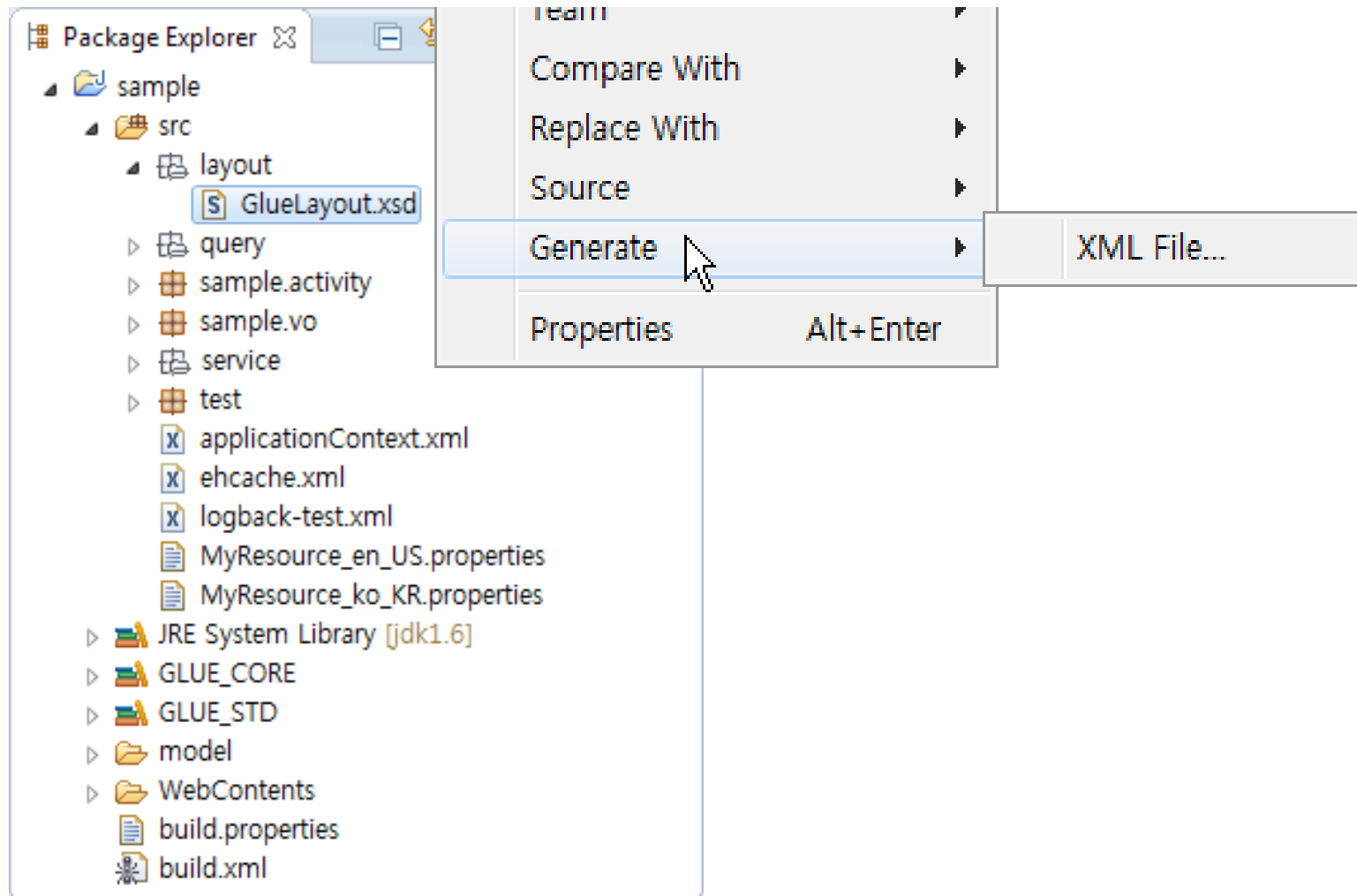
- GlueSample.war! \WEB-INF \ classes \ layout
- user-application.jar! \ layout

- Glue Project 상의 위치

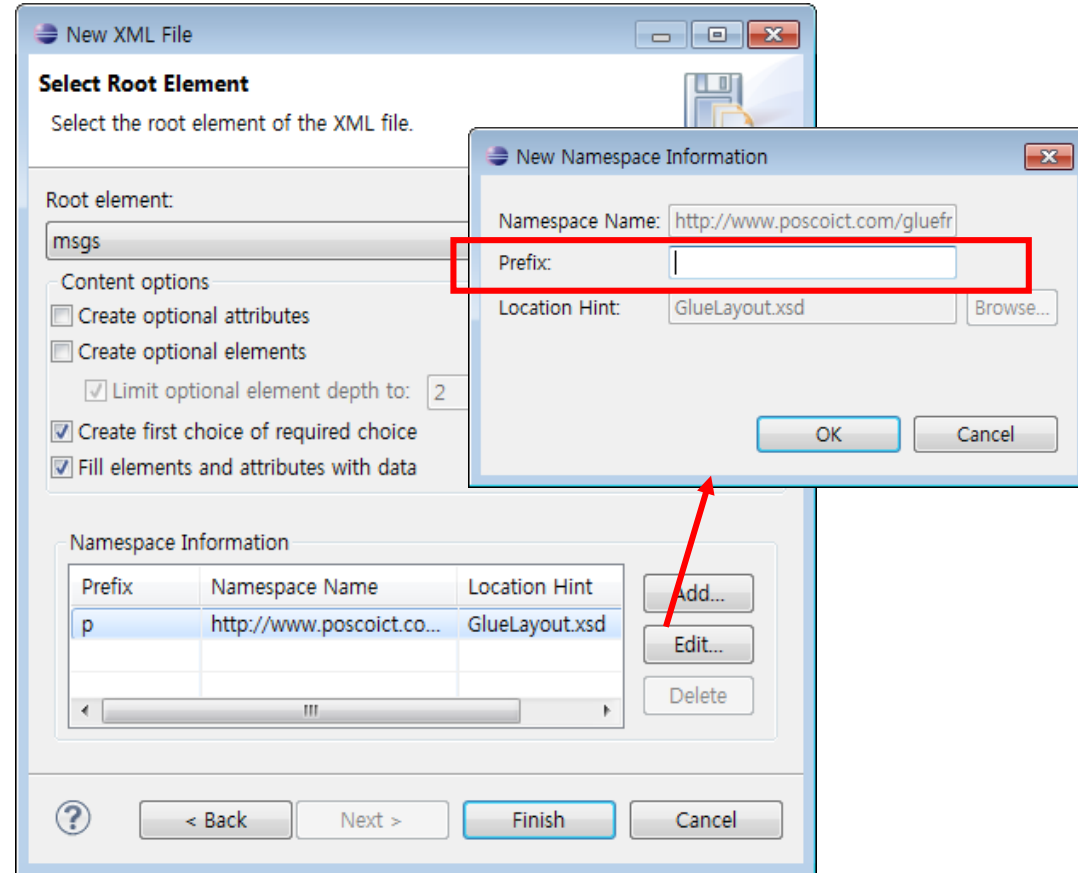
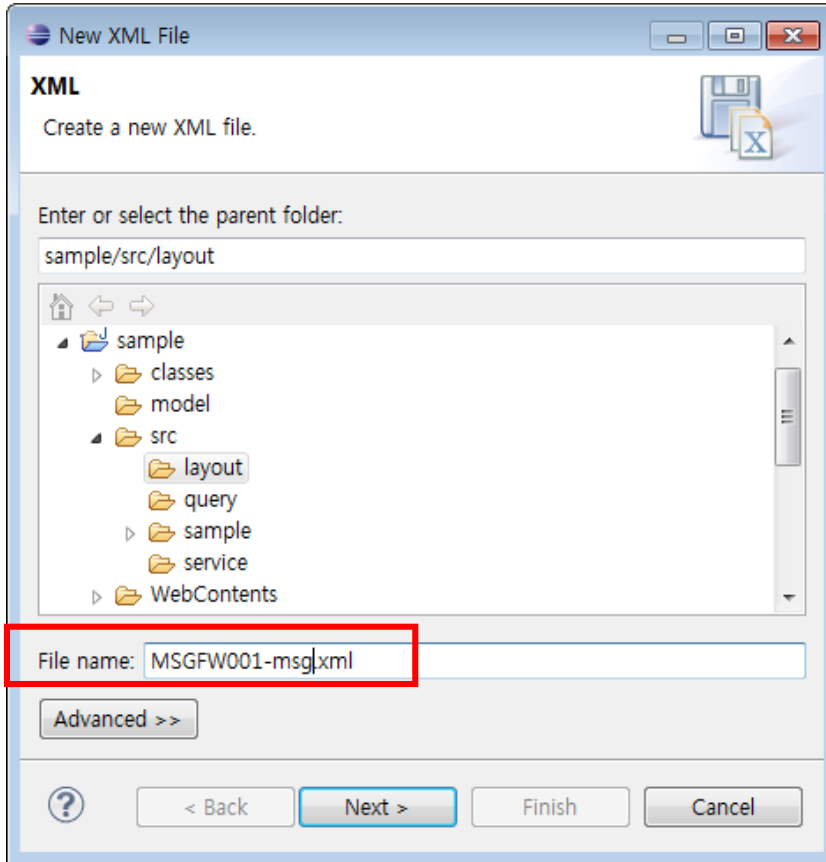
- src \ layout

- <sample> \ src \ layout
- <sample-maven> \ src \ main \ resource \ layout

- XML Schema를 통해 XML를 생성함.
- GlueLayout.xsd > 우클릭 > Generate > XML File



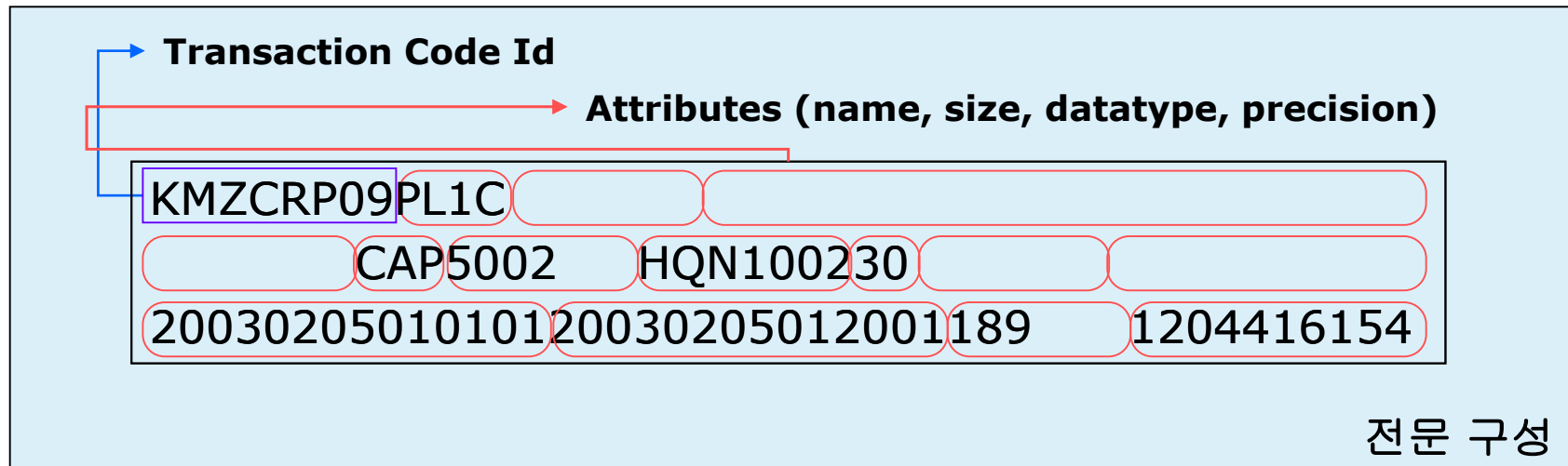
- TC id를 파일 이름으로 입력 : MSGFW001-msg.xml
- Prefix 는 삭제.





- msg
  - id : Message Layout에 대한 TC ID를 의미  
ID는 파일명에 사용됨
  - name : Message 명 및 Message 설명을 정의
- attributes
  - type : 항목이 일반 항목인지 그룹항목인지 여부를 정의
    - E : Element, G : Group , GE : Elements of a Group
  - seq : 항목의 순서를 정의
  - id : 항목ID
  - name : 항목명
  - datatype : 항목의 데이터 타입
    - STRING, NUMBER, DATE, ARRAY
  - length : 항목의 총 길이.  
Type이 G인 경우는 반복 회수를 의미
  - precision : datatype이 NUMBER type일 경우 소수점 자리수

- Layout Manager 종류
  - XML Layout Manager
  - DB Layout Manager
- 관련 Reusable Activity
  - MessageParse : String 을 GlueMessage객체로
  - MessageCreate : GlueMessage객체를 String으로
- TC 예제



- MSGFW001-msg.xml : 일반 항목 사용
  - TRANSACTION\_CODE
  - CRUD
  - DEPTNO
  - DNAME
  - LOC
  
- MSGFW002-msg.xml : 그룹 항목 사용
  - TRANSACTION\_CODE
  - CRUD
  - DEPTNO
  - DNAME
  - LOC
  - EMPNO : 그룹 항목(3회 반복)
  - ENAME : 그룹 항목(3회 반복)
  - ETC

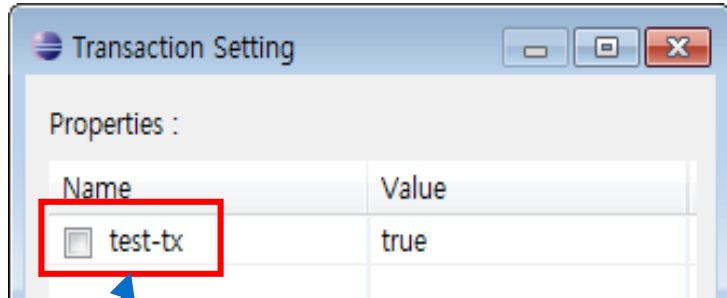
- MSGFW001-msg.xml : 일반 항목 사용

```
<?xml version="1.0" encoding="UTF-8"?>
<msgs ...>
<!-- 아래 msg 부분만 복사 -->
  <msg id="MSGFW001" name="DEPT">
    <attribute type="E" seq="1" id="TRANSACTION_CODE" name="TransactionCode"
      datatype="STRING" length="8" />
    <attribute type="E" seq="2" id="CRUD" name="CRUD" datatype="STRING" length="1" />
    <attribute type="E" seq="3" id="DEPTNO" name="부서번호"
datatype="NUMBER" length="2" precision="0" />
    <attribute type="E" seq="4" id="DNAME" name="부서명" datatype="STRING" length="14" />
    <attribute type="E" seq="5" id="LOC" name="부서위치" datatype="STRING" length="13" />
  </msg>
</msgs>
```

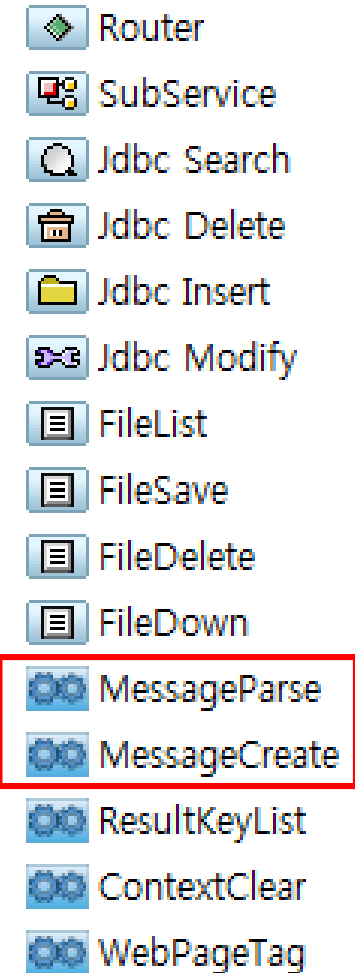
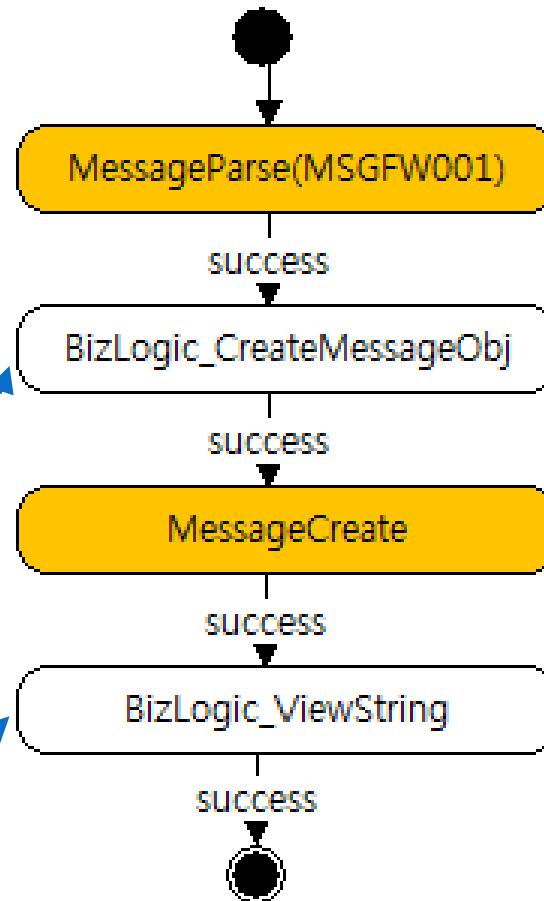
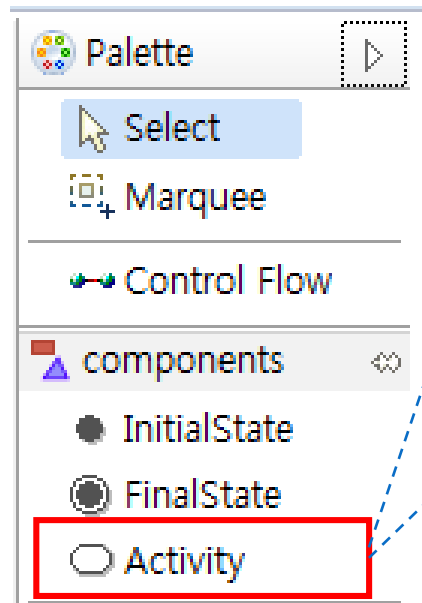
- MSGFW002-msg.xml : 그룹 항목 사용

```
<?xml version="1.0" encoding="UTF-8"?>
<msgs ...>
<!-- 아래 msg 부분만 복사 -->
  <msg id="MSGFW002" name="DEPT">
    <attribute type="E" seq="1" id="TRANSACTION_CODE" name="TransactionCode"
      datatype="STRING" length="8" />
    <attribute type="E" seq="2" id="CRUD" name="CRUD" datatype="STRING" length="1" />
    <attribute type="E" seq="3" id="DEPTNO" name="부서번호"
      datatype="NUMBER" length="2" precision="0" />
    <attribute type="E" seq="4" id="DNAME" name="부서명" datatype="STRING" length="14" />
    <attribute type="E" seq="5" id="LOC" name="부서위치" datatype="STRING" length="13" />
    <attribute type="G" seq="6" id="EMP_SET" name="부서소속 사원"
      datatype="ARRAY" length="3" />
    <attribute type="GE" seq="7" id="EMPNO" name="사원번호"
      datatype="NUMBER" length="4" precision="0" />
    <attribute type="GE" seq="8" id="ENAME" name="이름" datatype="STRING" length="10" />
    <attribute type="E" seq="9" id="ETC" name="기타" datatype="STRING" length="1" />
  </msg>
</msgs>
```

- MSGFW001.glue\_uml\_ad



- Transaction manager 설정



- MSGFW001.glue\_uml\_ad



MessageParse(MSGFW001)

Properties :

Name	Value
<input type="checkbox"/> class	com.poscoict.glueframework.biz.activity.GlueMessageParse
<input type="checkbox"/> layout	<b>layoutManager</b>

- applicationContext에서 정의

```

MSGFW001R50dname:abc      dloc:seoul
MSGFW001D50
MSGFW001C50dname:abc      dloc:seoul
MSGFW001U50dname:ABC      dloc:SEOUL
  
```

- MSGFW001.glue\_uml\_ad



MessageCreate

Properties :

Name	Value
<input type="checkbox"/> class	com.poscoict.glueframework.biz.activity.GlueMessageCreate
<input type="checkbox"/> layout	layoutManager
<input type="checkbox"/> message-key	messageObj
<input type="checkbox"/> result-key	stringObj



- MSGFW001.glue\_uml\_ad



BizLogic\_CreateMessageObj

Properties :

Name	Value
<input type="checkbox"/> class	sample.activity.GetEmp

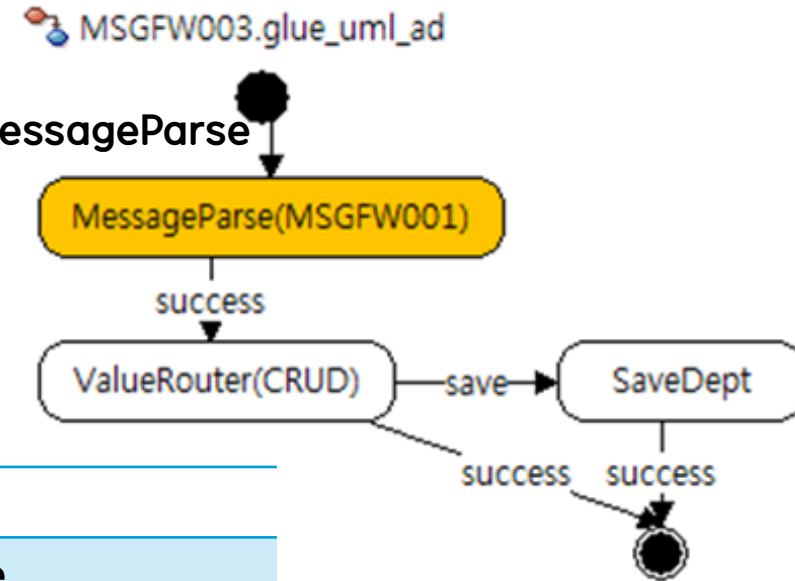
BizLogic\_ViewString

Properties :

Name	Value
<input type="checkbox"/> class	sample.activity.ViewString

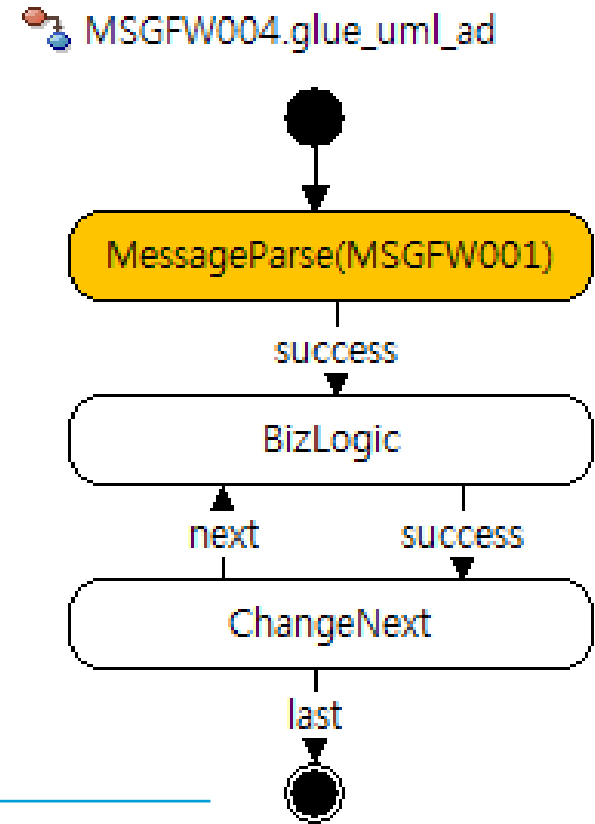
- sample.activity.GetEmp
- sample.activity.ViewString

- MSGFW003.glue\_uml\_ad
  - com.poscoict.glueframework.biz.activity.GlueMessageParse
  - sample.activity.ValueRouter
  - sample.activity.SaveDept
  - transaction 설정 (test-tx : true)



Activity name	Properties	
	name	value
MessageParse (MSGFW001)	layout message-id	layoutManager MSGFW001
SaveDept	class dao insert-sql insert-param update-sql update-param delete-sql delete-param	sample.activity.SaveDept test-dao dept.insert DEPTNO,DNAME,LOC dept.update DNAME,LOC,DEPTNO dept.delete deptno=DEPTNO

- MSGFW004.glue\_uml\_ad
  - MessageParse 의 경우, MSGFW003 AD 와 같게 설정
  - sample.activity.BizLogic
  - sample.activity.ChangeNext



## Activity name

## Properties

name

value

BizLogic

class

sample.activity.BizLogic

ChangeNext

class

sample.activity.ChangeNext

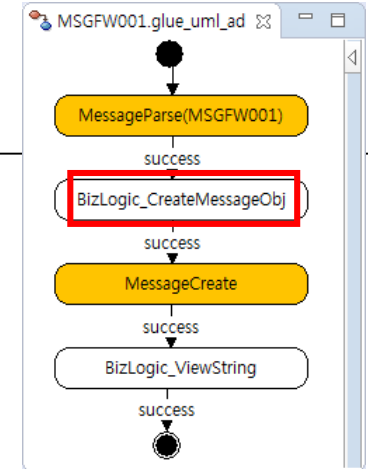
- **sample.activity.GetEmp (1/3)**

```
package sample.activity;
import java.util.*;
import com.poscoict.glueframework.biz.activity.*;
import com.poscoict.glueframework.biz.control.*;
import com.poscoict.glueframework.context.*;
import com.poscoict.glueframework.message.*;
public class GetEmp extends GlueActivity<GlueContext> {
    public String runActivity( GlueContext ctx ) {
        GlueMessage message = ctx.getMessage ();

        GlueMessage messageObj = new GlueMESMessageImpl ();
        messageObj.setTCID ( "MSGFW002" );
        messageObj.setObject ( "TRANSACTION_CODE", "MSGFW002" );
        messageObj.setObject ( "CRUD", message.get ( "CRUD" ) );
        messageObj.setObject ( "DEPTNO", message.get ( "DEPTNO" ) );
        messageObj.setObject ( "DNAME", message.get ( "DNAME" ) );
        messageObj.setObject ( "LOC", message.get ( "DLOC" ) );

        List<Object> empno = new ArrayList<Object>();
        empno.add( 1101 );
        empno.add( 1102 );
        empno.add( 1103 );

        messageObj.setObject ( "EMPNO", empno );
    }
}
```



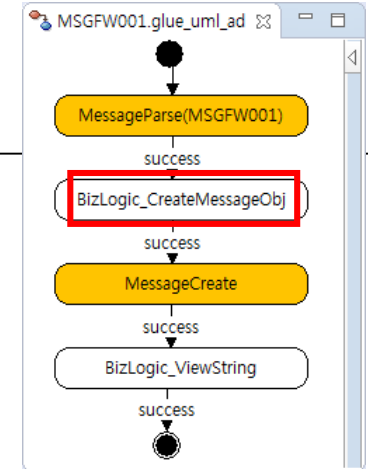
- **sample.activity.GetEmp (2/3)**

```
List<Object> ename = new ArrayList<Object>();
ename.add( "name#1" );
ename.add( "name#2" );
ename.add( "name#3" );
```

```
messageObj.setObject( "ENAME", ename );
ctx.put( "messageObj", messageObj );
```

```
/*
```

```
List<GlueMessage> messageList = new ArrayList<GlueMessage>();
GlueGenericDao dao = this.getDao( "test-dao" );
List<Object> paramList = new ArrayList<Object>();
paramList.add( message.get( "DEPTNO" ) );
GlueParameter<List<Object>> param = new GlueParameter<List<Object>>();
param.setParameter( paramList );
List<Map<String, Object>> rowSet = dao.find( "sample01.emp.select" , param );
for(int cnt=rowSet.size(), i=0; i<cnt; ){
    GlueMessage msg = new GlueMESMessageImpl();
    msg.setTCID( "MSGFW002" );
    msg.setObject( "TRANSACTION_CODE", "MSGFW002" );
    msg.setObject( "CRUD", message.get( "CRUD" ) );
    msg.setObject( "DEPTNO", message.get( "DEPTNO" ) );
    msg.setObject( "DNAME", message.get( "DNAME" ) );
    msg.setObject( "LOC", message.get( "DLOC" ) );
```



