

23-11-24

- 웹서버

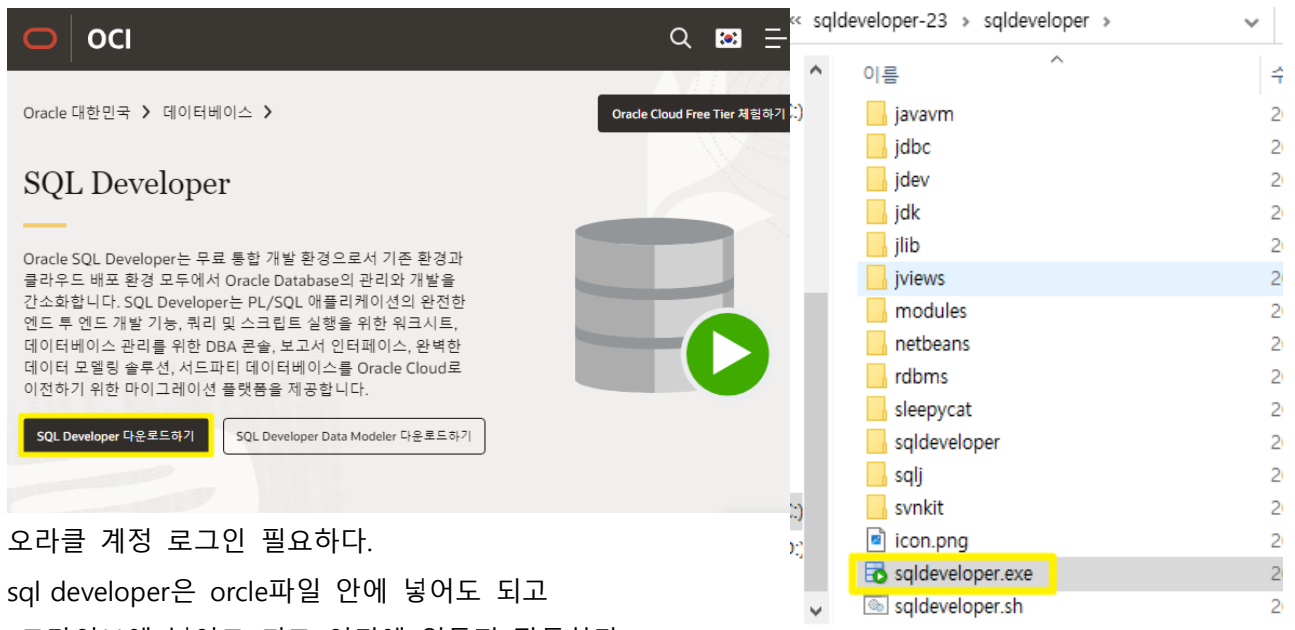
아파치 웹서버와 톰캣

Werserver port 80

Tomcat port 8090 (JSP할 때는 이렇게 잡기_오라클과 같은 포트8080)

톰캣 접속페이지 내 이름으로 출력되게 만들기

<오라클 sql developer 설치>



오라클 계정 로그인 필요하다.

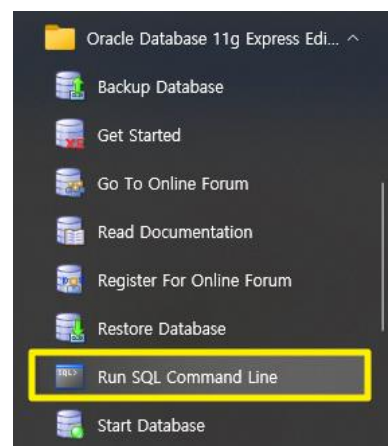
sql developer은 orcle파일 안에 넣어도 되고

c드라이브에 넣어도 되고 어디에 있든지 작동한다

실행파일 바탕화면에 바로가기 만들기

시작 > oracle에서 SQL command Line 실행

사용자 대해서 account 추가할거다



```

SQL*Plus: Release 11.2.0.2.0 Production on 금 11월 24 10:46
Copyright (c) 1982, 2014, Oracle. All rights reserved.

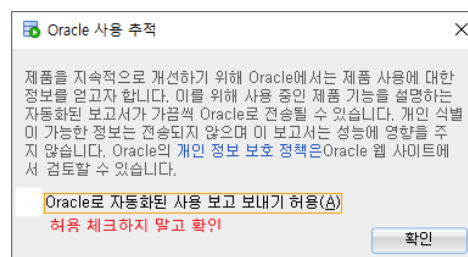
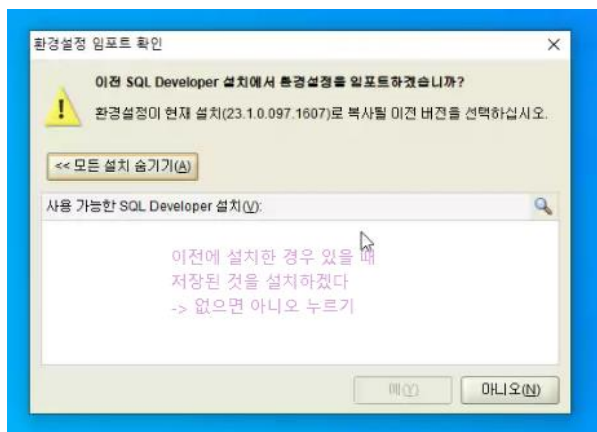
SQL> conn /as sysdba 처음 시작할 때 해야하는 명령어
Connected.
SQL> alter user hr account unlock identified by hr;
2
        사용자이름                비밀번호
SQL> alter user hr account unlock identified by hr;
User altered.

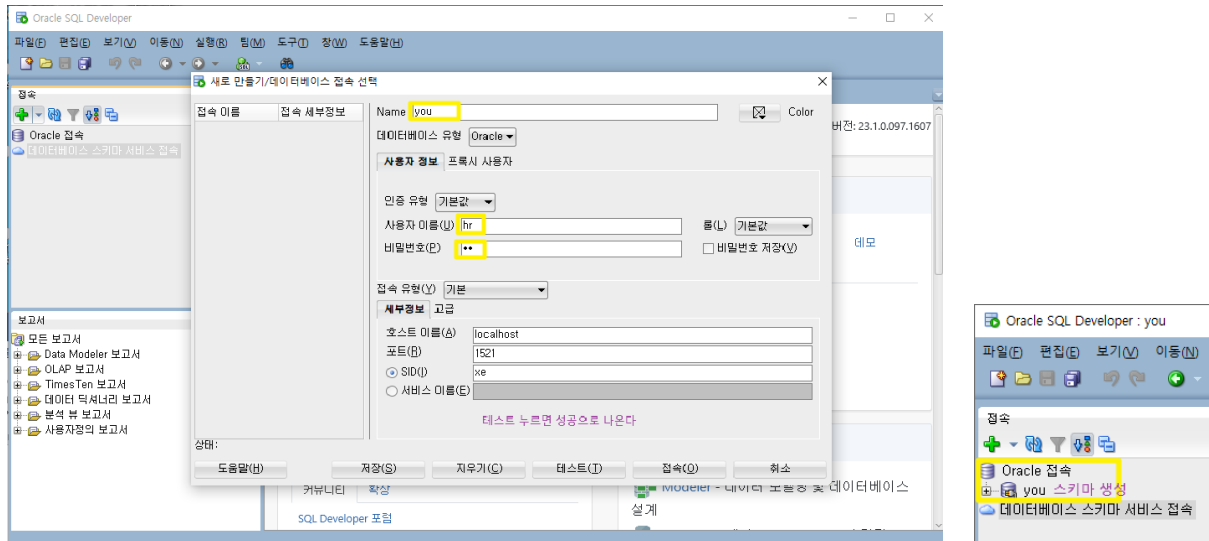
```

사용자이름과 비밀번호 설정 => User altered 가 나옴
 사용자이름을 hr로 하였고, 비밀번호도 hr로 설정하였다.

