# Examples of JPA, Hibernate, and Spring Data JPA

## 1. JPA Example (Standard Java with EntityManager)

JPA is just a specification. Below is a standard JPA example using `EntityManager` without any Spring framework.  
  
Entity Class:  
@Entity  
public class User {  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private Long id;  
 private String name;  
 // getters and setters  
}  
  
Main Class:  
EntityManagerFactory emf = Persistence.createEntityManagerFactory("example-unit");  
EntityManager em = emf.createEntityManager();  
em.getTransaction().begin();  
User user = new User();  
user.setName("John");  
em.persist(user);  
em.getTransaction().commit();  
em.close();

## 2. Hibernate Example (Standalone ORM)

Hibernate can be used standalone without JPA.  
  
Configuration:  
hibernate.cfg.xml with database and mapping settings  
  
Entity Class:  
public class User {  
 private Long id;  
 private String name;  
 // annotate with @Entity, @Id etc. using Hibernate annotations  
}  
  
Session Code:  
SessionFactory factory = new Configuration().configure().buildSessionFactory();  
Session session = factory.openSession();  
Transaction tx = session.beginTransaction();  
User user = new User();  
user.setName("Alice");  
session.save(user);  
tx.commit();  
session.close();

## 3. Spring Data JPA Example (Using JpaRepository)

Spring Data JPA builds on JPA and Hibernate to simplify database access.  
  
Entity Class:  
@Entity  
public class User {  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private Long id;  
 private String name;  
}  
  
Repository Interface:  
public interface UserRepository extends JpaRepository<User, Long> {  
 List<User> findByName(String name);  
}  
  
Service or Main Class:  
@Autowired  
private UserRepository userRepository;  
  
userRepository.save(new User("Emma"));  
List<User> users = userRepository.findByName("Emma");