|  |
| --- |
| C:\Users\Marine\Dropbox\Pôle Communication\Logos\Logos 2014\TagCommander_logo-M.png  Tc\_vars Collector  V1.0  05/06/2015 |
|  |

# Summary

[Summary 2](#_Toc425146923)

[1. Introduction 3](#_Toc425146924)

[2. Structure et fonctionnement général du code 3](#_Toc425146925)

[3. Liste des fonctions et variables JavaScript 5](#_Toc425146926)

[4. Liste des fonctions et variables PHP 6](#_Toc425146927)

# Introduction

L’outil est actuellement à un point qui ne lui permet d’être fonctionnel qu’en local.

Le volume de données envoyé pouvant être conséquent, il peut excéder le volume gérable par un navigateur.

De plus, la fonction d’envoi étant en Ajax, il ne tolère pas le cross-domain.

La fonction d’envoi et de traitement des données devra être repensée.

Le document servira d’abord de description du contenu de mon script en trois étapes :

1. L’explication de mon script, son fonctionnement actuel et comment il devrait fonctionner et une proposition de fonctionnement alternatif
2. La définition des fonctions et variables JavaScript
3. La définition des fonctions et variables PHP

**En Annexe :**

* Le Descriptif en format doc
* Le dossier QA vars avec les fichiers php et excel :
  + Class = > contenant la classe collectedvars
  + PhpExcel => Contenant les classes PhpExcel pour la génération de fichiers Excel
  + Temp => contenant le fichier Excel datalayer.xls
* Le fichier recupvars.js => le Javascript de collecte et d’envois des variables
* Un exemple d’export de datalayer généré par le script

# Structure et fonctionnement général du code

Le fonctionnement recherché est le suivant :

* Le démarrage de la navigation doit se faire à volonté de l’utilisateur
* Lors de la navigation
  + Elle doit se poursuivre sur toutes les pages quand elle est activée
  + Elle doit pouvoir se mettre sur pause et recommencer
  + Elle ne doit pas prendre de pages en doublon
  + A la fin de la navigation, le fichier Excel généré doit pouvoir se télécharger automatiquement

Point à noter :

* Le développement inclus une partie front en JavaScript et une partie back en PHP.
* Le code PHP utilise la librairie PHPExcel, dont le dossier est inclue.
* Le traitement se fait dans le fichier ***index.php*** incluant les classes PHP Excel et la Classe **collectedvars**
* De dossier contient le fichier **index.php**, le dossier **PHP Excel** et le dossier ***tmp*** qui contient le classeur Excel de référence **datalayer.xlsx** et les classeurs générés.
* Le fichier compose un fichier Excel à partir d’un tableau Json de ce format :

{

navigation:{

0:"//URL.com/1",

1:"//URL.com/2",

2:"//URL.com/3",

...

},

QA\_category\_name:{

0:"Name\_1",

1:"Name\_2",

2:"Name\_3",

...

},

var1:{

0:"Value\_1",

1:"Value\_2",

2:"Value\_3",

...

},

var2:{

0:"Value\_1",

1:"Value\_2",

2:"Value\_3",

...

},

...

}

Ce format sert de base pour la construction du tableau Excel, ou chaque clef correspond à une ligne et chaque valeur à une colonne. Les valeurs en questions peuvent être aussi bien des chaînes de caractères, des chiffres ou des tableaux.

Fonctionnement actuel de l’application :

* L’application ne se doit se lancer que si le cookie 'StartQA' est reconnu:

1. La fonction javascript **getAllVars()** collecte les variables dans un tableau en format Json.

Toutes les tc\_vars collectent alors leur valeur en fonction de la page où elles sont chargées => toutes les tc\_vars sont des tableaux

* + 1. Ce tableau est stocké en local Storage sous forme d’une chaîne de caractères
    2. Ce Tableau est mis à jour par la même fonction

1. La fonction javascript **sendVars()\*** envoie toutes les données en une fois en format post vers le serveur à destination de **index.php**
2. Les données sont récupérées et stockées dans la variable **$datalayer**
3. Les données sont placées dans le fichier Excel, chaque variable étant une ligne, et chacun des éléments du tableau une colonne
4. Les valeurs sous forme de tableau à deux dimension tels que **order\_products** ou **list\_products** etc.. sont traités, le premier élément est décortiqué et les suivants sont affichés sous la forme d’un Json stringifié
5. Après traitement, **index.php** affiche le nom du fichier Excel généré
6. La fonction javascript **sendVars()\*** récupère le nom du fichier et le télécharge

\* La fonction javascript *sendVars()* exécute une action en Ajax en deux temps qui consiste en l’envoie des variable et la récupération du fichier généré.