- **sample.activity.GetEmp (3/3)**

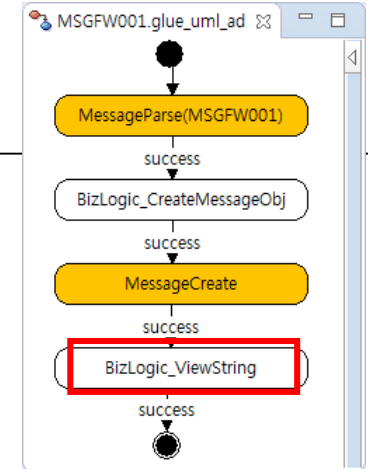
```
List<Object> empnoList = new ArrayList<Object>();
List<Object> enameList = new ArrayList<Object>();
for(int j=0; j<3; j++, i++){
    if(i<cnt){
        Map<String, Object> row = rowSet.get( i );
        empnoList.add( row.get( "EMPNO" ) );
        enameList.add( row.get( "ENAME" ) );
    }else{
        empnoList.add( "" );
        enameList.add( "" );
    }
}
msg.setObject( "EMPNO", empnoList );
msg.setObject( "ENAME", enameList );
messageList.add( msg );
}
ctx.put( "messageObj", messageList );
*/
return GlueBizControlConstants.SUCCESS;
}
}
```



- **sample.activity.ViewString**

```
package sample.activity;
import java.util.*;
import com.poscoict.glueframework.biz.activity.*;
import com.poscoict.glueframework.biz.control.*;
import com.poscoict.glueframework.context.*;
public class ViewString extends GlueActivity<GlueContext> {
    public String runActivity( GlueContext ctx ) {
        Object send = ctx.get( "stringObj" );

        if ( send instanceof Map<?, ?> ) {
            Map<String, List<String>> sendMap = (Map<String, List<String>>) send;
            for ( String key : sendMap.keySet() ) {
                List<String> sendList = sendMap.get( key );
                for ( String tc : sendList ) {
                    System.out.println( key + "["+tc+"]" );
                }
            }
        }
        return GlueBizControlConstants.SUCCESS;
    }
}
```

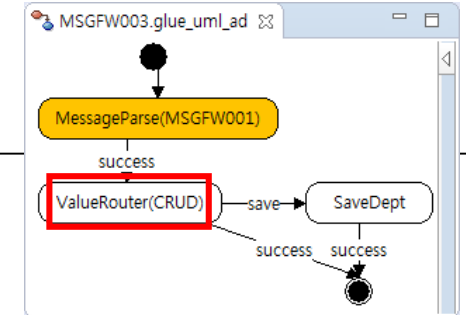


- sample.activity.ValueRouter

```
package sample.activity;

import com.poscoict.glueframework.biz.activity.*;
import com.poscoict.glueframework.biz.control.*;
import com.poscoict.glueframework.context.*;
import com.poscoict.glueframework.message.*;

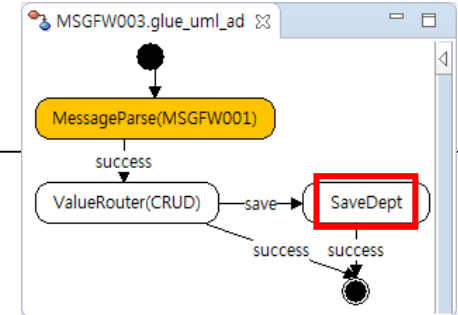
public class ValueRouter extends GlueActivity<GlueContext> {
    public String runActivity( GlueContext ctx ) {
        /* Value Router : 값에 따른 분기 */
        GlueMessage message = ctx.getMessage();
        if ( message != null ) {
            String crud = (String) message.get( "CRUD" );
            String crud2 = (String) ctx.get( "CRUD" );
            this.logger.info( "{},{}", crud, crud2 );
            if ( "C".equals( crud ) || "U".equals( crud ) || "D".equals( crud ) ){
                return "save";
            } else if ( "S".equals( crud ) ) {
                return "send"; /* ERROR */
            }
        }
        return GlueBizControlConstants.SUCCESS;
    }
}
```





- **sample.activity.SaveDept (1/2)**

```
package sample.activity;
import java.util.*;
import com.poscoict.glueframework.biz.activity.*;
import com.poscoict.glueframework.biz.control.*;
import com.poscoict.glueframework.context.*;
import com.poscoict.glueframework.dao.*;
import com.poscoict.glueframework.dao.vo.*;
import com.poscoict.glueframework.message.*;
public class SaveDept extends GlueActivity<GlueContext> {
    public String runActivity(GlueContext ctx) {
        GlueGenericDao dao = this.getDao( this.getProperty( GlueActivityConstants.DAO ) );
        GlueMessage msg = ctx.getMessage();
        String crud = (String)msg.get( "CRUD" );
        if(crud.equals( "C" )) {
            String paramInfo = this.getProperty( "insert-param" );
            String[] paramNames = paramInfo.split( "," );
            List<Object> paramList = new ArrayList<Object>();
            for ( String name : paramNames ) {
                paramList.add( msg.get( name ) );
            }
            GlueParameter<List<Object>> param
                = new GlueParameter<List<Object>>( paramList );
            int cnt = dao.insert( this.getProperty( "insert-sql" ), param );
            this.logger.info( "{}건이 추가 되었습니다.", cnt );
        }
    }
}
```

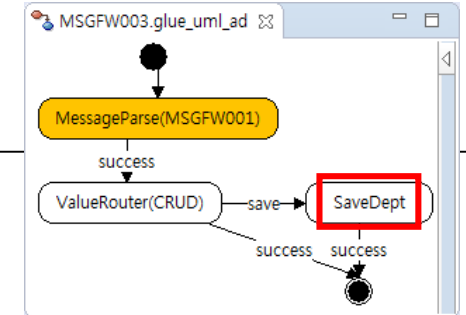


- sample.activity.SaveDept (2/2)

```

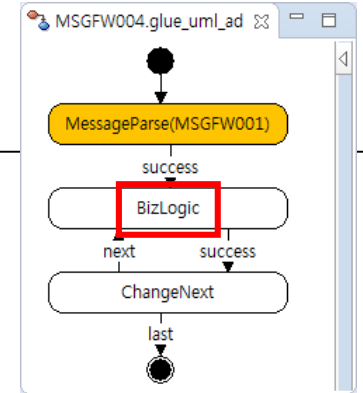
}else if(crud.equals( "U" )){
    String paramInfo = this.getProperty( "update-param" );
    String[] paramNames = paramInfo.split( "," );
    List<Object> paramList = new ArrayList<Object>();
    for ( String name : paramNames ) {
        paramList.add( msg.get( name ) );
    }
    int cnt = dao.update( this.getProperty( "update-sql" ) ,
        new GlueParameter<List<Object>>( paramList ));
    this.logger.info( "{}건이 수정 되었습니다.", cnt );
}else if(crud.equals( "D" )){
    String paramInfo = this.getProperty( "delete-param" );
    String[] paramNames = paramInfo.split( "," );
    Map<String, Object> paramMap = new HashMap<String, Object> ();
    for ( String nameInfo : paramNames ) {
        String[] name = nameInfo.split( "=" );
        paramMap.put( name[0], msg.get( name[1] ) );
    }
    int cnt = dao.delete( this.getProperty( "delete-sql" ) ,
        new GlueParameter<Map<String, Object>>( paramMap ));
    this.logger.info( "{}건이 삭제 되었습니다.", cnt );
}
return GlueBizControlConstants.SUCCESS;
}
}

```



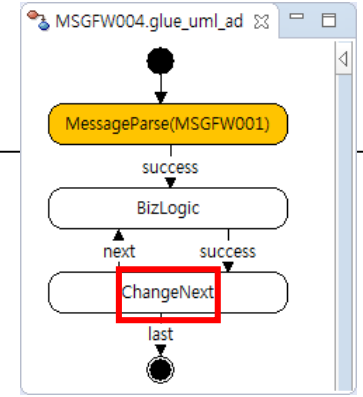
- **sample.activity.BizLogic**

```
package sample.activity;
import java.util.*;
import com.poscoict.glueframework.biz.activity.*;
import com.poscoict.glueframework.biz.control.*;
import com.poscoict.glueframework.context.*;
import com.poscoict.glueframework.message.*;
public class BizLogic extends GlueActivity<GlueContext> {
    public String runActivity( GlueContext ctx ) {
        List<GlueMessage> messageList = ctx.getMessages();
        this.logger.info( "수신 : [{}]", ctx.get( GlueBizControlConstants.RECEIVE_TC ) );
        this.logger.info( "Parsing TC 개수 : {}", messageList.size() );
        int idx = ctx.get( "tc-idx" ) != null
            ? ( (Integer) ctx.get( "tc-idx" ) ).intValue()
            : 0;
        GlueMessage message = messageList.get( idx );
        this.logger.info( "TC : [{}]", message.getTC() );
        /* SubService 실행 */
        String mainServiceName = (String)ctx.get( GlueBizControlConstants.SERVICE_NAME );
        ctx.putAll( ctx.getMessages().get( idx ).getAttributes() );
        ctx.put( GlueBizControlConstants.SERVICE_NAME, "dept-service" );
        GlueBizProvider.getController().doSubController( ctx, true );
        ctx.put( GlueBizControlConstants.SERVICE_NAME, mainServiceName );
        return GlueBizControlConstants.SUCCESS;
    }
}
```



- **sample.activity.ChangeNext**

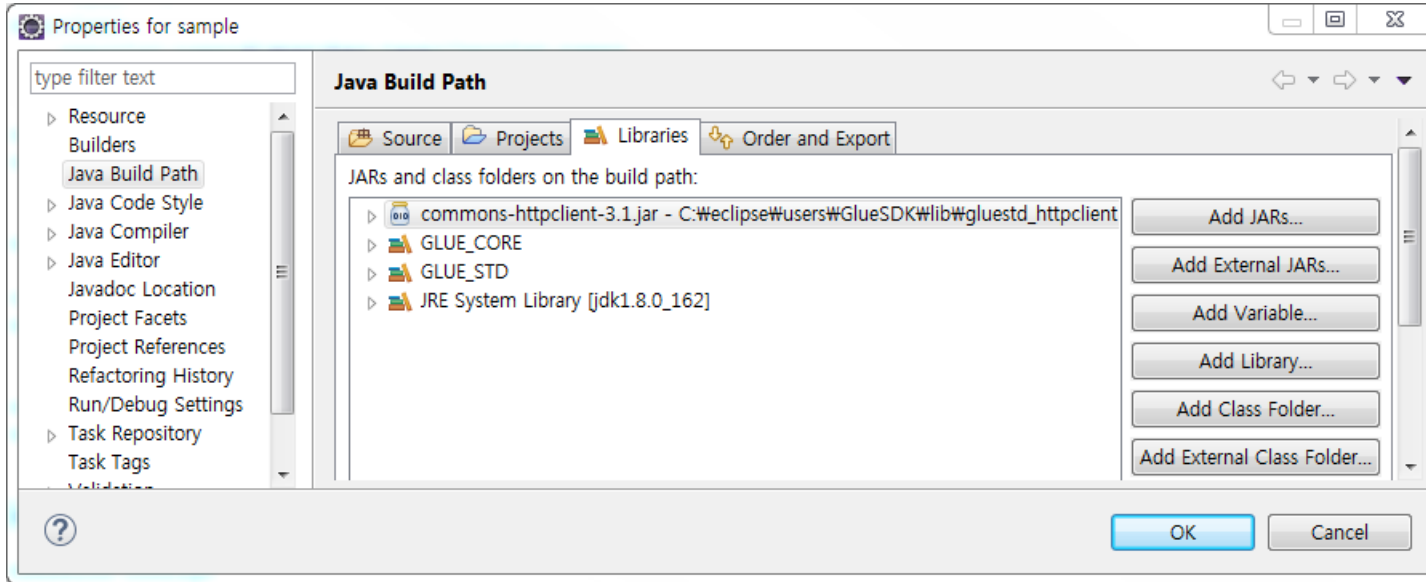
```
package sample.activity;
import com.poscoict.glueframework.biz.activity.*;
import com.poscoict.glueframework.context.*;
public class ChangeNext extends GlueActivity<GlueContext> {
    public String runActivity( GlueContext ctx ) {
        Object idx = ctx.get( "tc-idx" );
        if ( idx == null ){
            idx = new Integer( 0 );
        }
        int total = ctx.getMessage().size();
        int current = ( (Integer) idx ).intValue() + 1;
        ctx.put( "tc-idx", new Integer( current ) );
        if ( current == total )
        {
            return "last";
        } else
        {
            return "next";
        }
    }
}
```



- web.xml 설정
- HttpReceiver Servlet 추가.
  - com.poscoict.glueframework.web.GlueHttpReceiverAdapter

```
<servlet>
  <servlet-name>HttpReceiver</servlet-name>
  <servlet-class>
    com.poscoict.glueframework.web.GlueHttpReceiverAdapter
  </servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>HttpReceiver</servlet-name>
  <url-pattern>*.tc</url-pattern>
</servlet-mapping>
```

- 라이브러리 추가
  - GlueSDK\lib\gluestd\_httpclient\commons-httpclient-3.1.jar



- etc/MSGFW004.dat

```
MSGFW001D90who (공백11자리) SEOUL (공백8자리)  
MSGFW001C90abcde (공백9자리) seoul (공백8자리)  
MSGFW001U90HR (공백12자리) PANGYO (공백7자리)
```

- web.xml
- SendTC.java (테스트용)

- web.xml 설정

```
<web-app . . .>

  <servlet>
    <servlet-name>HttpReceiver</servlet-name>
    <servlet-class>
      com.poscoict.glueframework.web.GlueHttpReceiverAdapter
    </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HttpReceiver</servlet-name>
    <url-pattern>/receiver.tc</url-pattern>
  </servlet-mapping>

</web-app>
```

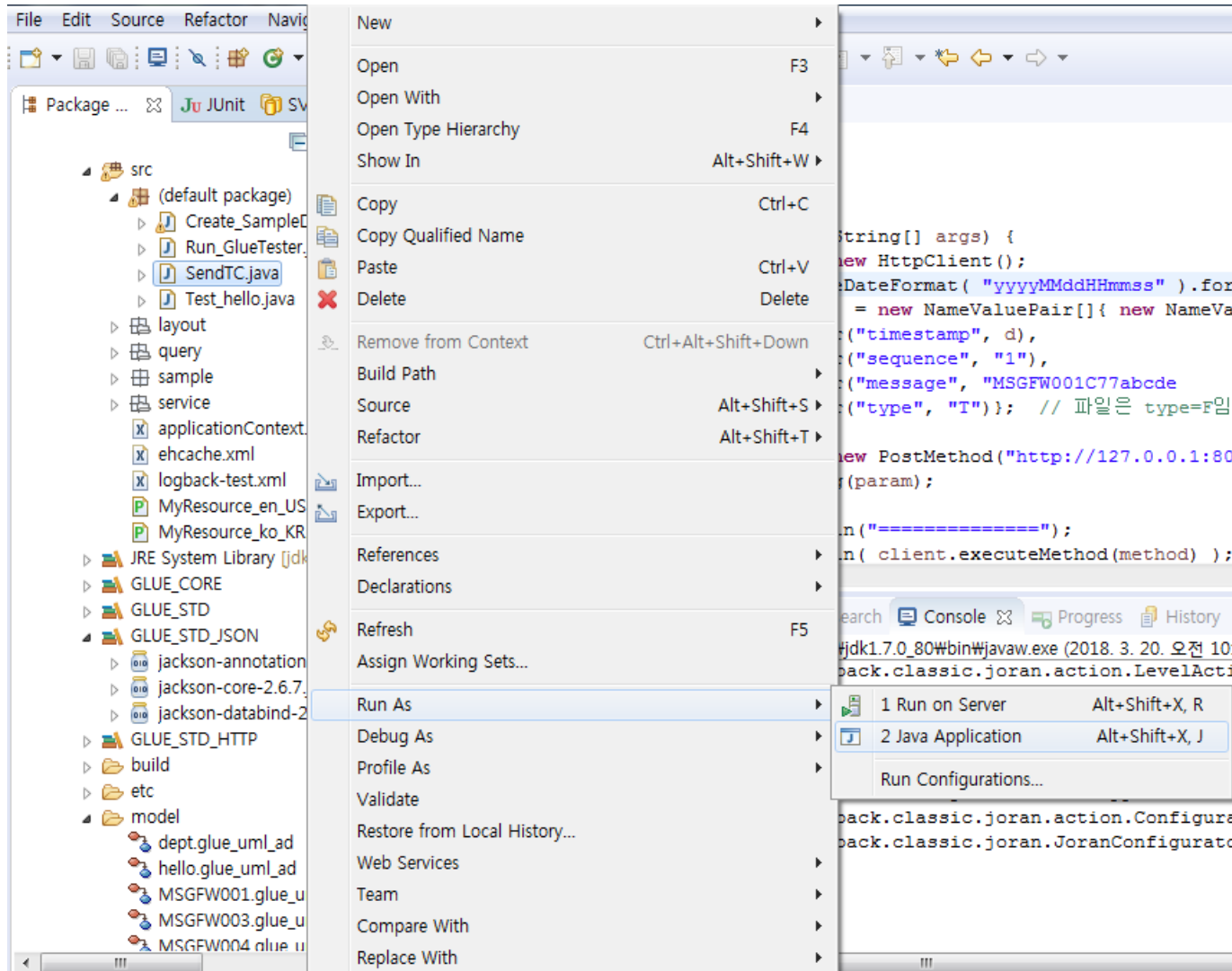
- SendTC.java(테스트용)

```
package test;
import java.text.*;
import java.util.*;
import org.apache.commons.httpclient.*;
import org.apache.commons.httpclient.methods.*;
public class SendTC {
    public static void main(String[] args) {
        HttpClient client = new HttpClient();
        String d = new SimpleDateFormat( "yyyyMMddHHmmss" ).format( new Date());
        NameValuePair param[] = new NameValuePair[]{ new NameValuePair("ifd", "TEST_TC_TC"),
            new NameValuePair("timestamp", d),
            new NameValuePair("sequence", "1"),
            //new NameValuePair("message", "C:/eclipse/workspace/sample/etc/MSGFW004.dat"),
            //new NameValuePair("type", "F")};
            new NameValuePair("message", "MSGFW001C77abcde                seoul
"),
            new NameValuePair("type", "T")}; /* 파일은 type=F임. message는 path포함 파일명임 */
        HttpMethod method = new PostMethod("http://127.0.0.1:8080/GlueSample/receiver.tc");
        method.setQueryString(param);
        try {
            System.out.println( "response code : " + client.executeMethod(method) );
        } catch ( Exception e ) {
            e.printStackTrace();
        }finally{
            method.releaseConnection();
        }
    }
}
```



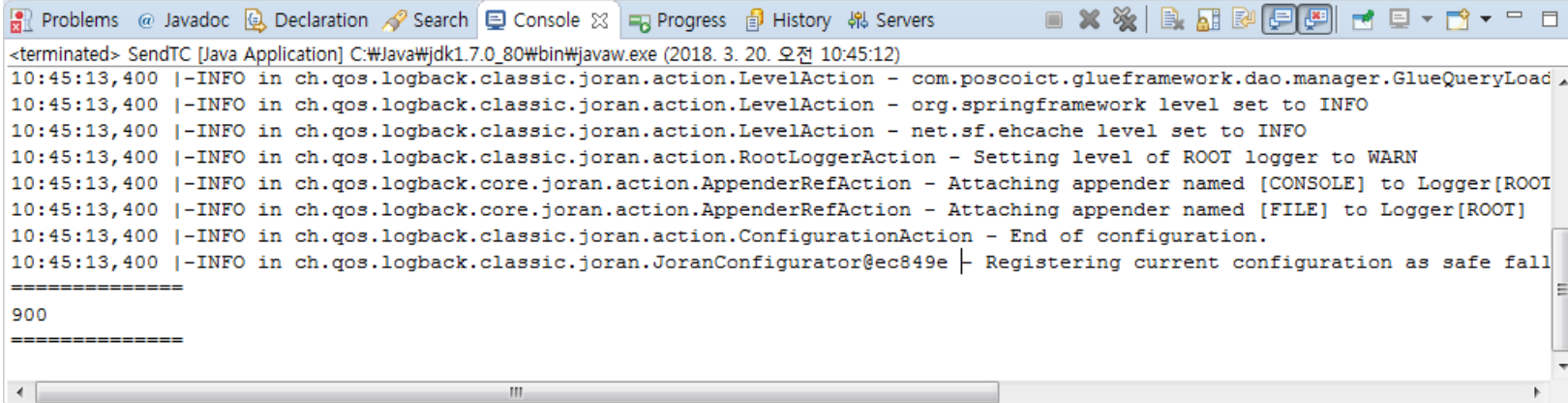
- **SendTC.java Parameter 설명**
  - type : T(TC), F(File)
  - message : type 이 T 일 경우 message string, F 일 경우 파일경로명
  - 이외의 파라미터는 u-CUBE 엔진에서 호출한것처럼 만들기 위한 의미없는 값
  - type 이 T 일 경우, message 에서 TC Length(기본 8자리)를 기준으로, type 이 F 일 경우, 파일명 앞에서 TC Length(기본 8자리)를 기준으로 서비스명을 생성해 Glue Service 를 수행함
  - 예를 들어 MSGFW001.dat 파일일 경우 수신 처리시 MSFGW001-service 를 수행.
- **다양한 테스트 예**
  - **MSGFW001-service 테스트**
    - type : T
    - message : MSGFW001C77abcde                      seoul
  - **MSGFW003-service 테스트**
    - type : T
    - message : MSGFW003C77abcde                      seoul
  - **MSGFW004-service 테스트**
    - type : F
    - message : C:/eclipse/workspace/sample/etc/MSGFW004.dat

## • 실행방법



- 실행결과 확인

- response code 가 900 이면 정상, 700 이면 실패
- <https://www.solutionpot.co.kr/doc/framework5/apidocs/com/poscoict/glueframework/web/GlueHttpReceiverAdapter.html>



```
<terminated> SendTC [Java Application] C:\Java\jdk1.7.0_80\bin\javaw.exe (2018. 3. 20. 오전 10:45:12)
10:45:13,400 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - com.poscoict.glueframework.dao.manager.GlueQueryLoad
10:45:13,400 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - org.springframework level set to INFO
10:45:13,400 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - net.sf.ehcache level set to INFO
10:45:13,400 |-INFO in ch.qos.logback.classic.joran.action.RootLoggerAction - Setting level of ROOT logger to WARN
10:45:13,400 |-INFO in ch.qos.logback.core.joran.action.AppenderRefAction - Attaching appender named [CONSOLE] to Logger[ROOT
10:45:13,400 |-INFO in ch.qos.logback.core.joran.action.AppenderRefAction - Attaching appender named [FILE] to Logger[ROOT]
10:45:13,400 |-INFO in ch.qos.logback.classic.joran.action.ConfigurationAction - End of configuration.
10:45:13,400 |-INFO in ch.qos.logback.classic.joran.JoranConfigurator@ec849e | Registering current configuration as safe fall
=====
900
=====
```

- 실행결과 확인
  - WAS Application Log 에서 서비스 수행관련 로그 확인

```

Tomcat
exec-1] DEBUG com.poscoict.glueframework.message.layout.GlueMessageUtil - Truncat Number :: 1102
exec-1] DEBUG com.poscoict.glueframework.message.layout.GlueMessageUtil - getStringFromNumber() value=1102
exec-1] DEBUG com.poscoict.glueframework.message.layout.GlueXmlMessageLayout - EMPNO = 1102
exec-1] DEBUG com.poscoict.glueframework.message.layout.GlueXmlMessageLayout - ENAME = name#2
exec-1] DEBUG com.poscoict.glueframework.message.layout.GlueMessageUtil - Original Number :: 1103
exec-1] DEBUG com.poscoict.glueframework.message.layout.GlueMessageUtil - Truncat Number :: 1103
exec-1] DEBUG com.poscoict.glueframework.message.layout.GlueMessageUtil - getStringFromNumber() value=1103
exec-1] DEBUG com.poscoict.glueframework.message.layout.GlueXmlMessageLayout - EMPNO = 1103
exec-1] DEBUG com.poscoict.glueframework.message.layout.GlueXmlMessageLayout - ENAME = name#3
exec-1] DEBUG com.poscoict.glueframework.message.layout.GlueXmlMessageLayout - ETC =
exec-1] INFO com.poscoict.glueframework.biz.control.GlueService$GlueActivityHandler - ActivityName:[MSGFW001-service][Mes
ime[Tue Mar 20 10:45:16 KST 2018] RunTime[16]
exec-1] INFO com.poscoict.glueframework.biz.control.GlueService$GlueActivityHandler - ActivityName:[MSGFW001-service][Biz
Tue Mar 20 10:45:16 KST 2018]
      1101name#1      1102name#2      1103name#3      ]
exec-1] INFO com.poscoict.glueframework.biz.control.GlueService$GlueActivityHandler - ActivityName:[MSGFW001-service][Biz
e Mar 20 10:45:16 KST 2018] RunTime[0]
exec-1] DEBUG com.poscoict.glueframework.transaction.GlueDataSourceTransactionManager - Initiating transaction commit
exec-1] DEBUG com.poscoict.glueframework.transaction.GlueDataSourceTransactionManager - Committing JDBC transaction on Con
ache-tomcat-8.5.16/sample.db, UserName=null, SQLiteJDBC]
exec-1] DEBUG com.poscoict.glueframework.transaction.GlueDataSourceTransactionManager - Releasing JDBC Connection [jdbc:sq
16/sample.db, UserName=null, SQLiteJDBC] after transaction
exec-1] INFO com.poscoict.glueframework.transaction.GlueDataSourceTransactionManager - Commit Complete
exec-1] INFO com.poscoict.glueframework.transaction.GlueDataSourceTransactionManager - Transaction Status is cleared
exec-1] DEBUG com.poscoict.glueframework.biz.control.GlueBizControllerImpl - Commit TX : test-tx
exec-1] INFO com.poscoict.glueframework.biz.control.GlueBizControllerImpl - ServiceName:[MSGFW001-service] EndTime[Tue Ma
ime:[202]
exec-1] INFO com.poscoict.glueframework.web.GlueHttpReceiverAdapter - Send Code to EAI Adapter[900]
    
```

- **GlueHttpReceiverAdapter**
  - Glue 측 TC 수신 Adapter
    - <https://www.solutionpot.co.kr/doc/framework/apidocs/com/poscoict/glueframework/web/GlueHttpReceiverAdapter.html>
    - 정상수신일 경우 200 이 아닌 900을 u-CUBE adapter 로 리턴
- **GlueMessageSender**
  - 송신관련 환경설정은 솔루션포털 > Glue > FAQ 에서 u-CUBE 로 검색
  - Glue 환경에서 u-CUBE 설정관련 내용 게시물 참고
  - interface type : TC, file 2가지
  - File type 전송일 경우 전송할 파일명 포함한 경로를 입력
  - 확장자 앞 파일명에서 8자리를 수신측에서는 TC code 로 이용함.
  - 송신처리 시 u-CUBE manager 를 이용해 송신, u-CUBE adapter 는 수신측 glue 시스템에 http call
- **GlueMessageParse**
  - Parse message (TC) Activity
- **GlueMessageCreate**
  - create message (TC) Activity

- **applicationContext.xml**

```
<beans ...>
...
<!-- MES향 (seadapter.jar 사용시) -->
<bean id="ucubeManager" class="com.poscoict.glueframework.ucube.GlueUCubeManagerImpl">
  <property name="configFile" value="C://SEND.xml"/>
</bean>
<!-- ERP향 (ucube-adapter.jar 사용시) -->
<!--
<bean id="ucubeManager" class="com.poscoict.glueframework.ucube4.GlueUCubeManagerImpl" >
  <property name="propertyFileName"
value="D:/uCUBE/services/uCUBE4/Send/config/GLUESEND_TLCENGA01_U.properties"/>
  <property name="serviceBrokerNames">
    <list>
      <value>SEND_FILE_HEADER</value>
    </list>
  </property>
  <property name="period" value="100"/>
</bean>
-->
...
</baans>
```

- 수신측의 시스템에 설치된 u-CUBE 버전에 따라 설정파일(xml 혹은 properties)이 다름

- GlueMessageSender 사용 예 (u-CUBE3 : MES 향)

The screenshot illustrates the configuration of a GlueMessageSender activity. On the left, an activity diagram shows a flow from a start node to a yellow box labeled "Message Send(u-CUBE 3)", which then leads to an end node labeled "success".

