LUXOFT TRAINING

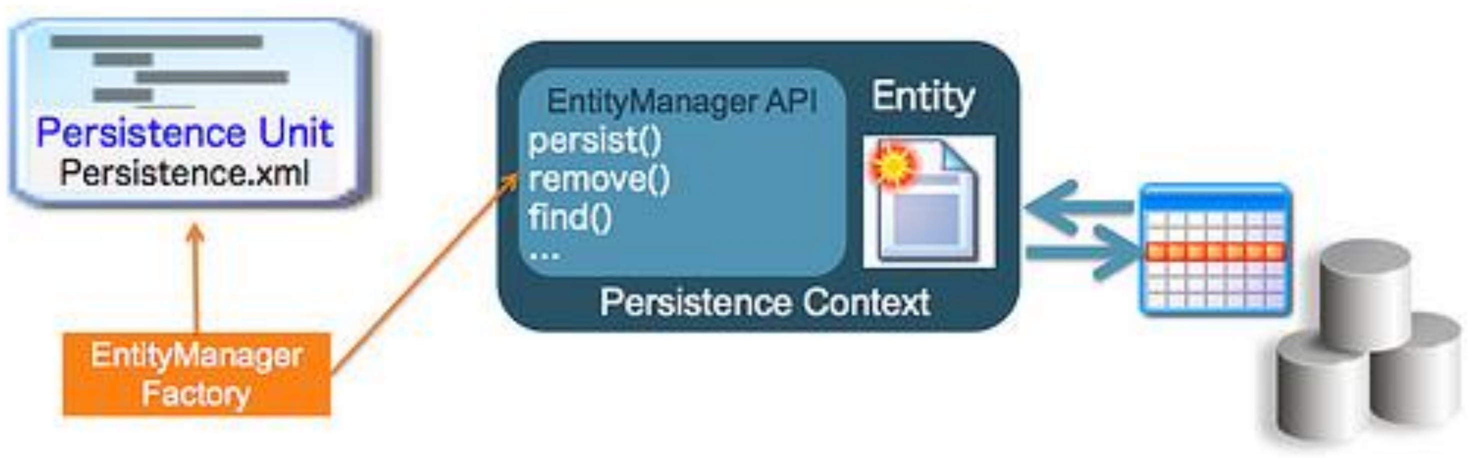
JVA-000 Java Persistence with Hibernate

Module 4
Packaging

Objectives

- Understand what the Persistence Unit is
- Learn how to describe the Persistence Unit
- Learn structure of `persistence.xml` file

Persistence Unit



Persistence unit defines the details that are required to acquire entity manager

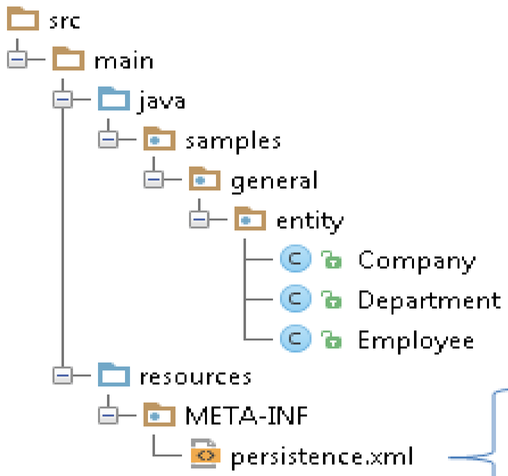
Persistence Unit

A persistence unit (PU) is logical group that includes:

- Entity **manager factory** + **its entity managers** + **configuration**
- The **set of managed classes** included in the persistence unit and managed by the entity managers
- **Mapping metadata** that specifies the mapping of the classes to the database

PU is defined in `/META-INF/persistence.xml` file

Persistence Unit



```java
@Entity
@Table(name = "T_COMPANY")
public class Company {
    @Id
    @Column(name = "COMPANY_ID")
    private int id;
    @Column(name = "COMPANY_NAME")
    private String name;
}
```

```xml
<persistence-unit name="samples.demo" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
    <non-jta-data-source>jdbc/DataSource</non-jta-data-source>
    <class>samples.general.entity.Company</class>
    <class>samples.general.entity.Department</class>
    <class>samples.general.entity.Employee</class>
    <properties>
        <property name="hibernate.show_sql" value="true"/>
    </properties>
</persistence-unit>
```

