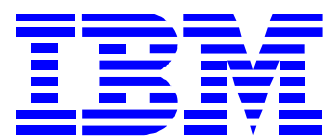


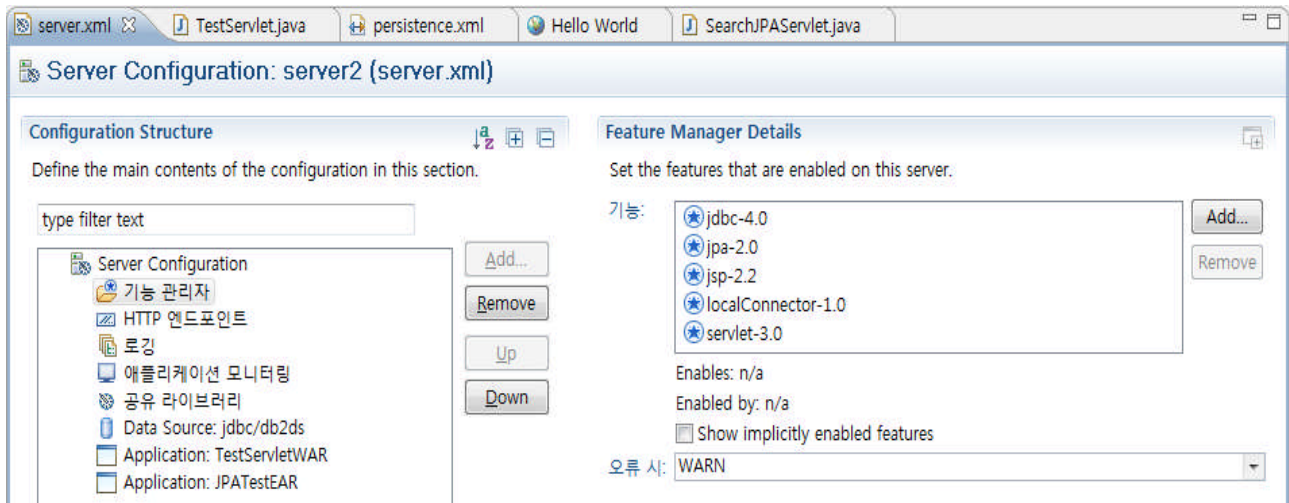
**WebSphere Application Server v8.5**  
**Liberty Profile – JPA Test**



## 1) IBM WAS V8.5 Liberty Profile - JPA 설정

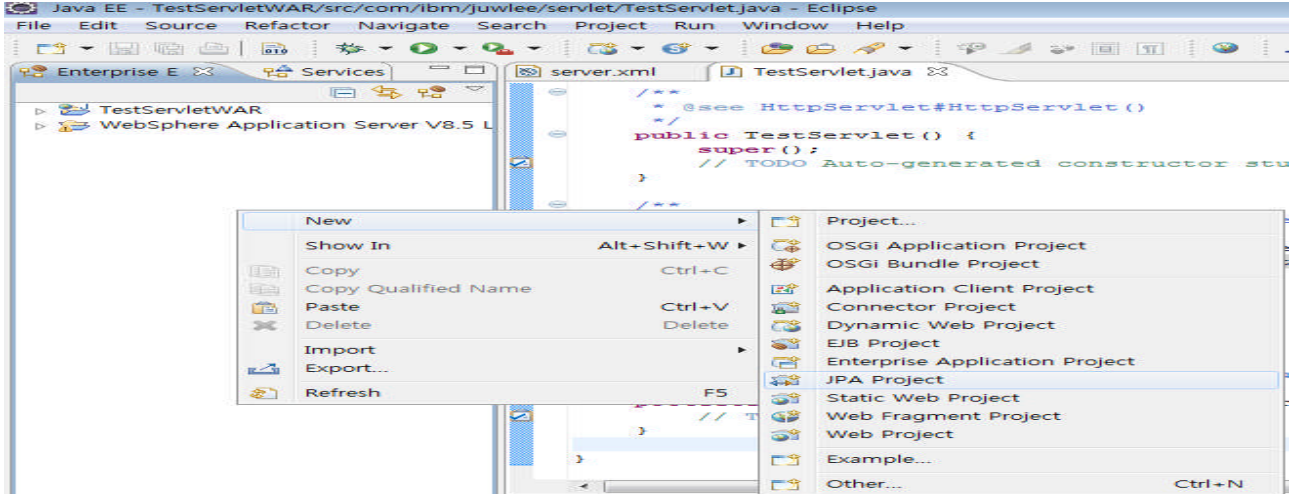
1. 이번 강좌에서는 IBM WAS V8.5 Liberty Profile을 이용해서 JPA 애플리케이션을 테스트 해보도록 하겠습니다. 해당 테스트를 위해서 먼저 server.xml > 기능 관리자에서 Add를 클릭하여 jpa-2.0 기능을 추가합니다.

(jpa의 경우에는 당연히 datasource를 이용해서 DB Connection을 해야하기 때문에 jdbc-4.0 기능도 같이 포함되어야 합니다. 또한 JDBC 연결 부분의 설정은 지난 강좌인 JDBC 테스트를 참고하시기 바라겠습니다.)

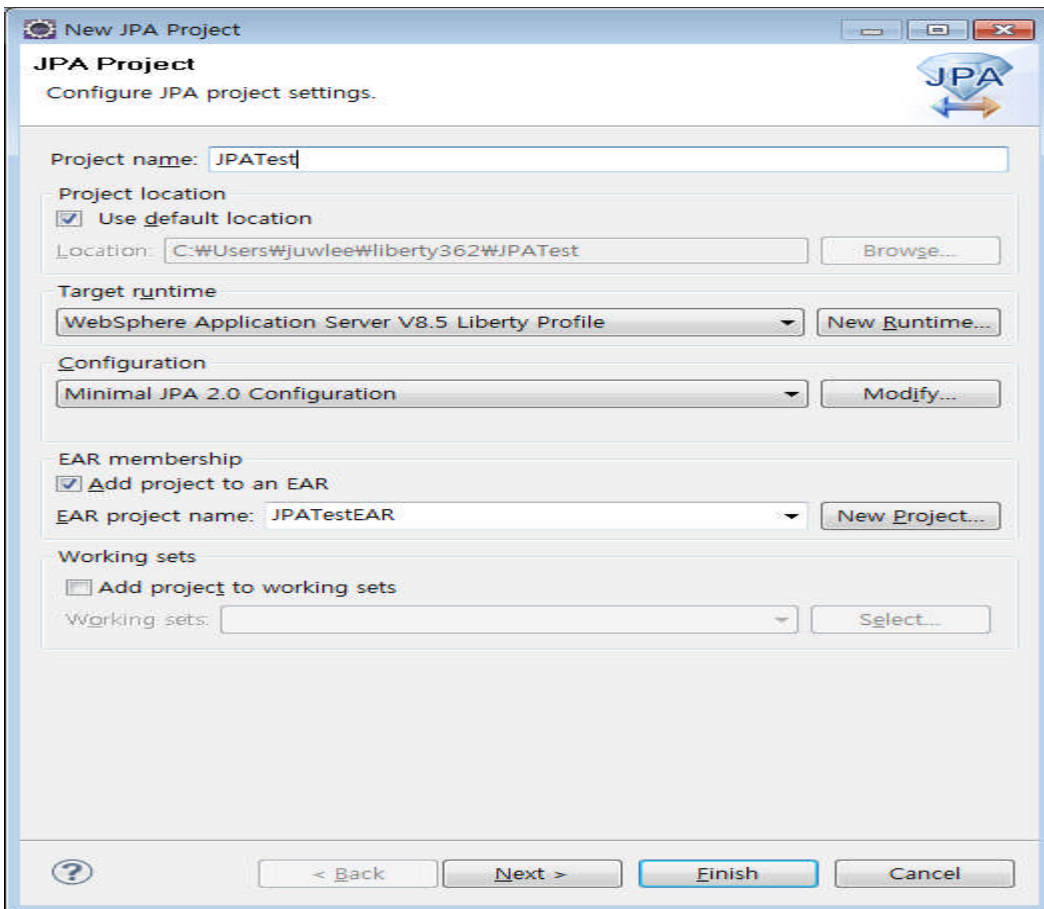


## 2) JPA Sample 개발

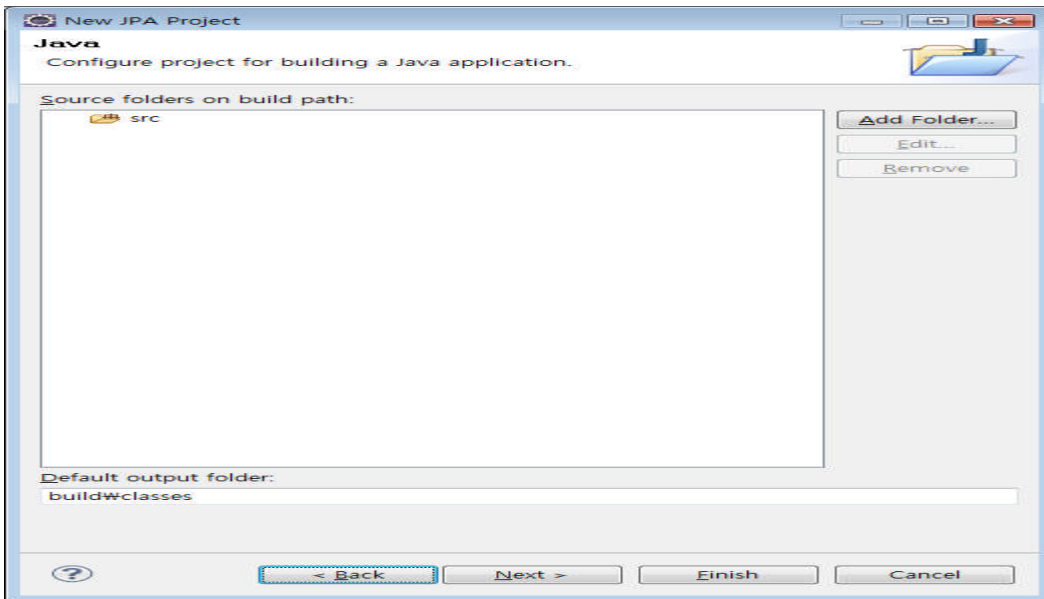
1. IBM WAS V8.5 Liberty Profile을 이용해서 JPA 를 테스트하기 위하여 간단하게 JPA Sample 애플리케이션을 개발 해보도록 하겠습니다. 먼저 왼쪽 탭에서 마우스 우 클릭하여 New > JPA Project를 선택합니다.