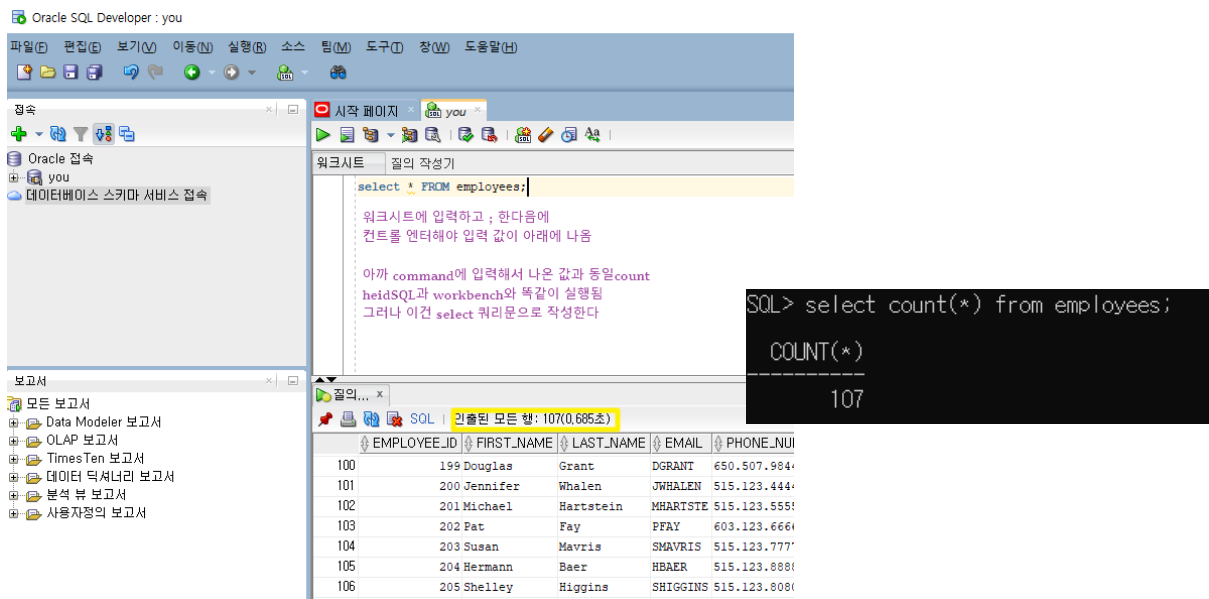
뭔가 작업할 수 있는 것이 없다 -> sql developer가 필요하다

<sql developer 설치>





왼쪽 접속 누르면 새로만들기에서 사용자이름과 비밀번호 설정해주기 -> 테스트 -> 상태 : 성공
스키마가 Oracle 안에 생성됨



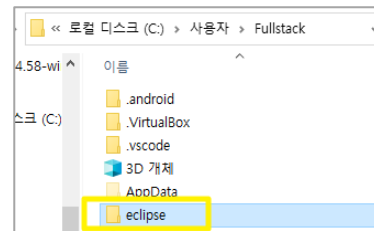
워크시트에 입력하고 ; 한다음에 컨트롤 엔터해야 입력 값이 아래에 나옴

인출된 모든 값은 command에 107이 나온 것과 동일하게 나온다

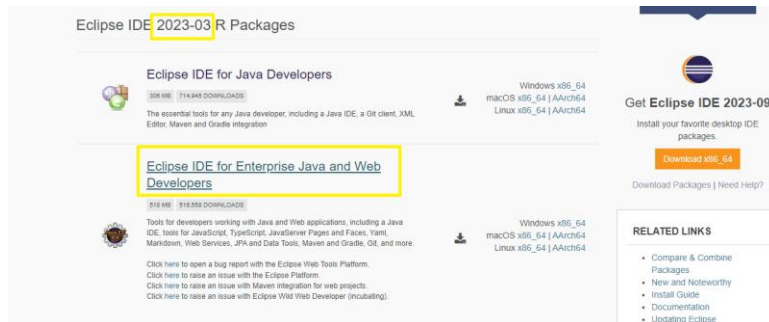
이클립스 vscode intelliJ

3개 다 설치하면 꼬이는 상황이 발생한다

그래서 이클립스를 C:\Users\Fullstack 이 경로에 붙여넣었다

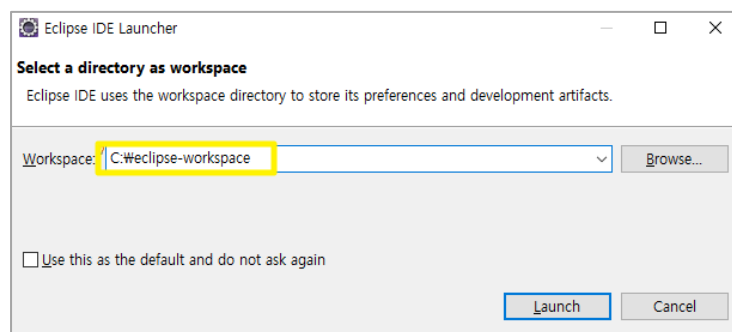


<이클립스 설치>



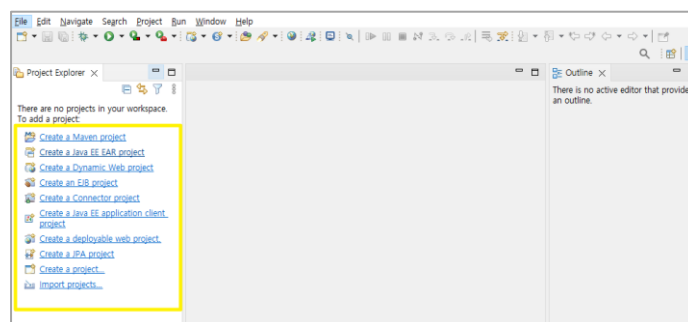
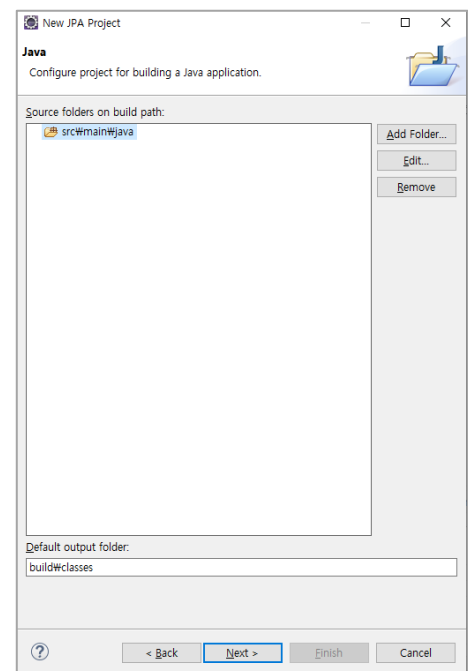
처음에 2021-06 버전을 다운받았는데 선생님이 실행하셨을 때 에러가 많이나서

2023-03 으로 다시 다운받았다. -> 64버전으로 다운받기



경로를 중간에 지우고 eclipse-workspace로 설정해주기

이클립스 용도로 따로 만들어라 해서 변경해줌



vs에서는 못보던 기능들이다

vs와 intelliJ는 IDE라서 프로젝트 생성이 주 목적이다.