Create EntityManager for PU and find an entity:

```java
EntityManagerFactory emf = Persistence.createEntityManagerFactory("samples.demo");
EntityManager em = emf.createEntityManager();

em.getTransaction().begin();
Company company = em.find(Company.class, 1);
em.getTransaction().commit();
```

Simple example of how to define PU and use it in JPA application

Persistence Unit

PU must have name that’s unique within application

It’s recommended to explicitly list entity classes in PU to insure the portability of application

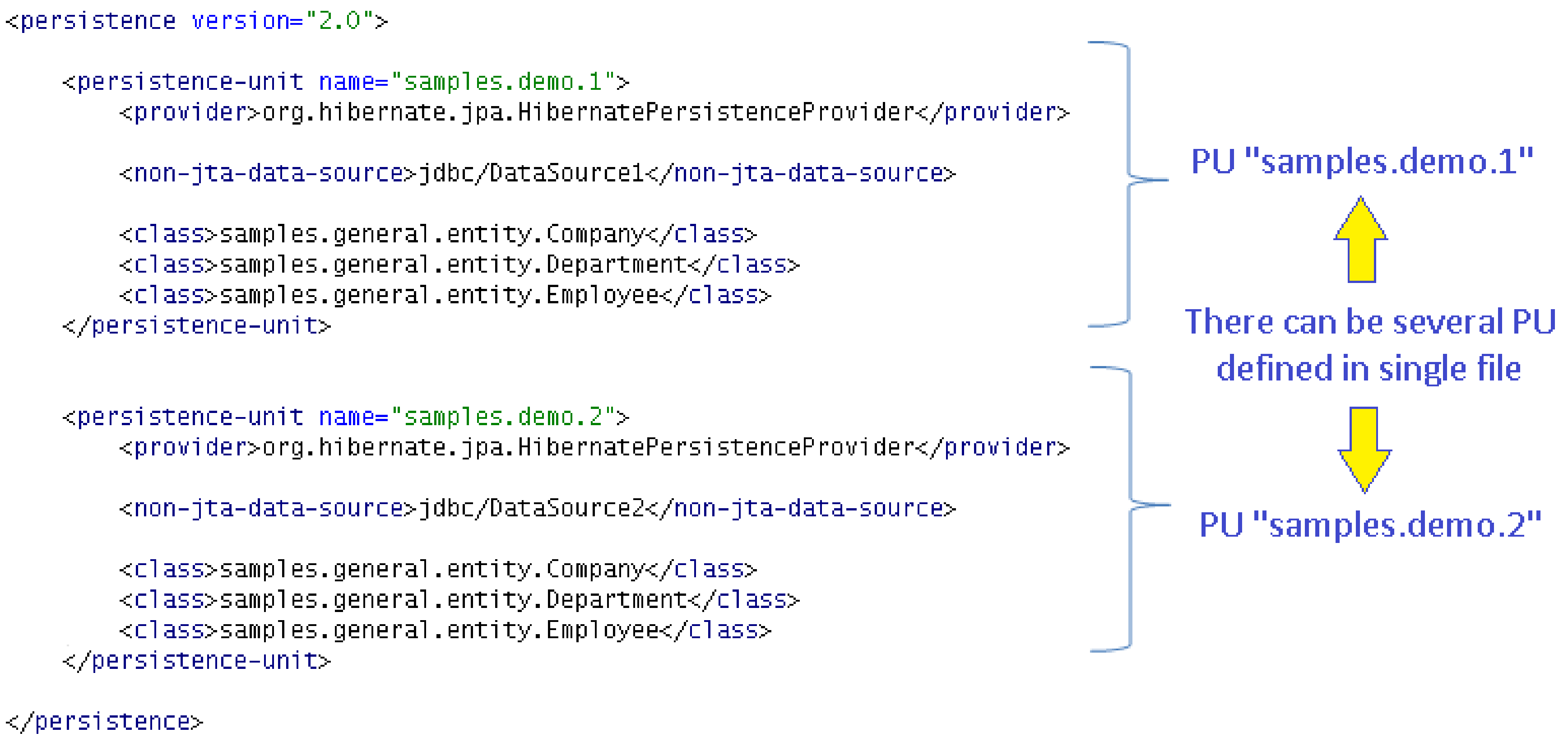
The `persistence.xml` file may contain more than one persistence unit within the same application

