

커넥션 풀 (Connection Pool) 기법

DB 연동시에 Connection 객체를 만드는데 시간이 오래 걸린다.

클라이언트에서 다수의 요청이 들어오면 일일이 Connection 객체를 생성하게 되면

데이터베이스에 부하가 발생하게 된다. 이를 위해서 커넥션 풀 기법을 사용한다.

지금까지는 아래와 같이 1단계(드라이버 로드) ~ 7단계(연결해제) 과정을 거쳤는데

커넥션 풀 기법을 통해서 1단계(드라이버 로드)와 2단계(Connection=DB연결 객체)를 간단히 작성

```
<%!  
String driver = "oracle.jdbc.driver.OracleDriver";  
String url    = "jdbc:oracle:thin:@127.0.0.1:1521:xe";  
String uid    = "scott";  
String upw    = "tiger";  
%>  
<%  
Connection conn = null;  
Statement stmt = null;  
ResultSet rs = null;  
String sql = "SELECT * FROM EMP";  
try{  
    Class.forName(driver); //1단계 : 드라이버 로드  
    conn = DriverManager.getConnection(url, uid, upw); // 2단계 : DB연결객체  
    stmt = conn.createStatement(); // 3단계 : SQL전송객체  
    rs = stmt.executeQuery(sql); // 4단계(SQL전송) + 5단계(전송결과 받기)
```

Step 1

context.xml(Servers\Tomcat\context.xml)파일에 아래와 같은 코드를 추가

```
<Resource  
    auth="Container"  
    driverClassName = "oracle.jdbc.driver.OracleDriver"  
    url = "jdbc:oracle:thin:@localhost:1521:xe"  
    username = "scott"  
    password = "tiger"  
    name = "jdbc/Oracle11g"  
    type = "javax.sql.DataSource"  
    maxActive = "20"  
    maxWait = "1000"  
/>
```

Step 2

Connection 객체를 가져오는 함수 getConnection()을 아래와 같이 작성

```
public static Connection getConnection() throws Exception{  
    Connection conn = null;  
    try {  
        Context ctx = new InitialContext();  
        DataSource ds =
```

```
(DataSource)ctx.lookup("java:comp/env/jdbc/Oracle11g");
    conn = ds.getConnection();
} catch (NamingException e) {
    System.out.println(e.getMessage());
}
return conn;
}
```