



 eBook Gratuit

APPRENEZ datatables

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#datatables

Table des matières

À propos	1
Chapitre 1: Démarrer avec les données	2
Remarques	2
Versions	2
Exemples	2
Installation	3
Initialiser un DataTables minimal:	3
Fonction Activer / Désactiver (Options DataTables)	4
API DataTables	4
Chapitre 2: Ajouter des boutons d'exportation à la table dans Bootstrap 4	6
Introduction	6
Exemples	6
Ajouter des boutons à la table	6
Chapitre 3: Comment obtenir la valeur de recherche entrée dans Datatables par programme? ...	8
Exemples	8
Exemple	8
Chapitre 4: Datatables - Option Afficher les lignes sélectionnées	9
Exemples	9
Afficher uniquement les lignes sélectionnées	9
Chapitre 5: Migration de <1.10 à 1.10 et plus	10
Introduction	10
Syntaxe	10
Remarques	10
Exemples	10
Initialisation de Datatable 1.10+	10
Caractéristiques Non disponible dans Datatables 1.10+	11
Chapitre 6: Traitement des données côté serveur	13
Exemples	13
Charger des données en utilisant ajax avec le traitement côté serveur	13
DataTables 1.10+ Traitement des serveurs	13

Obtenir des données JSON à partir de la table MySQL.....	15
Chapitre 7: zone de saisie de recherche de données pour une recherche en temps réel.....	18
Exemples.....	18
Zone de saisie de recherche pour une recherche progressive sur la base de données.....	18
Crédits.....	20

À propos

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

Chapitre 1: Démarrer avec les données

Remarques

Principales caractéristiques

- Pagination de longueur variable
- Filtrage à la volée
- Tri multi-colonnes avec détection de type de données
- Gestion intelligente des largeurs de colonne
- Afficher des données à partir de presque n'importe quelle source de données
- DOM, tableau JavaScript, fichier Ajax et traitement côté serveur
- Options de défilement pour la fenêtre de tableau
- Entièrement internationalisable
- Prise en charge de jQuery UI ThemeRoller
- Grande variété de plug-ins
- C'est gratuit

Ressources

- [Usage](#)
- [Exemples](#)
- [coiffant](#)
- [API](#)
- [Développement](#)
- [Extras](#)
- [Plug-ins](#)
- [Blog](#)
- [FAQ](#)
- [Forums](#)
- [Exemples côté serveur](#)

Versions

Version	Notes de version	Date de sortie
1.9 et plus tôt (héritage)	https://datatables.net/forums/discussion/8332/datatables-1-9-0-released	2014-02-01
1.10 et plus tard	https://datatables.net/new/1.10	2015-08-12

Exemples

Installation

Avoir les fichiers JavaScript et CSS requis inclus dans votre `index.html` . Vous pouvez le faire soit en utilisant les fichiers CDN disponibles sur les chemins suivants:

```
<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>
```

Ou en téléchargeant des fichiers locaux individuels et en les hébergeant vous-même. Pour obtenir un ensemble complet de fichiers JavaScript et CSS requis, visitez le [générateur de téléchargement de DataTables](#) qui vous permettra de choisir les fonctionnalités dont vous avez besoin et de les condenser dans un seul package (ou d'offrir des fichiers individuels). Incluez-les dans l'ordre affiché au bas de la page.

DataTables dépend de jQuery , alors incluez-le avant `jquery.dataTables.js` :

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

Les DataTables sont également disponibles via NPM

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

et Bower

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

Initialiser un DataTables minimal:

Le code ci-dessous transformera la table avec un identifiant de `tableid` en un DataTable, et renverra une instance de l'API DataTables:

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

Comparez cela au code ci-dessous, qui transformera la table en DataTable mais ne renverra pas d'instance d'API DataTables:

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

Consultez la section de [documentation de l'API DataTables](#) pour plus de détails sur ce qui peut être fait avec l'instance de l'API DataTables.