-> create a JPA project

Target runtime

jre ▼ New Runtime...

java는 jvm(가상머신) jre(자바 실행하는 구동환경 설정) JDK(옆에 두개와 + 개발도구 포함한 것)
그중 jre로 타겟 설정해주기

New JPA Project

JPA Project
Configure JPA project settings.

Project name: jpa01

Project location
☒ Use default location
Location: C:\eclipse-workspace\jpa01 Browse...

Target runtime
jre ▼ New Runtime...

JPA version
2.2 ▼

Configuration
Default Configuration for jre ▼ Modify...
A good starting point for working with jre runtime. Additional facets can later be installed to add new functionality to the project.

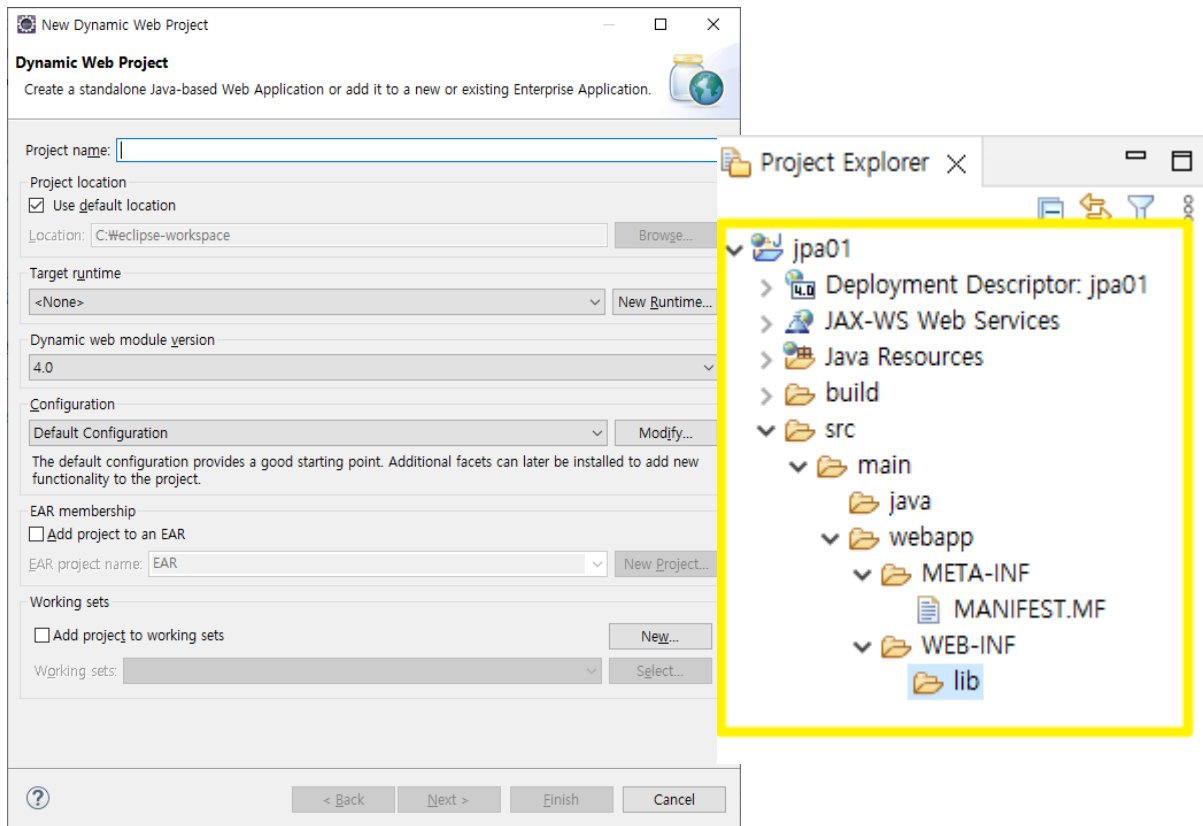
EAR membership
☐ Add project to an EAR
EAR project name: ▼ New Project ...

Working sets
☐ Add project to working sets New...
Working sets: ▼ Select...

