



**Kostenloses eBook**

# LERNEN datatables

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#datatables**

# Inhaltsverzeichnis

Über.....	1
<b>Kapitel 1: Erste Schritte mit Datentabellen.....</b>	<b>2</b>
Bemerkungen.....	2
Versionen.....	2
Examples.....	2
Installation.....	3
Initialisieren einer minimalen DataTables:.....	3
Funktion aktivieren / deaktivieren (DataTables-Optionen).....	4
DataTables-API.....	4
<b>Kapitel 2: datatables - Option "Ausgewählte Zeilen anzeigen".....</b>	<b>6</b>
Examples.....	6
Nur ausgewählte Zeilen anzeigen.....	6
<b>Kapitel 3: Datatables-Suchfeld für Echtzeitsuche.....</b>	<b>7</b>
Examples.....	7
Suchfeld für die progressive Suche auf der Datentabelle.....	7
<b>Kapitel 4: Fügen Sie in Bootstrap 4 Exportschaltflächen zur Tabelle hinzu.....</b>	<b>8</b>
Einführung.....	8
Examples.....	8
Fügen Sie der Tabelle Schaltflächen hinzu.....	8
<b>Kapitel 5: Migration von &lt;1.10 auf 1.10 und höher.....</b>	<b>10</b>
Einführung.....	10
Syntax.....	10
Bemerkungen.....	10
Examples.....	10
Initialisierung von Datatable 1.10+.....	10
Features In DataTables 1.10+ nicht verfügbar.....	11
<b>Kapitel 6: Serverseitige Datenverarbeitung.....</b>	<b>13</b>
Examples.....	13
Laden Sie Daten mit ajax mit serverseitiger Verarbeitung.....	13
DataTables 1.10+ Serverseitige Verarbeitung.....	13

JSON-Daten aus MySQL-Tabelle abrufen.....	15
<b>Kapitel 7: Wie kann der Suchwert programmgesteuert in Datatables eingegeben werden?.....</b>	<b>18</b>
Examples.....	18
Beispiel.....	18
<b>Credits.....</b>	<b>19</b>



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](mailto:info@zzzprojects.com)

# Kapitel 1: Erste Schritte mit Datentabellen

## Bemerkungen

### Hauptmerkmale

- Variable Paginierung
- Sofortige Filterung
- Mehrspaltensortierung mit Datentyperkennung
- Intelligente Handhabung von Säulenbreiten
- Zeigen Sie Daten aus nahezu jeder Datenquelle an
- DOM, JavaScript-Array, Ajax-Datei und serverseitige Verarbeitung
- Bildlaufoptionen für das Tabellendarstellungsfenster
- Vollständig internationalisierbar
- Unterstützung für jQuery UI ThemeRoller
- Große Auswahl an Plug-Ins
- Es ist kostenlos

### Ressourcen

- [Verwendungszweck](#)
- [Beispiele](#)
- [Styling](#)
- [API](#)
- [Entwicklung](#)
- [Extras](#)
- [Plugins](#)
- [Blog](#)
- [FAQ's](#)
- [Foren](#)
- [Serverseitige Beispiele](#)

## Versionen

Ausführung	Versionshinweise	Veröffentlichungsdatum
1.9 und früher (Erbe)	<a href="https://datatables.net/forums/discussion/8332/datatables-1-9-0-veroeffentlicht">https://datatables.net/forums/discussion/8332/datatables-1-9-0-veroeffentlicht</a>	2014-02-01
1.10 und später	<a href="https://datatables.net/new/1.10">https://datatables.net/new/1.10</a>	2015-08-12

## Examples

## Installation

Haben Sie die erforderlichen JavaScript und CSS Dateien in Ihrer `index.html`. Sie können dies tun, indem Sie entweder die CDN-Dateien verwenden, die unter den folgenden Pfaden verfügbar sind:

```
<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>
```

Oder indem Sie einzelne lokale Dateien herunterladen und selbst hosten. Um ein umfangreiches Paket erforderlicher JavaScript und CSS Dateien zu erhalten, besuchen Sie den [DataTables-Download-Builder](#). Hier können Sie die gewünschten Funktionen auswählen und sie in einem einzigen Paket zusammenfassen (oder einzelne Dateien anbieten). Fügen Sie diese in der Reihenfolge hinzu, die am unteren Rand der Seite angezeigt wird.