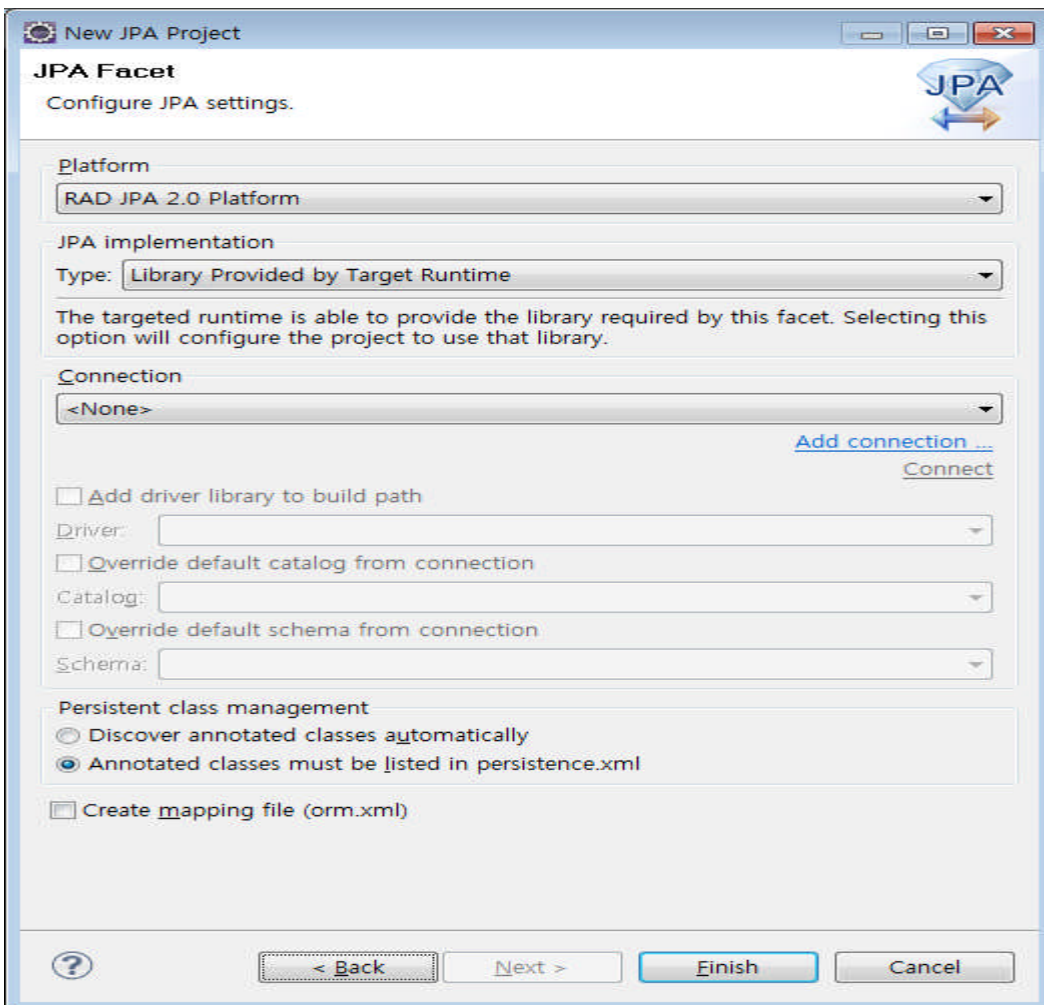
2. JPA Project 마법사가 나오면 JPATest 라는 이름을 입력하고 Target runtime과 Configuration을 하단과 같이 설정한 후 다음을 클릭합니다.



3. Source folder의 위치를 확인하고 다음을 클릭합니다.

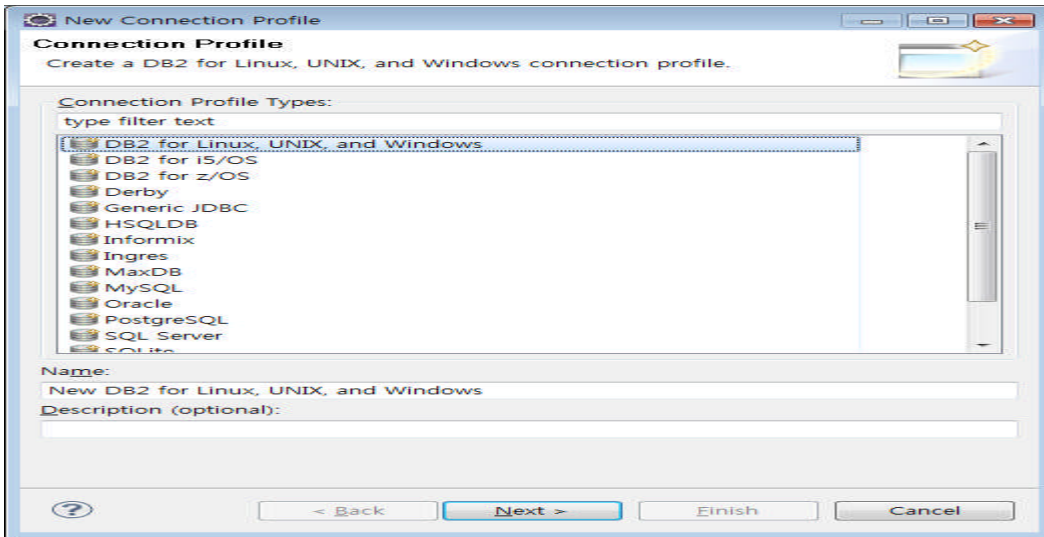


4. JPA Facet 화면을 확인한 후 DB Connection을 생성하기 위하여 Add connection을 클릭합니다.

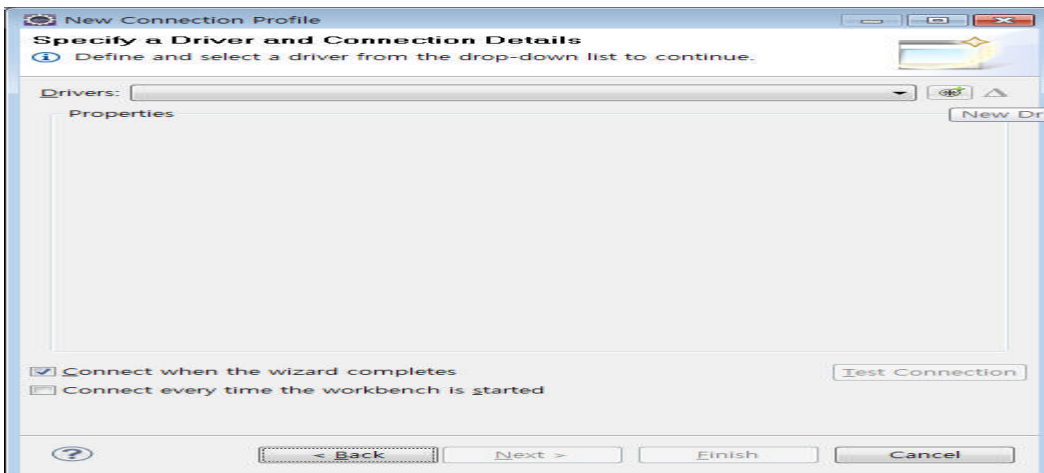


5. Connection Profile 화면이 나오면 설정하고자 하는 DB를 선택하고 다음을 클릭합니다.

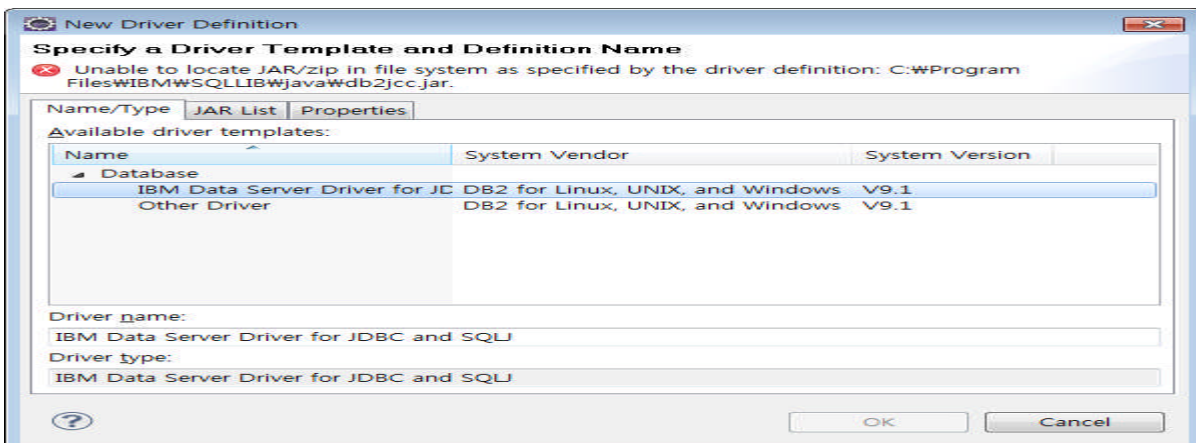
(이번 강좌에서는 이미 언급한 것처럼 DB2를 사용하도록 하겠습니다.)