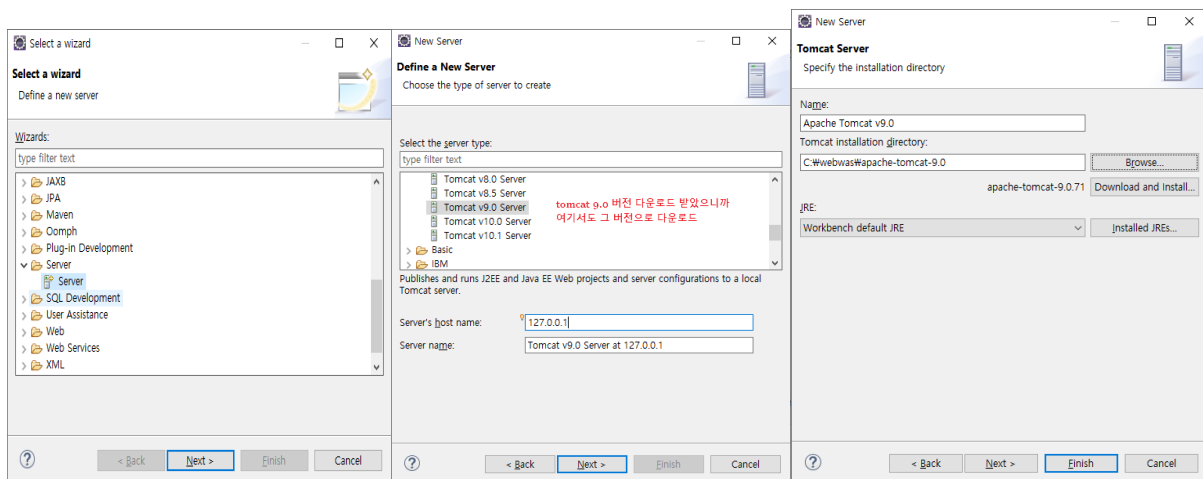
? < Back Next > Finish Cancel

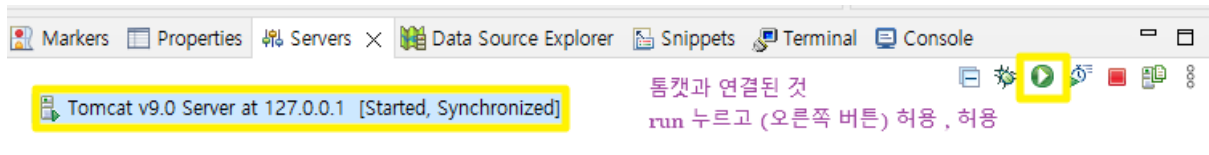
JPA project 설정해주기

데이터 베이스 무슨 프로그램과 연결할건지 선택할 수 있음
mysql과 연결하고 싶으면 mysql 선택하면 됨

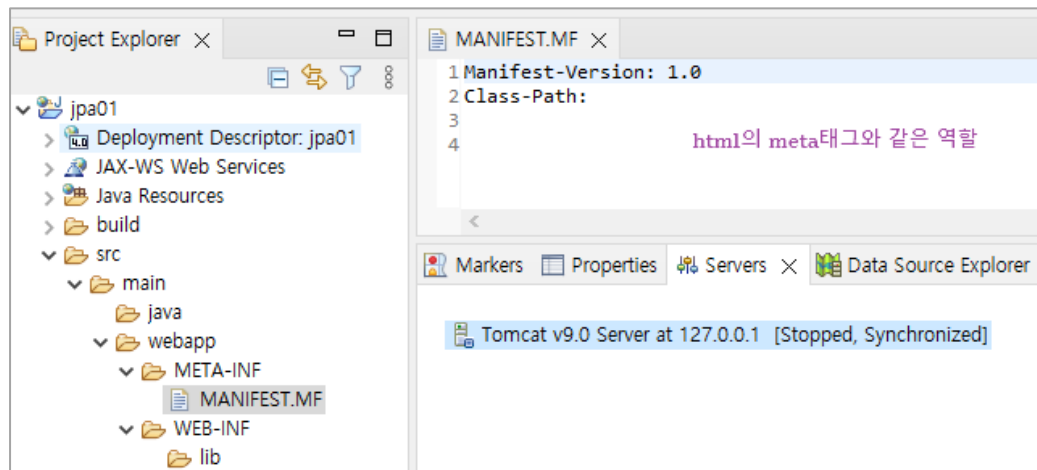
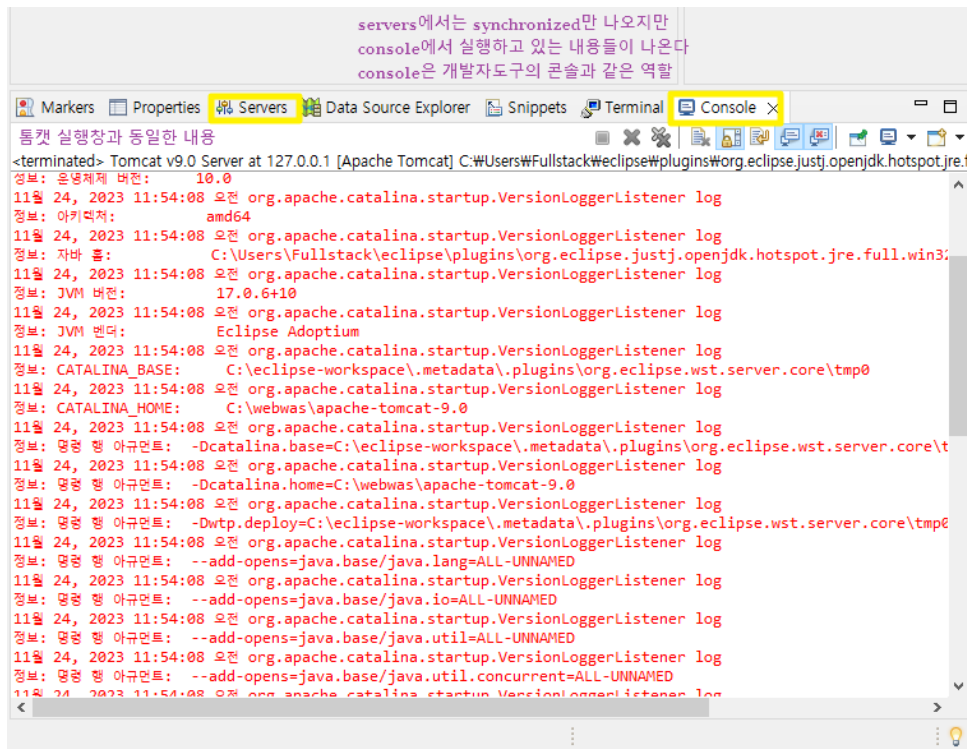


동적 웹 프로그램을 만들어라
계산작업이 만들어지는 프로그램 만들 때 다이나믹 웹 프로젝트인 것이다.

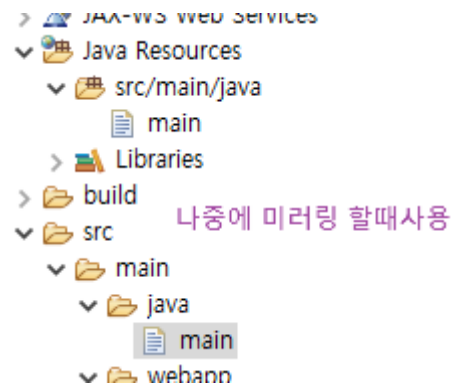
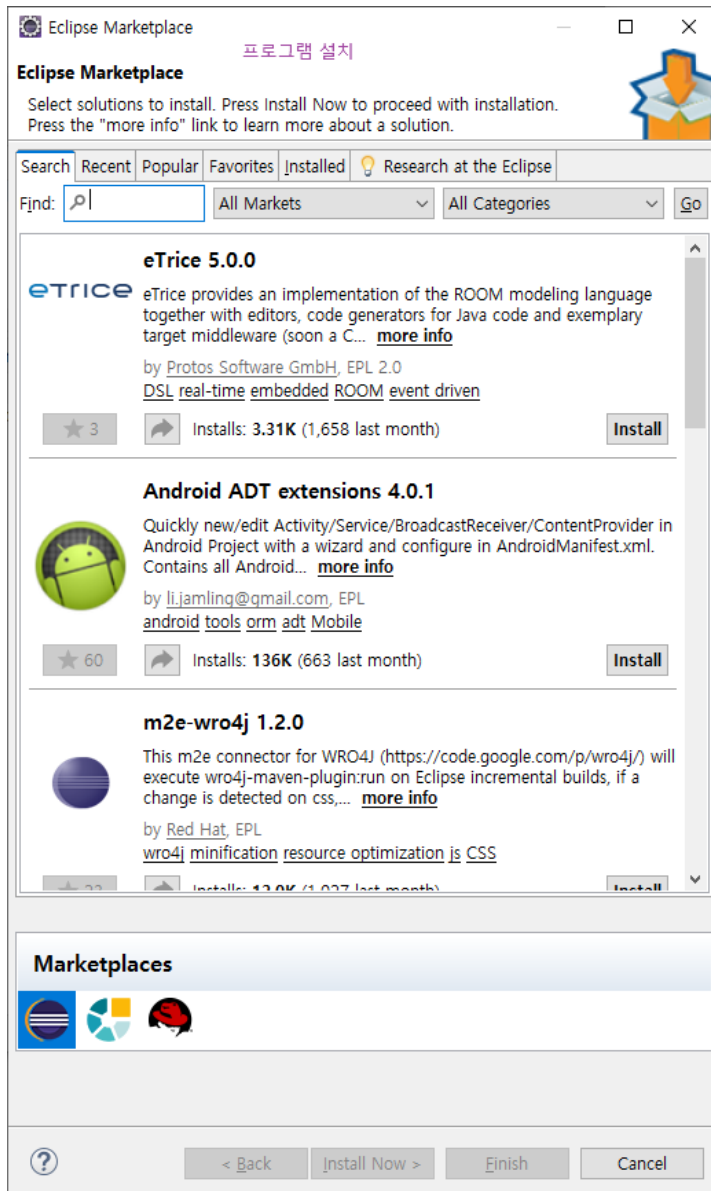
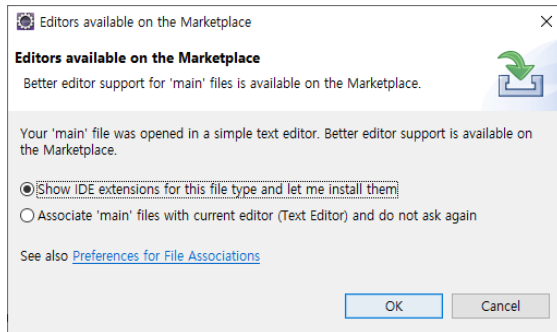


