***** With DataBase CRUD SpringBoot ******

```
@RestController
public class StudentController {

ArrayList<Student> al = new ArrayList<>();

@Autowired
SessionFactory sf;
Student stud;

@RequestMapping("Multiple Record ShowStudent List")
public List multipleRecord() {
        Session ss = sf.openSession();
        Criteria criteria = ss.~~createCriteria~~(Student.class);
        List<Student> StudentList = criteria.list();
        System.out.println(StudentList);

        return StudentList;
}


@GetMapping("One Record ShowStudent List/{id}")
public Student getStudent(@PathVariable int id) {

        Session ss = sf.openSession();
        stud = ss.load(Student.class, id);
        System.out.println(stud);

        Student student = stud;
        for (Student ss1 : al) {
                if (ss1.id == id) {
                        student = ss1;
                }
        }
        System.out.println("One Record Show Successfully Done... " + student);
        return student;
}

@PostMapping("insertStudent")
Student addStudent(@RequestBody Student student) {

        Session ss = sf.openSession();
        Transaction tx = ss.beginTransaction();

        ss.save(student);
        System.out.println("Record Inserted Successfully Done... " + student);
        tx.commit();
        return student;
}
@PostMapping("multipleInsertRecord")
public List<Employee> multipleinsert(@RequestBody List<Employee> list) {
    for (Employee employee2 : list) {
        Session ss=sf.openSession();
        Transaction tt=ss.beginTransaction();
        ss.save(employee2);
        tt.commit();
    }
    return list;
}

@PutMapping("updateStudent")
Student updateStudent(@RequestBody Student student) {

    Session ss = sf.openSession();
    Transaction tx = ss.beginTransaction();

    ss.update(student);
    System.out.println("Record Updated Successfully Done... " + student);
    tx.commit();
    return student;
}

@PutMapping("multipleupdateRecord")
public List<Employee> multipleupdate(@RequestBody List<Employee> list) {
    for (Employee employee2 : list) {
            Session ss=sf.openSession();
            Transaction tt=ss.beginTransaction();
            ss.update(employee2);
            tt.commit();
        }
        return list;
    }

@DeleteMapping("deleteStudent/{id}")
public int deleteStudent(@PathVariable int id) {

    Session ss = sf.openSession();
    Transaction tx = ss.beginTransaction();

    stud = ss.load(Student.class, id);
    System.out.println(stud);

    ss.delete(stud);

    System.out.println("Record Deleted Successfully Done... " + id);
    tx.commit();
    return id;
    }
}
```