

Guide du développeur

Utilisation de DataTable

	Date de naissance	est Grand	Peut se connecter	Group
	1991-06-13	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Admir
	1991-06-24	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Redac
	1991-06-24	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Admir
	1987-11-28	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Redac
avec accent &é"(\$§è!!	2014-01-17	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Utilise
	2014-02-20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Utilise
	2014-02-20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admir
	2014-02-21	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admir
	2014-02-22	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admir
t avec accent &é"(\$§è	2014-02-23	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admir

First
Previous
1
2
Next

Edouard COMTET

Génie Informatique et Statistiques - Promo 2014

Février 2014 sous la direction de Olivier CARON

Guide du développeur

Utilisation de DataTable

Introduction	3
Installation	4
Création d'un objet Table	5
Création de la page HTML	5
Mise en place du PHP	5
Personnalisation des options	6
Ajout des colonnes et des filtres	7
Ajout d'un filtre	7
Ajout d'une colonne	7
Affinement des autres options	9
Fonctions CRUD	9
Pagination & tri	9
Icône d'extension	10
Langue	10
Rendu de l'interface	12
Modifier	12
Ajouter	12
Supprimer	13
Introspection sur le type des champs	13
Affichage des erreurs	13
Outils de détection des erreurs	14
Bibliographie & Annexes	15

1. Introduction

Lors de la création d'une partie d'administration, la plupart des développeurs doivent régulièrement mettre en place des tableaux pour la gestion (tableaux des utilisateurs, des news, ...).

Le but de ce framework est de permettre, en quelques lignes de code, de générer un tableau HTML/CSS/Javascript relié à une base de données. La modification du tableau sera, de manière automatique, répercutée dans la base.

Ce document est destiné aux développeurs qui devront l'utiliser. Il explique quelle fonction utiliser et comment le paramétrer.

Dans la suite de ce document, on considèrera que le framework est installé dans un dossier du nom *DataTable* et que le fichier utilisant ce dernier est à la racine et se nomme *index.php*.

Ce document est rattaché à son homologue qui permet de comprendre l'architecture du framework et comment le modifier.

Ce framework se base sur DataTable, disponible à l'adresse <http://datatables.net/>. Le code source de cette amélioration est disponible depuis <https://github.com/doudou34/DataTable>.

2. Installation

L'installation du framework se décompose en 4 étapes élémentaires:

- Le framework fonctionne avec des versions de PHP supérieur à la 5.3, l'utilisateur doit avoir autorisé javascript et activé les cookies dans son navigateur.
- Copier l'intégralité des fichiers dans le répertoire d'installation.
- Configurer le dossier *DataTable/temp/* pour que le serveur web¹ puisse y accéder en lecture et en écriture. En cas d'erreur d'accès, lors de la modification, de la suppression ou de l'ajout, un message apparaîtra: "File not found. Maybe a right's problem".
- Initialiser les variables de connexion disponibles au début du fichier *DataTable/php/Table.php*.

```
// Default parameters
protected static $BDD_URL = "127.0.0.1";
protected static $BDD_LOGIN = "root";
protected static $BDD_PASSWORD = "";
protected static $BDD_DATABASE = "pfe";
protected static $BDD_DRIVER = "mysql";
protected static $LANGUAGE = "en";
protected static $ENCODING = "utf8";
```

Pour la gestion des langues, se référer à la partie "Langue" du présent document; pour ajouter un driver, se référer à son homologue.

Ces paramètres seront utilisés par défaut lors de la connexion à la base de données et lors du rendu. Il est important de ne pas mettre de constante dans ces variables. En effet, pour pouvoir assurer la transition entre javascript et php, un mécanisme de sérialisation est nécessaire. L'utilisation de constante, avec l'instruction "define" par exemple, n'assure pas la désérialisation.

Ce type d'erreurs peut-être facilement détecté, référez vous à la partie "Outils de détection des erreurs".

¹ Service qui exécutera les scripts PHP (traditionnellement l'utilisateur www-data).