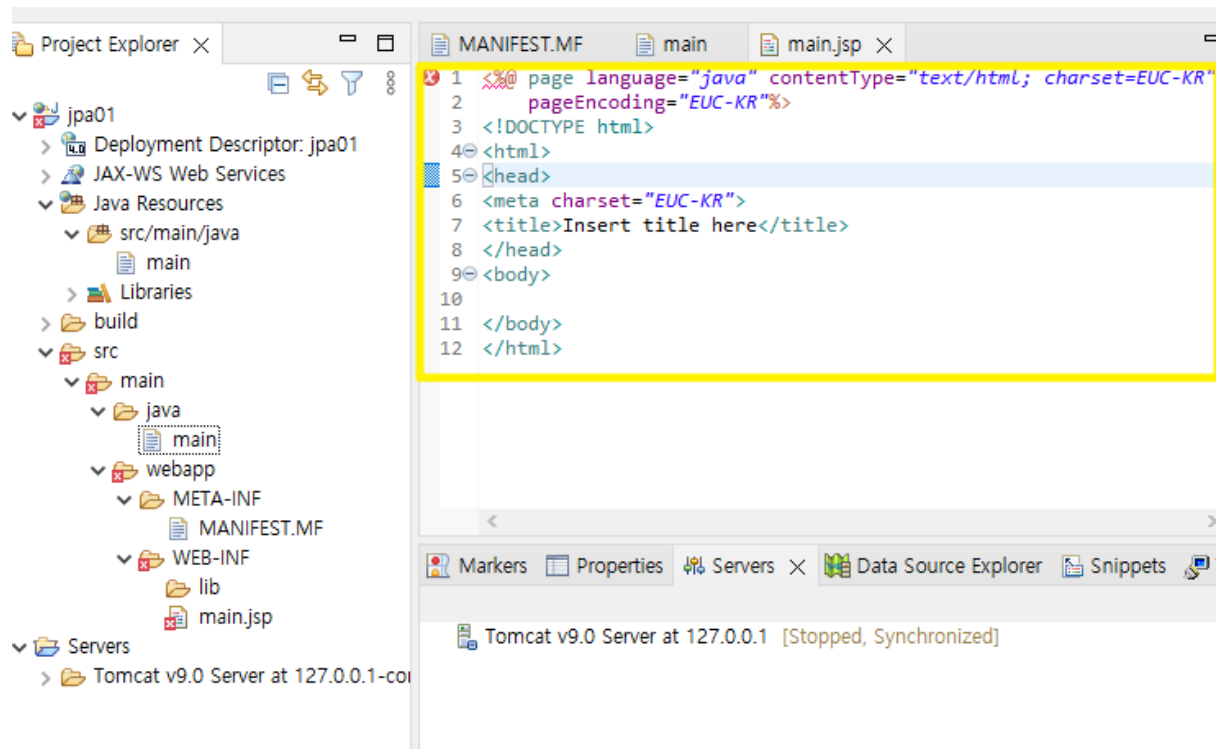


tomcat과 연결된 화면(run누르고 설치함)

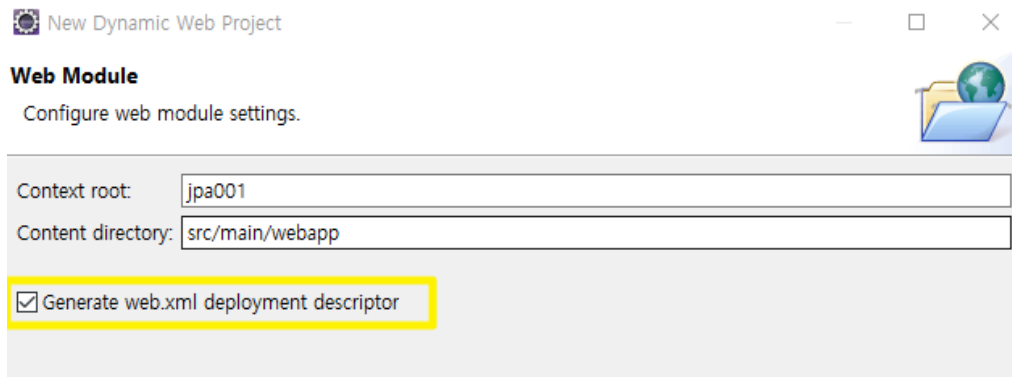


html의 meta 태그와 같은 역할을 한다.

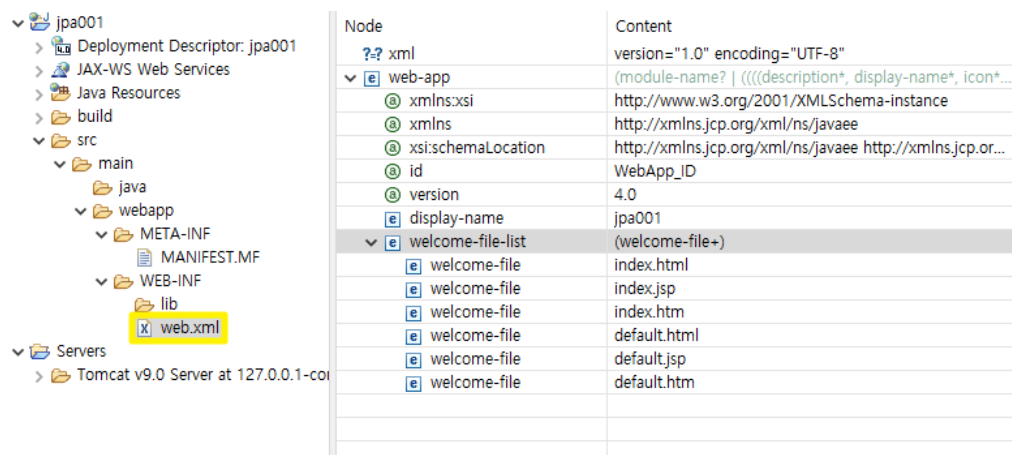




New file 만들기 > Dynamic web project로



체크를 해주면 web.xml파일이 생성되고, 각 설명이 되어 있는 것을 확인할 수 있다.



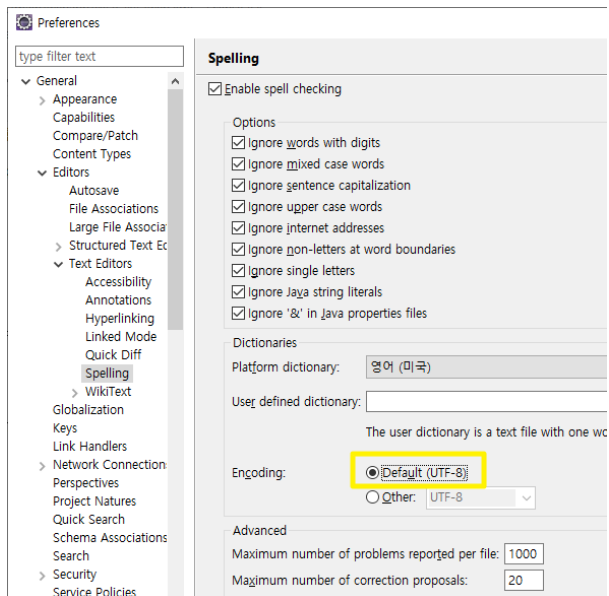
최종적으로 빌드도구의 형태로 만든다

Web.xml은 각 설명 달아놓은 파일이다 (이것 위치 파일 형태정도 알아 둘 것)

