Day 4 Agenda

JPA – Java Persistence API

Quick Recap

1. Spring Core
2. Spring Modules
3. Design Patterns (IoC & DI)
4. SpringBoot (SpringBoot CLI,Using Spring Initializr, Using STS)
5. Spring Data JPA

Controller, Service, DAO (Repository)

<https://spring.io/guides>

JPA - Adv Java Concept

Core JAVA – Java Standard Edition (J2SE) Collections, Generics, Thread, Data Time, IO, File Handling

Adv JAVA – Java Enterprise Edition (J2EE) Web, Servlet, JSP, JMS, JPA, Mail, Portlet, JSF

JPA – Java Persistence API

Persistence – Is a way of storing the data in a RDBMS entity.

In Java, Everything is Object. (Object is the Root of JAVA Lang)

Built-in Classes (Classes created by the JAVA Language creators) System, Date, File, Thread, Object

Built-in Classes = Pre-defined class (Creator defined classes)

Custom Classes/User defined classes = These classes are created by the Java developers around the world.

Abstract means in-complete (Not fully written) Abstraction means hiding something.

Abstract method – A method with only the declaration and no definition/implementation. (A method without body)

JPA is a specification

JDBC is a Specification (It contains Interfaces (abstract methods) which enforces important methods for implementing classes)

JPA Implementations

1. Hibernate (ORM framework)
2. iBatis
3. Spring Data JPA (JPA Implementaion – It uses hibernate)
4. EclipseLink

Important Interfaces of JDBC API (Java DataBase Connectivity) package name :java.sql

1. Connection
2. Driver
3. Statement
4. PreparedStatement
5. CallableStatement
6. ResultSet
7. RowId
8. DatabaseMetaData
9. ResultSetMetaData

Important Classes in JDBC

1. DriverManger

JPA Interfaces (package name :javax.persistence)

1. EntityManager
2. EntityManagerFactory
3. EntityTransaction
4. PersistenceUtil
5. Query
6. TypedQuery
7. StoredProcedureQuery

Class in JPA

1. Persistence

Connecting to Database using JDBC

1. Loading & Registering the Driver (Driver Class Name – fully qualified name of driver class)
2. Establish the connection (URL, username & password)
3. Create & Execute Query (CURD operation query)
4. Process the Result
5. Close all the opened resources

Connecting to Database using JPA

1. Define Entity (@Entity) & Entity Manager (Each Entity need to have a primary key added by @Id annotation)
2. Begin Transaction, perform the operation (save(Entity obj), saveorupdate(Entity obj) ,delete(int id), readAll(), readById(int id))
3. Commit transaction

Downoad EclipseLink2.5.2 using the below link (Download the installer zip file 40mb in size)

<https://www.eclipse.org/eclipselink/releases/2.5.php>

Extract the content in a folder.

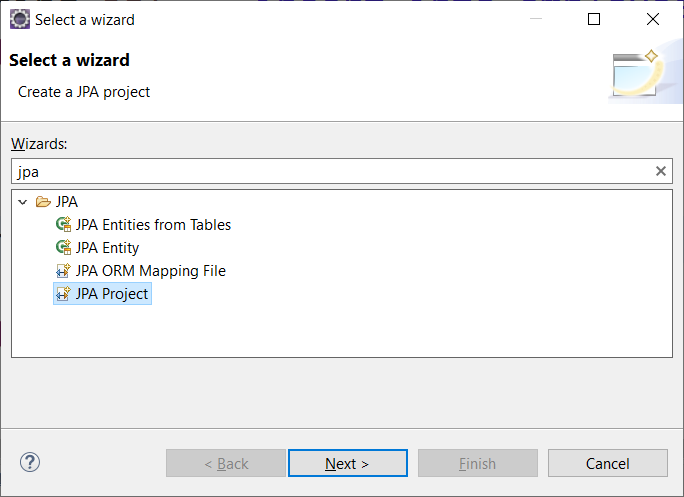
D:\Rakuten\eclipselink-2.5.2\eclipselink\jlib\jpa --- Copy the following 3 files from this location

* javax.persistence,
* org.eclipse.persistence.modelgen,
* org.eclipse.persistence.jpars

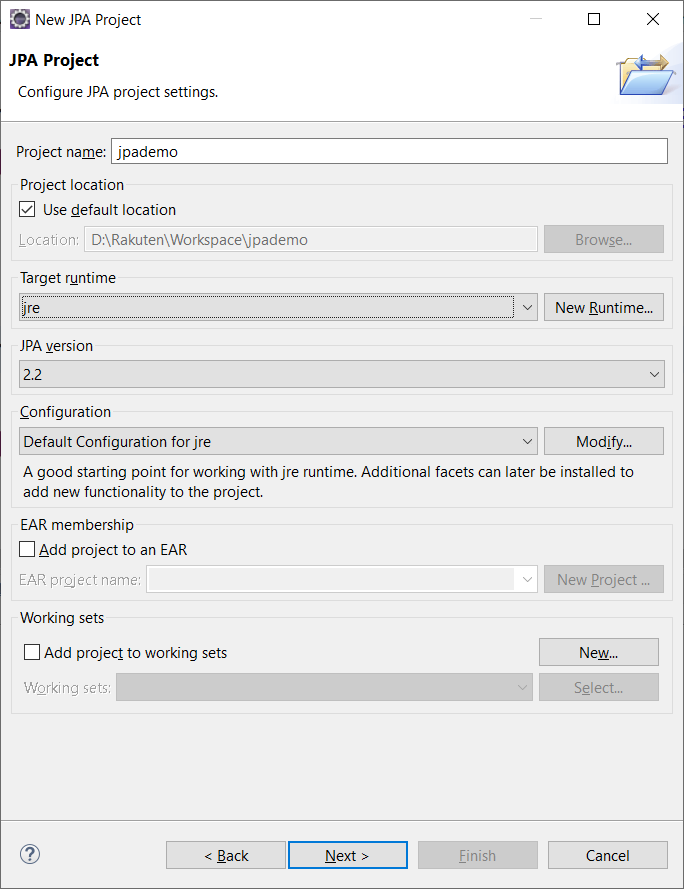
copy the “eclipselink.jar” file from D:\Rakuten\eclipselink-2.5.2\eclipselink\jlib

