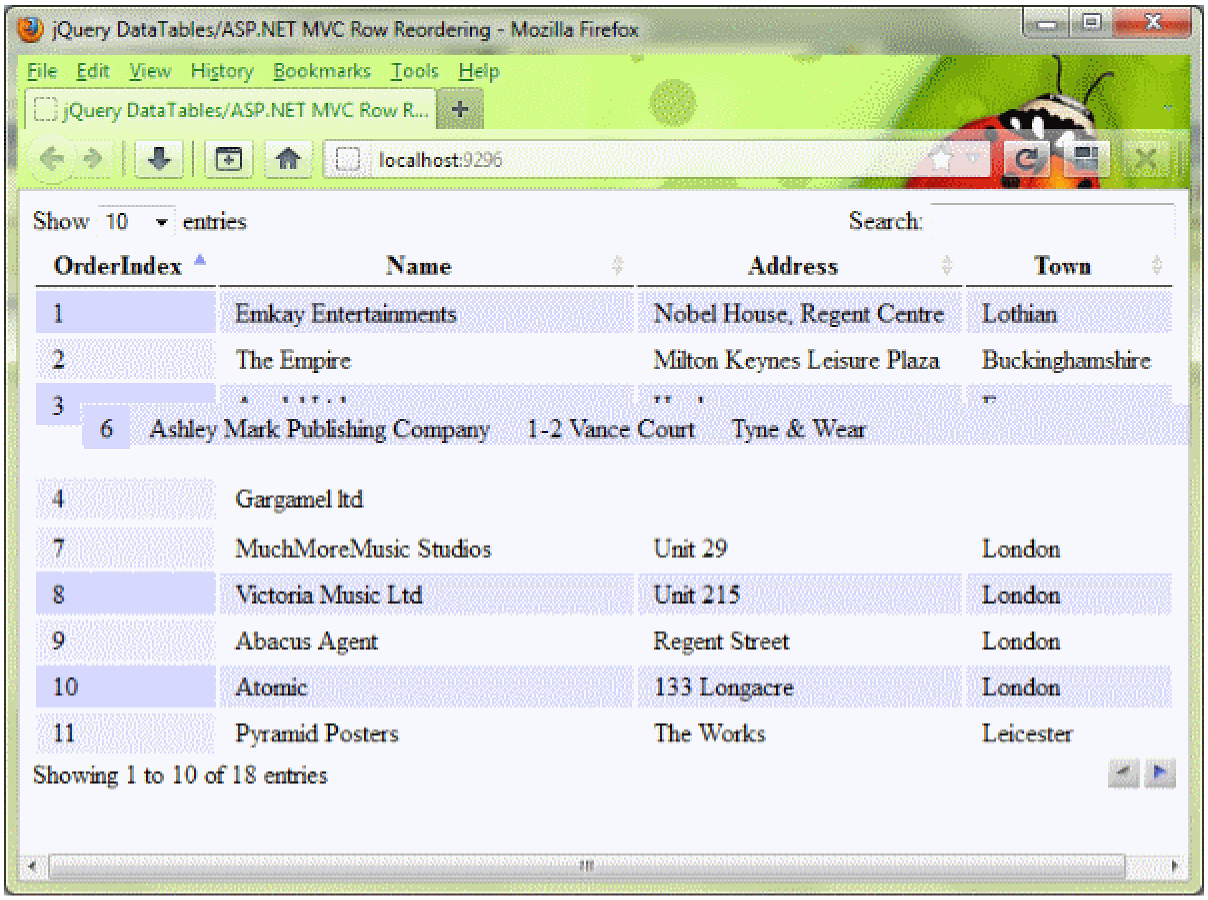
Drag and Drop in ASP.NET MVC (jQuery DataTables and ASP.NET MVC Integration - Part VI)

