

1 Connection Pooling?

- 2 1)Database와 연결된 connection을 미리 일정 갯수만큼 생성하여 pool속에 저장해 놓고 필요할 때마다 이 pool에 접근하여 Connection 객체를 사용하고, 작업이 끝나면 다시 반환하는 것
- 3 2)사용자가 connection이 필요할 때마다 Connection 객체를 생성하여 연결한다는 것은 매우 비효율적
- 4 3)이 pool을 사용하면 pool 속에 미리 connection이 생성되어 있기 때문에 connection을 생성하는데 드는 시간이 소비되지 않는다.
- 5 4)재 사용이 가능하기 때문에 사용자가 접속할 때마다 계속해서 connection을 생성할 필요가 없다.
- 6 5)Program의 효율과 성능 개선의 효과

8 6)JDBC 방식

```
9 Class.forName("com.mysql.jdbc.Driver");  
10 Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root",  
11 "1234");
```

12 7)Connection Pool 방식

```
13 Class.forName("org.apache.commons.dbcp.PoolingDriver");  
14 Connection conn = DriverManager.getConnection("jdbc:apache:commons:dbcp:/pool");
```

16 8)Connection Pool의 대표적인 Open Source에는 DBCP와 C3P0가 있다.

19 DBCP2 2.6 API for JDBC 4.1 의 사용방법

20 -refer to : <https://hsp1116.tistory.com/8>
21 <https://sjh836.tistory.com/148>

23 1. DBCP 관련 Jar file 및 JDBC driver Jar file 설치하기

24 1)Homepage

25 -DBCP : <http://commons.apache.org/proper/commons-dbcp/>
26 -Pool : <http://commons.apache.org/proper/commons-pool/>
27 -Logging : <http://commons.apache.org/proper/commons-logging/>

29 2)Downloads

30 -DBCP API 관련 jar file : commons-dbcp2-2.6.0-bin.zip or commons-dbcp2-2.6.0-bin.tar.gz
31 -Pool API 의 jar file : commons-pool2-2.6.2-bin.zip or commons-pool2-2.6.2-bin.tar.gz
32 -Logging API의 jar file : commons-logging-1.2-bin.zip or commons-logging-1.2-bin.tar.gz

34 3)위의 file의 압축을 풀고 각각의 jar file을 WEB-INF/lib folder에 import 한다.