Fonction Activer / Désactiver (Options DataTables)

DataTables peut activer ou désactiver un certain nombre de ses fonctionnalités, telles que la pagination ou la recherche. Pour choisir ces options, sélectionnez-les simplement dans votre initialisation:

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

Notez que les guillemets entourant les noms d'options sont facultatifs:

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

Est également parfaitement valide.

Une liste complète des options peut être trouvée [ici](#) , avec des descriptions des utilisations de chaque option.

Ces options ne peuvent être définies qu'une seule fois, lorsque la table est initialisée. Cependant, vous pouvez contourner cette limitation en ajoutant:

```
destroy: true
```

API DataTables

DataTables est livré avec une API étendue qui est utilisée pour manipuler ou obtenir des informations sur les DataTables sur une page.

L'API est accessible de 3 manières:

```
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');
```

Une fois l'objet défini, vous pouvez appeler n'importe laquelle des fonctions de l'API sur cet objet.

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

Un exemple plus complexe consiste à [ajouter des lignes](#) à votre table:

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

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

La liste complète des fonctions de l'API peut être trouvée [ici](#) .

Lire Démarrer avec les données en ligne: <https://riptutorial.com/fr/datatables/topic/1844/demarrer-avec-les-donnees>

Chapitre 2: Ajouter des boutons d'exportation à la table dans Bootstrap 4

Introduction

Avec le plug-in de données, vous pouvez ajouter des boutons d'exportation à votre table.

Vous pouvez exporter vos données de table pour exceller, pdf ou les copier dans le presse-papier.

Ce manuel est destiné au framework bootstrap 4.

Exemples

Ajouter des boutons à la table

Dans votre fichier JS, ajoutez cette **option** à votre datatable:

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

Cela ressemblera à:

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

Ajoutez les fichiers **CSS** nécessaires pour le datatable avec les boutons:

```
<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"/>
```

Ajoutez les fichiers **javascript** nécessaires pour le datatable avec les boutons:

```
<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>
```

Cela ressemblera à cette image:



Si vous ne voyez pas les boutons, ajoutez cette option:

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

à la liste des options pouvant être datées. Donc, ça ressemble à:

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

Vous trouverez plus d'informations pour définir les éléments de contrôle de la table à afficher sur la page et dans quel ordre sur cette [page](#) .

Remarque: La condition préalable est que les fichiers jQuery et bootstrap4 soient installés dans votre projet.

Lire Ajouter des boutons d'exportation à la table dans Bootstrap 4 en ligne:

<https://riptutorial.com/fr/datatables/topic/10149/ajouter-des-boutons-d-exportation-a-la-table-dans-bootstrap-4>

Chapitre 3: Comment obtenir la valeur de recherche entrée dans Datatables par programme?

Exemples

Exemple

C'est le code pour filtrer les Datatables [1.10.7] par valeur par programmation, vous pouvez le trouver sur la documentation officielle.

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

C'est le code pour obtenir la valeur par la recherche précédente.

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

Cette approche est utile lorsque le cache est actif (*"stateSave": true*) et que vous devez connaître la valeur de recherche précédente après avoir rechargé la page.

Lire [Comment obtenir la valeur de recherche entrée dans Datatables par programme? en ligne:](https://riptutorial.com/fr/datatables/topic/6085/comment-obtenir-la-valeur-de-recherche-entree-dans-datatables-par-programme-)
<https://riptutorial.com/fr/datatables/topic/6085/comment-obtenir-la-valeur-de-recherche-entree-dans-datatables-par-programme->

Chapitre 4: Datatables - Option Afficher les lignes sélectionnées

Exemples

Afficher uniquement les lignes sélectionnées

Il est fréquent que les données aient une case à cocher pour sélectionner plusieurs lignes. Si les données sont réparties sur plusieurs pages, il peut être difficile pour l'utilisateur d'afficher les enregistrements qu'il a sélectionnés. Pour permettre à l'utilisateur d'afficher tous les enregistrements sélectionnés en une seule fois, nous utilisons généralement un lien hypertexte qui, lorsqu'il est cliqué, affiche uniquement les lignes sélectionnées dans le fichier pouvant être traité. Ce lien peut être utilisé pour basculer entre l'affichage des enregistrements sélectionnés et tous les enregistrements.

