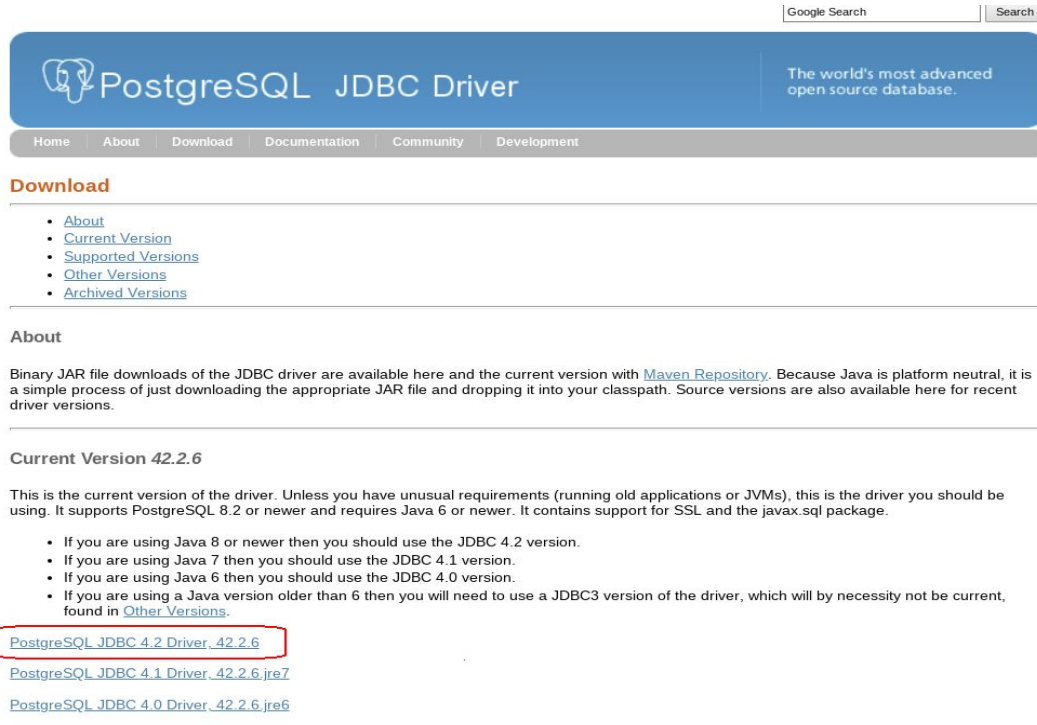


POSTGRESQL JDBC CONNECTION EXAMPLE

1. Visit <http://jdbc.postgresql.org/download.html> to download the latest PostgreSQL JDBC Driver.



The screenshot shows the PostgreSQL JDBC Driver download page. At the top, there is a blue header with the PostgreSQL logo and the text "PostgreSQL JDBC Driver". To the right of the header is a search bar with "Google Search" and "Search" buttons. Below the header is a navigation bar with links: Home, About, Download, Documentation, Community, and Development. The main content area is titled "Download" and contains a list of links: About, Current Version, Supported Versions, Other Versions, and Archived Versions. Below this is an "About" section with text about binary JAR file downloads and a link to the Maven Repository. The "Current Version 42.2.6" section follows, stating that this is the current version and providing instructions for different Java versions. A red box highlights the link "PostgreSQL JDBC 4.2 Driver, 42.2.6". Below this are links for "PostgreSQL JDBC 4.1 Driver, 42.2.6.jre7" and "PostgreSQL JDBC 4.0 Driver, 42.2.6.jre6".

Google Search Search

PostgreSQL JDBC Driver

The world's most advanced open source database.

Home About Download Documentation Community Development

Download

- [About](#)
- [Current Version](#)
- [Supported Versions](#)
- [Other Versions](#)
- [Archived Versions](#)

About

Binary JAR file downloads of the JDBC driver are available here and the current version with [Maven Repository](#). Because Java is platform neutral, it is a simple process of just downloading the appropriate JAR file and dropping it into your classpath. Source versions are also available here for recent driver versions.

