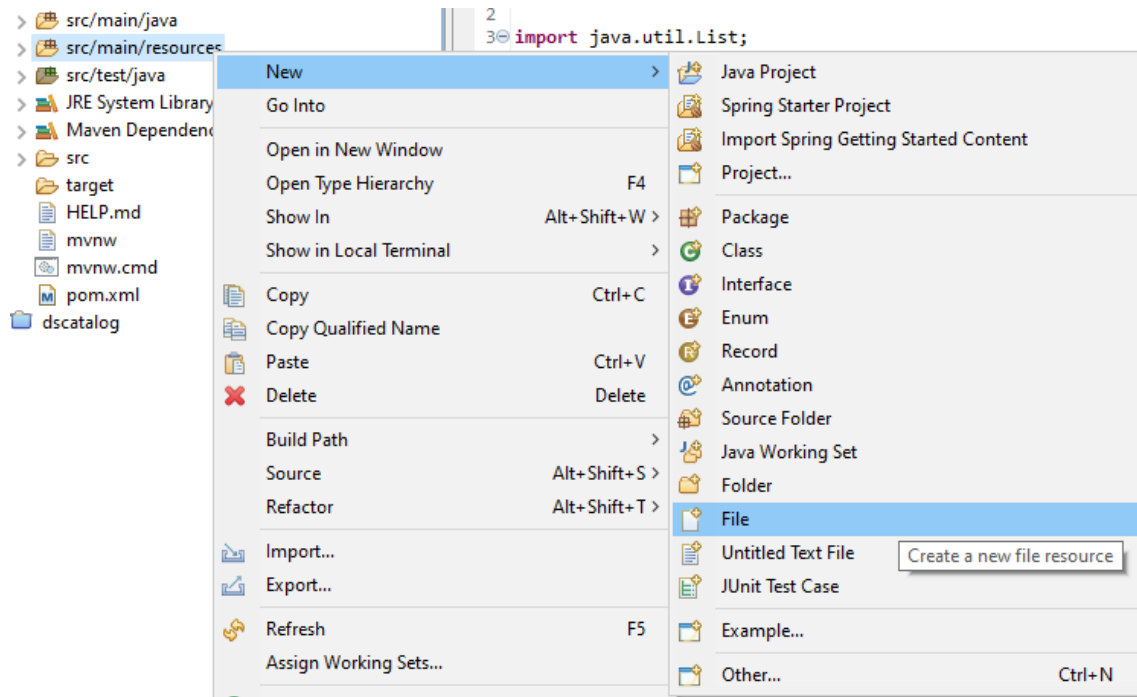
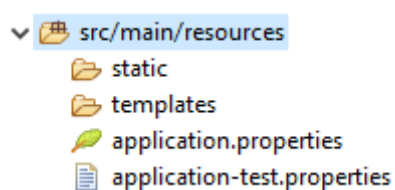
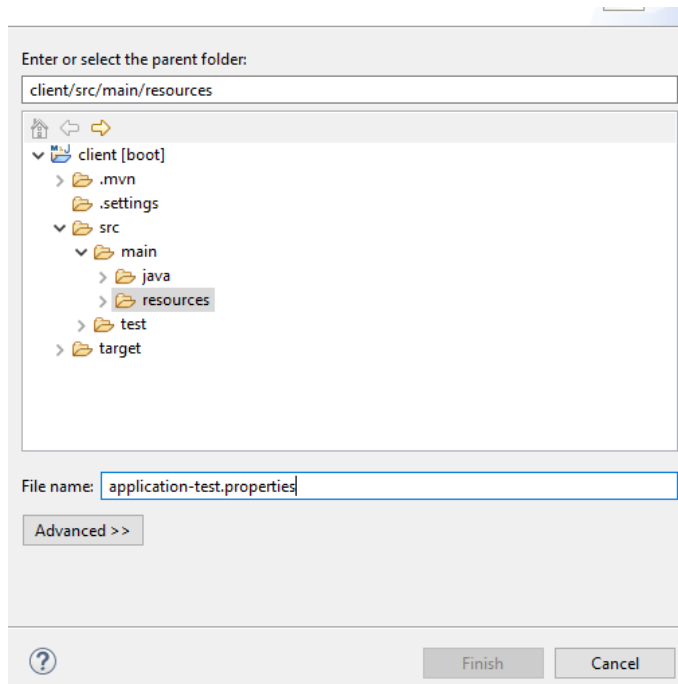


- Now, we gonna make the H2 database configuration properties
- Go in Resources and make a new file



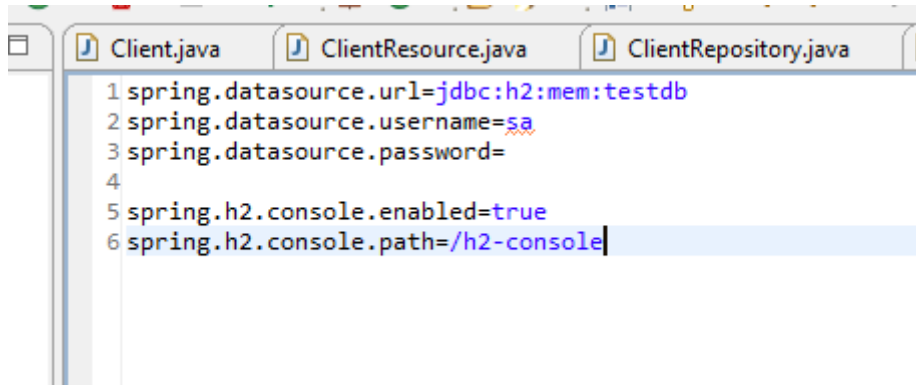
- Make a file with the name application-test.properties