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

Dans la méthode ci-dessus, lorsque vous cliquez sur l'hyperlien Sélectionner tout, si le texte div du lien hypertexte est "Afficher les rapports sélectionnés", nous filtrons la valeur datatable pour afficher uniquement les lignes pour lesquelles la case à cocher est cochée. Nous utilisons la fonction API Datatables intégrée **fnFilter**.

Nous passons 2 paramètres à cette méthode - la chaîne de requête et l'index de la colonne à filtrer. Dans ce cas, la valeur de la case à cocher sera «**Checked**» si elle est sélectionnée dans l'interface utilisateur et l'index de la colonne contenant les cases à cocher est 8. Nous passons donc «Checked» et 8 en tant que paramètres à la fonction fnFilter. Après le filtre, nous basculons le lien pour afficher "Afficher tous les rapports".

Lorsque l'utilisateur clique sur Afficher tous les rapports, nous transmettons une chaîne vide à la fonction fnFilter en tant que chaîne de requête. Donc, il affiche tous les enregistrements.

Lire Datatables - Option Afficher les lignes sélectionnées en ligne:

<https://riptutorial.com/fr/datatables/topic/6805/datatables---option-afficher-les-lignes-selectionnees>

Chapitre 5: Migration de <1.10 à 1.10 et plus

Introduction

Datatables 1.10.x est la dernière version à ce jour. Bien qu'il soit rétrocompatible avec les versions précédentes (1.9 etc.), il est fortement recommandé d'utiliser la dernière version qui renvoie directement un objet `datatable api`. Un autre changement majeur, le plus visible, est le passage de [la notation hongroise](#) à la [méthode camelCase](#).

Syntaxe

- Le seul changement de syntaxe majeur est l'utilisation de camelCase partout au lieu de la notation hongroise.
- Un [guide plus détaillé peut être trouvé ici](#) qui implique la conversion des paramètres dans <1.10 à 1.10+

Remarques

Pour plus de détails, essayez de visiter les pages suivantes:

1. [Mise à niveau vers les bases de données 1.10](#)
2. [Historique des bases de données 1.10](#)
3. [Convertir les noms de paramètres de 1.9 à 1.10](#)
4. [Utiliser l'API Datatable](#)

Exemples

Initialisation de Datatable 1.10+

Auparavant, les données étaient initialisées comme suit:

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

Cela permet de retourner un objet jQuery qui serait stocké dans la variable oTable. Et puis, pour accéder à l'API afin de modifier les propriétés de la table, nous avons dû initialiser l'API différemment, comme indiqué ci-dessous:

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

L'initialisation est maintenant modifiée comme suit:

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

Notez que cela retourne une `datatable` `api` et que vous pouvez directement utiliser la `table` variables pour manipuler les propriétés pouvant être datées.

Caractéristiques Non disponible dans Datatables 1.10+

Les 3 fonctionnalités suivantes (qui étaient obsolètes dans la version 1.9) ne sont plus disponibles dans la version 1.10:

1. **fnRender** : Selon le developer:

L'ancienne option `fnRender` offrait une méthode de manipulation d'une cellule lors de sa création. Cependant, il était fourni avec une liste d'options déroutantes comme arguments et nécessitait une structure particulière dans `DataTables` en interne qui posait des problèmes de performances. La suppression de `fnRender` a conduit à une amélioration significative des performances des `DataTables` avec des ensembles de données volumineux et à la possibilité de fournir des instances d'objet à `DataTables` en tant qu'objets de source de données (par exemple, objets observables `Knockout`).

