

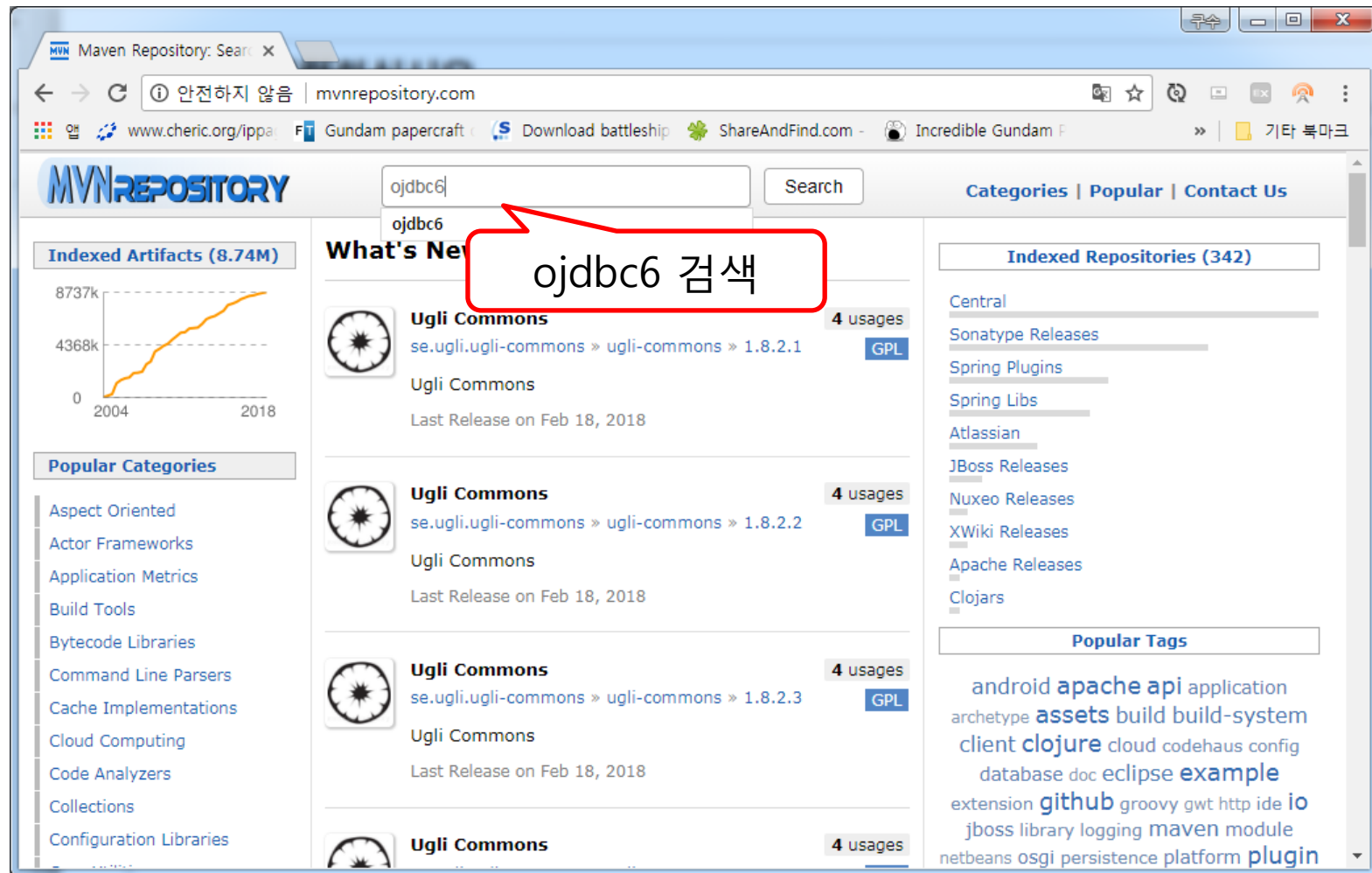
JDBC 프로그래밍

- 커넥션 : 데이터베이스 연결 -

JDBC 프로젝트 설정

❖ 외부 라이브러리 검색 (ojdbc6)

○ <http://mvnrepository.com/>



JDBC 프로젝트 설정

❖ 외부 라이브러리 검색 (ojdbc6)