Ce mode de fonctionnement posant les problèmes cités plus haut (problème de cross-domain, trop de données chargées en une fois via post..) voici une alternative de fonctionnement :

1. La fonction javascript **getAllVars()** collecte les variables et les envois directement à **index.php**
2. **Index.php** traite chaque requête individuellement et stocke sous un format texte en se basant sur le tCID (utilisation possible du navigator.sendBeacon("//preprod.tagcommander.com/~yann/QA/index.php", a); )
3. Quand la navigation est terminée, il faut alors recréer le format Json utilisé pour la création du fichier Excel (cf format précédent)
4. La fonction javascript **sendVars()\*** crée un iframe avec en SRC une requête avec en paramètres le TCID à destination de **index.php**
5. Le Json est reconstitué en PHP
6. Les données sont récupérées et stockées dans la variable **$datalayer**
7. Les données sont placées dans le fichier Excel, chaque variable étant une ligne, et chacun des éléments du tableau une colonne
8. Les valeurs sous forme de tableau à deux dimension tels que **order\_products** ou **list\_products** etc.. sont traités, le premier élément est décortiqué et les suivants sont affichés sous la forme d’un Json stringifié
9. Après traitement, **index.php** crée une redirection PHP vers le fichier Excel créé, téléchargé via l’iframe

# Liste des fonctions et variables JavaScript

* tCQAvars

C’est la variable qui stocke toutes les fonctions = > toutes les fonctions sont name-spacées tCQAvars

* tCQAvars.navigation

Est la variable qui stocke toutes les url des pages visitées

* tCQAvars.QA\_category\_name

Est la variable qui stocke tous les titres des pages

* tCQAvars.Launched\_tags

Est la variable qui stocke les tags chargés par le dernier conteneur de la page ( tC.array\_launched\_tags)

* tCQAvars.verifIfObj()

Fonction qui vérifie si la valeur d’une variable est une erreur, objet ou un string

Si c’est un objet, elle est traitée avec la fonction *tCQAvars.getObject()*

* tCQAvars.getObject()

Fonction qui traite l’objet de manière à stocker tous les éléments dans un nouvel objet, vérifie tous les éléments avec *tCQAvars.getObject()*

* tCQAvars.getAllVars()

Fonction principale servant à récupérer toutes les variables, les stocker dans tCQAvars.tc\_newvars sous forme de tableau, dans l’ordre de la navigation.

* tCQAvars.sendVars()

Fonction envoyant les informations en post et téléchargeant le fichier excel à la réception de la valeur

* tCQAvars.stopQA()

Fonction supprimant le cookie

* tCQAvars.startQA()

Fonction démarrant l’enregistrement de la navigation à travers la pose du cookie et l’appel de la fonction tCQAvars.getAllVars()

# Liste des fonctions et variables PHP

* $collectedVars

Classe définie pour le traitement des données

* $arrayToString(array)

Fonction transformant en chaîne de caractères un tableau

* $setArrayValues(array,colonne)

Fonction décortiquant le premier élément d’un tableau, chacun entré individuellement dans le tableau Excel, dans la ligne et colonne donnée

* $getVars(variable)

Fonction récupérant toutes les variables d’une variable données

* $setTitles()

Fonction posant tous les titres

* $setCellColor(colonne, ligne, couleur)

Fonction qui pose la bonne couleur dans la cellule donnée

* $error;  
  couleur des erreurs
* $empty;

Couleur des valeurs vides

* $array;

Couleur des tableaux

* $subarray;

Couleur du tableau à deux dimensions

* $title;

Couleur des tires

* $row;

Numéro de ligne Excel

* $columnOrigin;

Numéro de colonne d’origine

* $Alphabet;

Redéfinition des colonnes pour le fichier excel

* $json

Variable qui stocke les données envoyées en post

* $datalayer;

Tableau de variables

* $objPHPExcel

Stocke le fichier Excel

* ***$sheet***

Stocke la feuille Excel active

* ***$nmbVars***

Chiffrage du nombre de variables

* ***$predefined***

Variables définissant les variables standart et connues TagCommander

* ***$records***

Nom du fichier généré et enregistré

* ***$writer***

Ecriture du fichier Excel