

ENSAE

## Projet informatique

Sauvegarde des données de son *Cloud*



Martin GUBRI

Mastère Spécialisé — Data Science

Année universitaire 2014-2015

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Principe</b>	<b>3</b>
<b>3</b>	<b>Technologies utilisées</b>	<b>4</b>
<b>4</b>	<b>Notes techniques</b>	<b>5</b>
<b>5</b>	<b>Base de données</b>	<b>5</b>
<b>6</b>	<b>Utilisation</b>	<b>6</b>
<b>7</b>	<b>Licences</b>	<b>9</b>

# 1 Introduction

Framasoft<sup>1</sup> est une association à but non-lucratif dont l'objet est la promotion et la diffusion du logiciel libre en particulier et de la culture libre en général. Dans cet objectif elle propose différents services en ligne libres. Ces services de *Cloud* fonctionnent grâce à des logiciels libres, comme *Etherpad*. L'objectif visé de l'association est de faire découvrir des alternatives aux géants propriétaires du Web. Ces logiciels peuvent être installés sur d'autres serveurs. L'association insiste même pour que chacun installe son propre *Cloud* chez soi. Cependant, la complexité du processus fait que tout le monde ne peut pas avoir ses propres instances. Framasoft a donc une position intermédiaire entre l'auto-hébergement et le recours à des services fermés aux conditions générales d'utilisation très souvent obscures.

Les deux logiciels les plus populaires proposés sont *Etherpad*, un *Google doc like*, et *Framadate*, un *Doodle like*. Etherpad est un éditeur collaboratif en ligne et en temps réel dans lequel chacun possède une couleur et peut contribuer sur un document même temps que d'autres. Framadate est un outil de sondage en ligne proposant des sondages pour trouver une date ou des sondages classiques pour tous les autres types de votes.

La popularité grandissante de l'instance d'Etherpad de l'association a impliqué de nombreuses instabilités ces derniers mois<sup>2</sup>. Un bug encore non résolu dans Etherpad fait tomber le service régulièrement. Il se relance automatiquement au bout d'un certain temps, mais parfois une intervention manuelle est nécessaire. De ce fait, nous recevons très souvent sur le support de Framasoft<sup>3</sup> des demandes urgentes de personnes n'ayant pas de sauvegardes locales. De plus, comme toute organisation, il est possible que faute de finances suffisantes, l'association doivent fermer certains de ses services. Dans un objectif de pérennité des données, il m'a donc paru pertinent d'aider à la création de sauvegardes locales.

## 2 Principe

Le but de ce projet est de fournir une interface pour facilement garder des copies locales de ses données issues d'instances d'Etherpad et de Framadate. Ces logiciels fonctionnant en ligne, la création des documents d'Etherpad, appelés *pads*, et des sondages de Framadate sont accessibles via une URL. Le problème est à la fois de pouvoir conserver ses URL et le contenu associé. Garder un fichier à jour à la main listant les URL de ses contenus est fastidieux. De plus, les URL sont souvent composées de chaînes de caractères aléatoires, et ne sont donc pas explicites sur la description du contenu associé.

Le projet répond à ces problématiques. Premièrement, *BackupMyCloud* récupère et stocke automatiquement les URL depuis l'historique et les marques-pages de Firefox, Chrome et Chromium. Ainsi, la liste de ses URL est toujours à jour sans effort. L'historique permet d'avoir les URL sauvegardées sans manipulation particulière, et les marques-pages permettent de s'assurer que les URL importantes soient toujours disponibles même si l'historique est vidé avant de lancer *BackupMyCloud*.

Pour remédier au fait que l'extraction depuis l'historique peut contenir beaucoup de contenu non-pertinent, il est possible de *blacklister* certains contenus pour éviter d'avoir une liste trop longue et gagner en clarté.

Dans un objectif d'auto-hébergement, il était nécessaire de ne pas supporter seulement la sauvegarde de contenu venant des instances de Framasoft, mais également de toute instances ex-

---

1. <http://framasoftware.org/>

2. cf : <http://framacloud.org/category/news/>

3. Je fais partie du CA de Framasoft. Nous sommes quelques uns à répondre aux tickets de support.

térieures. Ainsi, l'utilisateur peut ajouter des instances de ces logiciels comme il le souhaite. Ceci permet de sauvegarder des contenus provenant d'instances différentes d'Etherpad par exemple.

Pour garder un contenu, il est possible de lancer une sauvegarde à la main en temps voulu. Une sauvegarde de ce contenu sera alors entrée en base de données, et la date d'extraction sera conservée. Les contenus étant amenés à évoluer et le nombre d'URL à sauvegarder pouvant être élevé, *BackupMyCloud* inclut la possibilité de sélectionner les contenus que nous souhaitons sauvegarder en lot. Ainsi, un seul clic permet de garder tous ses contenus importants.

Il est également possible de rechercher un contenu. Pour des mots donnés, la recherche s'effectue à la fois sur le nom, la description et l'intérieur du contenu. Cela permet d'arriver rapidement à l'information voulue.

Si une sauvegarde spécifique veut être récupérée, il suffit de sélectionner le contenu à la date voulue. Le fichier est disponible dans le format choisi.