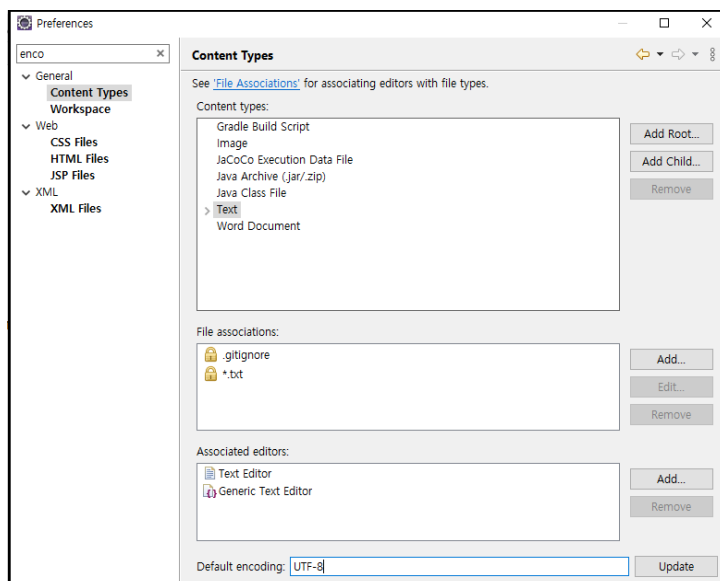
Html은 index.html로 잡아라

??? 공통서식 적용할 때 사용한다

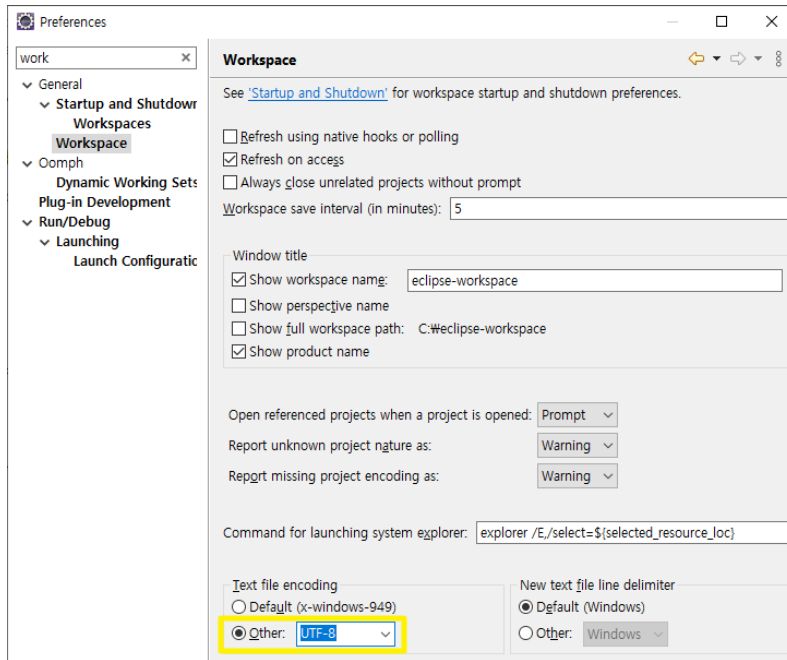
Window > preferences : 어떤모양으로 설정할 것인지



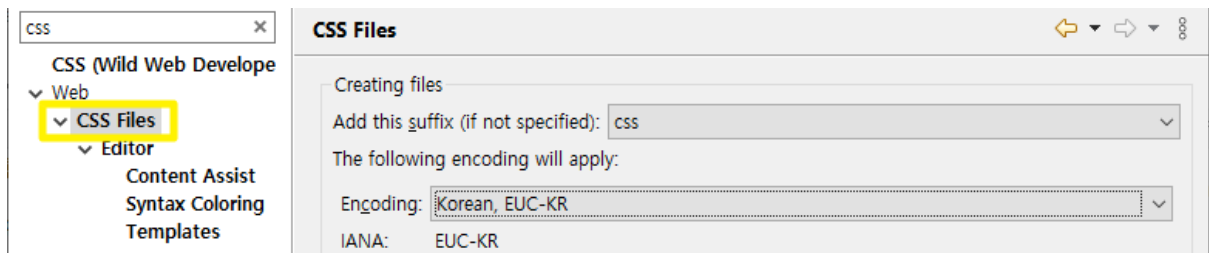
1. spelling -> UTF-8로 바꾼다. -> apply하기



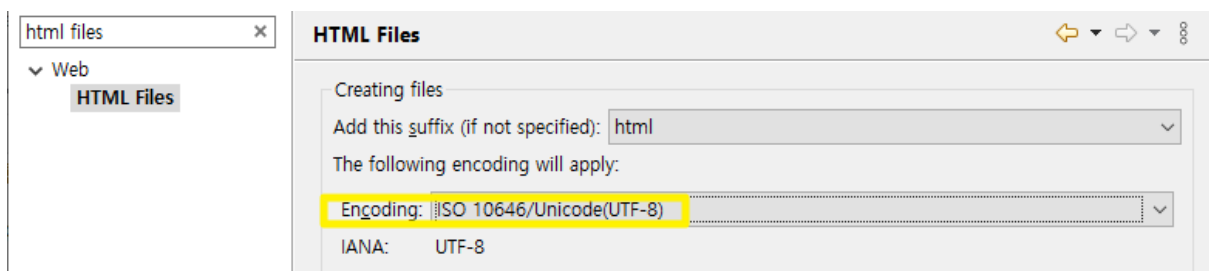
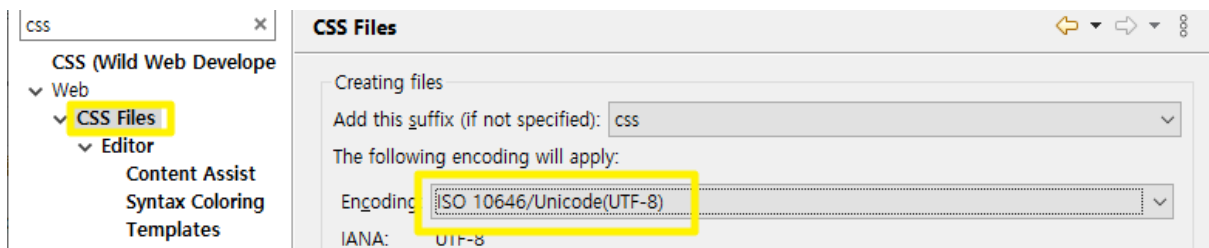
2. enco검색해서 나오는 text에 밑 default 값을 UTF-8로 바꿔준다.