The screenshot shows a web browser window with the URL `mvnrepository.com/search?q=ojdbc6`. The page displays the Maven Repository logo and a search bar containing 'ojdbc6'. On the left, there is a section titled 'Indexed Artifacts (8.74M)' with a line graph showing growth from 2004 to 2018, and a 'Popular Categories' list including Aspect Oriented, Actor Frameworks, Application Metrics, Build Tools, Bytecode Libraries, and Command Line Parsers. The main content area, titled 'Found 8 results', shows two results. The first result is '1. Ojdbc6' by 'com.oracle', with 219 usages and a last release on Apr 26, 2016. The second result is '2. Oracle JDBC Driver' by 'oracle', with 4 usages and a last release on Jan 20, 2014. This second result is highlighted with a red rectangular box.

Maven Repository: ojdbc x

mvnrepository.com/search?q=ojdbc6

www.cheric.org/ippa Gundam papercraft Download battleship ShareAndFind.com Incredi

MVNREPOSITORY ojdbc6 Search

Indexed Artifacts (8.74M)

8737k
4368k
0
2004 2018

Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers

Found 8 results

Sort: relevance | popular | newest

1. Ojdbc6 219 usages
com.oracle » ojdbc6
Ojdbc6
Last Release on Apr 26, 2016

2. Oracle JDBC Driver 4 usages
oracle » ojdbc6
Oracle JDBC driver classes for use with JDK1.4
Last Release on Jan 20, 2014

JDBC 프로젝트 설정

❖ 외부 라이브러리 검색 (ojdbc6)

The screenshot shows the Maven Repository website for the Oracle JDBC Driver. The browser address bar displays `mvnrepository.com/artifact/oracle/ojdbc6`. The page features a search bar at the top and a sidebar with navigation links. The main content area displays the Oracle JDBC Driver information, including a graph of indexed artifacts and a table of versions.

Indexed Artifacts (8.74M)

Graph showing indexed artifacts from 2004 to 2018. The y-axis ranges from 0 to 8737k, and the x-axis shows the years 2004 and 2018.

Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries

Oracle JDBC Driver

Oracle JDBC driver classes for use with JDK1.4

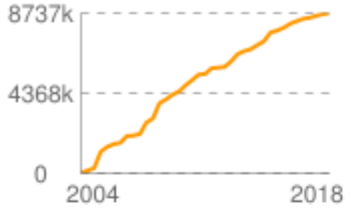
Used By 4 artifacts

Datanucleus (1)

Version	Repository	Usages	Date
11.2.0.3	Datanucleus	2	(Jan, 2014)


JDBC 프로젝트 설정

❖ 외부 라이브러리 검색 (ojdbc6)

Indexed Artifacts (8.74M)**Popular Categories**

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection

Home » oracle » ojdbc6 » 11.2.0.3

**Oracle JDBC Driver » 11.2.0.3**

Oracle JDBC driver classes for use with JDK1.4

Organization	Oracle Corporation
HomePage	http://www.oracle.com/technology/software/tech/java/sqlj_jdb...
Date	(Jan 20, 2014)
Files	pom (895 bytes) jar (2.6 MB) View All
Repositories	Datanucleus
Used By	4 artifacts

[Maven](#) [Gradle](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#) [Buildr](#)

```
<!-- https://mvnrepository.com/artifact/oracle/ojdbc6 -->
<dependency>
  <groupId>oracle</groupId>
  <artifactId>ojdbc6</artifactId>
  <version>11.2.0.3</version>
</dependency>
```

☒ Include comment with link to declaration

Note: this artifact is located at [Datanucleus](http://www.datanucleus.org/downloads/maven2/) repository
(<http://www.datanucleus.org/downloads/maven2/>)