DataTables hängt von `jQuery` `jquery.dataTables.js` es daher vor `jquery.dataTables.js` :

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

DataTables sind auch über NPM verfügbar

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

und Bower

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

## Initialisieren einer minimalen DataTables:

Der folgende Code `tableid` die Tabelle mit der ID der `tableid` in eine DataTable um und gibt eine DataTables-API-Instanz zurück:

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

Vergleichen Sie dies mit dem folgenden Code, der die Tabelle in eine DataTable umwandelt, jedoch keine DataTables-API-Instanz zurückgibt:

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

Weitere Informationen dazu, was mit der DataTables-API-Instanz möglich ist, finden Sie im Abschnitt zur [Dokumentation](#) der DataTables-API.

## Funktion aktivieren / deaktivieren (DataTables-Optionen)

DataTables kann eine Reihe von Funktionen aktivieren oder deaktivieren, z. B. Paging oder Suche. Um diese Optionen auszuwählen, wählen Sie sie einfach in Ihrer Initialisierung aus:

```
$(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
  } );
});
```

Beachten Sie, dass die Anführungszeichen um die Optionsnamen optional sind:

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

Ist auch vollkommen gültig.

Eine vollständige Liste der Optionen gefunden werden kann [hier](#) , zusammen mit Beschreibungen der Verwendungen der einzelnen Optionen.

Diese Optionen können nur einmal festgelegt werden, wenn die Tabelle initialisiert wird. Sie können diese Einschränkung jedoch umgehen, indem Sie Folgendes hinzufügen:

```
destroy: true
```

## DataTables-API

DataTables enthält eine umfangreiche API, mit der Informationen zu den DataTables auf einer Seite bearbeitet oder abgerufen werden können.

Auf die API kann auf drei Arten zugegriffen werden:

```
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');
```

Nachdem das Objekt festgelegt wurde, können Sie alle API-Funktionen für dieses Objekt aufrufen.

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

Ein komplexeres Beispiel ist das [Hinzufügen einiger Zeilen](#) zu Ihrer Tabelle:

```
table.rows.add( [ {
  "name":      "John Doe",
  "employee_id": "15135",
  "department": "development",
```

```
}, {  
  "name":      "Jane Smith",  
  "employee_id": "57432",  
  "department": "quality assurance",  
} ] )  
.draw();
```

Die vollständige Liste der API-Funktionen finden Sie [hier](#) .

Erste Schritte mit Datentabellen online lesen: <https://riptutorial.com/de/datatables/topic/1844/erste-schritte-mit-datentabellen>



# Kapitel 2: datatables - Option "Ausgewählte Zeilen anzeigen"

## Examples

### Nur ausgewählte Zeilen anzeigen

Es ist üblich, dass Datentabellen ein Kontrollkästchen zum Auswählen mehrerer Zeilen haben. Wenn die Daten über mehrere Seiten verteilt sind, kann es für den Benutzer schwierig sein, die von ihm ausgewählten Datensätze anzuzeigen. Damit der Benutzer alle ausgewählten Datensätze auf einmal sehen kann, verwenden wir normalerweise einen Hyperlink, der beim Klicken nur die ausgewählten Zeilen der Datentabelle anzeigt. Dieser Link kann verwendet werden, um zwischen ausgewählten Datensätzen und allen Datensätzen umzuschalten.

```
$(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;
    });
});
```

Wenn Sie in der obigen Methode auf den Hyperlink Alle auswählen klicken: Wenn der div-Text des Hyperlinks 'Ausgewählte Berichte anzeigen' enthält, filtern Sie die Datentabelle so, dass nur die Zeilen angezeigt werden, für die das Kontrollkästchen aktiviert ist. Wir verwenden die eingebaute Datatables-API-Funktion **fnFilter** .

