



EBook Gratis

APRENDIZAJE datatables

Free unaffiliated eBook created from
Stack Overflow contributors.

#datatables

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con los datos	2
Observaciones.....	2
Versiones.....	2
Examples.....	3
Instalación.....	3
Inicializando un DataTable mínimo:.....	3
Función habilitar / deshabilitar (opciones de tablas de datos).....	4
API de tablas de datos.....	4
Capítulo 2: ¿Cómo obtener el valor de búsqueda ingresado en Datatables mediante programaci.....	6
Examples.....	6
Ejemplo.....	6
Capítulo 3: Agregar botones de exportación a la tabla en Bootstrap 4.....	7
Introducción.....	7
Examples.....	7
Añadir botones a la tabla.....	7
Capítulo 4: Cuadro de entrada de búsqueda de datos para una búsqueda en tiempo real.....	9
Examples.....	9
Cuadro de entrada de búsqueda para búsqueda progresiva en el datatable.....	9
Capítulo 5: datatables - opción Mostrar filas seleccionadas.....	10
Examples.....	10
Mostrar solo las filas seleccionadas.....	10
Capítulo 6: Migración de <1.10 a 1.10 y superior.....	11
Introducción.....	11
Sintaxis.....	11
Observaciones.....	11
Examples.....	11
Inicialización de Datatable 1.10+.....	11
Características No disponible en Datatables 1.10+.....	12
Capítulo 7: Procesamiento de datos del lado del servidor.....	14

Examples.....	14
Cargar datos usando ajax con procesamiento del lado del servidor.....	14
DataTables 1.10+ Serverside Processing.....	14
Obtener datos JSON de la tabla MySQL.....	16
Creditos.....	19

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [datatables](#)

It is an unofficial and free datatables ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official datatables.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con los datos

Observaciones

Características clave

- Paginación de longitud variable
- Filtrado sobre la marcha
- Clasificación multi-columna con detección de tipo de datos
- Manejo inteligente de anchos de columna.
- Muestra datos de casi cualquier fuente de datos
- DOM, matriz de JavaScript, archivo Ajax y procesamiento del lado del servidor
- Opciones de desplazamiento para la vista de tabla
- Totalmente internacionalizable
- Soporte de jQuery UI ThemeRoller
- Amplia variedad de plug-ins
- Es gratis

Recursos

- [Uso](#)
- [Ejemplos](#)
- [estilo](#)
- [API](#)
- [Desarrollo](#)
- [Extras](#)
- [Plug-ins](#)
- [Blog](#)
- [Preguntas frecuentes](#)
- [Los foros](#)
- [Ejemplos del lado del servidor](#)

Versiones

Versión	Notas de lanzamiento	Fecha de lanzamiento
1.9 y anteriores (legado)	https://datatables.net/forums/discussion/8332/datatables-1-9-0-released	2014-02-01
1.10 y después	https://datatables.net/new/1.10	2015-08-12

Examples

Instalación

Tenga archivos JavaScript y CSS requeridos incluidos en su `index.html` . Puede hacerlo usando los archivos CDN disponibles en las siguientes rutas:

```
<link rel="stylesheet" type="text/css"
href="//cdn.datatables.net/1.10.12/css/jquery.dataTables.css">

<script type="text/javascript" charset="utf8"
src="//cdn.datatables.net/1.10.12/js/jquery.dataTables.js"></script>
```

O descargando archivos locales individuales y alojándolos usted mismo. Para obtener un paquete completo de archivos JavaScript y CSS , visite el [generador de descargas de DataTables](#) que le permitirá seleccionar y elegir las características que necesita y condensarlas en un solo paquete (u ofrecer archivos individuales). Incluya estos en el orden que se muestra en la parte inferior de la página.