Pom.xml의
<dependencies>에 등록

pom.xml의
<repositories>에 등록

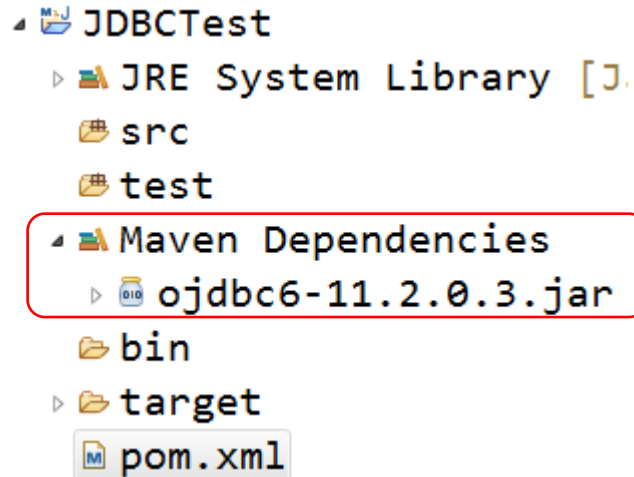
JDBC 프로젝트 설정

❖ pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" ...>
  <modelVersion>4.0.0</modelVersion>
  <groupId>edu.iot</groupId>
  <artifactId>JDBCTest</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <repositories>
    <repository>
      <id>ojdbc</id>
      <url>http://www.datanucleus.org/downloads/maven2/</url>
    </repository>
  </repositories>
  <dependencies>
    <dependency>
      <groupId>oracle</groupId>
      <artifactId>ojdbc6</artifactId>
      <version>11.2.0.3</version>
    </dependency>
  </dependencies>
  :
</project>
```

JDBC 프로젝트 설정

❖ pom.xml



다운로드 받은
oracle jdbc 라이브러리

- Maven 로컬 레파지토리
 - C:\Users\<사용자ID>\.m2\repository
 - 다시 다운로드 받고 싶은 경우 해당 라이브러리를 제거 후 pom.xml 을 다시 저장

JDBC 프로그래밍

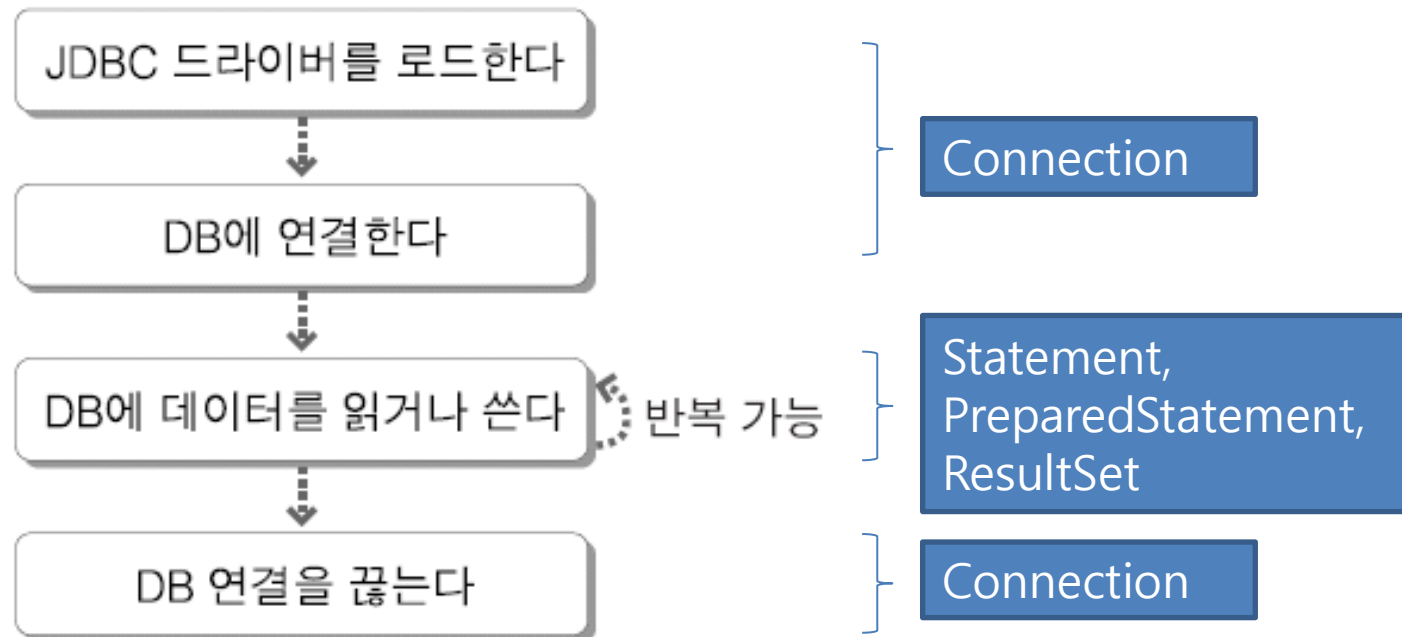
❖ JDBC 주요 클래스

- Connection
 - JDBC 드라이버 로드 및 DBMS 접속 및 해제
- Statement
 - SQL문 실행
- PreparedStatement
 - 매개변수가 있는 SQL문 실행
- ResultSet
 - SELECT 쿼리의 결과 행 집합 관리

JDBC 프로그래밍

❖ JDBC를 이용한 프로그램의 작성 방법

- 자바 프로그램에서 데이터베이스를 사용하는 4단계의 과정



Connection 관리

❖ JDBC 드라이버의 로드

- JDBC 드라이버를 로드하는 방법
 - 오라클 JDBC 드라이버의 클래스 이름을 지정