The "Activity Setting" dialog is open, showing the following configuration:

- Name : Message Send(u-CUBE 3)
- Properties :
 

Name	Value
<input type="checkbox"/> class	com.poscoict.glueframework.ucube.activity.GlueMessageSender
<input type="checkbox"/> interface-id	value
<input type="checkbox"/> interface-type	value
<input type="checkbox"/> message	ctxName
<input type="checkbox"/> sender	ucube3Manager

The "components" palette on the right shows the "Message Send(u-CUBE 3)" component highlighted with a red box.

- GlueMessageSender 사용 예 (u-CUBE : ERP 향)

The screenshot displays the configuration for the 'Message Send(u-CUBE 4)' activity. The 'Activity Setting' dialog box is open, showing the following details:

- Name:** Message Send(u-CUBE 4)
- Properties:**

Name	Value
<input type="checkbox"/> class	com.poscoict.glueframework.ucube4.activity.GlueMessageSender
<input type="checkbox"/> interface-id	S24H-0142-02
<input type="checkbox"/> interface-type	file
<input type="checkbox"/> message	message
<input type="checkbox"/> sender	ucubeManager
<input checked="" type="checkbox"/> service-broker	SEND_FILE_HEADER

The 'components' palette on the right side of the IDE shows the 'Message Send(u-CUBE 4)' component selected, which is also highlighted with a red box. The 'service-broker' property in the configuration dialog is also highlighted with a red box.



# IV. Glue Framework 제작 가이드

## 3장 심화 학습

### 2. RESTful API 만들기





## 목표

## 실습

- RESTful URL에 API를 매칭시키기 위해
  - Web application을 설정할 수 있다.
  - RESTful Controller를 구현할 수 있다.

- dispatcher-restful-servlet.xml 변경하기
- sample.controller.RestController 구현하기

- RESTful API를 통해 GlueService를 호출하기 위해
  - GlueService를 만들 수 있다.
  - Jsp에서 RESTful URL로 GlueService를 호출할 수 있다.
  - API에 Parameter 를 추가할 수 있다.

- restful-service 구현하기
- <http://localhost:8080/sample-restful-server/restful/glue/restful-service> 호출하기
- http://localhost:8080/sample-restful-server/restful/glue/restful-service/data/30 호출하기

- RESTful
  - web.xml 에 URL pattern 지정
  - /WebContents/WEB-INF/dispatcher-restful-servlet.xml
  - /WebContents/WEB-INF/views.xml
  - restful-query.glue\_sql
  - sample.controller.RestController
  - restful.glue\_uml\_ad
  - sample.activity.GlueJdbcSearch,  
GlueJdbcInsert, GlueJdbcModify, GlueJdbcDelete
  - /src/test/RestfulTest.java
  - /WebContents/client\_test/restful\_client\_test.jsp
  - <https://www.solutionpot.co.kr/doc/framework/docs/webservice.html#RESTful>

- RESTful

- web.xml

```
<web-app ...>
...
  <servlet>
    <servlet-name>dispatcher-restful</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>dispatcher-restful</servlet-name>
    <url-pattern>/restful/*</url-pattern>
  </servlet-mapping>
...
</web-app>
```

- build.xml

- `<include name="**/*.jsp"/>` 아래 `<include name="**/*.js"/>`
    - jackson library 추가 되도록 수정
    - build.properties 에서 WEB\_NAME=sample-restful-server 로 변경해서 빌드

- RESTful
  - dispatcher-restful-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-4.0.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd">

  <context:component-scan base-package="sample.controller" />
  <mvc:annotation-driven />
  <bean id="viewResolver" class="org.springframework.web.servlet.view.XmlViewResolver">
    <property name="order" value="1"/>
    <property name="location" value="/WEB-INF/views.xml"/>
  </bean>
</beans>
```

- RESTful

- views.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
       http://www.springframework.org/schema/beans/spring-beans-4.0.xsd">
  <bean name="jsonView"
        class="org.springframework.web.servlet.view.json.MappingJackson2JsonView"/>
</beans>
```

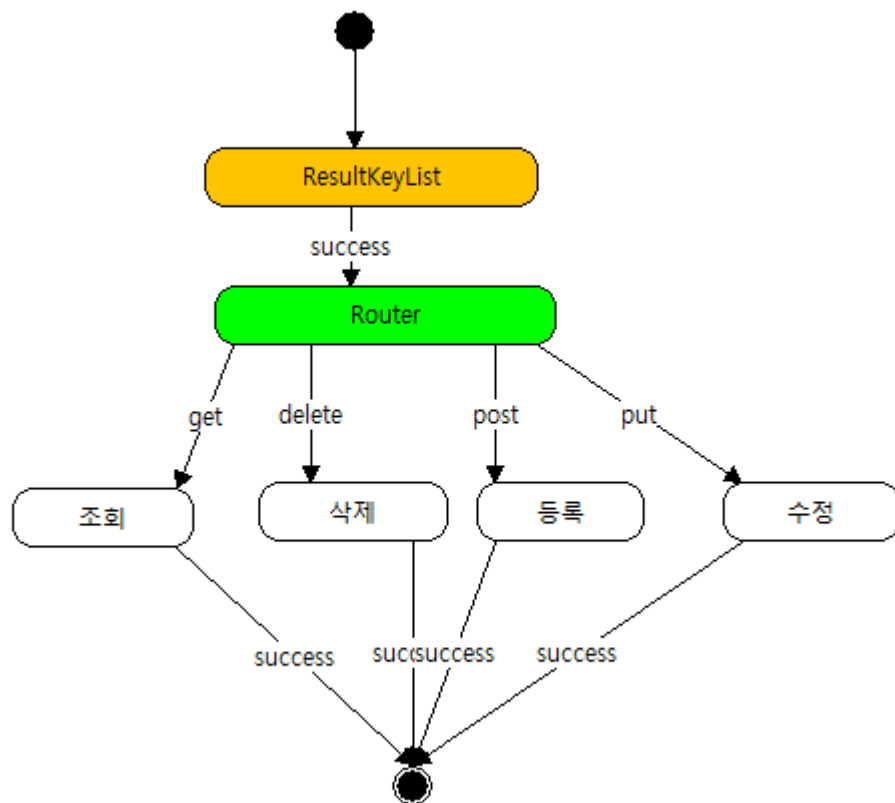
- RESTful

- `restful-query.glue_sql`

```
<?xml version="1.0" encoding="UTF-8"?>
<queryMap desc="queries for restful" xmlns="http://www.poscoict.com/glueframework/query">
  <query id="restful.emp.select" desc="EMP 조회" resultType="" isNamed="false"><![CDATA[
select EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, DEPTNO
from EMP
where DEPTNO=?
]]></query>
  <query id="restful.emp.update" desc="EMP 수정" resultType="" isNamed="false"><![CDATA[
update emp
set sal=?, ename=?
where empno=?
]]></query>
  <query id="restful.emp.delete" desc="EMP 삭제" resultType="" isNamed="false"><![CDATA[
delete from emp
where empno=?
]]></query>
  <query id="restful.emp.insert" desc="EMP 등록" resultType="" isNamed="false"><![CDATA[
insert into emp(empno, ename, sal, deptno)
values(?, ?, ?, ?)
]]></query>
</queryMap>
```



- RESTful
  - restful.glue\_uml\_ad



**Activity Setting**

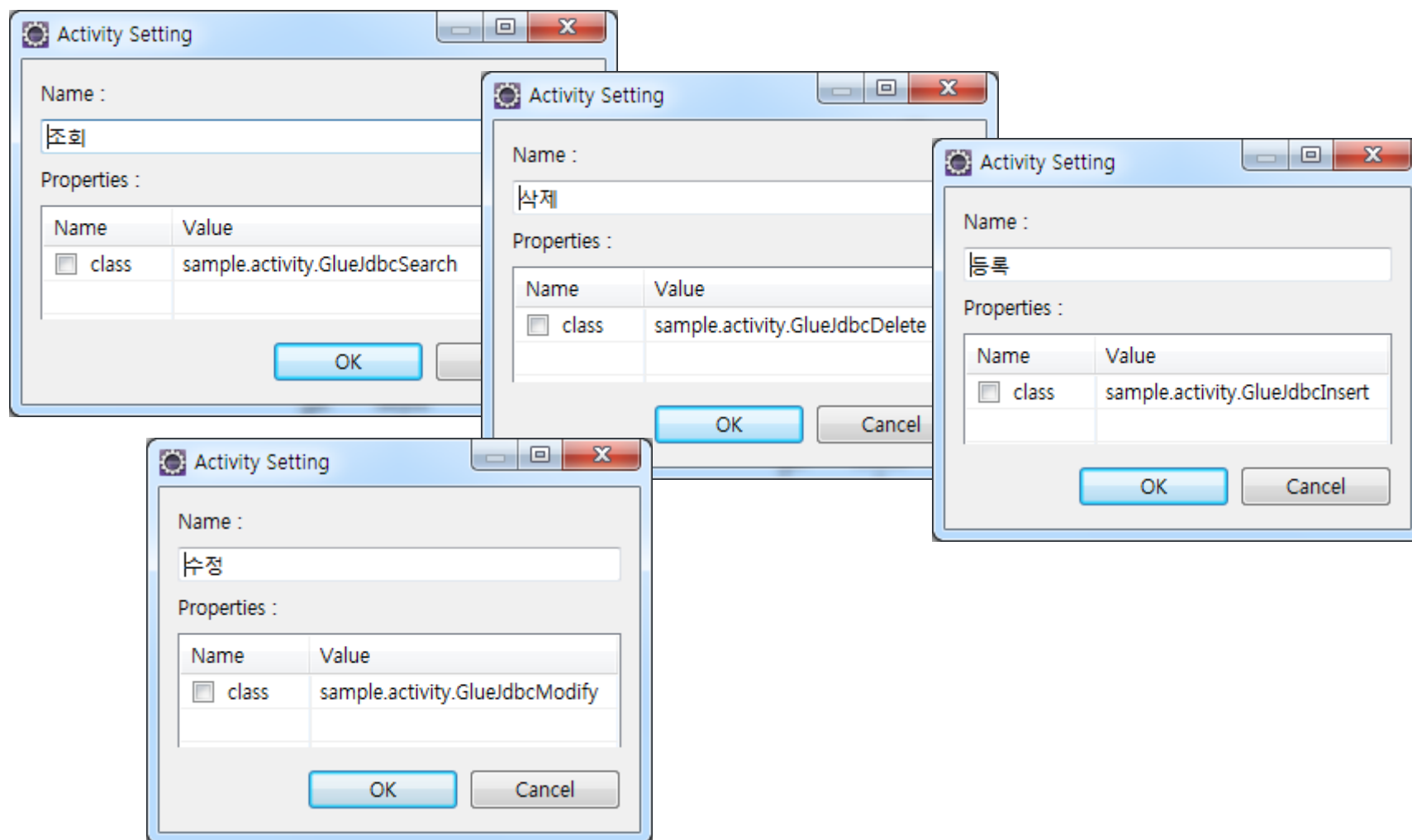
Name :

Properties :

Name	Value
<input type="checkbox"/> class	com.poscoict.glueframework.biz.activity.GlueResultKeyList
<input type="checkbox"/> ctx-keys	search_result delete_cnt insert_cnt update_cnt

Description :

- RESTful
  - `restful.glue_uml_ad`



- RESTful
  - **sample.activity.GlueJdbcSearch**

```
package sample.activity;

import java.util.ArrayList;
import java.util.List;

import com.poscoict.glueframework.biz.activity.GlueActivity;
import com.poscoict.glueframework.biz.control.GlueBizControlConstants;
import com.poscoict.glueframework.context.GlueContext;
import com.poscoict.glueframework.dao.jdbc.GlueGenericJdbcDao;
import com.poscoict.glueframework.dao.vo.GlueParameter;

public class GlueJdbcSearch extends GlueActivity<GlueContext>
{
    public String runActivity( GlueContext ctx ) {
        GlueGenericJdbcDao dao = (GlueGenericJdbcDao) this.getDao( "test-dao" );
        GlueParameter<List> param = new GlueParameter<List>();
        List paramList = new ArrayList();
        paramList.add( ctx.get( "restfulData" ) );
        param.setParameter( paramList );
        List<?> result = dao.find( "restful.emp.select", param );
        ctx.put( "search_result", result );
        return GlueBizControlConstants.SUCCESS;
    }
}
```

- RESTful
  - **sample.activity.GlueJdbcDelete**

```
package sample.activity;

import java.util.ArrayList;
import java.util.List;

import com.poscoict.glueframework.biz.activity.GlueActivity;
import com.poscoict.glueframework.biz.control.GlueBizControlConstants;
import com.poscoict.glueframework.context.GlueContext;
import com.poscoict.glueframework.dao.jdbc.GlueGenericJdbcDao;
import com.poscoict.glueframework.dao.vo.GlueParameter;

public class GlueJdbcDelete extends GlueActivity<GlueContext>
{
    public String runActivity( GlueContext ctx ) {
        GlueGenericJdbcDao dao = (GlueGenericJdbcDao) this.getDao( "test-dao" );
        GlueParameter<List> param = new GlueParameter<List>();
        List paramList = new ArrayList();
        paramList.add( ctx.get( "restfulData" ) );
        param.setParameter( paramList );
        int result = dao.delete( "restful.emp.delete", param );
        ctx.put( "delete_cnt", result );
        return GlueBizControlConstants.SUCCESS;
    }
}
```

- RESTful
  - **sample.activity.GlueJdbcInsert**

```
package sample.activity;

import java.util.ArrayList;
import java.util.List;
import com.poscoict.glueframework.biz.activity.GlueActivity;
import com.poscoict.glueframework.biz.control.GlueBizControlConstants;
import com.poscoict.glueframework.context.GlueContext;
import com.poscoict.glueframework.dao.jdbc.GlueGenericJdbcDao;
import com.poscoict.glueframework.dao.vo.GlueParameter;

public class GlueJdbcInsert extends GlueActivity<GlueContext> {
    public String runActivity( GlueContext ctx ) {
        GlueGenericJdbcDao dao = (GlueGenericJdbcDao) this.getDao( "test-dao" );
        GlueParameter<List> param = new GlueParameter<List>();
        List paramList = new ArrayList();
        paramList.add( ctx.get( "param1" ) );
        paramList.add( ctx.get( "param2" ) );
        paramList.add( ctx.get( "param3" ) );
        paramList.add( ctx.get( "param4" ) );
        param.setParameter( paramList );
        int result = dao.insert( "restful.emp.insert", param );
        ctx.put( "insert_cnt", result );
        return GlueBizControlConstants.SUCCESS;
    }
}
```

- RESTful

- **sample.activity.GlueJdbcModify**

```
package sample.activity;
import java.util.ArrayList;
import java.util.List;
import com.poscoict.glueframework.biz.activity.GlueActivity;
import com.poscoict.glueframework.biz.control.GlueBizControlConstants;
import com.poscoict.glueframework.context.GlueContext;
import com.poscoict.glueframework.dao.jdbc.GlueGenericJdbcDao;
import com.poscoict.glueframework.dao.vo.GlueParameter;
public class GlueJdbcModify extends GlueActivity<GlueContext> {
    public String runActivity( GlueContext ctx ) {
        GlueGenericJdbcDao dao = (GlueGenericJdbcDao) this.getDao( "test-dao" );
        GlueParameter<List> param = new GlueParameter<List>();
        List paramList = new ArrayList();
        List<String> param1 = (List<String>) ctx.get( "param1" );
        List<String> param2 = (List<String>) ctx.get( "param2" );
        List<String> param3 = (List<String>) ctx.get( "param3" );
        paramList.add( param1.get( 0 ) );
        paramList.add( param2.get( 0 ) );
        paramList.add( param3.get( 0 ) );
        param.setParameter( paramList );
        int result = dao.update( "restful.emp.update", param );
        ctx.put( "update_cnt", result );
        return GlueBizControlConstants.SUCCESS;
    }
}
```

- RESTful

- **sample.controller.RestfulController(1/9)**

```
package sample.controller;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.slf4j.MDC;
import org.springframework.stereotype.Controller;
import org.springframework.util.MultiValueMap;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

import com.poscoict.glueframework.biz.control.GlueBizControlConstants;
import com.poscoict.glueframework.biz.control.GlueBizController;
import com.poscoict.glueframework.biz.control.GlueBizProvider;
import com.poscoict.glueframework.context.GlueDefaultContext;
import com.poscoict.glueframework.util.log.GlueLog;
import com.poscoict.glueframework.util.log.GlueLogFactory;
import com.poscoict.glueframework.web.control.GlueWebConstants;
```

- RESTful

- sample.controller.RestfulController(2/9)

```
/**
 * RESTful방식의 웹서비스를 제공하는 Controller URI 패턴으로는 아래 두가지 방식이 제공된다.
 * 1) /glue/{serviceName} : URI의 serviceName과 Service명이 일치하는 Glue Service가 실행 된다.
 * 2) /glue/{serviceName}/data/{gluedata} : URI의 serviceName과 Service명이 일치하는 Glue Service가 실행되며
 * URI의 gluedata값은 "gluedata"를 Key로 GlueContext에 저장된다.
 * GET,POST,PUT,DELETE 방식이 지원되며 어떤 방식의 요청이었는지는 "action"을 Key로 GlueContext에 소문자로 저장된다.
 * 해당 요청이 Key로도 등록되므로 Default Router로 분기도 가능하다.
 * ajax방식이나 RestTemplate을 사용하여 서비스를 호출 할 수있으며 JSON 형식의 데이터가 Return된다.
 */
@Controller
public class RestfulController
{
    /** logger */
    protected GlueLog logger = GlueLogFactory.getLogger( getClass() );

    /**
     * @param serviceName
     * @return
     */
    @RequestMapping( value = "/glue/{serviceName}", method = RequestMethod.GET )
    public ModelAndView doGlueServiceGet( @PathVariable String serviceName )
    {
        ModelAndView mav = new ModelAndView( GlueWebConstants.JSON_VIEW );
        Map<String, Object> glueResult = doGlueService( serviceName, GlueWebConstants.GET, null, null );
    }
}
```



- RESTful

- **sample.controller.RestfulController(3/9)**

```
        mav.addAllObjects( glueResult );
        return mav;
    }

    /**
     * @param serviceName
     * @param gluedata
     * @return
     */
    @RequestMapping( value = "/glue/{serviceName}/data/{gluedata}", method = RequestMethod.GET )
    public ModelAndView doGlueServiceGetData( @PathVariable String serviceName, @PathVariable
String gluedata )
    {
        ModelAndView mav = new ModelAndView( GlueWebConstants.JSON_VIEW );
        Map<String, Object> glueResult = doGlueService( serviceName, GlueWebConstants.GET, gluedata,
null );
        mav.addAllObjects( glueResult );
        return mav;
    }

    /**
     * @param serviceName
     * @return
     */
    @RequestMapping( value = "/glue/{serviceName}", method = RequestMethod.DELETE )
    public ModelAndView doGlueServiceDelete( @PathVariable String serviceName )
```

- RESTful

- sample.controller.RestfulController(4/9)

```
    {
        ModelAndView mav = new ModelAndView( GlueWebConstants.JSON_VIEW );
        Map<String, Object> glueResult = doGlueService( serviceName,
GlueWebConstants.DELETE, null, null );
        mav.addAllObjects( glueResult );
        return mav;
    }

    /**
     * @param serviceName
     * @param gluedata
     * @return
     */
    @RequestMapping( value = "/glue/{serviceName}/data/{gluedata}", method =
RequestMethod.DELETE )
    public ModelAndView doGlueServiceDeletetByData( @PathVariable String serviceName,
@PathVariable String gluedata )
    {
        ModelAndView mav = new ModelAndView( GlueWebConstants.JSON_VIEW );
        Map<String, Object> glueResult = doGlueService( serviceName,
GlueWebConstants.DELETE, gluedata, null );
        mav.addAllObjects( glueResult );
        return mav;
    }
}
```

- RESTful

- sample.controller.RestfulController(5/9)

```
/**
 * @param serviceName
 * @param params
 * @return
 */
@RequestMapping( value = "/glue/{serviceName}", method = RequestMethod.POST )
public ModelAndView doGlueServicePost( @PathVariable String serviceName, @RequestParam
Map<String, Object> params )
{
    ModelAndView mav = new ModelAndView( GlueWebConstants.JSON_VIEW );
    Map<String, Object> glueResult = doGlueService( serviceName, GlueWebConstants.POST,
null, params );
    mav.addAllObjects( glueResult );
    return mav;
}

/**
 * @param serviceName
 * @param gluedata
 * @param params
 * @return
 */
@RequestMapping( value = "/glue/{serviceName}/data/{gluedata}", method =
RequestMethod.POST )
public ModelAndView doGlueServicePostByData( @PathVariable String serviceName,
@PathVariable String gluedata, @RequestParam Map<String, Object> params )
```

- RESTful

- sample.controller.RestfulController(6/9)

```
{
    ModelAndView mav = new ModelAndView( GlueWebConstants.JSON_VIEW );
    Map<String, Object> glueResult = doGlueService( serviceName, GlueWebConstants.POST,
gluedata, params );
    mav.addAllObjects( glueResult );
    return mav;
}

/**
 * @param serviceName
 * @param params
 * @return
 */
@RequestMapping( value = "/glue/{serviceName}", method = RequestMethod.PUT )
public ModelAndView doGlueServicePut( @PathVariable String serviceName, @RequestBody
MultiValueMap<String, String> params )
{
    ModelAndView mav = new ModelAndView( GlueWebConstants.JSON_VIEW );
    Map<String, Object> glueResult = doGlueService( serviceName, GlueWebConstants.PUT,
null, params );
    mav.addAllObjects( glueResult );
    return mav;
}
```

- RESTful
  - **sample.controller.RestfulController(7/9)**

```
/**
 * @param serviceName
 * @param gluedata
 * @param params
 * @return
 */
@RequestMapping( value = "/glue/{serviceName}/data/{gluedata}", method =
RequestMethod.PUT )
public ModelAndView doGlueServicePutByData( @PathVariable String serviceName,
@PathVariable String gluedata, @RequestBody MultiValueMap<String, String> params )
{
    ModelAndView mav = new ModelAndView( GlueWebConstants.JSON_VIEW );
    Map<String, Object> glueResult = doGlueService( serviceName, GlueWebConstants.PUT,
gluedata, params );
    mav.addAllObjects( glueResult );
    return mav;
}

/**
 * @param serviceName
 * @param action
 * @param data
 * @param params
 * @return
 */
```

- RESTful

- **sample.controller.RestfulController(8/9)**

```
private Map<String, Object> doGlueService( String serviceName, String action, String
data, Map<String, ?> params )
{
    logger.debug( "Glue ServiceName : {}", serviceName );
    logger.debug( "RESTful Action : {}", action );
    logger.debug( "RESTful Data : {}", data );
    logger.debug( "RequestBody : {}", params );

    GlueDefaultContext ctx = new GlueDefaultContext();
    Map<String, Object> result = new HashMap<String, Object>();
    List<?> result_key_list = null;
    if ( params != null )
    {
        ctx.putAll( params );
    }
    ctx.put( GlueBizControlConstants.SERVICE_NAME, serviceName );
    ctx.put( GlueWebConstants.RESTFUL_ACTION, action );
    ctx.put( action, action );
    ctx.put( GlueWebConstants.RESTFUL_DATA, data );

    try
    {
        MDC.put( GlueBizControlConstants.SERVICE_NAME, serviceName );
        GlueBizController bizIF = GlueBizProvider.getController();
        bizIF.doAction( ctx );
        MDC.remove( GlueBizControlConstants.SERVICE_NAME );
    }
}
```

- RESTful

- **sample.controller.RestfulController(9/9)**

```
        result_key_list = (List<?>) ctx.get( GlueBizControlConstants.RESULT_KEY_LIST );
    } catch ( Exception e )
    {
        result.put( GlueWebConstants.RESTFUL_ERROR, e );
        logger.error( e.getMessage(), e );
    }

    if ( ( result_key_list != null ) && ( result_key_list.size() > 0 ) )
    {
        for ( int i = 0; i < result_key_list.size(); i++ )
        {
            result.put( (String) result_key_list.get( i ),
ctx.get( result_key_list.get( i ) ) );
        }
    }

    return result;
}
}
```

- RESTful
  - test.RestfulTest

```
package test;
import java.util.List;
import java.util.Map;
import org.springframework.web.client.RestTemplate;
public class RestfulTest {
    public static void main( String[] args ) throws Exception {
        System.out.println( "=====getForObject String Start=====" );
        RestTemplate restTemplate = new RestTemplate();
        String resultStr = restTemplate.getForObject( "http://localhost:8080/sample-
restful-server/restful/glue/restful-service/data/30", String.class );
        System.out.println( "result:::::" + resultStr );
        System.out.println( "=====getForObject String End=====" );
        System.out.println( "=====getForObject Map Start=====" );
        Map<?, ?> result = restTemplate.getForObject( "http://localhost:8080/sample-
restful-server/restful/glue/restful-service/data/30", Map.class );
        System.out.println( "result:::::" + result );
        List<Map> resultList = (List<Map>) result.get( "search_result" );
        for ( int i = 0; i < resultList.size(); i++ ) {
            Map emp = resultList.get( i );
            System.out.println( emp );
        }
        System.out.println( "=====getForObject Map End=====" );
    }
}
```

- 이클립스 프로젝트 build path 에 jackson 라이브러리 추가



- RESTful

- `restful_client_test.jsp` (1/4)

```
<%@ page contentType="text/html; charset=UTF-8"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Sample</title>
<script src="https://code.jquery.com/jquery-2.1.3.min.js"></script>
<script type="text/javascript">
$.doService = function(opt){
    $('#result').html('실행중');
    if(opt=='POST'){
        var formdata = $("form[name=formdata1]").serializeArray();

        $.ajax({
            type: opt,
            url: "http://localhost:8080/sample-restful-server/restful/glue/restful-service/",
            data: formdata,
            success: function(msg){
                $('#result').html(JSON.stringify(msg));
            }
        });
    }
}
} else if(opt=='PUT'){
    var formdata = $("form[name=formdata2]").serializeArray();
```

- RESTful
  - `restful_client_test.jsp (2/4)`

```
$.ajax({
    type: opt,
    url: "http://localhost:8080/sample-restful-server/restful/glue/restful-service/",
    data: formdata,
    success: function(msg) {
        $('#result').html(JSON.stringify(msg));
    } });
} else if (opt == 'GET') {
    var data = $("#getvalue").val();
    $.ajax({
        type: opt,
        url: "http://localhost:8080/sample-restful-server/restful/glue/restful-
service/data/"+data,
        success: function(msg) {
            $('#result').html(JSON.stringify(msg));
        } });
} else if (opt == 'DELETE') {
    var data = $("#deletevalue").val();
    $.ajax({
        type: opt,
        url: "http://localhost:8080/sample-restful-server/restful/glue/restful-
service/data/"+data,
        success: function(msg) {
            $('#result').html(JSON.stringify(msg));
        } });
} }
```

