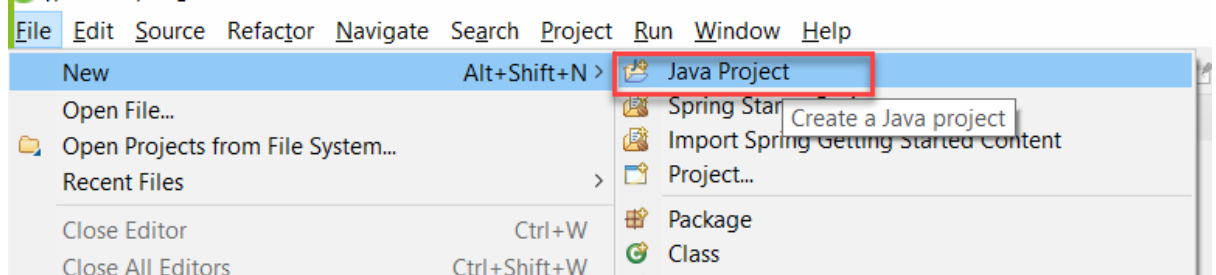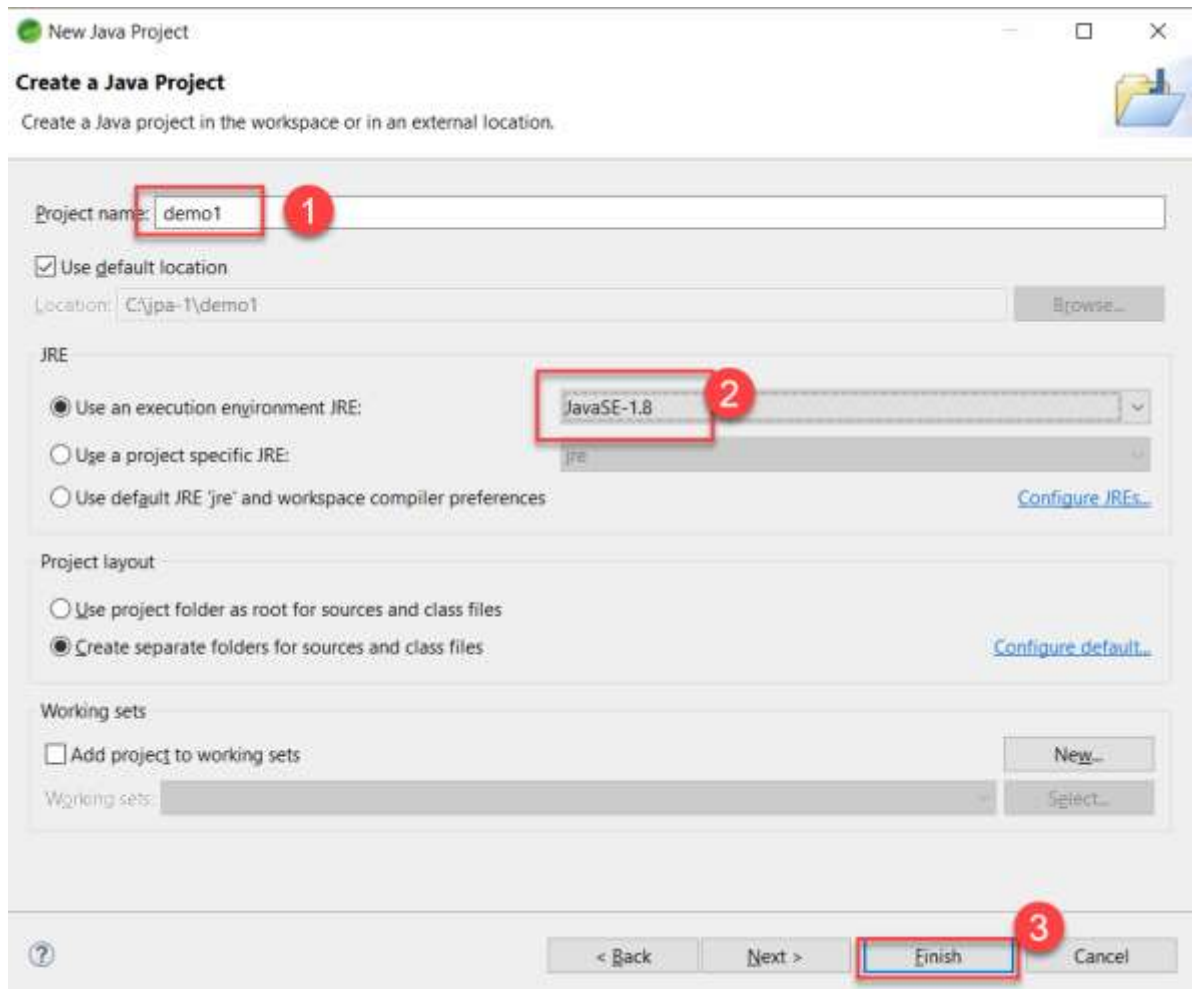