### 3 Technologies utilisées

Pour l'interface, j'ai choisi de proposer une interface web. Cela permet d'avoir une apparence plus attrayante que ce qu'aurait pu proposer *tkinter*. Une autre alternative envisagée aurait été d'utiliser pyGTK (pour GTK2), pyObject (pour GTK3), ou pyQT (pour QT), mais il m'a semblé que, pour les utilisateurs de Windows, télécharger une librairie de environ 150 Mio pour un logiciel faisant des sauvegardes n'était pas approprié et pourrait être un frein pour les plus néophytes.

J'ai choisi le mini-framework web *bottle* principalement parce qu'il tient en un seul script python de 3891 lignes et propose pour autant beaucoup de fonctionnalités. L'utilité d'avoir une dépendance qui ne tient qu'en un seul fichier est de pouvoir l'inclure dans le logiciel. Ainsi *BackupMyCloud* ne nécessite aucune dépendance extérieure, ce qui facilitera son utilisation. Il est distribuable en l'état. Bien sûr les utilisateurs doivent avoir python3 d'installé sur leurs machines. De plus, cela assure une durée de vie plus longue, puisque le logiciel n'est pas dépendant de bibliothèques extérieures qui peuvent avoir des régressions dans de futures versions.

J'ai également choisi d'utiliser le framework HTML, CSS et JS *Bootstrap*, afin d'avoir un rendu assez esthétique. De plus, le logiciel pourra être utilisé dans le cadre des activités de Framasoft. Le choix de Bootstrap me permet également de rajouter la surcouche *Framacss*<sup>4</sup> pour modifier son allure et être en accord avec la charte graphique de Framasoft.

Ce logiciel est multi-plateforme. Cependant n'ayant qu'un ordinateur sur Linux, je n'ai pu le tester intensivement que sur ce système. Il est donc possible que sur certaines versions de Windows, il n'arrive pas à accéder aux données d'historiques et de favoris des navigateurs. Si c'est le cas, je suis preneur de toute remontée de bugs.

La base de données SQLITE me semble être une bon choix, d'une part grâce au support natif dans python, et d'autre part du fait que les données sauvées seront de tailles assez raisonnables pour qu'un petit *SGBD* puisse le gérer.

Le code est découpé en petites fonctions, principalement dans le fichier *functions.py* pour une réutilisation et une relecture plus facile. Le fichier *main.py* se contente principalement d'appeler ces fonctions et de créer les redirections nécessaires avec *bottle*. Le script *functions.py* regroupe les fonctions par type de tâches effectuées.

L'utilisation de *bottle* permet également d'utiliser des *templates*. Ces derniers permettent d'inclure du code python qui sera évalué pour générer la page HTML finale. Toutes les pages utilisent ces *templates* (fichiers avec l'extension *tpl*) qui sont disponibles dans le dossier *views*. Afin d'avoir un code plus optimal, j'ai créé des templates *header* et *footer* permettant de réutiliser

---

4. <https://framalab.org/framacss/>

facilement ces codes sans redondance. Notons qu'un *template* peut recevoir en argument des variables python, ce qui permet par exemple de mettre en évidence dans la barre de gauche la page actuelle.

## 4 Notes techniques

*BackupMyCloud* supporte l'existence de plusieurs profils de Firefox. Les URL seront importées depuis les historiques et les marques-pages de tous les profils d'un utilisateur (listés dans le fichier de configuration *profile.ini* de Firefox).

L'historique et les marques-pages de Firefox sont stockés dans une base de données SQLITE. L'historique de Chrome et Chromium est aussi en SQLITE, mais les favoris sont dans un fichier JSON. Notons que les marques-pages peuvent être organisés en dossiers et sous-dossiers. *BackupMyCloud* sera capable d'explorer tous ces dossiers et sous-dossiers grâce à la définition récursive de la fonction *retrieve\_chrome\_bookmarks*. Si un ou plusieurs navigateurs parmi Firefox, Chrome, et Chromium n'est pas installé, le logiciel ne plantera pas.

Les URL peuvent indifféremment inclure un caractère '/' à la fin. Le contenu ne sera pas dupliqué. De plus, les URL sont aussi nettoyées en fonction du logiciel. Par exemple, pour Etherpad l'accès à l'historique d'un pad se fait via '/timeslider' à la fin de l'URL. Les URL correspondant à des étapes intermédiaires de création de sondage framadate ne sont pas entrées comme des contenus. De plus, le logiciel est capable de bien gérer le cas où un utilisateur a visité un framadate via l'interface d'administration d'un sondage (privée et réservée au créateur du sondage) et/ou la page publique d'un sondage sur laquelle les sondés peuvent entrer leurs votes. Le contenu ne sera pas dupliqué s'il a visité les deux. En effet, l'URL d'administration est constituée de 8 caractères aléatoires qui s'ajoutent aux 16 caractères aléatoires de l'identifiant d'un sondage<sup>5</sup>.

Dans l'optique de Framasoft d'essaimage des logiciels de Cloud personnel, il était important de permettre de sauvegarder le contenu provenant de n'importe quelle instance d'Etherpad ou de Framadate. Le choix du format d'export semble aussi important pour que chacun puisse bénéficier de ce qui lui convient le mieux. La valeur ajoutée de la configuration du nom et de la description de chaque contenu, ainsi que de la recherche étendue au nom, à la description et au contenu, sont un plus pour des personnes gérant beaucoup de données.

