Automation Testing

Introduction to JDBC

Learning Objectives

By the end of this lesson, you will be able to:

- Define JDBC
- Illustrate a connection in JDBC
- Classify connections and their types
- Explain the JDBC architecture

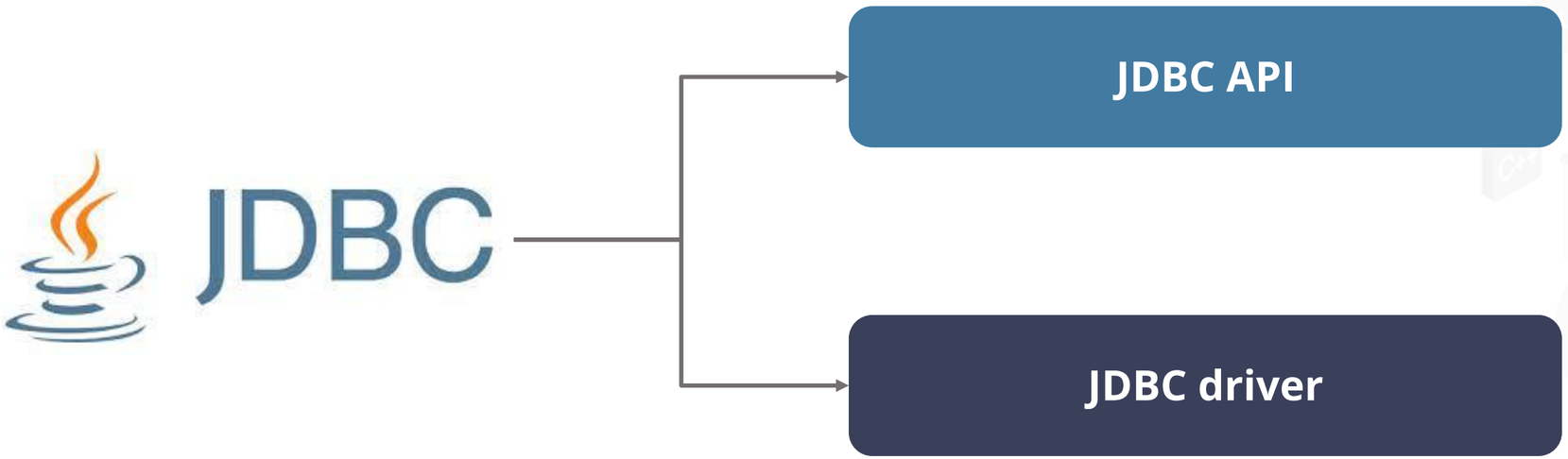
Overview of JDBC

Overview of JDBC

Java Database Connectivity (JDBC) is a database-agnostic, Java-based data access technology that describes how a client can access a database.

JDBC Architecture

The JDBC interface consists of two layers:



JDBC API

A JDBC API provides data access from the Java programming language, and it includes java.sql and javax.sql packages. It is implemented through a JDBC driver.

These are the main JDBC API objects:

- DataSource
- Connection
- Statement
- ResultSet

JDBC API

DataSource

Connection

Statement

ResultSet

It establishes connections.

JDBC API

DataSource

Connection

Statement

ResultSet

It controls connection to the database.

JDBC API

DataSource

Connection

Statement

ResultSet

It executes SQL queries.

