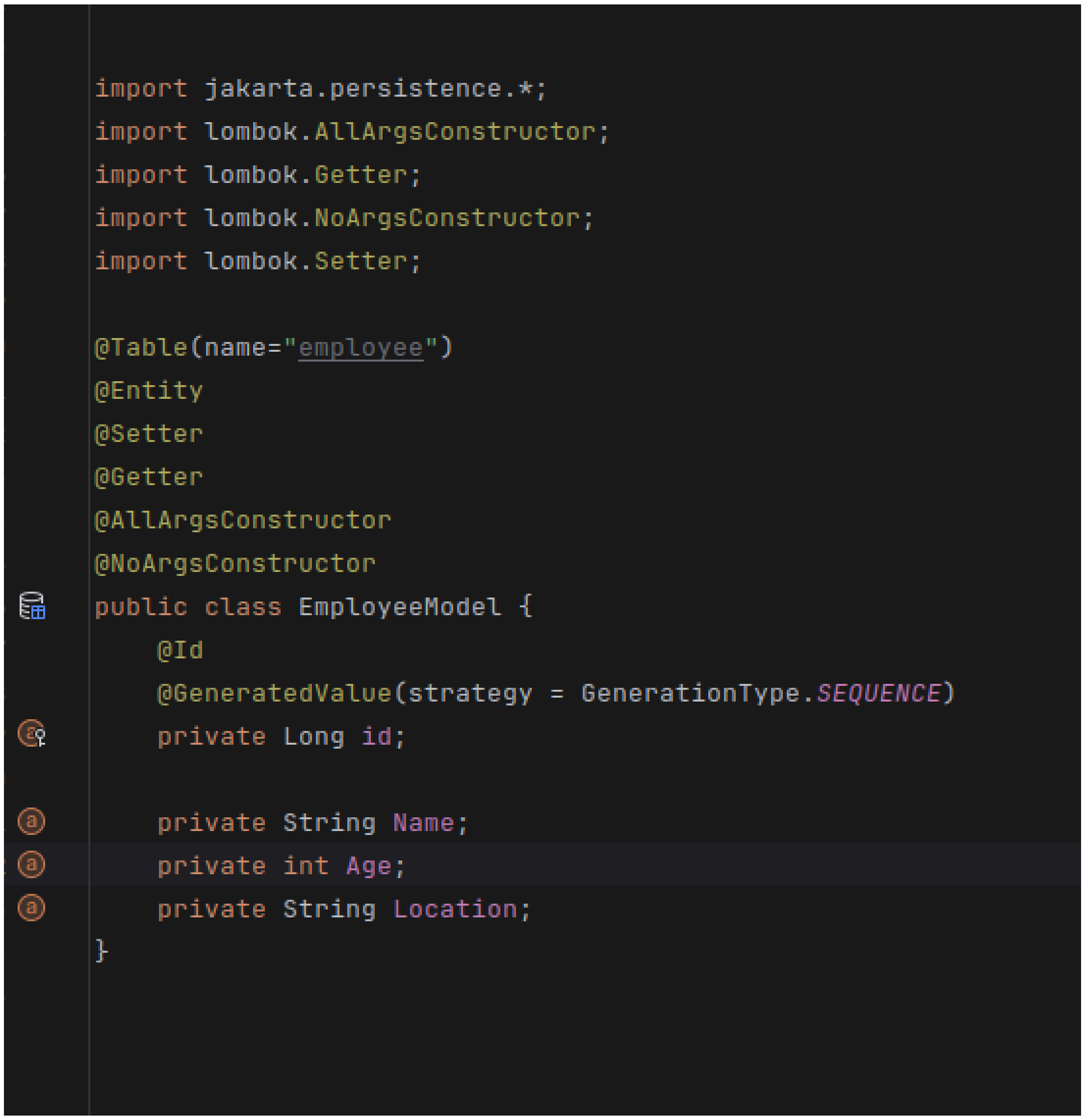
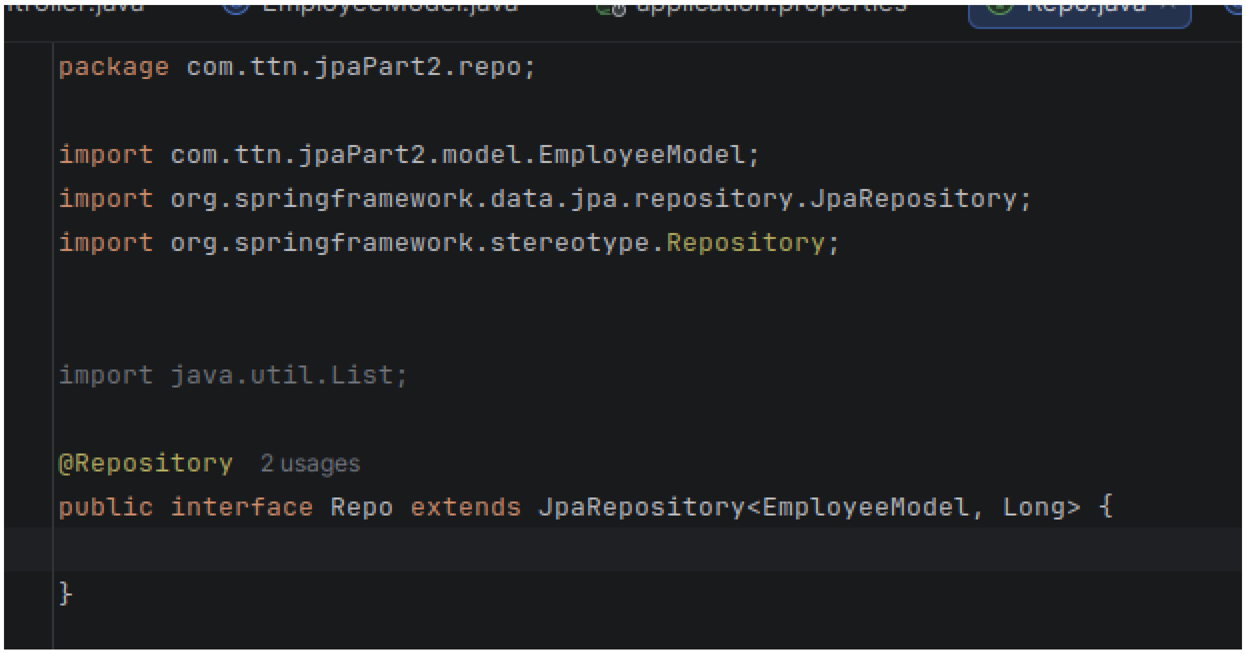
Q)1 Create an Employee Entity which contains the following fields: Name, Id, Age, Location