Un début de travail sur l'accessibilité du logiciel a été mené. Il faudrait continuer pour le rendre totalement accessible aux déficients visuels, mais par exemple dans la barre de gauche est ajoutée la balise '`<span class="sr-only">(current)</span>`' pour la page actuelle. Ainsi un navigateur en mode texte, ne voyant pas le surlignage de la page courante, pourra quand même accéder à cette information.

Enfin notons que le logo créé pour le logiciel est visible dans la barre supérieure et en favicon sur toutes les pages.

## 5 Base de données

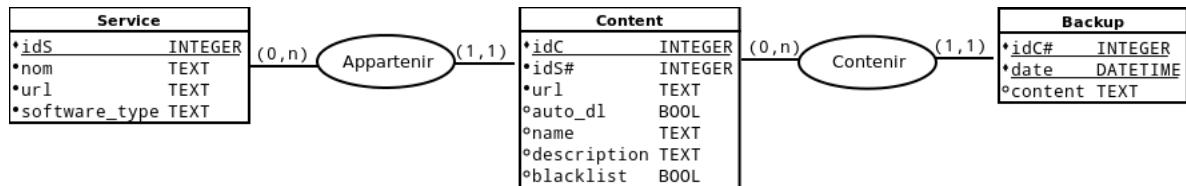
Toutes les informations (à l'exception de la configuration globale stockée dans un fichier JSON) sont stockées dans une base de données SQLITE3. La base de données rassemble ainsi les services configurés, les contenus et les sauvegardes. La figure 1 représente le schéma entité-association de notre base de données.

Les contraintes d'unicité ont été appliquées sur certaines colonnes, comme l'URL ou le nom des services, qui ne sont pas des clés primaires, afin de garantir une intégrité de la base.

---

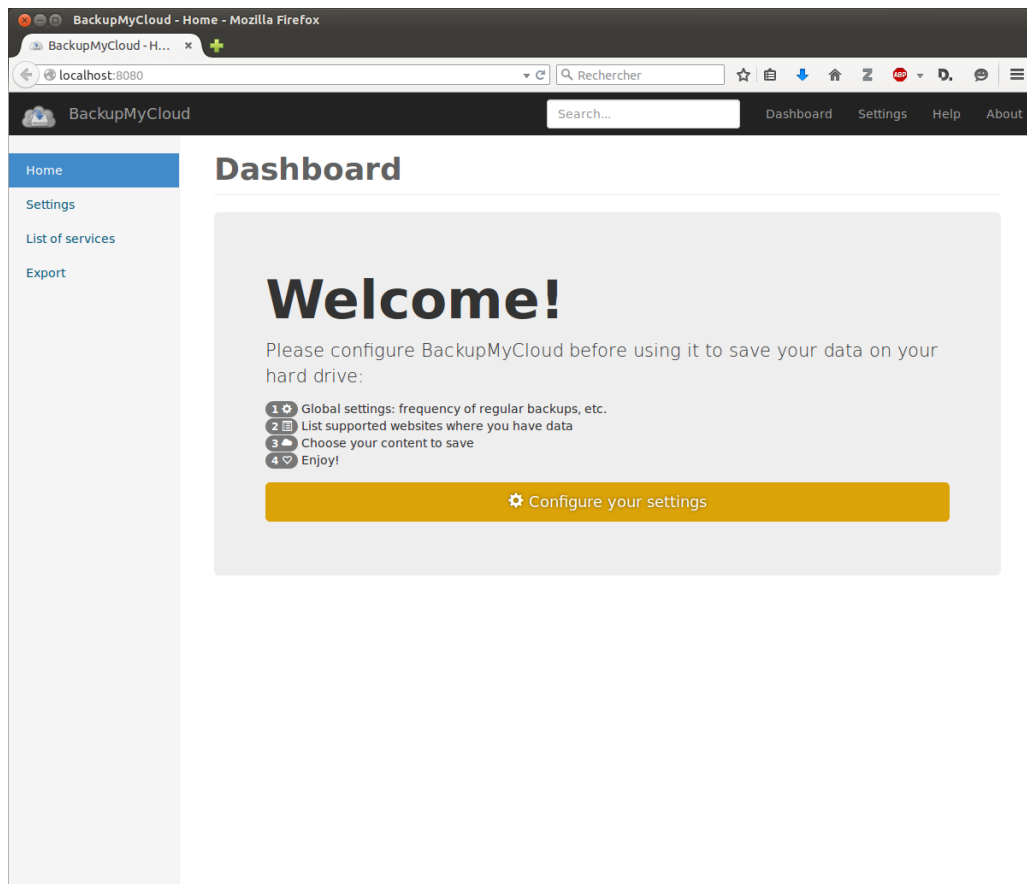
5. Pour plus de détails, voir les lignes 64 et 68 de ce fichier du code source : [https://git.framasoft.org/framasoft/framadate/blob/master/creation\\_sondage.php](https://git.framasoft.org/framasoft/framadate/blob/master/creation_sondage.php)