- RESTful
  - `restful_client_test.jsp (3/4)`

```
</script>
</head>
<body>
RESTFUL TEST<br/><br/>
GET<br/>
URL:http://localhost:<%=request.getLocalPort() %><%=request.getContextPath() %>/restful/glue/
restful-service/data/<input type='text' value='10' id='getvalue' >
<br/><br/>
DELETE<br/>
URL:http://localhost:<%=request.getLocalPort() %><%=request.getContextPath() %>/restful/glue/
restful-service/data/<input type='text' value='9999' id='deletevalue' >
<br/><br/>
POST DATA(INSERT)
<form id= 'formdata1' name= 'formdata1'>
param1(empno):<input type='text' value='9999' id='param1' name='param1'>
param2(ename):<input type='text' value='ename_test' id='param2' name='param2'>
param3(sal):<input type='text' value='800' id='param3' name='param3'>
param4(deptno):<input type='text' value='10' id='param4' name='param4'>
</form>
<br/><br/>
PUT DATA(UPDATE)
<form id= 'formdata2' name= 'formdata2'>
param1(sal):<input type='text' value='900' id='param1' name='param1'>
param2(ename):<input type='text' value='ename_test' id='param2' name='param2'>
param3(empno):<input type='text' value='9999' id='param3' name='param3'>
</form>
```

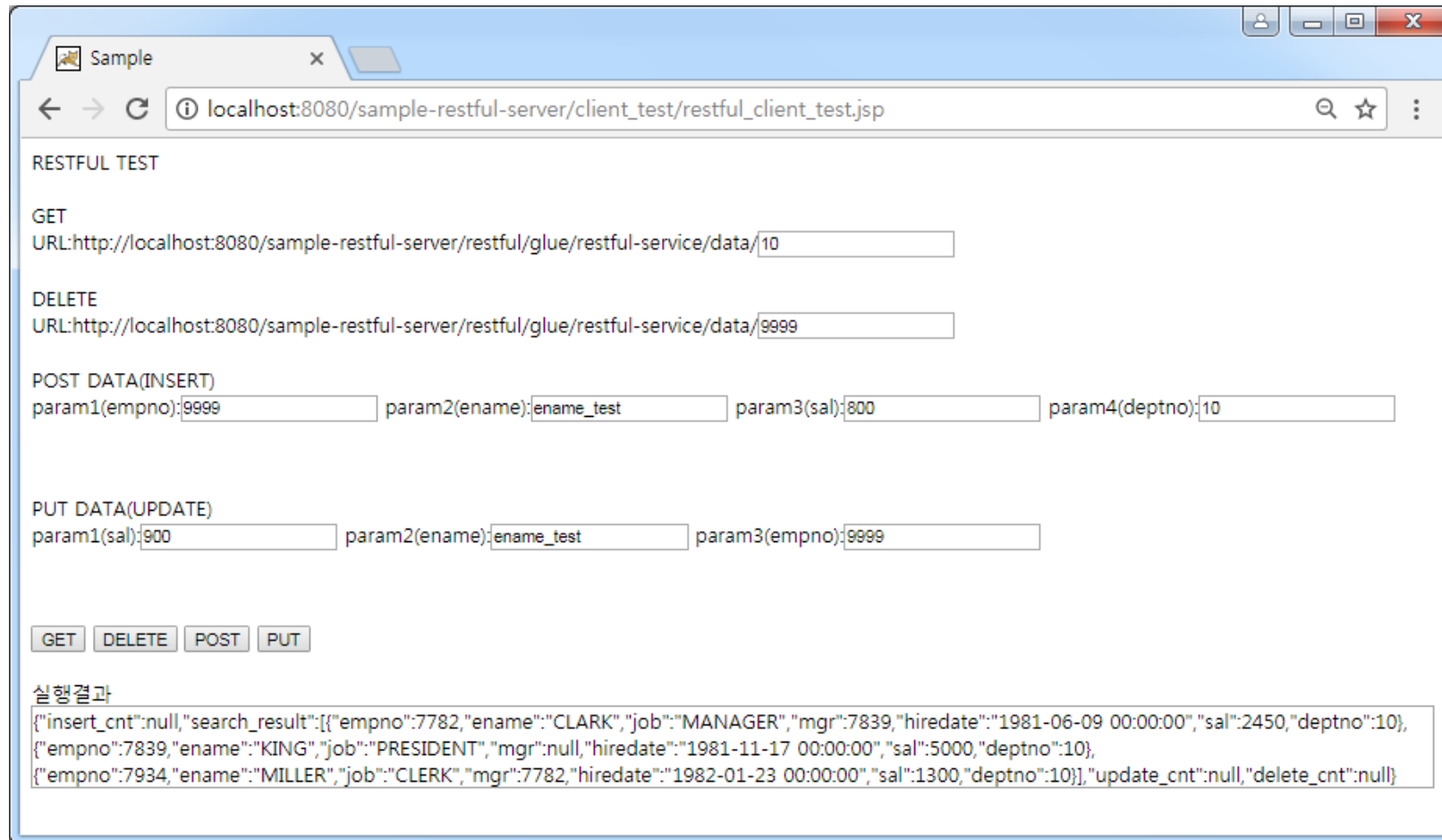
- RESTful
  - `restful_client_test.jsp (4/4)`

```
<br/><br/>
<input type='button' value='GET' onclick="$.doService('GET')" >
<input type='button' value='DELETE' onclick="$.doService('DELETE')" >
<input type='button' value='POST' onclick="$.doService('POST')" >
<input type='button' value='PUT' onclick="$.doService('PUT')" >
<br/><br/>
실행결과
<br/>
<div id="result" style="border-color:#a4a4a4;border-width: 1px;border-style: solid;">
</div>
</body>
</html>
```

- RESTful

- 화면 테스트

- [http://localhost:8080/sample-restful-server/client\\_test/restful\\_client\\_test.jsp](http://localhost:8080/sample-restful-server/client_test/restful_client_test.jsp)





# IV. Glue Framework 제작 가이드

## 3장 심화 학습

### 3. WebSocket 사용하기







---

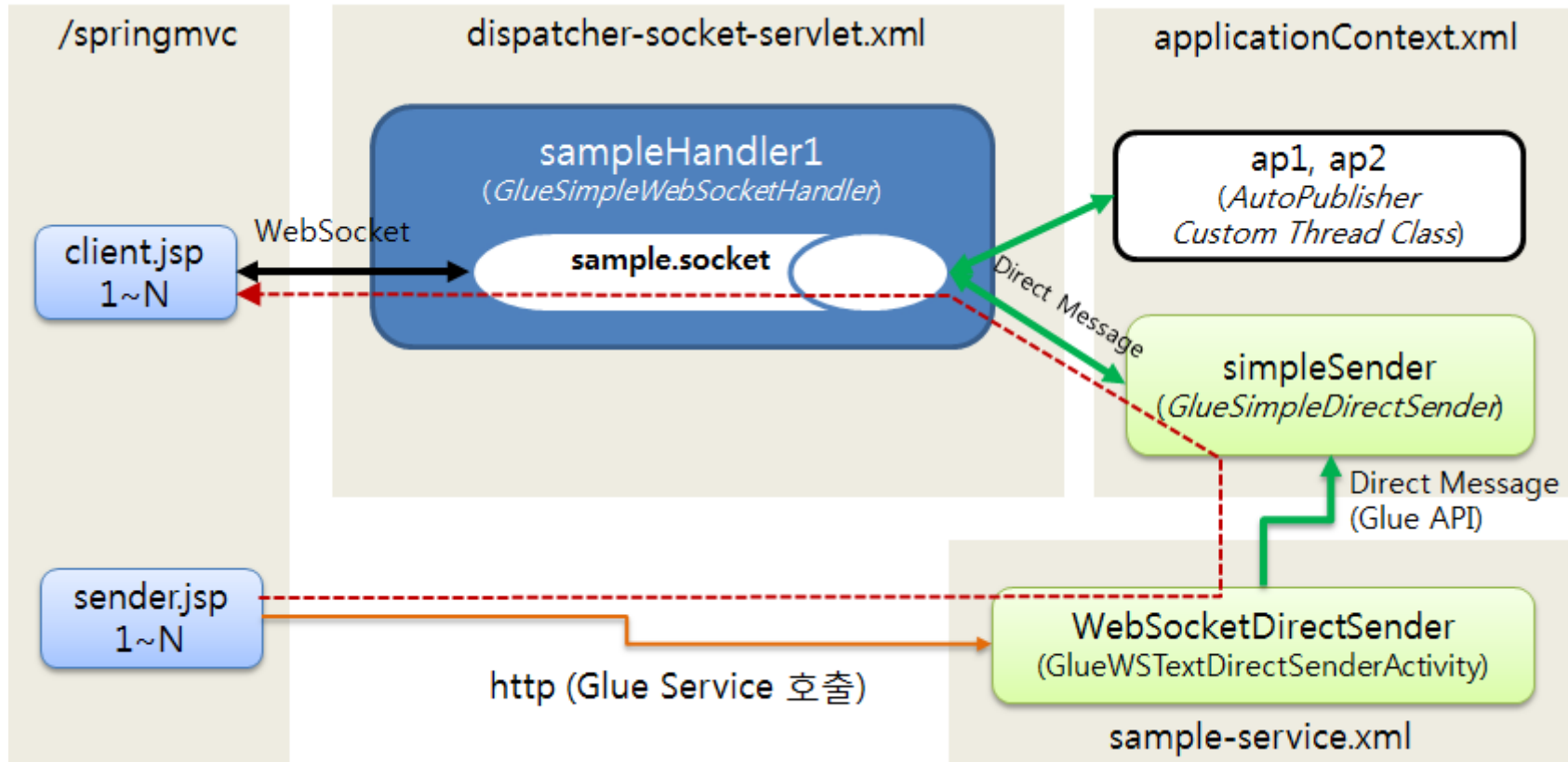
## 목표

- WebSocket을 사용하여 텍스트 송수신을 할 수 있다.

## 실습

- Websocket sample project를 생성하여 테스트
- 
- 
- 
- 
-

- WebSocket



- <https://www.solutionpot.co.kr/doc/framework5/guide/sample-project/howto-sample-websocket.html>

- applicationContext.xml

```
...
<bean id="simpleSender" class="com.poscoict.glueframework.websocket.GlueSimpleDirectSender" />
<bean id="ap1" class="sample.publisher.AutoPublisher">
  <property name="propertiesMap">
    <map>
      <entry key="id" value="ap1" />
      <entry key="ip" value="192.168.193.179" /> <!-- sample property. 실제로 사용하지는
않음 -->
      <entry key="port" value="12001" /> <!-- sample property. 실제로 사용하지는 않음 -->
      <entry key="timeout" value="5000" />
    </map>
  </property>
</bean>
<bean id="ap2" class="sample.publisher.AutoPublisher">
  <property name="propertiesMap">
    <map>
      <entry key="id" value="ap2" />
      <entry key="ip" value="172.18.49.90" /><!-- sample property. 실제로 사용하지는 않음 -
->
      <entry key="port" value="12001" /><!-- sample property. 실제로 사용하지는 않음 -->
      <entry key="timeout" value="5000" />
    </map>
  </property>
</bean>
...
```

- web.xml

```
...
    <servlet>
        <servlet-name>dispatcher</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>/WEB-INF/dispatcher-servlet.xml</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>dispatcher</servlet-name>
        <url-pattern>*.mvc</url-pattern>
    </servlet-mapping>

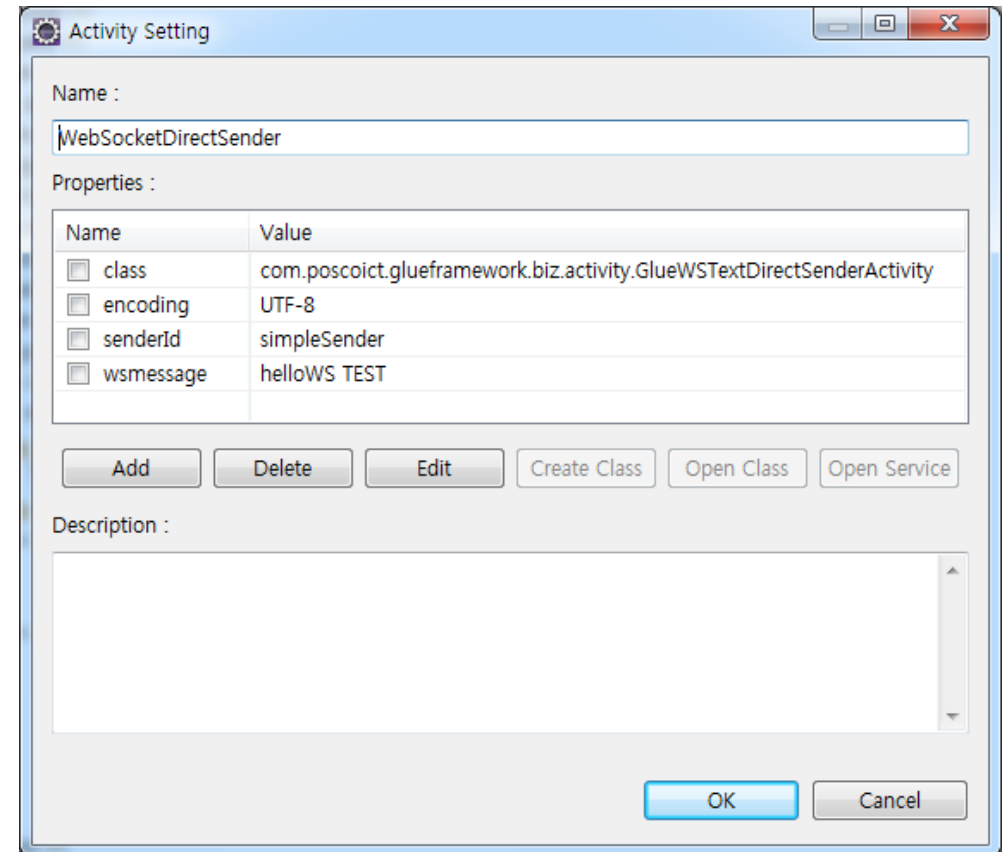
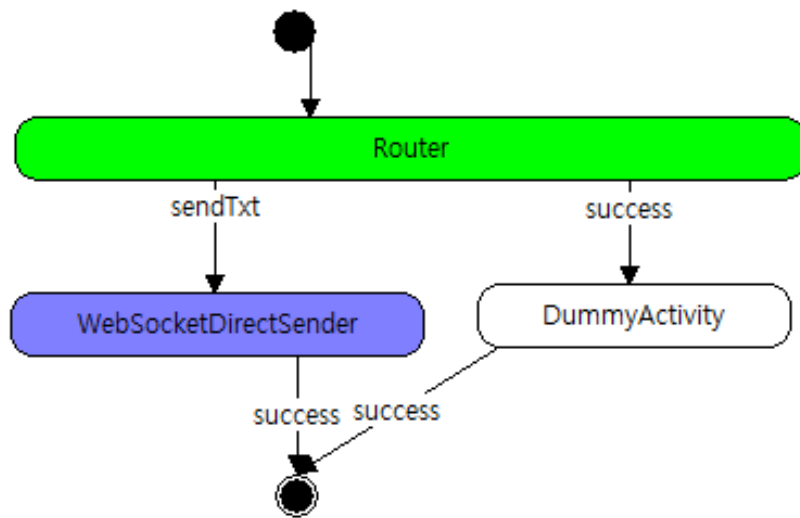
    <servlet>
        <servlet-name>dispatcher-socket</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>dispatcher-socket</servlet-name>
        <url-pattern>*.socket</url-pattern>
    </servlet-mapping>
...

```

- **dispatcher-socket-servlet.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:websocket="http://www.springframework.org/schema/websocket"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.0.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd
    http://www.springframework.org/schema/websocket
    http://www.springframework.org/schema/websocket/spring-websocket.xsd">
  <websocket:handlers>
    <websocket:mapping path="/sample.socket" handler="sampleHandler1"/>
  </websocket:handlers>
  <bean id="sampleHandler1"
class="com.poscoict.glueframework.websocket.GlueSimpleWebSocketHandler">
    <property name="directSenderOnStartup" value="ap1,ap2,simpleSender"/>
  </bean>
  <bean
class="org.springframework.web.socket.server.standard.ServletServerContainerFactoryBean">
-->    <property name="maxTextMessageBufferSize" value="10000"/> <!-- size 넘어가면 close
    <property name="maxSessionIdleTimeout" value="30000"/> <!-- milliseconds -->
  </bean>
</beans>
```

- sample.glue\_uml\_ad



- Context Clear
- WebPage Tag
- ExcelInfo
- WebSocket DirectSender**

- **sample.activity.Dummy**

```
package sample.activity;

import com.poscoict.glueframework.biz.activity.GlueActivity;
import com.poscoict.glueframework.biz.control.GlueBizControlConstants;
import com.poscoict.glueframework.context.GlueContext;

public class Dummy extends GlueActivity<GlueContext>{
    // Do not define the Class Member Variable!!
    public String runActivity( GlueContext ctx )
    {
        this.logger.info("Dummy, runActivity!");
        return GlueBizControlConstants.SUCCESS;
    }
}
```

- 주기적으로 text를 전송해주는 simulator이다.
- **sample.publisher.AutoPublisher (1/5)**

```
package sample.publisher;

import java.util.Map;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.DisposableBean;
import org.springframework.beans.factory.InitializingBean;

import com.poscoict.glueframework.websocket.GlueAbstractWebSocketHandler;
import com.poscoict.glueframework.websocket.GlueWebSocketDirectSender;

public class AutoPublisher implements GlueWebSocketDirectSender, InitializingBean,
    DisposableBean, Runnable {
    GlueAbstractWebSocketHandler handler;
    String id;
    private Logger logger = LoggerFactory
        .getLogger(AutoPublisher.class);

    private Map<String, String> propertiesMap;
    private Thread thisThread;

    public AutoPublisher() {
        super();
    }
}
```



- **sample.publisher.AutoPublisher (2/5)**

```
@Override
public void setWebSocketHandler(GlueAbstractWebSocketHandler handler) {
    this.handler = handler;
}

@Override
public void receiveDirectMessage(String msg) {
    //Do nothing
    logger.info("[RCV TXT from handler]" + msg);
}

@Override
public void receiveDirectMessage(byte[] msg) {
    //Do nothing
}

@Override
public void run() {
    String txtMsg = null;
    int index = 0;
    while (true) {
        txtMsg = "From Thread , msg : 你好 안녕하세요 test version :" +
index;
        index++;
    }
}
```

- `sample.publisher.AutoPublisher (3/5)`

```
        try {
            logger.info(this.id + ":" + txtMsg);
            if( handler != null ){
                this.handler.handleDirectMessage("TEST",
txtMsg.getBytes("UTF-8") );
            }
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    @Override
    public void afterPropertiesSet() throws Exception {

        String ip = propertiesMap.get("ip").toString();
        int port = -1;
        int timeout = 60000;
```

- `sample.publisher.AutoPublisher (4/5)`

```
        try {
            port = Integer.parseInt((String)propertiesMap.get("port"));
        } catch (Exception e) {
            port = 12001;
        }

        try {
            timeout =
Integer.parseInt((String)propertiesMap.get("timeout"));
        } catch (Exception e) {
            port = 12001;
        }

        this.thisThread = new Thread( this );
        this.thisThread.start();

        logger.info("***** [" + this.id + "] AutoPublisher
STARTED *****");
    }

    public void setPropertiesMap( Map<String, String> propertiesMap )
    {
        this.propertiesMap = propertiesMap;
        this.id = propertiesMap.get("id");
        if(id == null){
            this.id = "Undefined";
        }
    }
}
```

- **sample.publisher.AutoPublisher (5/5)**

```
setPropertiesMap ***** logger.info(" ***** [" + this.id + "]AutoPublisher
    }

    @Override
    public void destroy() throws Exception {
        logger.info("AutoPublisher DESTROYED");
    }

    @Override
    public void receiveCommand(String cmd) {
        // TODO Auto-generated method stub
        if( "stop".equals(cmd)){
            this.currentThread.interrupt();
        }

    }

    @Override
    public GlueAbstractWebSocketHandler getHandler() {
        return this.handler;
    }
}
```



- client.jsp (2/3)

```
        console.log('receive message : ' + message.data);
        $('#messages').val($('#messages').val() + message.data + '\n');
        document.getElementById('messages').scrollTop =
document.getElementById('messages').scrollHeight;
    };
    ws.onclose = function (event) {
        console.log(event);
        console.log('websocket closed');
    };
}

function disconnect() {
    if (ws) {
        ws.close();
        ws = null;
    }
}

$(function () {
    $('#connect').click(function () {
        connect();
        $(this).attr('disabled', true);
        $('#disconnect').removeAttr('disabled');
    });
});
```

- client.jsp (3/3)

```
    $('#disconnect').click(function () {
        disconnect();
        $(this).attr('disabled', true);
        $('#connect').removeAttr('disabled');
    });

    $('#message').keydown(event, function () {
        if (event.keyCode === 13) {
            ws.send($(this).val());
            $(this).val('');
        }
    });
});
</script>