Table of Contents

Introduction

In this article I will explain how you can implement drag and drop reordering of rows in web tables. For drag and drop functionality we will use the jQuery UI Sortable plug-in combined with jQuery DataTables. A prerequisite is that the table is enhanced with the jQuery DataTables plug-in. An example of a table enhanced with the jQuery DataTables plug-in where drag and drop reordering is implemented is shown in the following figure:


jQuery DataTables/ASP.NET MVC Row Reordering - Mozilla Firefox
File Edit View History Bookmarks Tools Help
jQuery DataTables/ASP.NET MVC Row R...
localhost:9296

Show 10 entries Search:

OrderIndex	Name	Address	Town
1	Emkay Entertainments	Nobel House, Regent Centre	Lothian
2	The Empire	Milton Keynes Leisure Plaza	Buckinghamshire
3	[illegible]	[illegible]	[illegible]
6	Ashley Mark Publishing Company	1-2 Vance Court	Tyne & Wear
4	Gargamel ltd		
7	MuchMoreMusic Studios	Unit 29	London
8	Victoria Music Ltd	Unit 215	London
9	Abacus Agent	Regent Street	London
10	Atomic	133 Longacre	London
11	Pyramid Posters	The Works	Leicester

Showing 1 to 10 of 18 entries

The jQuery DataTables plug-in enhances a plain HTML table and adds a lot of JavaScript functionalities. It has a lot of additional features such as Excel-like navigation through the table, inline cell editing, column reordering, etc. One of the features that is provided is row drag and drop. In this article, I will explain how you can use thejQuery DataTables drag and drop feature in your ASP.NET MVC application.

Background

This article is part of a series about integration of the jQuery DataTables plug-in with ASP.NET MVC web applications. You might also want to take a look at other articles in this group:

1. jQuery DataTables and ASP.NET MVC Integration
2. ASP.NET MVC Editable Table
3. Refreshing the content of DataTable using AJAX in ASP.NET MVC
4. Creating an expandable master-details table
5. jQuery DataTables Advanced Filtering in ASP.NET MVC

In these articles, you might find a lot of useful information about the usage of the jQuery DataTables plug-in in ASP.NET MVC. If you have not read the first article, I would recommend that you read it too because it contains some important details about the basic integration of jQuery DataTables with ASP.NET MVC applications.

