

Demo JHipster

Step3

Customisation avec héritage

Dans postman lancer un GET Customers :

Les customers retournés ont le champ order : null bien qu'ils possèdent des commandes.

Cela est dû au fetchType Lazy positionné sur la relation orders de l'entité Customer

Nous allons créer un nouveau endpoint pour retourner les customers ainsi que leurs orders

Repository

Créer le nouveau Repository "CustomerRepositoryExtended" dans le package "/src/main/java/com/decathlon/demo/repository"

```
package com.decathlon.demo.repository;

import com.decathlon.demo.domain.Customer;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

@Repository
public interface CustomerRepositoryExtended extends CustomerRepo

    @Query(value = "select c from Customer c inner join fetch c.
        countQuery = "select count(c) from Customer c inner join
    Page<Customer> findAllWithOrders(Pageable page);

}
```

Service

Créer le nouveau service "CustomerServiceExtended" dans le package "src/main/java/com/deathlon/demo/service"

```
package com.deathlon.demo.service;

import com.deathlon.demo.domain.Customer;
import com.deathlon.demo.repository.CustomerRepository;
import com.deathlon.demo.repository.CustomerRepositoryExtended;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

@Service
@Transactional
public class CustomerServiceExtended extends CustomerService{

    private CustomerRepositoryExtended customerRepository;

    public CustomerServiceExtended(CustomerRepository customerRe
        super(customerRepository);
        this.customerRepository = customerRepositoryExtended;
    }

    @Transactional(readOnly = true)
    public Page<Customer> findAllWithOrders(Pageable pageable) {
        return customerRepository.findAllWithOrders(pageable);
    }
}
```

Resouce

Créer le nouveau fichier de resource "CustomerResourceExtended" dans la package "src/main/java/com/deathlon/demo/web/rest"

```
package com.deathlon.demo.web.rest;

import com.deathlon.demo.domain.Customer;
import com.deathlon.demo.service.CustomerServiceExtended;
import com.deathlon.demo.service.dto.CustomerCriteria;
import com.deathlon.demo.web.rest.util.PaginationUtil;
import org.springframework.data.domain.Page;
```

```

import org.springframework.data.domain.Pageable;
import org.springframework.http.HttpHeaders;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
@RequestMapping("/api/extended")
public class CustomerResourceExtended {

    private CustomerServiceExtended customerServiceExtended;

    public CustomerResourceExtended(CustomerServiceExtended cust
        this.customerServiceExtended = customerServiceExtended;
    }

    @GetMapping("/customers")
    public ResponseEntity<List<Customer>> getAllCustomers(Pageab
        Page<Customer> page = customerServiceExtended.findAllWit
        HttpHeaders headers = PaginationUtil.generatePaginationH
        return ResponseEntity.ok().headers(headers).body(page.ge
    }
}

```

Lancer l'application

Dans postman lancer la requete GET "Customers With Orders"

Ajouter une nouvelle colonne dans l'entité order du fichier entities.jdl

Lancer la commande `jhipster import-jdl entities.jdl`

Redemarrer l'application

Dans postman lancer la requete GET "Customers With Orders"