Wir übergeben dieser Methode zwei Parameter - Abfragezeichenfolge und den Index der zu filternden Spalte. In diesem Fall wird der Wert des Kontrollkästchens **'Markiert' sein**, wenn es auf der Benutzeroberfläche ausgewählt ist und der Index der Spalte, die die Ankreuzfelder enthält, 8 ist. Daher übergeben wir 'Checked' und 8 als Parameter an die Funktion fnFilter. Nach dem Filter schalten wir den Link um, um "Alle Berichte anzeigen" anzuzeigen.

Wenn der Benutzer auf Alle Berichte anzeigen klickt, übergeben wir eine leere Zeichenfolge als Abfragezeichenfolge an die Funktion fnFilter. So werden alle Datensätze angezeigt.

[datatables - Option "Ausgewählte Zeilen anzeigen" online lesen:](https://riptutorial.com/de/datatables/topic/6805/datatables---option--ausgewahlte-zeilen-anzeigen-)

<https://riptutorial.com/de/datatables/topic/6805/datatables---option--ausgewahlte-zeilen-anzeigen->

# Kapitel 3: Datatables-Suchfeld für Echtzeitsuche

## Examples

### Suchfeld für die progressive Suche auf der Datentabelle

Im Folgenden finden Sie ein Beispiel zum Implementieren eines Sucheingabefelds, das Benutzern hilft, die Vorkommen eines bestimmten Werts in der Datentabelle zu durchsuchen.

In dem folgenden Beispiel ist #report die Div-ID des Div, das das Sucheingabefeld enthält. Diese Funktion wird aufgerufen, sobald der Benutzer einen Wert in dieses Eingabefeld eingibt. Da es viele Vorkommen eines einzelnen Zeichens geben kann, rufen wir die eigentliche Suchfunktion nur auf, wenn mehr als ein Zeichen in das Suchfeld eingegeben wird.

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

In der searchTable-Funktion verwenden wir die eingebaute Datatables-Funktion **fnFilter** , um die übereinstimmenden Vorkommen der **Eingabezeichenfolge** zu finden. **Wir können die Suche auf eine bestimmte Spalte einschränken, indem Sie den Spaltenindex übergeben.** Hier übergeben wir den Spaltenindex 2.

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

Wenn Sie alle Spalten durchsuchen müssen, **stellen Sie sicher, dass Sie nicht den Indexparameter r übergeben.**

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

Datatables-Suchfeld für Echtzeitsuche online lesen:

<https://riptutorial.com/de/datatables/topic/6806/datatables-suchfeld-fur-echtzeitsuche>

---

# Kapitel 4: Fügen Sie in Bootstrap 4 Exportschaltflächen zur Tabelle hinzu

## Einführung

Mit dem Datatables-Plugin können Sie Ihrer Tabelle Exportschaltflächen hinzufügen.

Sie können Ihre Tabellendaten nach Excel exportieren, als PDF exportieren oder in die Zwischenablage kopieren.

Dieses Handbuch ist für das Bootstrap 4-Framework gedacht.

## Examples

### Fügen Sie der Tabelle Schaltflächen hinzu

Fügen Sie in Ihrer JS-Datei diese **Option** zu Ihrer Datentabelle hinzu:

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

Es wird so aussehen:

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

Fügen Sie die erforderlichen **CSS**-Dateien für die Datentabelle mit den Schaltflächen hinzu:

```
<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"/>
```

Fügen Sie die erforderlichen **Javascript**-Dateien für die Datentabelle mit den Schaltflächen hinzu:

```
<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>
```

Es wird wie dieses Bild aussehen:



Wenn Sie die Schaltflächen nicht sehen, fügen Sie diese Option hinzu:

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

zur datierbaren Optionsliste. So sieht es aus:

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

Sie finden weitere Informationen, um festzulegen, welche Steuerelemente für die Tabelle auf der Seite erscheinen sollen und in welcher Reihenfolge auf dieser [Seite](#) .