DataTables depende de jQuery , así que `jquery.dataTables.js` antes de `jquery.dataTables.js` :

```
<script type="text/javascript" src="https://code.jquery.com/jquery-3.1.0.min.js"></script>
```

Las tablas de datos también están disponibles a través de NPM

```
npm install datatables.net      # Core library
npm install datatables.net-dt  # Styling
```

y Bower

```
bower install --save datatables.net
bower install --save datatables.net-dt
```

Inicializando un DataTable mínimo:

El código a continuación convertirá la tabla con un ID de `tableid` en un DataTable, así como devolverá una instancia de la API de DataTables:

```
$(document).ready(function() {
    $('#tableid').DataTable();
});
```

Compare esto con el siguiente código, que convertirá la tabla en un DataTable pero no devolverá una instancia de la API de DataTables:

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

Consulte la sección de [documentación de la API de DataTables](#) para obtener más detalles sobre lo que se puede hacer con la instancia de la API de DataTables.

Función habilitar / deshabilitar (opciones de tablas de datos)

DataTables tiene la capacidad de habilitar o deshabilitar varias de sus funciones, como la paginación o la búsqueda. Para elegir estas opciones, simplemente selecciónelas en su inicialización:

```
$(document).ready(function() {  
    $('#tableid').DataTable( {  
        "paging":    false, //Turn off paging, all records on one page  
        "ordering":  false, //Turn off ordering of records  
        "info":      false  //Turn off table information  
    } );  
} );
```

Tenga en cuenta que las comillas alrededor de los nombres de las opciones son opcionales:

```
paging: false,  
ordering: false,  
info: false
```

También es perfectamente válido.

Una lista completa de opciones se puede encontrar [aquí](#) , junto con descripciones de los usos de cada opción.

Estas opciones solo se pueden configurar una vez, cuando se inicializa la tabla. Sin embargo, puede solucionar esta limitación agregando:

```
destroy: true
```

API de tablas de datos

DataTables viene con una API extensa que se utiliza para manipular u obtener información sobre las DataTables en una página.

Se puede acceder a la API de 3 maneras:

```
var table = $('#tableid').DataTable(); //DataTable() returns an API instance immediately  
var table = $('#tableid').dataTable().api(); //dataTable() returns a jQuery object  
var table = new $.fn.dataTable.Api('#tableid');
```

Una vez que el objeto se ha establecido, puede llamar a cualquiera de las funciones de la API en ese objeto.

```
var columns = table.columns();
```

Un ejemplo más complejo es [agregar algunas filas](#) a su tabla:

```
table.rows.add( [ {  
    "name":      "John Doe",  
    "employee_id": "15135",  
    "department": "development",  
}, {  
    "name":      "Jane Smith",  
    "employee_id": "57432",  
    "department": "quality assurance",  
} ] )  
table.draw();
```

La lista completa de funciones API se puede encontrar [aquí](#) .

Lea [Empezando con los datos en línea](#):

<https://riptutorial.com/es/datatables/topic/1844/empezando-con-los-datos>

Capítulo 2: ¿Cómo obtener el valor de búsqueda ingresado en Datatables mediante programación?

Examples

Ejemplo

Este es el código para filtrar los Datatables [1.10.7] por valor programáticamente, puede encontrarlo en la documentación oficial.

```
function setFilterValue(datatable, value){
    if(datatable !== undefined){
        datatable
            .columns(0)
            .search(value)
            .draw();
    }
}
```

Este es el código para obtener el valor de la búsqueda anterior.

```
function getFilterValue(datatable){
    var value;
    if(datatable !== undefined){
        value = datatable
            .settings()[0]
            .oSavedState
            .columns[0]
            .search.search;
    }
    return value;
}
```

Este enfoque es útil cuando tiene el caché activo (*"stateSave": verdadero*) y necesita conocer el valor de búsqueda anterior después de volver a cargar la página.

Lea ¿Cómo obtener el valor de búsqueda ingresado en Datatables mediante programación? en línea: <https://riptutorial.com/es/datatables/topic/6085/-como-obtener-el-valor-de-busqueda-ingresado-en-datatables-mediante-programacion->

Capítulo 3: Agregar botones de exportación a la tabla en Bootstrap 4

Introducción

Con el complemento datatables puede agregar botones de exportación a su tabla.

Puede exportar los datos de su tabla a Excel, PDF o copiarlos en el portapapeles.

Este manual está destinado para el framework bootstrap 4.

