JPA with Hibernate 3.0

Lesson 3-Java Persistence API

Lesson Objectives

After completing this lesson, participants will be able to understand:

- Java Persistence API
- Working with JPA
- Managing entities using EntityManager

JPA Overview

Entity Classes

EntityManager

- Persistence Context

EntityManagerFactory

EntityTransaction

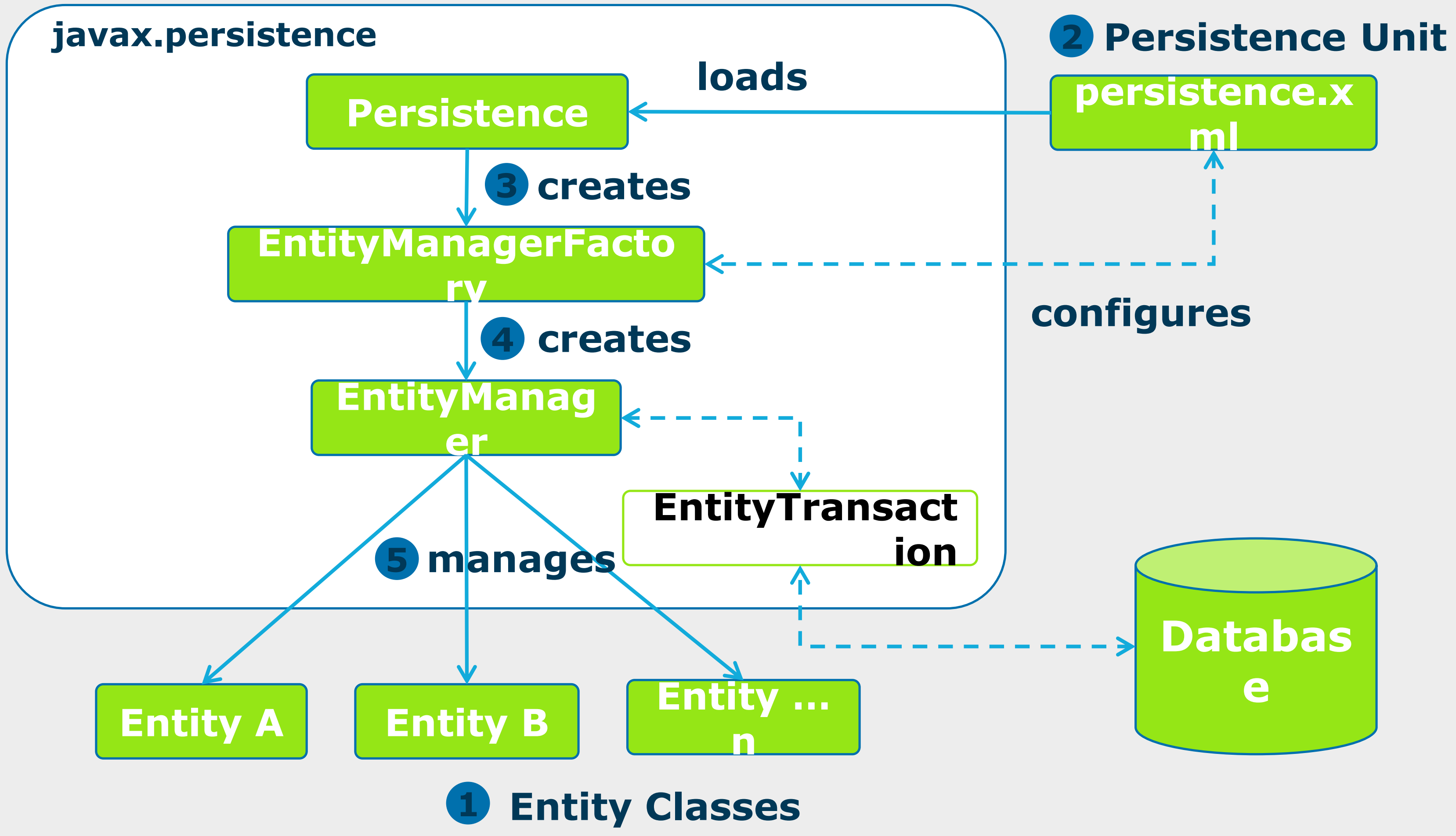
Persistence Unit

- persistence.xml

Java Persistence Query Language (JPQL)