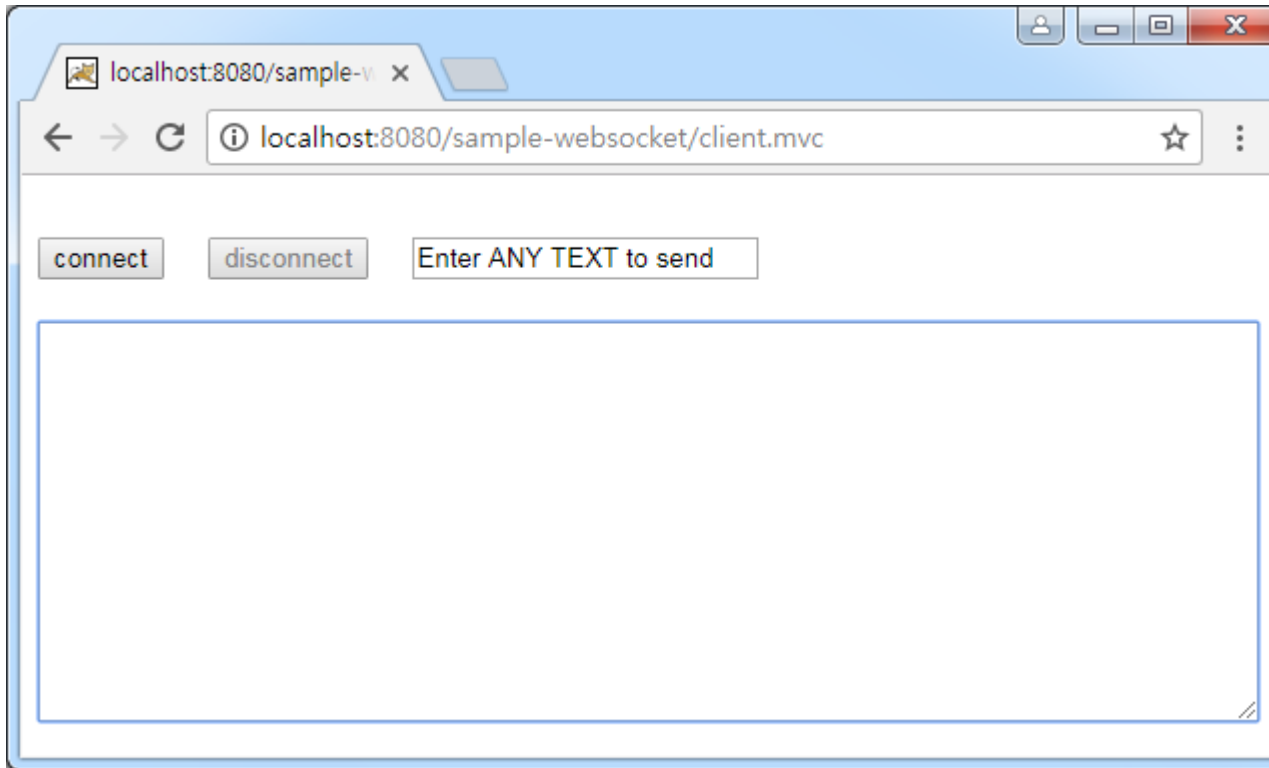
</body>
</html>
```

- sender.jsp

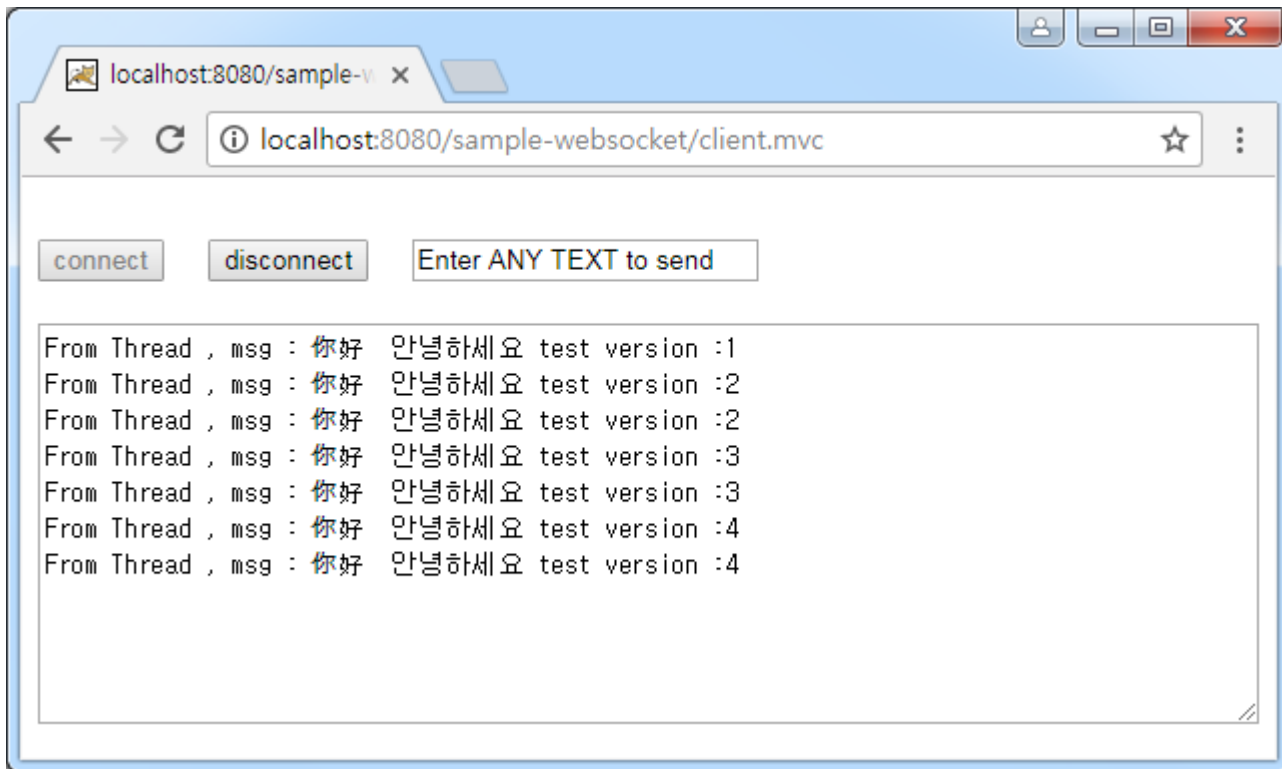
```
<%@ page contentType="text/html; charset=EUC-KR"%>
<%@ page import="com.poscoict.glueframework.context.GlueContext" %>
<%@ page import="com.poscoict.glueframework.web.control.GlueWebConstants" %>
<%@ page import="com.poscoict.glueframework.biz.control.GlueBizControlConstants" %>
<%@ page import="java.util.*" %>
<html>
<head>
<title>Sample</title>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<body bgcolor="#FFFFFF" text="#000000" topmargin=10 leftmargin=10>
<form name="form1" method="post" action="sender.mvc">
<b> WebSocket sample </b>
<hr>
<input type="submit" name="sendTxt" value="Send text message">
<hr>
Glue Service Name : <input type="text" name="ServiceName" value="sample-service">
</form>
</body>
</html>
```



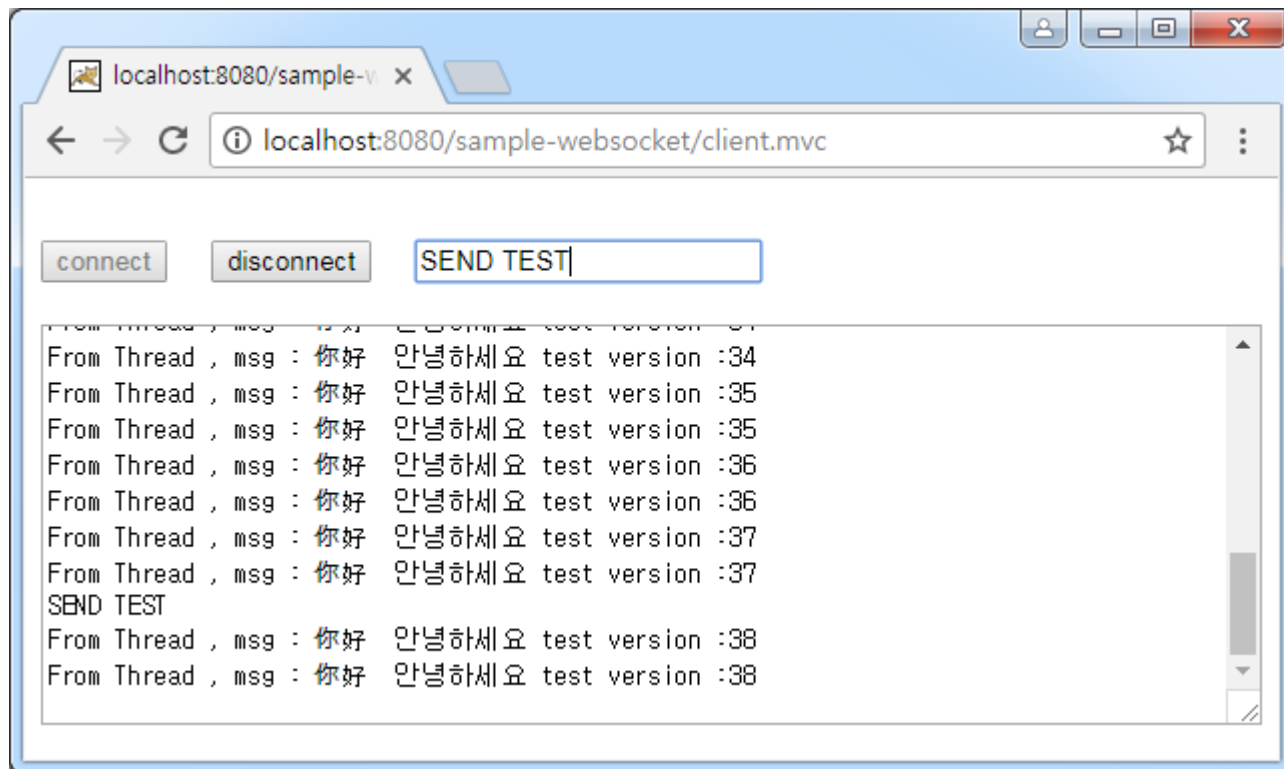
- **사용법**
  - WAR 를 build 하여 local WAS 에 deploy 한 후, 브라우저에서 아래의 경로로 접속



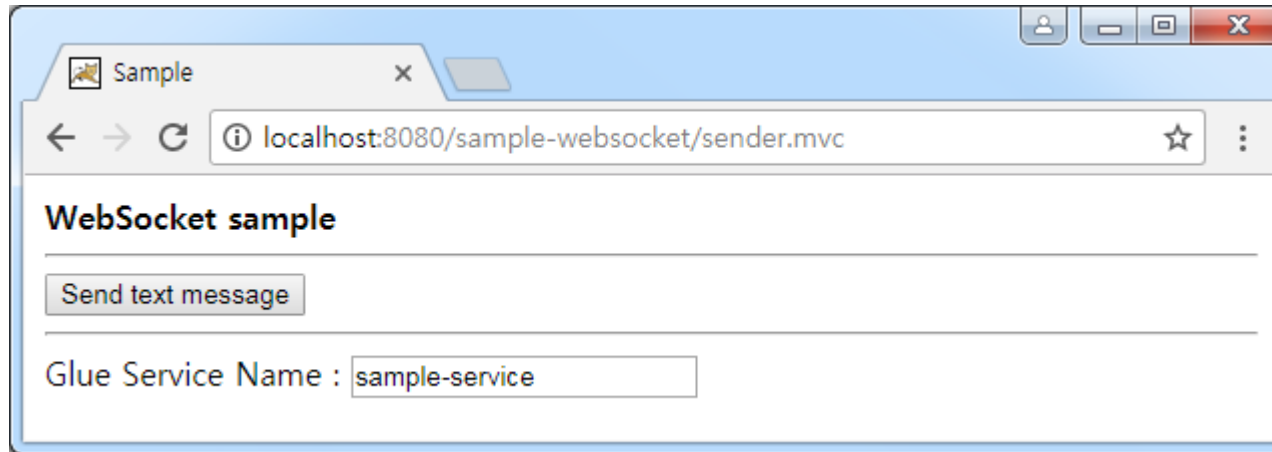
- **사용법**
  - connect 버튼을 클릭한다. 5초 정도 기다리면 자동으로 test 문자가 보여진다.
  - AutoPublisher Thread가 보내는 Direct Message



- **사용법**
  - TextForm 안에 전송하고 싶은 text를 입력하고 엔터를 친다
  - Test 입력 문자를 입력하면 N개의 client에 전송된다

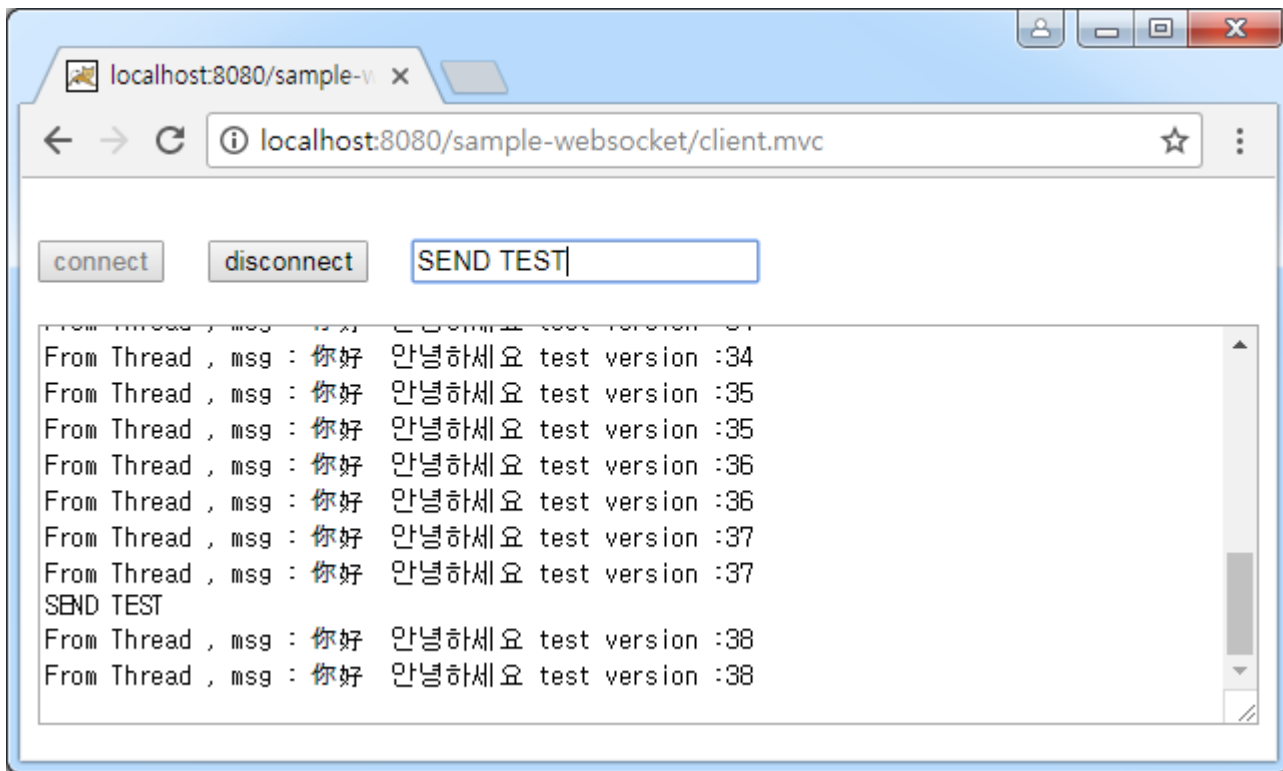


- **사용법**
  - 또 다른 브라우저에 아래의 경로로 접속한다.



- Send text message 버튼을 클릭하고 client 창을 확인하면 Sender가 보낸 message가 수신되어 표시된다.

- 사용법
  - Glue service (activity) 를 통해서 보내지는 Direct Message





# IV. Glue Framework 제작 가이드

## 3장 심화 학습

### 4. Batch Job 만들기





## 목표

## 실습

- batch job 생성용 project 구성

- 기존 프로젝트를 참고하여 필수관련소스로만 경량화 하여 구성

- 정주기에 실행되는 batch job을 구현할 수 있다.
- Batch job용으로 jar를 build 할 수 있다.

- build.xml 구성

- Spring Scheduler 를 사용하여 batch job을 실행 할 수 있다.

- spring-scheduler.xml 만들기.

- Quartz를 사용하여 batch job을 실행 할 수 있다.

- quartz-scheduler.xml 만들기

- 7개의 단위 표현식으로 구성된 문자열
  - 초 (Seconds) – 0에서 59 사이의 숫자
  - 분 (Minutes) – 0에서 59 사이의 숫자
  - 시 (Hours) – 0에서 23 사이의 숫자
  - 일 (Day of month) – 1에서 31 사이의 숫자
  - 월 (Months) – 1에서 12사이의 숫자, 또는 JAN, FEB, .. , DEC
  - 주 (Days of week) – 1에서 7사이의 숫자, 또는 MON, TUE, .. , SUN
  - 연 (Year) - 생략 가능
  - 특수문자 : [\*][?][-],[/][L]

```
0 15 12 * * ? : 매일 12시 15분마다 실행 (모든값[*], 설정값없음[?])
0 15 12 ? * * 2015 : 2015년도에 모든 요일마다 12시 15분마다 실행
0 15 12 * * ? 2010-2014 : 2010~14년도에 매일 12시 15분마다 실행(범위[-])
0 0 8 ? * MON,FRI : 월금 8시마다(나열[,])
0 0/30 8-9 * * * : 매일 8시 10시 사이에 30분 간격으로 실행(간격[/])
                        8:00, 8:30, 9:00, 9:30
0 0 23 L * ? : 매월 마지막날 23시에 실행 [마지막일, 마지막요일[L])
```

- <http://www.quartz-scheduler.org/documentation/quartz-2.2.x/tutorials/crontrigger.html>

- `GlueSDK\lib\gluelib\glue-schedule-x.x.x.jar`
- `GlueSDK\lib\gluestd_schedule\backport-util-concurrent-3.0.jar`
- `GlueSDK\lib\gluestd_schedule\mina-core-1.0.10.jar`
- `GlueSDK\lib\gluestd_schedule\quartz-2.2.1.jar`

- batch job 생성용 Glue Java Project 생성 : sample-scheduler
- 기존 프로젝트에서 관련 service 파일 복사
  - sample/src/service/hello-service.xml, dept-service.xml 를
  - sample-scheduler/src/service 폴더로 복사
- 관련 Activity 파일 복사
  - /src/sample/activity/Dept\*.java
  - /src/sample/HelloActivity.java
- applicationContext.xml
  - test-datasource url 수정, maxActive 1로 수정(sqlite db lock 방지)
  - messageSource 삭제
  - logback-test.xml 에서 logger 추가 (sample 패키지)
- batch job 에서 사용하지 않는 불필요한 파일 삭제
  - sample-scheduler/src/default package 아래 java 파일 삭제
  - layout 폴더 삭제 (전문사용할 경우 유지)
  - MyResource\*.properties 삭제
  - WebContents 폴더 삭제

- build.xml (1/2)

```
<?xml version="1.0"?>
<project name="GlueSample" default="jar">
  <property file="build.properties"/>
  <path id="class.path">
    <fileset dir="${GlueSDK.dir}/lib/gluelib"><include name="*.jar"/></fileset>
    <fileset dir="${GlueSDK.dir}/lib/gluestd"><include name="*.jar"/></fileset>
    <fileset dir="${GlueSDK.dir}/lib/gluestd_schedule"><include name="*.jar"/></fileset>
    <!-- add here -->
  </path>
  <pathconvert property="class-path" pathsep=" " dirsep="/">
    <path refid="class.path"></path>
    <map from="${GlueSDK.dir}/lib/gluelib" to="lib"/>
    <map from="${GlueSDK.dir}/lib/gluestd_schedule" to="lib"/>
    <map from="${GlueSDK.dir}/lib/gluestd" to="lib"/>
  </pathconvert>
  <target name="init" description="clean build dir before packaging">
    <delete failonerror="false">
      <fileset dir="${build.dir}">
        <include name="*.*"/>
      </fileset>
    </delete>
    <mkdir dir="${build.dir}"/>
    <mkdir dir="${build.dir}/lib"/>
  </target>

```

- build.xml (2/2)

```
<target name="jar" depends="init" description="Creates the JAR file">
  <jar destfile="${build.dir}/user-application.jar">
    <manifest>
      <section name="POSCO DX Glue Sample Project">
        <attribute name="Implementation-Version" value="${project_version}"/>
      </section>
      <attribute name="Main-Class"
value="com.poscoict.glueframework.scheduling.server.GlueSchedulerHttpServer"/>
      <attribute name="Class-Path" value="${class-path}" />
    </manifest>
    <fileset dir="${class.dir}">
      <patternset>
        <include name="**/*.*" />
      </patternset>
    </fileset>
  </jar>
  <copy todir="${build.dir}/lib">
    <fileset dir="${GlueSDK.dir}/lib/gluelib"><include name="*.jar"/></fileset>
    <fileset dir="${GlueSDK.dir}/lib/gluestd"><include name="*.jar"/></fileset>
    <fileset dir="${GlueSDK.dir}/lib/gluestd_schedule"><include
name="*.jar"/></fileset>
  </copy>
</target>
</project>
```

- spring-scheduler.xml (1/2)

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:task="http://www.springframework.org/schema/task"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/task
    http://www.springframework.org/schema/task/spring-task.xsd">

  <task:scheduled-tasks scheduler="scheduler">
    <task:scheduled ref="glue-task-1" method="doGlueService" cron="0/8 * * * * MON-
FRI"/>
    <task:scheduled ref="glue-task-2" method="doGlueService" fixed-rate="70000"/>
    <task:scheduled ref="glue-task-3" method="doGlueService" fixed-delay="50000"/>
  </task:scheduled-tasks>

  <task:scheduler id="scheduler" pool-size="3"/>

  <bean id="glue-task-1"
    class="com.poscoict.glueframework.scheduling.task.GlueTaskScheduler">
    <property name="ServiceName" value="hello-service"/>
  </bean>
```

- spring-scheduler.xml (2/2)

```
<bean id="glue-task-2"
      class="com.poscoict.glueframework.scheduling.task.GlueTaskScheduler">
  <property name="ServiceName" value="dept-service"/>
  <property name="dataMap">
    <map>
      <entry key="DEPTNO" value="50"/>
      <entry key="DNAME" value="Marketing"/>
      <entry key="LOC" value="SEOUL"/>
      <entry key="insert" value="event"/>
    </map>
  </property>
</bean>
<bean id="glue-task-3"
      class="com.poscoict.glueframework.scheduling.task.GlueTaskScheduler">
  <property name="ServiceName" value="dept-service"/>
  <property name="dataMap">
    <map>
      <entry key="DEPTNO" value="50"/>
      <entry key="delete" value="event"/>
    </map>
  </property>
</bean>
</beans>
```



- quartz-scheduler.xml (1/3)

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
  <bean id="scheduler"
    class="org.springframework.scheduling.quartz.SchedulerFactoryBean">
    <property name="triggers">
      <list>
        <ref local="simpleTrigger1"/>
        <ref local="simpleTrigger2"/>
        <ref local="cronTrigger"/>
      </list>
    </property>
  </bean>

  <bean id="cronTrigger"
    class="org.springframework.scheduling.quartz.CronTriggerFactoryBean">
    <property name="jobDetail" ref="jobDetail-A"/>
    <property name="cronExpression" value="0 40 15 * * ?"/>
  </bean>
</beans>
```

- quartz-scheduler.xml (2/3)

```
<bean id="simpleTrigger1"
      class="org.springframework.scheduling.quartz.SimpleTriggerFactoryBean">
    <property name="jobDetail" ref="jobDetail-B"/>
    <property name="repeatInterval" value="10000"/>
    <property name="repeatCount" value="3"/>
    <property name="startDelay" value="5000"/>
</bean>
<bean id="simpleTrigger2"
      class="org.springframework.scheduling.quartz.SimpleTriggerFactoryBean">
    <property name="jobDetail" ref="jobDetail-C"/>
    <property name="repeatInterval" value="7000"/>
</bean>

<bean id="jobDetail-A"
      class="org.springframework.scheduling.quartz.JobDetailFactoryBean">
    <property name="jobClass"
              value="com.poscoict.glueframework.scheduling.GlueQuartzJobBean"/>
    <property name="jobDataAsMap">
        <map>
            <entry key="ServiceName" value="hello-service"/>
        </map>
    </property>
</bean>
```

- quartz-scheduler.xml (3/3)

```
<bean id="jobDetail-B"
      class="org.springframework.scheduling.quartz.JobDetailFactoryBean">
  <property name="jobClass"
            value="com.poscoict.glueframework.scheduling.GlueQuartzJobBean"/>
  <property name="jobDataAsMap">
    <map>
      <entry key="ServiceName" value="dept-service"/>
      <entry key="DEPTNO" value="60"/>
      <entry key="DNAME" value="Testing"/>
      <entry key="LOC" value="seoul"/>
      <entry key="insert" value="event"/>
    </map>
  </property>
</bean>
<bean id="jobDetail-C"
      class="org.springframework.scheduling.quartz.JobDetailFactoryBean">
  <property name="jobClass"
            value="com.poscoict.glueframework.scheduling.GlueQuartzJobBean"/>
  <property name="jobDataAsMap">
    <map>
      <entry key="ServiceName" value="dept-service"/>
      <entry key="DEPTNO" value="60"/>
      <entry key="delete" value="event"/>
    </map>
  </property>
</bean>
</beans>
```

- 해당 프로젝트 build 폴더로 이동, 명령 창 실행
  - 예) cd C:\eclipse\workspace\sample-scheduler\build

- JVM으로 batch job 실행

- Quartz 사용시

```
java -jar user-application.jar quartz quartz-scheduler.xml
```

- Spring Scheduler 사용 시

```
java -jar user-application.jar spring spring-scheduler.xml
```

# IV. Glue Framework 제작 가이드

## 3장 심화 학습

### 5. Glue Job Scheduler용 User Job 만들기



---

## 목표

## 실습

- 
- Glue Job Scheduler 에 등록하여 실행할 수 있는 user job을 유형별로 build 할 수 있다.
  - build.xml 수정
-

- user job(biz app) 패키징
  - [https://www.solutionpot.co.kr/doc/scheduler/glue-scheduler-job/dev\\_job\\_packaging.html](https://www.solutionpot.co.kr/doc/scheduler/glue-scheduler-job/dev_job_packaging.html)
- 테스트 방법
  - scheduler server 가 user job 을 기동하는 방식대로 테스트
  - user job(biz app) 을 sample 프로젝트의 build 디렉토리로 복사
  - user job 관련 라이브러리를 build/lib 아래에 복사
  - Glue Job Scheduler 패키지에서 glue-scheduler-job-1.2.4-RELEASE.jar 복사
  - 실행유형이 class 일 경우 아래와 같이 job 실행 가능
    - **java** -Dlog.file=C:/logs/test.log **-cp ./sample.jar;./lib/\*;**  
**com.poscoict.app.job.GlueSimpleJob** -l "ServiceName=job001-service@S" "sleep-second=30@S"
  - 실행유형이 jarfile 일 경우 아래와 같이 job 실행 가능
    - **java** -Dlog.file=C:/logs/test.log **-jar user-application.jar** -l "ServiceName=job001-service@S" "sleep-second=30@S"
  - 위의 2가지 유형의 차이점은 main-class, class-path 지정 여부



- 실행유형이 class일 때

Option Type	Option Key	Option Value
-D 옵션	CONFIG_PATH	C:\scheduler-server\apps\conf
기타	directory	C:\scheduler-server\logs
classpath	biz-app	C:\scheduler-server\apps\sample1.jar
classpath	ref-all	C:\scheduler-server\apps\lib\*
JavaPath	java	C:\Program Files\Java\jdk1.7.0_25\bin\java
실행유형	class	com.poscoict.app.job.GlueSimpleJob

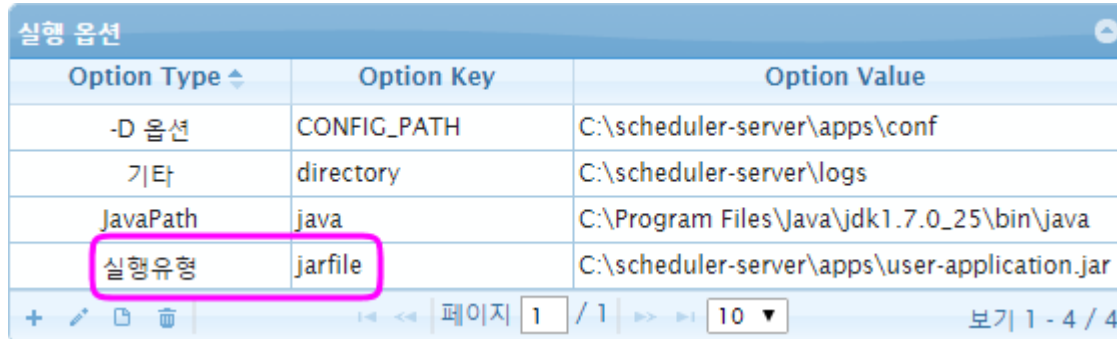
- build.xml (1/2)

```
<project name="GlueSample" default="jar" basedir=". ">
  <property file="build.properties"/>
  <target name="init">
    <mkdir dir="${basedir}/build"/>
    <mkdir dir="${basedir}/buildfile"/>
  </target>
  <target name="javac" depends="init">
    <javac srcdir="${basedir}/src" destdir="${basedir}/buildfile" encoding="UTF-8"
includeantruntime="false">
      <classpath>
        <pathelement location="${GlueSDK.dir}/lib/gluelib/glue-core-5.0.0-
RELEASE.jar" />
```

- **build.xml (2/2)**

```
RELEASE.jar" /> <pathelement location="${GlueSDK.dir}/lib/gluelib/glue-schedule-5.0.0-  
    <pathelement location="${GlueSDK.dir}/lib/gluestd/slf4j-api-1.7.7.jar" />  
    </classpath>  
    </javac>  
</target>  
<target name="jar" depends="javac">  
    <jar destfile="${basedir}/build/sample1.jar">  
        <fileset dir="${basedir}/buildfile">  
            <patternset>  
                <include name="**/*.*" />  
            </patternset>  
        </fileset>  
        <fileset dir="${basedir}/src">  
            <patternset>  
                <include name="applicationContext.xml" />  
                <include name="ehcache.xml" />  
                <include name="logback-test.xml" />  
                <include name="layout/*.xml" />  
                <include name="query/*.glue_sql" />  
                <include name="service/*.xml" />  
            </patternset>  
        </fileset>  
    </jar>  
</target>  
</project>
```

- 실행유형이 jarfile 일 때



Option Type	Option Key	Option Value
-D 옵션	CONFIG_PATH	C:\scheduler-server\apps\conf
기타	directory	C:\scheduler-server\logs
JavaPath	java	C:\Program Files\Java\jdk1.7.0_25\bin\java
실행유형	jarfile	C:\scheduler-server\apps\user-application.jar

- MANIFEST.MF 파일에 Main-Class 와 Class-Path 가 포함되어야 한다

- build.xml (1/3)

```
<project name="GlueSample" default="user-application" basedir=". ">
  <property file="build.properties"/>
  <target name="init">
    <mkdir dir="${basedir}/build"/>
    <mkdir dir="${basedir}/build/lib"/>
    <mkdir dir="${basedir}/buildfile"/>
  </target>
  <path id="class.path">
    <fileset dir="${GlueSDK.dir}/lib/gluelib" >
      <include name="*.jar"/>
    </fileset>
  </path>
</project>
```

- build.xml (2/3)

```
        <fileset dir="${GlueSDK.dir}/lib/gluestd" >
            <include name="*.jar"/>
        </fileset>
        <!-- add here .. -->
    </path>
    <pathconvert property="class-path" pathsep=" " dirsep="\">
        <path refid="class.path"/>
        <map from="${GlueSDK.dir}/lib/gluestd" to="lib"/>
        <map from="${GlueSDK.dir}/lib/gluelib" to="lib"/>
        <!-- add here .. -->
    </pathconvert>
    <target name="javac" depends="init">
        <javac srcdir="${basedir}/src" destdir="${basedir}/buildfile" encoding="UTF-8"
includeantruntime="false">
            <classpath>
                <pathelement location="${GlueSDK.dir}/lib/gluelib/glue-core-5.0.0-
RELEASE.jar" />
                <pathelement location="${GlueSDK.dir}/lib/gluelib/glue-schedule-5.0.0-
RELEASE.jar" />
                <pathelement location="${GlueSDK.dir}/lib/gluestd/slf4j-api-1.7.7.jar" />
            </classpath>
        </javac>
    </target>
    <target name="user-application" depends="javac">
        <jar destfile="${basedir}/build/sample2.jar">
            <manifest>
```

- build.xml (3/3)

