

« [Part 5 – Downloading PrimeFaces themes / creating and installing custom themes and overriding PrimeFaces style sheets.](#)
[Reviewing PrimeFaces Cookbook – Packt Publishing – Oleg Varaksin, Mert Çalışkan](#) »

Part 6 – Generic Primefaces LazySorter

10 Jan 2013 | Java EE

When we lazy load our data table we will need to sort our object list which will be bound to the data table. All we need to do is to add `sortBy="#{user.username}"` property to the username column i.e in `userList.xhtml` file (in this project).



Our data table will look like this

```

1  <p:dataTable id="dataTable" var="user" value="#{userController.lazy?l
2          paginator="true" rows="10" selection="#{user
3          paginatorTemplate="{CurrentPageReport} {Fir
4          lazy="true" rowsPerPageTemplate="10,15,50">
5      <f:facet name="header">
6          User List
7      </f:facet>
8      <p:column selectionMode="multiple" style="width:18px
9      <p:column sortBy="#{user.username}">
10         <f:facet name="header">
11             <h:outputText value="Username" />
12         </f:facet>
13         <p:commandLink value="#{user.username}" update="
14             <f:setPropertyActionListener value="#{user}"
15         </p:commandLink>
16     </p:column>
17
18     <p:column sortBy="#{user.firstname}">
19         <f:facet name="header">
20             <h:outputText value="Firstname" />
21         </f:facet>
22         <h:outputText value="#{user.firstname}" />
23     </p:column>
24
25     <p:column sortBy="#{user.lastname}">
26         <f:facet name="header">
27             <h:outputText value="Lastname" />
28         </f:facet>
29         <h:outputText value="#{user.lastname}" />
30     </p:column>
31
32     <p:column sortBy="#{user.email}">
33         <f:facet name="header">
34             <h:outputText value="Email" />
35         </f:facet>
36         <h:outputText value="#{user.email}" />
37     </p:column>
38
39     <f:facet name="footer">
40         <p:commandButton value="New User" oncomplete="ne
41         <p:commandButton value="Delete Users" actionList
42     </f:facet>
43 </p:dataTable>

```

The next step is to check if `sortField` parameter is not null and sort our object list (LazyUserDataModel.java file / load function).

```

1  /**
2   * Lazy loading user list with sorting ability
3   * @param first
4   * @param pageSize
5   *

```

Blogroll

- › [Arquillian testing tool for Java EE](#)
- › [PrimeFaces JSF Component Suite](#)

Recent Posts

- › [Glassfish4 and JSF 2.2 ViewScope support \(JSF 2.2, CDI, JPA 2 \(ORM\) / EclipseLink, JAAS, MySQL, example CRUD web application\)](#)
- › [My Book Review – Instant PrimeFaces Starter- Packt Publishing](#)
- › [Reviewing Instant PrimeFaces Starter- Packt Publishing](#)
- › [My Book Review – PrimeFaces Cookbook by Oleg Varaksin and Mert Çalışkan](#)
- › [Reviewing PrimeFaces Cookbook – Packt Publishing – Oleg Varaksin, Mert Çalışkan](#)

Recent Comments

- › [simtay on Part 4 – Lazy Loading and Pagination with JSF 2.1 / Primefaces 3.4.2 DataTable and Form validation](#)
- › [simtay on Simple CRUD Web Application with JSF 2.1, PrimeFaces 3.5, Maven and JPA](#)
- › [Mohamed on Part 4 – Lazy Loading and Pagination with JSF 2.1 / Primefaces 3.4.2 DataTable and Form validation](#)
- › [dukgo on Simple CRUD Web Application with JSF 2.1, PrimeFaces 3.5, Maven and JPA](#)
- › [simtay on Simple CRUD Web Application with JSF 2.1, PrimeFaces 3.5, Maven and JPA](#)

Archives

- › [November 2013](#)
- › [August 2013](#)
- › [April 2013](#)
- › [March 2013](#)
- › [January 2013](#)
- › [September 2012](#)

Categories

```

5      * @param sortField
6      * @param sortOrder
7      * @param filters
8      * @return List<User>
9      */
10     @Override
11     public List<User> load(int first, int pageSize, String sortField
12         datasource = crudService.findWithNamedQuery(User.ALL, first,
13         // if sort field is not null then we sort the field accordin
14         if(sortField != null) {
15             Collections.sort(datasource, new LazySorter(sortField, s
16         }
17         setRowCount(crudService.countTotalRecord(User.TOTAL));
18         return datasource;
19     }

```

[Java EE](#)

Meta

[Log in](#)
[Entries RSS](#)
[Comments RSS](#)
[WordPress.org](#)

The last thing to do is to create a LazySorter class that implements java.util.Comparator interface. This class works with generic types and captures the specified fields with the help of Java reflection.

```

1  package com.nz.simplecrud.util;
2
3  import java.lang.reflect.Field;
4  import java.util.Comparator;
5  import org.primefaces.model.SortOrder;
6
7  /**
8   * Generic sorting utility class
9   * @param <T>
10  */
11  public class LazySorter<T> implements Comparator<T> {
12
13      private String sortField;
14
15      private SortOrder sortOrder;
16
17      /**
18       * initializing sorting field and sorting order
19       * @param sortField
20       * @param sortOrder
21       */
22      public LazySorter(String sortField, SortOrder sortOrder) {
23          this.sortField = sortField;
24          this.sortOrder = sortOrder;
25      }
26
27      /**
28       * Comparing object1 and object2 with reflection
29       * @param object1
30       * @param object2
31       * @return
32       */
33      @Override
34      public int compare(T object1, T object2) {
35          try {
36              Field field1 = object1.getClass().getDeclaredField(this.
37              Field field2 = object2.getClass().getDeclaredField(this.
38              field1.setAccessible(true);
39              field2.setAccessible(true);
40              Object value1 = field1.get(object1);
41              Object value2 = field2.get(object2);
42
43              int value = ((Comparable)value1).compareTo(value2);
44              return SortOrder.ASCENDING.equals(sortOrder) ? value : -
45          }
46          catch (Exception e) {
47              throw new RuntimeException();
48          }
49      }
50  }

```

Download the source code [here](#)

Please try live [demo](#) (Username: Admin, Password:1234)

Leave a Reply

 Author (required)

 Email (will not be published)(required)

Website

Just to prove you are a human, please answer the following math challenge either out loud or typing into the text box

 × eight = forty eight**b****i**

link

b-quote

code

close tags

« [Part 5 – Downloading PrimeFaces themes / creating and installing custom themes and overriding PrimeFaces style sheets.](#)

[Reviewing PrimeFaces Cookbook – Packt Publishing – Oleg Varaksin, Mert Çalışkan](#) »

© 2013 Diary of a Java Enthusiast — All Rights Reserved.