

One To One

Means one employee can have one address

1) Create Spring Boot Project with dependencies
spring web h2dilect and spring JPA

2) Create a Entity class as new class ---->Employee

```
import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.OneToOne;
import jakarta.persistence.Table;
import jakarta.transaction.Transactional;

@Entity
@Transactional
@Table(name="emp")
public class Employee {
    @Id
    @GeneratedValue (strategy = GenerationType.AUTO)

    @Column(name="emp_id")

    private Long empid;
    private String empname;
    private Integer empage;

    @OneToOne( cascade = CascadeType.ALL )
    @JoinColumn(name="fk_add_id")
    private Address address;

    public Long getEmpid() {
        return empid;
    }

    public void setEmpid(Long empid) {
        this.empid = empid;
    }

    public String getEmpname() {
        return empname;
    }

    public void setEmpname(String empname) {
```

```

        this.empname = empname;
    }

    public Integer getEmpage() {
        return empage;
    }

    public void setEmpage(Integer empage) {
        this.empage = empage;
    }

    public Address getAddress() {
        return address;
    }

    public void setAddress(Address address) {
        this.address = address;
    }
}

```

3) Create EmployeeRepository interface:- new----->Interface-----
>EmployeeRepository

```

import org.springframework.data.jpa.repository.JpaRepository;

public interface EmployeeRepository extends
JpaRepository<Employee, Long> {

}

```

4) Create a new class Address and write the following code

```

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import jakarta.transaction.Transactional;

@Entity
@Transactional
@Table(name="address")
public class Address {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "add_id")

```

```

    private Long addressId;
    private String City;
    private String addressType;

    public Long getAddressId() {
        return addressId;
    }
    public void setAddressId(Long addressId) {
        this.addressId = addressId;
    }
    public String getCity() {
        return City;
    }
    public void setCity(String city) {
        City = city;
    }
    public String getAddressType() {
        return addressType;
    }
    public void setAddressType(String addressType) {
        this.addressType = addressType;
    }
}

```

5) Create a employeecontroller class as ----->new class->EmpController

```

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

@RestController

public class EmpController {

    @Autowired
    private EmployeeRepository employeeRepository;

    @GetMapping("/saveemp")
    public List<Employee> getAllProducts() {
        return employeeRepository.findAll();
    }
}

```

```

        @PostMapping("/saveemp")
        public ResponseEntity<String> saveEmployees
        (@RequestBody Employee empdata)
        {

            employeeRepository.save(empdata);
            return ResponseEntity.ok("Data saved");
        }

    }

```

in the application.properties write the following code

```

spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=password

```

run the application
 create json and use link localhost:8080/saveemp
 run post url
 and provide json as

```

{
    "empid":1,
    "empname":"abc",
    "empage":3,
    "address":
    {
        "addressId":2,
        "City":"Dhule",
        "addressType":"dd"
    },
}

```