```
        <attribute name="Main-Class" value="com.poscoict.app.job.GlueSimpleJob"/>
        <attribute name="Class-Path" value="${class-path}" />
    </manifest>
    <fileset dir="${basedir}/buildfile">
        <patternset>
            <include name="**/*.xml"/>
        </patternset>
    </fileset>
    <fileset dir="${basedir}/src">
        <patternset>
            <include name="applicationContext.xml"/>
            <include name="ehcache.xml"/>
            <include name="logback-test.xml"/>
            <include name="layout/*.xml"/>
            <include name="query/*.glue_sql"/>
            <include name="service/*.xml"/>
        </patternset>
    </fileset>
</jar>
</target>
</project>
```

- build 전에 SearchEmp.java, EmpVO.java 한글 encoding 확인
- default package java 파일 삭제 (glue-tester.jar 필요)



# V. Support 안내



License 관련 .....	243
Supervising Service .....	244
Documents .....	245







- License 종류


Mode	용도	특징	만료 시
EDUCATION	교육용	라이선스 키가 없을 때 기본 모드. 서비스 실행 횟수(50회)로 체크.	WAS 재기동 하면 횟수 초기화
DEVELOPMENT	개발용	유효 기간 체크	서비스 수행 시 10초 지연 
PRODUCT	운영용	유효 MAC Address 체크	서비스 수행 시 10초 지연 

- License 발급 요청 방법

- SmartFactory 운영섹션에 메일로 라이선스 발급요청 (김원준 프로) 

- 관련 오류 시 참고

- <https://www.solutionpot.co.kr/doc/framework/faq.html>
- [https://www.solutionpot.co.kr/doc/framework/troubleshooting\\_1.html](https://www.solutionpot.co.kr/doc/framework/troubleshooting_1.html)
- DCONFIG\_PATH 에 라이선스 키 파일(xml, license)이 위치해야 함.
- JBoss, Weblogic 의 경우 : [www.solutionpot.co.kr](http://www.solutionpot.co.kr) - Glue - FAQ - 2,3 번 게시물 참고

- License 구매 시 Glue Framework 에 대해 서비스 지원
- Glue Framework 관련 질문에 대한 우선순위
  -  1. 솔루션포털 ([www.solutionpot.co.kr](http://www.solutionpot.co.kr)) - guest 글쓰기 가능
  - 2. 이메일
  - 3. 포스톡
  - 4. 전화
- 솔루션포털 질문 작성 시 유의점
  - Q&A, FAQ 에서 관련 내용 먼저 검색
  - 프로젝트 및 작성자 이름과 이슈에 대한 구체적인 설명
  - 로그 첨부 시 앞뒤 내용을 여유롭게 캡처
  - Glue Framework 관련내용이 아닌 질문은 답변 대상이 아님.

- **Online** Documents

- <http://www.solutionpot.co.kr/doc/>
- GlueFramework 4, 5 로 분류
- 버전업 될 경우, Release History 참고
- 버전별 Example 페이지에 다양한 예제 파일

- **Offline** Documents

- GlueSDK/GlueAPI 아래 압축파일로 존재
- apidocs-aggregate : Glue Framework java API document
- site-javadoc : Glue Framework offline document



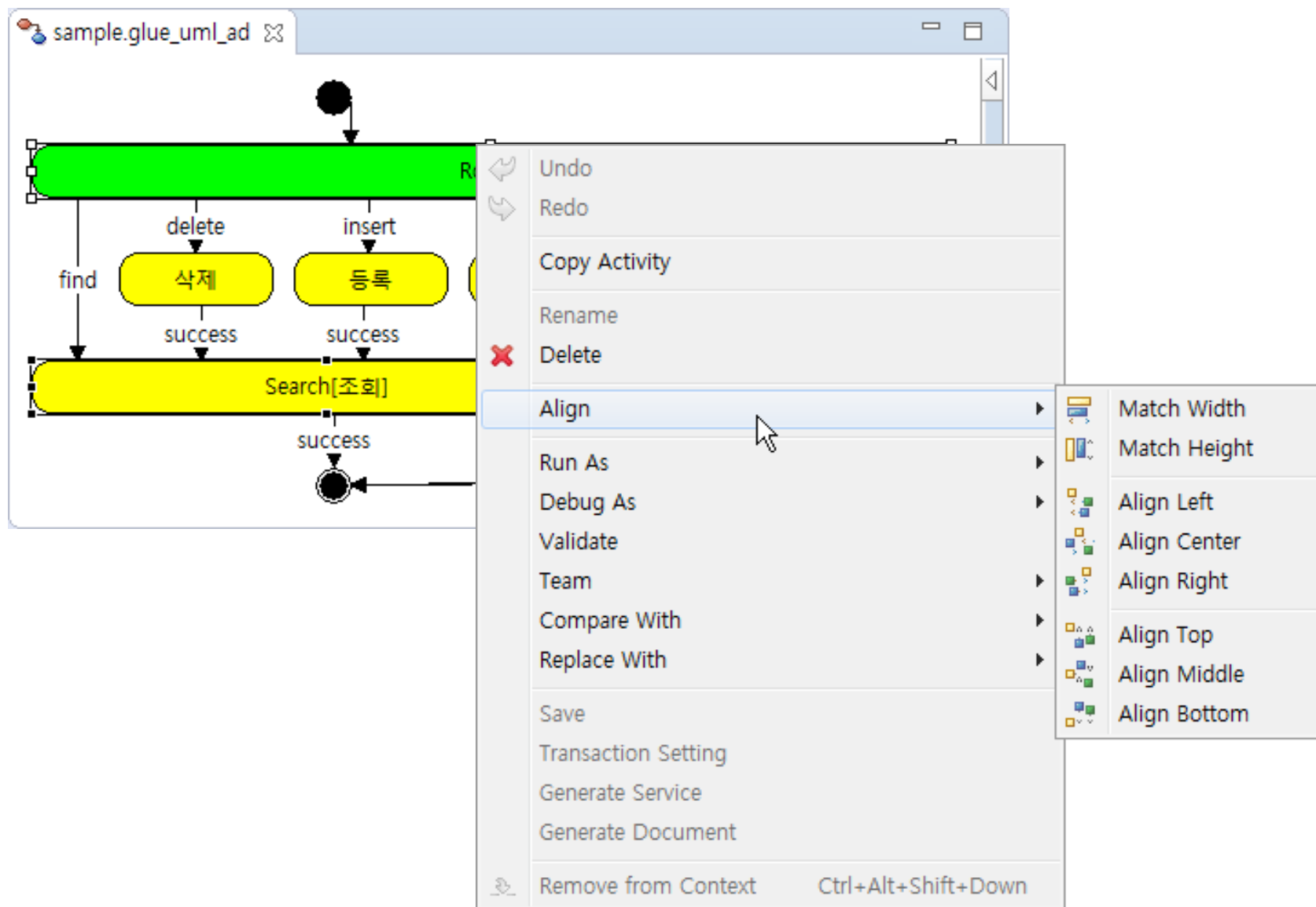
## VI. 별첨

개발프로세스 정의 .....	249
Activity Diagram 정렬 .....	250
glue-config.xml .....	252
Service 명세서 생성방법 .....	256
Exception .....	258
기타 .....	261

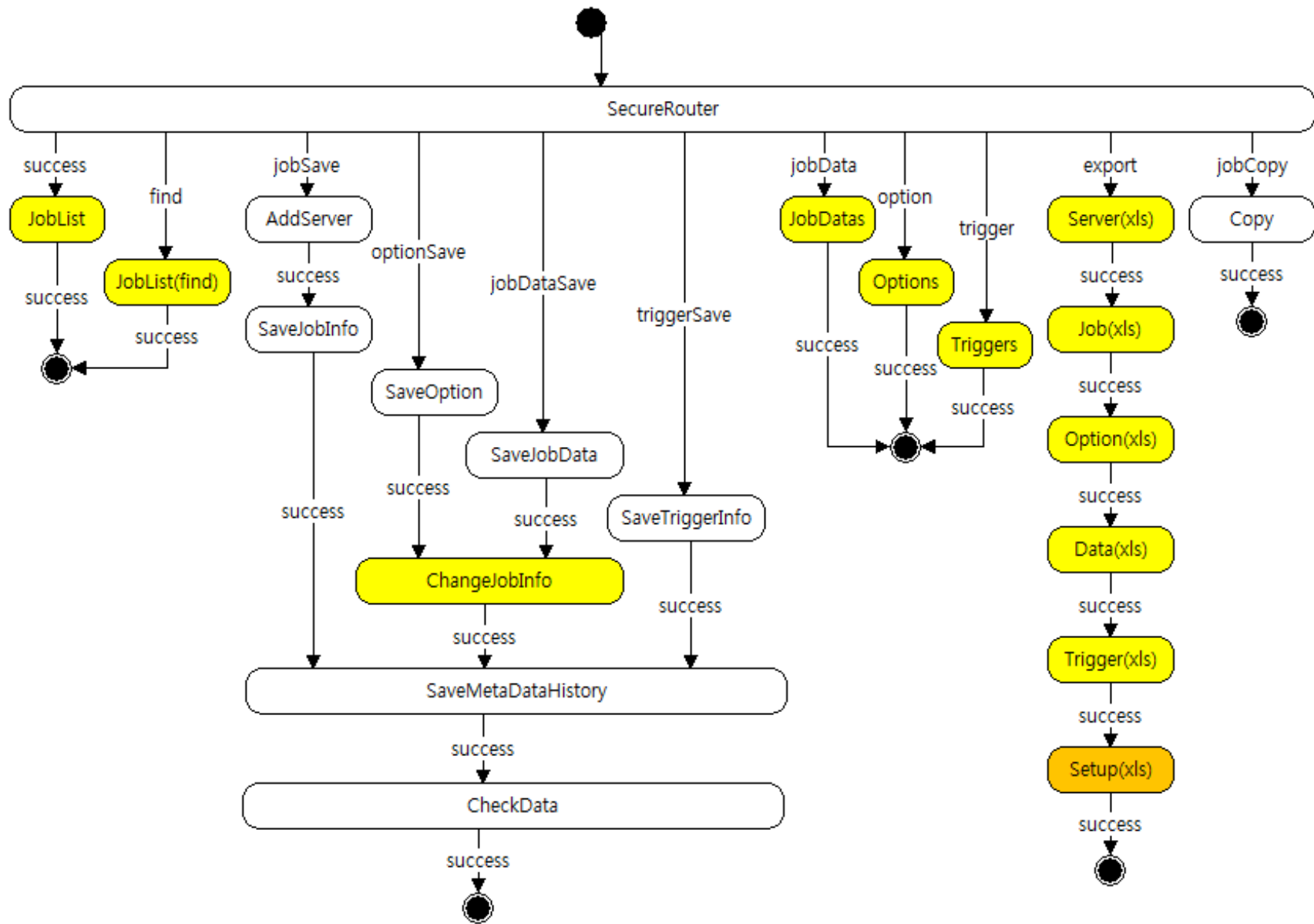


단계	주요 수행 업무	주요 산출물
<div style="border: 1px solid black; padding: 5px; width: 60px; margin: 0 auto;">분석</div> <div style="text-align: center; margin: 10px 0;">↓</div>	<p>현행 업무 및 시스템을 분석하고, 사용자 요구사항을 도출하며, 이를 토대로 업무 프로세스, 이벤트 모델링, 보안 관련 분석을 함</p>	<p>고객요구정의서, 현행시스템 분석서, 시스템 아키텍처 등</p>
<div style="border: 1px solid black; padding: 5px; width: 60px; margin: 0 auto;">설계</div> <div style="text-align: center; margin: 10px 0;">↓</div>	<p>사용자 요구사항 및 분석 결과에 근거하여 업무기능, 사용자 인터페이스 및 내/외부 인터페이스 등을 구현 가능한 수준으로 설계</p>	<p>업무, 어플리케이션, 화면, 시스템 인터페이스, DB, 보완, 데이터 이행 설계서</p> <p>Page Flow Diagram, <b>Activity Diagram</b>  <b>화면 입출력사양서, 화면 Layout (HTML)</b>  <b>Query정의서, 전문Layout</b></p>
<div style="border: 1px solid black; padding: 5px; width: 60px; margin: 0 auto;">구현</div> <div style="text-align: center; margin: 10px 0;">↓</div>	<p>설계에 따라 응용시스템 기능의 충분성, 완전성, 무결성, 편의성, 적정성을 확보할 수 있도록 구현하고, 단위 기능에 대한 검증을 수행</p>	<p><b>프로그램 Source Code, 단위테스트 계획서 및 결과서</b></p>
<div style="border: 1px solid black; padding: 5px; width: 60px; margin: 0 auto;">시험</div> <div style="text-align: center; margin: 10px 0;">↓</div>	<p>통합시험, 시스템 시험을 통하여 구현된 시스템이 통합관점에서의 기능 완전성과 성능, 안정성, 보완성 확보 여부를 검증</p>	<p>결합/시스템/운영 테스트 계획서 및 결과서, 운영교육 계획서 및 결과서, 사용자/시스템 운영 매뉴얼</p>
<div style="border: 1px solid black; padding: 5px; width: 60px; margin: 0 auto;">전개</div>	<p>시스템을 운영하기 위한 시스템 설치 및 배포, 초기 데이터 구축 등의 준비를 완료하고, 시스템이 사용자에게 이관 및 운영될 수 있도록 준비</p>	<p>이행 계획서 및 결과서</p>

※ 위 개발프로세스는 권고사항이며, 프로젝트의 개발 방법론 및 PM에 따라서 변경 가능함







- activities 와 common 으로 구성.
- glue-schema-5.x.x.jar 의 Schema 참고
  - [schemaorg\\_apache\\_xmlbeans/src/GluePluginConfig.xsd](http://www.poscoict.com/glueframework/plugin/config)



```
<?xml version="1.0" encoding="UTF-8"?>
<glue-eclipse-config
  xmlns="http://www.poscoict.com/glueframework/plugin/config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.poscoict.com/glueframework/plugin/config
    GluePluginConfig.xsd ">
  <activities>
    <activity name="Router" icon="icons/activity_router.gif" addSeperator="true">
    <activity name="SubService" icon="icons/activity_subservice.gif">
    <activity name="Jdbc Search" icon="icons/activity_search.gif">
    <activity name="Jdbc Insert" icon="icons/activity_insert.gif">
    <activity name="Jdbc Modify" icon="icons/activity_modify.gif">
    <activity name="Jdbc Delete" icon="icons/activity_delete.gif">
    <activity name="Message Parse" icon="icons/components.gif" addSeperator="true">
    <activity name="Message Create" icon="icons/components.gif">
  </activities>
  <common>
    <document>
    <service>
    <query>
    <class-generator>
    <activity-template>
  </common>
</glue-eclipse-config>
```

- Reusable Activity 관리
- icon : icons/파일명 으로 정의함. 허용 파일명은 그림 참고.
- addSeparator : Palette의 구분 선을 정의. Reuse Activity의 Grouping 용도.
- color : 허용 color는 schema 참고.

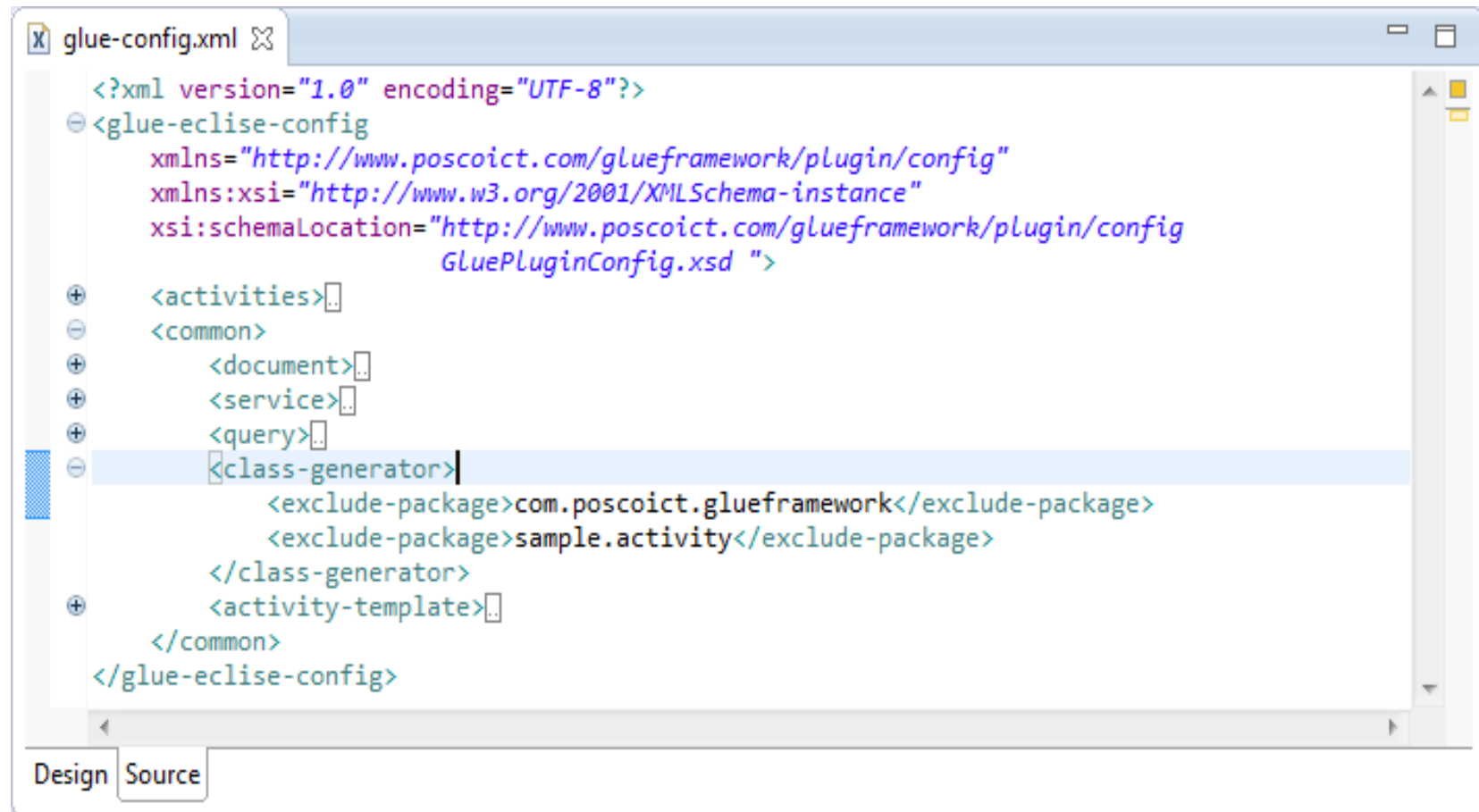
The screenshot displays an IDE window titled 'glue-config.xml' with the following XML content:

```
<?xml version="1.0" encoding="UTF-8"?>
<glue-eclipse-config
  xmlns="http://www.poscoict.com/glueframework/plugin/config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.poscoict.com/glueframework/plugin/config
    GluePluginConfig.xsd ">
  <activities>
    <activity name="Router" icon="icons/activity_router.gif" addSeperator="true">
      <bubble-tip>Default Router Activity</bubble-tip>
      <template>Glue Router template</template>
      <color>green</color>
      <default-properties>
        <property name="class" value="com.poscoict.glueframework.biz.activity.GlueDefaultRouter" />
      </default-properties>
    </activity>
    <activity name="SubService" icon="icons/activity_subservice.gif">
      <bubble-tip>SubService Activity</bubble-tip>
      <template>Glue SubService template</template>
      <color>lightBlue</color>
      <default-properties>
        <property name="class" value="com.poscoict.glueframework.biz.activity.GlueSubService" />
        <property name="service-name" value="sample-service" />
        <property name="new-transaction" value="false" />
      </default-properties>
  </activities>
</glue-eclipse-config>
```

On the right side, a palette shows various activity icons with their corresponding file names:

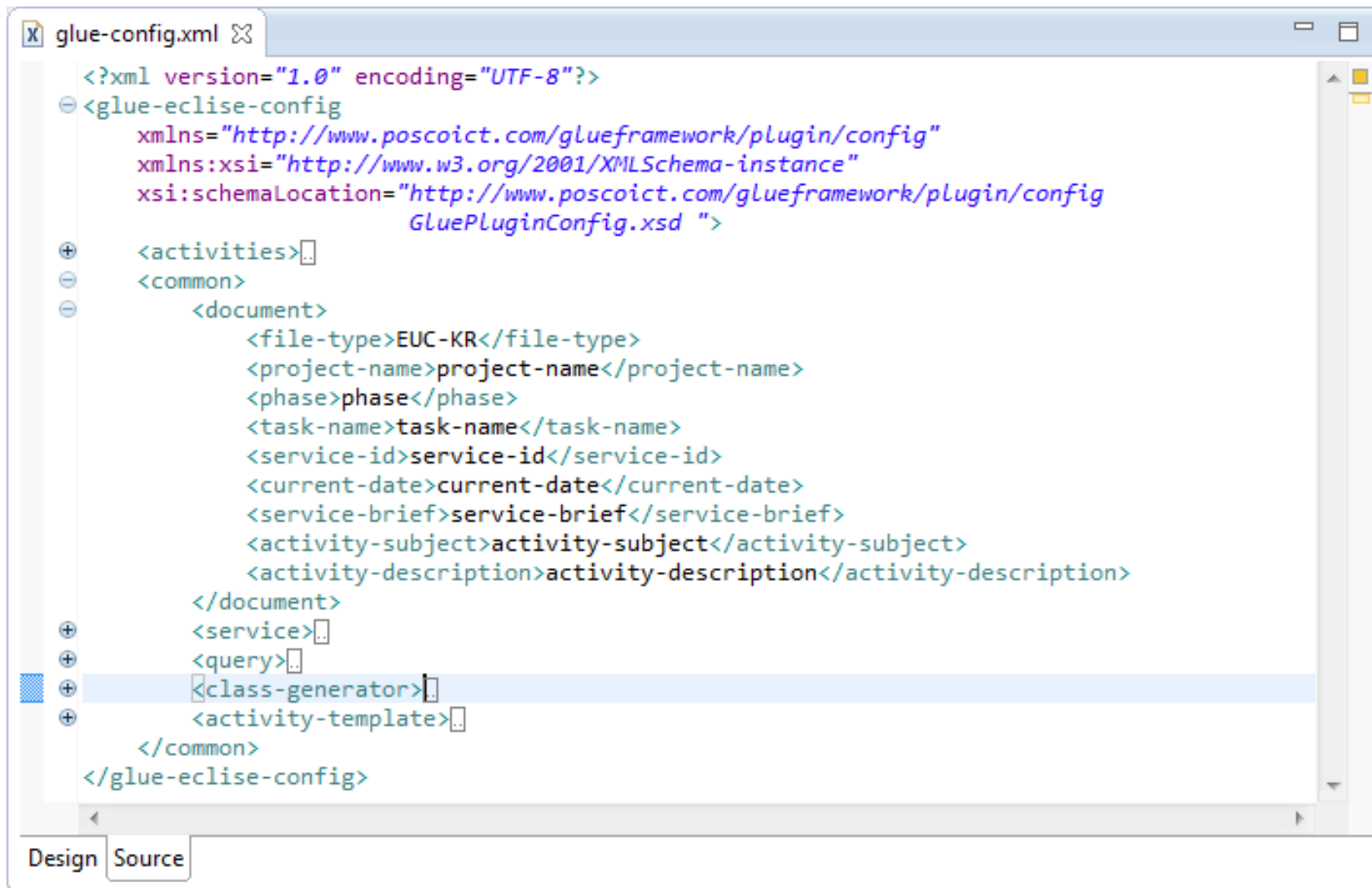
- activity.gif
- activity\_delete.gif
- activity\_insert.gif
- activity\_modify.gif
- activity\_pageset.gif
- activity\_router.gif
- activity\_search.gif
- activity\_subservice.gif
- components.gif

- service : GlueService 파일의 인코딩 유형 정의
- query : GlueQueryEditor 파일의 인코딩 유형 정의
- class-generator : Activity Diagram의 Class 생성 제외 대상 관리
- activity-template : Custom Activity의 template 코드



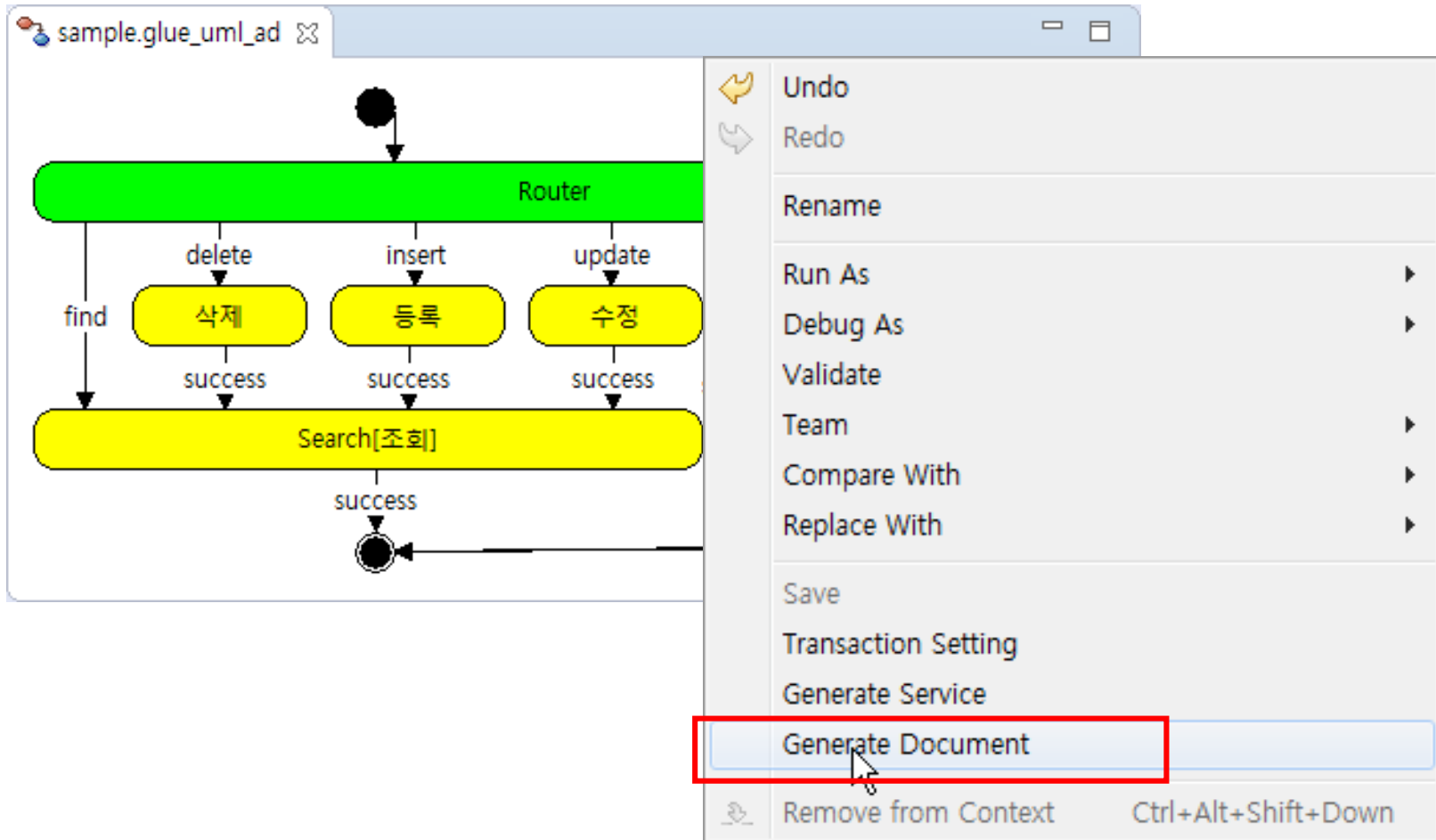
```
<?xml version="1.0" encoding="UTF-8"?>
<glue-eclipse-config
  xmlns="http://www.poscoict.com/glueframework/plugin/config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.poscoict.com/glueframework/plugin/config
    GluePluginConfig.xsd ">
  <activities>[]
  <common>
    <document>[]
    <service>[]
    <query>[]
    <class-generator>
      <exclude-package>com.poscoict.glueframework</exclude-package>
      <exclude-package>sample.activity</exclude-package>
    </class-generator>
    <activity-template>[]
  </common>
</glue-eclipse-config>
```

- Service 명세서 관련 환경설정

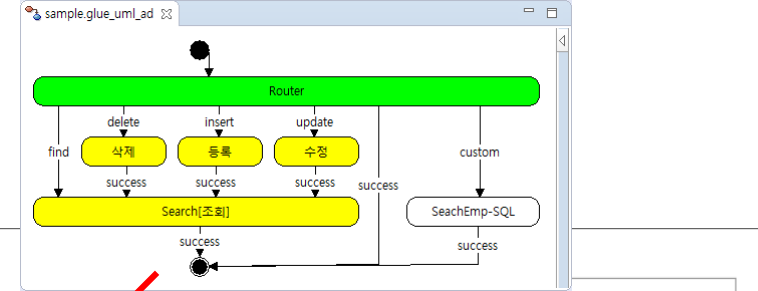


