

Filtrage, pagination

LicencePro 2017

Olivier Perrin
Université de Lorraine

Récupération de la QueryString

```
scheme://host:port/path?queryString
```

```
GET /sandwichs?type=baguette
```

```
@Path("/sandwichs")
public class SandwichsResource {
    @GET
    @Produces("application/json")
    public Response getSandwichs(@QueryParam("type") String ptype) {
        // ptype contient "baguette"
        ...
    }
}
```

Filtrage (cont.)

Traitement en JPA

```
Query query = em.createQuery(  
    "SELECT s FROM Sandwich s WHERE s.type = :type");  
  
query.setParameter("type", "baguette");  
  
List<Sandwich> sandwiches = query.getResultList();
```

```
@NamedQueries({  
    @NamedQuery(name = "sandwichsType",  
        query = "SELECT s FROM Sandwich s " +  
            "WHERE s.type = :type ORDER BY s.id DESC"),  
})
```

Pagination

Récupération de la QueryString

```
GET /sandwichs?page=0&size=10
```

```
@Path("/sandwichs")
public class SandwichsResource {
    @GET
    @Produces("application/json")
    public Response getSandwichs(@QueryParam("page") int page,
                                  @DefaultValue("10") @QueryParam("size") int size {
        // on gère la pagination
        ...
    }
}
```

Pagination (cont.)

Traitement en JPA

- ▶ **setFirstResult(int)**: positionne l'offset pour débiter la pagination
- ▶ **setMaxResults(int)**: fixe le nombre maximum d'entités qui doivent être retournées dans la page

```
Query query = entityManager.createQuery("Select ...");  
int numPage = 1;  
int taillePage = 10;  
query.setFirstResult((numPage-1) * taillePage);  
query.setMaxResults(taillePage);  
List <...> liste = query.getResultList();
```

Pagination (cont.)

Dénombrement

```
Query nbElementsQuery = entityManager.createQuery  
    ("Select count(f.id) from Foo f");  
long nbElements = (long) nbElementsQuery.getSingleResult();
```

Dernière page

```
int taillePage = 10;  
int numPage = (int) ((nbElements / taillePage) + 1);
```