

Telecom-service

Entity

```
package com.springtelecom.entities;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import lombok.Data;

@Data
@Entity
public class Telecom {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer companyId;

    private String companyName;

    private String companyCEO;
```

```
    private String companyServices;

    private Integer companyEstablishedYear;

}
```

Telecom Controller

```
package com.springtelecom.controllers;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
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.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.springtelecom.entities.Telecom;
import com.springtelecom.services.TelecomService;

@RestController
@RequestMapping("api/v1")
public class TeelcomController {
```

```

@Autowired
TelecomService service;

@GetMapping("/telecoms/{companyName}")
public Optional<Telecom> getByName(String companyName){
    return service.getByName(companyName);
}

@GetMapping("/telecoms")
public Iterable<Telecom> getAllCompanies()
{
    return service.getAllCompanies();
}

@PostMapping("/telecoms")
public void addCompany(@RequestBody Telecom telecom) {
    service.addCompany(telecom);
}
}

```

Telecom Service

```
package com.springtelecom.services;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.springtelecom.entities.Telecom;
import com.springtelecom.repositories.TelecomRepository;
```

```
@Service
```

```
public class TelecomService {
```

```
    @Autowired
```

```
    TelecomRepository repository;
```

```
    public Optional<Telecom> getByName(String companyName){
```

```
        return repository.findByCompanyName(companyName);
```

```
    }
```

```
    public Iterable<Telecom> getAllCompanies() {
```

```

        // TODO Auto-generated method stub
        return repository.findAll();
    }

    public void addCompany(Telecom telecom) {
        // TODO Auto-generated method stub
        repository.save(telecom);
    }
}

```

Telecom Repository

```

package com.springtelecom.repositories;

import java.util.Optional;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import com.springtelecom.entities.Telecom;

```

@Repository

```
public interface TelecomRepository extends CrudRepository<Telecom, Integer>{
```

```
    Optional<Telecom> findByCompanyName(String companyName);  
}
```

Outputs

	company_id	companyceo	company_established_year	company_name	company_services
*	NULL	NULL	NULL	NULL	NULL

HTTP <http://localhost:8081/api/v1/telecoms>

POST <http://localhost:8081/api/v1/telecoms>

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON

```
1 {
2   "companyName": "Vodafone VI",
3   "companyCEO": "Chandran",
4   "companyServices": "Prepaid and PostPaid",
5   "companyEstablishedYear": 1999
6 }
7
8
```

Body Cookies Headers (4) Test Results 200 OK 240 ms

HTTP <http://localhost:8081/api/v1/telecoms>

GET <http://localhost:8081/api/v1/telecoms>

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary

This request does not have a body

Body Cookies Headers (5) Test Results 200 OK 41 ms 304 B Save

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "companyId": 1,
4     "companyName": "Vodafone VI",
5     "companyCEO": "Chandran",
6     "companyServices": "Prepaid and PostPaid",
7     "companyEstablishedYear": 1999
8   }
9 ]
```

Activate Windows

Customer Entity

```
package com.springcustomerservice.entities;
```

```
import jakarta.persistence.Entity;
```

```
import jakarta.persistence.GeneratedValue;
```

```
import jakarta.persistence.GenerationType;
```

```
import jakarta.persistence.Id;
```

```
import jakarta.persistence.ManyToMany;
```

```
import lombok.Data;
```

```
@Entity
```

```
@Data
```

```
public class Customer {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Integer customerId;
```

```
    private String customerName;
```

```
    private String customerSubscription;
```

```
    private String customerTelecomCompany;
```



```
}
```

Customer Controller

```
package com.springcustomerservice.controllers;
```

```
import java.util.Optional;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
import com.springcustomerservice.entities.Customer;
```

```
import com.springcustomerservice.services.CustomerService;
```

```
@RestController
```

```
@RequestMapping("api/v1")
```

```
public class CustomerController {
```

```
    @Autowired
```

```
    CustomerService customerService;
```

```
    @GetMapping("/customers/name/{name}")
```

```

        Optional<Customer> getByCustomerName(String name){
            return customerService.getByCustomerName(name);
        }

        @GetMapping("/customers/subscriptionpaln/{subscriptionpaln}"
    )
        Iterable<Customer> getByCustomerBySubscription(String
subscriptionpaln){
            return
customerService.getByCustomerBySubscription(subscriptionpaln
);
        }
    }
}

```

Customer Service

```

package com.springcustomerservice.services;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

import com.springcustomerservice.entities.Customer;
import com.springcustomerservice.repositories.CustomerRepository;

```

```

@Service
public class CustomerService {

    @Autowired
    RestTemplate restTemplate;

    @Autowired
    CustomerRepository customerRepository;

    public Optional<Customer> getByCustomerName(String name){
        return customerRepository.findByCustomerName(name);
    }

    public Iterable<Customer> getByCustomerBySubscription(String
subscriptionpaIn){
        return
customerRepository.findByCustomerSubscription(subscriptionpa
ln);
    }

}

```

Customer Repo

```

package com.springcustomerservice.repositories;

import java.util.Optional;

```

```
import org.springframework.data.repository.CrudRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.springcustomerservice.entities.Customer;
```

```
@Repository
```

```
public interface CustomerRepository extends  
CrudRepository<Customer, Integer> {
```

```
    Optional<Customer> findByCustomerName(String customerName);
```

```
    Iterable<Customer> findByCustomerSubscription(String  
customerSubscription);
```

```
}
```

```
App config
```

```
package com.springcustomerservice.config;
```

```
import org.springframework.context.annotation.Bean;
```

```
import org.springframework.context.annotation.Configuration;
```

```
import org.springframework.web.client.RestTemplate;
```

```
@Configuration
```

```
public class AppConfig {
```

```
    @Bean
```

```
public RestTemplate getRestTemplate() {  
    return new RestTemplate();  
}  
}
```

The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons for file operations, execution, and search. Below the toolbar, a SQL query is entered in a text area: `1 • SELECT * FROM customer.customer;`. Below the query area, there is a section for the results. It includes a 'Result Grid' tab, a 'Filter Rows' input field, and an 'Edit' button. The results are displayed in a table with four columns: `customer_id`, `customer_name`, `customer_subscription`, and `customer_telecom_company`. The first row of data shows all four fields as `NULL`.

	customer_id	customer_name	customer_subscription	customer_telecom_company
*	NULL	NULL	NULL	NULL