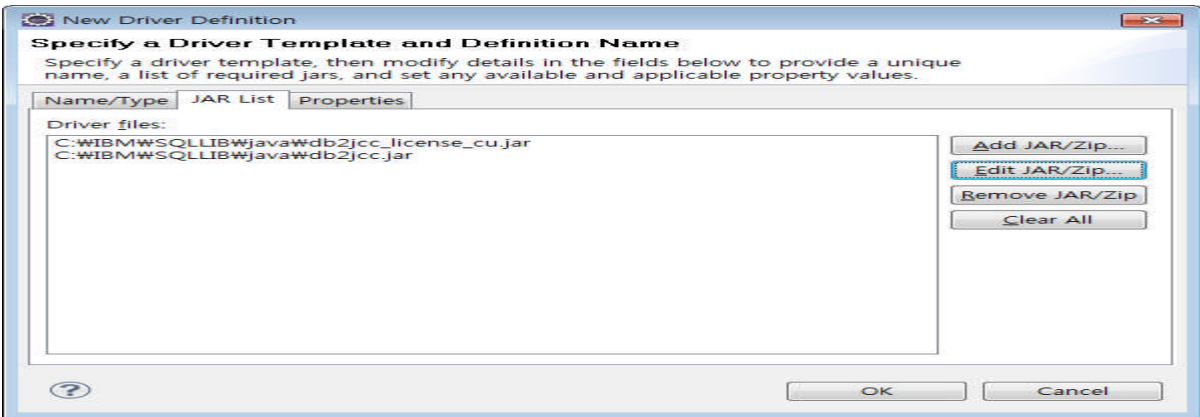
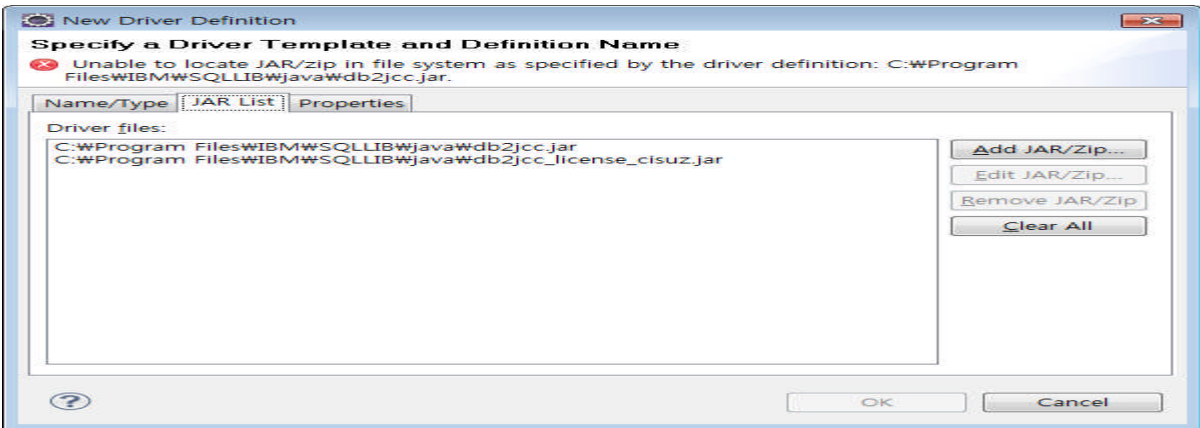
6. Drivers 선택화면이 나오면 아직 만들어둔 Driver가 없으므로 오른쪽 끝에 있는 New Driver를 클릭합니다.



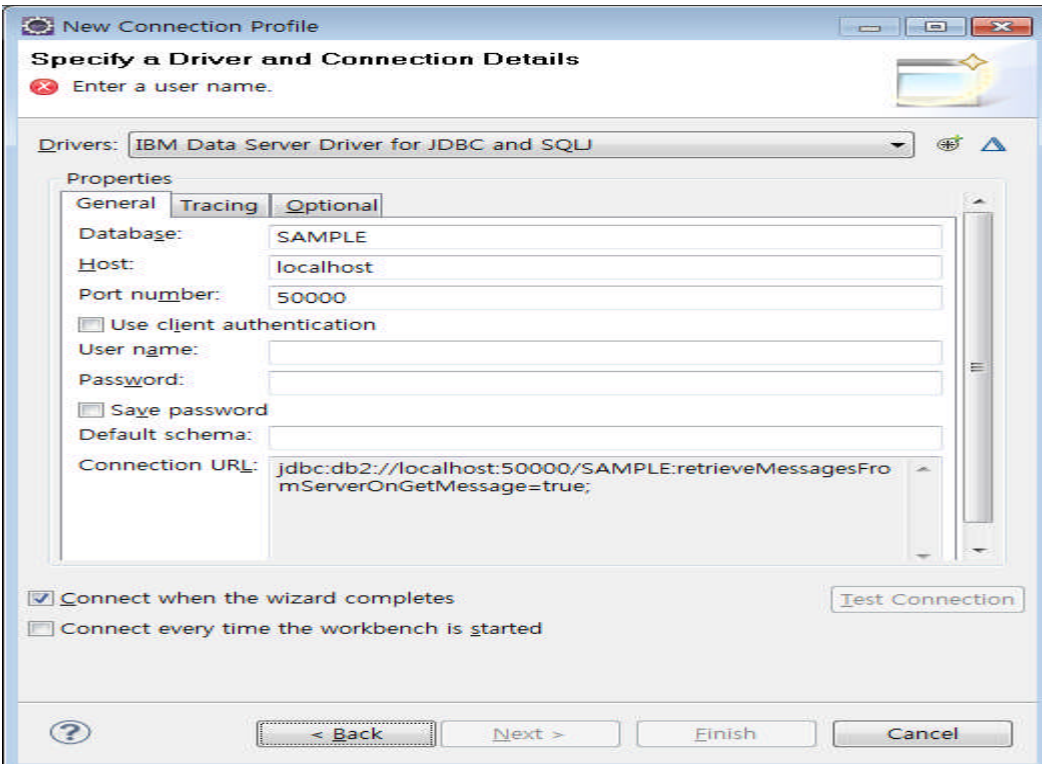
7. Driver Template에서 IBM Data Server Driver for JDBC를 클릭하고 JAR List를 선택합니다.



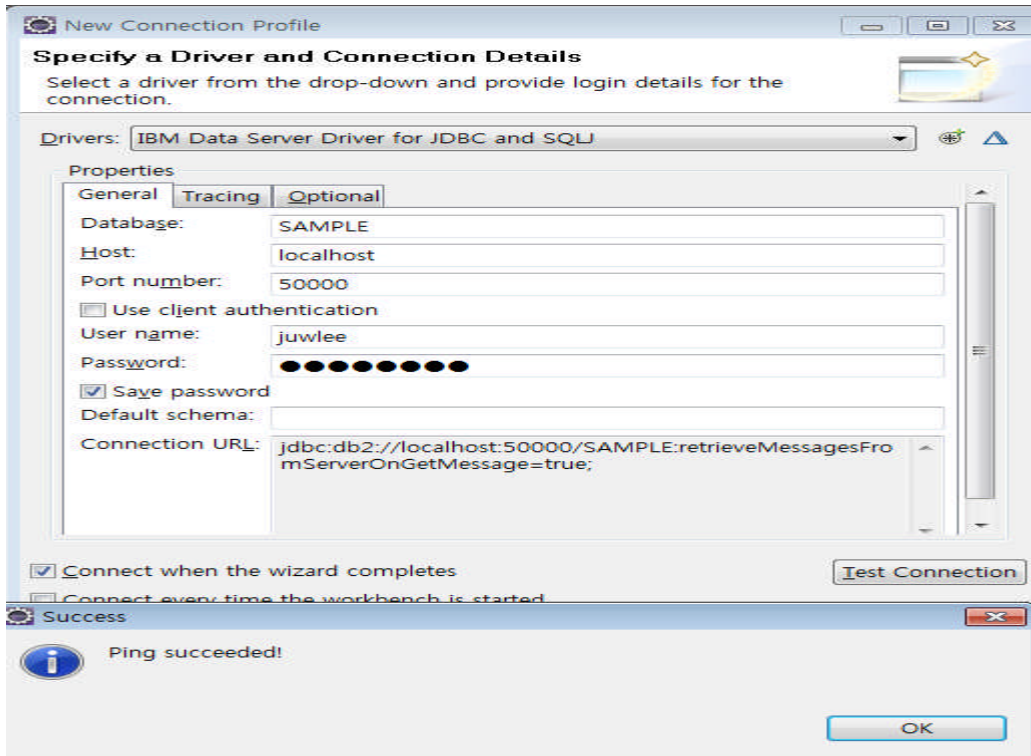
8. JAR List에서 DB2의 JDBC Driver jar 파일을 확인하고 설치되어 있는 정상적인 위치로 변경하고 OK를 클릭합니다.