**Hinweis:** Voraussetzung ist, dass die Dateien jQuery und bootstrap4 in Ihrem Projekt installiert sind.

Fügen Sie in Bootstrap 4 Exportschaltflächen zur Tabelle hinzu online lesen:

<https://riptutorial.com/de/datatables/topic/10149/fugen-sie-in-bootstrap-4-exportschaltflachen-zur-tabelle-hinzu>

---

# Kapitel 5: Migration von <1.10 auf 1.10 und höher

## Einführung

Datatables 1.10.x ist ab sofort die neueste Version. Obwohl es abwärtskompatibel zu den Vorgängerversionen (1.9 usw.) ist, wird dringend empfohlen, die neueste Version zu verwenden, die direkt ein `datatable api` Objekt zurückgibt. Eine weitere wichtige Änderung, die am deutlichsten sichtbar ist, ist die Änderung der [ungarischen Notation](#) in [camelCase](#)

## Syntax

- Die einzige wesentliche Änderung der Syntax ist die Verwendung von camelCase überall anstelle der ungarischen Notation.
- Eine [detailliertere Anleitung finden Sie hier](#), die die Umwandlung von Parametern in <1.10 auf 1.10+ beinhaltet

## Bemerkungen

Weitere Informationen erhalten Sie auf den folgenden Seiten:

1. [Upgrade auf Datenblätter 1.10](#)
2. [Änderungsprotokoll 1.10](#)
3. [Konvertieren Sie Parameternamen von 1.9 in 1.10](#)
4. [Verwenden der Datatable-API](#)

## Examples

### Initialisierung von Datatable 1.10+

Bisher wurden die Datentabellen wie folgt initialisiert:

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

Dies wurde verwendet, um ein jQuery-Objekt zurückzugeben, das in der Variablen oTable gespeichert wurde. Und um dann auf die API zuzugreifen, um die Tabelleneigenschaften zu ändern, mussten wir die API anders initialisieren, wie unten gezeigt:

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

Nun wird die Initialisierung wie folgt geändert:

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

Beachten Sie, dass dies eine `datatable api` - Instanz und Sie direkt Variable verwenden können `table` Datatable Eigenschaften zu manipulieren.

## Features In Datatables 1.10+ nicht verfügbar

Die folgenden 3 Funktionen (die in 1.9 nicht mehr unterstützt wurden) sind in 1.10 nicht mehr verfügbar. Sie sind:

### 1. **fnRender** : Laut Developer:

Die alte Option `fnRender` bot eine Methode zum Bearbeiten einer Zelle, wenn diese erstellt wurde. Es wurde jedoch eine verwirrende Liste von Optionen als Argumente bereitgestellt und erforderte intern eine bestimmte Struktur in DataTables, die Leistungsprobleme verursachte. Das Entfernen von `FnRender` hat zu einer erheblichen Verbesserung der Leistung von DataTables mit großen Datensätzen und der Möglichkeit geführt, DataTables Objektinstanzen als Datenquellenobjekte bereitzustellen (z. B. beobachtbare Knockout-Objekte).

Alternativen zu `fnRender` sind als `columns.render` und `columns.createdCell`

### 2. **bScrollInfinite** : Laut Entwickler:

Die eingebaute Fähigkeit von DataTables 1.9, ein unendlich langes Raster mit der Option `bScrollInfinite` anzuzeigen, wurde aufgrund der Inkonsistenzen in der API entfernt. Das Entfernen hat auch einfach dazu beigetragen, den internen Code erheblich zu verbessern.

Als Alternative steht eine Erweiterung mit dem Namen `scroller` zur Verfügung.

### 3. **Cookie-basierte Zustandsspeicherung** :

Die Speicherung von Cookies auf Basis von Status wurde in DataTables 1.10 durch die Speicherung von Status auf Basis von LocalStorage ersetzt. Cookies waren mit ihrem 4KiB-Limit sehr begrenzt und hatten einen Performance-Nachteil, da sie Teil jeder HTTP-Anfrage waren. LocalStorage ist viel schneller und flexibler und wird als Standardspeicher für Statusinformationen in DataTables 1.10 verwendet.