JDBC API

DataSource

Connection

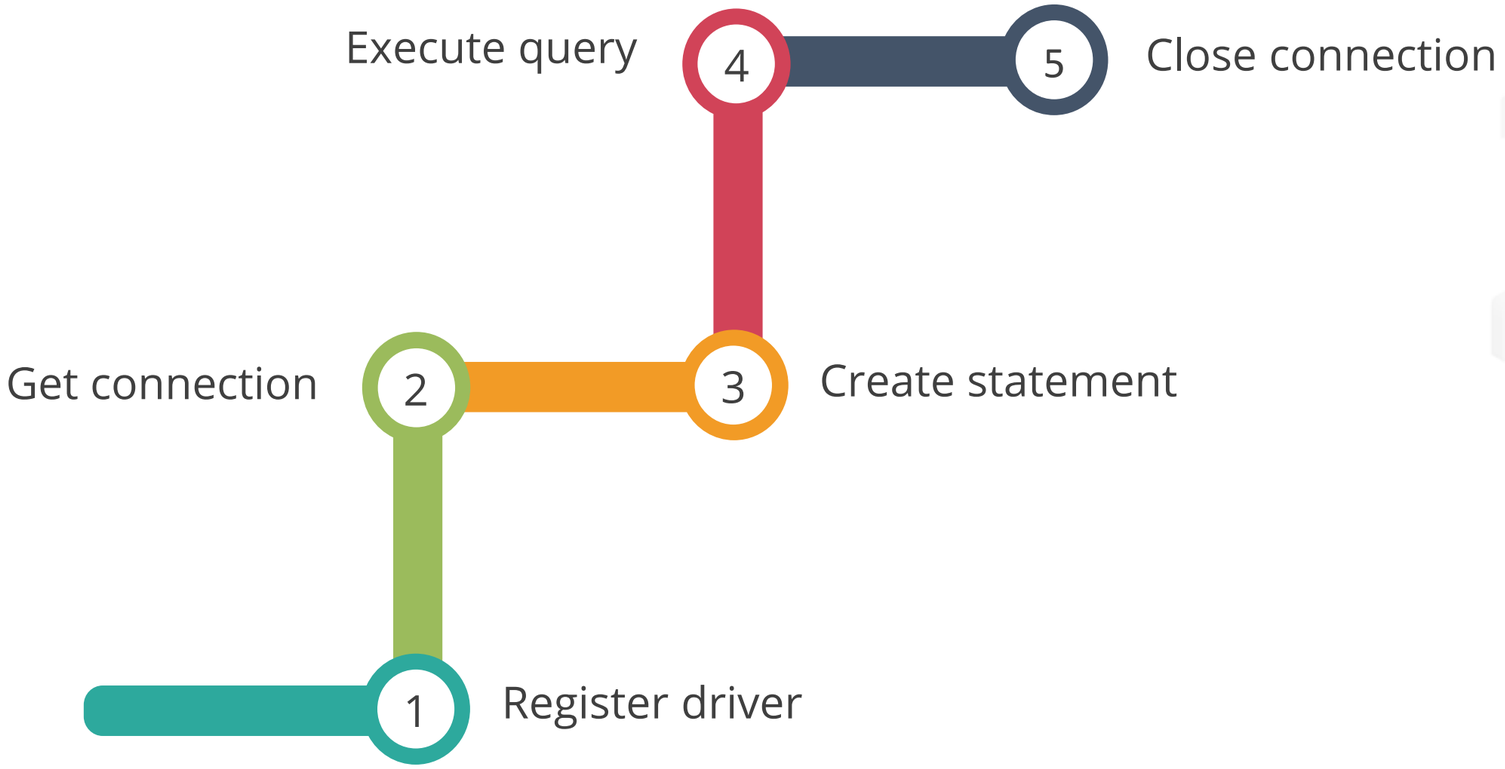
Statement

ResultSet

It holds the result of a query.

JDBC Connectivity

These are the steps involved in JDBC connectivity:

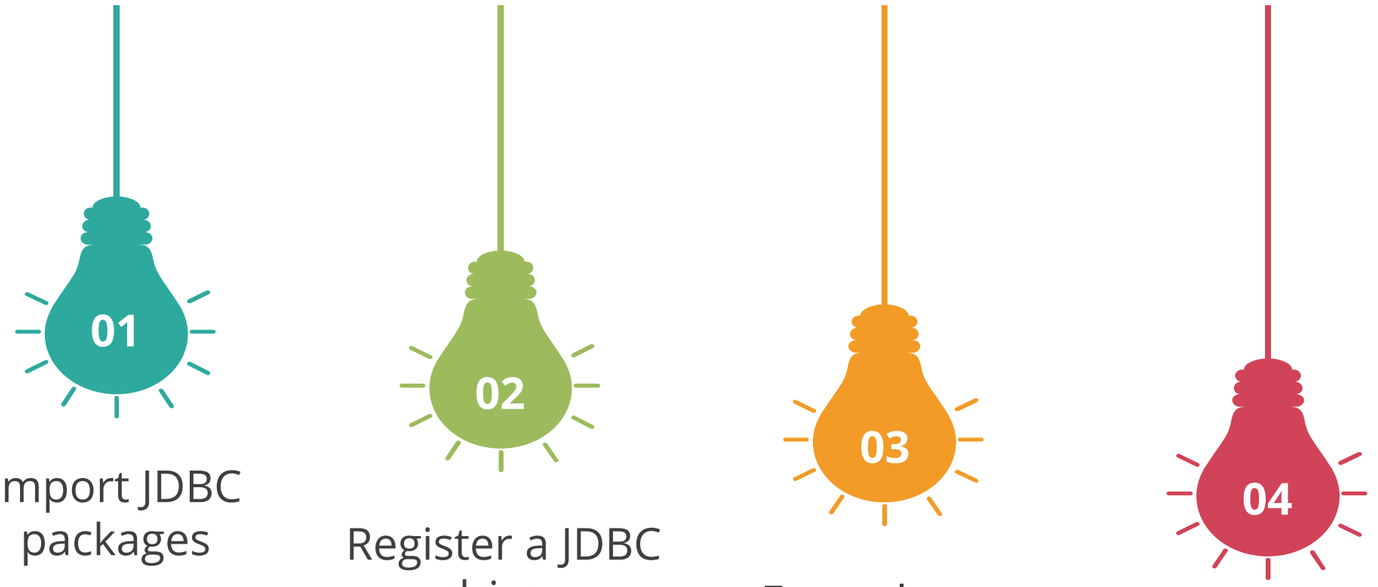


Connections and Their Types

JDBC Connection

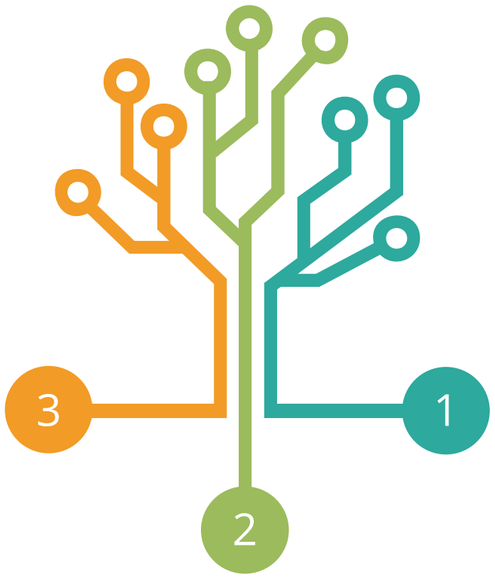
JDBC connection represents the connection to a relational database. This connection is established using an object of connection interface.

These are the steps to establish a JDBC connection:



JDBC Connection

A Connection object is instantiated using the DriverManager.getConnection method. The various overloaded methods of getConnection are listed below:



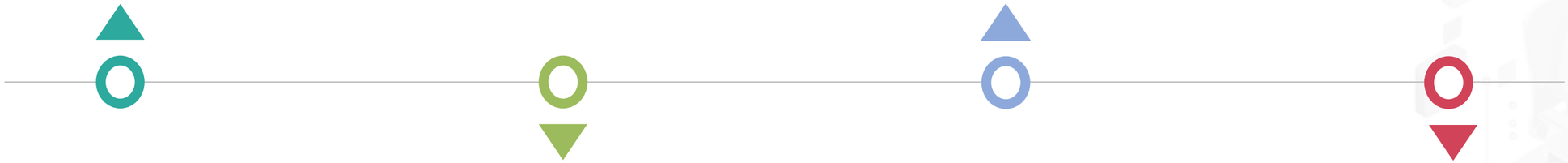
getConnection (String urlString, String username, String password)
h

getConnection (String urlString)

getConnection (String urlString, Properties)

JDBC Driver and Its Types

A JDBC driver is a software component that enables a Java application to interact with a database.
These are the different types of JDBC drivers:



JDBC Driver and Its Types

JDBC-ODBC bridge driver

- The JDBC-ODBC bridge driver converts JDBC method calls to ODBC function calls.
- It is also known as the Universal driver since it can connect to any database.

