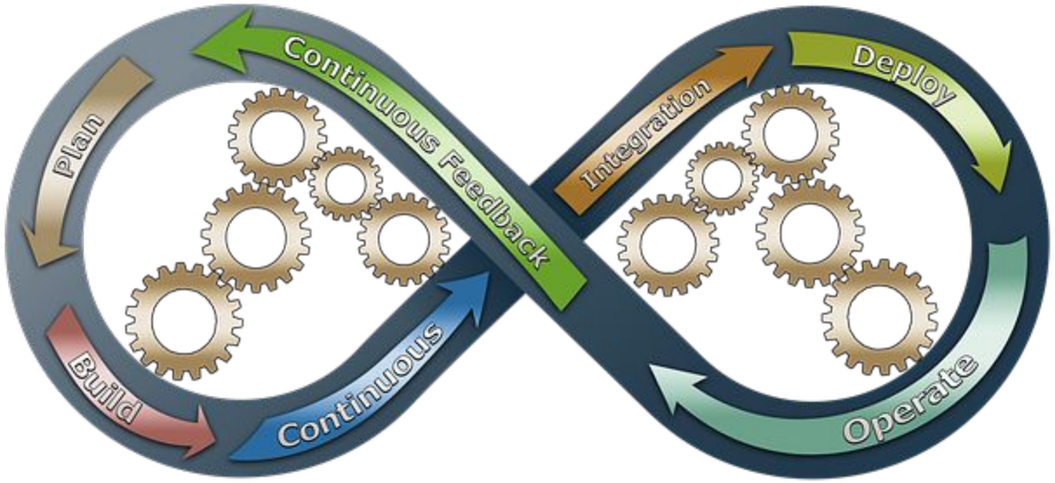


Session 3.10

Database Connection (JDBC)

AN INITIATIVE BY

UNICAL ACADEMY

Introduction

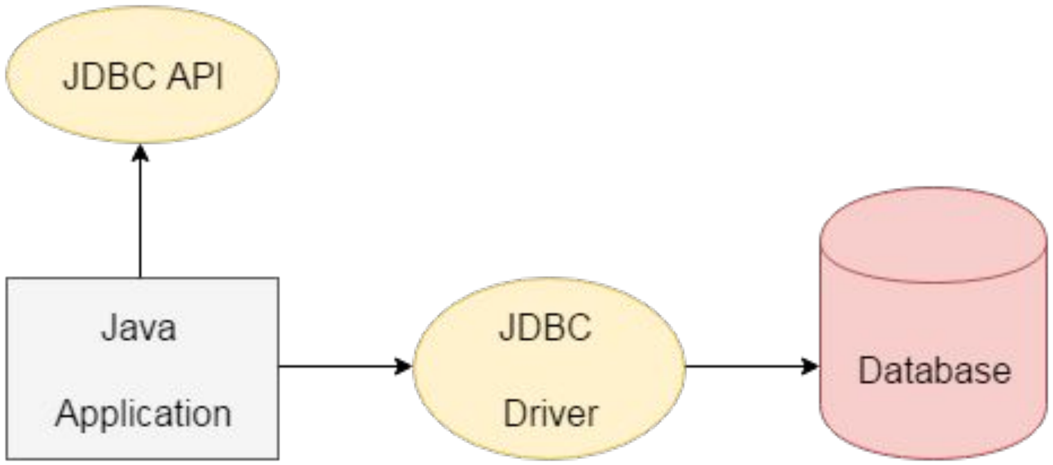
Let’s go!!!

Database Connection (JDBC)