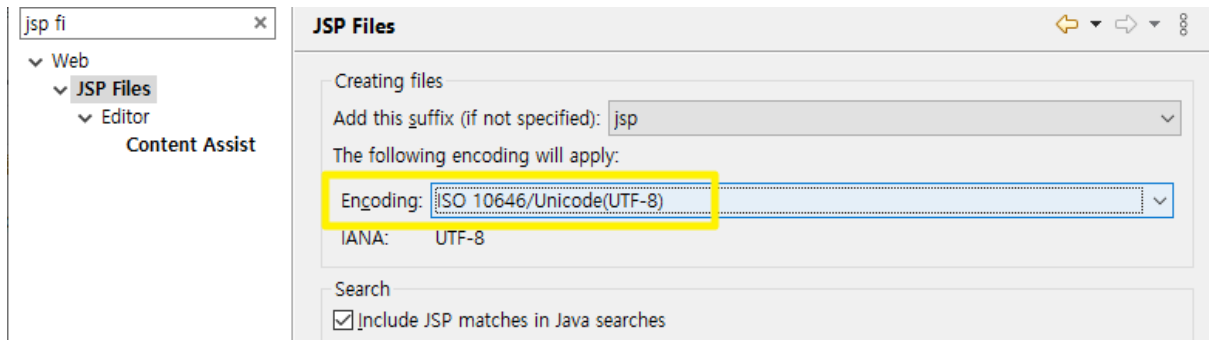
3. Workspace 에 UTF-8로 바꿔준다



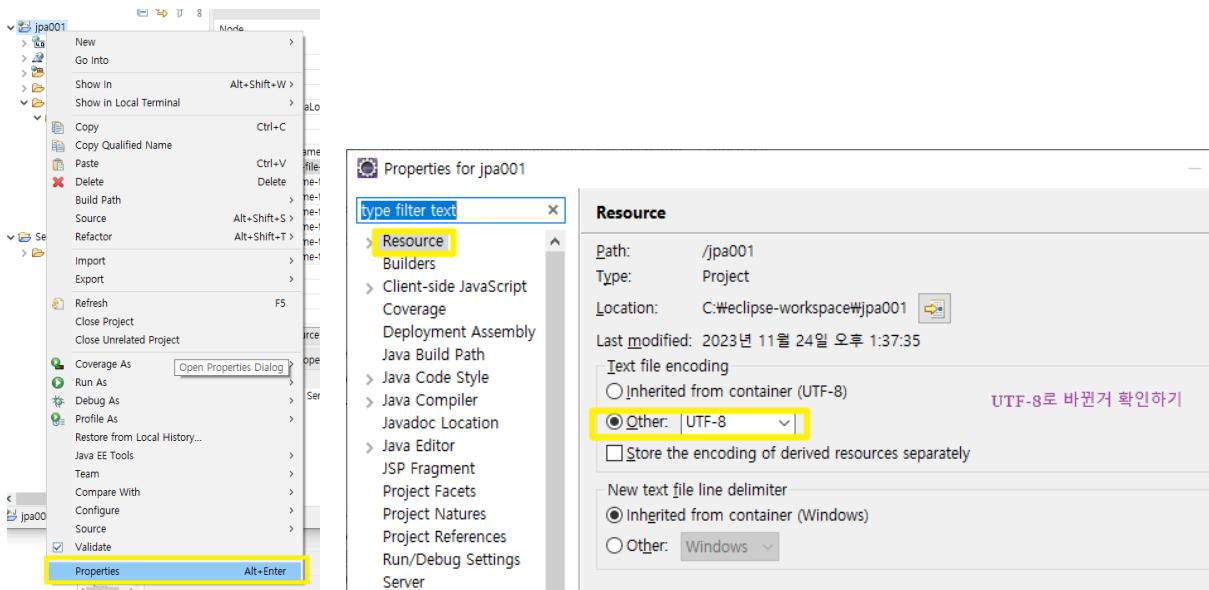
4. CSS Files에 들어가면 처음에 korean으로 되어 있는 것을 아래와 같이 바꿔준다.



5. HTML Files

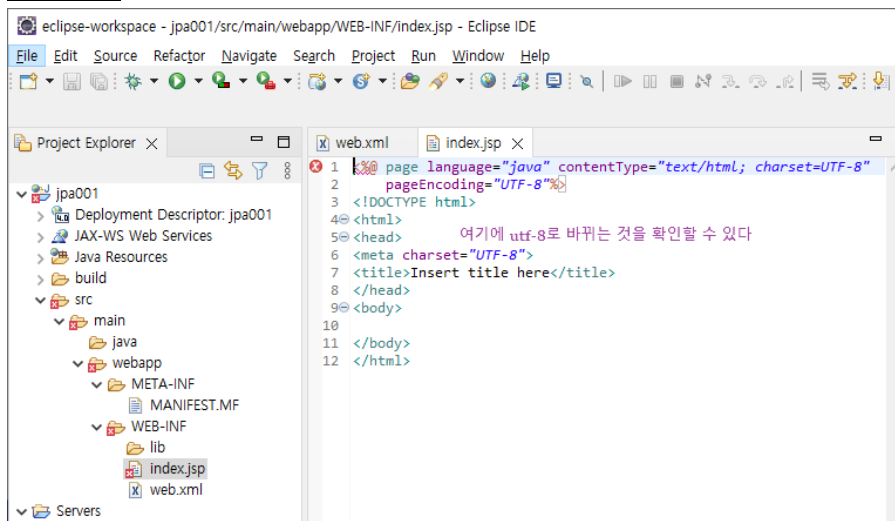


6. JSP Files



7. properties에서 -> 옆과 같이 UTF-8이 되어 있는 것을 확인한다

Index.jsp



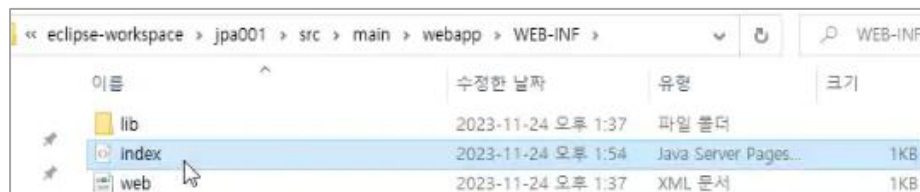
Index.jsp 파일로 메인페이지 만들고 있다

Pre 태그 입력한 그대로 보여줌(br태그 쓰지 않아도)

ㄱ -> 한자키 : 기본 특수문자

ㅇ -> 한자키 : c 특수문자 (푸터에 필요한 특수기호)

index파일 위치



Jsbe 사용하기 위해 넣는 명령어다 => 구조 알아보자

```
<% page import="java.sql.*, javax.sql.*" %>
```

```
<%
```

```
    Connection con=null;
```

```
    PreparedStatement pstmt=null;
```

```
    ResultSet rs=null;
```

```
    try{
```

```
        Class.forName("oracle.jdbc.OracleDriver");
```

```
        con=DriverManager.getConnection
```

```
        ("jdbc:oracle:thin:@//localhost:1521/xes", "system", "1234");
```

```
        pstmt=con.prepareStatement("select * from member_tbl_02");
```

```
        rs=pstmt.executeQuery();
```

```
    }%>
```

```

Connection con=null;
PreparedStatement pstmt=null;
ResultSet rs=null;

```

오라클 데이터베이스 문서 꺼내오라는 명령문 함수이다.???

```
"select * from member_tbl_02"
```

필드의 내용을 가지고 와라, member_tbl_02 테이블의 자리 내용을 가지고 와라

```
("jdbc:oracle:thin:@//localhost:1521/xe","system","1234");
```

Jdbc : 자바데이터베이스컬렉션 / 오라클 데이터베이스에 있는 것을 찾아서 가지고 올거다

localhost:1521 -> 데이터베이스 기본이름

system 사용자 이름 : 오라클 처음 만들 때 생성되었다

초기에는 웹서버에 DB서버가 붙어있는 것으로 되어 있다.

```
<% page import="java.sql.*, javax.sql.*" %>
```

Import는 외부에서 만들어진 것을 내것에서 사용할 때 사용

java.sql.*, sql과 관련된 명령어 추가하라는 것

```

try {
} catch ( ) {
} finally {
}

```

Java 예외처리

```

try {
    예외가 발생할 수 있는 코드
} catch (Type 1) {
    1번 유형의 예외처리 코드
} finally {
    예외가 발생하든 하지 않든 실행코드
}

```

이렇게 사용하는 구문을 jsp에서 볼 수 있을 것

- Try : 자바에서 예외 처리할 때 사용 { 안에는 예외가 발생할 수 있는 코드를 작성
- Catch (type 1) { 1번 유형의 예외처리 코드이다
- Finally { 예외가 발생하던 발생하지 않던 실행되는 코드

