

در [قسمت قبلی](#) شما را با DataTables آشنا کردم. به طور خلاصه نحوه اعمال کردن DataTables به یک جدول ساده html را گفتیم که با این کار به صورت پیش فرض، امکاناتی مثل فیلتر کردن داده ها، صفحه بندی و مرتب سازی آنها و نیز اعمال شدن استایل های css به همین جدول html خام اضافه می شود. نکته مهم در مثال قبلی این بود که داده های درون این جدول با کدنویسی خام html فراهم شدند، اما این را در نظر داشته باشید که اکثریت مواقع باید داده ها از یک بانک اطلاعاتی دریافت شوند و سپس درون جدول قرار بگیرند.

در این قسمت سعی خواهیم کرد تا منبع داده جدول را یک آرایه جاوا اسکریپتی و سپس کالکشنی از آبجکتهای جاوا اسکریپتی (json) در نظر بگیریم و نیز برخی ویژگی های پیش فرض پلاگین را غیر فعال نمائیم.

فرض کنید می خواهید لیستی از اطلاعات دانشجویان شامل نام (FirstName)، نام خانوادگی (LastName)، و سن (Age) را نمایش دهید. اطلاعات قرار است در جدول زیر قرار بگیرند:

```
<table id="std-grid">
  <thead>
    <th>نام</th>
    <th>نام خانوادگی</th>
    <th>سن</th>
  </thead>
  <tbody>
    <tr>
      <td>
      </td>
    </tr>
  </tbody>
</table>
```

مشاهده می کنید که این جدول فقط شامل قسمت header است و در بدنه آن هیچ سطری قرار نگرفته است. در این مثال اطلاعات از یک آرایه جاوا اسکریپتی باید خوانده شوند و تبدیل به html شده و در نهایت درون قسمت <tbody></tbody> آن تزریق شوند. خوشبختانه DataTables برای این کار امکانات آماده ای را در اختیار قرار می دهد. این کار بدین صورت قابل انجام است:

```
<script>
$(document).ready(function() {
  $('#std-grid').dataTable({
    "aaData": [
      ["24", "پژمان", "پارسائی"],
      ["25", "سعید", "الیاسی"],
      ["20", "محمد رضا", "گلزار"],
      ["19", "آرش", "ایرانی"],
      ["22", "مرتضی", "فرمانی"],
      ["23", "سعید", "حمیدیان"],
      ["23", "امین", "پارساآیا"],
      ["24", "محمد امین", "فقیهی"],
      ["25", "محمد", "خرمی"],
      ["20", "سینا", "امیربان"],
      ["19", "آرش", "ایرانی"],
      ["22", "وحید", "فرزانه"],
      ["23", "امیر علی", "فرمانی"],
      ["23", "امین", "حسینی"]
    ]
  });
});
</script>
```

شرح کد:

aaData : یک آرایه دو بعدی (که به آن ماتریس یا آرایه ای از آرایه ها هم گفته می شود) است که مقادیر سلول هایی را نشان

می‌دهد که در جدول قرار خواهند گرفت. تعداد ستون‌ها در این آرایه دو بعدی باید با تعداد ستون‌های جدول html متناظر یکسان باشند.

در مثال بالا از یک ماتریس به عنوان منبع داده استفاده شد. منبع داده می‌تواند به فرمت json نیز باشد. البته در این صورت باید ستون‌های جدول html را هم به پلاگین معرفی کنید، بدین صورت:

```
$(document).ready(function() {
    $('#std-grid').dataTable({
        "aaData": [
            { "FirstName": "پژمان", "LastName": "پارسائی", "Age": "24" },
            { "FirstName": "سعید", "LastName": "الیاسی", "Age": "25" },
            { "FirstName": "محمد رضا", "LastName": "گلزار", "Age": "24" },
            { "FirstName": "آرش", "LastName": "ایرانی", "Age": "24" },
            { "FirstName": "مرتضی", "LastName": "فرمانی", "Age": "24" },
            { "FirstName": "سعید", "LastName": "حمیدیان", "Age": "24" },
            { "FirstName": "امین", "LastName": "پارسانیا", "Age": "24" },
            { "FirstName": "محمد امین", "LastName": "فقیهی", "Age": "24" },
            { "FirstName": "محمد", "LastName": "خرمی", "Age": "24" },
            { "FirstName": "سینا", "LastName": "امیریان", "Age": "24" },
            { "FirstName": "آرش", "LastName": "ایرانی", "Age": "24" },
            { "FirstName": "وحید", "LastName": "فرزانه", "Age": "24" },
            { "FirstName": "امیر علی", "LastName": "فرمانی", "Age": "24" },
            { "FirstName": "امین", "LastName": "حسینی", "Age": "24" },
        ],
        "aoColumns": [
            { "mDataProp": "FirstName" },
            { "mDataProp": "LastName" },
            { "mDataProp": "Age" }
        ]
    });
});
```

aaData: همان طور که گفته شد در این قسمت دیتاهای درون جدول آورده می‌شوند و در این مثال آنها به فرمت json نوشته شده‌اند.

aoColumns: در این قسمت باید اسم ستون‌های جدول ذکر شوند.

غیرفعال کردن بعضی از ویژگی‌های پیش فرض DataTables

همان طور که گفته شد پلاگین DataTables به صورت پیش فرض ویژگی‌های مرتب سازی (sorting)، صفحه بندی (paging)، فیلتر کردن داده‌ها (filtering)، و غیره را به جدول مورد نظرش اعمال می‌کند. و بدین صورت قابل تغییر است:

```
$('#std-grid').dataTable({
    "bPaginate": false,
    "bLengthChange": false,
    "bFilter": false,
    "bSort": false,
    "bInfo": true,
    "bAutoWidth": false
});
```

bPaginate: بیان می‌کند آیا صفحه بندی سطرهای جدول فعال باشد یا نه.

bLengthChange: در صورتی که قابلیت صفحه بندی فعال باشد، بیان می‌کند که کاربر بتواند اندازه صفحه را تغییر دهد یا نه.

bFilter: بیان می‌کند آیا قابلیت فیلتر کردن داده‌ها فعال باشد یا نه.

bSort: بیان می‌کند قابلیت مرتب سازی داده‌های جدول فعال باشد یا نه.

bInfo: بیان می‌کند که قسمت info زیر گرید نشان داده شود یا نه (در این قسمت اطلاعاتی راجع به تعداد کل رکوردهای بایند شده به جدول و نیز رکوردهای درون صفحه جاری نشان داده می‌شود)

bAutoWidth: در صورتی که این گزینه فعال باشد اندازه عرض هر ستون به صورت خودکار توسط DataTables مقدار دهی خواهد شد.

مقدارهای قابل قبول برای هر کدام از این خصوصیات : true یا false

کدهای مربوط به این مثال را می‌توانید از لینک زیر دریافت کنید:

[DataTables-DoteNetTips-Tutorial-02.zip](#)

نظرات خوانندگان

نویسنده: farzad

تاریخ: ۲۰:۵۱ ۱۳۹۲/۰۴/۲۵

با سلام

آیا این پلاگین قابلیت حذف و یا ویرایش داده‌ها رو هم می‌ده؟

نویسنده: پژمان پارسائی

تاریخ: ۲۱:۲۷ ۱۳۹۲/۰۴/۲۵

سلام

بله، نمونه پیاده سازی شده در MVC رو می‌تونید توی لینک زیر مشاهده کنید:

[\(ASP.NET MVC Editable Table \(jQuery DataTables and ASP.NET MVC integration - Part II](#)