```
try {  
    Class.forName("oracle.jdbc.driver.OracleDriver");  
    System.out.println("오라클 JDBC 드라이버 로드 성공");  
} catch (ClassNotFoundException e) {  
    System.err.println("드라이버 로드 실패" + e.getMessage());  
}
```

Connection 관리

❖ JDBC 드라이버를 로드 프로그램

```
public class JDBCEx1 {  
    public static void main(String[] args) {  
        try {  
            Class.forName("oracle.jdbc.driver.OracleDriver");  
            System.out.println("오라클 JDBC 드라이버 로드 성공");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Connection 관리

❖ 데이터베이스로의 연결(사용자 로그인)

- 데이터베이스 연결 객체 생성

- Connection 객체 - DriverManager.getConnection() 메서드의 리턴값

```
String url = "데이터베이스 접속 url";
```

```
Connection conn = (Connection) DriverManager.getConnection(  
    url, "db 사용자명", "패스워드");
```

```
// 접속 url : "jdbc드라이이버@db_서버_ip:포트번호:SID"
```

```
String url = "jdbc:oracle:thin:@localhost:1521:XE";
```

```
Connection conn = (Connection) DriverManager.getConnection(  
    url, "hr", "hr");
```

Connection 관리

❖ 데이터베이스 연결 끊기

- 데이터베이스와 연결을 끊는 방법

`conn.close();`

↑
데이터베이스로의
연결을 끊는 메소드

- 예외처리 필수

Connection 관리

❖ JDBC 드라이버를 로드하고 데이터베이스에 연결하는 프로그램

```
public class JDBCEx2 {  
    public static void main(String[] args) {  
  
        String url = "jdbc:oracle:thin:@localhost:1521:XE";  
        try(Connection conn =  
            (Connection) DriverManager.getConnection(  
                url, "hr", "hr")) {  
            System.out.println("데이터베이스에 접속했습니다.");  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Connection 관리

❖ 접속 정보의 분리

- Source 폴더 추가
 - src/main/java
 - database.properties 파일 생성(문자셋 : UTF-8)

```
driver=oracle.jdbc.driver.OracleDriver  
url=jdbc:oracle:thin:@localhost:1521:xe  
username=hr  
password=hr
```

Connection 관리

❖ Properties 파일 읽기

```
public class JDBCEx3 {  
    public static void main(String[] args) {  
  
        try {  
            Properties properties = new Properties();  
            String path = JDBCEx3.class.getResource(  
                "/database.properties").getPath();  
            properties.load(new FileReader(path));  
            String driver = properties.getProperty("driver");  
            String url = properties.getProperty("url");  
            String username = properties.getProperty("username");  
            String password = properties.getProperty("password");  
            System.out.println(driver);  
  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```


Connection 관리

❖ database.properties 정보로 데이터베이스 연결 클래스

```
public class Database {  
    private static Connection connection;  
    static {  
        try {  
            Properties properties = new Properties();  
            String path = Database.class.getResource(  
                "/database.properties").getPath();  
            properties.load(new FileReader(path));  
            String driver = properties.getProperty("driver");  
            String url = properties.getProperty("url");  
            String username = properties.getProperty("username");  
            String password = properties.getProperty("password");  
  
            Class.forName(driver);  
            connection = (Connection) DriverManager.getConnection(  
                url, username, password);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Connection 관리

❖ database.properties 정보로 데이터베이스 연결 클래스

```
public static Connection getConnection() {  
    return connection;  
}  
  
public static void close() {  
    if(connection!= null) {  
        try {  
            connection.close();  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }  
}  
}
```

Connection 관리

❖ 데이터베이스 연결

```
public class JDBCEx4 {  
    public static void main(String[] args) {  
  
        Connection conn = Database.getConnection();  
  
        System.out.println("데이터베이스 준비 완료");  
  
        Database.close();  
    }  
}
```