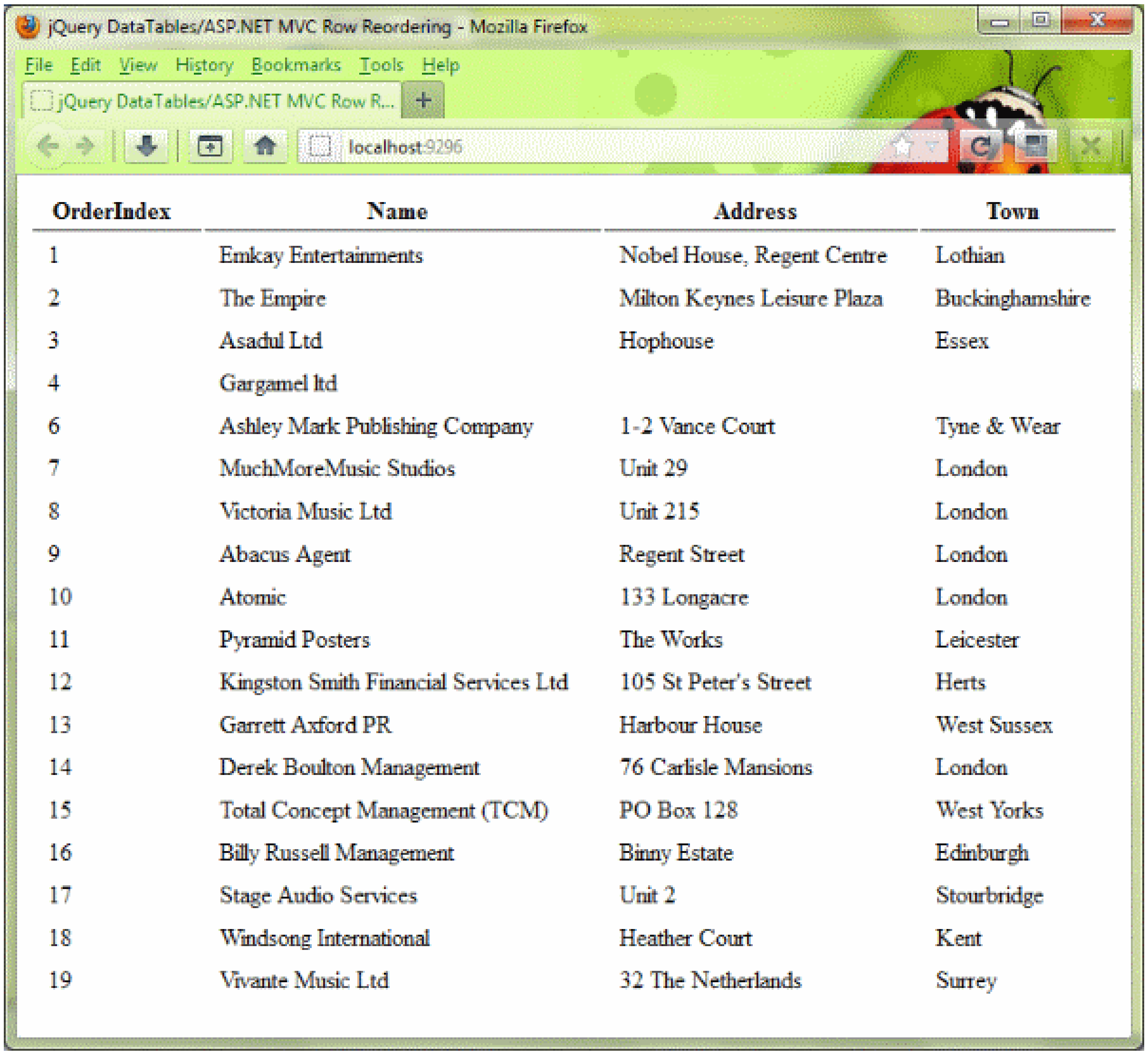
jQuery DataTables is an excellent jQuery plug-in that enables you to display and manage information in a table. Detailed instructions about integrating the jQuery DataTables plug-in in an ASP.NET MVC application is described in the "jQuery DataTables and ASP.NET MVC Integration" article, but here I will shortly summarize how you can integrate this plug-in.

First, you will need to place a plain HTML table in the page. An example of a table is shown in the following code:

Hide Copy Code

```
<table class="display" id="example">
     <thead>
          <tr>
               <th>OrderIndex</th><th>Name</th><th>Address</th><th>Town</th>
          </tr>
     </thead>
     <tbody>
          <tr>
               <td>1</td><td>Gowi</td><td>Town</td><td>Pancevo</td>
          </tr>
          <tr>
               <td>2</td><td>Microsoft</td><td></td><td>Redmond</td>
          </tr>
          <tr>
               <td>3</td><td>IBM</td><td></td><td></td>
          </tr>
          <tr>
               <td>4</td><td>Oracle</td><td></td><td></td>
          </tr>
     </tbody>
</table>
```

This will be generated as a standard web table - very similar to the one on the following figure:


jQuery DataTables/ASP.NET MVC Row Reordering - Mozilla Firefox
File Edit View History Bookmarks Tools Help
jQuery DataTables/ASP.NET MVC Row R...
localhost:9296

OrderIndex	Name	Address	Town
1	Emkay Entertainments	Nobel House, Regent Centre	Lothian
2	The Empire	Milton Keynes Leisure Plaza	Buckinghamshire
3	Asadul Ltd	Hophouse	Essex
4	Gargamel ltd		
6	Ashley Mark Publishing Company	1-2 Vance Court	Tyne & Wear
7	MuchMoreMusic Studios	Unit 29	London
8	Victoria Music Ltd	Unit 215	London
9	Abacus Agent	Regent Street	London
10	Atomic	133 Longacre	London
11	Pyramid Posters	The Works	Leicester
12	Kingston Smith Financial Services Ltd	105 St Peter's Street	Herts
13	Garrett Axford PR	Harbour House	West Sussex
14	Derek Boulton Management	76 Carlisle Mansions	London
15	Total Concept Management (TCM)	PO Box 128	West Yorks
16	Billy Russell Management	Binny Estate	Edinburgh
17	Stage Audio Services	Unit 2	Stourbridge
18	Windsong International	Heather Court	Kent
19	Vivante Music Ltd	32 The Netherlands	Surrey