### 4. `two_button` **Paginierungskontrolle** :

DataTables 1.10 hat die Paging-Steuererelemente von DataTables (siehe `pagingType`) erheblich aktualisiert. Dies hat zur Folge, dass die alte, eingebaute `Two_button`-Form von Paging entfernt wurde.

Sie haben sich um Menschen gekümmert, die noch die `two_button` Paginierungsmethode verwenden `two_button`, indem sie eine zusätzliche Javascript-Datei namens `two_button.js`. Die Verwendung ist wie folgt:

Fügen Sie diese Datei einfach in Ihr Dokument ein, nachdem Sie DataTables geladen haben, aber bevor Sie Ihre Tabelle initialisieren, und die Seitenauslagerung mit zwei\_button wird genauso wie in 1.9 wiederhergestellt (einschließlich Klassennamen usw.).

Migration von <1.10 auf 1.10 und höher online lesen:

<https://riptutorial.com/de/datatables/topic/8369/migration-von--lt-1-10-auf-1-10-und-hoher>

# Kapitel 6: Serverseitige Datenverarbeitung

## Examples

Laden Sie Daten mit ajax mit serverseitiger Verarbeitung.

```
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
});
```

Die Antwort des Aufrufs an `MY_AJAX_ACTION_URL` sollte streng im folgenden Format sein:

```
{
  "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" }
  ]
}
```

Wenn die Ausgabe des Aufrufs nicht mit dem obigen Format übereinstimmt, führt dies zu einem Fehler bei der Initialisierung der `table`.

## DataTables 1.10+ Serverseitige Verarbeitung

### Beispieltabelle

Es gibt verschiedene Möglichkeiten, Ihre Daten in DataTables einzubringen. Serverseitige



Verarbeitung ist nur eine Methode. Auf diese Weise verfügt DataTables über einen vorkonfigurierten Endpunkt, von dem Daten abgerufen werden können, und dieser Endpunkt ist dafür verantwortlich, alle Paging-, Filter- und Sortieranforderungen zu akzeptieren, auf die DataTables angewendet wird. Es gibt eine Reihe von Vor- und Nachteilen, anstatt das komplette Dataset vom Server zurückzusenden und DataTables dies alles clientseitig tun zu lassen, abhängig von Ihrem Anwendungsfall.

```
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'
  }]
});
```

### Beispiel für das Hook-Ereignis `preXhr` zum Senden zusätzlicher Daten an eine Ajax-Anforderung

Dieses Ereignis wird direkt vor dem Aufruf von Ajax ausgelöst, sodass Sie den Anfragetext ändern können. Dies ist nützlich, wenn es ein Formular gibt, das die in die Tabelle zurückgegebenen Daten beeinflusst (z. B. eine Min- / Max-Leiste zum Filtern innerhalb eines Datumsbereichs oder ähnliches).

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

### Erläuterung der serverseitigen Anforderungen für die Verarbeitung der Anforderung

In einer typischen Instanz von DataTables ohne Option werden das gesamte Filtern, Sortieren und Paging im Clientbrowser ausgeführt. Wenn die Verarbeitung von Serverseiten aktiviert ist, werden diese Aufgaben auf den Webserver verlagert. Für sehr große Datensätze, die möglicherweise nicht vollständig an den Client gesendet werden, kann dies hilfreich sein.

Es gibt mehrere Standardparameter, die von der Datatables-Anforderung gesendet werden, wenn Sie einen Ajax-Endpunkt konfigurieren. Es kann sehr lang werden, daher kann ein allgemeiner Überblick über die Server-Zuständigkeiten hilfreicher sein, als jede einzelne aufzuschreiben.

Beginnen Sie mit beliebigen optionalen Parametern, die Sie der Anfrage bereitgestellt haben, kompilieren Sie Ihr Dataset und bereiten Sie es für DataTables-Vorgänge vor. Verwenden Sie die Parameter `search[value]` und `search[regex]`, um eine Filterung auf alle Spalten / Eigenschaften

anzuwenden. Es gibt auch `columns[i][search][value]` und `columns[i][search][regex]` für einzelne Spaltenfilter. Diese werden jedoch normalerweise nicht in einfachen Instanzen von DataTables verwendet.