Native-API driver

- This driver translates JDBC method calls to database native calls.
- It requires a local API to connect with other databases, which is why data transfer is considerably more secure than with a JDBC-ODBC bridge driver.

JDBC Driver and Its Types

Network-protocol driver

- The Network Protocol driver makes use of middleware (application server) to convert JDBC calls into vendor-specific database protocols, either directly or indirectly.
- There is no need for unique client-side installation because all database connectivity drivers are present on a single server.

Thin driver

- This driver communicates with the database directly.
- It is known as Thin driver because it does not require any native database libraries.

JDBC Statement and Its Types

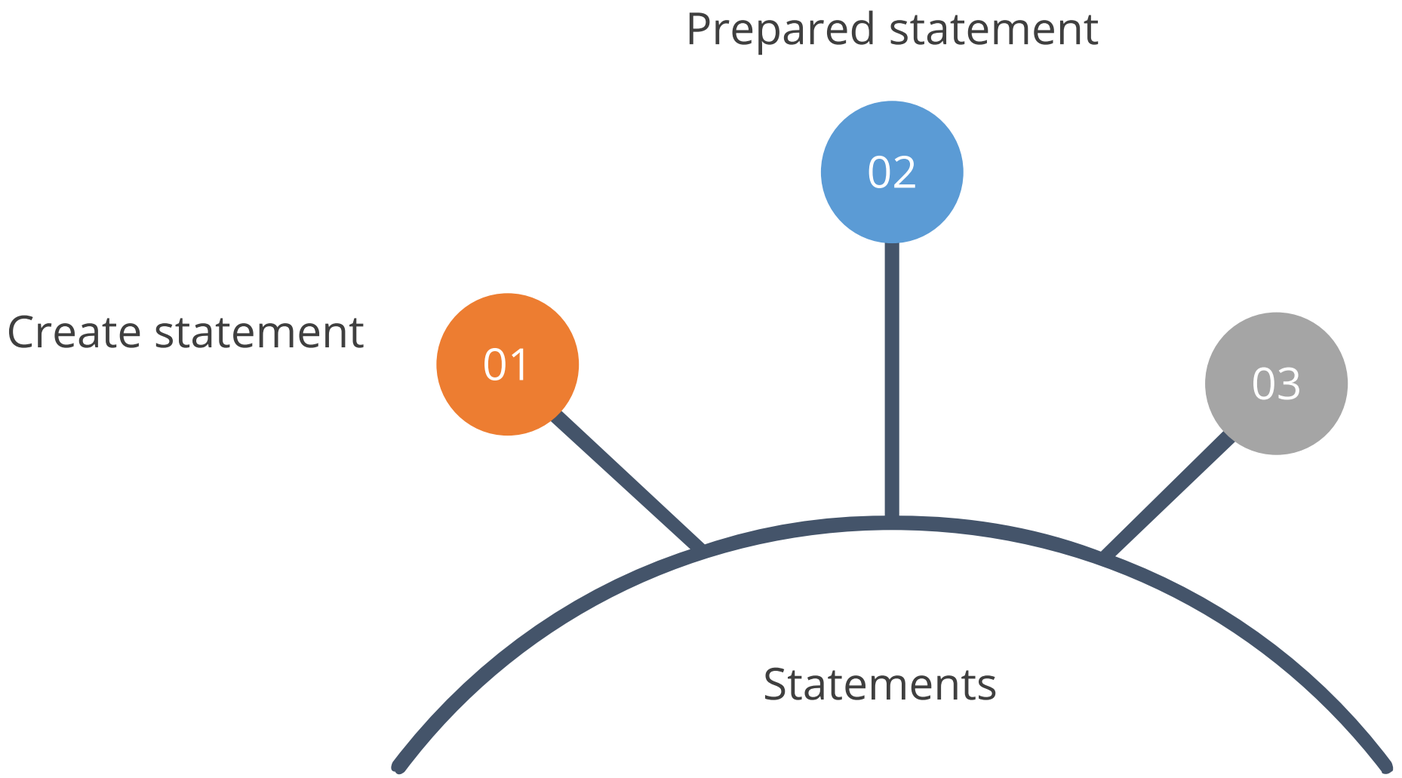
JDBC Statement

A JDBC Statement is used to execute queries. The methods in the Statement, CallableStatement, and PreparedStatement interfaces are used to execute queries against the database.

The statement interfaces are as follows:

Interfaces	Description	Parameters
Statement	Used for general-purpose database access and executing static SQL statements at run-time	Parameters not accepted
CallableStatement	Used to access stored procedures in the database	Input parameters accepted at run-time
PreparedStatement	Used to execute SQL statements repeatedly	Input parameters accepted at run-time

Types of Statements



Types of Statements

- Create Statement
- Prepared Statement
- Callable Statement

- It is typically utilized for general–purpose database access and comes in handy when running static SQL commands
- Users can build the object for this interface via the connection interface.

Types of Statements

Create Statement