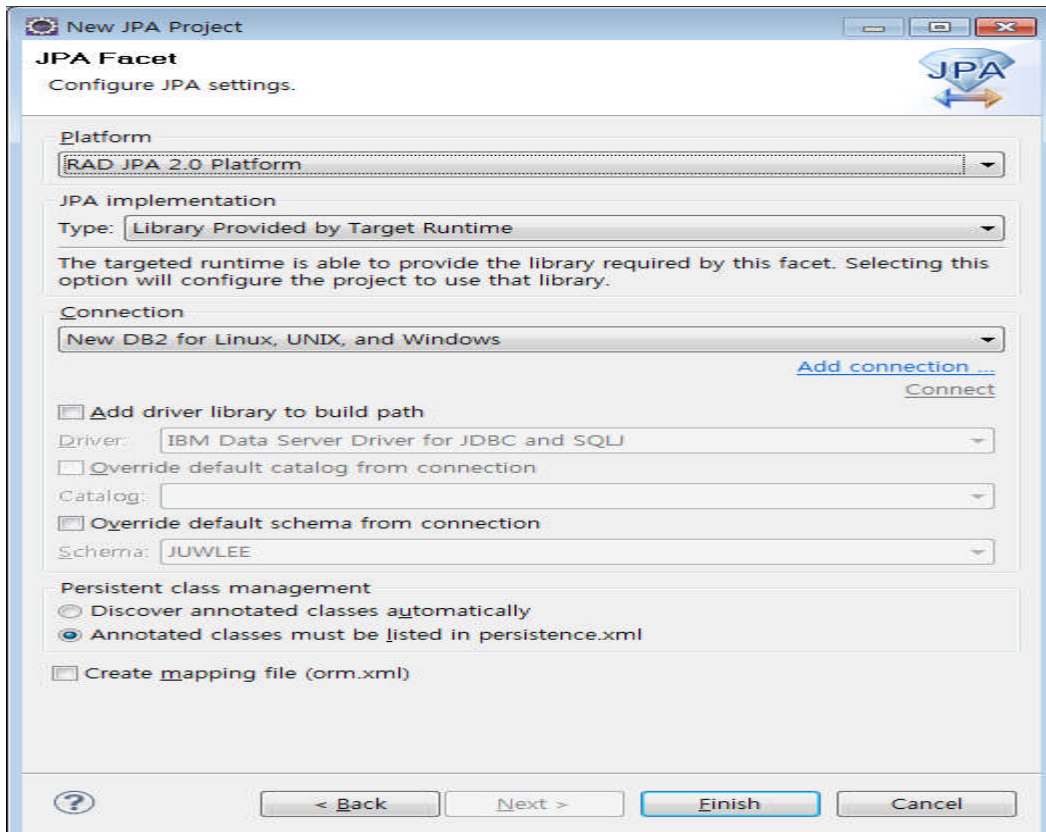
9. 세부 정보 화면에서 Host와 Port를 확인하고 User name과 Password를 입력합니다.



10. 설정이 다 되었으면 Test Connection버튼을 클릭하여 정상적으로 ping이 되는지 확인합니다. Ping이 성공적으로 완료되면 Driver 설정을 무사히 완료하신 것 입니다.

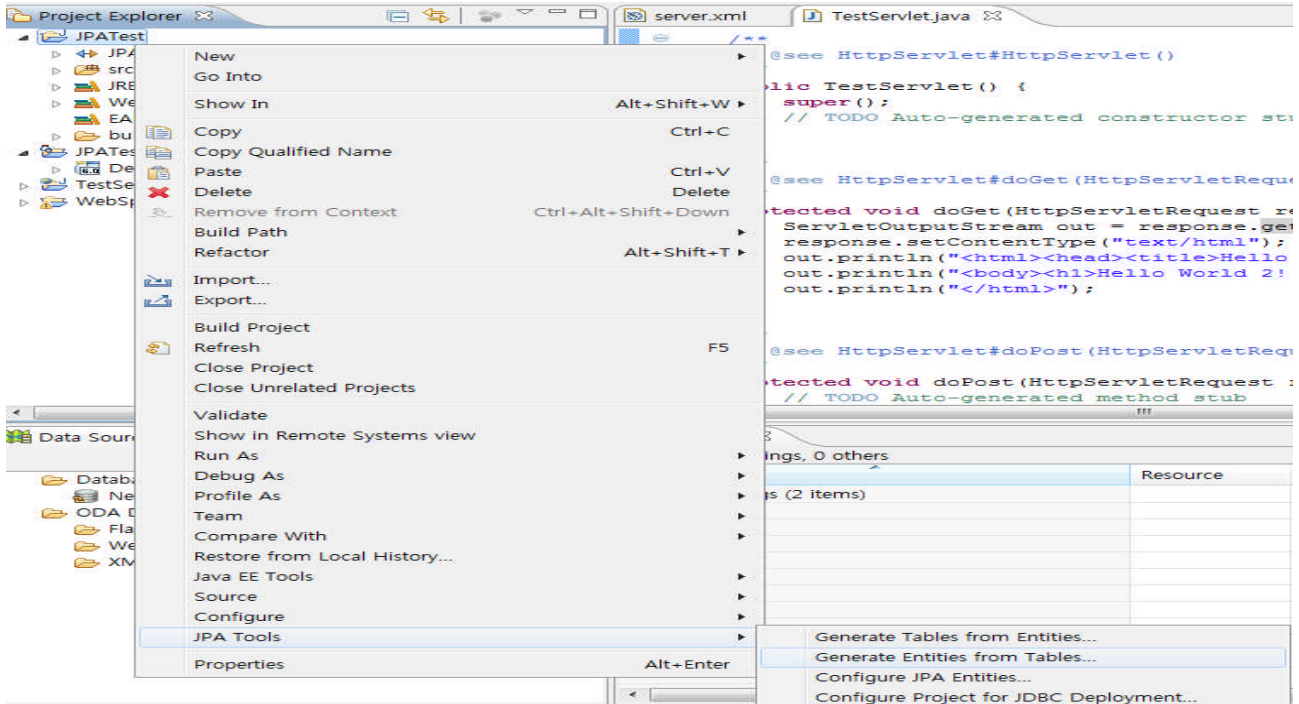


11. 새롭게 정의한 Connection이 정상적으로 입력된 것을 확인하고 완료를 클릭합니다.

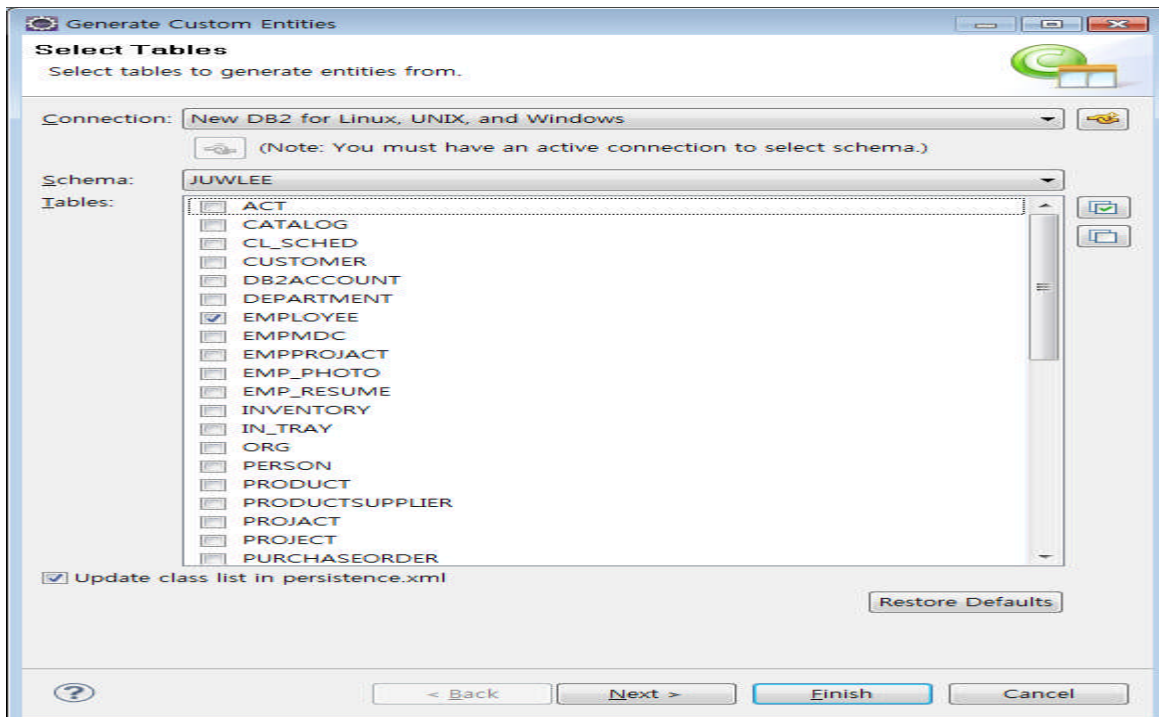


12. JPATest 애플리케이션의 기본 프로젝트가 생성된 것을 확인합니다. 다음으로 DB에서 만들어진 Table 을 기반으로 자동으로 JPA관련 Entity class를 생성하기 위하여 마우스 우 클릭하여 JPA Tools > Generate Entities from Tables을 선택합니다.

(JPA를 생성하는 다양한 방법이 있지만 여기서는 Table 기반으로 Entities를 만드는 방식을 테스트로 사용하며 다른 방법들은 JPA Programming 부분이기 때문에 다른 자료를 참고하시기 바라겠습니다.)