Then you will need to bind the table with the plug-in. This is done using the following jQuery call:

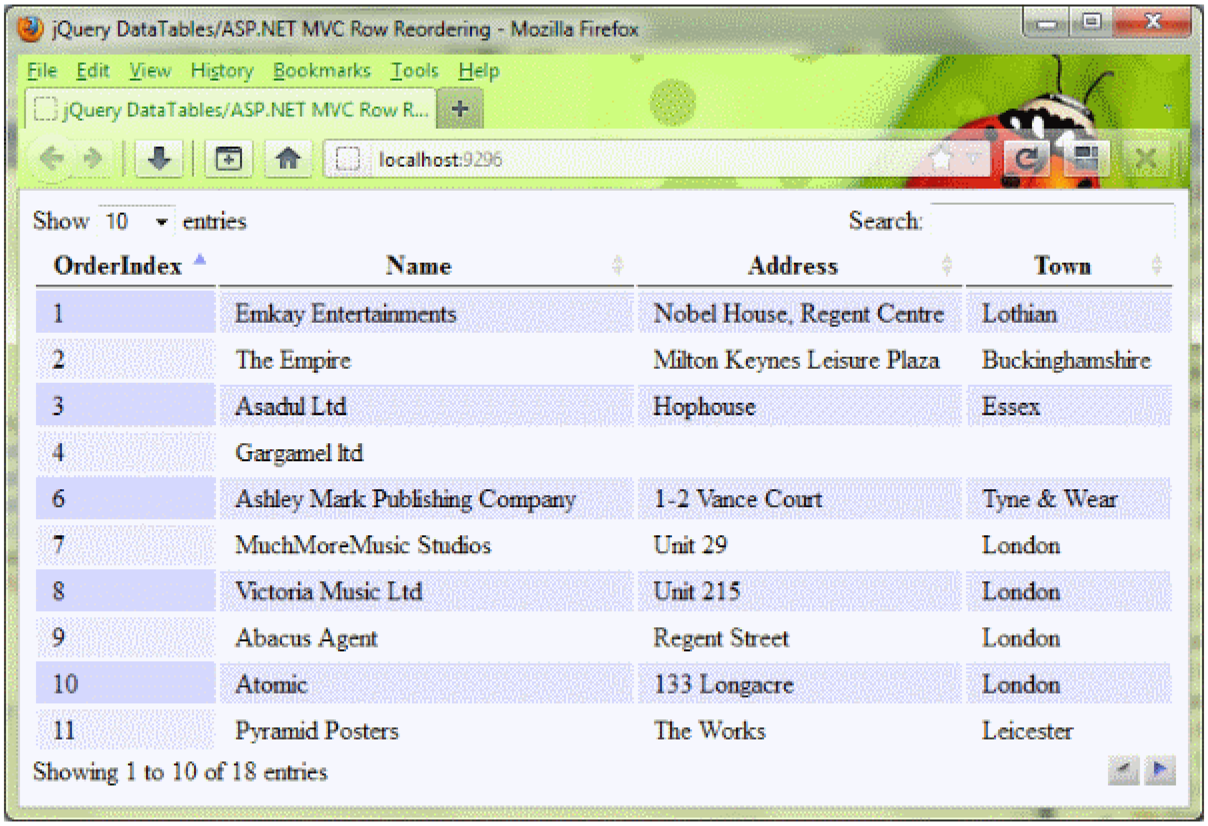
Hide Copy Code

```
<script type="text/javascript" charset="utf-8">
$(document).ready( function () {
       $('#example').dataTable();

 });

</script>
```