FIGURE 1 – Schéma entité-association de la base de données



## 6 Utilisation

FIGURE 2 – Page d'accueil

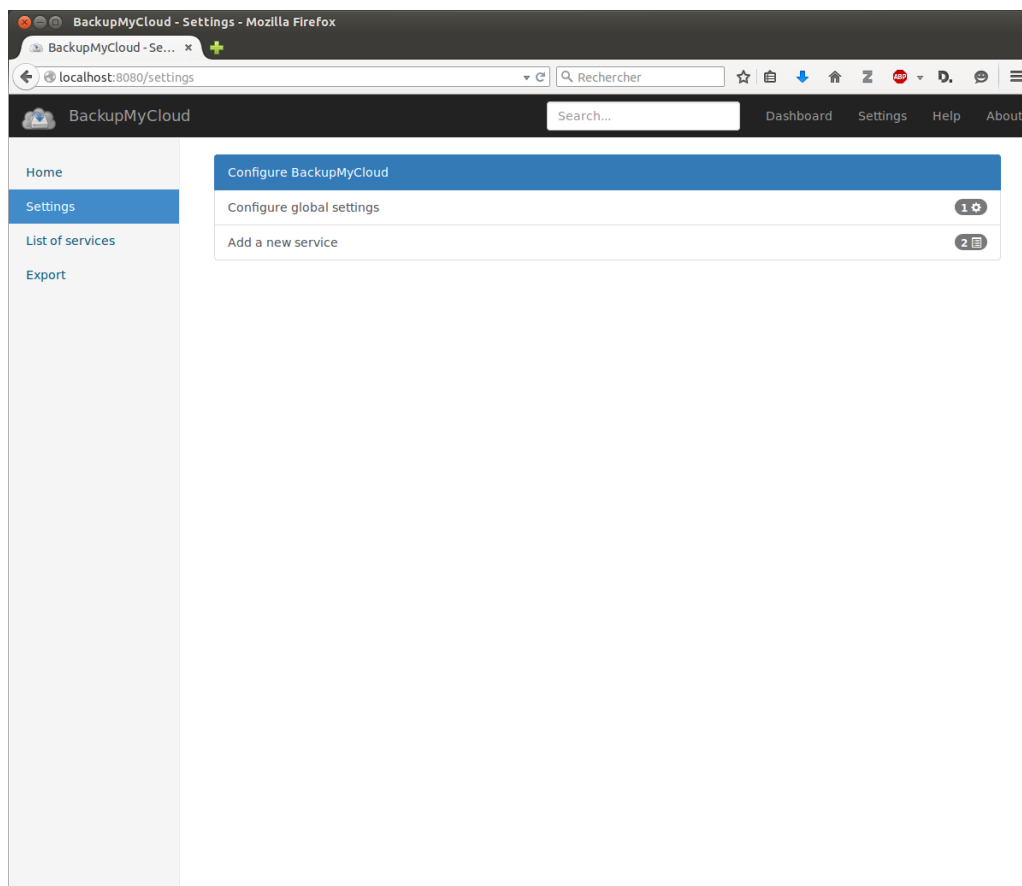


Le logiciel se lance avec le script *main.py*, par exemple avec la commande : *python3 main.py*. Il importe alors la configuration, les modules nécessaires, les fonctions utiles, et si besoin crée un fichier de configuration par défaut et/ou les tables de la base de données SQLITE. Le logiciel importe les URL dans la table *content* des instances des services entrés, depuis l'historique et les marques-pages de Firefox, Chrome et Chromium. Les routes sont définies avec *bottle*, et le mini-serveur web se lance. L'interface est alors accessible à l'adresse <http://localhost:8080/>.

À la première utilisation, la page d'accueil visible en figure 2 est alors affichée. L'utilisateur

est invité à se rendre sur la page principale de la configuration (figure 3) à partir de laquelle, il pourra accéder à la page de choix des formats d'export (figure 4) et à celle d'ajout de nouvelles instances (services) d'Etherpad ou de Framadate (figure 5).