```
<?xml version="1.0" encoding="UTF-8"?>
<glue-eclipse-config
  xmlns="http://www.poscoict.com/glueframework/plugin/config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.poscoict.com/glueframework/plugin/config
    GluePluginConfig.xsd ">
  <activities>
    <common>
      <document>
        <file-type>EUC-KR</file-type>
        <project-name>project-name</project-name>
        <phase>phase</phase>
        <task-name>task-name</task-name>
        <service-id>service-id</service-id>
        <current-date>current-date</current-date>
        <service-brief>service-brief</service-brief>
        <activity-subject>activity-subject</activity-subject>
        <activity-description>activity-description</activity-description>
      </document>
      <service>
      <query>
      <class-generator>
      <activity-template>
    </common>
  </glue-eclipse-config>
```

- 설계 산출물로 사용할 수 있는 document를 생성해 준다.
- AD 우클릭 > Generate Document



- Activity Diagram은 자동으로 붙여지지 않고 화면을 Capture 하여 넣어야 함.



Service Name : sample-service

1. service-brief

GlueSDK에 포함된 예제 화면을 위한 Activity Diagram 입니다.  
sample-service는 사원정보를 관리하는 서비스입니다.

2. Activity Diagram

Please put the Activity Diagram Image!

3. activity-subject

3.1 Router

Properties:

name	value
class	com.poscoict.glueframework.biz.activity.GlueDefaultRouter

activity-description:

화면에서 발생하는 Event에 따라 분기하는 Activity입니다.  
GlueContext에는 find, delete, insert, update 중 한개의 정보가 있습니다.

3.3 삭제

Properties:

name	value
class	com.poscoict.glueframework.biz.activity.GlueJdbcDelete
sql-key	[sample.emp.delete] delete from emp where empno=?
chk-name	chk
dao	test-dao
param0	EMPNO
param-count	1

activity-description:

사원정보를 삭제하는 Activity입니다.  
result-key를 설정하지 않았으므로, 삭제결과는 '{sql-key}\_deleteCnt'가 사용됩니다.

3.4 수정

Properties:

name	value
name	value

## Service Specification

project-name	
phase	
task-name	
service-id	sample-service
current-date	2015-02-11

### Service Name : sample-service

#### 1. service-brief

GlueSDK에 포함된 예제 화면을 위한 Activity Diagram 입니다.  
sample-service는 사원정보를 관리하는 서비스입니다.

#### 2. Activity Diagram

Please put the Activity Diagram Image!

#### 3. activity-subject

##### 3.1 Router

Properties:

name	value
class	com.poscoict.glueframework.biz.activity.GlueDefaultRouter

activity-description:

화면에서 발생하는 Event에 따라 분기하는 Activity입니다.  
GlueContext에는 find, delete, insert, update 중 한개의 정보가 있습니다.

##### 3.2 등록

Properties:

name	value
class	com.poscoict.glueframework.biz.activity.GlueJdbcInsert
sql-key	[sample.emp.insert] insert into emp(empno, ename, sal, deptno) values(?, ?, ?, ?)
chk-name	chk_insert
dao	test-dao
param0	empno_insert

- **Glue Framework 에서의 Exception 처리 원칙**
  - 다음의 경우를 제외하고 모든 Exception을 Unchecked Exception을 사용한다.
    - 회복 가능한 에러가 발생한 경우 (시간이 지나면 복구되는 에러)
    - 모든 호출자들이 발생 되는 Exception을 꼭 처리해야 하는 경우
      - 비즈니스적인 Exception
  - GlueException(Unchecked Exception)을 확장하여 정의한다.
    - GlueDaoException, GlueCacheException, GlueMessageParserException 등등..
- **Checked Exception 의 단점**
  - 소스코드의 복잡도 증가
  - 소스코드의 가독성(Readability) 저하
    - 유지보수의 어려움
  - 비즈니스적인 Exception을 제외하고는 개발자가 프로그램적으로 처리할 수 있는 것이 상당히 제한적이다.



- **Checked Exception**

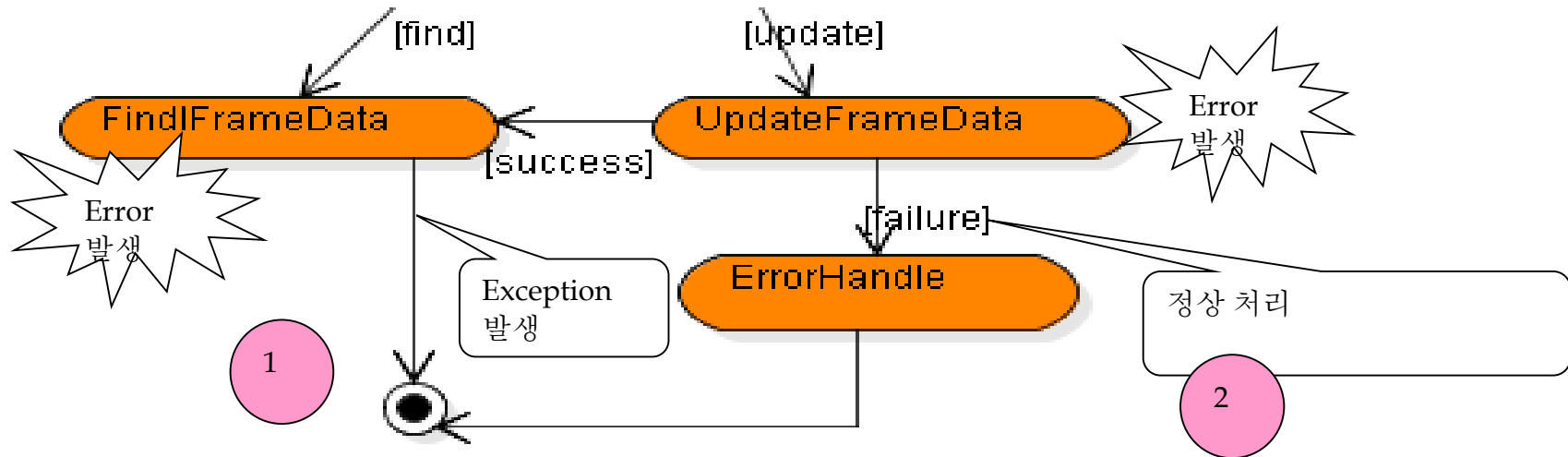
- java.lang.Exception을 상속한 Exception으로 try~catch 구문으로 처리하지 않을 경우 컴파일 에러가 발생한다. 그러므로, 컴파일 시점에 Exception 처리가 제대로 구현되어 있는지를 확인이 가능하다. 예를 들어 java.io.IOException이 대표적인 Checked Exception이다.

- **Unchecked Exception**

- java.lang.RuntimeException을 상속한 Exception으로 try~catch 구문으로 처리하지 않아도 컴파일 에러가 발생하지 않는다. 단, 런타임 시점에 에러가 발생할 경우 Exception이 발생한다. 예를 들어, java.lang.NullPointerException이 대표적인 Unchecked Exception이다.

Checked Exception	Unchecked Exception
<pre>public Task getTask() throws Exception;  public void performTask() {     try {         // ... pre processing         Task task = getTask();         task.start();         // ... post processing     } catch (Exception e) {         // handle exception     } }</pre>	<pre>public Task getTask() throws RuntimeException;  public void performTask() {     Task task = getTask();     task.start(); }</pre>

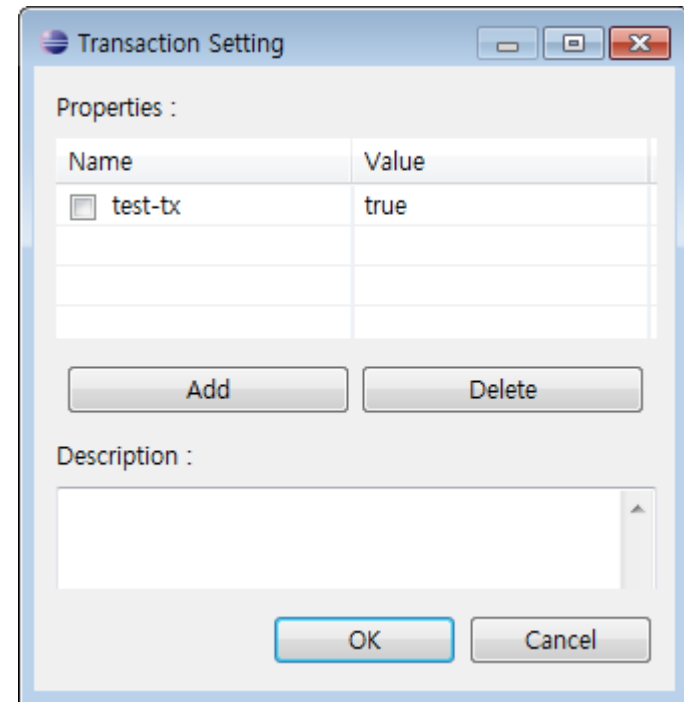
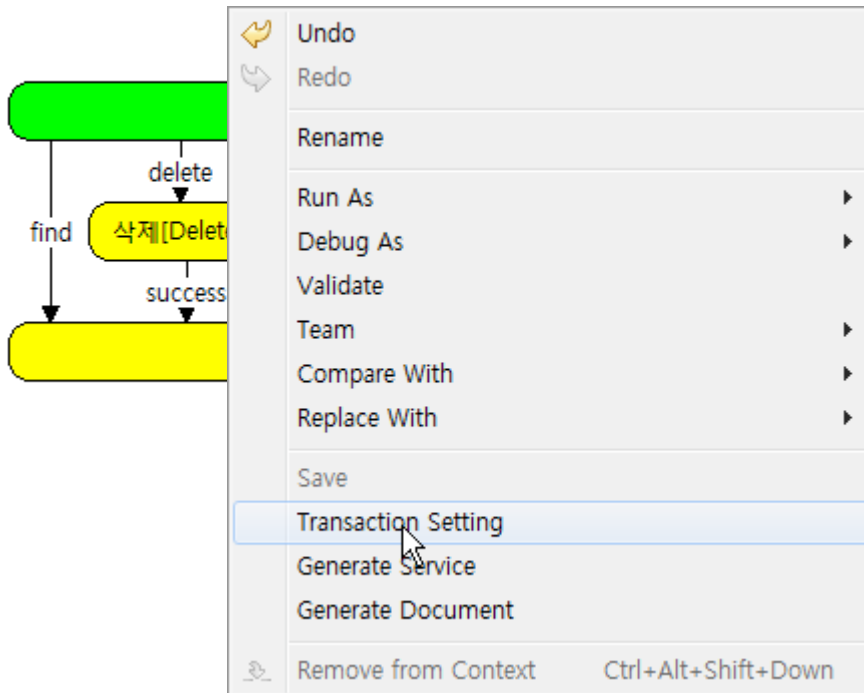
- Case #1 : Error 처리를 하지 않은 경우
  - 예러가 난 Activity에서 서비스 종료.
  - 상위 Layer (BizController) 에서 DB Transaction이 처리됨.
- Case #2 : Error 처리를 위한 Flow가 존재할 경우
  - Error 처리를 위한 별도의 비즈니스가 있는 경우에 해당



- transaction-manager

```
<transaction-manager id="tx1" commit="true"/>
```

- id
  - Spring의 applicationContext.xml에 정의 되어 있는 Transaction Manager의 bean ID
- Commit
  - 해당 Transaction을 commit 할 지에 대한 여부



- Transaction Manager 이용 : `commitTransaction()`, `rollbackTransaction()`

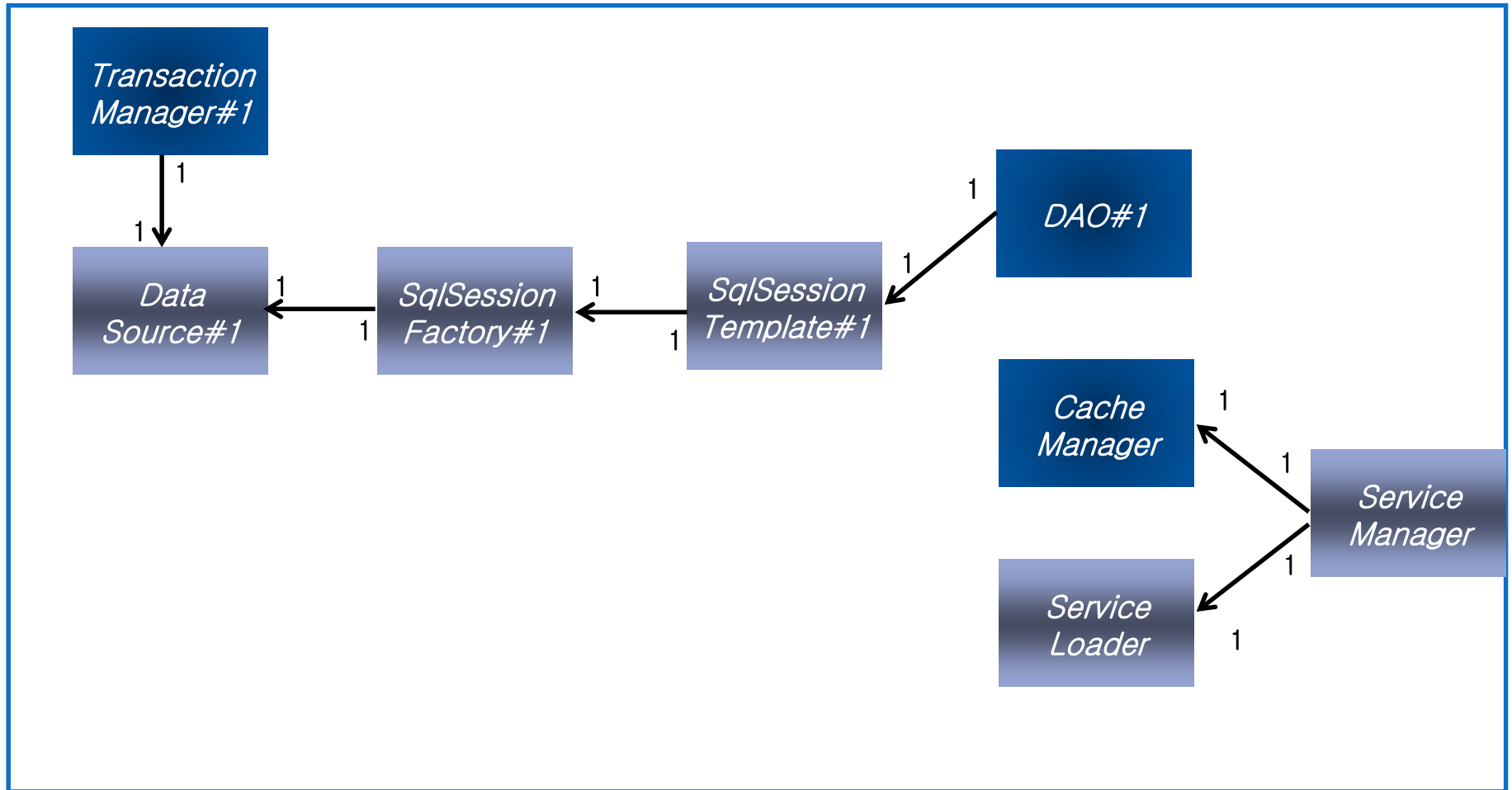
```
public class CustomActivity extends GlueActivity<GlueContext> {
    public String runActivity( GlueContext ctx ) {
        . . .

        //service.xml 에 기술된 것을 사용함
        //<transaction-manager id="test-tx" commit="true"/>
        this.commitTransaction( "test-tx" );
        . . .
        return GlueBizControlConstants.SUCCESS;
    }
}
```

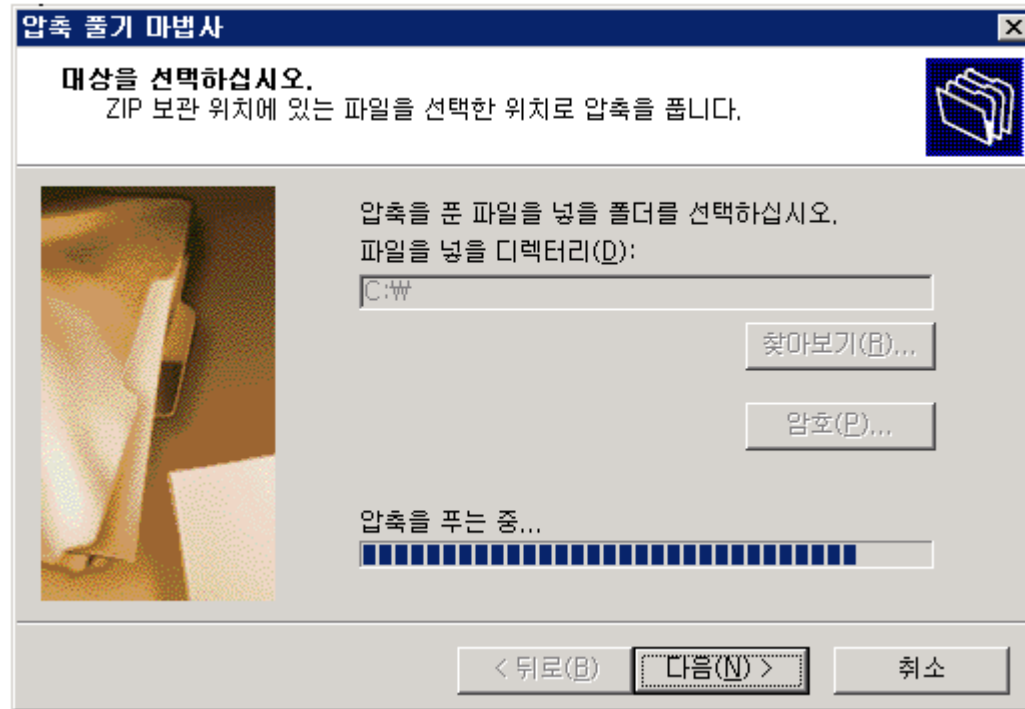
- Glue Framework 에서는 Service 시작 전 transaction 시작
- Service 종료 후 transaction 종료
- 프로그램 내에서 transaction 처리가 필요한 경우, 위와 같이 사용 가능

- MyBatis 용 JDBC 관련 Reusable Activity 제공
  - GlueMybatisSearch, GlueMybatisInsert, GlueMybatisDelete, GlueMybatisModify
- mybatis 사용시에는 Query Editor, xxx-query.glue\_sql 미사용
- 예제 파일
  - <https://www.solutionpot.co.kr/doc/framework5/guide/sample-project/howto-sample-mybatis.html>

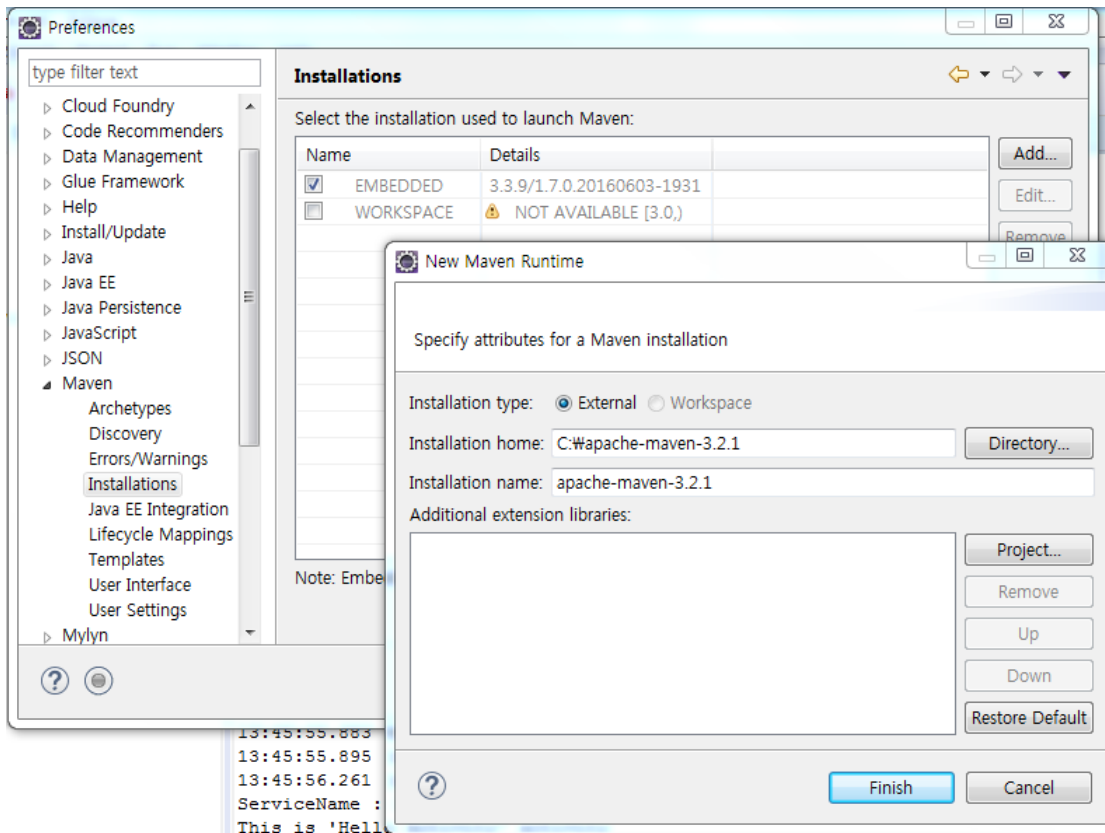
- DAO 관련 Bean
  - A→B : A가 B를 참조한다(Referencing)



- apache-maven-x.x.x-bin.zip 압축 해제



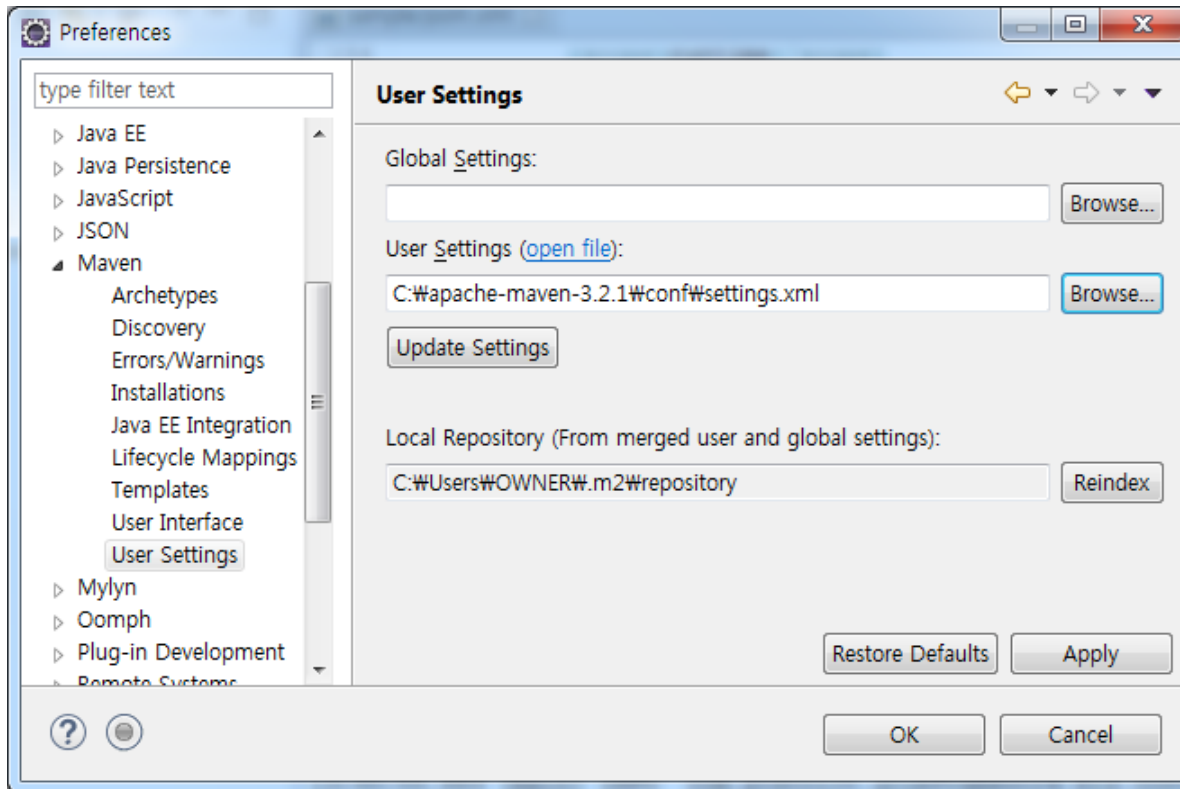
- Window - Preferences - Maven - Installations
- Installation home 에 C:\apache-maven-3.2.1 을 지정



- Finish 후, 추가한 항목을 선택 후 Apply



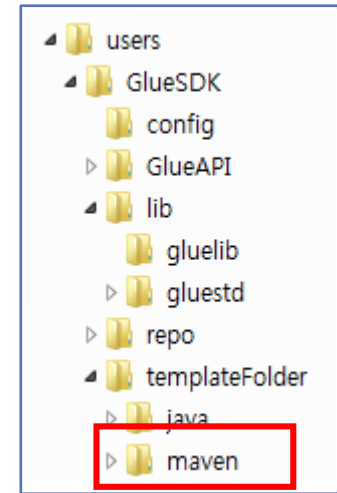
- Window - Preferences - Maven - User Settings



- User Setting 파일 지정과 Local Repository 변경 시 여기서 적용

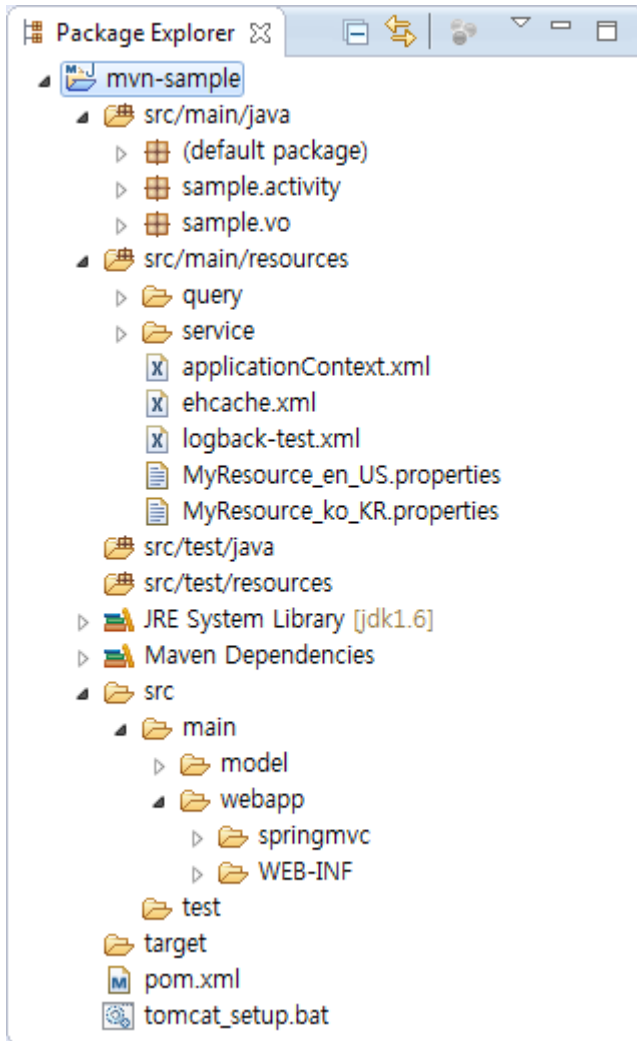
- GlueSDK repository 수정
  - pom.xml 수정 : %GlueSDK% / templateFolder / maven / pom.xml
    - repo의 위치(<url>)을 수정

