



# 제11장 커넥션 풀(DBCP)

# 1. DAO, DTO

## 1. DAO

Data Access Object -> DB에 직접 접근하여 DB와 관련된 일(로직)을 하는 객체를 칭한다.

데이터 베이스에 접속해서 데이터 추가, 삭제, 수정 등의 작업을 하는 클래스입니다.

일반적인 JSP 혹은 Servlet 페이지 내에 위의 로직을 함께 기술할 수도 있지만, 유지보수 및 코드의 모듈화를 위해 별도의 DAO클래스를 만들어 사용 합니다.

## 2. DTO

Data Transfer Object -> DAO가 DB에 접근하여 가져온 데이터를 난잡하게 가져오지 않고, 클래스로 만들어서 가져오는 객체를 칭한다.

DAO클래스를 이용하여 데이터 베이스에서 데이터를 관리할 때 데이터를 일반적인 변수에 할당하여 작업할 수도 있지만, 해당 데이터의 클래스를 만들어 사용 합니다.



## 2. 커넥션 풀(DBCP)



### 1. 커넥션 풀을 이용한 DB연동

Pool은 웅덩이라는 뜻을 가지고 있다.

클라이언트에서 다수의 요청이 발생할 경우 데이터베이스에 부하가 발생하게 됩니다.

이러한 문제를 해결 하기 위해서 커넥션 풀(DataBase Connection Pool)기법을 이용합니다.

자바 강의 때 스레드 풀에 대한 개념을 배웠다. 역시 이 데이터베이스 커넥션 풀 개념도 비슷하다.

다시 말해, 미리 생성해 놓고 클라이언트들이 접속할 때 가져다 쓰는 개념이며, 다 쓰고 난 커넥션 객체를 다시 재활용을 한다.



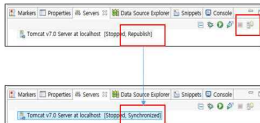
ex) 직장인이 한정된 1시간의 점심시간을 지낸다. 근데 수타 짜장면이라면 30분 이상 걸릴지도 모르니 미리 만들어 둬으로써 그 시간을 줄일 수가 있을 것이다.

# 3. 커넥션 풀(DBCP) 코드

## 1. 코드 작성하기

tomcat컨테이너가 데이터베이스 인증을 하도록 context.xml 파일을 열어서 아래의 코드를 추가한다  
아울러 publish를 해준다. 반드시 해당 프로젝트 lib폴더에 jdbc드라이버를 추가해야 한다.

```
<Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
-->
<Resource name="jdbc/dbtool"
  auth="Container"
  driverClassName = "com.mysql.jdbc.Driver"
  url = "jdbc:mysql://localhost:3306/sqlldb"
  username = "root"
  password = "1234"
  type = "javax.sql.DataSource"
  maxActive = "100"
  maxIdle="10"
  removeAbandoned="true"
  maxWait = "5000"
/>
```



## 4. ConnectionPool로 연결할 DB속성

속성	설명
name	DataSource에 대한 JNDI 이름
auth	인증 주체
driverClassName	연결할 데이터베이스 종류에 따른 드라이버 클래스 이름
factory	연결할 데이터베이스 종류에 따른 ConnectionPool 생성 클래스 이름
maxActive	동시에 최대로 데이터베이스에 연결할 수 있는 Connection 수
maxIdle	동시에 idle 상태로 대기할 수 있는 최대 수
maxWait	새로운 연결이 생길 때까지 기다릴 수 있는 최대 시간
user	데이터베이스 접속 ID
password	데이터베이스 접속 비밀번호
type	데이터베이스 종류별 DataSource
url	접속할 데이터베이스 주소와 포트 번호 및 SID

다른 속성들은 고정적으로 사용되며, 주로 `driverClassName`, `user`, `password`, `url`만 변경해서 사용합니다.

## 5. 커넥션 풀 사용

### Connection Pool 사용하기

더 이상 아래와 같이 DB연결은 안 해도 된다.

```
String url = "jdbc:mysql://localhost:3306/testdb";
String user = "root";
String password = "1234";
String driver = "com.mysql.jdbc.Driver";

Class.forName(driver);
connection = DriverManager.getConnection(url, user, password);
```

DBCP를 활용하여 단 한번 만들어놓으면 언제든지 연동이 가능하다.

```
try {
    context = new InitialContext();
    dataSource = (DataSource)context.lookup("java:comp/env/"+ "jdbc/dbtool");
    conn = dataSource.getConnection();
    System.out.println("DBCP 연동 성공");
} catch (Exception e) { e.printStackTrace(); }
```

감사합니다.

