

DBCP

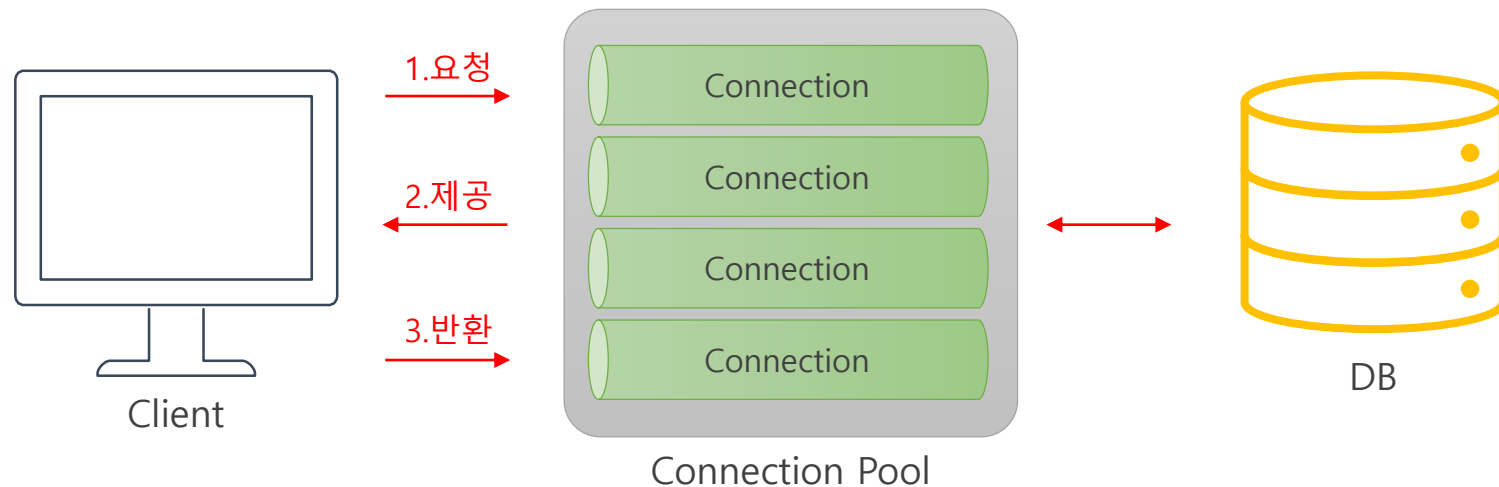
DBCP

- DBCP

- ✓ DataBase Connection Pool
- ✓ DB에 접근할 때는 반드시 Connection 객체가 필요함
- ✓ DB에 접근할 때마다 Connection 객체를 새로 만들면 시스템의 속도가 떨어짐
- ✓ 미리 여러 개의 Connection 객체를 만들어 두고 DB에 접근할 때 사용하는 방식을 '커넥션 풀(Connection Pool)'이라고 함

- DBCP 동작 방식

- ① 미리 Connection 객체를 생성해 둠
- ② Connection 요청이 들어오면 Connection Pool에 있는 커넥션을 하나 대여해 줌
- ③ Connection 사용이 끝나면 Connection을 다시 회수함