Current Version 42.2.6

This is the current version of the driver. Unless you have unusual requirements (running old applications or JVMs), this is the driver you should be using. It supports PostgreSQL 8.2 or newer and requires Java 6 or newer. It contains support for SSL and the javax.sql package.

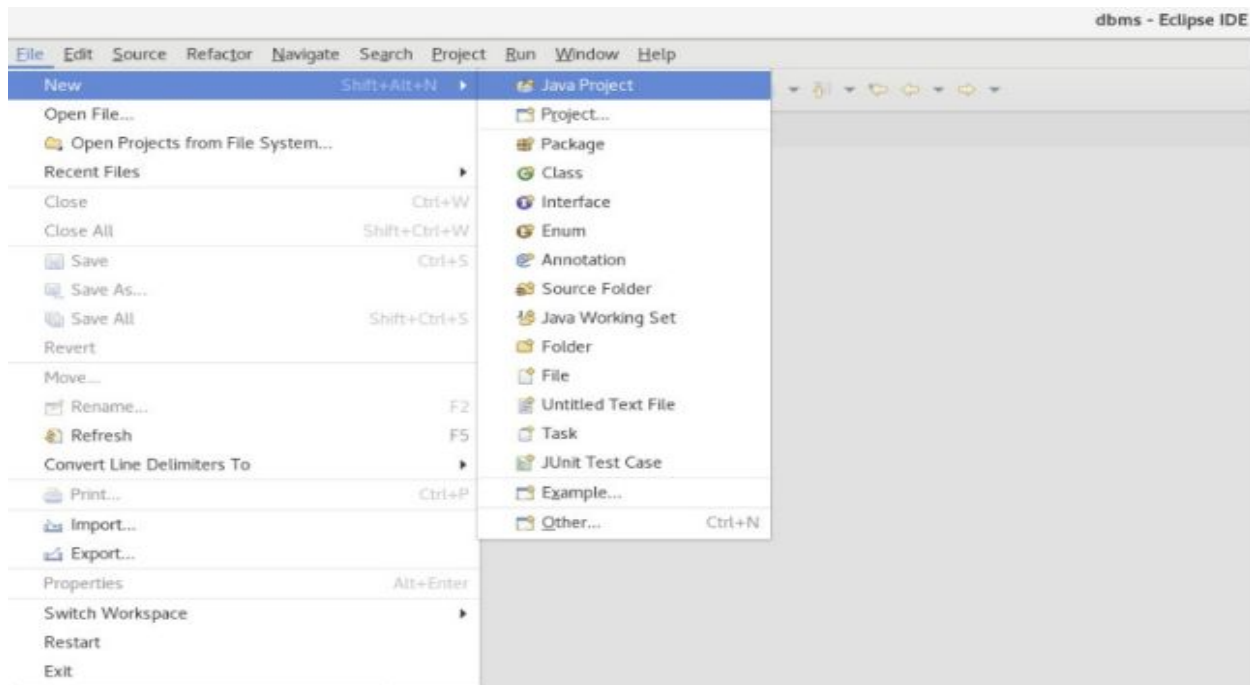
- If you are using Java 8 or newer then you should use the JDBC 4.2 version.
- If you are using Java 7 then you should use the JDBC 4.1 version.
- If you are using Java 6 then you should use the JDBC 4.0 version.
- If you are using a Java version older than 6 then you will need to use a JDBC3 version of the driver, which will by necessity not be current, found in [Other Versions](#).

[PostgreSQL JDBC 4.2 Driver, 42.2.6](#)

[PostgreSQL JDBC 4.1 Driver, 42.2.6.jre7](#)

[PostgreSQL JDBC 4.0 Driver, 42.2.6.jre6](#)

2. Open Eclipse and click File -> New-> Java project. Give proper project name and click on Finish.



New Java Project



Create a Java Project

Create a Java project in the workspace or in an external location.