Examples

Añadir botones a la tabla

En su archivo JS agregue esta **opción** a su datatable:

```
buttons: [ 'excel', 'pdf', 'copy' ]
```

Se verá como

```
$('#yourTableID').DataTable({  
  buttons: [ 'excel', 'pdf', 'copy' ]  
});
```

Agregue los archivos **CSS** necesarios para la datatable con los botones:

```
<link rel="stylesheet" type="text/css"  
href="//cdn.datatables.net/1.10.15/css/dataTables.bootstrap4.min.css"/>  
<link rel="stylesheet" type="text/css"  
href="//cdn.datatables.net/buttons/1.3.1/css/buttons.bootstrap4.min.css"/>
```

Agregue los archivos **javascript** necesarios para la datatable con los botones:

```
<script type="text/javascript"  
src="//cdn.datatables.net/1.10.15/js/jquery.dataTables.min.js"></script>  
<script type="text/javascript"  
src="//cdn.datatables.net/1.10.15/js/dataTables.bootstrap4.min.js"></script>  
<script type="text/javascript"  
src="//cdn.datatables.net/buttons/1.3.1/js/dataTables.buttons.min.js"></script>  
<script type="text/javascript"  
src="//cdn.datatables.net/buttons/1.3.1/js/buttons.bootstrap4.min.js"></script>  
  
<script type="text/javascript"  
src="//cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>  
<script type="text/javascript"  
src="//cdn.rawgit.com/bpampuch/pdfmake/0.1.27/build/pdfmake.min.js"></script>  
<script type="text/javascript"
```

```
src="//cdn.rawgit.com/bpampuch/pdfmake/0.1.27/build/vfs_fonts.js"></script>
<script type="text/javascript"
src="//cdn.datatables.net/buttons/1.3.1/js/buttons.html5.min.js"></script>
```

Se verá como esta imagen:



Si no ve los botones, agregue esta opción:

```
dom: 'Blfrtip',
```

a la lista de opciones de datos. Así se ve:

```
$('#yourTableID').DataTable({
  dom: 'Blfrtip',
  buttons: [ 'excel', 'pdf', 'copy' ]
});
```

Encontrará más información para definir los elementos de control de la tabla que aparecerán en la página y en qué orden en esta [página](#) .

Nota: el requisito previo es que los archivos jQuery y bootstrap4 estén instalados en su proyecto.

Lea [Agregar botones de exportación a la tabla en Bootstrap 4 en línea](#):

<https://riptutorial.com/es/datatables/topic/10149/agregar-botones-de-exportacion-a-la-tabla-en-bootstrap-4>

Capítulo 4: Cuadro de entrada de búsqueda de datos para una búsqueda en tiempo real

Examples

Cuadro de entrada de búsqueda para búsqueda progresiva en el datatable

A continuación se muestra un ejemplo para implementar un cuadro de entrada de búsqueda que ayuda a los usuarios a buscar las ocurrencias de un valor en particular a través del datatable.

En el ejemplo siguiente, #report es el ID de div del div que contiene el cuadro de entrada de búsqueda. Esta función se llama tan pronto como el usuario ingresa un valor en este cuadro de entrada. Como puede haber muchas apariciones de un solo carácter, llamamos a la función de búsqueda real solo cuando se ingresa más de 1 carácter en el cuadro de búsqueda.

```
$('#report').on('input',function(e){
    if($(this).data("lastval")!= $(this).val()){
        $(this).data("lastval",$(this).val());
        //change action
        if($('#report').val().length >1 ){
            searchTable($(this).val());
        }
    }
});
```

En la función searchTable usamos la función incorporada de datos **fnFilter** para encontrar las **apariciones** coincidentes de la cadena de entrada. **Podemos restringir la búsqueda a una columna en particular al pasar el índice de la columna.** Aquí estamos pasando el índice de columna 2.

```
function searchTable(inputVal) {
    var dataTable = $("#report_table").dataTable();
    dataTable.fnFilter(inputVal,2);
}
```