Sortieren Sie Ihre gefilterten Daten anhand der verschiedenen `order`. Wie bei den Suchparametern gibt es eine Reihe von Sortierparametern pro Spalte, für die die Sortierung aktiviert ist. `order[i][column]` und `order[i][column][dir]`

Nach Filterung und Sortierung abgeschlossen ist, ist es Zeit, auf Seite der Daten auf Basis der Datentabellen Anforderung der mit `start` und `length` Parametern. In .NET könnte das so aussehen:

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

## Beispielantwort

Dies ist die grobe Antwortstruktur, die DataTables erwartet. Was auch immer Ihre Daten sind (2d-Array, Array von Objekten usw.), ist in der `data`-Eigenschaft verschachtelt, wobei die anderen Eigenschaften den DataTable-Kern bilden, um Informationen zum Seitenwechsel und zum Filtern zu zeichnen (z. )

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

Siehe auch `.rows.add(data)` und die `data` [Option](#) für alternative Methoden zum Festlegen des Inhalts Ihrer DataTable mithilfe von JSON-Daten. Natürlich können DataTables auch mit [statischen HTML- oder HTML5- data- initialisiert werden](#)

## JSON-Daten aus MySQL-Tabelle abrufen

Auf der offiziellen Website von DataTable finden Sie ein [Beispiel dafür](#), wie ein serverseitiger Prozess mit PHP und MySQL aussehen kann. Dieses Beispiel ist **veraltet** und kann nicht mehr mit PHP 7 verwendet werden (die Funktion `mysql_pconnect` und die zugehörigen Funktionen sind veraltet, siehe diesen [Beitrag](#) ).

Diese Funktion gibt Ihnen also als Antwort eine gute JSON-Daten:

```
<?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
?>

```

Die Antwort sieht wie aus und kann dann von der DataTable verarbeitet werden:

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

Serverseitige Datenverarbeitung online lesen:

<https://riptutorial.com/de/datatables/topic/4176/serverseitige-datenverarbeitung>

---

# Kapitel 7: Wie kann der Suchwert programmgesteuert in Datatables eingegeben werden?

## Examples

### Beispiel

Dies ist der Code, um die Datatables [1.10.7] programmgesteuert nach Werten zu filtern. Sie finden ihn in der offiziellen Dokumentation.

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

Dies ist der Code, um den Wert bei der vorherigen Suche abzurufen.

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

Diese Vorgehensweise ist nützlich, wenn der Cache aktiv ist ( "*stateSave*": *true* ) und Sie den vorherigen *Suchwert* nach dem Neuladen der Seite kennen müssen.

Wie kann der Suchwert programmgesteuert in Datatables eingegeben werden? online lesen: <https://riptutorial.com/de/datatables/topic/6085/wie-kann-der-suchwert-programmgesteuert-in-datatables-eingegeben-werden->

# Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit Datentabellen	<a href="#">andialles</a> , <a href="#">Andrei Zhytkevich</a> , <a href="#">Chris H.</a> , <a href="#">Community</a> , <a href="#">cookypuss</a>
2	datatables - Option "Ausgewählte Zeilen anzeigen"	<a href="#">saikris</a>
3	Datatables-Suchfeld für Echtzeitsuche	<a href="#">saikris</a>
4	Fügen Sie in Bootstrap 4 Exportschaltflächen zur Tabelle hinzu	<a href="#">Martin</a>
5	Migration von <1.10 auf 1.10 und höher	<a href="#">philantrovert</a>
6	Serverseitige Datenverarbeitung	<a href="#">Aswathy S</a> , <a href="#">BLSully</a> , <a href="#">Martin</a> , <a href="#">philantrovert</a>
7	Wie kann der Suchwert programmgesteuert in Datatables eingegeben werden?	<a href="#">Domenico Campagnolo</a>