- Now put the H2 database configuration

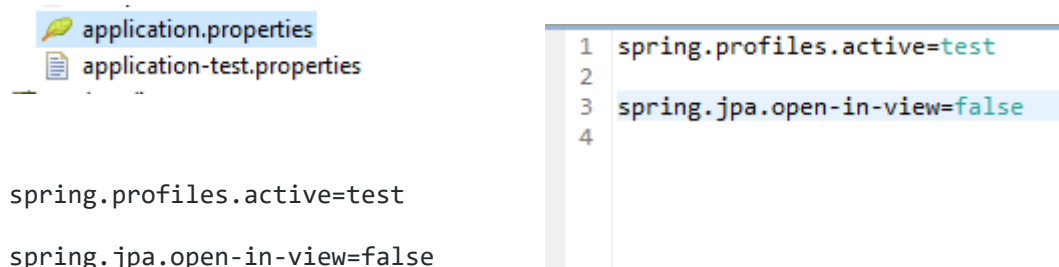
```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.username=sa
spring.datasource.password=
```

```
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console
```



```
1 spring.datasource.url=jdbc:h2:mem:testdb
2 spring.datasource.username=sa
3 spring.datasource.password=
4
5 spring.h2.console.enabled=true
6 spring.h2.console.path=/h2-console
```

- Now, go to the file application.properties and put the configuration below

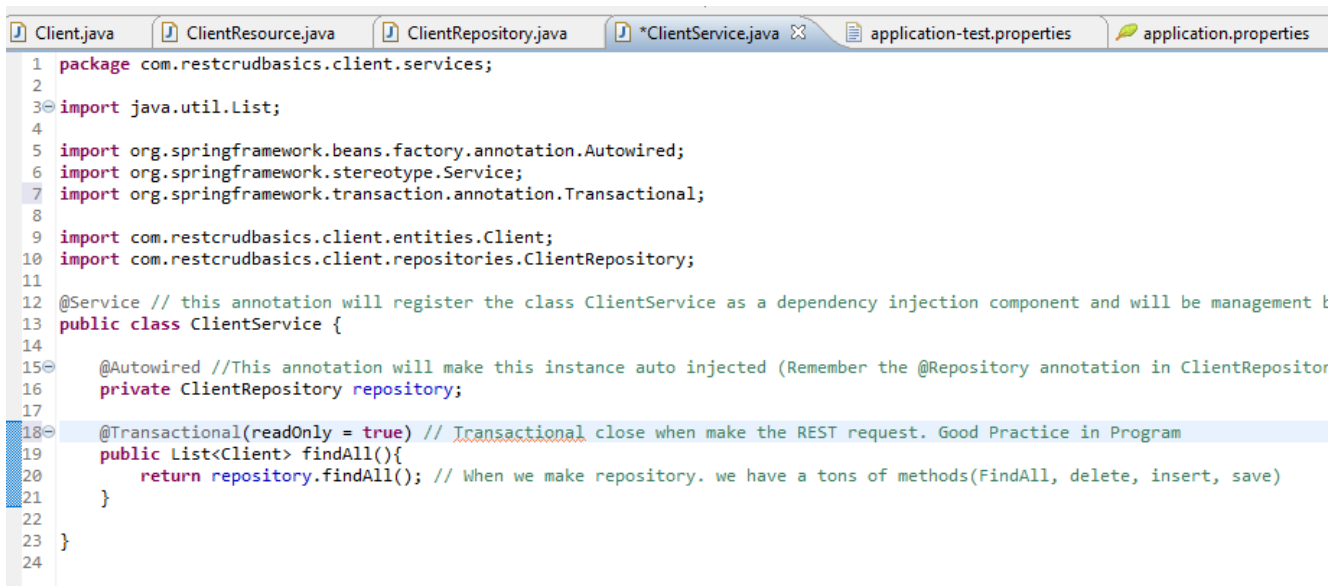


```
1 spring.profiles.active=test
2
3 spring.jpa.open-in-view=false
4
```

```
spring.profiles.active=test

spring.jpa.open-in-view=false
```

- This is a profile test. We gonna use the H2 database as database to make test's. Now, enter the database H2. In a application, we use 3 profiles - Test, Development and Production.
Test - Normally, use H2 as database.
Development - The same database used in Production or client. Normally, PostgreSQL
Production - The final test.
- The Property view = false is to close the transactions (Service, Repository) to not be open when make a REST request.
- To make this, use the annotation @Transactional in the service method



```
1 package com.restcrudbasics.client.services;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7 import org.springframework.transaction.annotation.Transactional;
8
9 import com.restcrudbasics.client.entities.Client;
10 import com.restcrudbasics.client.repositories.ClientRepository;
11
12 @Service // this annotation will register the class ClientService as a dependency injection component and will be management t
13 public class ClientService {
14
15     @Autowired //This annotation will make this instance auto injected (Remember the @Repository annotation in ClientRepository
16     private ClientRepository repository;
17
18     @Transactional(readOnly = true) // Transactional close when make the REST request. Good Practice in Program
19     public List<Client> findAll(){
20         return repository.findAll(); // When we make repository. we have a tons of methods(FindAll, delete, insert, save)
21     }
22
23 }
24
```