



## I Utilisation des paginateurs

### Exemple

Objet	Un exemple d'utilisation des paginateurs.
Référence	Formation esup-commons V1.3.9
Version	1.0

---

#### **Utilisation et diffusion de ce document**

*Il est de la responsabilité de chacun des destinataires de ce document de ne pas le rediffuser en dehors du cadre pour lequel il a été écrit.*

## I Utilisation des paginateurs

### Exemple 1

Utilisation et diffusion de ce document

1

### A. Paginateur

3

n°1. Le paginateur simple

3

n°2. Le paginateur Hibernate

4

### B. Bean 5

n°3. Création de la classe TestPaginator

5

n°4. Création de la classe TestHQPaginator

6

### C. JSF 7

Utilisation et diffusion de ce document

2

---

## **Utilisation et diffusion de ce document**

*Il est de la responsabilité de chacun des destinataires de ce document de ne pas le rediffuser en dehors du cadre pour lequel il a été écrit.*

## A. Paginateur

Il n'est pas rare dans les applications web de vouloir afficher des listes de valeurs.

Parce que l'espace visible sur l'écran du client est limité, on souhaite souvent afficher les résultats sur plusieurs pages ; c'est ce qu'on appelle la pagination.

La première difficulté de la pagination est donc de n'afficher qu'une partie des résultats d'un ensemble plus important, et de proposer une navigation visuelle entre les pages.

La deuxième difficulté consiste, pour les accès aux bases de données, à ne récupérer que les valeurs de la base qui doivent être affichées ; cela est indispensable lorsque le nombre de résultat est énorme, assez gros en tout cas pour saturer la mémoire des processus.

Il existe donc, deux types de paginateurs :

- le paginateur simple qui permet de paginer des données sans se soucier de leur récupération.
- le paginateur hibernate qui ne récupère de la base de données que les données à afficher.

Quelque soit le type de paginateurs choisi, la déclaration dans un controller reste sensiblement la même.

### v°1. Le paginateur simple

```
private TestPaginator testPaginator;

    public void reset() {
        super.reset();
        setTestPaginator(new TestPaginator(getDomainService()));
    }
```

### Les accesseurs (Getters et Setters)

```
public void setTestPaginator(TestPaginator testPaginator) {
    this.testPaginator = testPaginator;
}

public TestPaginator getTestPaginator() {
    return testPaginator;
}
```

---

### Utilisation et diffusion de ce document

*Il est de la responsabilité de chacun des destinataires de ce document de ne pas le rediffuser en dehors du cadre pour lequel il a été écrit.*

## v°2. Le paginateur Hibernate

```
private TestHQPaginator testHQPaginator;
```

### Les accesseurs (Getters et Setters)

```
public void setTestHQPaginator(TestHQPaginator testHQPaginator) {  
    this.testHQPaginator = testHQPaginator;  
}  
  
public TestHQPaginator getTestHQPaginator() {  
    return testHQPaginator;  
}
```

---

### Utilisation et diffusion de ce document

*Il est de la responsabilité de chacun des destinataires de ce document de ne pas le rediffuser en dehors du cadre pour lequel il a été écrit.*

## B. Bean

### v°3. Création de la classe *TestPaginator*

```
public class TestPaginator extends ListPaginator<Test> {

    1. private DomainService domainService;

    /**
    * Constructor.
    */
    2. public TestPaginator() {
    3.     super();
    4. }
    5.
    6. public TestPaginator(final DomainService domainService) {
    7.     super();
    8.     this.domainService = domainService;
    9. }
    10.
    11. /**
    12.     * @see org.esupportail.commons.web.beans.ListPaginator#getData()
    13.     */
    14. @Override
    15. protected List<Test> getData() {
    16.     return getDomainService().getTests();
    17. }
    18.
    19. public DomainService getDomainService() {
    20.     return domainService;
    21. }
    22.
    23. public void setDomainService(final DomainService domainService) {
    24.     this.domainService = domainService;
    25. }
    26. }
```

---

### Utilisation et diffusion de ce document

*Il est de la responsabilité de chacun des destinataires de ce document de ne pas le rediffuser en dehors du cadre pour lequel il a été écrit.*

## v°4. Création de la classe *TestHQPaginator*

### 1. Création de la classe

```
public class TestHQPaginator extends AbstractHibernateQueryPaginator<Test> {

    /**
     * Constructor.
     */
    public TestHQPaginator() {
        super();
    }

    /**
     * @see org.esupportail.commons.dao.AbstractHibernateQueryPaginator#getQueryString()
     */
    @Override
    protected String getQueryString() {
        return HqlUtils.from("Test");
    }

}
```

### 2. Déclaration spring de *entryHQPaginator* :

```
<bean id="testHQPaginator"
      class="org.esupportail.formation.web.beans.TestHQPaginator">
    <property name="daoService" ref="daoService"/>
</bean>
```

### 3. Injection du bean *entryHPaginator* dans le bean *test1Controller*

---

## Utilisation et diffusion de ce document

*Il est de la responsabilité de chacun des destinataires de ce document de ne pas le rediffuser en dehors du cadre pour lequel il a été écrit.*

## C. JSF

```
<%@include file="_include.jsp"%>
<e:page stringsVar="msgs" menuItem="test1"
    locale="#{sessionController.locale}">
    <%@include file="_navigation.jsp"%>
    <h:form id="formTest1">
        <e:messages />
        <e:dataTable rendered="#{not empty test1Controller.testPaginator.visibleItems}"
            id="data"
            rowIndexVar="variable"
            value="#{test1Controller.testPaginator.visibleItems}"
            var="test" border="0"
            style="width:100%"
            cellpadding="0">
            <f:facet name="header">
                <h:panelGroup>
                    <e:paginator id="testPaginator"
                        paginator="#{test1Controller.testPaginator}"
                        itemsName="#{msgs['MON.MESSAGE.i18n']}"
                        onchange="javascript:
{simulateLinkClick('formTest1:submitPageSize');}" />
                    <e:commandButton id="submitPageSize"
                        action="#{test1Controller.testPaginator.forceR
eload}" />
                </h:panelGroup>
            </f:facet>
            <t:column>
                <e:text value="#{test.value}" />
            </t:column>
            <t:column>
                <e:text value="#{test.date}" />
            </t:column>
        </e:dataTable>
    </h:form>
</e:page>
```

---

### Utilisation et diffusion de ce document

Il est de la responsabilité de chacun des destinataires de ce document de ne pas le rediffuser en dehors du cadre pour lequel il a été écrit.