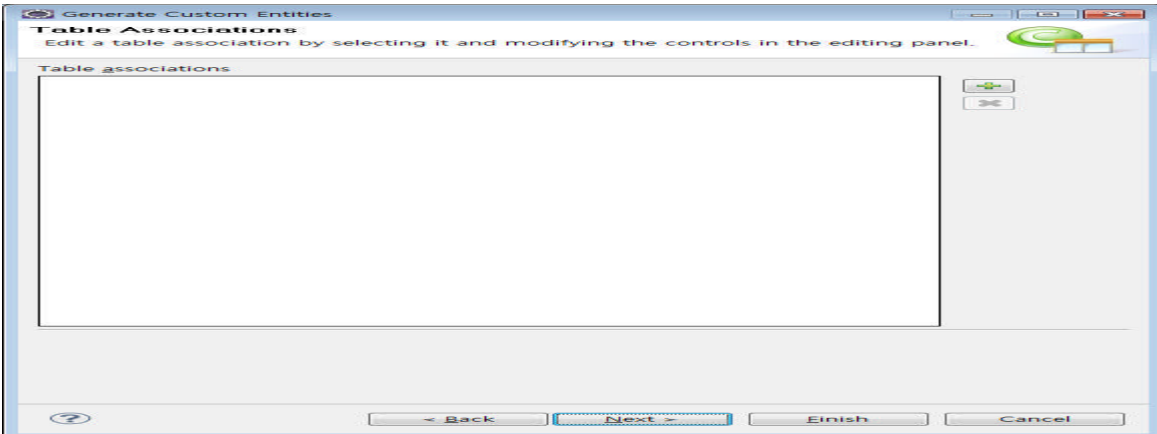


13. Entities 마법사가 나오면 이전에 만든 connection을 선택하고 원하는 Schema와 Table을 선택하고 다음을 클릭합니다.

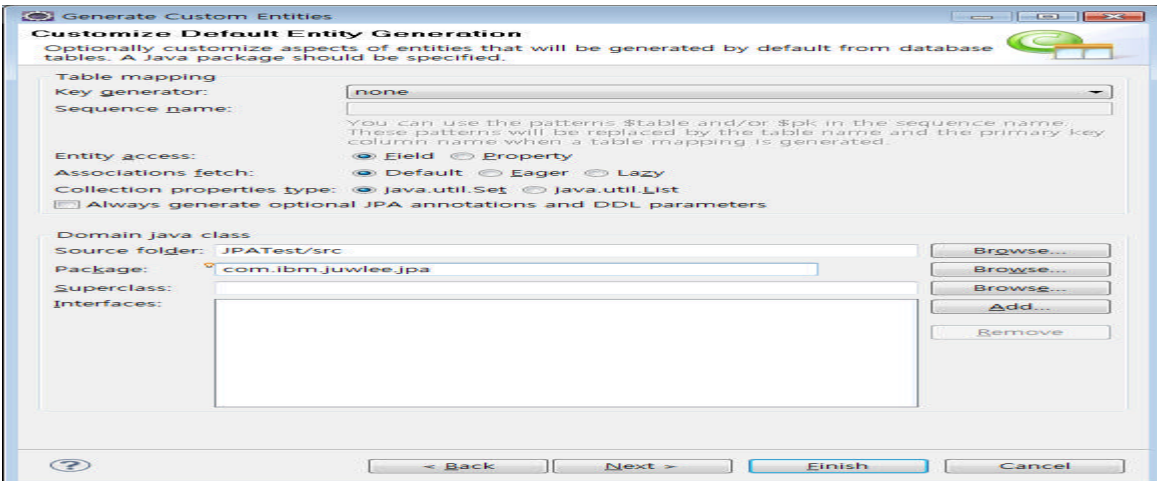




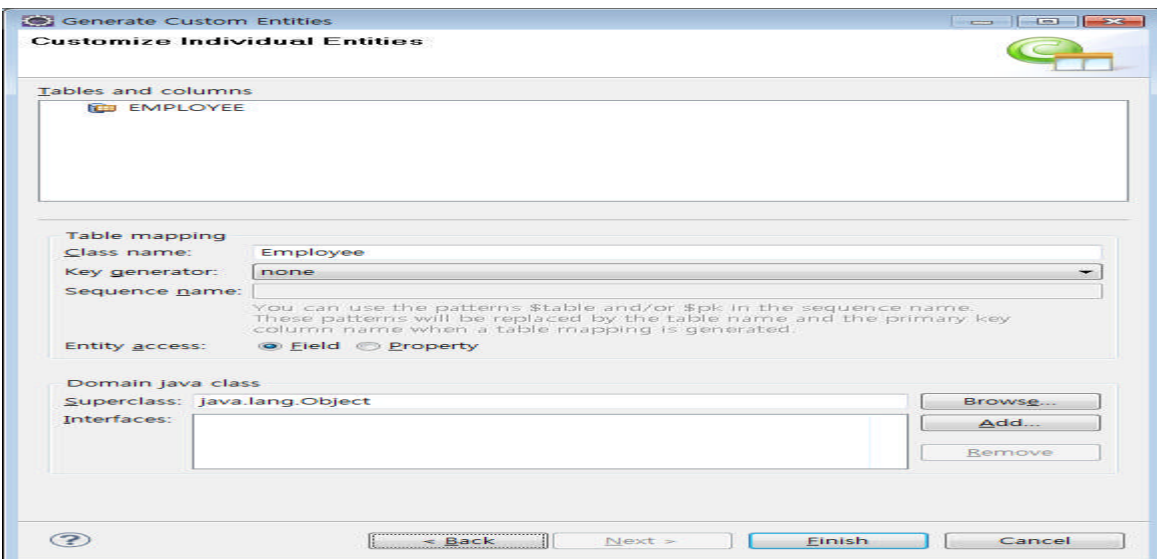
14. 테이블 연관관계입력 부분에서는 그냥 다음을 클릭합니다.



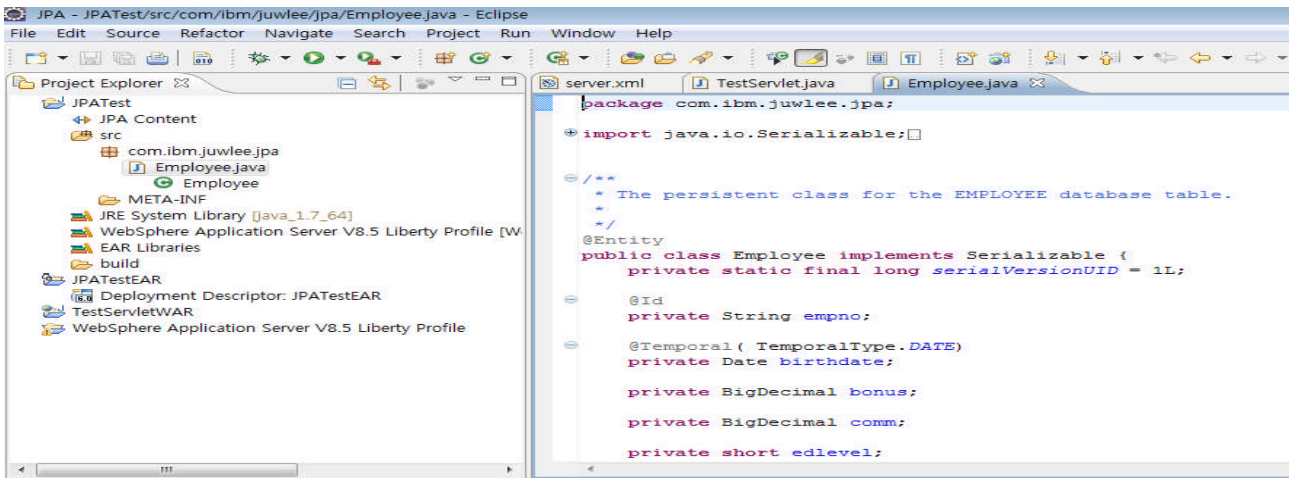
15. Entity Generation 부분에서는 패키지명만 입력하고 다음을 클릭합니다.



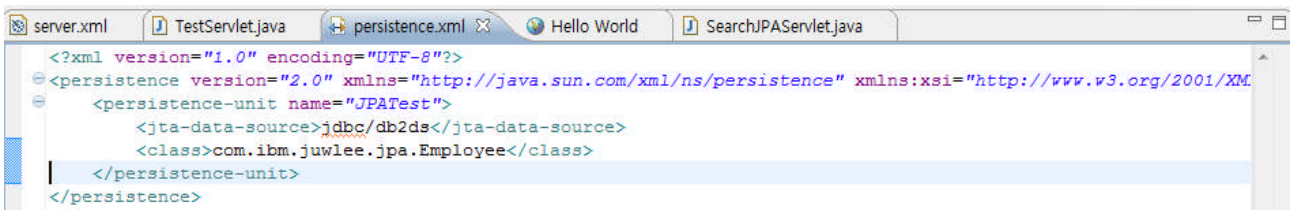
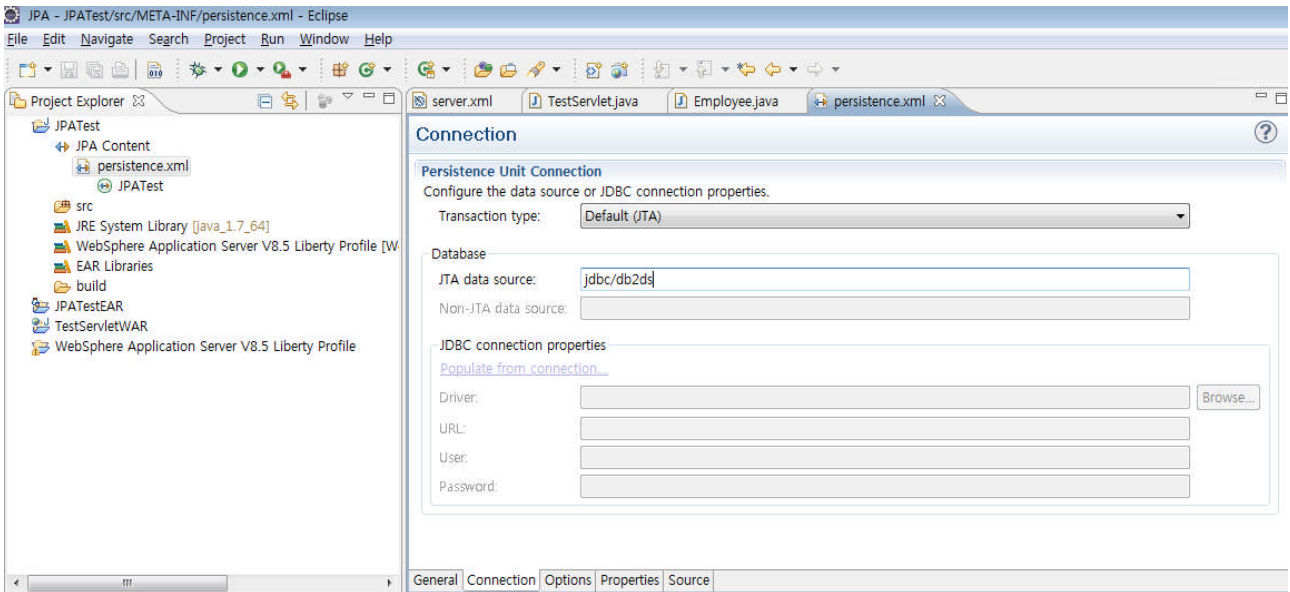
16. 요약화면을 확인하고 문제가 없다면 Finish를 클릭하여 Table에서 Entity 생성을 시작합니다.