35 -commons-dbcp2-2.6.0.jar, commons-pool2-2.6.2.jar, commons-logging-1.2.jar

38 2. Connection Pool 관련 설정 file 및 Connection Pool 관련 driver loading하기

39 -src/com.example.utils.DBCPInit.java

```
40 package com.example.utils;
```

```
42 import java.sql.DriverManager;
```

```
44 import javax.servlet.ServletException;
```

```
45 import javax.servlet.http.HttpServlet;
```

```
47 import org.apache.commons.dbcp2.ConnectionFactory;
```

```
48 import org.apache.commons.dbcp2.DriverManagerConnectionFactory;
```

```
49 import org.apache.commons.dbcp2.PoolableConnection;
```

```
50 import org.apache.commons.dbcp2.PoolableConnectionFactory;
```

```
51 import org.apache.commons.dbcp2.PoolingDriver;
```

```
52 import org.apache.commons.pool2.impl.GenericObjectPool;
53 import org.apache.commons.pool2.impl.GenericObjectPoolConfig;
54
55 public class DBCPInit extends HttpServlet {
56     private final String driverClass = "oracle.jdbc.driver.OracleDriver";
57     private final String url = "jdbc:oracle:thin:@localhost:1521:XE";
58     private final String username = "hr";
59     private final String password = "hr";
60
61     @Override
62     public void init() throws ServletException{
63         loadJDBCdriver();
64         initconnectionPool();
65     }
66
67     private void loadJDBCdriver() {
68         try {
69             //Connection Pool에서 사용할 JDBC Driver Loading
70             Class.forName(this.driverClass);
71         }catch(ClassNotFoundException ex) {
72             throw new RuntimeException("Driver Loading Failure");
73         }
74     }
75
76     private void initconnectionPool() {
77         try {
78             //ConnectionFactory 생성, ConnectionFactory는 새로운 Connection을 생성할 때 사용.
79             ConnectionFactory connFactory = new DriverManagerConnectionFactory(this.url,
80                                     this.username, this.password);
81
82             //DBCP가 Connection Pool에 connection을 보관할 때 사용하는 PoolableConnectionFactory 생성
83             //실제로 내부적으로 connection을 담고 있고, connection을 관리하는데 기능을 제공한다.
84             //Connection을 close하면 종료하지 않고 Connection Pool에 반환한다.
85             PoolableConnectionFactory poolableConnFactory = new
86                 PoolableConnectionFactory(connFactory, null);
87             //Connection이 유효한지 확인할 때 사용하는 query를 설정한다.
88             poolableConnFactory.setValidationQuery("select 1 from dual");
89
90             //Connection Pool의 설정 정보를 생성한다.
91             GenericObjectPoolConfig poolConfig = new GenericObjectPoolConfig();
92             //유휴 connection 검사 주기
93             poolConfig.setTimeBetweenEvictionRunsMillis(1000L * 60 * 1L);
94             //Pool에 있는 connection이 유효한지 검사 유무 설정
95             poolConfig.setTestWhileIdle(true);
96             //Connection 최소 갯수 설정
97             poolConfig.setMinIdle(4);
98             //Connection 최대 갯수 설정
99             poolConfig.setMaxTotal(50);
100
101             //Connection Pool 생성, parameter는 위에서 생성한 PoolableConnectionFactory와
102             GenericObjectPoolConfig를 사용
103             GenericObjectPool<PoolableConnection> connectionPool = new
104                 GenericObjectPool<>(poolableConnFactory, poolConfig);
```

```

102 //PoolableConnectionFactory에도 Connection Pool 연결
103 poolableConnFactory.setPool(connectionPool);
104
105 //Connection Pool을 제공하는 JDBC Driver 등록.
106 Class.forName("org.apache.commons.dbcp2.PoolingDriver");
107
108 PoolingDriver driver = (PoolingDriver)
109 DriverManager.getDriver("jdbc:apache:commons:dbcp:");
110
111 //위에서 Connection Pool Driver에 생성한 Connection Pool을 등록한다.
112 //이름은 cp이다.
113 driver.registerPool("cp", connectionPool);
114 }catch(Exception ex) {
115     throw new RuntimeException(ex);
116 }
117 }
118
119

```

120 3. Web Application이 시작될 때 DBCPInit Servlet class가 시작될 수 있도록 지정하기

```

121 -WEB-INF/web.xml
122 <?xml version="1.0" encoding="UTF-8"?>
123 ....
124 ....
125 <servlet>
126     <servlet-name>DBCPInit</servlet-name>
127     <servlet-class>com.example.utils.DBCPInit</servlet-class>
128     <load-on-startup>1</load-on-startup>
129 </servlet>
130 </web-app>
131
132

```

133 4. Connection을 가져오는 class

134 -Connection을 구하는 class는 별도의 DBConnection class를 작성하는것이 개발하는데 편리하다.

```

135 -src/com.example.utils.DBConnection.java
136
137 package com.example.utils;
138
139 import java.sql.Connection;
140 import java.sql.DriverManager;
141 import java.sql.SQLException;
142
143 public class DBConnection {
144     public static Connection getConnection() throws SQLException {
145         return DriverManager.getConnection("jdbc:apache:commons:dbcp:cp");
146     }
147 }
148
149

```

150 5. 사용방법

151 -Connection을 구하는 곳에 다음과 같이 해주면 된다.

```

152 try {
153     conn = DBConnection.getConnection();
154

```

155

156 6. DBCP Configuration 정보 :

<https://commons.apache.org/proper/commons-dbcp/configuration.html>

157 -Refer to : <https://sjh836.tistory.com/148>