Project name:

☒ Use default location

Location:

JRE

☐ Use an execution environment JRE:

☐ Use a project specific JRE:

☒ Use default JRE (currently 'java-8-openjdk-amd64')

[Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files

[Configure default...](#)

Working sets

☐ Add project to working sets

Working sets:



The current workspace uses a 1.4 JRE with compiler compliance level 1.7. This is not recommended and either the JRE or the compiler compliance level should be changed. [Configure...](#)



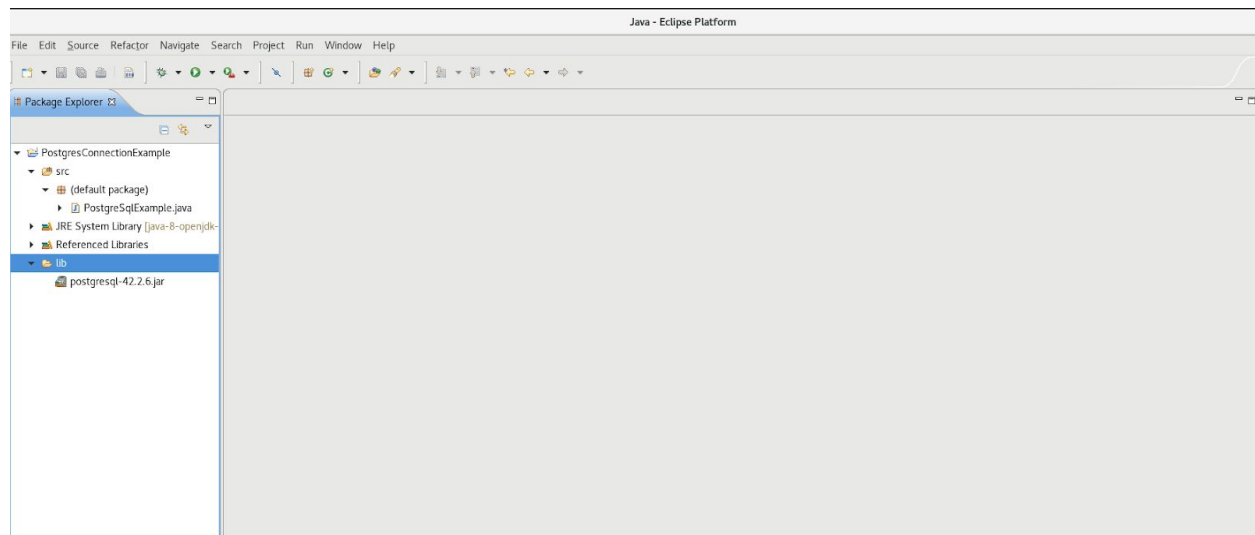
< Back

Next >

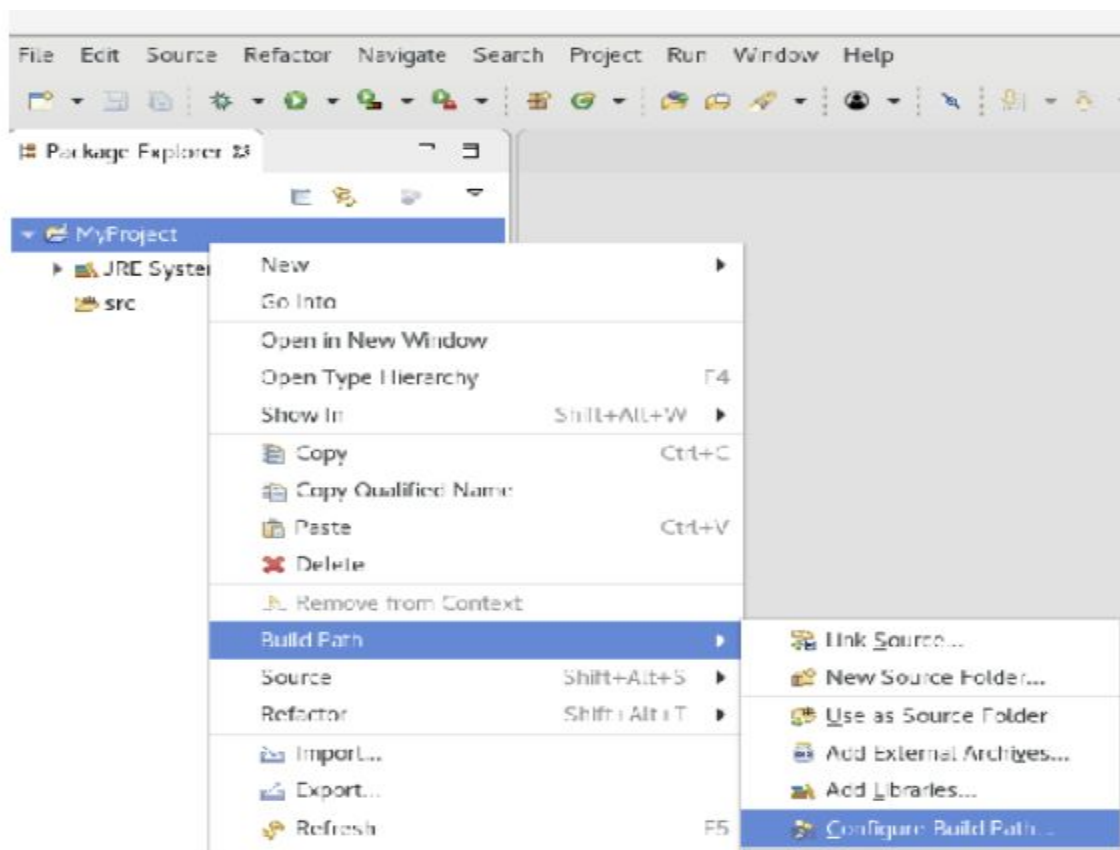
Cancel

Finish

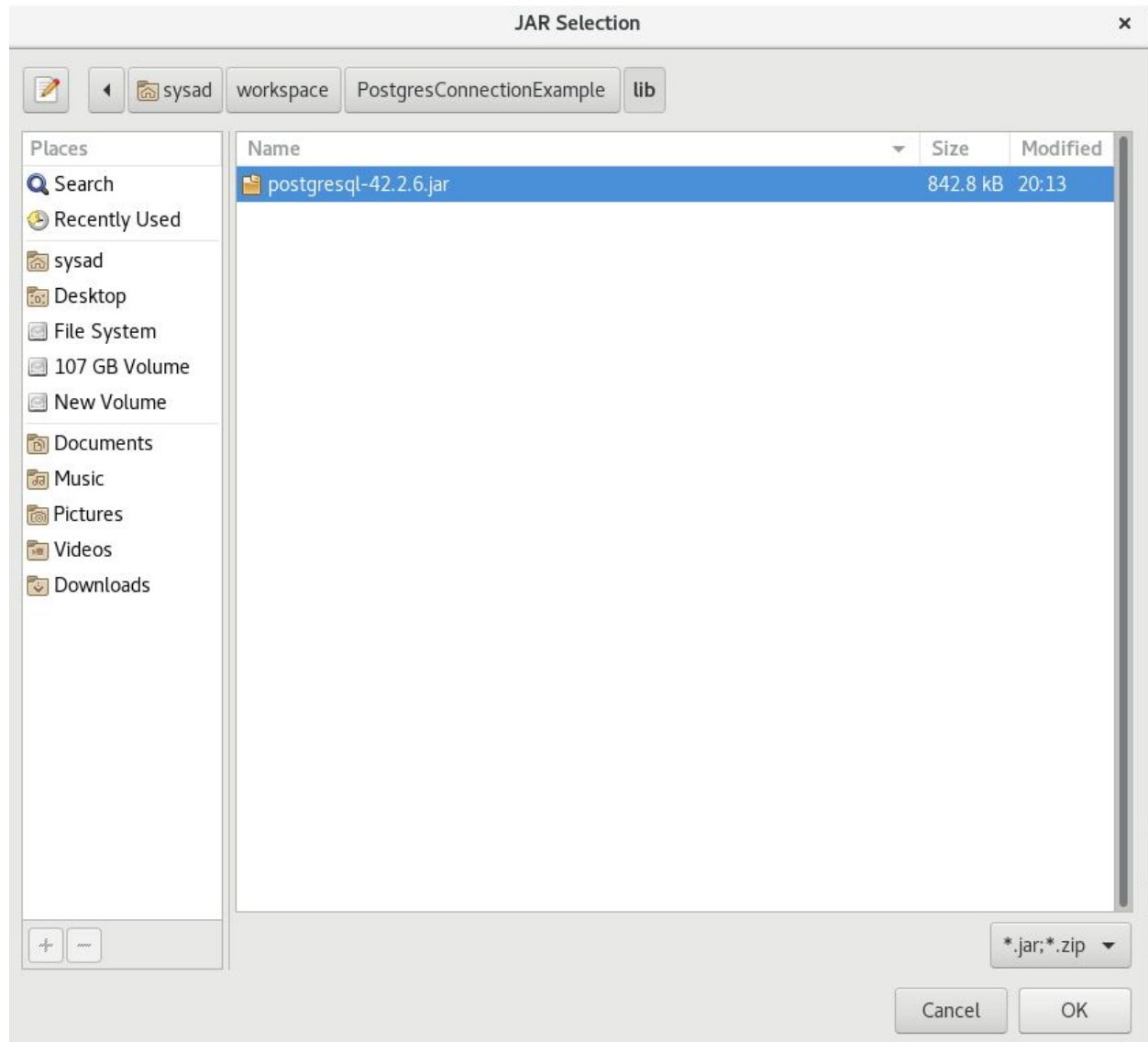
3. Create a folder lib under your project directory and add the postgresql jdbc driver jar file to that.