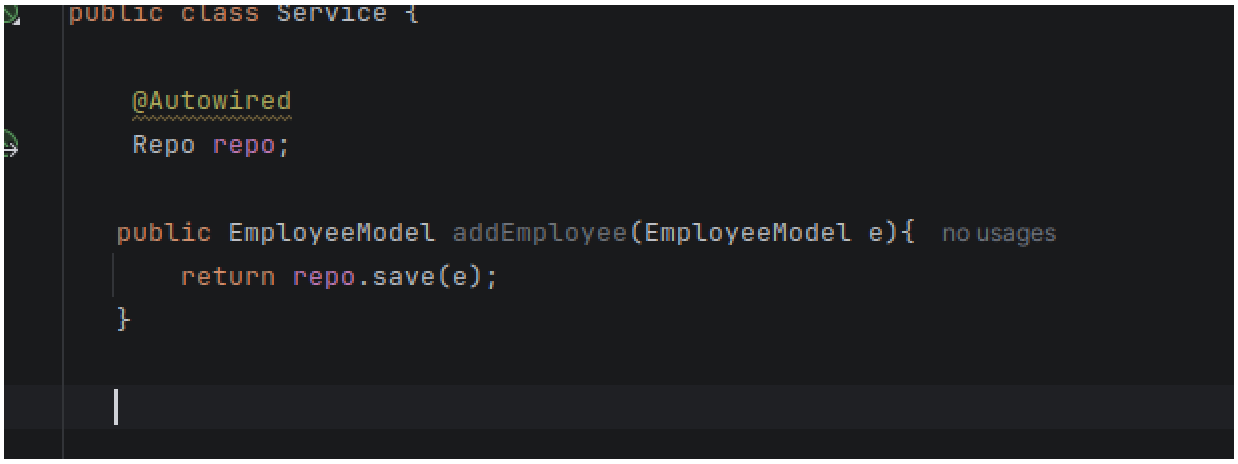
```java
import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

@Table(name="employee")
@Entity
@Setter
@Getter
@AllArgsConstructor
@NoArgsConstructor
public class EmployeeModel {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private Long id;

    private String Name;
    private int Age;
    private String Location;
}
```