- Query

Working with JPA



Creating Entity Classes

Java POJO classes can be made entity via either one of the following way:

- XML configuration (orm.xml)
- Annotations

A single xml file i.e. orm.xml is required to map entire set of entity classes within an application.

In case of Annotations, it should be marked in individual classes.

```
@Entity
public class Student implements Serializable {
	@Id
	private int studentId;
	private String name;
	//getters and setters
}
```

Entity Annotations

Mandatory Annotations

- @Entity
- @Id

Few more Annotations

- @GeneratedValue
- @Table
- @Column
- @Transient

```
@Entity
@Table(name="student_masters")
public class Student implements Serializable {
        @Id
        @GeneratedValue(strategy=GenerationType.
AUTO)
        @Column(name = "stud_id")
        private int studentId;
        private String name;
```

Persistence Configuration

JPA persistence configuration is done with an XML file called "persistence.xml" which contains information about how to connect to the underlying database.

```
<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="unit-name ">
        <provider> <!-- JPA provider name like hibernate-->
</provider>
       <properties>  <!-- Database properties --></properties>
     </persistence-unit>
</persistence>
```

Obtaining Entity Manager

An EntityManager is responsible for managing entities.
Below are the steps to create instance of EntityManager.

- Create EntityManagerFactory by using the Persistence class
- Call createEntityManager() on an EntityManagerFactory.

```
EntityManagerFactory emf =
Persistence.createEntityManagerFactory(name);
EntityManager em = emf.createEntityManager();
```

Working with Entity Manager

The EntityManager interface is providing the API for interacting with the Entity.

- Creates and removes persistent entity instances
- Finds entities by their primary key
- Allows for data querying
- Interacts with the persistence context

Following are some of the methods of EntityManager

Task	EntityManager method
Save new/detached entity	persist(entity-instance)
Update state of entity	merge(entity-instance)
Remove entity	remove(entity-instance)
Search for entity	find(class,idvalue)

Persisting entity with Entity Manager

```
public static void main(String[] args) {
        EntityManagerFactory factory =
                        Persistence.createEntityManagerFactory("JPA-PU");
        EntityManager em = factory.createEntityManager();
        em.getTransaction().begin();
        Student student = new Student();
        student.setName("John");
        em.persist(student);
        em.getTransaction().comn
        em.close();
        factory.close();
}
```

```
@Entity
public class Student implements
Serializable {
        @Id
        private int studentId;
        private String name;
}
```

3.2: Working with JPA

Demo

JPAStarter Demo

3.2: Working with JPA

Entity CRUD with Entity Manager

```
public Student getStudentById(int id) {
        Student student = entityManager.find(Student.class, id);
        return student;
}

public void addStudent(Student student) {
        entityManager.persist(student);
}

public void removeStudent(Student stude
        entityManager.remove(stude
}

public void updateStudent(Student studen
        entityManager.merge(studen
}
```

```
@Entity
public class Student impl….
{
    @Id
    private int studentId;
    private String name;
}
```

3.2: Working with JPA

Demo

JPACrudOperations Demo

Lab

Lab 1

Summary

In this lesson, you have learned about:

- Setting up JPA in an application
- Configuring database with JPA
- Entity operations using EntityManager

Review Question

Question 1: Which of the following method/s is/are used to complete the transaction?

- Option 1: EntityManager.commit()
- Option 2: EntityManager.getTransaction().commit()
- Option 3: EntityManager.getTransaction().rollback()
- Option 4: All of above

Question 2: Persistence configuration file “persistence.xml” must be saved under META-INF folder.

- True/False