위에 Try 부분과 연결되는 부분이다.

```

Connection con=null;
PreparedStatement pstmt=null;
ResultSet rs=null;

try{
    Class.forName("oracle.jdbc.OracleDriver");
    con=DriverManager.getConnection
        ("jdbc:oracle:thin:@//localhost:1521/xe","system","1234");

    pstmt=con.prepareStatement("select * from member_tbl_02");

    rs=pstmt.executeQuery();
}
%>

```

null 부분은 조건 설정해준 것

```

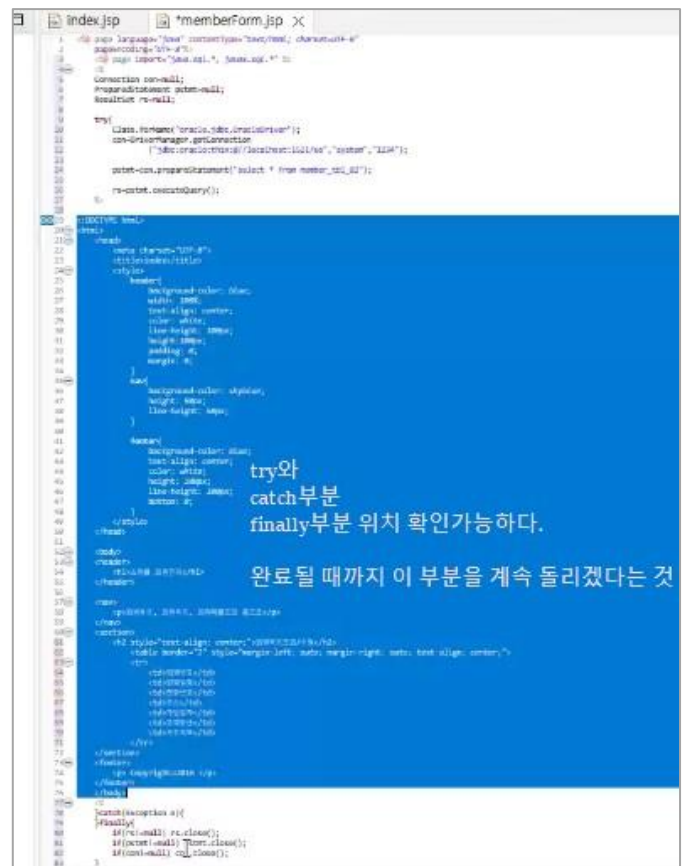
<%
} catch(Exception e){
} finally{
    if(rs!=null) rs.close();
    if(pstmt!=null) pstmt.close();
    if(con!=null) con.close();
}
%>

```

JSP는 라인별로 끝나는 것이다

처음시작과 끝 안에 함수 등을 다 넣을 수 있다

이클립스 바로 사이트 연결하는 방법



<nav>

```

<nav>
    <a href = "joinForm.jsp">회원등록</a>
    <a href = "memberForm.jsp">회원목록조회/수정</a>
    <a href = "saleForm.jsp">회원매출조회</a>
    <a href = "index.jsp">홈으로</a>
</nav>

```

 한칸씩 띄고 시작하는 것

joinForm.jsp

```
try{
    Class.forName("oracle.jdbc.OracleDriver");
    con=DriverManager.getConnection

    ("jdbc:oracle:thin:@//localhost:1521/xe","system","1234");
    바뀌지 않는다

    pstmt=con.prepareStatement("select * from member_tbl_02");

    바뀔 수 있다 가지고 오는 테이블마다 달라질 것이기 때문에

    int custno=100001;
    if(rs.next()){
        custno=rs.getInt(1)+1;
    }
    추가해줌
```

Catch와 finally부분은 바뀔 것이 없다 (예외처리해줄 것이기 때문에)

```
<form name="f" action="joinProcess.jsp" method="post" onsubmit="return
checkForm(this)">
```

내 것 맞는지 확인하는 것이 post

```
<tr>
    <td>회원번호</td>
    <td><input type="text" name="custno" value="<%= custno %>"></td>
```

매개변수나 파라미터 값 들어온다

modForm.jsp

```
pstmt=con.prepareStatement("select * from member_tbl_02 where custno=?");
pstmt.setString(1, custno);
```

1, custno가 위에 where에 들어가는 부분인 것

```
pstmt=con.prepareStatement("select * from member_tbl_02 where custno=>= 100");
pstmt.setString(1, custno);
```

원래 이렇게 쓴다
회원정보 수정할 거라서 어느 값이 들어올지 몰라서 '?'를 써준다

```
<% if(){ }%>
```

이렇게 있으면 아래 <% %> 이 부분이 더 추가가 되어야 함 if 문이기 때문에

seleForm.jsp

```
String sql="select m1.custno, custname, grade, sum(price)"
        +" from MEMBER_TBL_02 m1 inner join money_tbl_02 m2"
        +" on m1.custno=m2.custno"
        +" group by m1.custno, m1.custname, m1.grade order by sum(price) desc";
```

+ : 문장연결, 너무 길어서 한줄이 아닌 여러줄로 나눠쓴다는 것(위에는 한줄내용임)

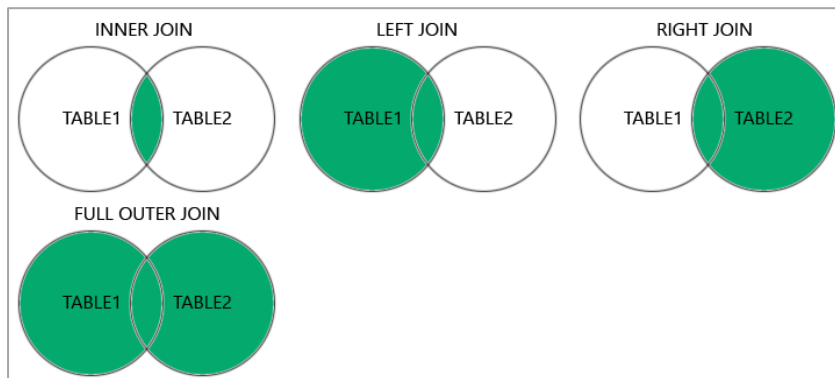
M1.custno(기준)에 쫓아가거나 m2 테이블에 포함된 것일 것이다.

M1.custno는 m2에도 있을 것이다

Inner join : m1과 m2를 한번에 묶어서 출력하라는 것

(where m1.custno=m2.custno 와 같은 의미의 문장이다)

Outer join :



왼쪽에 붙는지, 오른쪽에 붙는지에 따라서 join 설정을 다르게 해줄 수 있다.

두개의 테이블을 하나의 테이블처럼 만들어준 것

중복 최소화 하려고 DB하는데 엑셀로 나오면 의미가 없다, 그런데 엑셀로 출력할 때가 필요할 때도 있다 그 때 필요한 것이 inner join, outer join이다.

Group 넘버로 하나 묶고 이름으로 하나 묶고..

(나라이름의 튜플이 있을 때, 중복되는 값을 하나로 묶을 때 group by를 사용한다.

그러면 하나의 나라의 여러값이 나올 수 있는 것)

sum(price)를 desc내림차순으로!!

Select *필드* *from* *테이블* *where* *조건* *group by* *묶음* *order by* *정렬*

sql 문장으로 쓸 수 있는 최고 긴 문장이다.

- **Select 필드**

A,B,C 찾을거야 (필드)

여러 필드를 가지고 올 수 있다.

- **Group by 묶음**

하나의 그룹으로 묶어서 찾는것

- **Order by 정렬**

하나로 묶지만 정렬이 되어 있지 않으면