Free JPA Project1

Prepared by: Nadir ÇİFTÇİ

February, 2018

Concepts

- Entity
- Entity Manager
- Persistence Context
- Unidirectional/bidirectional
- JPQL

Maven Coordinates - JPA

```
<dependencies>
       <dependency>
          <groupId>org.hibernate</groupId>
          <artifactId>hibernate-entitymanager</artifactId>
          <version>5.1.4.Final</version>
       </dependency>
</dependencies>
```

Maven Coordinates – Apache Derby

```xml
<dependencies>
       <dependency>
          <groupId>org.apache.derby</groupId>
          <artifactId>derby</artifactId>
          <version>10.14.1.0</version>
       </dependency>
       <dependency>
          <groupId>org.apache.derby</groupId>
          <artifactId>derbyClient</artifactId>
          <version>10.14.1.0</version>
       </dependency>
       <dependency>
          <groupId>org.apache.derby</groupId>
          <artifactId>derbynet</artifactId>
          <version>10.14.1.0</version>
       </dependency>
</dependencies>
```

Persistence.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" 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">
     <persistence-unit name="SAMPLE_PU_LOCAL" transaction-type="RESOURCE_LOCAL">
          <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
          <class>com.ndr.free.entity.SampleEntity</class>
          <properties>
               <property name="javax.persistence.jdbc.driver"
                         value="org.apache.derby.jdbc.EmbeddedDriver" />
               <property name="javax.persistence.jdbc.url"
                         value="jdbc:derby:./src/main/resources/SampleDb;create=true" />
               <property name="javax.persistence.jdbc.user" value="sample_user" />
               <property name="javax.persistence.jdbc.password" value="sample_password" />
               <property name="hibernate.dialect" value="org.hibernate.dialect.DerbyDialect"/>
               <property name="hibernate.show_sql" value="true"/>
               <property name="hibernate.hbm2ddl.auto" value="update" />
          </properties>
    </persistence-unit>
</persistence>
```

SampleMain1.java

```
public class SampleMain1 {

    public static void main(String[] args) {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("SAMPLE_PU_LOCAL");
        EntityManager em = emf.createEntityManager();
        try {
            EntityTransaction trans = em.getTransaction();
            SampleEntity sampleEntity = new SampleEntity(1, "strVal");
            trans.begin();
            em.persist(sampleEntity);
            trans.commit();
        }catch(Exception e){
            e.printStackTrace();
        }
        em.close();
        emf.close();
    }

}
```

SampleEntity.java

```java
@Entity
@Table(name="SAMPLE_TABLE")
public class SampleEntity {

	@Id
	@Column(name = "ID")
	private long longProperty;

	@Column(name = "STR_COL")
	private String strProperty;

	public SampleEntity() {
		super();
	}

	public SampleEntity(long longProperty, String strProperty) {
		this.longProperty = longProperty;
		this.strProperty = strProperty;
	}

	public long getLongProperty() {
		return longProperty;
	}

	public void setLongProperty(long longProperty) {
		this.longProperty = longProperty;
	}

	public String getStrProperty() {
		return strProperty;
	}

	public void setStrProperty(String strProperty) {
		this.strProperty = strProperty;
	}

}
```

SampleMain2.java

```
public class SampleMain2 {

     public static void main(String[] args) {
          EntityManagerFactory emf = Persistence.createEntityManagerFactory("SAMPLE_PU_LOCAL");
          EntityManager em = emf.createEntityManager();
          SampleEntity sampleEntity = em.find(SampleEntity.class, new Long(1));
          System.out.println(sampleEntity.getStrProperty());
          em.close();
          emf.close();
    }

}
```

SampleMain3.java

```java
public class SampleMain3 {

    public static void main(String[] args) {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("SAMPLE_PU_LOCAL");
        EntityManager em = emf.createEntityManager();
        try {
            EntityTransaction trans = em.getTransaction();
            SampleParentEntity sampleParentEntity = new SampleParentEntity("strVal0");

            SampleChildEntity sampleChildEntity1 = new SampleChildEntity("strVal1");
            sampleChildEntity1.setSampleParentEntity(sampleParentEntity);
            sampleParentEntity.getLstChildEntity().add(sampleChildEntity1);

            SampleChildEntity sampleChildEntity2 = new SampleChildEntity("strVal2");
            sampleChildEntity2.setSampleParentEntity(sampleParentEntity);
            sampleParentEntity.getLstChildEntity().add(sampleChildEntity2);

            SampleChildEntity sampleChildEntity3 = new SampleChildEntity("strVal3");
            sampleChildEntity3.setSampleParentEntity(sampleParentEntity);
            sampleParentEntity.getLstChildEntity().add(sampleChildEntity3);

            trans.begin();
            em.persist(sampleParentEntity);
            trans.commit();

        }catch(Exception e){
            e.printStackTrace();
        }
        em.close();
        emf.close();
    }

}
```

SampleParentEntity.java

```java
@Entity
@Table(name="SAMPLE_PARENT_TABLE")
@NamedQuery(name="SampleParentEntity.findById", query="SELECT sampleParentEntity FROM SampleParentEntity
     sampleParentEntity WHERE sampleParentEntity.id = :id")
public class SampleParentEntity {

      @Id
      @GeneratedValue(strategy = GenerationType.AUTO)
      @Column(name = "ID")
      private Long id;

      @Column(name = "STR_COL")
      private String strProperty;

      @OneToMany(cascade=CascadeType.ALL, fetch=FetchType.EAGER)
      @JoinColumn(name="SAMPLE_PARENT_TABLE_ID")
      private List<SampleChildEntity> lstChildEntity = new ArrayList<SampleChildEntity>();

      public SampleParentEntity() {
            super();
      }

      public SampleParentEntity(String strProperty) {
            this.strProperty = strProperty;
      }

      public Long getId() {
            return id;
      }

      public void setId(Long id) {
            this.id = id;
      }
```

SampleParentEntity.java(continued)