FIGURE 3 – Page de configuration

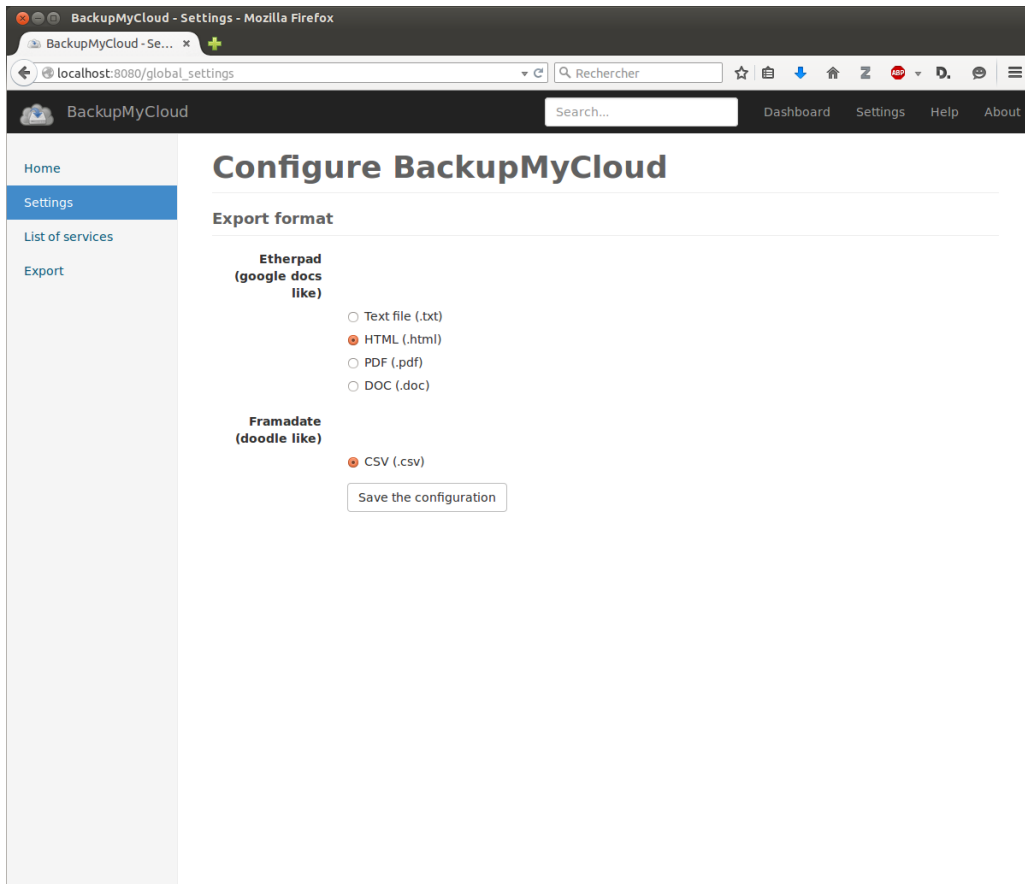


Notons que comme l'indique la figure 6, l'utilisateur reçoit des notifications lors d'une nouvelle entrée. Ces alertes sont vertes en cas de succès, et rouges en cas d'erreur.

Ici nous avons configuré Framapad l'instance d'Etherpad de Framasoft et Framadate. Ces deux instances sont visibles sur la page *List of services*, comme indiqué sur la figure 7. Dans le même temps, le logiciel a importé en base de données, les URL des deux instances depuis l'historique et les marques pages du navigateur.

L'accès aux contenus se fait alors très facilement, soit en sélectionnant le nom du service voulu dans la barre de gauche, soit via la liste des services. La page qui s'affiche alors liste tous les contenus du service sélectionné. Depuis cette page, nous pouvons par exemple ajouter ou modifier les noms et descriptions des contenus. Les noms et descriptions sont automatiquement importés pour Framadate, cependant il faudra les définir à la main pour Etherpad, car Etherpad ne permet pas d'entrer un nom ou une description pour un pad. Dans l'exemple, nous avons modifié la description du premier framadate (figure 8). Une fois validé, une alerte apparaît, nous confirmant la prise en compte de la modification (figure 9). Notons que cette alerte apparaît dans la page au niveau du contenu modifié. De plus, pour tous les changements faits sur cette page,

FIGURE 4 – Configuration du format d'export



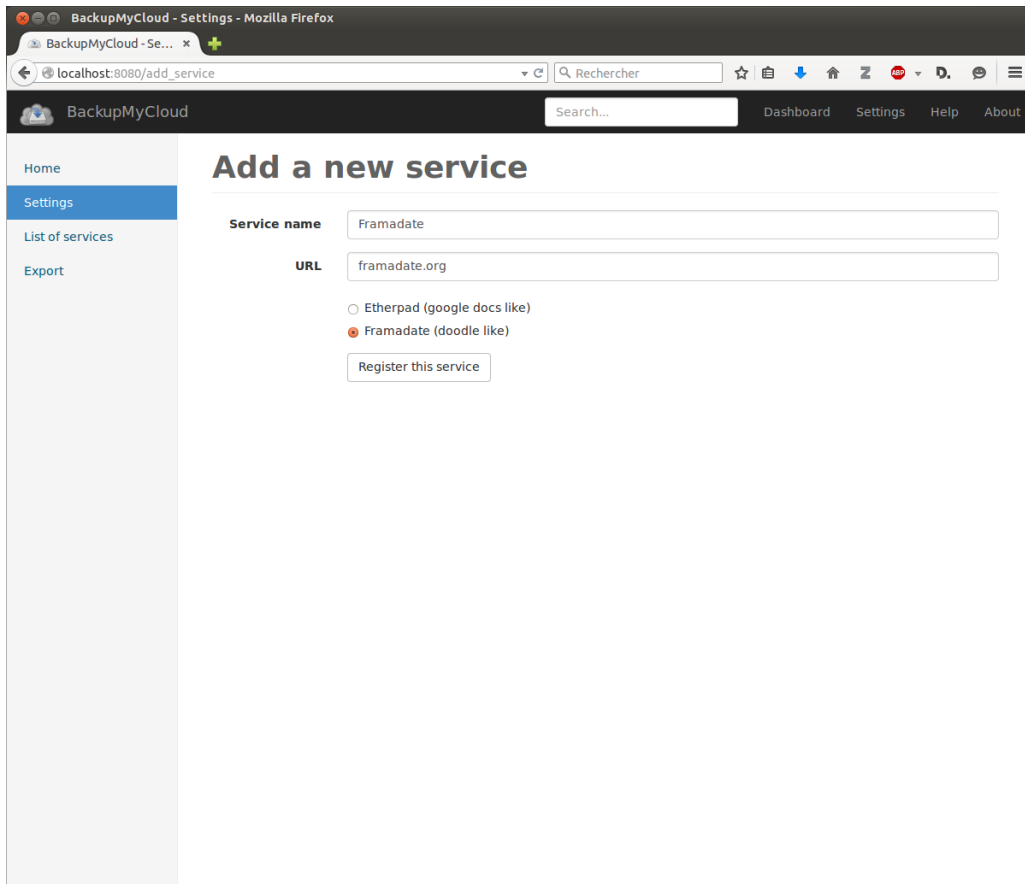
l'utilisateur est ramené au bon endroit grâce à une ancre. Ceci permet d'être efficace et ne pas avoir à constamment naviguer dans la page à chaque modification.

