Lead’s FACTORY

Documentation

Documentation technique et fonctionnelle.

EXOL

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | http://www.exol.fr  Eric.wallet@exol.fr |
|  |  |  |

Table des matières

[Introduction 1](#_Toc424934680)

[Documentation fonctionnelle 2](#_Toc424934681)

[Documentation technique 3](#_Toc424934682)

[Référence de tags 4](#_Toc424934683)

[Annexes 8](#_Toc424934684)

# Introduction

« Vous avez quelque chose de très important à dire à vos lecteurs ?   
Mettez-le en évidence au moyen d’un encadré. »

# Installation

Un projet Lead’s factory se presente avec deux éléments :

1 – Une archive des fichiers du serveur.

2 – Un Dump de la base SQL

Pré-requis

* Une version récente de Java pour le module segmentation marketing de la lead’s factory : <https://www.elastic.co/guide/en/elasticsearch/guide/current/_installing_elasticsearch.html>
* PHP 5.5 minimum
* MySQL 5.4 minimum

## Les fichiers

Les fichiers sont à copier dans le VHost applicatif. Il faut s’assurer que les droits sont positionnés en www-data et 775.

## Le Dump

Le dump doit etre importé dans votre base MySQL

mysql –h 127.0.0.1 –u root –p <mabase> < <mon\_fichier\_sql>

mabase : Nom de la base de données sur votre serveur. Cette base doit etre en UTF8\_Unicode\_CI.

mon\_fichier\_sql : Fichier SQL à importer

## ElasticSearch

ElasticSearch se trouve dans le dossier install de la lead’s.

* <https://github.com/tellaw/leadsfactory/tree/master/Install>

Pour Installer ElasticSearch :

1. Le dezipper dans le dossier de destination.
2. Lui donner les droits souhaités (vous pouvez lui attribuer www-data) via un chown
3. Dezipper et copier head et gui dans un dossier plugins à la racine (le créer si il n’existe pas).
4. Copier le script de lancement automatique (elasticsearch) en prenant soin d’adapter le chemin avec celui ou vous avez installé ElasticSearch.
   * ES\_HOME=/var/www/elasticsearch-1.7.1
5. Fixez ses droits :
   1. sudo chmod 0755 /etc/init.d/elasticsearch
6. Ajoutez le au lancement automatique
   1. sudo update-rc.d elasticsearch defaults
7. Executez un lancement manuel /etc/init.d/elasticsearch start

## Kibana

1. Le dezipper dans le dossier de destination (l’archive se trouve dans le dossier Install de la leads)..
2. Lui donner les droits souhaités (vous pouvez lui attribuer www-data) via un chown
3. Copier le script de lancement automatique (kibana) en prenant soin d’adapter le chemin avec celui ou vous avez installé ElasticSearch.
   1. DAEMON=/var/www/kibana-4.1.1-linux-x64/bin/$NAME
4. Fixez ses droits :
   1. sudo chmod 0755 /etc/init.d/kibana
5. Ajoutez le au lancement automatique
   1. sudo update-rc.d kibana defaults
6. Executez un lancement manuel /etc/init.d/kibana start

## La Crontab

Configurer la crontab suivante pour l’utilisateur www-data :

> crontab –u www-data -e

\*/5 \* \* \* \* cd /data/apps/leads-factory/current;php app/console leadsfactory:crontasks:run > /data/apps/leads-factory/shared/app/logs/cron-leadsfactory.log

Pensez à adapter les chemins de cette crontab avec le chemin de votre installation.

# Documentation fonctionnelle

Utilisez cette section pour fournir un bref résumé de vos données financières, en mettant en évidence les faits saillants.

C’est également l’endroit idéal pour insérer quelques graphiques afin d’illustrer les principales informations financières. Pour ajouter un graphique, dans l’onglet Insertion, cliquez sur Graphique. Le graphique est automatiquement adapté à l’aspect de votre rapport.

Vous avez besoin d’aide pour choisir un type de graphique ? Aucun souci.

* Pour afficher les valeurs d’une catégorie à une autre, par exemple pour comparer les chiffres d’affaires de différentes unités commerciales, essayez un histogramme ou un graphique à barres.
* Pour afficher les valeurs sur une période de temps, par exemple pour l’évolution du chiffre d’affaires ou des profits, essayez un graphique en courbes.
* Pour comparer deux jeux de valeurs connexes, par exemple pour comparer les salaires des dirigeants par rapport au nombre d’années au sein de l’entreprise, essayez un graphique en nuages.

Et lorsque vous êtes prêt à personnaliser l’aspect de votre graphique, cliquez simplement sur le graphique, puis reportez-vous aux icônes à droite pour toute modification, depuis le style et la mise en forme jusqu’à la gestion des données.

# Documentation technique

Cette section présente une conception technique de la lead’s factory.

# Référence de tags

Cette section liste les principaux tags disponibles pour la création de formulaires dans la lead’s factory.

## Objet Texte

Cet objet est un champ texte classique (input type text).

### Appel TWIG

{{ field({'type': 'textarea', 'attributes': {'id': 'lastName'}}) }}

### Rendu HTML

<input type="text" id="lastName" name="lastName"/>

## Objet zone de texte multiligne

Cet objet est un champ textarea.