Create a new JPA project

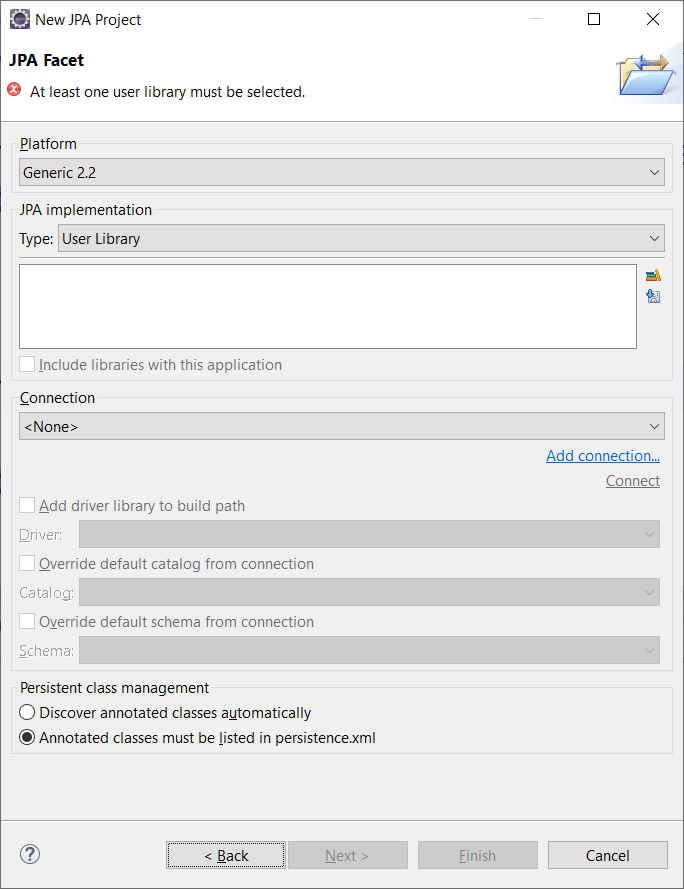
1. File 🡪 New 🡪others (search for JPA) 🡪 select JPA Project



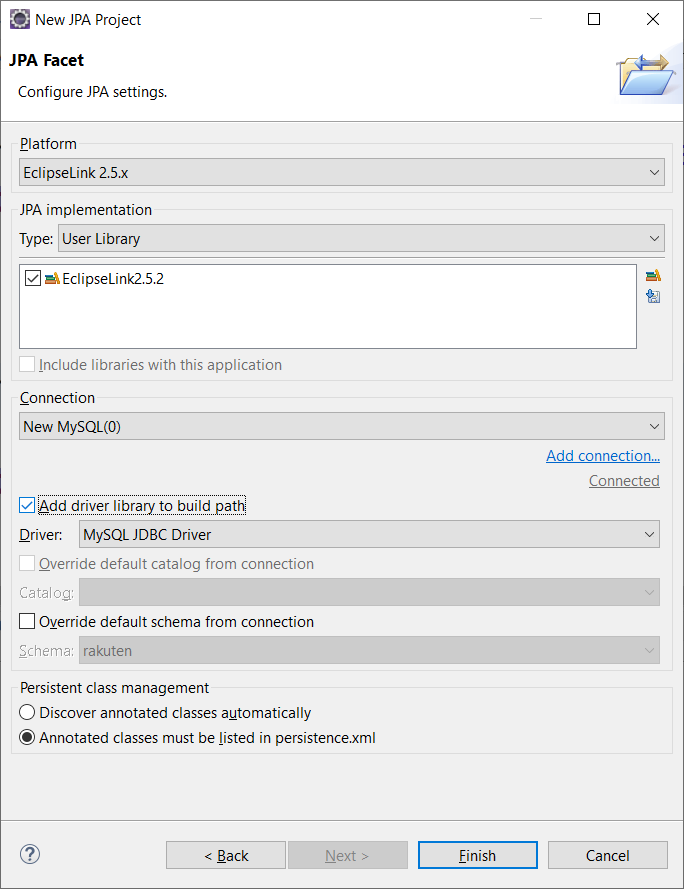
1. Fill the project details as below (make sure to select JPA version number as 2.1)



1. Setup JPA facet (important step)



4. C:\Program Files (x86)\MySQL\Connector J 8.0 – Copy MySQL Connector from here.



<http://wiki.eclipse.org/EclipseLink/Examples>

<https://www.javatips.net/blog/java-persistence-jpa-2-0-tutorial-with-eclipselink>