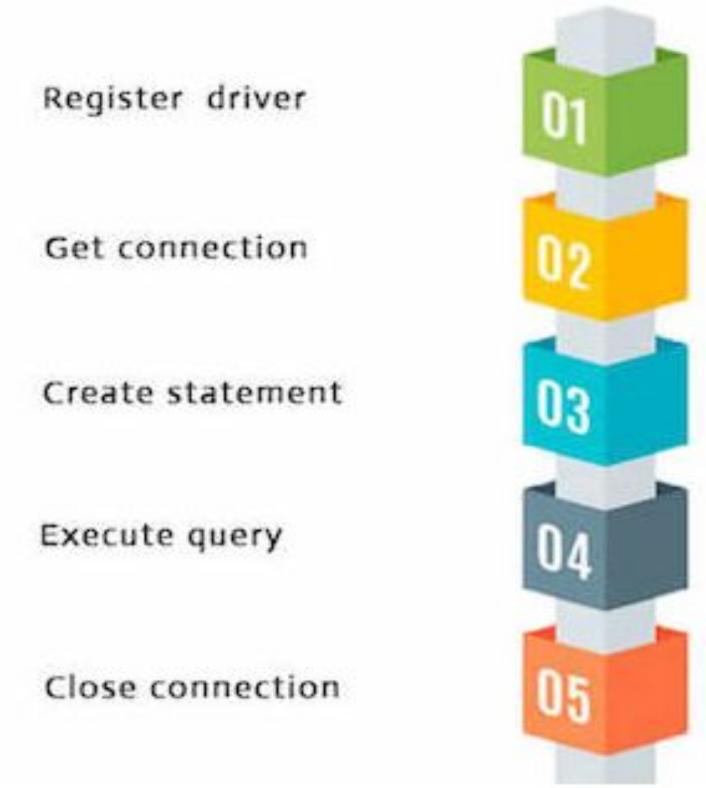
JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database.

Types of JDBC drivers:

- JDBC-ODBC Bridge Driver,
- Native Driver,
- Network Protocol Driver, and
- Thin Driver



Steps to connect application with the database using JDBC:



How to configure MySQL database

Database Connectivity with MySQL:

- **Driver class:** The driver class for the mysql database is com.mysql.jdbc.Driver.
- **Connection URL:** The connection URL for the mysql database is **jdbc:mysql://localhost:3306/databasename** where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and database name. **Username:** The default username for the mysql database is **root.**
- **Password:** It is the password given by the user at the time of installing the mysql database. In this example, we are going to use root as the password.

- To connect java application with the mysql database, mysqlconnector.jar file is required to be loaded. download the jar file mysql-connector.jar

Two ways to load the jar file:

1. Paste the mysqlconnector.jar file in jre/lib/ext folder

Download the mysqlconnector.jar file. Go to jre/lib/ext folder and paste the jar file here.

2. Set classpath

There are two ways to set the classpath:

- temporary
- permanent

How to set the temporary classpath:

open command prompt and write:

```
C:>set classpath=c:\folder\mysql-connector-java-5.0.8-bin.jar;.;
```

How to set the permanent classpath:

- Go to environment variable then click on new tab. In variable name write classpath and in variable value paste the path to the mysqlconnector.jar file by appending mysqlconnector.jar;.; as

```
C:\folder\mysql-connector-java-5.0.8-bin.jar;.;
```

How to configure with Eclipse

Install Java:

Step 1: Go to the Java Downloads Page and click on the option of **Download**

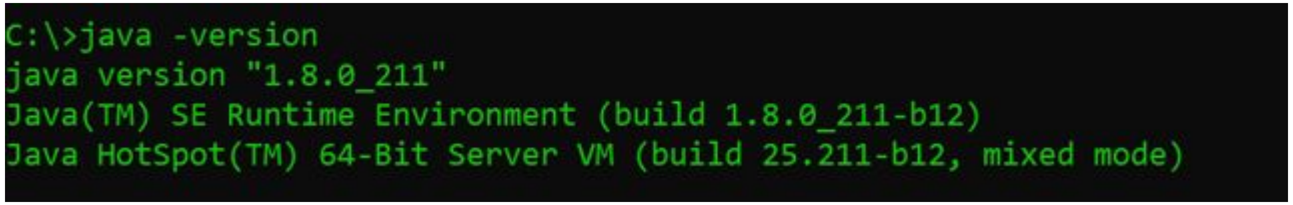
Step 2: Once you click on **Download**, select the **Accept License Agreement** radio button. After that, choose the download link according to your matching system configuration.

Step 3: Now, once the file is downloaded, run the installer and keep clicking on **Next**, till you finally get a dialog box, which say, you have finished downloading.

Step 4: Once the installation is over follow the below instructions to set the path of the file.