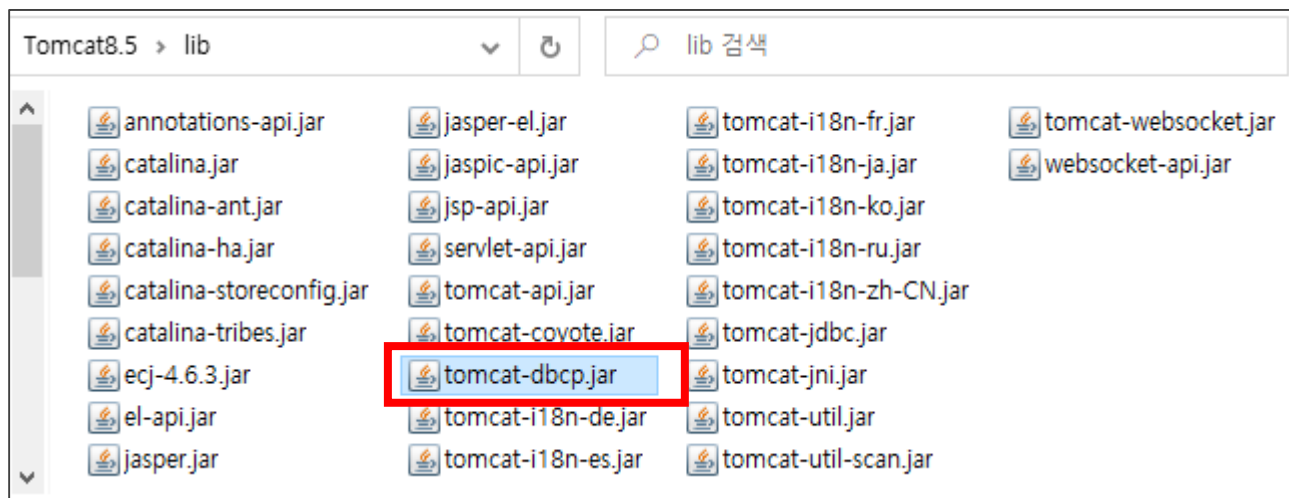


DBCP 사용을 위한 준비-1

- DBCP 라이브러리

- ✓ DBCP : 자카르타(Jakarta) DBCP API 관련 jar
- ✓ Pool : 자카르타(Jakarta) DBCP API가 사용하는 Pool API 관련 jar
- ✓ Collections : 자카르타(Jakarta) Pool API가 사용하는 Collection API 관련 jar

다만, Tomcat에 모든 라이브러리가 포함되어 있으므로 Tomcat을 사용하는 경우 준비할 필요는 없음
%CATALINA_HOME%\lib\tomcat-dbc.jar



DBCP 사용을 위한 준비-2

- context.xml
 - ✓ Connection Pool 설정에 필요한 정보를 context.xml에 작성해야 함
 - ✓ Web Root/META-INF/context.xml 파일 생성
 - ✓ <Resource> 태그 작성
- <Resource> 태그 주요 속성

| 속성명 | 의미 |
|-----------------|---|
| auth | 자원 관리 주체(Container/Application) |
| name | JNDI 이름 (Context의 lookup()으로 검색) |
| type | Resource 타입 |
| driverClassName | JDBC 드라이버 이름 |
| url | 데이터베이스 커넥션 URL |
| username | 데이터베이스 사용자 |
| password | 데이터베이스 사용자 암호 |
| maxTotal | DataSource로부터 꺼낼 수 있는 최대 Connection(기본 8개) |
| maxIdle | DataSource가 유지하는 최대 유휴 Connection(기본 8개) |
| maxWait | 남은 Connection이 없을 때 Connection 반납을 기다리는 최대 밀리초(기본 -1 : 반납할때까지 계속 기다림) |

```
<Resource
  auth="Container"
  name="jdbc/oracle11g"
  type="javax.sql.DataSource"
  driverClassName="oracle.jdbc.OracleDriver"
  url="jdbc:oracle:thin:@localhost:1521:xe"
  username="SCOTT"
  password="1111"
  maxTotal="8"
  maxIdle="2"
  maxWait="5000" />
```

context.xml 예시

JNDI

- JNDI
 - ✓ JNDI, Java Naming and Directory Interface API
 - ✓ 서버의 특정 Resource를 찾는 방식
 - ✓ Java EE 서버에서 Resource를 찾는 경우 정해진 기본 이름이 있음
- 주요 JNDI

| JNDI 이름 | 의미 |
|--------------------|-----------------|
| java:comp/env | 응용프로그램 환경 |
| java:comp/env/jdbc | JDBC DataSource |
| java:comp/ejb | EJB 컴포넌트 |
| java:comp/env/mail | JavaMail 객체 |
| java:comp/env/url | URL 정보 |
| java:comp/env/jms | JMS 연결 객체 |

DBCP 설정이 저장된
Resource를 찾는 경우
사용해야 하는 JNDI

* DataSource : 커넥션 풀 관리 객체

JNDI 예시

- DataSource 객체 생성 시 context.xml에 작성한 <Resource> 태그의 name 속성이 필요함

* DataSource 객체 생성

```
private static DataSource dataSource;  
static {  
    try {  
        Context context = new InitialContext();  
        dataSource = (DataSource)context.lookup("java:comp/env/jdbc/oracle11g");  
    } catch (NamingException e) {  
        System.out.println("Resource name을 찾을 수 없습니다.");  
    }  
}
```

* context.xml

```
<Resource  
    auth="Container"  
    name="jdbc/oracle11g"  
    type="javax.sql.DataSource"  
    driverClassName="oracle.jdbc.OracleDriver"  
    url="jdbc:oracle:thin:@localhost:1521:xe"  
    username="SCOTT"  
    password="1111"  
    maxTotal="8"  
    maxIdle="2"  
    maxWait="5000" />
```