Si necesita buscar en todas las columnas, **asegúrese de no pasar el parámetro de índice r.**

```
function searchTable(inputVal) {
    var dataTable = $("#report_table").dataTable();
    dataTable.fnFilter(inputVal);
}
```

Lea Cuadro de entrada de búsqueda de datos para una búsqueda en tiempo real en línea:
<https://riptutorial.com/es/datatables/topic/6806/cuadro-de-entrada-de-busqueda-de-datos-para-una-busqueda-en-tiempo-real>

Capítulo 5: datatables - opción Mostrar filas seleccionadas

Examples

Mostrar solo las filas seleccionadas

Es común que las tablas de datos tengan una casilla de verificación para seleccionar varias filas. Si los datos se distribuyen en varias páginas, puede ser difícil para el usuario ver los registros que seleccionó. Para permitir que el usuario vea todos los registros seleccionados de una sola vez, usualmente usamos un hipervínculo que, cuando se hace clic, muestra solo las filas seleccionadas de la base de datos. Este enlace se puede utilizar para alternar entre la visualización de registros seleccionados y todos los registros.

```
$(sAnchor).click(function() {
    $('#show_selected').text(function(_,txt) {
        var checked = 0;
        var ret='';
        var dataTable = $("#report_table").dataTable();
        if ( txt == 'Show Selected Reports' ) {
            dataTable.fnFilter('Checked',8);
            ret = 'Show All Reports';
        }else{
            dataTable.fnFilter('',8);
            ret = 'Show Selected Reports';
        }
        return ret;
    });
});
```

En el método anterior, al hacer clic en el hipervínculo Seleccionar todo, si el texto div del hipervínculo es "Mostrar informes seleccionados", filtramos la base de datos para mostrar solo aquellas filas en las que la casilla está marcada. Usamos la función incorporada de la API de **Datatables fnFilter** .

Pasamos 2 parámetros a este método: la cadena de consulta y el índice de la columna a filtrar. En este caso, el valor de la casilla de verificación se **'Verificará'** si se selecciona en la interfaz de usuario y el índice de la columna que contiene las casillas de verificación es 8. Por lo tanto, estamos pasando 'Verificados' y 8 como parámetros a la función fnFilter. Después del filtro, cambiamos el enlace para mostrar "Mostrar todos los informes".

Cuando el usuario hace clic en Mostrar todos los informes, pasamos una cadena vacía a la función fnFilter como la cadena de consulta. Así se muestra todos los registros.

Lea [datatables - opción Mostrar filas seleccionadas en línea](https://riptutorial.com/es/datatables/topic/6805/datatables---opcion-mostrar-filas-seleccionadas):

<https://riptutorial.com/es/datatables/topic/6805/datatables---opcion-mostrar-filas-seleccionadas>

Capítulo 6: Migración de <1.10 a 1.10 y superior

Introducción

Datatables 1.10.x es la última versión a partir de ahora. A pesar de que es compatible con versiones anteriores (1.9, etc.), es altamente recomendable usar la última versión que devuelve directamente un objeto `datatable api` **ser** `datatable api`. Otro cambio importante, que es el más visible, es el cambio de [la notación húngara](#) a `camelCase`

Sintaxis

- El único cambio importante en la sintaxis es el uso de `camelCase` en todas partes en lugar de la notación húngara.
- [Puede encontrar una guía](#) más [detallada aquí](#) que implica la conversión de parámetros en <1.10 a 1.10+

Observaciones

Para más detalles, intente visitar las siguientes páginas:

1. [Actualización a Datatables 1.10](#)
2. [Datos de registro de registro de cambios 1.10](#)
3. [Convertir nombres de parámetros de 1.9 a 1.10](#)
4. [Usando la API de datos](#)

Examples

Inicialización de Datatable 1.10+

Anteriormente, los datos se inicializaron de la siguiente manera:

```
var oTable = $("#selector").dataTable();
```

Esto se utiliza para devolver un objeto jQuery que se almacenaría en la variable `tabla`. Y luego, para acceder a la `api` para modificar las propiedades de la tabla, tuvimos que inicializar la `api` de manera diferente, como se muestra a continuación:

```
var api = oTable.api()  
//r  
var api = new $.fn.dataTable.api("#selector")
```

Ahora, la inicialización se cambia a lo siguiente:

```
var table = $("#selector").DataTable();
```

Tenga en cuenta que esto devuelve una instancia de `datatable api` y puede utilizar directamente la `table` variables para manipular las propiedades de datos.