- **Step 4.1:** Go to start and search for ‘**System**’. Then, click on ‘**System**’ and go to **Advanced System Settings**.
- **Step 4.2:** Now, click on ‘**Environment Variables**’ under ‘**Advanced**’ tab
- **Step 4.3:** Next, under **System Variables** choose **New.**
- **Step 4.4:** Enter the variable name as ‘**JAVA_HOME**’ and the full path to Java installation directory as per your system
- **Step 4.5:** Next thing that you have to do is to configure your environment variables. Let’s see how to do that. Here, you have to edit the path of the system variable
- **Step 4.6:** Under ‘**Variable value**’, at the end of the line, enter the path of the folder. Now, you can click ‘**OK**’ and you are done.

cross-check the installation, just run following command in cmd – ***java -version***. It should display the installed version of Java in your system.



```
C:\>java -version
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
```

Install Eclipse:

Step 1: Navigate to the following URL – https://www.eclipse.org/downloads/packages/ and select the download link depending on your system architecture – (Windows, Mac OS or Linux) and download it.

Step 2: Once the download is over, extract the zipped file by right-clicking on the folder and choose **Extract All**.

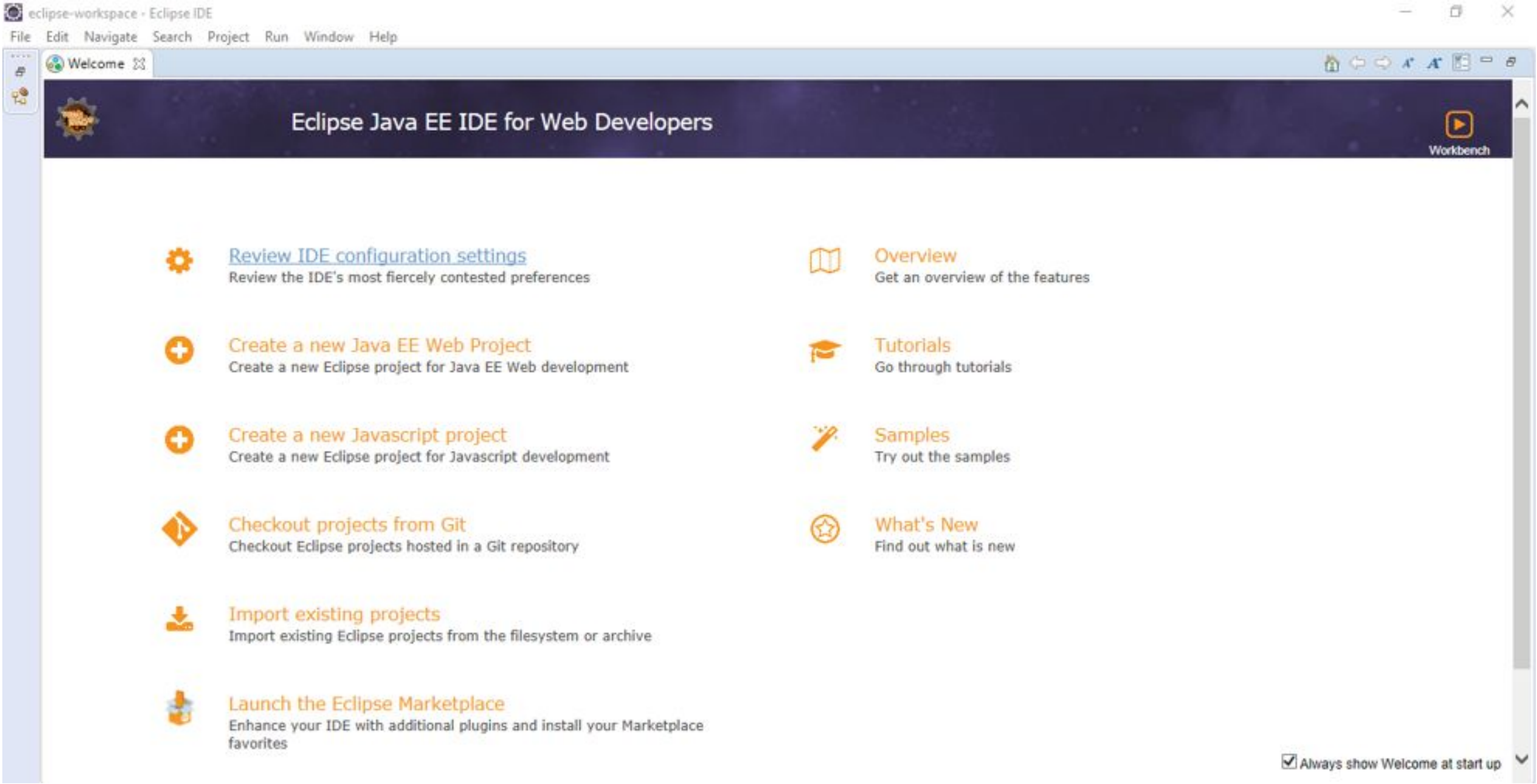
Step 3: You will be then redirected to a dialog box, where you have to choose the directory in which you wish to extract the files. Then click on **Extract**.