2) Set up EmployeeRepository with Spring Data JPA



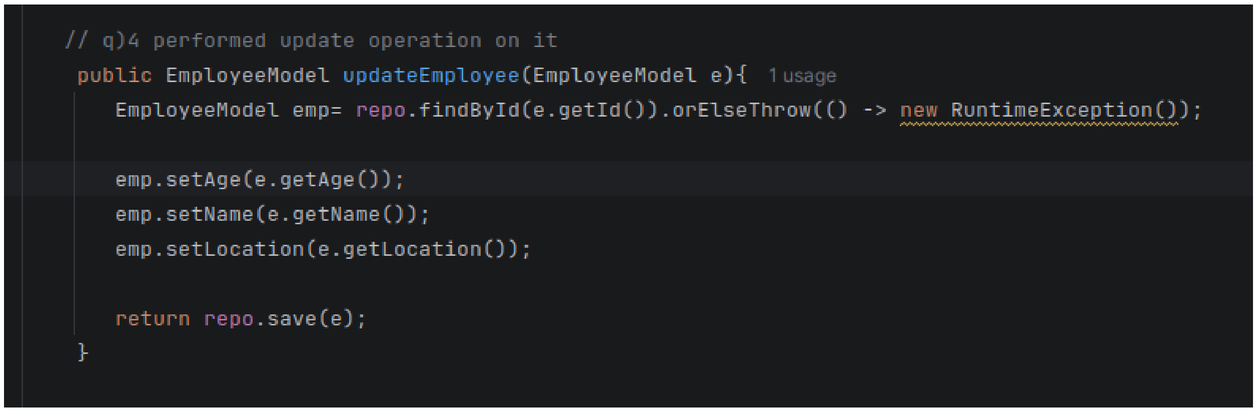
```
package com.ttn.jpaPart2.repo;

import com.ttn.jpaPart2.model.EmployeeModel;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;


import java.util.List;

@Repository  2 usages
public interface Repo extends JpaRepository<EmployeeModel, Long> {

}
```

3) Perform Create Operation on Entity using Spring Data JPA



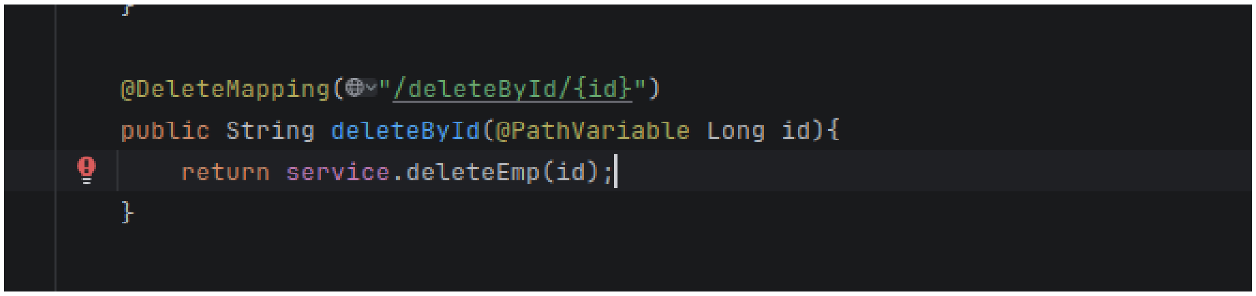
```
public class Service {

    @Autowired
    Repo repo;

   public EmployeeModel addEmployee(EmployeeModel e){  no usages
       return repo.save(e);
   }