```java
    public String getStrProperty() {
        return strProperty;
    }

    public void setStrProperty(String strProperty) {
        this.strProperty = strProperty;
    }

    public List<SampleChildEntity> getLstChildEntity() {
        return lstChildEntity;
    }

}
```

SampleChildEntity.java

```java
@Entity
@Table(name="SAMPLE_CHILD_TABLE")
public class SampleChildEntity {

      @Id
      @GeneratedValue(strategy = GenerationType.AUTO)
      @Column(name = "ID")
      private long id;

      @Column(name ="SAMPLE_PARENT_TABLE_ID")
      private Long sampleParentTableId;

      @Column(name = "STR_COL")
      private String strProperty;

      @ManyToOne
      private SampleParentEntity sampleParentEntity;

      public SampleChildEntity() {
            super();
      }

      public SampleChildEntity(String strProperty) {
            this.strProperty = strProperty;
      }

      public long getId() {
            return id;
      }

      public void setId(long id) {
            this.id = id;
      }
```

SampleChildEntity.java(continued)

```java
    public Long getSampleParentTableId() {
        return sampleParentTableId;
    }

    public void setSampleParentTableId(Long sampleParentTableId) {
        this.sampleParentTableId = sampleParentTableId;
    }

    public String getStrProperty() {
        return strProperty;
    }

    public void setStrProperty(String strProperty) {
        this.strProperty = strProperty;
    }

    public SampleParentEntity getSampleParentEntity() {
        return sampleParentEntity;
    }

    public void setSampleParentEntity(SampleParentEntity sampleParentEntity) {
        this.sampleParentEntity = sampleParentEntity;
    }

}
```

SampleMain4.java

```
public class SampleMain4 {

      public static void main(String[] args) {
            EntityManagerFactory emf = Persistence.createEntityManagerFactory("SAMPLE_PU_LOCAL");
            EntityManager em = emf.createEntityManager();
            try {
                  Query query = em.createQuery("Select sampleParentEntity from SampleParentEntity
                        sampleParentEntity",SampleParentEntity.class);
                  List<SampleParentEntity> sampleParentEntityLst = query.getResultList();
                  for(SampleParentEntity sampleParentEntity :sampleParentEntityLst){
                        System.out.println("Parent id: " + sampleParentEntity.getId() + " strProperty:
                              " + sampleParentEntity.getStrProperty());
                        for(SampleChildEntity sampleChildEntity:  sampleParentEntity.getLstChildEntity()){
                              System.out.println("Child id: " + sampleChildEntity.getId() + " strProperty: "
                                    + sampleChildEntity.getStrProperty());
                        }
                  }
            }catch(Exception e){
                  e.printStackTrace();
            }
            em.close();
            emf.close();
      }

}
```

SampleMain5.java

```
public class SampleMain5 {

      public static void main(String[] args) {
            EntityManagerFactory emf = Persistence.createEntityManagerFactory("SAMPLE_PU_LOCAL");
            EntityManager em = emf.createEntityManager();
            try {
                  Query query = em.createQuery("Select sampleChildEntity from SampleChildEntity
                        sampleChildEntity",SampleChildEntity.class);
                  List<SampleChildEntity> sampleChildEntityLst = query.getResultList();
                  for(SampleChildEntity sampleChildEntity: sampleChildEntityLst){
                        System.out.println("Parent id: " + sampleChildEntity.getSampleParentEntity().getId()
                              + " Child id: " + sampleChildEntity.getId() + " strProperty: " +
                              sampleChildEntity.getStrProperty());
                  }
            }catch(Exception e){
                  e.printStackTrace();
            }
            em.close();
            emf.close();
      }

}
```

SampleMain6.java

```java
public class SampleMain6 {

	public static void main(String[] args) {
		EntityManagerFactory emf = Persistence.createEntityManagerFactory("SAMPLE_PU_LOCAL");
		EntityManager em = emf.createEntityManager();
		try {
			TypedQuery<SampleParentEntity> query = em.createNamedQuery("SampleParentEntity.findById",
				SampleParentEntity.class);
			query.setParameter("id", new Long(1));
			SampleParentEntity sampleParentEntity = query.getSingleResult();
			if ( sampleParentEntity != null ){
				System.out.println("Parent id: " + sampleParentEntity.getId() + " strProperty: " +
					sampleParentEntity.getStrProperty());
				for(SampleChildEntity sampleChildEntity: sampleParentEntity.getLstChildEntity()){
					System.out.println("Child id: " + sampleChildEntity.getId() + " strProperty: " +
						sampleChildEntity.getStrProperty());
				}
			}
		}catch(Exception e){
			e.printStackTrace();
		}
		em.close();
		emf.close();
	}

}
```

SampleMain7.java

```java
public class SampleMain7 {

    public static void main(String[] args) {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("SAMPLE_PU_LOCAL");
        EntityManager em = emf.createEntityManager();
        try {
            Query query = em.createNativeQuery("SELECT * FROM SAMPLE_PARENT_TABLE WHERE ID = :id",
                SampleParentEntity.class);
            query.setParameter("id", new Long(1));
            SampleParentEntity sampleParentEntity = (SampleParentEntity) query.getSingleResult();
            if ( sampleParentEntity != null ){
                System.out.println("Parent id: " + sampleParentEntity.getId() + " strProperty: " +
                    sampleParentEntity.getStrProperty());
            }
        }catch(Exception e){
            e.printStackTrace();
        }
        em.close();
        emf.close();
    }

}
```