4. Right click on your project in the Project Explorer, then Build Path and Configure Build Path



5.. Goto the libraries tab --> Add external Jar --> postgresql.jar file. Click apply and close.



6. Right click on your project in the Project Explorer, then File -> Class and create the java class
- PostgreSQLExample.java

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
  
public class PostgreSQLExample {  
    public static void main(String[] args) {
```

```

try (Connection connection =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/university", "postgres",
"123456")) {

    System.out.println("Java JDBC PostgreSQL Example");
    System.out.println("Connected to PostgreSQL database!");
    Statement statement = connection.createStatement();
    System.out.println("Reading student records...");
    System.out.printf("%s %s", "id", "name");
    ResultSet resultSet = statement.executeQuery("SELECT * FROM student");
    while (resultSet.next()) {
        System.out.printf("%s %s\n", resultSet.getString("id"), resultSet.getString("name"));
    }

} catch (SQLException e) {
    System.out.println("Connection failure.");
    e.printStackTrace();
}
}
}

```

7. Run the java code.

Output:

```

<terminated> PostgresSqlExample [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (19-Aug-2019 8:56:56 PM)
Java JDBC PostgreSQL Example
Connected to PostgreSQL database!
Reading student records...
id name24746 Schrefl
79352 Runat
76672 Miliko
14182 Moszkowski
44005 Prieto
96052 Marcol
35175 Quimby
44271 Sowerby
40897 Coppens
92839 Cirsto
79329 Velikovs
97101 Marek
24865 Tran-
36052 Guerra
98940 Hawthorne
21395 Louen
55059 Eguchi
74016 Moei
93061 Alfaro
94908 Krishnakumar
30772 Inoue
1968 Sahn
90567 Tomasos
49611 Karande
81538 Wunderli
60990 Greve
74639 Cerime
66954 Crick
792 Hashimoto
60267 Dage
17769 Pearlman
39580 Macias
89186 Dawson
37856 Gav

```

NOTE :

JDBC database URL for PostgreSQL

The syntax of database URL for PostgreSQL looks like the following forms:

`jdbc:postgresql:database`

`jdbc:postgresql://host/database`

`jdbc:postgresql://host:port/database`

`jdbc:postgresql://host:port/database?param1=value1 & param2=val2& ...`

Where:

- host: hostname or IP address of the machine on which PostgreSQL server is running. If omitted, default is localhost.
- port: port number on which the server is listening. If omitted, default is 5432.
- database: name of the database to connect to.
- param1=value1¶m2=value2&...: specify additional connection parameters in pairs of key=value form. Most common parameters are user and password.

Here are some examples:

- Connect to the database ProductDB on localhost:

`jdbc:postgresql:ProductDB`

- Connect to a remote PostgreSQL server on the host dbserver:

`jdbc:postgresql:dbserver:ProductDB`

- Using host name and port number explicitly:

`jdbc:postgresql:dbserver:5432:ProductDB`

- Specify username and password for the connection:

`jdbc:postgresql:ProductDB&user=root&password=secret`