```

4) Perform Update Operation on Entity using Spring Data JPA

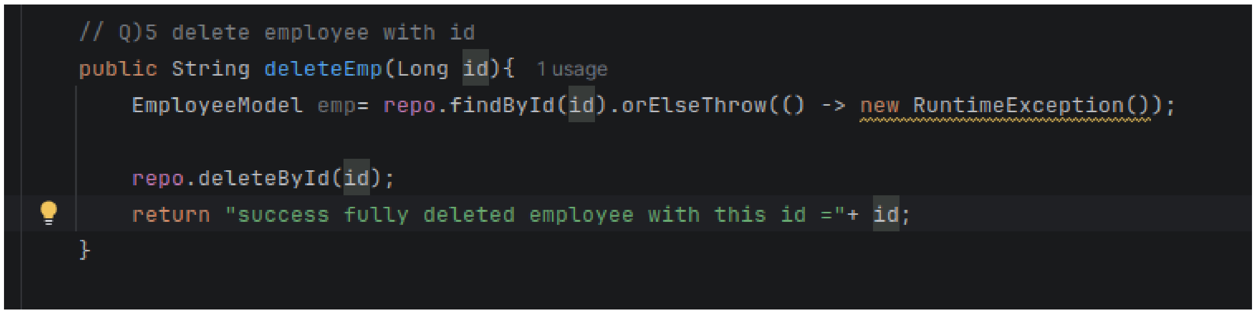


```
// q)4 performed update operation on it
public EmployeeModel updateEmployee(EmployeeModel e){  1 usage
    EmployeeModel emp= repo.findById(e.getId()).orElseThrow(() -> new RuntimeException());

    emp.setAge(e.getAge());
    emp.setName(e.getName());
    emp.setLocation(e.getLocation());

    return repo.save(e);
}
```

5)
Perform Delete Operation on Entity using Spring Data JPA

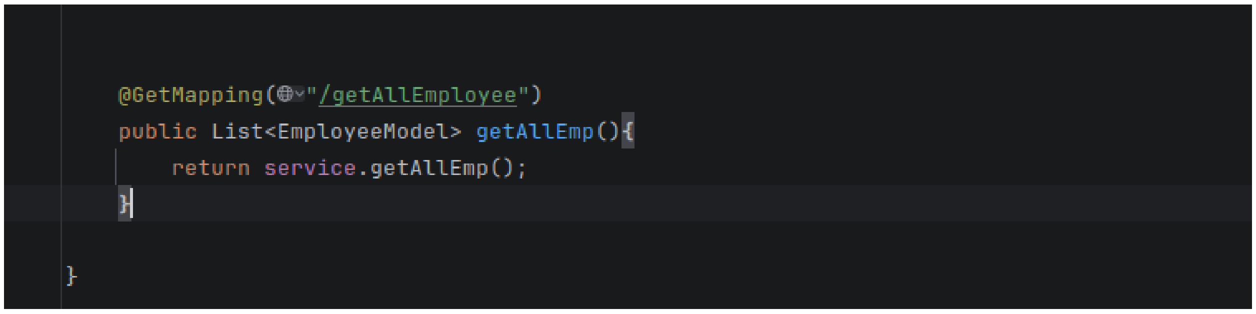


```
    @DeleteMapping("/deleteById/{id}")
    public String deleteById(@PathVariable Long id){
        return service.deleteEmp(id);
    }
```