3. Création d'un objet Table

A. Création de la page HTML

Pour créer une nouvelle page avec un Table dedans, il faut d'abord créer une page HTML. Voici un exemple:

```
<html>
  <head>
    <title>Test DataTable</title>

    <link href="./DataTable/css/bootstrap.css" rel="stylesheet">
    <link href="./DataTable/css/bootstrap-theme.css" rel="stylesheet">
    <link href="./DataTable/css/dataTables.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <div class="row">
        <h1>Test du PFE avec affichage en bootstrap</h1>
      </div>
      <div class="row">
        <?php
          // CODE PHP ICI
        ?>
      </div>
    </div>

    <script type="text/javascript" src="./DataTable/js/jquery.js"></script>
    <script type="text/javascript" src="./DataTable/js/bootstrap.js"></script>
    <script type="text/javascript" src="./DataTable/js/dataTables.js"></script>
    <script type="text/javascript" src="./DataTable/js/table.js"></script>
  </body>
</html>
```

Il est nécessaire d'inclure la librairie Bootstrap (autant au niveau Javascript que CSS), la librairie DataTable de base ainsi que le fichier CSS du framework et son équivalent Javascript.

B. Mise en place du PHP

La structure de base étant en place, la création d'un objet Table sera infantine. Dans la partie PHP, vous pouvez y inscrire ceci:

```
require_once('./DataTable/php/include.php');

$table = new DataTable\php\Table();
$table->init("user");

$table->show("./DataTable/");
```

Cette partie de code permet de créer un objet Table, de le connecter à la table *user* et de l'afficher. La fonction *show* prend un seul paramètre: le chemin où est installé le framework par rapport au fichier d'utilisation. Il doit se terminer par un */*.

C. Personnalisation des options

Il est possible de modifier les paramètres de connexion et de langue grâce à la fonction *init*.

S'il n'y a pas de paramètres passés ou qu'ils sont vides, les valeurs par défaut vues plus haut seront utilisées.

```
public function init(  
    $name,  
    $url = "",  
    $login = "",  
    $password = "",  
    $database = "",  
    $driver = "",  
    $language = "",  
    $encoding = ""  
);
```

Nous étudierons plus tard comment modifier le style de pagination, la gestion du tri, de la modification, de la suppression, de l'ajout et des langues.

Dans le deuxième document, il est également expliqué comment ajouter un driver et la personnalisation du CSS.

La partie suivante permet d'ajouter des colonnes et des filtres pour afficher une table selon vos besoins.

4. Ajout des colonnes et des filtres

Pour ajouter des colonnes et des filtres, il faut utiliser des fonctions pour chacun.

A. Ajout d'un filtre

Un filtre permet de n'afficher qu'une certaine partie des données d'une table, par exemple si on ne souhaite afficher que les utilisateurs connectés on écrira:

```
$table->addFilter("connected", "= TRUE");
```

Cette fonction prend en paramètre la colonne sur laquelle on fera le tri et la condition à prendre en compte.

B. Ajout d'une colonne

Il est possible d'ajouter différents types de colonnes à un objet *Table* qui seront, suivant les paramètres, affichées ou non, modifiables ou non, ... Voici la liste intégrale des colonnes disponibles et leurs caractéristiques:

- *Col*: Une colonne classique. La fonction prend pour paramètre un libellé affiché à l'utilisateur et le nom de la colonne dans la table.

```
public function addCol($label, $name)
```

- *ColIndex*: Cette colonne sera utilisée comme index lors de la modification, il est impossible de la modifier. Il faudra supprimer et recréer la ligne. La fonction prend en paramètre un libellé pour la colonne, le nom de la colonne dans la table, un premier booléen pour gérer l'affichage et un deuxième pour préciser s'il faut proposer la colonne lors de l'ajout.

```
public function addColIndex($label, $name, $visible, $creatable)
```

- *ColHidden*: Cette colonne n'est jamais affichée à l'utilisateur mais lors de l'ajout, la valeur par défaut de la colonne sera ajoutée. La fonction prend en paramètre un libellé, non utilisé, le nom de la colonne dans la table et la valeur par défaut (non vide).

```
public function addColHidden($label, $name, $default)
```