`persistence.xml` file

File `/META-INF/persistence.xml` contains:

- Managed persistence classes included in the persistence unit
- Object/Relational mapping information for those classes
- Scripts for use in schema generation
- Bulk loading of data
- Configuration information for PU

persistence.xml file



```xml
<persistence version="2.0">

    <persistence-unit name="samples.demo.1">
        <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>

        <non-jta-data-source>jdbc/DataSource1</non-jta-data-source>

        <class>samples.general.entity.Company</class>
        <class>samples.general.entity.Department</class>
        <class>samples.general.entity.Employee</class>
    </persistence-unit>


    <persistence-unit name="samples.demo.2">
        <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>

        <non-jta-data-source>jdbc/DataSource2</non-jta-data-source>

        <class>samples.general.entity.Company</class>
        <class>samples.general.entity.Department</class>
        <class>samples.general.entity.Employee</class>
    </persistence-unit>

</persistence>
```

The `persistence.xml` file example

`persistence.xml` file

The root element of the `persistence.xml` file is the `<persistence>` element

The `<persistence>` element consists of one or more `<persistence-unit>` elements

The `<persistence-unit>` element has attributes:

- **`name`** – name of PU *(Mandatory)*
- **`transaction-type`**:
    - `JTA` – for entity managers supporting JTA transaction management
    - `RESOURCE_LOCAL` – for entity managers that manages transactions via JPA `Transaction` interface

`persistence.xml` file

Nested elements of `<persistence-unit>` element:

- **`description`** – provides information about PU *(Optional)*
- **`provider`** – JPA provider class name (implements `javax.persistence.spi.PersistenceProvider`)
- **`jta-data-source`** – JNDI name of SQL data source supporting JTA (used with `transaction-type=JTA`)
- **`non-jta-data-source`** – JNDI name of SQL data source not supporting JTA (used with `transaction-type=RESOURCE_LOCAL`)
- **`class`** – defines JPA entity class

`persistence.xml` file

The `<persistence-unit>` elements *(continue)*:

- **`exclude-unlisted-classes`** – if TRUE the only listed classes are scanned for entities definition
- **`shared-cache-mod`**e – enable/disable entities caching
- **`validation-mode`** – validation mode to be used for the persistence unit
- **`properties`** – list of standard and vendor-specific properties and hints

`persistence.xml` file

The `<shared-cache-mode>` values:

- **`ALL`** – all entities and entity-related state and data are cached
- **`NONE`** – caching is disabled for the persistence unit
- **`ENABLE_SELECTIVE`** – caching is enabled for all entities with `@Cacheable(true)` is specified
- **`DISABLE_SELECTIVE`** – caching is enabled for all entities except those for which `@Cacheable(false)` is specified
- **`UNSPECIFIED`** – caching behavior is undefined (provider-specific defaults may apply)

`persistence.xml` file

The `<validation-mode>` values:

- **`AUTO`** – if a Bean Validation provider is present in the environment the persistence provider must perform the automatic validation of entities. If no Bean Validation provider is present in the environment, no lifecycle event validation takes place. This is the default behavior.
- **`CALLBACK`** – the persistence provider must perform the lifecycle event validation
- **`NONE`** – the persistence provider mustn’t perform lifecycle event validation

`persistence.xml` file

Standard JPA properties for `<properties>`:

- **`javax.persistence.lock.timeout`** – value in milliseconds for pessimistic lock timeout
- **`javax.persistence.query.timeout`** – value in milliseconds for query timeout
- **`javax.persistence.jdbc.driver`** – fully qualified name of the driver class
- **`javax.persistence.jdbc.url`** – driver-specific URL
- **`javax.persistence.jdbc.user`** – username used by database connection
- **`javax.persistence.jdbc.password`** – username used by database connection

`persistence.xml` file

Standard JPA properties for `<properties>` (continue):

- **`javax.persistence.schema-generation.create-script-source`** – URL to file with database schema creation script
- **`javax.persistence.schema-generation.drop-script-source`** – URL to file with database schema deletion script
- **`javax.persistence.sql-load-script-source`** – URL to file with initial data loading to database

LUXOFT TRAINING

Thank you for your attention!

Questions?