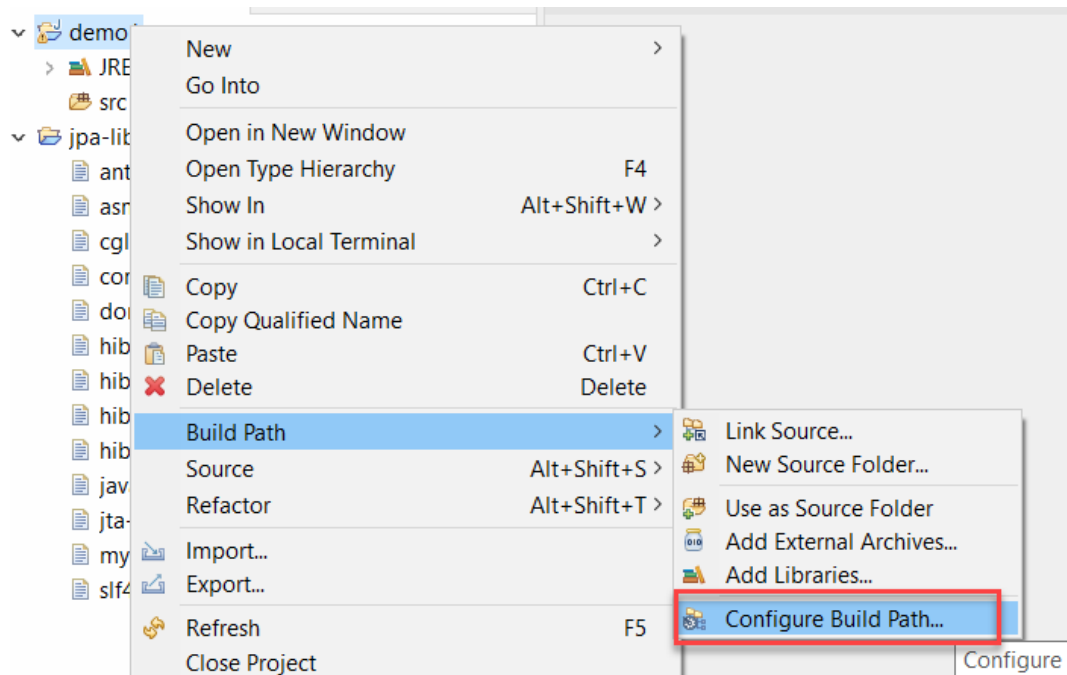# Demo1