Características No disponible en Datatables 1.10+

Las siguientes 3 características (que quedaron en desuso en 1.9) ya no están disponibles en 1.10, son:

1. **fnRender** : De acuerdo con el developer:

La antigua opción `fnRender` proporcionaba un método para manipular una celda cuando se creó. sin embargo, se proporcionó una confusa lista de opciones como sus argumentos y requirió una estructura particular en `DataTables` internamente que causó problemas de rendimiento. La eliminación de `fnRender` ha conducido a una mejora significativa en el rendimiento de `DataTables` con grandes conjuntos de datos y la capacidad de proporcionar instancias de objetos a `DataTables` como objetos de origen de datos (por ejemplo, objetos observables `Knockout`).

Las alternativas a `fnRender` están disponibles como `columns.render` y `columns.createdCell`

2. **bScrollInfinite** : Según el desarrollador:

La capacidad incorporada de `DataTables` 1.9 para mostrar una cuadrícula de desplazamiento infinito a través de la opción `bScrollInfinite` se ha eliminado debido a las inconsistencias que causó en la API. La eliminación también ha ayudado simplemente al código interno de manera significativa.

Una extensión que lleva el nombre de `Scroller` está disponible como alternativa.

3. **Ahorro de estado basado en cookies** :

El ahorro de estado basado en cookies se ha reemplazado con el almacenamiento de estado basado en almacenamiento local en `DataTables` 1.10. `Cookie's`, con su límite de 4KiB, era muy limitado e incurrió en una penalización de rendimiento ya que formaba parte de cada solicitud HTTP. `LocalStorage` es mucho más rápido y más flexible, y se usa como el almacenamiento predeterminado para la información de estado en `DataTables` 1.10.

4. **two_button paginación de dos** `two_button` :

`DataTables` 1.10 ha mejorado significativamente los controles de paginación de `DataTables` (ver `pagingType`), una consecuencia de la cual es que se eliminó la antigua forma integrada de paginación `two_button`.

Se han ocupado de las personas que todavía quieren usar el método de paginación de dos `two_button` al proporcionar un archivo javascript adicional llamado `two_button.js`. El uso es el siguiente:

Simplemente incluya este archivo en su documento, después de cargar las Tablas de datos, pero antes de inicializar su tabla y la paginación de dos botones se restaurará exactamente como estaba en 1.9 (incluidos los nombres de clase, etc.).

Lea Migración de <1.10 a 1.10 y superior en línea:

<https://riptutorial.com/es/datatables/topic/8369/migracion-de--lt-1-10-a-1-10-y-superior>

Capítulo 7: Procesamiento de datos del lado del servidor

Examples

Cargar datos usando ajax con procesamiento del lado del servidor.

```
var MY_AJAX_ACTION_URL = "path/to/controller.php";

var table = $('#user_list_table').DataTable({
    "autoWidth": true,
    "paging": true,
    "searching": true,
    "ordering": true,
    "language": {
        "zeroRecords": "No data Found",
        "processing": 'Loading'
    },
    "info": false,
    "stripeClasses": [ "odd nutzer_tr", "even nutzer_tr"],
    "columns": [
        { 'data': 'uid', 'visible': false },
        { 'data': 'name', 'orderable': true },
        { 'data': 'phone', 'orderable': true },
        { 'data': 'email', 'orderable': true },
        { 'data': 'address', 'orderable': true }
    ],
    "order": [[ 1, "desc" ]],
    "processing": true,
    "serverSide": true,
    "ajax": MY_AJAX_ACTION_URL
});
```

La respuesta de la llamada a MY_AJAX_ACTION_URL debe estar estrictamente en el siguiente formato:

```
{
  "draw": 1,
  "recordsTotal": 2,
  "recordsFiltered": 2,
  "data": [
    { "name": "XYZ", "phone": "678654454", "email": "xyz@gmail.com", "address": "true" },
    { "name": "ABC", "phone": "678654455", "email": "abc@gmail.com", "address": "true" }
  ]
}
```

Tenga en cuenta que si la salida de la llamada no coincide con el formato anterior, se producirá un error en la inicialización de la `table`.