Depuis la figure 9, nous pouvons également voir la liste des sauvegardes disponibles d'un contenu. Ce menu déroulant liste les sauvegardes disponibles par date et heure. Au clic sur l'une d'entre elles, l'utilisateur peut bénéficier directement du fichier dans le format voulu (figure 10). Il n'est pas redirigé et peut une fois le fichier sauvegardé à l'endroit souhaité continuer à récupérer d'autres contenus. Notons également la présence du bouton *Backup Now* qui permet de télécharger et enregistrer en base de données une sauvegarde du contenu choisi. Elle s'ajoutera alors à la liste déroulante, et redirigera l'utilisateur sur la page visible en figure 11, attestant de la réussite (ou non) de l'enregistrement de la sauvegarde.

Un exemple de recherche est disponible en figure 12. Ici, nous avons recherché la description que nous avons entrée précédemment. La requête fonctionne correctement. Notons que le champ de recherche est disponible à tout moment dans la barre supérieure. De plus, chaque recherche s'effectuera à la fois sur le nom, la description et toutes les sauvegardes des contenus. L'utilisateur aura accès aux mêmes fonctionnalités que sur la page listant les contenus d'un service.



FIGURE 5 – Ajout d'un nouveau service



## 7 Licences

*BackupMyCloud* fonctionne grâce à :

- Bottle Python Web Framework : <http://bottlepy.org/docs/dev/index.html>
- Python3 : <https://www.python.org/>
- Bootstrap HTML, CSS, and JS framework : <http://getbootstrap.com/>
- La CSS accessible de Bootstrap modifié par Framasoft : <https://framalab.org/framacss/>

*BackupMyCloud* est distribué sous la licence *GNU GPL3*. Il inclut les logiciels tiers sous les licences suivantes :

**Bottle** MIT license

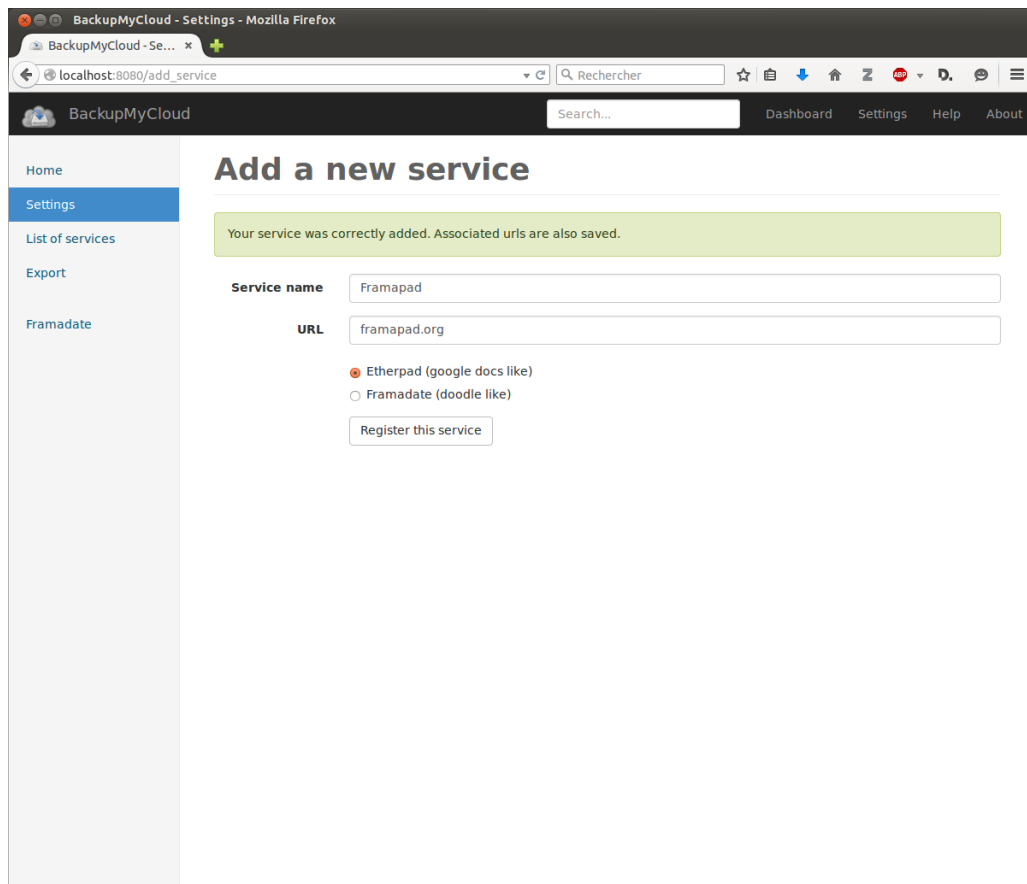
**Bootstrap** MIT license

**Framasoft Bootstrap** MIT license

Le logo de *BackupMyCloud* a été créé par moi-même à partir des images suivantes :