1. New "Java Project"
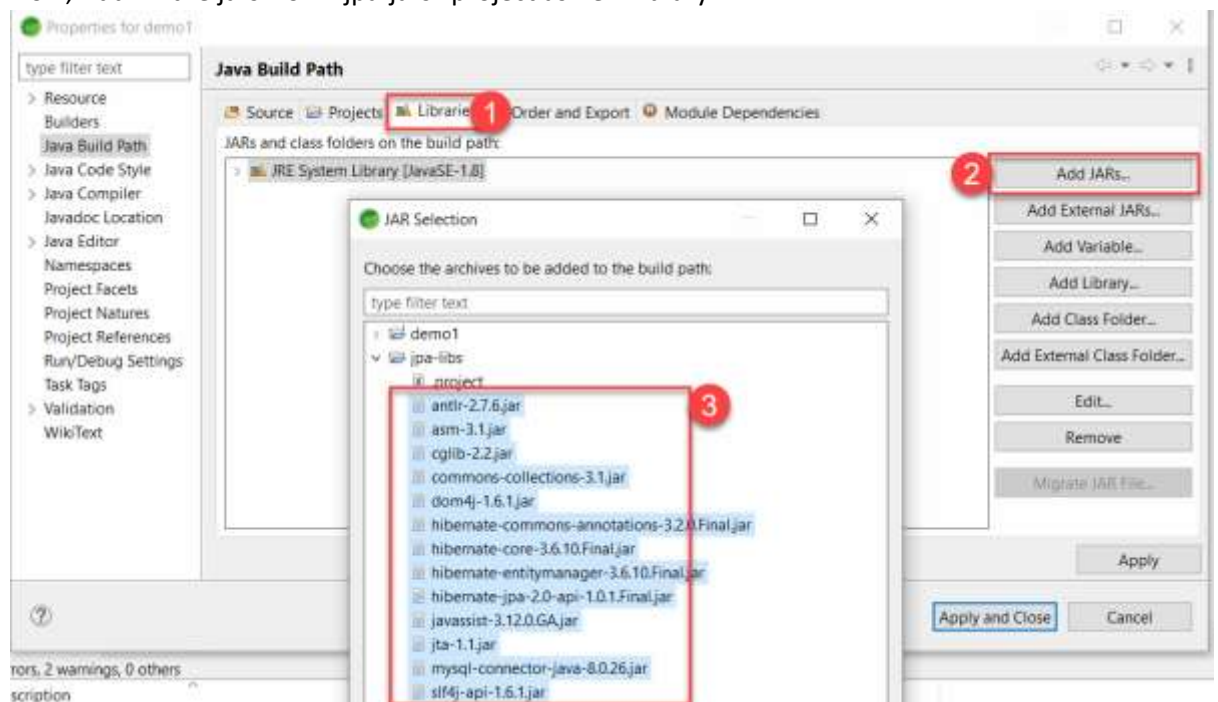

2. Create "demo1" with Java-SE8
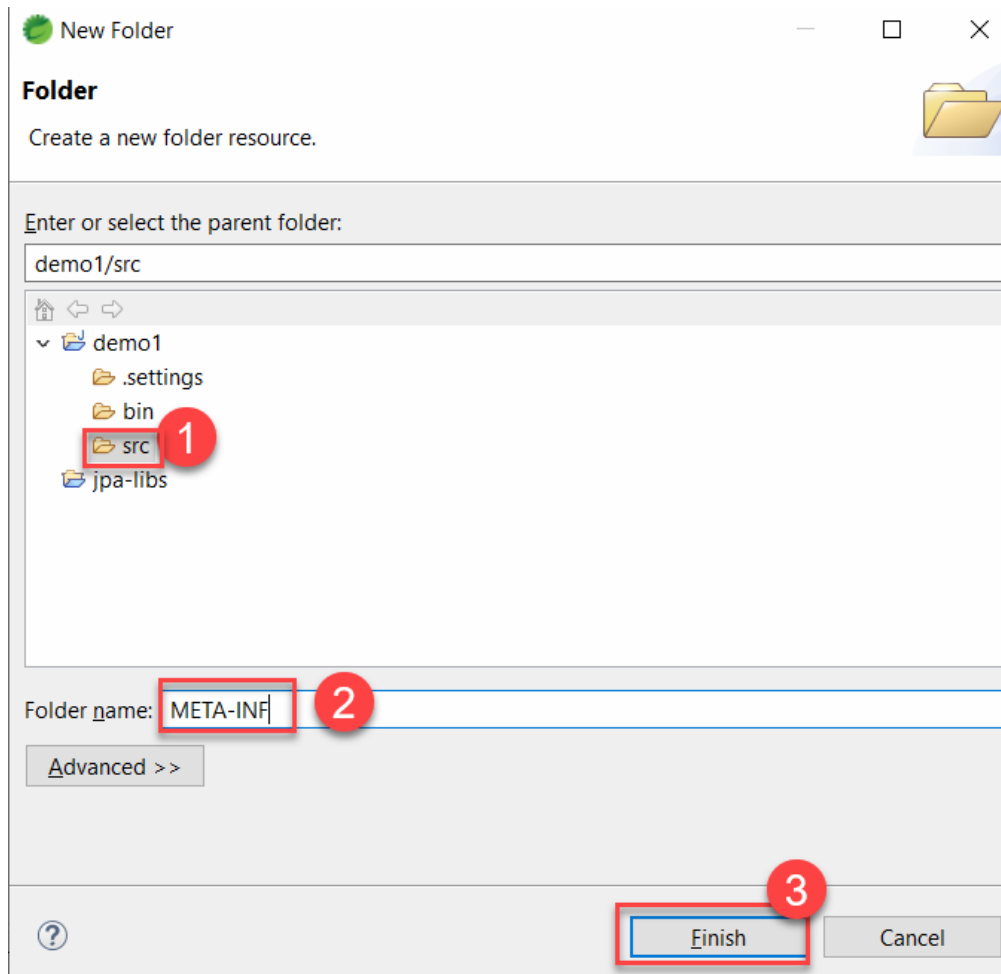