As a result, you will get a fully functional table with pagination, sorting, and filtering as shown in the following figure:


jQuery DataTables/ASP.NET MVC Row Reordering - Mozilla Firefox
File Edit View History Bookmarks Tools Help
jQuery DataTables/ASP.NET MVC Row R...
localhost:9296
Show 10 entries
Search:

OrderIndex	Name	Address	Town
1	Emkay Entertainments	Nobel House, Regent Centre	Lothian
2	The Empire	Milton Keynes Leisure Plaza	Buckinghamshire
3	Asadul Ltd	Hophouse	Essex
4	Gargamel ltd		
6	Ashley Mark Publishing Company	1-2 Vance Court	Tyne & Wear
7	MuchMoreMusic Studios	Unit 29	London
8	Victoria Music Ltd	Unit 215	London
9	Abacus Agent	Regent Street	London
10	Atomic	133 Longacre	London
11	Pyramid Posters	The Works	Leicester

Showing 1 to 10 of 18 entries

As you can see, only oe line of JavaScript code is required to enhance a plain table with advanced sorting, filtering, and pagination functionalities. The goal of this article is to show how you can implement reordering table rows using drag and drop on top of these functionalities.

Using the code

The code is organized in a standard Model-View-Controller architecture.

Model

The Model comes to a simple class containing company data. The fields that we need are company ID, name, address, and town. Also, each company object contains a unique order index information. Companies will be ordered by this index in the view. The source code of the company model class is shown below:

Hide Copy Code

```
public class Company
 {

    public int ID { get; set; }
    public int OrderIndex{ get; set; }
    public string Name { get; set; }
    public string Address { get; set; }
    public string Town { get; set; }
 }
```

View

Since the data presentation is done on the client-side, the classic View page is fairly simple. It contains a simple HTML table "decorated" with the jQuery DataTables plug-in. For example:

Hide Shrink ▲ Copy Code

```
@{

     Layout = null;

 }


<!DOCTYPE html>
<html>
     <head>
          <title>jQuery DataTables/ASP.NET MVC Row Reordering</title>
          <link href="/Content/dataTables/demo_table.css" rel="stylesheet" type="text/css" />
          <script src="/Scripts/jQuery-1.5.1.min.js" type="text/javascript"></script>
          <script src="/Scripts/jQuery-ui-1.8.11.min.js" type="text/javascript"></script>
          <script src="/Scripts/jQuery.dataTables.min.js" type="text/javascript"></script>
          <script src="/Scripts/jQuery.dataTables.rowReordering.js" type="text/javascript"></script>
          <script type="text/javascript">
          $(document).ready(function () {
                     $('#myDataTable').dataTable();

           });

          </script>
     </head>
     <body>
                         <table id="myDataTable" class="display">
                              <thead>
                                   <tr>
                                        <th></th>
                                        <th>Company name</th>
                                        <th>Address</th>
                                        <th>Town</th>
                                   </tr>
                              </thead>
                              <tbody>
                               @foreach(var company in Model) {

                                   <tr id="@company.ID">
                                        <tr>@company.OrderIndex</tr>
                                        <tr>@company.Name</tr>
                                        <tr>@company.Address</tr>
                                        <tr>@company.Town</tr>
                                   </tr>

                               }

                              </tbody>
                         </table>
     </body>
</html>
```
The assumption is that the controller has passed an enumeration of companies to view.
The View generates a valid HTML table according to the jQuery DataTables requirements. Additionally, this View places the ID of each company as an ID attribute of rows, and the order index in the first column. This is not required for the jQuery DataTables plug-in, but this information will be used to determine what row is moved on some position.
Also, as part of the View we would need to initialize a table with the jQuery DataTables plug-in. This initialization script will add standard DataTables features (pagination, filtering by keyword, and sorting).