- The tango document save icon, under Public Domain : <https://openclipart.org/detail/32245/tango-documnet-save>

FIGURE 6 – Notification de l'ajout d'un nouveau service



- The Stylized Basic Cloud icon, under Public Domain : <https://openclipart.org/detail/179894/stylized-basic-cloud>

FIGURE 7 – Liste des services configurés

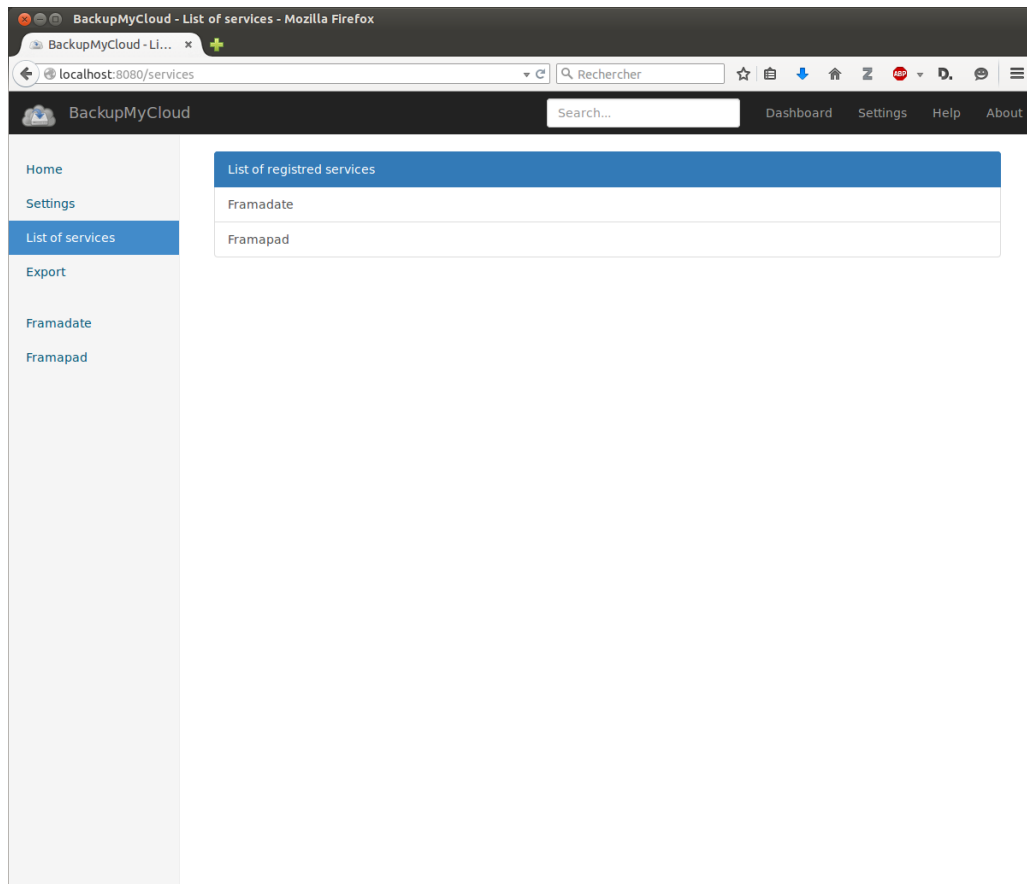


FIGURE 8 – Exemple de liste de contenus d'un service

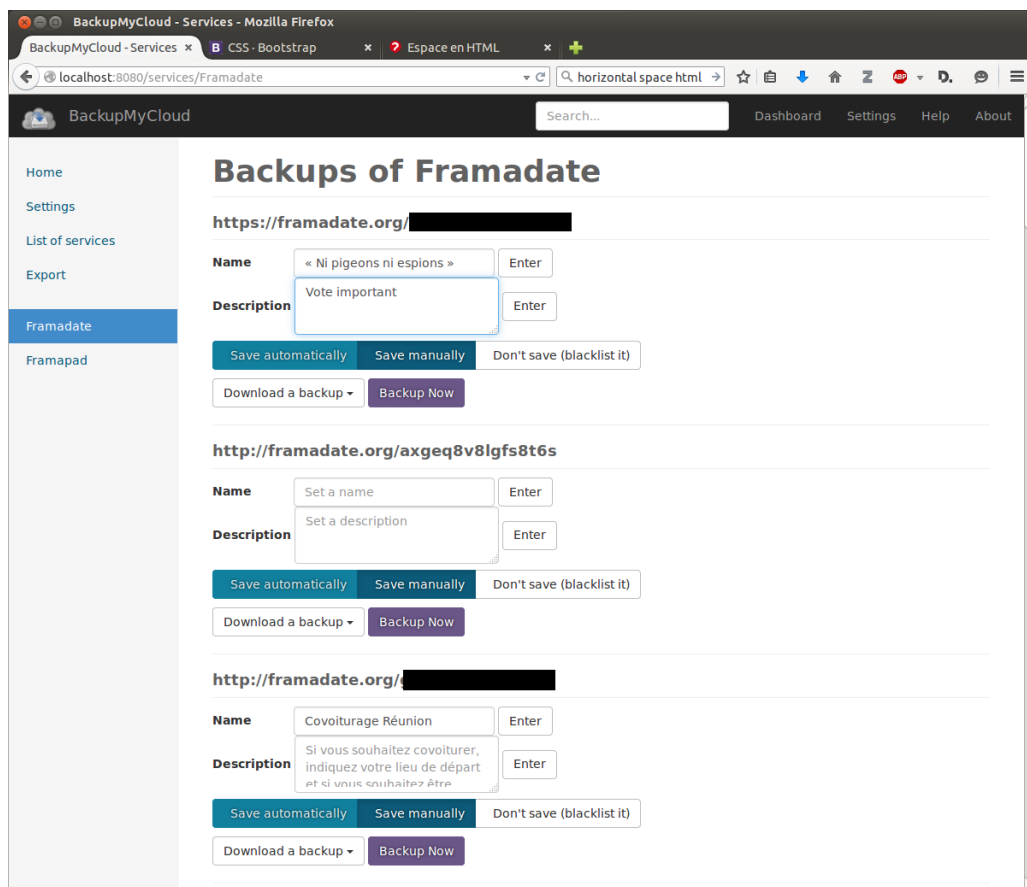


FIGURE 9 – Notification de la modification d'un champ et liste des sauvegardes

BackupMyCloud - Services - Mozilla Firefox

BackupMyCloud - Services x CSS - Bootstrap x Espace en HTML x

localhost:8080/services/Framadate#anchor\_1

horizontal space.html

BackupMyCloud Search...

Dashboard Settings Help About

Home

Settings

List of services

Export

Framadate

Framapad