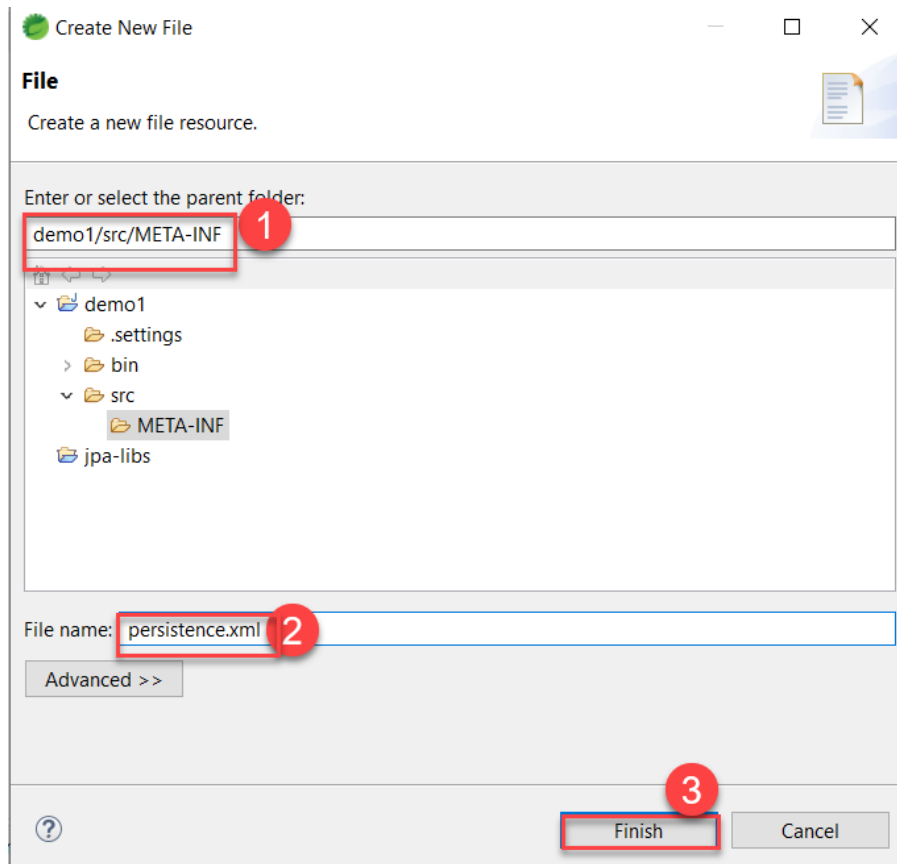3. Once project is created, goto "Build Path" > Configure Build Path

