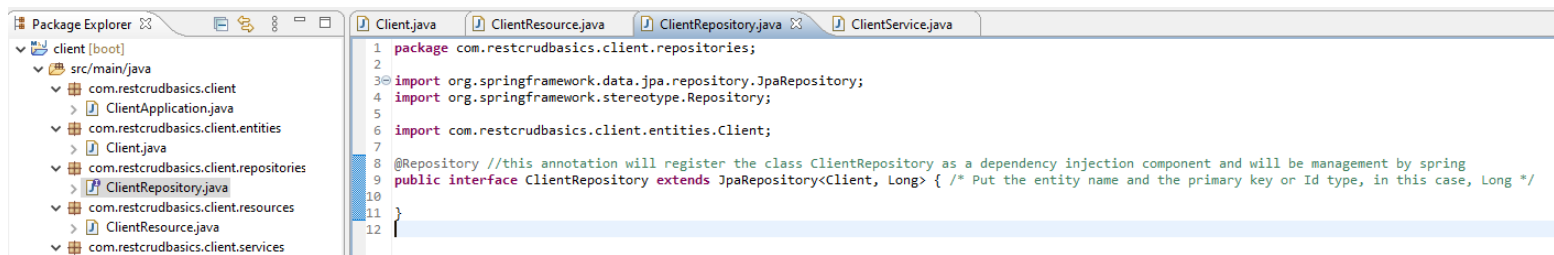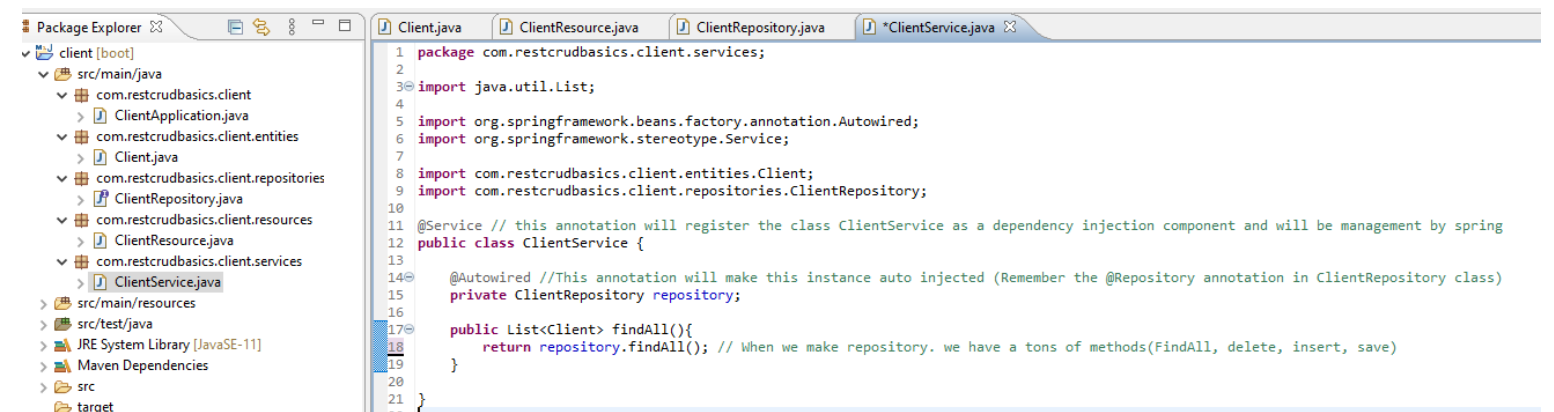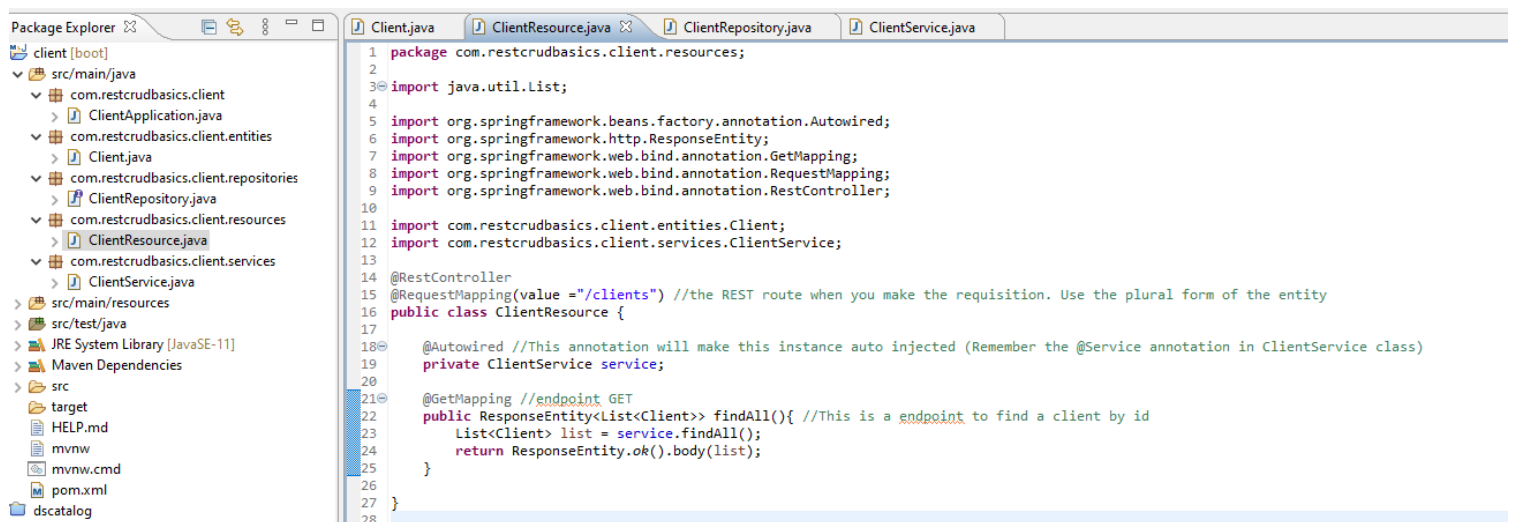- First, make the dependency injection in the ClientRepository class with the specific annotation @Repository

```
Package Explorer                    Client.java    ClientResource.java    ClientRepository.java    ClientService.java
client [boot]                      1  package com.restcrudbasics.client.repositories;
  src/main/java                    2
    com.restcrudbasics.client       3  import org.springframework.data.jpa.repository.JpaRepository;
      ClientApplication.java        4  import org.springframework.stereotype.Repository;
    com.restcrudbasics.client.entities    5
      Client.java                   6  import com.restcrudbasics.client.entities.Client;
    com.restcrudbasics.client.repositories    7
      ClientRepository.java         8  @Repository //this annotation will register the class ClientRepository as a dependency injection component and will be management by spring
    com.restcrudbasics.client.resources    9  public interface ClientRepository extends JpaRepository<Client, Long> { /* Put the entity name and the primary key or Id type, in this case, Long */
      ClientResource.java          10
    com.restcrudbasics.client.services   11  }
                                   12
```
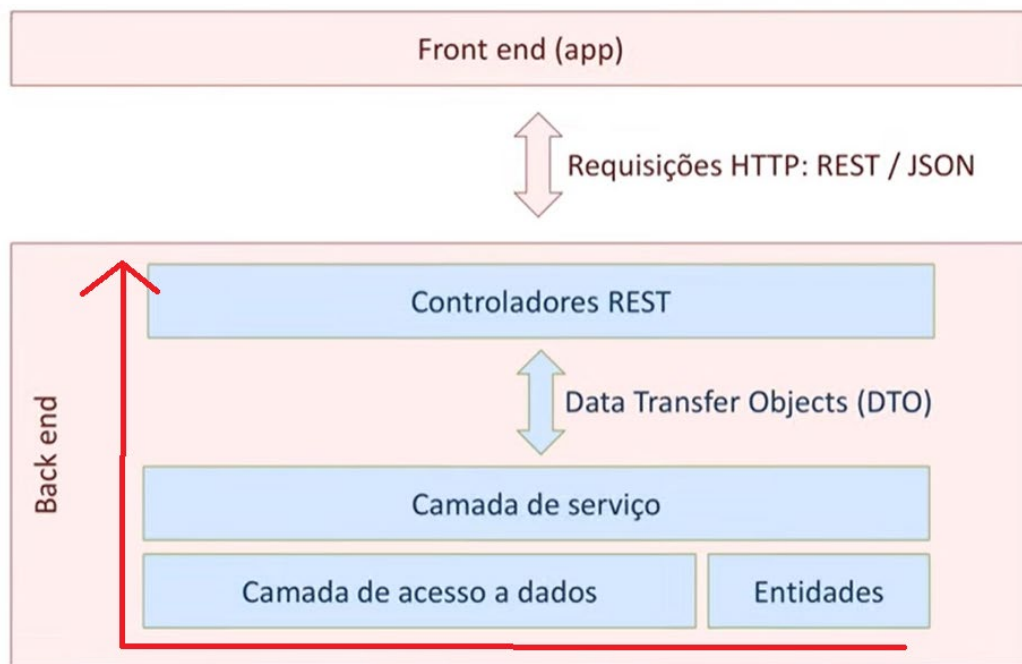
- Now, the ClientRepository class is a dependency injection componente management by spring
- Now, implemente the service Layer

```
Package Explorer                    Client.java    ClientResource.java    ClientRepository.java    *ClientService.java
client [boot]                      1  package com.restcrudbasics.client.services;
  src/main/java                    2
    com.restcrudbasics.client       3  import java.util.List;
      ClientApplication.java        4
    com.restcrudbasics.client.entities    5  import org.springframework.beans.factory.annotation.Autowired;