DataTables 1.10+ Serverside Processing

Tabla de ejemplo

Hay varias formas de inyectar sus datos en DataTables. Serverside Processing es solo un método. De esta manera, DataTables tiene un punto final preconfigurado para recuperar datos, y ese punto final es responsable de aceptar todas las solicitudes de paginación / filtrado / clasificación a las que se aplica DataTables. Hay una variedad de ventajas y desventajas para esto, en lugar de enviar su conjunto de datos completo desde el servidor y dejar que DataTables lo haga todo del lado del cliente, dependiendo de su caso de uso.

```
var tbl = $('#example').DataTable({
  processing: true,
  serverSide: true,
  ajax: {
    url: '/echo/json/',
    method: 'post'
  },
  columns: [{
    data: 'First',
    title: 'First Name'
  }, {
    data: 'Last',
    title: 'Last Name'
  }]
});
```

Ejemplo de enlace del evento preXhr para enviar datos adicionales a una solicitud ajax

Este evento se activa directamente antes de que se realice la llamada ajax, lo que le permite modificar el cuerpo de la solicitud. Esto es útil si hay un formulario que influye en los datos devueltos a la tabla (prevea tal vez una barra de mínimo / máximo para filtrar dentro de un rango de fechas o similar).

```
tbl.on('preXhr.dt', function(ev, settings, data) {
  $.extend(data, {
    min: $('form [name=min]').val(),
    max: $('form [name=max]').val()
  });
});
```

Explicación de los requisitos del lado del servidor para la solicitud de procesamiento

En una instancia típica sin opción de DataTables, todo el filtrado, la clasificación y la paginación se manejan en el navegador del cliente. Con Serverside Processing habilitado, estas tareas se desplazan al servidor web. Para conjuntos de datos muy grandes que pueden ser ineficientes para enviar en su totalidad al cliente, esto puede ayudar.

Hay varios parámetros predeterminados que son enviados por la solicitud de Datos de Datos cuando configura un punto final ajax. Puede llegar a ser muy largo, así que en lugar de detallar cada uno, una descripción general de las responsabilidades del servidor puede ser más útil.

Comenzando con cualquier parámetro opcional que haya proporcionado a la solicitud, compile su conjunto de datos y prepárelo para las operaciones de Tablas de datos. Utilice los parámetros de `search[value]` y `search[regex]` para aplicar cualquier filtrado en todas las columnas / propiedades. También hay `columns[i][search][value]` y `columns[i][search][regex]` para los filtros de columnas

individuales, aunque estos no se usan comúnmente en instancias simples de DataTables.

Ordene sus datos filtrados utilizando los diversos parámetros de `order`. Al igual que con los parámetros de búsqueda, habrá un conjunto de parámetros de ordenamiento por columna en los que se habilita la clasificación. `order[i][column]` y `order[i][column][dir]`

Una vez que se completa el filtrado y la clasificación, es hora de paginar los datos según la solicitud de DataTables utilizando los parámetros de `start` y `length`. En .NET esto podría parecer:

```
int start = Int32.TryParse(Request["start"]);
int length = Int32.TryParse(Request["length"]);
return MyData.Skip(start).Take(length);
```

Ejemplo de respuesta

Esta es la estructura de respuesta aproximada que DataTables espera. Cualquiera que sea su información (la matriz 2d, la matriz de objetos, etc.) está anidada en la propiedad de `data` con las otras propiedades que integran el núcleo de DataTable para obtener información sobre la paginación y el filtrado (por ejemplo, "Mostrar registros 11-20 de 500 (filtrados de 1000)")

```
{
  "draw": 1,
  "recordsTotal": 57,
  "recordsFiltered": 57,
  "data": [/*your data goes here*/]
}
```