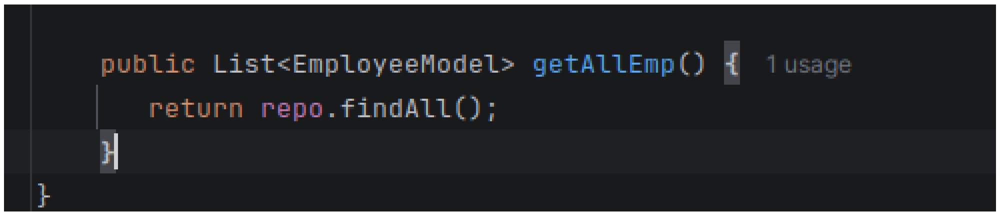


```
// Q)5 delete employee with id
public String deleteEmp(Long id){  1 usage
    EmployeeModel emp= repo.findById(id).orElseThrow(() -> new RuntimeException());

    repo.deleteById(id);
    return "success fully deleted employee with this id ="+ id;
}
```

6) Perform Read Operation on Entity using Spring Data JPA

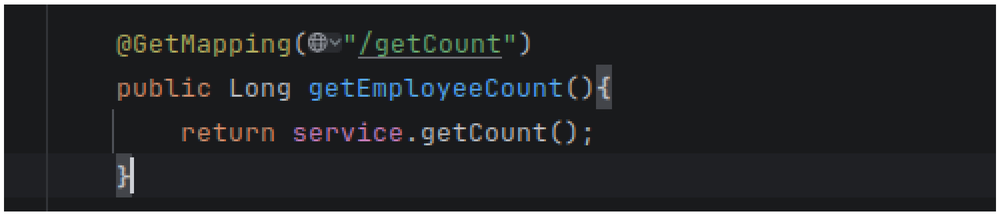


```
    @GetMapping("/getAllEmployee")
    public List<EmployeeModel> getAllEmp(){
        return service.getAllEmp();
    }

}
```

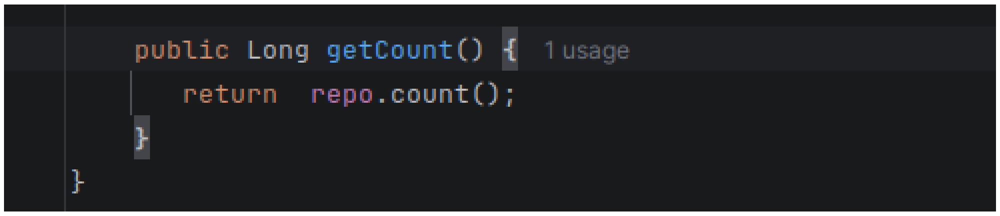


```
    public List<EmployeeModel> getAllEmp() {  1 usage
        return repo.findAll();
    }
}
```

7) Get the total count of the number of Employees

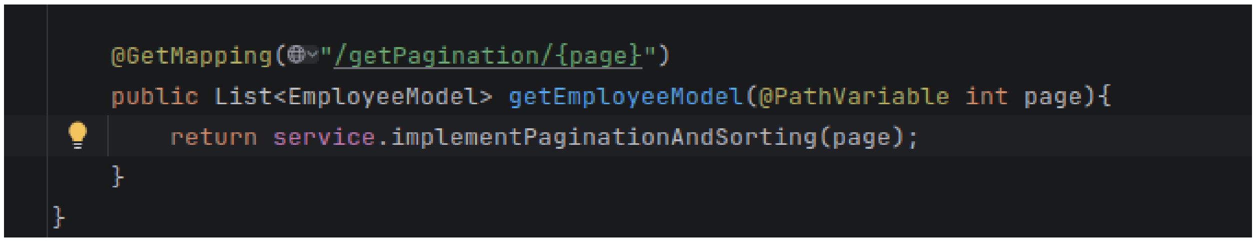


```
    @GetMapping("/getCount")
    public Long getEmployeeCount(){
        return service.getCount();
    }
```