      Client.java                   6  import org.springframework.stereotype.Service;
    com.restcrudbasics.client.repositories    7
      ClientRepository.java         8  import com.restcrudbasics.client.entities.Client;
    com.restcrudbasics.client.resources    9  import com.restcrudbasics.client.repositories.ClientRepository;
      ClientResource.java          10
    com.restcrudbasics.client.services   11  @Service // this annotation will register the class ClientService as a dependency injection component and will be management by spring
      ClientService.java           12  public class ClientService {
  src/main/resources               13
  src/test/java                    14     @Autowired //This annotation will make this instance auto injected (Remember the @Repository annotation in ClientRepository class)
  JRE System Library [JavaSE-11]   15     private ClientRepository repository;
  Maven Dependencies               16
  src                              17     public List<Client> findAll(){
  target                           18        return repository.findAll(); // When we make repository. we have a tons of methods(FindAll, delete, insert, save)
                                   19     }
                                   20
                                   21  }
```

- Now, make the depency injection in the ClientResource

```
Package Explorer                    Client.java    ClientResource.java    ClientRepository.java    ClientService.java
client [boot]                      1  package com.restcrudbasics.client.resources;
  src/main/java                    2
    com.restcrudbasics.client       3  import java.util.List;
      ClientApplication.java        4
    com.restcrudbasics.client.entities    5  import org.springframework.beans.factory.annotation.Autowired;
      Client.java                   6  import org.springframework.http.ResponseEntity;