- *ColLinked*: Cette colonne est liée à une autre table (notion de clef étrangère). La liaison sera faite automatiquement avec la seconde table et affichera la valeur

de la seconde table. La fonction prend en paramètre un libellé affiché, le nom de la colonne dans la table, le nom de la table pour la liaison, l'index de la seconde table (une seule colonne) et la(es) colonne(s) qui sera(ont) affichée(s) à l'utilisateur. Sur ce type de colonne il est également possible d'ajouter un filtre avec la fonction *getCol* et *addFilter*.. Il est possible d'afficher plusieurs champs, pour cela on entourera les champs par des *. Il est important de ne pas afficher une colonne portant le même nom qu'une colonne déjà affichée.

```
public function addColLinked($label, $name, $otherTable, $otherIndex, $otherName)
$table->addColIndexLinked("Utilisateur", "user_id", "user", "id", "*username* - *expired*");
$table->getCol("user_id")->addFilter("id", "<> '1'");
```

- *ColIndexLinked*: Une colonne qui servira d'index et qui est liée à une autre table. Elle prend les mêmes paramètres que la fonction *ColLinked*. Il n'est pas nécessaire de gérer l'affichage et l'ajout car l'utilisation de *ColIndex* remplit ces cas d'utilisations.

```
public function addColIndexLinked($label, $name, $otherTable, $otherIndex, $otherName)
```

Il est également possible d'ajouter l'intégralité des colonnes avec détection des clef primaire (affichée mais non visible lors de l'ajout) grâce à la fonction *addAllCols*.

5. Affinement des autres options

A. Fonctions CRUD

Lors de la création d'une fonction, il existe principalement 4 sous fonctions: la création (Create), la lecture (Read), la modification (Update) et la suppression (Delete). Ce framework permet de distinguer ces 4 possibilités.

```
// Operation available
public static $CREATE = 0x01;
public static $READ = 0x02;
public static $UPDATE = 0x04;
public static $DELETE = 0x08;

public function canCreate($create);
public function canRead($read);
public function canUpdate($update);
public function canDelete($delete);

public function can($flags)
```

La dernière fonction utilise les 4 premiers drapeaux. On pourra par exemple donner le droit de création et de lecture avec (ce qui enlèvera le droit de modification et de suppression):

```
$table->can(DataTable\php\Table::$CREATE | DataTable\php\Table::$READ);
```

B. Pagination & tri

De la même manière, on peut activer la pagination ou la désactiver avec les drapeaux et la fonction suivante:

```
// Style of pagination
public static $PAGINATION_FULL_NUMBERS = "full_numbers";
public static $PAGINATION_NO = "";

public function setPaginateStyle($paginate)
```

Et la fonction pour activer ou désactiver le tri qui prend en paramètre un type de tri:

```
// Style of sort
public static $SORT_NATURAL = "natural";
public static $SORT_BASIC = "";

public function setSort($sort);
```

C. Icône d'extension

Il est possible d'ajouter une troisième icône pour permettre de réaliser une action personnalisée. Pour ceci, utilisez la fonction *setExtension*. Une nouvelle icône apparaît. Pour déclencher cette fonction, tout comme lors de la suppression, sélectionnez les lignes et cliquez sur l'icône ou glissez-déposez les lignes. Un événement *fnExtension* est alors déclenché avec pour paramètre le *<tr />* concerné.