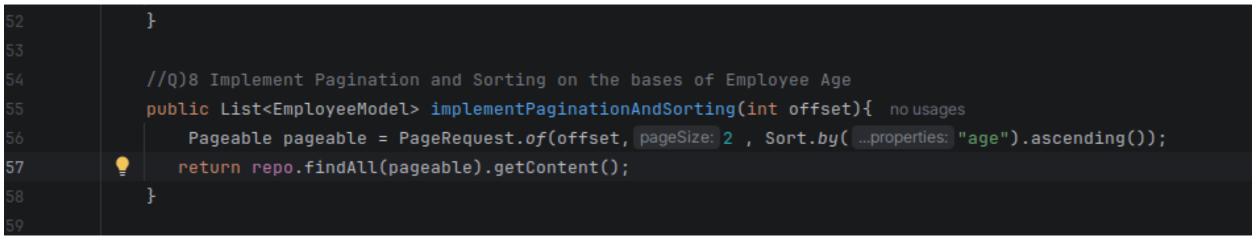


```
    public Long getCount() {  1 usage
        return  repo.count();
    }
}
```

8) Implement Pagination and Sorting on the bases of Employee Age

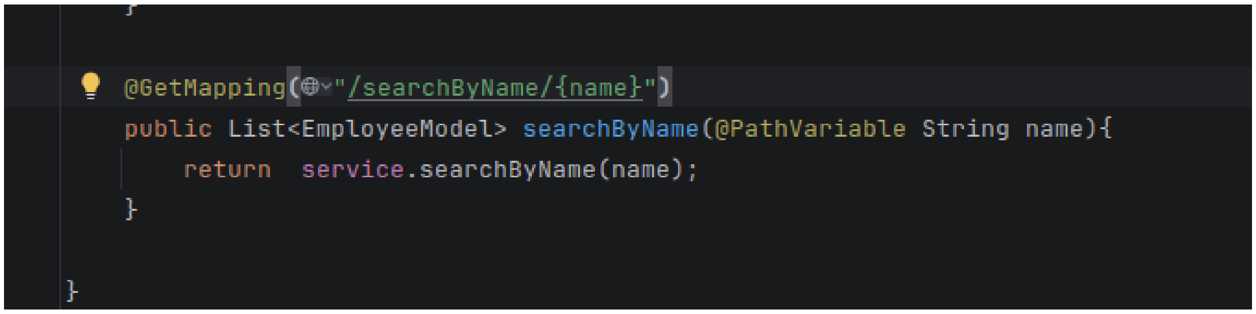


```
    @GetMapping("/getPagination/{page}")
    public List<EmployeeModel> getEmployeeModel(@PathVariable int page){
        return service.implementPaginationAndSorting(page);
    }
}
```

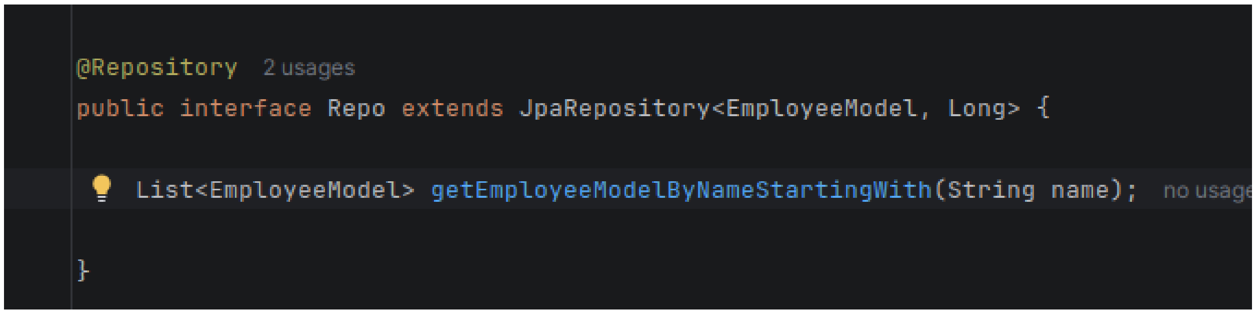


```
    }

    //Q)8 Implement Pagination and Sorting on the bases of Employee Age
    public List<EmployeeModel> implementPaginationAndSorting(int offset){  no usages
        Pageable pageable = PageRequest.of(offset, pageSize: 2 , Sort.by( ...properties: "age").ascending());
        return repo.findAll(pageable).getContent();
    }