17. 정상적으로 상단의 작업이 완료되었으면 하단과 같이 Table이 자동으로 Entity class로 생성된 것을 하단과 같이 확인할 수 있습니다.



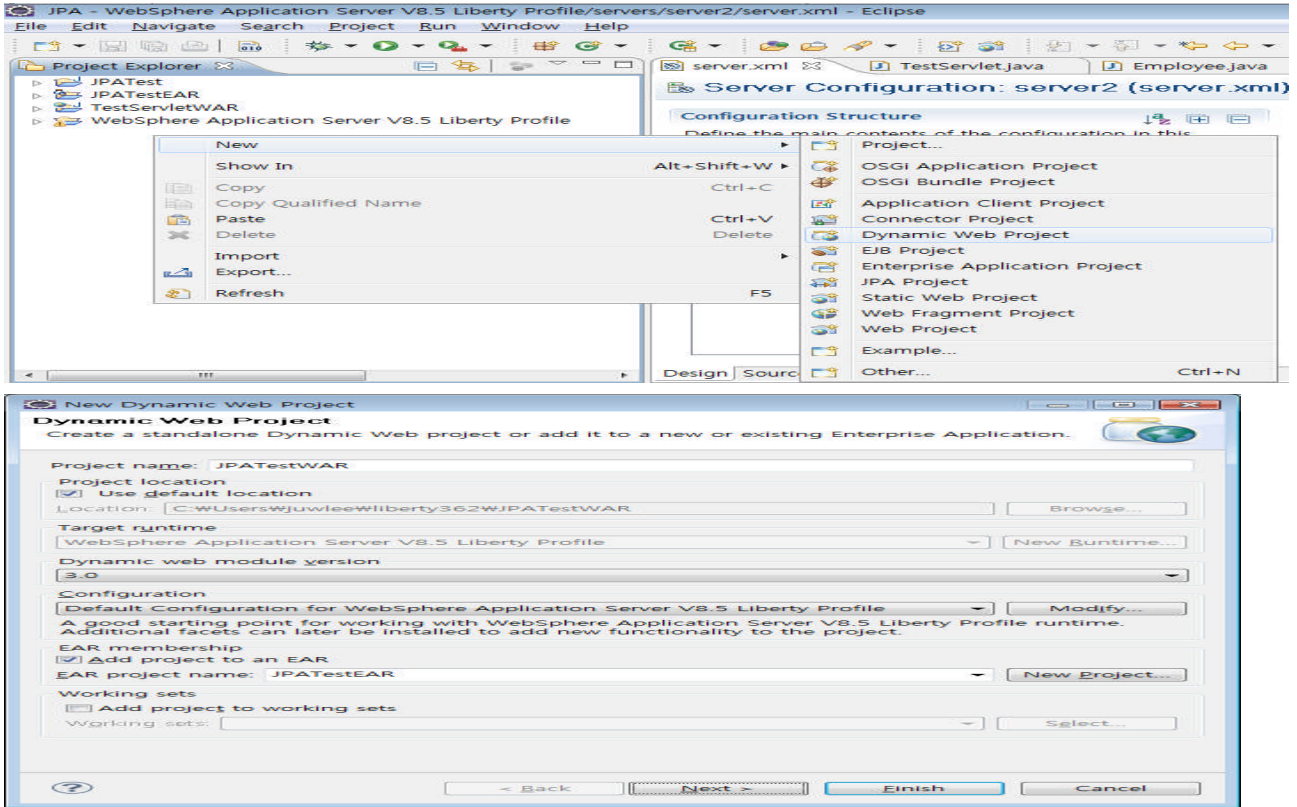
18. 또한 OR Mapping 설정 정보인 persistence.xml이 제대로 생성된 것을 확인하고 connection 탭에서 사용할 JTA data source 이름에 JDBC Datasource의 JNDI 이름을 넣어줍니다. (여기서는 지난 강좌에서 미리 만들어둔 jdbc/db2ds를 사용하도록 하겠습니다.)



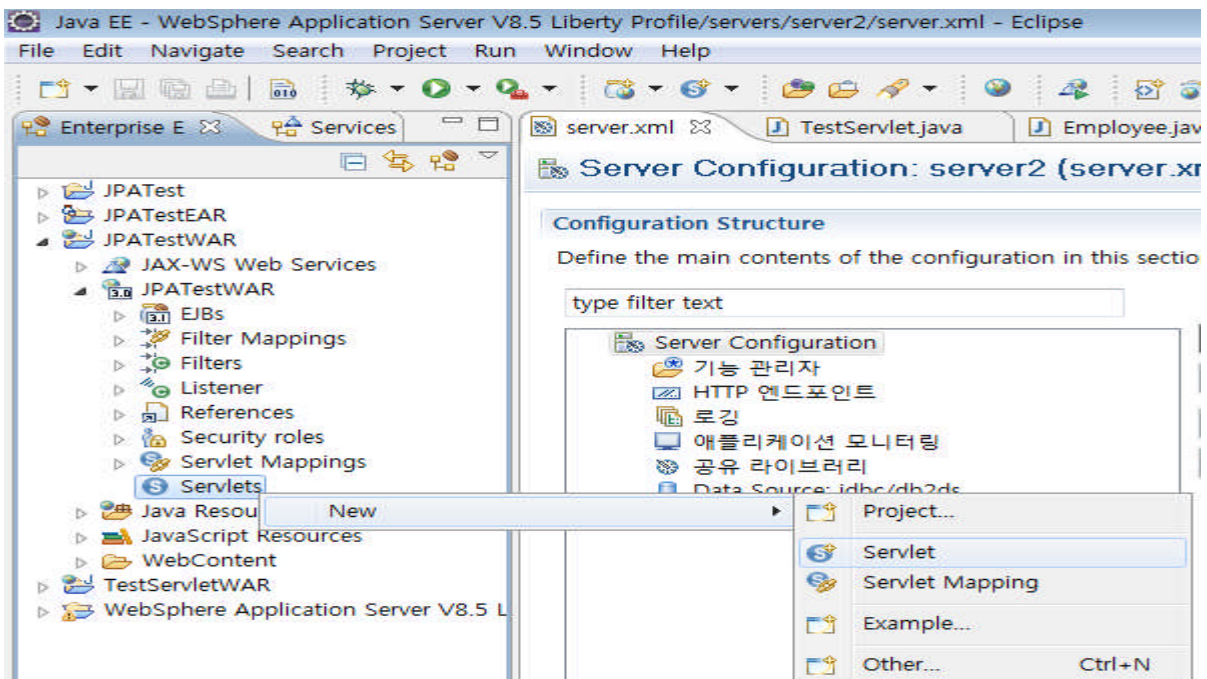
19. 이렇게 하시면 테스트를 위한 JPA Sample 애플리케이션 작성을 정상적으로 완료하신 것 입니다.

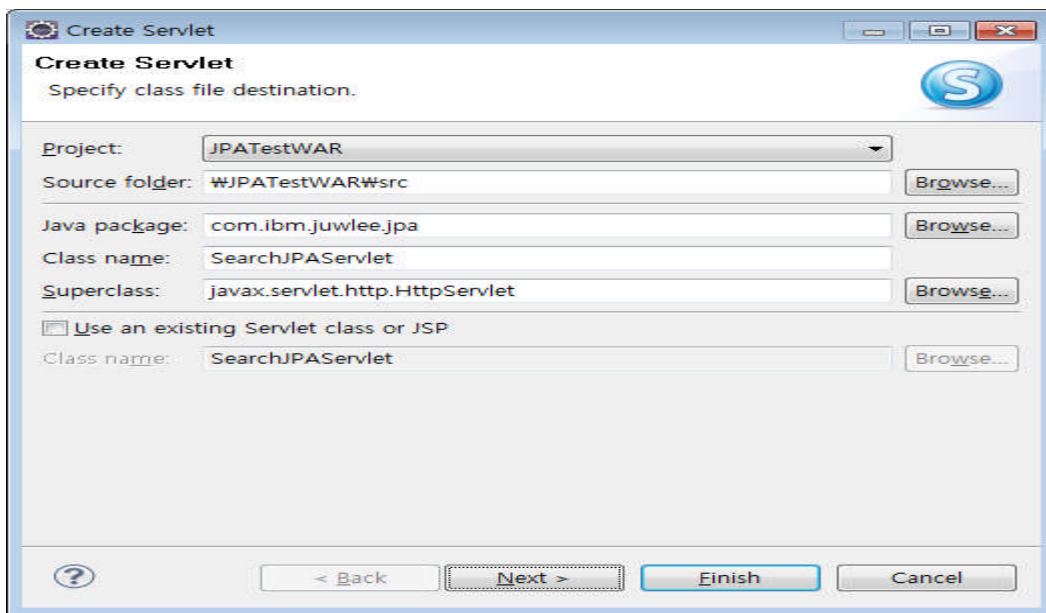
### 3) JPA Sample 테스트

1. 지금 까지 만든 JPA Sample 애플리케이션을 테스트하기 위하여 간단하게 동적 웹 프로젝트를 하나 작성합니다.