```
sample/pom.xml
154         <scope>runtime</scope>
155     </dependency>
156     <dependency>
157         <groupId>commons-dbc</groupId>
158         <artifactId>commons-dbc</artifactId>
159         <version>1.4</version>
160         <scope>runtime</scope>
161     </dependency>
162 </dependencies>
163 <repositories>
164     <repository>
165         <id>repo</id>
166         <name>repo</name>
167         <url>file:///C://eclipse/users/GlueSDK/repo</url>
168     </repository>
169 </repositories>
170 </project>
```



공용 repository가 있다면 <repositories>는 삭제

- repository 찾는 순서 : 로컬 repository -> pom.xml 지정 repository -> maven 공용 repository



이름	설명
src/main/java	배포대상 Java 소스의 위치
src/main/resource	배포대상 Java 소스를 제외한 리소스의 위치
src/test/java	테스트용 Java 소스 위치
src/test/resource	테스트용 Java 소스를 제외한 리소스의 위치
src/main/model	Activity Diagram 등의 설계산출물
src/main/webapp	Web 소스 폴더 WebContents 와 동일한 역할
pom.xml	maven 용 POM(Project Object Model)

- Logging 정의 및 역할
  - 어떤 원인 때문에 에러가 발생했는지에 대한 세부 정보를 제공
  - 체계적이고 상세한 logging 메시지는 개발 시점뿐만 아니라 운영하는 시점에도 상당한 위력을 발휘함.
  - SLF4J 를 통해 Logging System 변경 가능.

- Logging system

- logback
- log4j
- Jakarta common logging
- ...

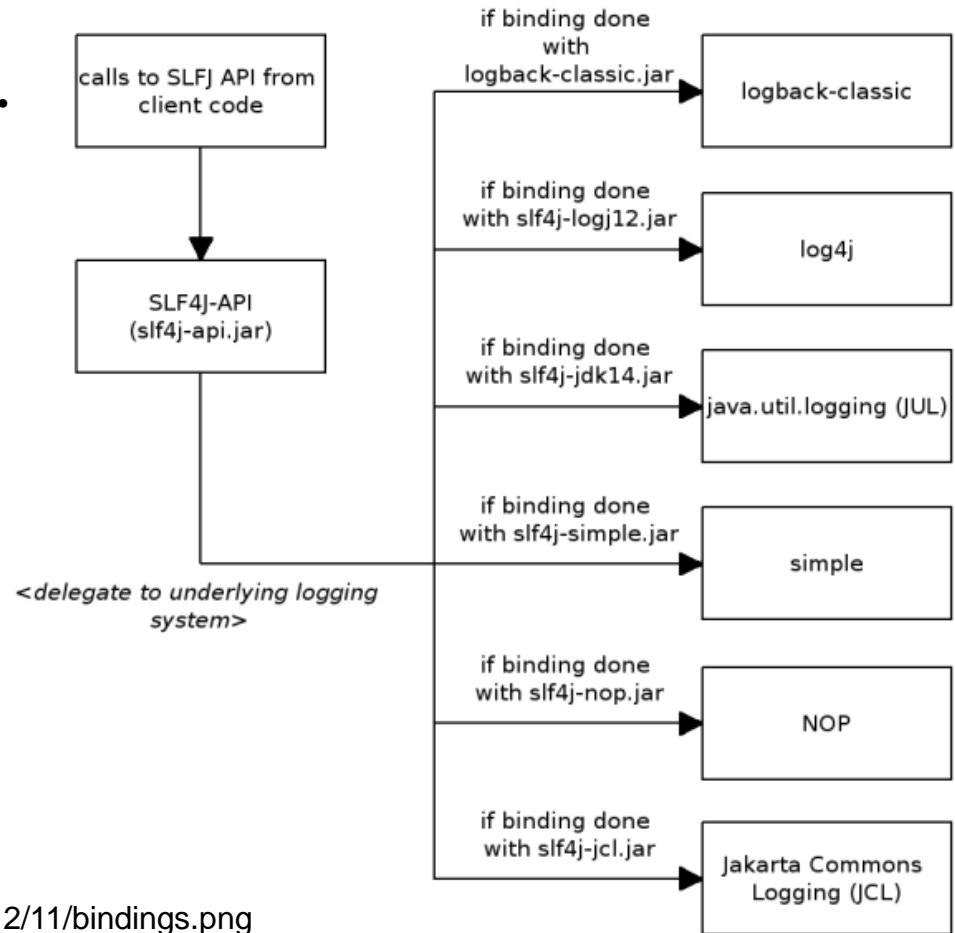


그림 출처 : <http://beyondj2ee.files.wordpress.com/2012/11/bindings.png>

- 메소드의 상세한 사용 방법은 java doc을 참조

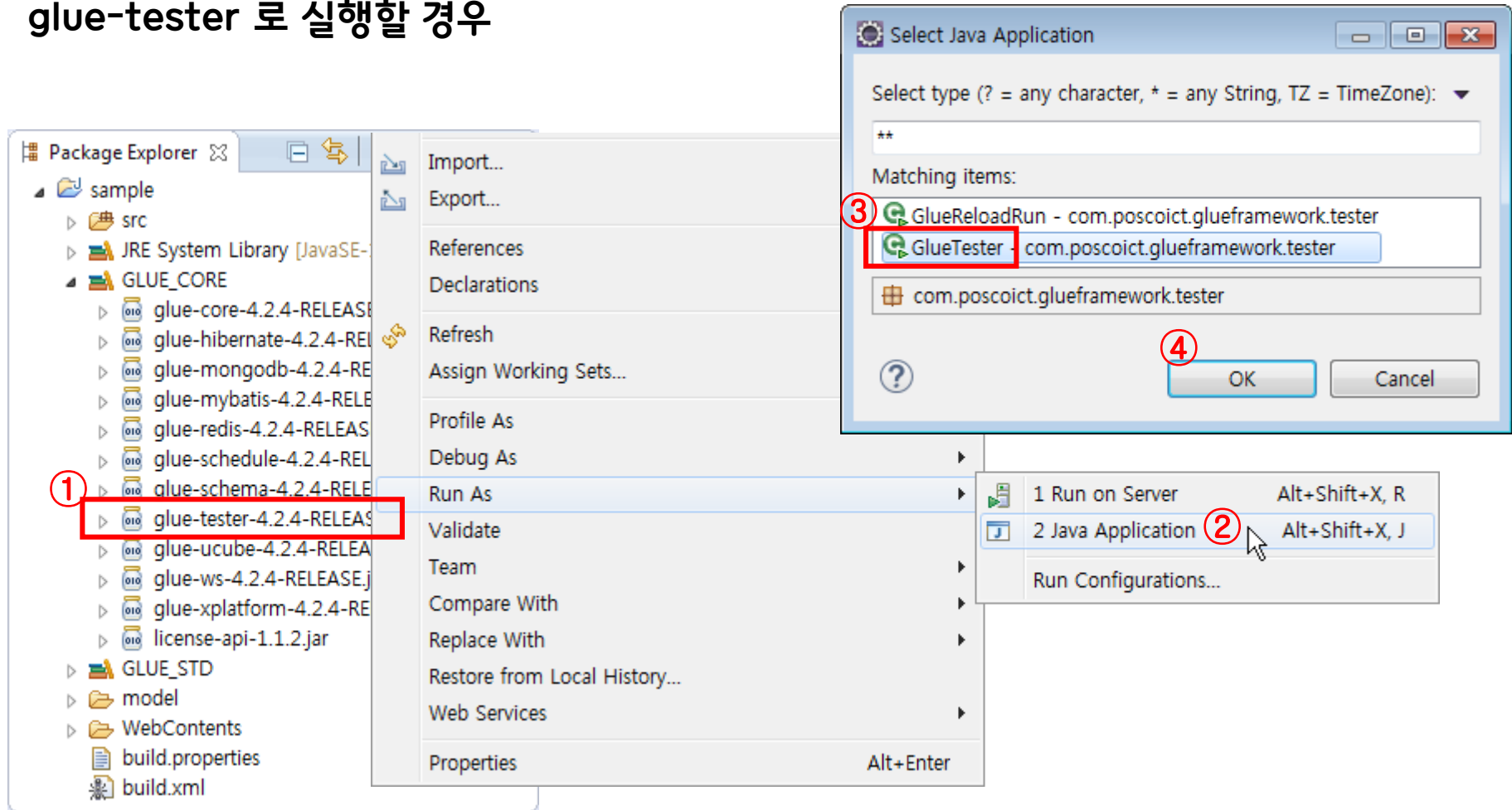
Method	Description
trace	Level : trace > debug > info > warn > error
debug	
info	
warn	
error	

```
public class LogTestActivity extends GlueActivity<GlueContext> {
    public String runActivity(GlueContext ctx) {
        // 현재 로그 레벨이 디버그 이상이면
        // (로그레벨, trace < debug < info < warn < error )
        if(this.logger.isDebugEnabled()){
            this.logger.debug("debug logging test");
        }
        logger.info("info logging test");
        if( 조건체크 ){
            try{
                // Biz Logic
            }catch(Exception e){
                // 에러 메시지와 Exception Stacktrace를 로깅한다.
                logger.error("error logging test : " + e.getMessage(), e);
            }
        }else{
            logger.warn("warn logging test");
        }
        return GlueBizControlConstants.SUCCESS;
    }
}
```

```
public class LoggingClient {
    // 각 클래스별로 GlueLog 를 등록함
    protected GlueLog logger = GlueLogFactory.getLogger(LoggingClient.class);

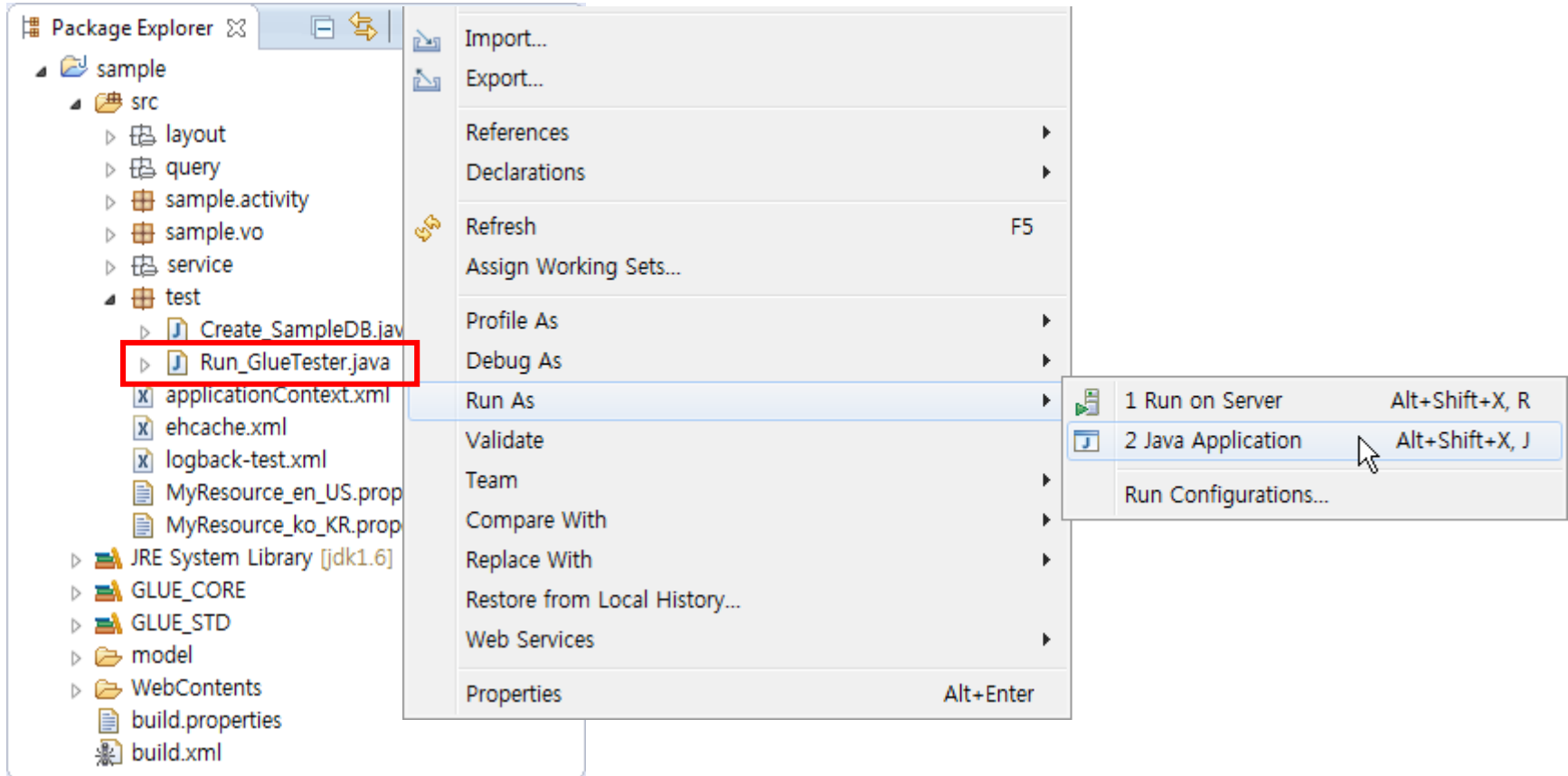
    public void performTask() {
        try {
            // 현재 로그 레벨이 디버그 이상이면
            // (로그레벨, trace < debug < info < warn < error )
            if (logger.isDebugEnabled()) {
                // debug logging
                logger.debug("debug logging test");
            }
            // 로깅할 메시지가 info 레벨 이상이 되는 경우는
            // 현재 로그 레벨을 체크하지 않아도 상관없다.
            logger.info("info logging test");
        } catch (GlueException ex) {
            // 에러 메시지와 Exception Stacktrace를 로깅한다.
            logger.error("Fail to perform task: " + ex.getMessage(), ex);
        }
    }
}
```

- GlueTester실행
  - glue-tester 로 실행할 경우

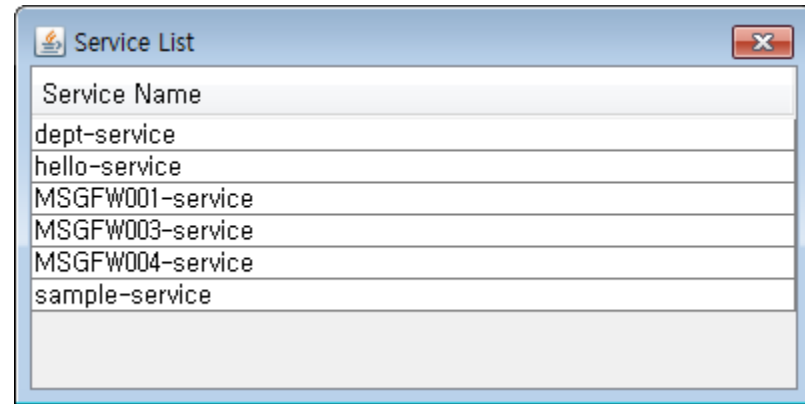
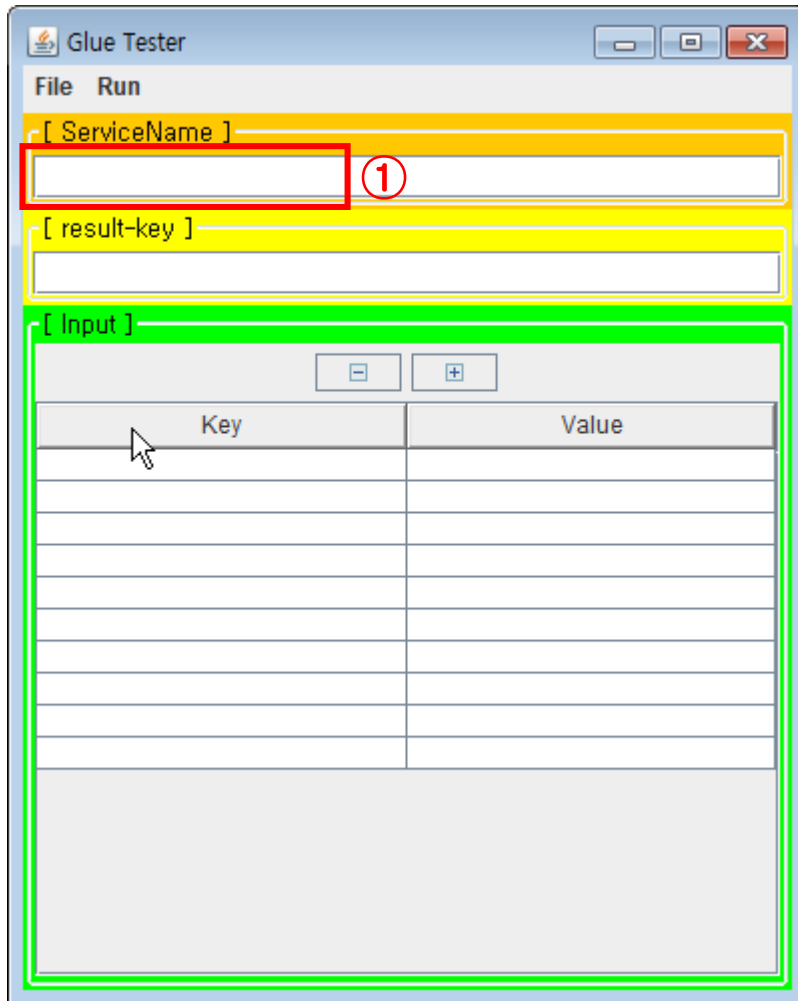




- GlueTester실행
  - Run\_GlueTester 로 실행할 경우

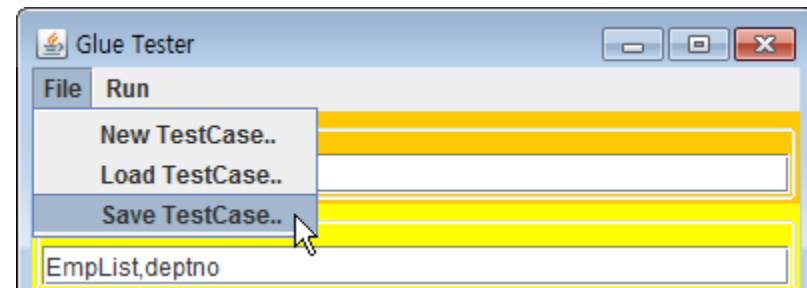
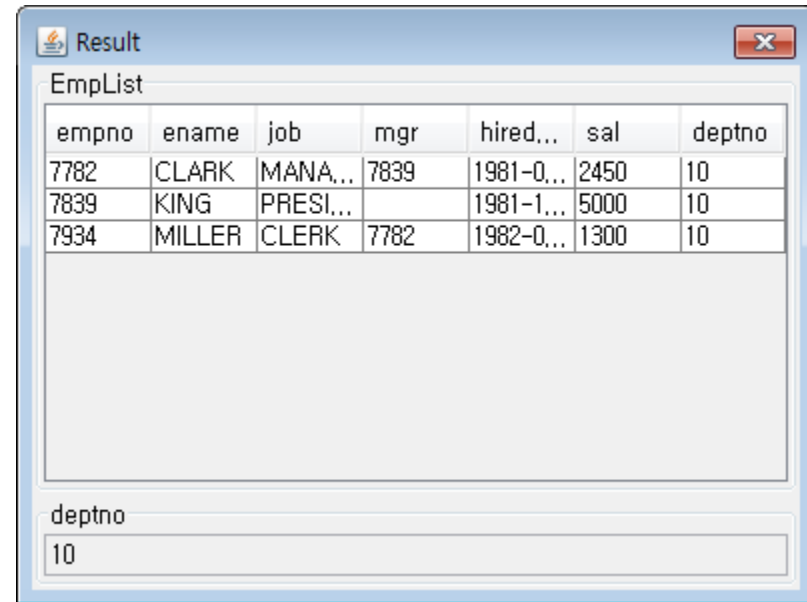
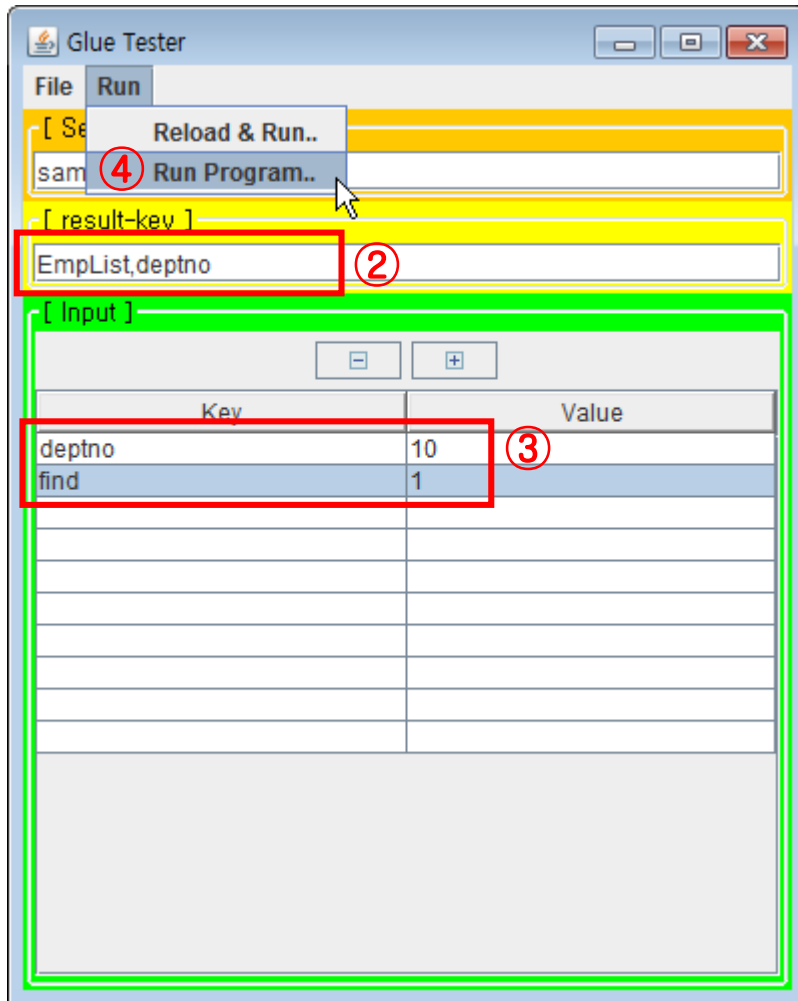


- GlueTester실행
  - Glue Tester 및 Service List

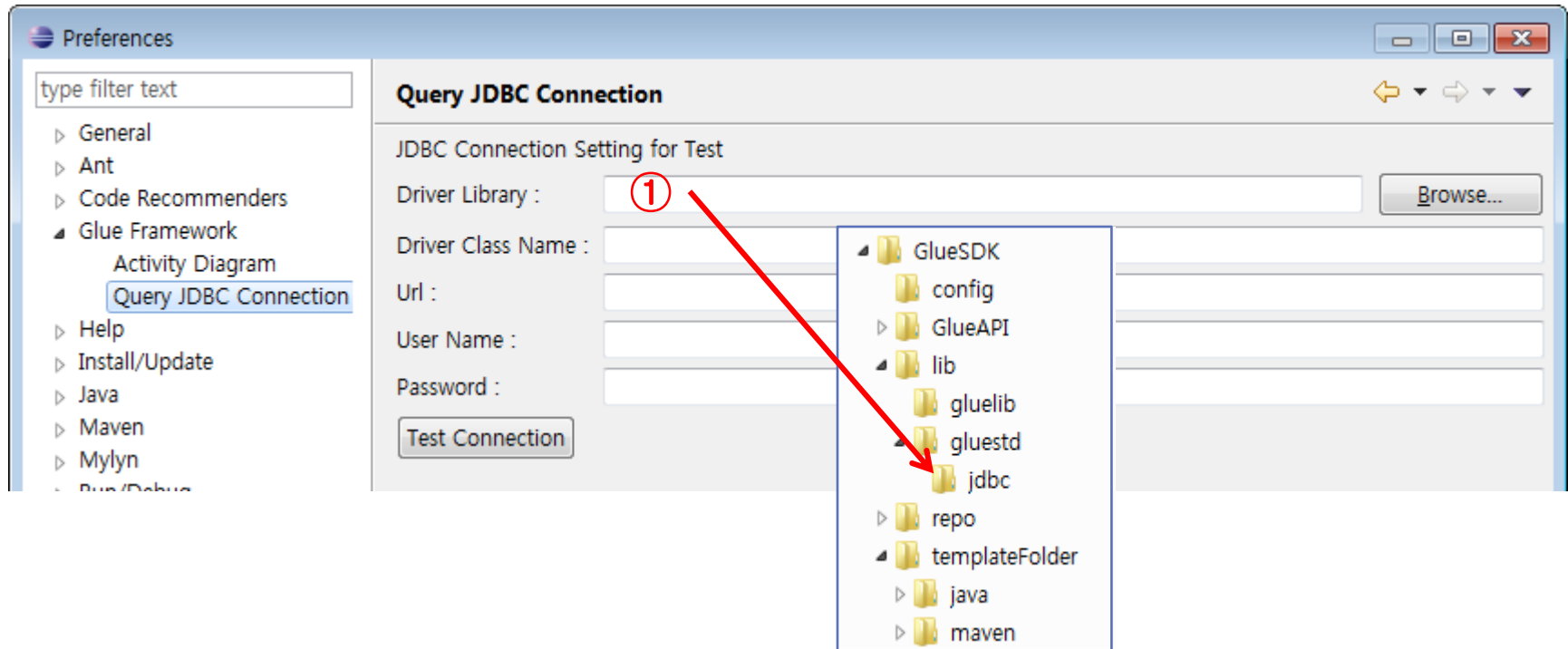


**ServiceName** 을 더블클릭하면  
프로젝트에 생성되어 있는 **Service List** 창 팝업

- GlueTester실행
  - 실행결과



- ‘?’ 를 포함하는 조회용 SQL의 테스트 지원
- SQL 테스트의 DB 연결정보는 Eclipse Workspace 별로 관리됨.
- Query 테스트시의 DB 연결정보는 applicationContext.xml 의 설정과 무관함.
- Driver Library 파일은 GlueSDK/lib/gluestd 외의 위치도 가능함.



- Bean 설정

- **messageSource를 Bean으로 등록**

```
<bean id="messageSource"
  class="org.springframework.context.support.ResourceBundleMessageSource">
  <property name="basenames">
    <list>
      <value>message</value>
    </list>
  </property>
</bean>
```

- Property 파일 등록

- **messageSource에 등록된 이름 뒤에 Locale정보를 붙여서 Property 파일 생성**

- 예: message\_ko.properties, message\_en.properties
- 우리 나라와 같이 2byte 문자를 사용하는 경우는 MessageSource에 변경 사항이 발생할 때마다 native2ascii 명령어를 실행하여 파일을 다시 생성해 주어야 하는 번거로움이 있다. 그러나 ResourceBundleEditor 이클립스 플러그 인(이클립스 실행 JVM이 JDK1.7 이상)을 사용하면 쉽게 작성할 수 있다.

- Locale 우선 순위

- User 설정 Locale -> Glue Property 설정(default.msg.locale) -> System Locale

- Globalization 기능 이용

- Java Util : GlueStaticContext.getResourceMessage(...)

- getResourceMessage(String beanName, String key, Object[] arguments, Locale locale)
  - beanName : ResourceBundleMessageSource가 등록된 Bean ID
  - key : property 파일에 정의된 message ID
  - arguments : Message에 바인딩 되는 값들
  - locale : 적용 Locale, Locale이 ko\_KR인 경우 message\_ko\_KR.properties파일의 값을 읽어온다.
- getResourceMessage(key)
- getResourceMessage(key , locale)
- getResourceMessage(key, arguments)
- getResourceMessage(key, arguments, locale) :
  - Parameter로 Locale 이 없을 경우 선 순위
    - User 설정 Locale -> Glue Property 설정 Locale(default.msg.locale) -> System Locale
  - Parameter로 beanName 이 없을 경우 Default값
    - messageSource

- **사용예**

```
String key = "showtable.header.empinfo";
System.out.println("Default::::"+GlueStaticContext.getResourceMessage(key));
System.out.println("English::::"+GlueStaticContext.getResourceMessage(key, Locale.US));
System.out.println("한국어::::"+GlueStaticContext.getResourceMessage(key, Locale.KOREA));
```

- **Multi Time Zone 기능 이용**

- **Java Util : GlueStaticContext. getTimeZoneFormat(...)**

- **getTimeZoneFormat(Date date, String timezoneid, String textformat)**

- date : Text 형태로 변환 할 Date
- timezoneid : Text 형태로 변환 시 적용 할 Time Zone ID
- textformat : Text 형태로 변환 시 적용 될 Text Format (Default 값 : yyyy-MM-dd HH:mm:ss (z Z))

- **getTimeZoneFormat(Date date, int offset, String textformat)**

- date : Text 형태로 변환 할 Date
- offset : Text 형태로 변환 시 적용 할 Time Zone 의 OffSet값(분 단위 : 120일 경우 GMT +02:00)
- textformat : Text 형태로 변환 시 적용 될 Text Format

- **사용예**

```
Date date = new Date();  
String textformat = "yyyy-MM-dd HH:mm:ss (z Z)";
```

```
System.out.println("Default::::"+GlueStaticContext.getTimeZoneFormat(date, TimeZone.getDefault().getID(), textformat));  
System.out.println("Asia/Seoul::::"+GlueStaticContext.getTimeZoneFormat(date, "Asia/Seoul", textformat));  
System.out.println("Canada/Eastern::::"+GlueStaticContext.getTimeZoneFormat(date, "Canada/Eastern", textformat));  
System.out.println("Asia/Shanghai::::"+GlueStaticContext.getTimeZoneFormat(date, "Asia/Shanghai", textformat));  
System.out.println("+05:00::::"+GlueStaticContext.getTimeZoneFormat(date, 5*60, textformat));  
System.out.println("-06:00::::"+GlueStaticContext.getTimeZoneFormat(date, -6*60, textformat));
```