Step 4: After extracting files, open the folder and launch **eclipse.exe.**

Step 5: Then, you have to choose the Launch directory for Eclipse and then click on Launch. Refer below.

How to configure with Eclipse

Once Eclipse launches, you will see the below window:



eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Welcome
Workbench

Eclipse Java EE IDE for Web Developers

Review IDE configuration settings
Review the IDE's most fiercely contested preferences

Create a new Java EE Web Project
Create a new Eclipse project for Java EE Web development

Create a new Javascript project
Create a new Eclipse project for Javascript development

Checkout projects from Git
Checkout Eclipse projects hosted in a Git repository

Import existing projects
Import existing Eclipse projects from the filesystem or archive

Launch the Eclipse Marketplace
Enhance your IDE with additional plugins and install your Marketplace favorites

Overview
Get an overview of the features

Tutorials
Go through tutorials

Samples
Try out the samples

What's New
Find out what is new

☑ Always show Welcome at start up

Connection Interface Commands

A Connection is the session between java application and database. The Connection interface is a factory of Statement, Prepared Statement, and Database Meta Data i.e. object of Connection can be used to get the object of Statement and Database Meta Data.

Methods of Connection Interface:

1) public Statement createStatement(): creates a statement object that can be used to execute SQL queries.

2) public Statement createStatement(int resultSetType,int resultSetConcurrency): Creates a Statement object that will generate ResultSet objects with the given type and concurrency.

3) public void setAutoCommit(boolean status): is used to set the commit status.By default it is true.

4) public void commit(): saves the changes made since the previous commit/rollback permanent.

5) public void rollback(): Drops all changes made since the previous commit/rollback.

6) public void close(): closes the connection and Releases a JDBC resources immediately.

Interface Commands:

Command Category	Description
Auth Commands	Use authorization commands to manage user accounts.
Config Commands	Use configuration commands to manage both management and integration configuration for the Management server.
Debug Commands	Use debugging commands to view queues and process stacks.
Mgmt Commands	Use management commands to enable and manage security settings.
Net Commands	Use network commands to manage network configuration.

Statement Interface Commands

Statement Interface Commands:

The Statement interface provides methods to execute queries with the database.

- **public ResultSet executeQuery(String sql):** is used to execute SELECT query. It returns the object of ResultSet.
- **public int executeUpdate(String sql):** is used to execute specified query, it may be create, drop, insert, update, delete etc.
- **public boolean execute(String sql):** is used to execute queries that may return multiple results.
- **public int[] executeBatch():** is used to execute batch of commands.

How to execute query

Execute Current Query:

- To execute a particular query in a batch, just place the cursor on the query to be executed (i.e. before the semicolon separating the query from the next query).
- Select Edit -> Execute Query -> Execute Current Query (or press F9 or click the 'single green arrow' icon in the icon bar). Only the current query would be executed.

Execute All Queries:

- To execute the whole batch of queries entered in the SQL window.
- Select Edit -> Execute Query -> Execute All Queries (or press Shift+F9 or click the 'double green arrow' icon in the icon bar). All queries would be executed one by one, showing result set in separate tab window.

Session Recap