2. 만들어진 동적 웹 프로젝트에서 JPA Sample 애플리케이션을 호출하는 Servlet을 작성합니다.





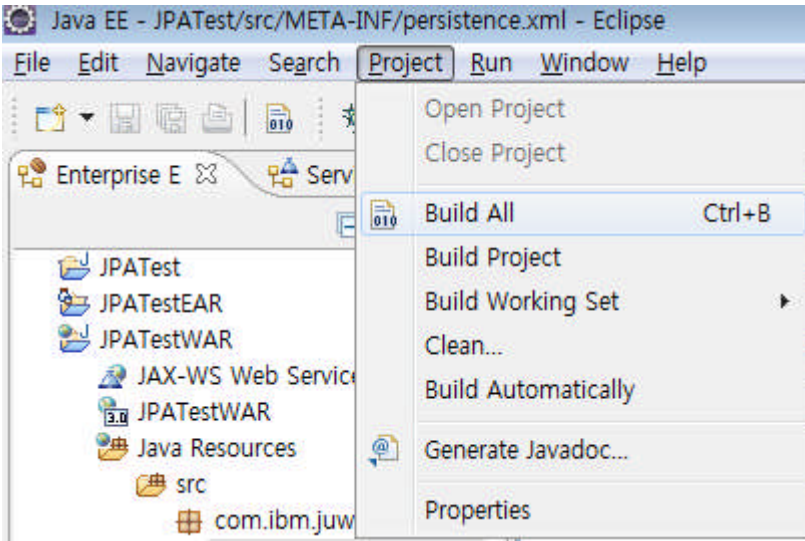
```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    ServletOutputStream out = response.getOutputStream();
    response.setContentType("text/html");
    out.println("<html><head><title>Hello World</title></head>");
    out.println("<body><h1>Hello World!!!</h1><br>");
    try {
        Employee e = SearchEmployee("000010");
        out.println("Hello " + e.getFirstname());
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    out.println("</html>");
}
```

```
@PersistenceUnit(unitName = "JPATest")
private EntityManagerFactory emf;
@Resource
private javax.transaction.UserTransaction utx;

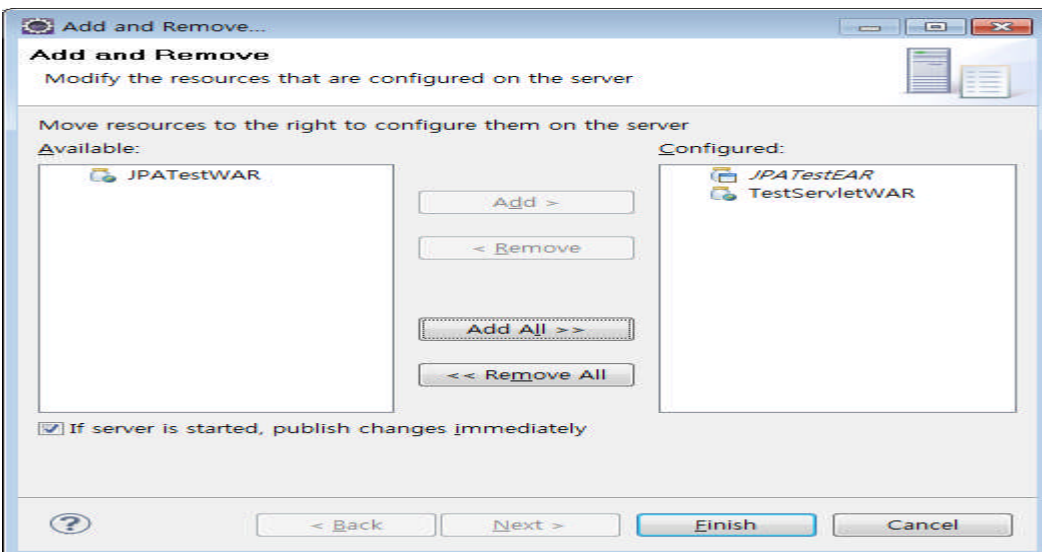
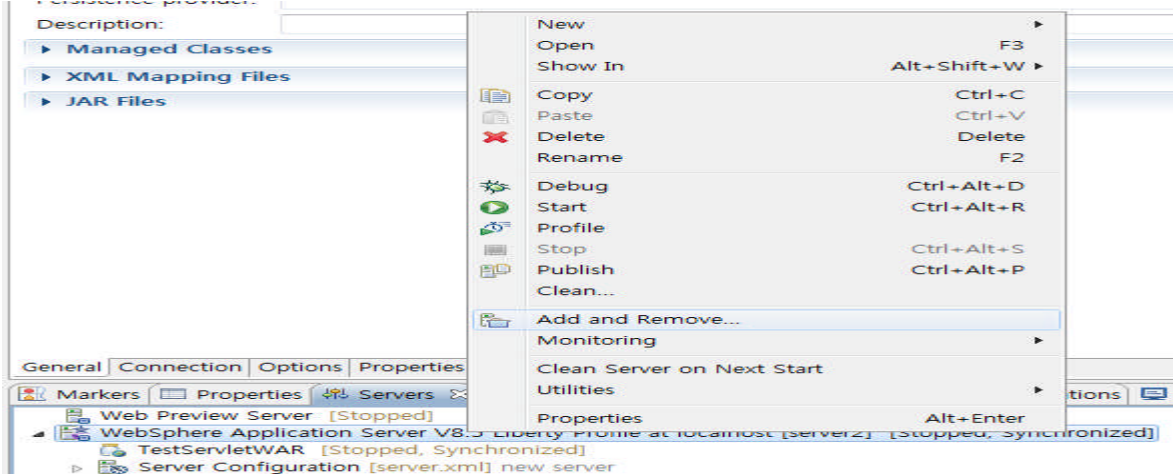
protected Employee SearchEmployee(String EmpNo) throws Exception {
    utx.begin();
    EntityManager em = emf.createEntityManager();

    Employee employee = (Employee) em.find(Employee.class, EmpNo);
    if (employee == null) {
        utx.commit();
        em.close();
        throw new Exception ("Employee is not found... EmpNo is " + EmpNo);
    }
    utx.commit();
    em.close();
    return employee;
}
```

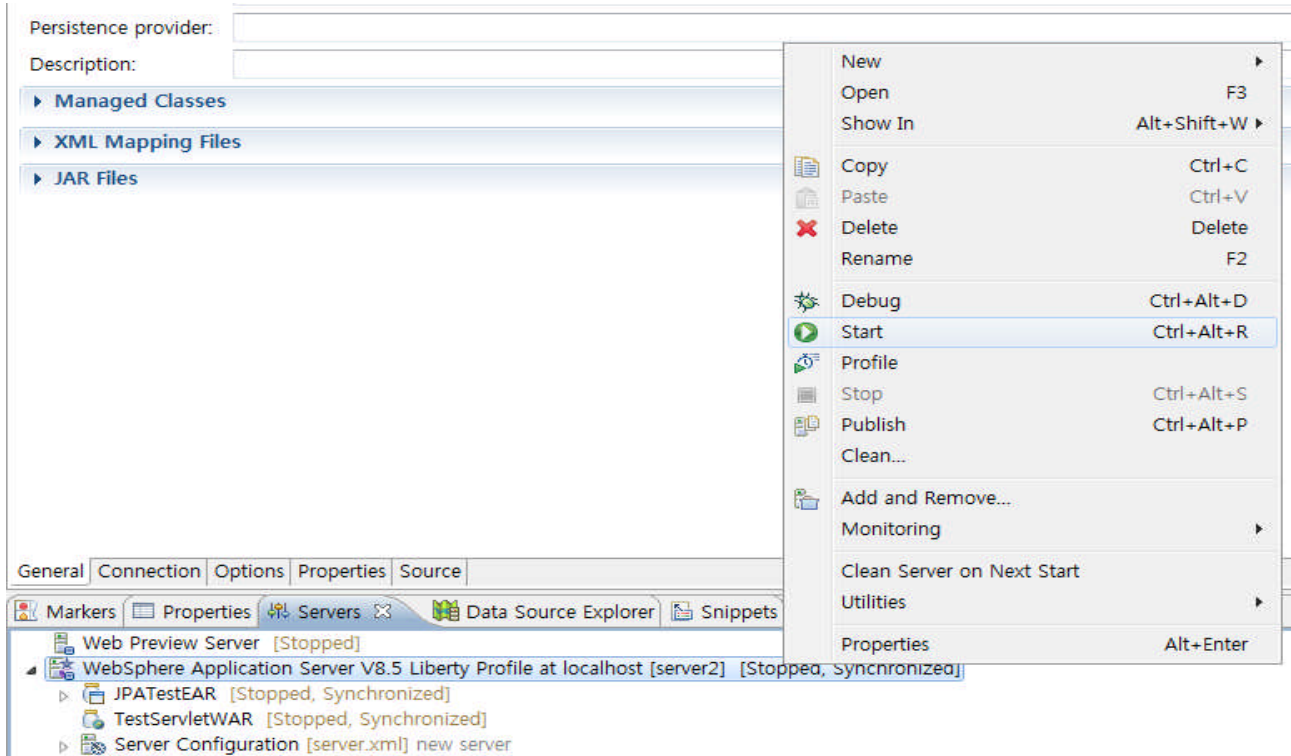
3. 해당 애플리케이션을 WAS에 배포하기 위하여 만들어진 프로젝트를 전부 빌드 합니다.