### Appel TWIG

{{ field({'type': 'textarea', 'attributes': {'id': 'lastName'}}) }}

### Rendu HTML

<textarea id="lastName" name="lastName"></textarea>

## Objet : Téléphone

C’est un champ texte à une ligne utilisable pour la saisie d’un numéro de téléphone. Attention, n’oubliez l’ajout des contrôles de validation sur ce champ.

### Appel TWIG

{{ field({'type': 'phone', 'attributes': {'id': 'myphone'}}) }}

### Rendu HTML

<input type="text" id="myphone"/>

## Objet : Email

C’est un champ texte à une ligne utilisable pour la saisie d’un email. Attention, n’oubliez l’ajout des contrôles de validation sur ce champ.

### Appel TWIG

{{ field({ 'type': 'email', 'attributes': {'id': 'myEmail'}}) }}

### Rendu HTML :

<input type="text" id="myphone"/>

## Objet : Checkbox

C’est une case à cocher. Par défaut non cochée.

### Appel TWIG :

{{ field({ 'type': 'checkbox',

'attributes': {'id': 'myCheckbox','checked' : 'checked'}}) }}

### Rendu HTML :

<input type="checkbox" id="myCheckbox" checked="checked"/>

### Attributs

|  |  |
| --- | --- |
| Attribut | Description |
| @checked=’checked’ | Case cochée par défaut |

## Objet : Liste

Ceci affiche une liste de référence contenant les valeurs de la liste configurée dans le backoffice sous l’identifiant ‘salutation’.

La liste va pouvoir etre affichée sous une des formes suivantes :

* Liste déroulante type ‘select’
* Liste de case à cocher type ‘checkbox’
* Liste de choix type ‘options’

### Appel TWIG

{{

field(

{

'type': 'reference-list',

'attributes': {

'id': 'salutation',

'data-list': 'salutation',

'validator': 'required' }

}

)

}}

### Rendu HTML

<select id="salutation">

<option... valeurs de la liste de ref />

</select>

### Attributs

|  |  |
| --- | --- |
| Attribut | Description |
| @display | Selectionne le type d’affichage :   * Checkbox * radio |

# L’API

## getLead

Méthode utilisable pour récupérer des données de la lead’s. Cette méthode retourne un objet JSON contenant le détail d’un lead.

**Route d’appel :**

|  |  |
| --- | --- |
| Attribut | Description |
| Route | /lead/{id}/{key} |
| Identifiant de route | Aucun |

**Paramètre :**

|  |  |
| --- | --- |
| Attribut | Description |
| id | Identifiant de la lead à charger |
| apikey | Code de sécurité de l’API. |

**Valeur de retour :**

|  |  |
| --- | --- |
| Type | Description |
| Json | Données encodées en JSON de la lead’s |

## validateEmail

Méthode POST.

Cette méthode permet de déclarer un email comme validé à la lead’s.

**Route d’appel :**

|  |  |
| --- | --- |
| Attribut | Description |
| Route | /email/validate |
| Identifiant de route | Aucun |

**Paramètre :**

|  |  |
| --- | --- |
| Attribut | Description |
| email | Email à déclarer comme valide |

**Valeurs de retour :**

|  |  |
| --- | --- |
| Type | Description |
| Json | Information sur l’action de validation |

## getLeads

Méthode POST.

Méthode utilisable pour récupérer des données de la lead’s. Cette méthode retourne plusieurs objets JSON contenant le détail d’un lead.

**Route d’appel :**

|  |  |
| --- | --- |
| Attribut | Description |
| Route | /leads |
| Identifiant de route | Aucun |

**Paramètre :**

|  |  |
| --- | --- |
| Attribut | Description |
| datemin | Date de début (date de création) : Format Y-m-d |
| datemax | Date de fin (date de création) : Format Y-m-d |
| Email | Possibilité de filtrer les lead’s par email |
| form | Possibilité de filtrer les lead’s par formulaire source |

**Valeur de retour :**

|  |  |
| --- | --- |
| Type | Description |
| Json | Données encodées en JSON des leads |

## postLead

Méthode permettant d’injecter un lead’s dans la lead’s factory en méthode POST.

Attention, l’utilisation de cette méthode n’enregistre pas de pages vues. Vous risquez donc d’avoir des statistiques faux.

**Route d’appel :**

|  |  |
| --- | --- |
| Attribut | Description |
| Route | /lead/post |
| Identifiant de route | Aucun |

**Paramètre :**

|  |  |
| --- | --- |
| Attribut | Description |
| content | Json de la lead’s à enregistrer |

**Valeur de retour :**

|  |  |
| --- | --- |
| Type | Description |
| boolean | 1 si succès & 0 en cas d’erreur. |

**Le JSON doit contenir des valeurs remarquables :**

|  |  |  |
| --- | --- | --- |
| Nom | Description | Obligatoire |
| utmcampaign | Code action associé à la lead’s | non |
| firstName | Prénom | Oui |
| lastName | Nom | oui |
| phone | Téléphone | Non |
| email | Email | Non |
| formCode | Code technique du formulaire lié. | oui |

# Les Paramètres de configuration

Module de recherche

|  |  |
| --- | --- |
| Attribut | Description |
|  |  |

# Annexes