```

9) Create and use finder to find Employee by Name

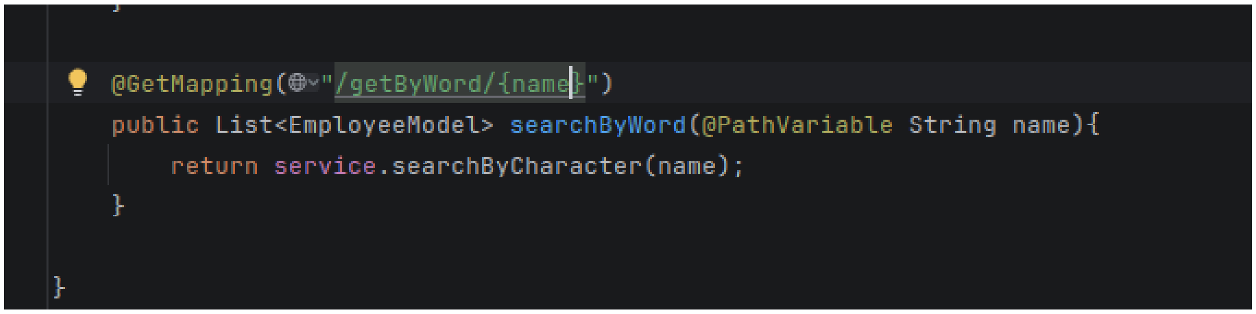


```
    }

    @GetMapping("/searchByName/{name}")
    public List<EmployeeModel> searchByName(@PathVariable String name){
        return  service.searchByName(name);
    }

}
```

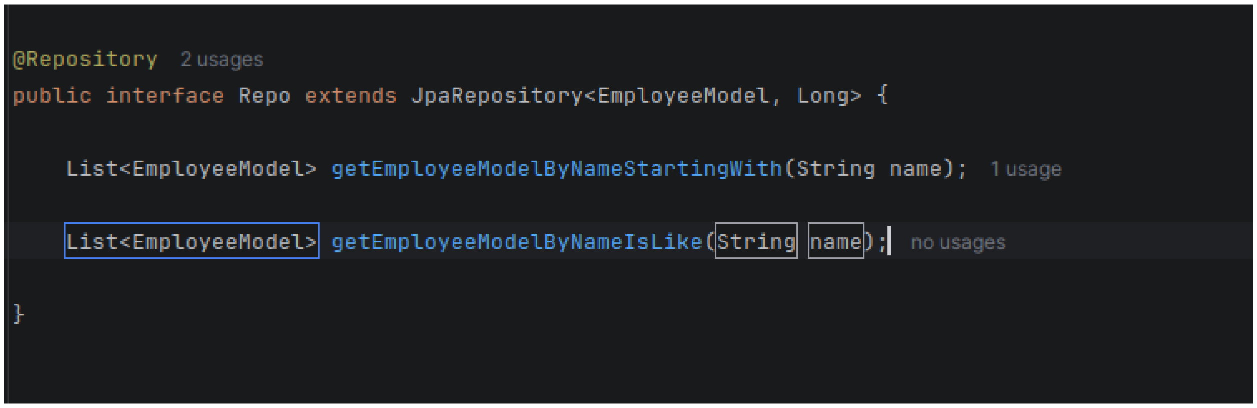


```
@Repository  2 usages
public interface Repo extends JpaRepository<EmployeeModel, Long> {

    List<EmployeeModel> getEmployeeModelByNameStartingWith(String name);  no usage

}
```