D. Langue

Il est également possible de personnaliser le framework en modifiant la langue avec celles disponibles et la fonction *init* vue précédemment.

```
{
  "sProcessing":      "Traitement...",
  "sSearch":          "Rechercher",
  "sLengthMenu":      "Afficher _MENU_ éléments",
  "sInfoThousands":   " ",
  "sInfo":             "Affichage des éléments _START_ à _END_ sur _TOTAL_ éléments",
  "sInfoEmpty":        "Aucun éléments",
  "sInfoFiltered":     "(filtré de _MAX_ éléments au total)",
  "sInfoPostFix":      "",
  "sLoadingRecords":  "Chargement...",
  "sZeroRecords":      "Aucun élément à afficher",
  "sEmptyTable":       "Aucune donnée disponible dans le tableau",
  "sDragDrop":         "Glissez-Déposez une ligne pour la supprimer",
  "sDragDropOption":   "Glissez-Déposez une ligne pour déclencher l'action",
  "oPaginate": {
    "sFirst":          "Premier",
    "sPrevious":        "Précédent",
    "sNext":            "Suivant",
    "sLast":            "Dernier"
  },
  "oAria": {
    "sSortAscending":  ": Activer pour trier la colonne par ordre croissant",
    "sSortDescending": ": Activer pour trier la colonne par ordre décroissant"
  },
  "eFunctionUnkown" : "Fonction inconnue",
  "eMissingPatameters": "Paramètres non trouvés",
  "eUnserialize": "Impossible de désérialiser",
  "eUnkonwn": "Erreur inconnu",
  "eAction": "Impossible d'effectuer cette action",
  "eQuery": "Erreur dans la requête",
  "eFindCol": "Impossible de trouver la colonne",
  "eFile": "Impossible de trouver le fichier",
  "eFail": "Erreur",
  "eNone": "Succès"
}
```

```
// Language available  
public static $FRANCAIS = "fr";  
public static $ENGLISH = "en";
```

Enfin, cet outil vous permet de rajouter une langue en créant un nouveau fichier dans *DataTable/translations/*. Le nom du fichier, *nomDeLaLangue.json*, se structure comme vu juste au dessus.

6. Rendu de l'interface

Test de DataTable

Show 10 entries

+

Search:

N	Login	Mail	Date de naissance	est Grand	Peut se connecter	Groupe
1	eco	ecomtet@test.com	1991-06-13	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Admin
2	farcham	farcham@test.com	1991-06-24	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Redacteur
3	mdarthos	mdarthos@test.com	1991-06-24	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Admin
4	svangaev	svangaev@test.com	1987-11-28	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Redacteur
16	tata	tata@test.com Ceci est un test avec accent &é"(\$\$è!!	2014-01-17	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Utilisateur
17	tonton	titi@test.com	2014-02-20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Utilisateur
18	titi	titi@test.com	2014-02-20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin
19	tete	tete@test.com	2014-02-21	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin
20	tyty	tyty@test.com	2014-02-22	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin
21	zaza	zaza@test.com Ceci est un test avec accent &é"(\$\$è	2014-02-23	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin

Showing 1 to 10 of 13 entries

First

Previous

1

2

Next

Last

A. Modifier

Pour modifier une case, il suffit de double-cliquer dessus. Un clique autre part sauvegarde la modification, une pression sur la touche *échap* l'annule. S'il y a une erreur, elle sera affichée sur la case en cours de modification.

B. Ajouter

Une icône avec un “+” permet d'ajouter une nouvelle ligne. Une nouvelle fenêtre s'ouvre et vous permet de remplir les cases. En cliquant sur *enregistrer*, la nouvelle ligne s'ajoute ou une erreur est affichée sur l'icône “+”.

Ajouter un enregistrement

Login

Mail

Date de naissance

dd/mm/yyyy

☒ est Grand

☒ Peut se connecter

Groupe

Admin

Annuler

Enregistrer

C. Supprimer

Il existe deux manières de supprimer:

- En glissant déposant une ligne sur l'icône “poubelle”.
- En sélectionnant des lignes et en cliquant sur la poubelle. Il est possible de sélectionner plusieurs lignes.

Id	Nom	Email	Date	Statut	Statut	Utilisateur
18	titi	titi@test.com	2014-02-20			Admin
19	tete	tete@test.com	2014-02-21			Admin
20	tyty	tyty@test.com	2014-02-22			Admin
21	zaza	zaza@test.com Ceci est un test avec accent &é"(\$\$è	2014-02-23			Admin

Les erreurs seront affichées sur l'icône de la poubelle.