4. Now, Add All the jars from "jpa-jars" project as new library:

5. Right click on project "demo1" and create new "Folder" (DO NOT USE Source Folder)

6. Inside META-INF create a persistence.xml file.
7. Create a new File (CTRL+N > File)

8. Please add following lines to persistence.xml
    NOTE: Please replace username "mahendra" with your mysql username and also replace the password.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
                    http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
    version="2.0">
        <persistence-unit name="JPA-PU" transaction-type="RESOURCE_LOCAL">
            <provider>org.hibernate.ejb.HibernatePersistence</provider>
            <class>com.mahendra.Country</class>
            <properties>
                    <property name="javax.persistence.jdbc.url"
value="jdbc:mysql://localhost:3306/world_x" />
                    <property name="javax.persistence.jdbc.user" value="mahendra" />
                    <property name="javax.persistence.jdbc.password" value="Pass@1234" />
                    <property name="javax.persistence.jdbc.driver"
value="com.mysql.cj.jdbc.Driver" />
                    <!-- Inform Hibernate about which Database to use! -->
                    <property name="hibernate.dialect"
                            value="org.hibernate.dialect.MySQLDialect"/>
                    <!-- Display all generated Queries!! -->
                    <property name="hibernate.show_sql" value="true"/>
                    <!-- Generate table schema (using DDL Queries)
                            drop if exists -->

            </properties>
        </persistence-unit>
</persistence>
```

9. Create a new Class "Country" in package "com.mahendra"

```java
package com.mahendra;

import javax.persistence.*;

@Entity
@Table(name="country")
public class Country {

    @Id
    @Column(length = 3, name="Code")
    private String code;

    @Column(length = 52, name="Name")
    private String name;

    @Column(name="Capital")
    private Integer capital;

    @Column(name="Code2", length = 2)
    private String code2;

    public String getCode() {
        return code;
    }

    public void setCode(String code) {
        this.code = code;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Integer getCapital() {
        return capital;
    }

    public void setCapital(Integer capital) {
        this.capital = capital;
    }

    public String getCode2() {
        return code2;
    }

    public void setCode2(String code2) {
        this.code2 = code2;
    }


    //Empty Constructor
    public Country() {
        super();
        // TODO Auto-generated constructor stub
    }
```

```
        }
```

10. Create a new Class "AppMain" with "main" method

11. Now, the code for Main

```java
package com.mahendra;

import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class AppMain {

        public static void main(String[] args) {
                // Load 'persistence.xml' from 'META-INF' folder and search
for 'JPA-PU'
                // Everything is Case Sensitive ( foldername, filename and
Unit name 'JPA-PU'
                EntityManagerFactory factory =
Persistence.createEntityManagerFactory("JPA-PU");

                EntityManager em = factory.createEntityManager();

                System.out.println("Entity Manager : "+ em);
```

```java
            List<Country> countries = em.createQuery("from Country
c").getResultList();
            System.out.println("Countries found: "+countries.size());

            for(Country c : countries) {
                System.out.println(c.getName());
            }
            em.close();

        }

    }
```

12. Run as Java Application.