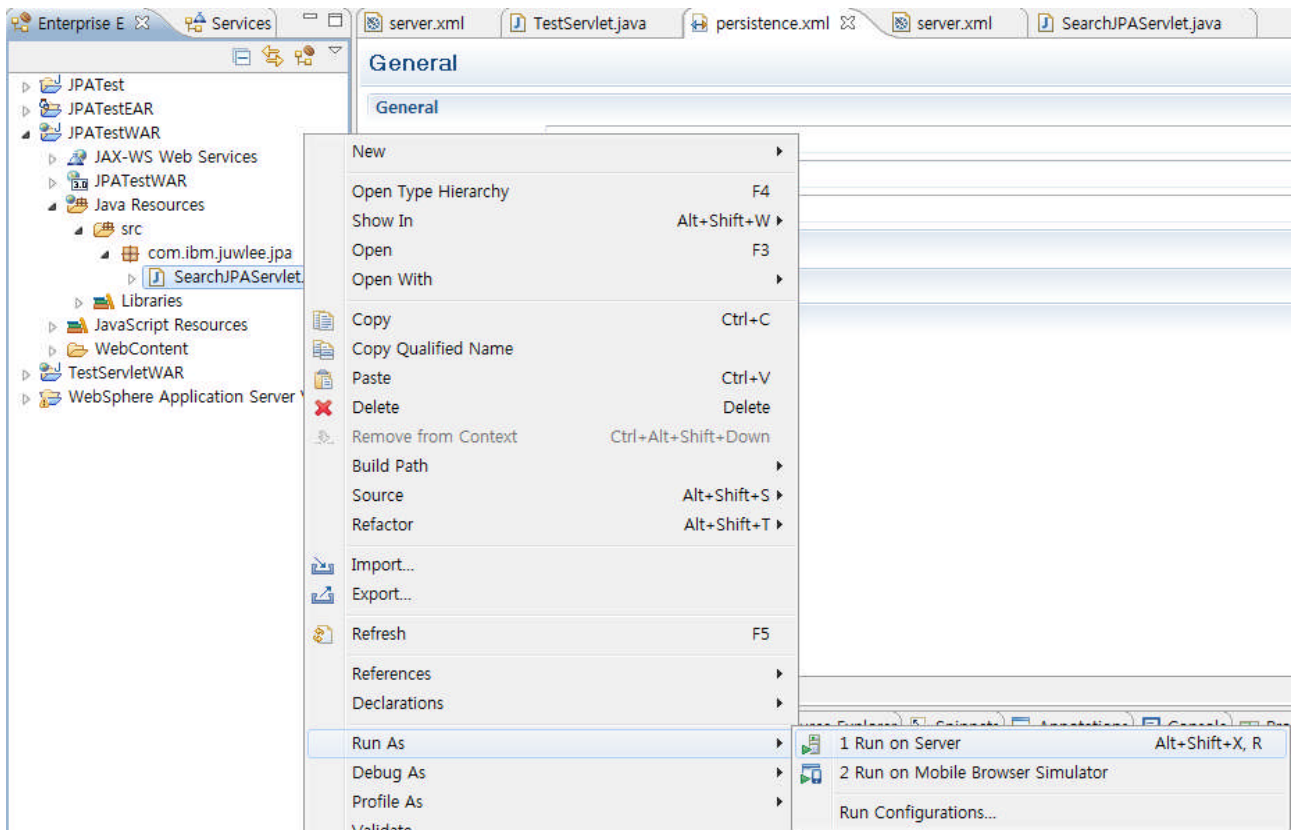
4. 빌드된 애플리케이션을 IBM WAS Liberty Profile 에 배포합니다.



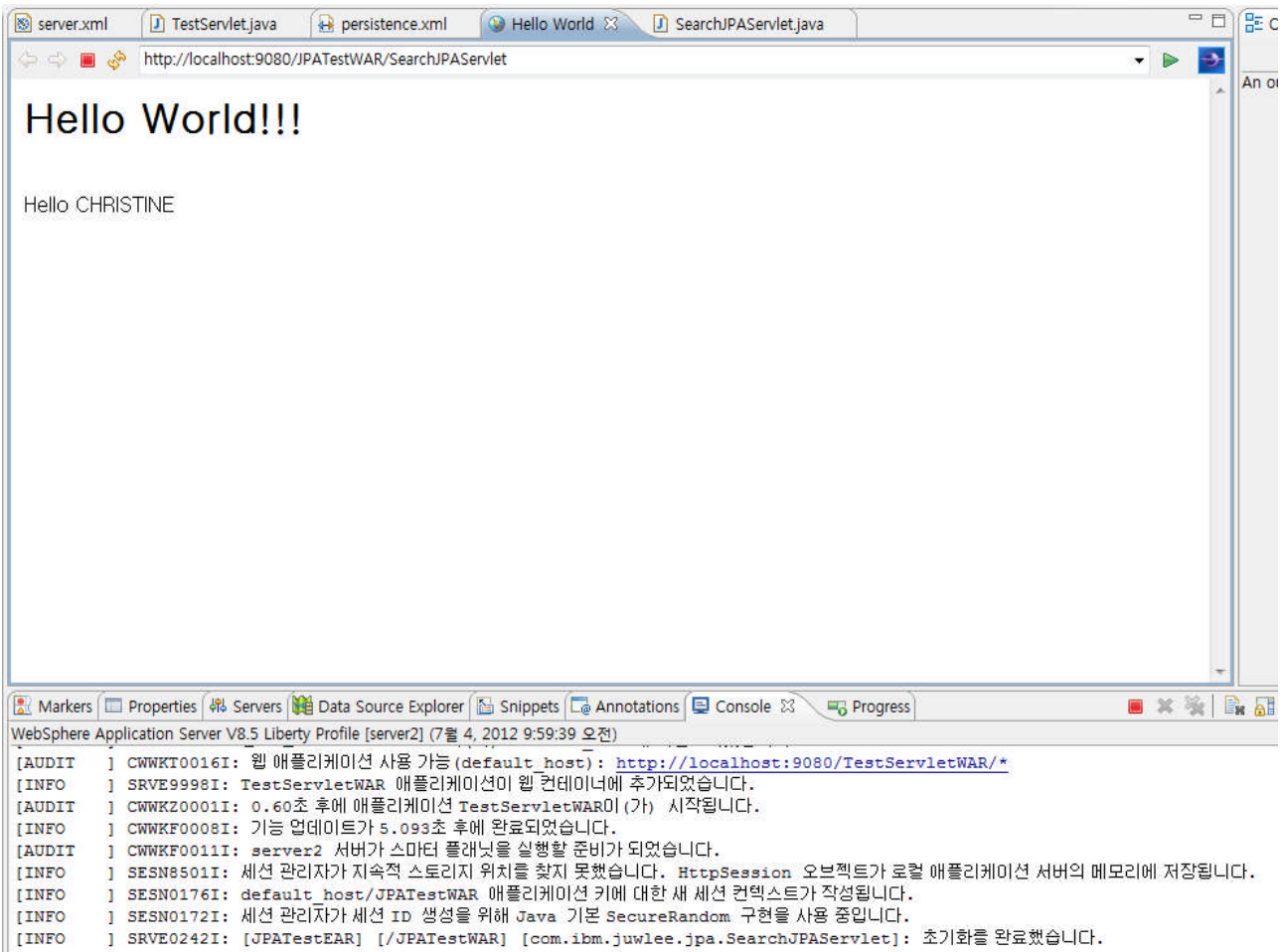
5. 배포가 완료되면 테스트를 위하여 IBM WAS Liberty Profile을 시작 시킵니다.



6. 테스트를 위해서 만들어둔 Servlet에서 마우스 우 클릭하여 Run As > Run on Server를 클릭하여 해당 Servlet을 수행합니다.



7. 하단과 같이 정상적으로 DB가 조회되어서 하단처럼 나오면 JPA Sample 테스트가 성공적으로 완료하신 것 입니다 .



#### 4) 결론

이번 강좌에서 IBM WAS V8.5 Liberty profile을 통한 JPA Sample 테스트를 수행하였습니다.

JPA 프로그래밍 자체가 조금 어렵기는 하지만 이번 강좌를 통해서 IBM WAS Liberty Profile를 이용하여 JPA 애플리케이션을 작성하고 운영하는 것은 그렇게 어렵지 않습니다. 설정하시는 것도 강좌를 통해서 확인해 보셨지만 JPA 기능만 추가하면 나머지 부분은 JDBC 테스트와 거의 동일합니다. 다시 말씀드려서 개발하기 참~ 쉽죠?

그럼 이번 강좌는 마무리하고 다음 강좌를 통해 IBM WAS V8.5의 다양한 장점을 더 만나보도록 하겠습니다. Go Go !!!



## 9) 참고 자료

1. 이 가이드는 IBM WAS V8.5 최초 사용자를 위한 기본 가이드입니다.
2. IBM WAS 자체에 아직 익숙하지 않으신 분들은 가급적 기본강좌인 '하나씩 쉽게 따라 해보는 IBM WAS v7' 강좌를 먼저 읽고 이 강좌를 읽으시는 것이 훨씬 이해에 됩니다.  
([http://www.websphere.pe.kr/xe/?mid=was\\_info\\_re&page=3&document\\_srl=800](http://www.websphere.pe.kr/xe/?mid=was_info_re&page=3&document_srl=800))
3. 가급적 IBM WAS V8.5 InfoCenter의 해당 카테고리를 한 번 읽어보고 난 후에 작업하시기 바랍니다.
4. InfoCenter – WebSphere Application Server V8.5  
(<http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp>)
5. InfoCenter – JPA for WebSphere Application Server  
([http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.base.doc/ae/cejb\\_jpanote.html?resultof=%22%6a%70%61%22%20](http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.base.doc/ae/cejb_jpanote.html?resultof=%22%6a%70%61%22%20))