Prepared Statement

Callable Statement

- This statement is a compiled SQL statement that can be used several times.
- It accepts parameterized SQL queries.

Types of Statements

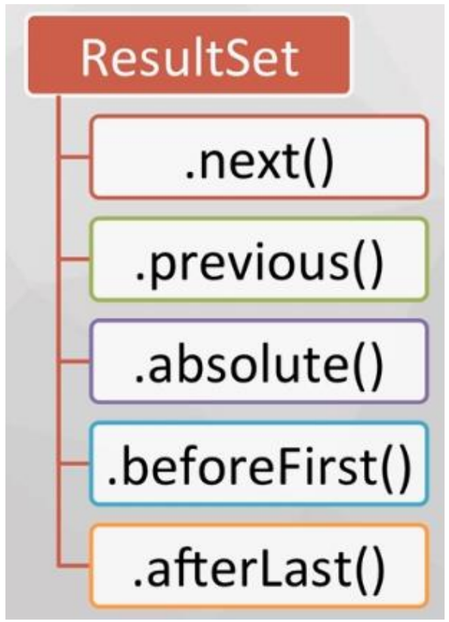
- Create Statement
- Prepared Statement
- Callable Statement

- The JDBC API Callable statement interface aids in the execution of stored procedures.
- When users are dealing with many tables in a complex scenario, instead of sending multiple queries to the database, they can use this feature.

ResultSet and Its Types

JDBC ResultSet

The ResultSet interface represents the result set of a database query. The current row in the result set is indicated by a cursor in a ResultSet object.

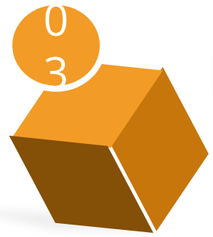


Types of ResultSet

The ResultSet object contains the result of a SQL query. There are three types of ResultSets:

Forward-only

Scroll-insensitive



Scroll-sensitive

Types of ResultSet

ResultSet Type	Description	Sensitivity
Forward-only	Default ResultSet type: In the result set, the cursor can only go forward	-
Scroll-insensitive	The cursor can scroll forward and backward	Not sensitive to database changes that occur after the creation of the result set
Scroll-sensitive	The cursor can scroll forward and backward	Sensitive to database changes that occur after the creation of the result set

Key Takeaways

- JDBC is a database-agnostic, Java-based data access technology that describes how a client can access a database.
- A JDBC driver is a software component that enables a Java application to interact with a database.
- There are three types of ResultSet: Forward-only, Scroll-insensitive, and Scroll-sensitive.
- The statement interface is used to provide methods to execute queries over the database.