    com.restcrudbasics.client.repositories    7  import org.springframework.web.bind.annotation.GetMapping;
      ClientRepository.java         8  import org.springframework.web.bind.annotation.RequestMapping;
    com.restcrudbasics.client.resources    9  import org.springframework.web.bind.annotation.RestController;
      ClientResource.java          10
    com.restcrudbasics.client.services   11  import com.restcrudbasics.client.entities.Client;
      ClientService.java           12  import com.restcrudbasics.client.services.ClientService;
  src/main/resources               13
  src/test/java                    14  @RestController
  JRE System Library [JavaSE-11]   15  @RequestMapping(value ="/clients") //the REST route when you make the requisition. Use the plural form of the entity
  Maven Dependencies               16  public class ClientResource {
  src                              17
  target                           18     @Autowired //This annotation will make this instance auto injected (Remember the @Service annotation in ClientService class)
  HELP.md                          19     private ClientService service;
  mvnw                             20
  mvnw.cmd                         21     @GetMapping //endpoint GET
  pom.xml                          22     public ResponseEntity<List<Client>> findAll(){ //This is a endpoint to find a client by id
  dscatalog                        23        List<Client> list = service.findAll();
                                   24        return ResponseEntity.ok().body(list);
                                   25     }
                                   26
                                   27  }
                                   28
```

- With this, we are following the architecture layer above

Front end (app)		
↕ Requisições HTTP: REST / JSON		
Back end	Controladores REST	
	↕ Data Transfer Objects (DTO)	
	Camada de serviço	
	Camada de acesso a dados	Entidades

- Now, is time for the REST requisition test