Controller

The Controller in this example has two functions - to provide an initial list of companies that will be displayed in the table and to update the order of companies once they are reordered by the user.
The first functionality is fairly simple - it should just pass a list of companies to view. This controller action is shown in the following listing:

Hide Copy Code

```
public class Controller
 {

    public ActionResult Index()
     {

         return View(DataRepository.GetCompanies());
     }
 }
```

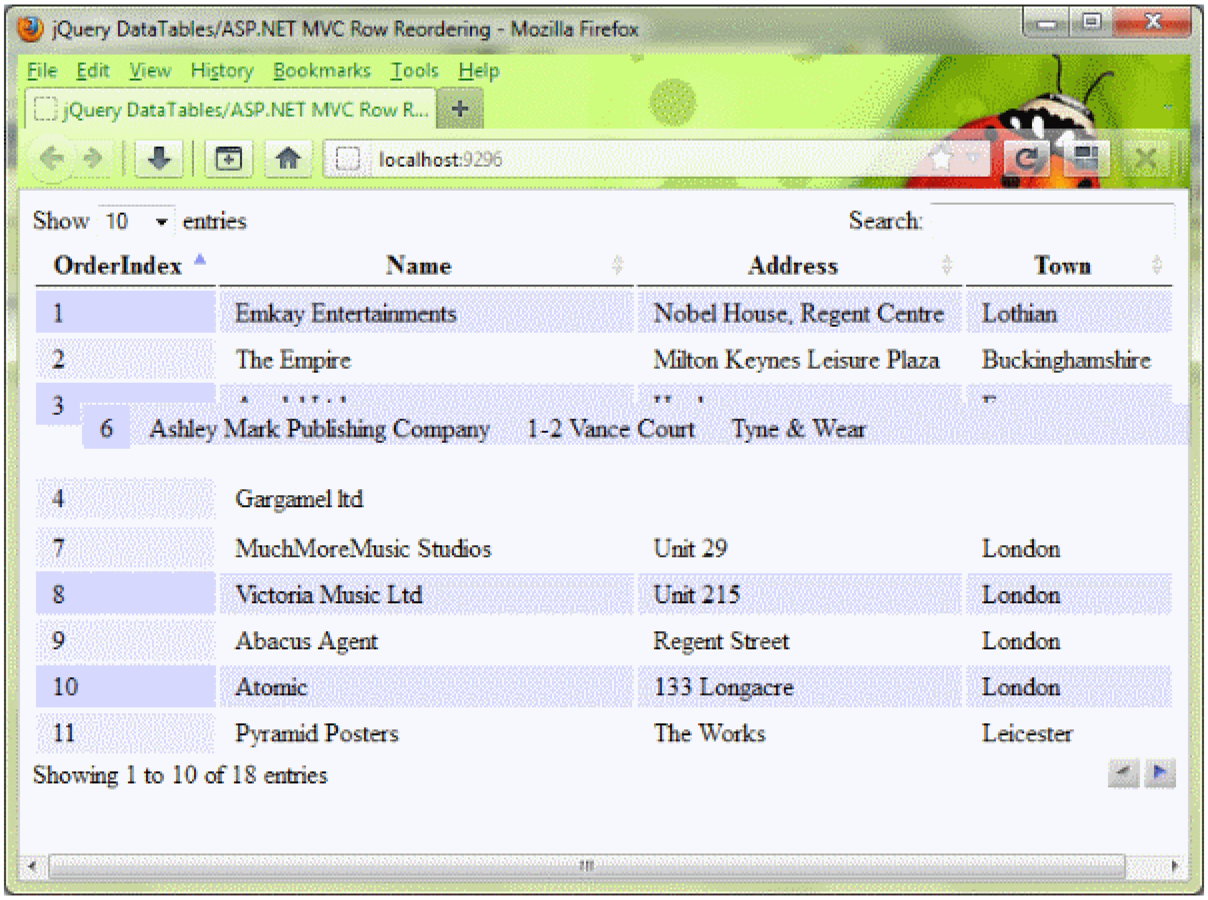
The second function is shown in the next section.