También vea `.rows.add(data)` y la [opción de data](#) para métodos alternativos de configuración del contenido de su DataTable usando datos JSON. Por supuesto, las tablas de datos también se pueden [inicializar a partir de HTML estático o atributos de data-HTML5](#)

Obtener datos JSON de la tabla MySQL

En el sitio web oficial de DataTable hay un [ejemplo](#) de cómo puede verse un proceso del lado del servidor con PHP y MySQL. Este ejemplo está en **desuso** y ya no se puede usar con PHP 7 (la función "mysql_pconnect" y las funciones asociadas están en desuso, consulte esta [publicación](#)).

Así que esta función le proporciona datos JSON bien formados como respuesta:

```
<?php
//include database connection file
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "name";

$conn = mysqli_connect($servername, $username, $password, $dbname) or die("Connection failed:
" . mysqli_connect_error());

/* check connection */
if (mysqli_connect_errno()) {
    printf("Connect failed: %s\n", mysqli_connect_error());
```

```

    exit();
}

// initilize all variable
$params = $columns = $totalRecords = $data = array();
$params = $_REQUEST;
//define index of column name
$columns = array(
    0 => 'id',
    1 => 'name',
    2 => 'salery',
);

$where = $sqlTot = $sqlRec = "";

// check search value exist
if( !empty($params['search']['value']) ) {
    $where .= " WHERE ";
    $where .= " ( id LIKE '". $params['search']['value']. "%' ";
    $where .= " OR name LIKE '". $params['search']['value']. "%' ";
    $where .= " OR salery LIKE '". $params['search']['value']. "%' )";
}

// getting total number records without any search
$sql = "SELECT * FROM `employees` ";
$sqlTot .= $sql;
$sqlRec .= $sql;

//concatenate search sql if value exist
if(isset($where) && $where != '') {
    $sqlTot .= $where;
    $sqlRec .= $where;
}

$sqlRec .= " ORDER BY ". $columns[$params['order'][0]['column']]."
". $params['order'][0]['dir']. " LIMIT ". $params['start']. " , ". $params['length']. " ";

$queryTot = mysqli_query($conn, $sqlTot) or die("database error:". mysqli_error($conn));

$totalRecords = mysqli_num_rows($queryTot);

$queryRecords = mysqli_query($conn, $sqlRec) or die("error to fetch employees data");

while( $row = mysqli_fetch_row($queryRecords) ) {
    $data[] = $row;
}

$json_data = array(
    "draw"           => intval( $params['draw'] ),
    "recordsTotal"   => intval( $totalRecords ),
    "recordsFiltered" => intval($totalRecords),
    "data"           => $data    // total data array
);

echo json_encode($json_data); // send data as json format
?>

```

La respuesta se ve y puede ser procesada por el DataTable:

```
{
```

```
"draw": 1,  
"recordsTotal": 3,  
"recordsFiltered": 2,  
"data": [  
  {  
    "id": "1",  
    "name": "Jim",  
    "salary": "1000"  
  },  
  {  
    "id": "2",  
    "name": "Claudia",  
    "salary": "3000"  
  },  
  {  
    "id": "3",  
    "name": "Tommy",  
    "salary": "2000"  
  }  
]  
}
```

Lea Procesamiento de datos del lado del servidor en línea:

<https://riptutorial.com/es/datatables/topic/4176/procesamiento-de-datos-del-lado-del-servidor>

Creditos

S. No	Capítulos	Contributors
1	Empezando con los datos	andialles , Andrei Zhytkevich , Chris H. , Community , cookypuss
2	¿Cómo obtener el valor de búsqueda ingresado en Datatables mediante programación?	Domenico Campagnolo
3	Agregar botones de exportación a la tabla en Bootstrap 4	Martin
4	Cuadro de entrada de búsqueda de datos para una búsqueda en tiempo real	saikris
5	datatables - opción Mostrar filas seleccionadas	saikris
6	Migración de <1.10 a 1.10 y superior	philantrovert
7	Procesamiento de datos del lado del servidor	Aswathy S , BLSully , Martin , philantrovert