10) Create and use finder to find Employees starting with A character

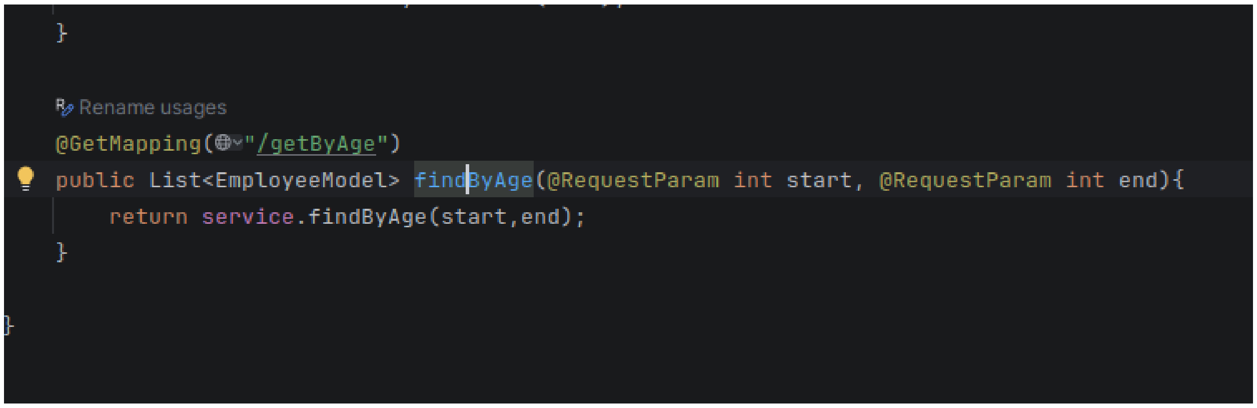


```
    @GetMapping("/getByWord/{name}")
    public List<EmployeeModel> searchByWord(@PathVariable String name){
        return service.searchByCharacter(name);
    }

}
```

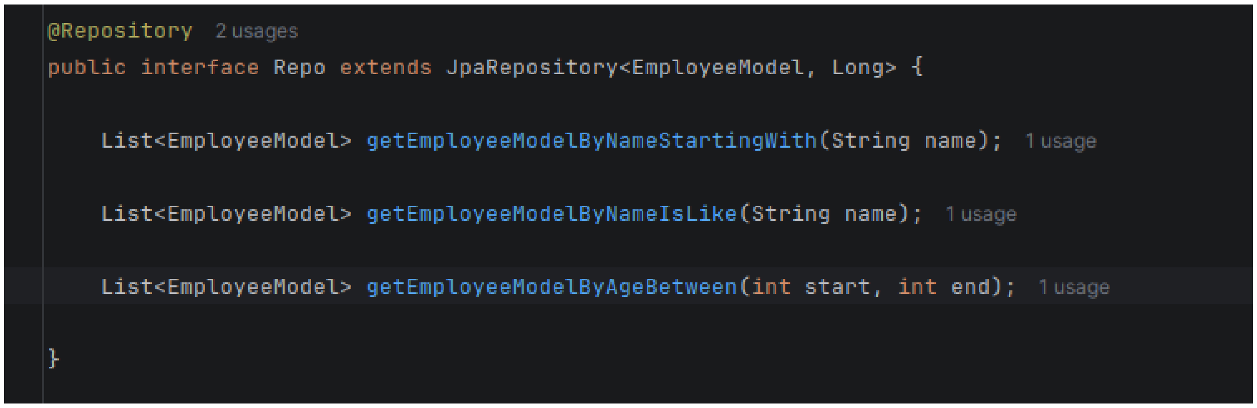


```
@Repository  2 usages
public interface Repo extends JpaRepository<EmployeeModel, Long> {

    List<EmployeeModel> getEmployeeModelByNameStartingWith(String name);  1 usage

    List<EmployeeModel> getEmployeeModelByNameIsLike(String name);  no usages

}
```

11) Create and use finder to find Employees Between the age of 28 to 32



```
    }

    Rename usages
    @GetMapping("/getByAge")
    public List<EmployeeModel> findByAge(@RequestParam int start, @RequestParam int end){
        return service.findByAge(start,end);
    }

}
```



```
@Repository  2 usages
public interface Repo extends JpaRepository<EmployeeModel, Long> {

    List<EmployeeModel> getEmployeeModelByNameStartingWith(String name);  1 usage

    List<EmployeeModel> getEmployeeModelByNameIsLike(String name);  1 usage

    List<EmployeeModel> getEmployeeModelByAgeBetween(int start, int end);  1 usage

}
```