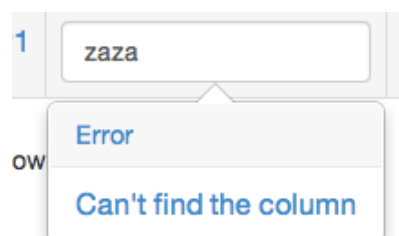
D. Introspection sur le type des champs

Ce framework détecte également automatiquement le type des champs de la base. Pour cela, une requête est exécutée et récupère le type de chaque colonne. L'affichage se transforme suivant le type (*linked*, *booléen*, *varchar*, *text*, *date*, ...).

The image shows a form with several input fields. At the top left is a dropdown menu with 'Admin' selected. Below it is a text input field containing 'titi'. To the right of these is a large text area containing 'titi@test.com'. Below the 'titi' field is a date input field showing '20/02/2014'. There are also some icons like a checkmark and a trash can.

E. Affichage des erreurs

Les erreurs s'affichent sous forme de bulle comme ceci:



7. Outils de détection des erreurs

Ce projet, encore en phase de test, peut présenter des erreurs. Il est possible de les signaler ici: <https://github.com/doudou34/DataTable/issues>. Une correction sera apportée le plus rapidement possible.

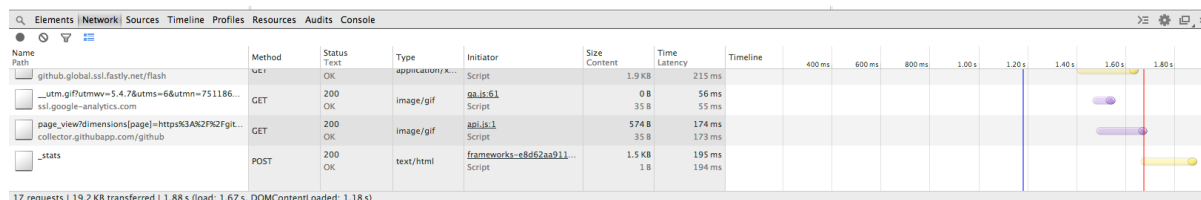
Il peut aussi s'agir d'une erreur de votre part suite à un manque de documentation. Pour vous aider, voici deux méthodes.

Les erreurs classiques, en PHP. Pour que PHP affiche les erreurs il est nécessaire de mettre la variable `display_errors` à "On". Deux manières, directement en PHP, ici <http://php.net/manual/fr/errorfunc.configuration.php>, ou dans le fichier `php.ini`. Pour trouver ce fichier, la fonction `phpinfo()` vous aidera.

Les erreurs en javascript peuvent être détectées sous Chrome avec la console. Pour cela, faites un clique droit, inspecter l'élément. Vous devriez voir apparaître une fenêtre en bas:



L'onglet console vous permet de détecter les erreurs javascript. Un deuxième onglet, Réseau, vous permettra de détecter les erreurs AJAX en récupérant les fichiers requis par la page, notamment le fichier `ajax.php` qui sert de passerelle entre le PHP et le javascript dans le framework.



8. Bibliographie & Annexes

PHP: <http://www.php.net/manual/fr/>

PHP est un langage informatique utilisé dans ce framework. Certains aspects complexes, comme la notion d'objet, sont utilisés dans celui-ci.

Bootstrap: <http://getbootstrap.com/>

Bootstrap est un framework qui permet de créer facilement et simplement des pages HTML en définissant des classes CSS. Grâce à sa partie jQuery, il permet également d'afficher des fenêtres, des tooltips, ...

DataTable: <http://datatables.net/>

DataTable est un framework qui permet, grâce à jQuery, de formater un tableau HTML avec une pagination, un champ de recherche, ...

PDO: <http://php.net/manual/fr/book.pdo.php>

PDO est une librairie de PHP qui permet de gérer des connexions avec une base de données.

jQuery: <http://jquery.com/>

jQuery est une librairie javascript qui permet d'étendre les fonctions du langage.

GitHub: <https://github.com/>

GitHub est un service internet qui permet de proposer à la communauté un espace de stockage pour un projet informatique. Il permet aussi d'améliorer le travail en équipe au même titre que SubVersion.