Implement row reordering

Now we have a table with all enhancements and we want to add row reordering. The implementation of reordering is simple: an additional call needs to be done in the DataTable initialization:

```
$(document).ready(function () {
    $('#myDataTable').dataTable().rowReordering();
 });
```

Once the `rowReordering` add-on is applied, you will be able to sort the rows in the table using drag'n'drop functionality. The plug-in itself will manage row positions and update the indexing column. Note that sorting in the table is disabled by the add-on because rows will need to be ordered by the order index column. In the following figure you can see how reordering works:


jQuery DataTables/ASP.NET MVC Row Reordering - Mozilla Firefox
File Edit View History Bookmarks Tools Help
jQuery DataTables/ASP.NET MVC Row R...
localhost:9296

Show 10 entries Search:

OrderIndex	Name	Address	Town
1	Emkay Entertainments	Nobel House, Regent Centre	Lothian
2	The Empire	Milton Keynes Leisure Plaza	Buckinghamshire
3	[illegible]	[illegible]	[illegible]
6	Ashley Mark Publishing Company	1-2 Vance Court	Tyne & Wear
4	Gargamel ltd		
7	MuchMoreMusic Studios	Unit 29	London
8	Victoria Music Ltd	Unit 215	London
9	Abacus Agent	Regent Street	London
10	Atomic	133 Longacre	London
11	Pyramid Posters	The Works	Leicester

Showing 1 to 10 of 18 entries

As you can see, jQuery UI sortable is applied on the table rows. Each time a row is moved, the plug-in will update the order index numbers.
There are a lot of options that can be used to configure the plug-in - you can see other options on the jQuery DataTables Row Reordering wiki.

Updating an order on the server-side

Updating an order on the web page is not very useful if you do not persist the changes. Therefore, the row reordering plug-in allows you to send information about

Hide Copy Code

```
<script type="text/javascript">
    $(document).ready(function () {
        $('#myDataTable').dataTable().rowReordering({ sURL: "/Company/UpdateOrder" });

     });

</script>
```

Each time rows are reordered, the plug-in will send an AJAX request with the following information:

- `id` - the ID of the row that is moved. This ID is placed in the ID attribute of the table row (TR) that is moved.
- `fromPosition` - the initial position of the row
- `toPosition` - position where the row is dropped
- `direction` - back or forward

The plug-in will update the order indexes in the table, and the Controller that accepts these parameters should update the order on the server-side. This controller action is shown in the following listing:

Hide Copy Code

```
public void UpdateOrder(int id, int fromPosition, int toPosition, string direction)
 {

    if (direction == "back")
```

```
     {
         var movedCompanies = DataRepository.GetCompanies()
                         .Where(c => (toPosition <= c.OrderIndex && c.OrderIndex <= fromPosition))
                         .ToList();


         foreach (var company in movedCompanies)
          {
               company.OrderIndex++;
          }
     }
    else
     {
         var movedCompanies = DataRepository.GetCompanies()
                         .Where(c => (fromPosition <= c.OrderIndex && c.OrderIndex <= toPosition))
                         .ToList();
         foreach (var company in movedCompanies)
          {
               company.OrderIndex--;
          }
     }


     DataRepository.GetCompanies().First(c => c.ID == id).OrderIndex = toPosition;
}
```

In this method, order indexes will be moved back or forward depending on the direction. As a result, the order index will be synchronized on both sides. In this example, I have used an in-memory list of objects, but in a real case you will probably update the order indexes in the database.

Conclusion

In this article I have explained how you can implement reordering of rows in a table. As you can see you need minimal effort on the View side - just a plain table and simple JavaScript that can be easily changed. The plug-in will handle everything for you on the client side and all you need to do is to create a controller action method that will handle AJAX calls sent by the plug-in and update the order on the server side too. You can see other features of the row reordering plug-in on the plug-in site.

Courtesy:
http://www.codeproject.com/Articles/331986/Table-Row-Drag-and-Drop-in-ASP-NET-MVC-jQuery-Data