Des alternatives à `fnRender` sont disponibles sous le nom `columns.render` et `columns.createdCell`

2. **bScrollInfinite** : Selon le développeur:

La capacité intégrée de `DataTables` 1.9 à afficher une grille défilant à l'infini via l'option `bScrollInfinite` a été supprimée en raison des incohérences provoquées dans l'API. La suppression a également contribué de manière significative au code interne.

Une extension `Scroller` est disponible en alternative.

3. **Enregistrement basé sur les cookies** :

La sauvegarde d'état basée sur les cookies a été remplacée par une sauvegarde d'état basée sur `localStorage` dans `DataTables` 1.10. Les cookies, avec leur limite de 4 Ko, étaient très limités et entraînaient une perte de performance car ils faisaient partie de chaque requête HTTP. `localStorage` est beaucoup plus rapide et plus flexible et est utilisé comme stockage par défaut pour les informations d'état dans `DataTables` 1.10.

4. **two_button pagination à deux** :

`DataTables` 1.10 a mis à jour de manière significative les contrôles de pagination de `DataTables` (voir `pagingType`), avec pour conséquence que l'ancienne forme de pagination intégrée de `two_button` a été supprimée.

Ils ont pris en charge les personnes souhaitant toujours utiliser la méthode de pagination `two_button` en fournissant un fichier javascript supplémentaire appelé `two_button.js`. L'utilisation est la suivante:

Il suffit d'inclure ce fichier dans votre document après avoir chargé `DataTables`, mais avant d'initialiser votre table et la pagination `two_button` sera restaurée exactement comme dans la version 1.9 (y compris les noms de classes, etc.).

Lire Migration de <1.10 à 1.10 et plus en ligne:

<https://riptutorial.com/fr/datatables/topic/8369/migration-de--lt-1-10-a-1-10-et-plus>

Chapitre 6: Traitement des données côté serveur

Exemples

Charger des données en utilisant ajax avec le traitement côté serveur.

```
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 réponse de l'appel à MY AJAX_ACTION_URL doit être strictement au format ci-dessous:

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

Notez que si la sortie de l'appel ne correspond pas au format ci-dessus, cela entraînera une erreur dans l'initialisation de la `table`.

DataTables 1.10+ Traitement des serveurs

Exemple de tableau

Il existe plusieurs façons d'injecter vos données dans DataTables. Serverside Processing n'est qu'une méthode. De cette manière, DataTables dispose d'un point de terminaison préconfiguré pour récupérer les données, et ce point d'extrémité est responsable de l'acceptation de toutes les requêtes de pagination / filtrage / tri que DataTables applique. Il existe une variété d'avantages et d'inconvénients par rapport à l'envoi de votre ensemble de données complet depuis le serveur et à ce que DataTables le fasse tout du côté du client en fonction de votre cas d'utilisation.

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

Exemple d'accrochage d'un événement preXhr pour envoyer des données supplémentaires à une requête ajax

Cet événement se déclenche directement avant que l'appel ajax ne soit effectué, vous permettant de modifier le corps de la requête. Ceci est utile si un formulaire influence les données renvoyées à la table (imaginez peut-être une barre min / max pour filtrer dans une plage de dates ou similaire).

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

Explication des exigences côté serveur pour le traitement des demandes

Dans une instance sans option de DataTables, tout le filtrage, le tri et la pagination sont gérés dans le navigateur client. Avec Serverside Processing activé, ces tâches sont transférées sur le serveur Web. Pour les très grands ensembles de données qui peuvent être inefficaces à envoyer dans leur intégralité au client, cela peut aider.

Plusieurs paramètres par défaut sont envoyés par la requête Datatables lorsque vous configurez un noeud final ajax. Cela peut prendre beaucoup de temps et, plutôt que de les détailler, une vue d'ensemble des responsabilités du serveur peut être plus utile.

En commençant par tous les paramètres facultatifs que vous avez pu fournir à la demande, compilez votre ensemble de données et préparez-le pour les opérations DataTables. Utilisez les paramètres `search[value]` et `search[regex]` pour appliquer un filtrage sur toutes les colonnes /

propriétés. Il existe également des paramètres de `columns[i][search][value]` et des `columns[i][search][regex]` pour les filtres de colonne individuels, bien qu'ils ne soient pas couramment utilisés dans les instances simples de DataTables.

Triez vos données filtrées en utilisant les différents paramètres de `order`. Comme pour les paramètres de recherche, il y aura un ensemble de paramètres de classement par colonne sur lesquels le tri est activé. `order[i][column]` et `order[i][column][dir]`

Une fois le filtrage et le tri terminés, il est temps de page les données en fonction de la requête DataTables en utilisant les paramètres de `start` et de `length`. Dans .NET, cela pourrait ressembler à:

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

Exemple de réponse

C'est la structure de réponse approximative attendue par DataTables. Quelles que soient vos données (tableau 2D, tableau d'objets, etc.), elles sont imbriquées dans la propriété `data` avec les autres propriétés du noyau DataTable pour en tirer des informations sur la pagination et le filtrage (par exemple:)

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

Voir aussi `.rows.add(data)` et l' [option data](#) pour d'autres méthodes de définition du contenu de votre DataTable à l'aide de données JSON. Bien sûr, DataTables peut également être [initialisé à partir d'attributs de data- HTML ou HTML5 statiques](#)

Obtenir des données JSON à partir de la table MySQL

Sur le site officiel de DataTable, on [peut voir](#) comment un processus côté serveur avec PHP et MySQL peut être considéré. Cet exemple est **obsolète** et ne peut plus être utilisé avec PHP 7 (la fonction `"mysql_pconnect"` et les fonctions associées sont obsolètes, voir ce [post](#)).

Donc, cette fonction vous donne des données JSON bien formées en réponse:

```
<?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 => 'salary',
);

$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 salary 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 réponse ressemble et peut être traitée par le 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"
    }
  ]
}
```

Lire Traitement des données côté serveur en ligne:

<https://riptutorial.com/fr/datatables/topic/4176/traitement-des-donnees-cote-serveur>

Chapitre 7: zone de saisie de recherche de données pour une recherche en temps réel

Exemples

Zone de saisie de recherche pour une recherche progressive sur la base de données

Vous trouverez ci-dessous un exemple d'implémentation d'une zone de saisie Recherche qui aide les utilisateurs à rechercher les occurrences d'une valeur particulière dans la base de données.

Dans l'exemple ci-dessous, #report est l'identifiant div du div qui contient la zone de saisie de recherche. Cette fonction est appelée dès que l'utilisateur entre une valeur dans cette zone de saisie. Comme il peut y avoir de nombreuses occurrences d'un seul caractère, nous appelons la fonction de recherche réelle uniquement lorsque plus d'un caractère est entré dans la zone de recherche.

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

Dans la fonction searchTable, nous utilisons la fonction de **datatables** intégrée fnFilter pour rechercher les **occurrences** correspondantes de la chaîne d'entrée. **Nous pouvons limiter la recherche à une colonne particulière en passant l'index de colonne.** Ici, nous passons l'index de la colonne 2.

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

Si vous avez besoin de rechercher dans toutes les colonnes, **assurez-vous simplement de ne pas passer le paramètre d'index r.**

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

Lire zone de saisie de recherche de données pour une recherche en temps réel en ligne:

<https://riptutorial.com/fr/datatables/topic/6806/zone-de-saisie-de-recherche-de-donnees-pour-une>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec les données	andialles , Andrei Zhytkevich , Chris H. , Community , cookypuss
2	Ajouter des boutons d'exportation à la table dans Bootstrap 4	Martin
3	Comment obtenir la valeur de recherche entrée dans Datatables par programme?	Domenico Campagnolo
4	Datatables - Option Afficher les lignes sélectionnées	saikris
5	Migration de <1.10 à 1.10 et plus	philantrovert
6	Traitement des données côté serveur	Aswathy S , BLSully , Martin , philantrovert
7	zone de saisie de recherche de données pour une recherche en temps réel	saikris