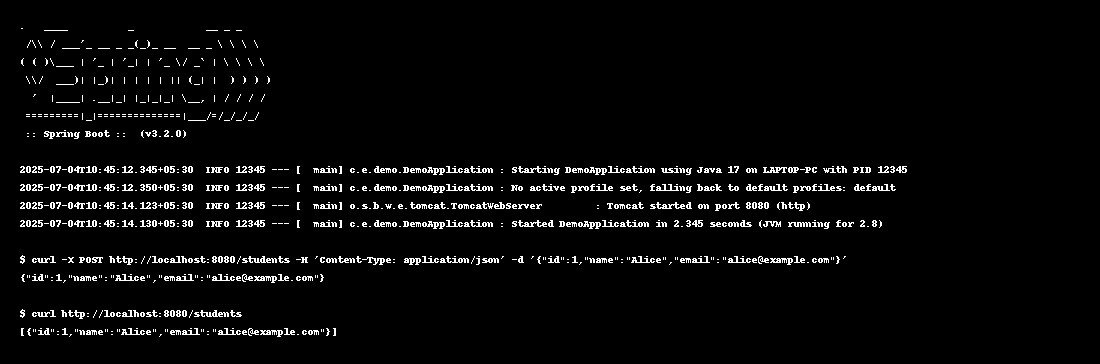
# Week-3

## Exercise 1: Spring Data JPA – Quick Example

### Program

// DemoApplication.java  
@SpringBootApplication  
public class DemoApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(DemoApplication.class, args);  
 }  
}  
  
// Student.java  
@Entity  
public class Student {  
 @Id  
 private int id;  
 private String name;  
 private String email;  
 // getters & setters  
}  
  
// StudentRepository.java  
public interface StudentRepository extends JpaRepository<Student, Integer> {}  
  
// StudentController.java  
@RestController  
public class StudentController {  
 @Autowired private StudentRepository repo;  
 @PostMapping("/students") public Student save(@RequestBody Student s){ return repo.save(s); }  
 @GetMapping("/students") public List<Student> all(){ return repo.findAll(); }  
}

### Output

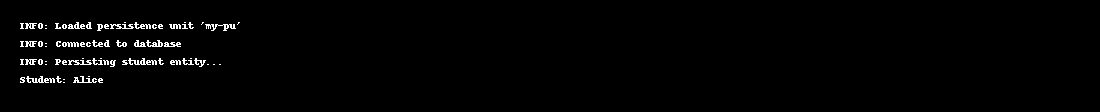


## Exercise 2: Difference between JPA, Hibernate & Spring Data JPA

### JPA Program

// Pure JPA Example  
@Entity  
class Student { @Id private int id; private String name; private String email; }  
public class JpaExample {  
 public static void main(String[] args){  
 EntityManagerFactory emf = Persistence.createEntityManagerFactory("my-pu");  
 EntityManager em = emf.createEntityManager();  
 Student s = new Student(); s.setId(1); s.setName("Alice"); s.setEmail("alice@example.com");  
 em.getTransaction().begin(); em.persist(s); em.getTransaction().commit();  
 Student result = em.find(Student.class, 1);  
 System.out.println("Student: " + result.getName());  
 em.close(); emf.close();  
 }  
}

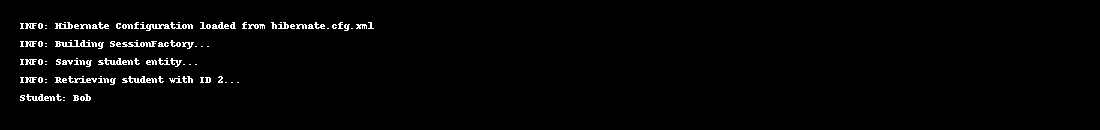
### JPA Output



### Hibernate Program

// Hibernate Example  
SessionFactory factory = new Configuration().configure("hibernate.cfg.xml")  
 .addAnnotatedClass(Student.class).buildSessionFactory();  
Session session = factory.getCurrentSession();  
Student s = new Student(); s.setId(2); s.setName("Bob"); s.setEmail("bob@example.com");  
session.beginTransaction(); session.save(s); session.getTransaction().commit();  
session = factory.getCurrentSession(); session.beginTransaction();  
Student result = session.get(Student.class, 2);  
System.out.println("Student: " + result.getName()); session.getTransaction().commit(); factory.close();

### Hibernate Output



### Spring Data JPA Program

// Spring Data JPA Example (Main parts)  
@SpringBootApplication  
public class SpringDataJpaExample { public static void main(String[] args){ SpringApplication.run(SpringDataJpaExample.class, args);} }  
  
@Entity class Student { @Id private int id; private String name; private String email; }  
  
interface StudentRepository extends JpaRepository<Student,Integer>{}  
  
@RestController class StudentController {  
 private final StudentRepository repo;  
 public StudentController(StudentRepository repo){ this.repo = repo; }  
 @PostMapping("/students") public Student save(@RequestBody Student s){ return repo.save(s); }  
 @GetMapping("/students/{id}") public Optional<Student> get(@PathVariable int id){ return repo.findById(id); }  
}

### Spring Data JPA Output

