JDBC API

Table of Contents

[JDBC API 2](#_Toc141737915)

[Packages 2](#_Toc141737916)

[Package java.sql 2](#_Toc141737917)

[Package javax.sql 2](#_Toc141737918)

[Interfaces and their uses 3](#_Toc141737919)

[Classes and their uses 4](#_Toc141737920)

# JDBC API

* JDBC or Java Database Connectivity is a Java API to connect and execute the query with the database.
* It contains a set of classes, interfaces and methods to connect and transact with the database.
* The latest JDBC API version is 4.3.
* The JDBC 4.3 API incorporates all of the previous JDBC API versions:
  1. The JDBC 4.2 API
  2. The JDBC 4.1 API
  3. The JDBC 4.0 API
  4. The JDBC 3.0 API
  5. The JDBC 2.1 core API
  6. The JDBC 2.0 Optional Package API   
     (Note that the JDBC 2.1 core API and the JDBC 2.0 Optional Package API together are referred to as the JDBC 2.0 API.)
  7. The JDBC 1.2 API
  8. The JDBC 1.0 API
* It consists of following:

1. Packages
2. Interfaces
3. Classes

Let us have a brief look at them one by one

## Packages

The JDBC 4.3 API includes both:

1. java.sql package
2. javax.sql package

### Package java.sql

* The java.sql package, is referred to as the JDBC Core API
* The java.sql package is included in the Java Standard Platform Edition (Java SE)

### Package javax.sql

* The javax.sql package, is referred as the JDBC Optional Package API.
* The javax.sql package extends the functionality of the JDBC API from a client-side API to a server-side API, and
* It is an essential part of the Java Enterprise Edition (Java EE) technology.

## Interfaces and their uses

|  |  |  |
| --- | --- | --- |
| **S. No** | **Interface name** | **Description** |
| 1. | Driver | Every database vendor has to provide a Driver class to connect with its database. This Driver class must implement java.sql.Driver interface |
| 2. | Connection | Represents a connection object to connect to the database. |
| 3. | Statement | Used to execute a **static SQL statement** & return the results it produces |
| 4. | PreparedStatement | The PreparedStatement interface is a subinterface of Statement.  It is used to execute parameterized query. |
| 5. | CallableStatement | Used to execute SQL stored procedures.  It is a sub-interface of PreparedStatement. |
| 6. | ResultSet | It is a Java object that contains the results of executing an SQL query. |
| 7. | RowSet | It is an extension of the ResultSet interface. It provides a more flexible and powerful way to work with tabular data retrieved from a database. |
| 8. | ResultSetMetaData | It is used to get the metadata (data about data) of a table like total number of column, column name, column type etc. from the ResultSet object. |
| 9. | DatabaseMetaData | It is used to get meta data of a database such as database product name & version, driver name, name of total number of tables etc. |

## ****Classes and their uses****

|  |  |  |
| --- | --- | --- |
| **S. No** | **Class names** | **Description** |
| 1. | com.mysql.cj.jdbc  .Driver | Not a part of JDBC API. Has to be provided by database vendor.  Used to connect with the database |
| 2. | DriverManager | Used to manage JDBC drivers. |
| 3. | Date | The representation of an SQL Date value in Java |
| 4 | Time | The representation of an SQL Time value in Java |
| 5. | CLOB | CLOB stands for **Character Large Object** in general, an SQL Clob is a built-in datatype and is used to store large amount of textual data. Using this datatype, you can store data up to 2,147,483,647 characters. |
| 5. | BLOB | A BLOB is binary large object that can hold a variable amount of data with a maximum length of **65535** characters.  These are used to store large amounts of binary data, such as images or other types of files. |
| 6. | SQLException | An exception that provides information on a database access error or other errors. It extends Exception class. It is a checked Exception |
| 7. | SQLIntegrityConstraintViolationException | It indicates that an integrity constraint (foreign key, primary key or unique key) has been violated. |
| 8. | MysqlDataTruncation | It indicates that we are trying to store a larger value in a smaller data type column |