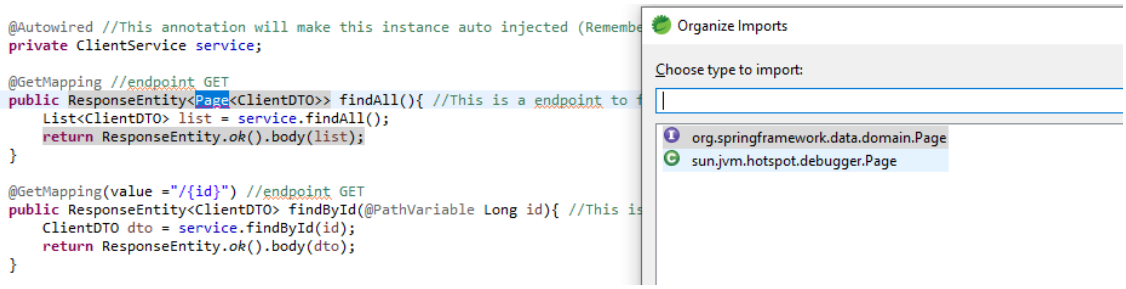
- Now, we will implement the pagination.
- First, we will modify the return of the HTTP request in the resource entity GET from List to Page.




- When we make a pagination, we can pass some attributes to the request like this.

```
@RequestParam(value = "page", defaultValue = "0") Integer page,
@RequestParam(value = "linesPerPage", defaultValue = "12") Integer linesPerPage,
@RequestParam(value = "orderBy", defaultValue = "name") String orderBy,
@RequestParam(value = "direction", defaultValue = "DESC") String direction
){
```

- The annotation @RequestParam is optional in the request. If we want a parameter obligate, we put the annotation @PathVariable.
- Page = The first exhibition page. linesPerPage = Number of pages we want. orderBy = The form we want to order. Direction = if will be Descending or ascending.
- Put the annotation in the body of the Getall method

```
@GetMapping //endpoint GET
public ResponseEntity<Page<ClientDTO>> findAll(

        @RequestParam(value = "page", defaultValue = "0") Integer page,
        @RequestParam(value = "linesPerPage", defaultValue = "12") Integer linesPerPage,
        @RequestParam(value = "orderBy", defaultValue = "moment") String orderBy,
        @RequestParam(value = "direction", defaultValue = "DESC") String direction
        ){

            //This is a endpoint to find all clients
    List<ClientDTO> list = service.findAll();
    return ResponseEntity.ok().body(list);
}
```

- Now, we will change the get method to return a page and instant a object in spring that is like a page object.

```
@GetMapping //endpoint GET
public ResponseEntity<Page<ClientDTO>> findAll(

        @RequestParam(value = "page", defaultValue = "0") Integer page,
        @RequestParam(value = "linesPerPage", defaultValue = "12") Integer linesPerPage,
        @RequestParam(value = "orderBy", defaultValue = "moment") String orderBy,
        @RequestParam(value = "direction", defaultValue = "DESC") String direction
        ){

    PageRequest pageRequest = PageRequest.of(page, linesPerPage, Direction.valueOf(direction), orderBy);

            //This is a endpoint to find all clients
    Page<ClientDTO> list = service.findAllPaged(pageRequest);

    return ResponseEntity.ok().body(list);
}
```

- Now, we have to change the findAll service.
- The findAllPaged will receive a PageRequest
- The repository method already have a findAll method receiving a PageRequest

```
@Transactional(readOnly = true) // Transactional close when make the REST request. Good Practice in Program
public Page<ClientDTO> findAllPaged(PageRequest pageRequest){
    Page<Client> list = repository.findAll(pageRequest); // We have to convert this Client list to ClientDTO list.
    return list.map(x -> new ClientDTO(x)); // Use Functional program
}
```

- Now, let test


```
{
    "content": [
        {
            "id": 34,
            "name": "ROSE",
            "cpf": "444.444.444-44",
            "income": 4000.0,
            "birthDate": "1993-12-03T10:15:30Z",
            "children": 4
        },
        {
            "id": 14,
            "name": "ROSE",
            "cpf": "444.444.444-44",
            "income": 4000.0,
            "birthDate": "1993-12-03T10:15:30Z",
            "children": 4
        },
        {
            "id": 44,
            "name": "ROSE",
            "cpf": "444.444.444-44",
```


- When we make the request GetAll, the response is a content, with the form of Descending for client name. Is this case, the page 0 have the name Rose.
- We can alter the path of the request, like the page, linesPerPage, direction and order by

`/clients?page=0&linesPerPage=6&direction=ASC&orderBy=name`

```
{
    "content": [
        {
            "id": 37,
            "name": "BRUNO",
            "cpf": "777.777.777-77",
            "income": 7000.0,
            "birthDate": "1996-12-03T10:15:30Z",
            "children": 7
        },
        {
            "id": 7,
            "name": "BRUNO",
            "cpf": "777.777.777-77",
            "income": 7000.0,
            "birthDate": "1996-12-03T10:15:30Z",
            "children": 7
        },
        {
            "id": 17,
            "name": "BRUNO",
            "cpf": "777.777.777-77",
```


/clients?page=0&linesPerPage=6&direction=ASC&orderBy=birthdate


```
            "name": "JOAO",
            "cpf": "111.111.111-11",
            "income": 1000.0,
            "birthDate": "1990-12-03T10:15:30Z",
            "children": 1
        },
        {
            "id": 1,
            "name": "JOAO",
            "cpf": "111.111.111-11",
            "income": 1000.0,
            "birthDate": "1990-12-03T10:15:30Z",
            "children": 1
        },
        {
            "id": 2,
            "name": "MARIA",
            "cpf": "222.222.222-22",
            "income": 2000.0,
            "birthDate": "1991-12-03T10:15:30Z",
            "children": 2
```


/clients?page=0&linesPerPage=6&direction=ASC&orderBy=cpf

```
        "birthDate": "1999-12-03T10:15:30Z",
        "children": 10
    },
    {
        "id": 40,
        "name": "ROBERTA",
        "cpf": "101.101.101-10",
        "income": 10000.0,
        "birthDate": "1999-12-03T10:15:30Z",
        "children": 10
    },
    {
        "id": 21,
        "name": "JOAO",
        "cpf": "111.111.111-11",
        "income": 1000.0,
        "birthDate": "1990-12-03T10:15:30Z",
        "children": 1
    }
```