## Backups of Framadate

https://framadate.org/ [redacted]

Description correctly updated.

**Name** « Ni pigeons ni espions » Enter

**Description** Vote important Enter

Save automatically Save manually Don't save (blacklist it)

Download a backup Backup Now

Dates of backups

rg/axgeq8v8lgs8t6s

2015-05-03 23:01:29

2015-05-03 23:02:15

**Description** Set a description Enter

Save automatically Save manually Don't save (blacklist it)

Download a backup Backup Now

http://framadate.org/ [redacted]

**Name** Covoiturage Réunion Enter

**Description** Si vous souhaitez covoiturer, indiquez votre lieu de départ et si vous souhaitez être Enter

Save automatically Save manually Don't save (blacklist it)

FIGURE 10 – Téléchargement d'une sauvegarde

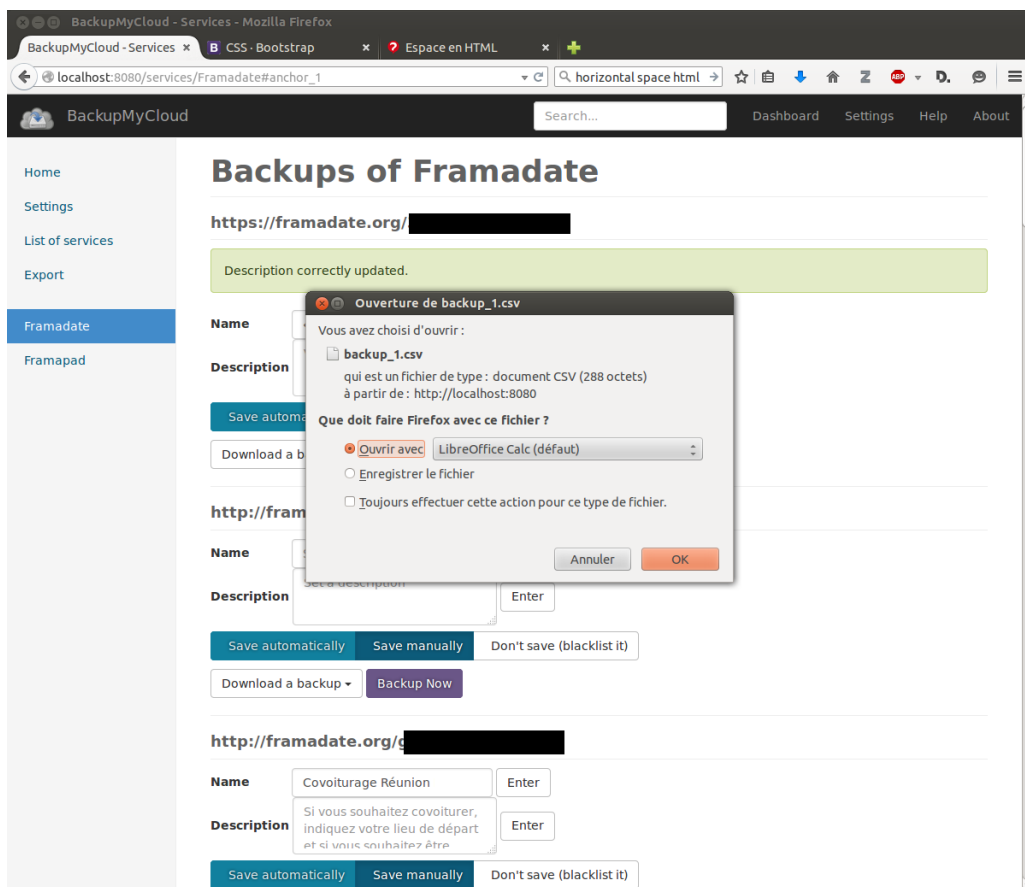


FIGURE 11 – Notification de l'enregistrement d'une sauvegarde manuelle

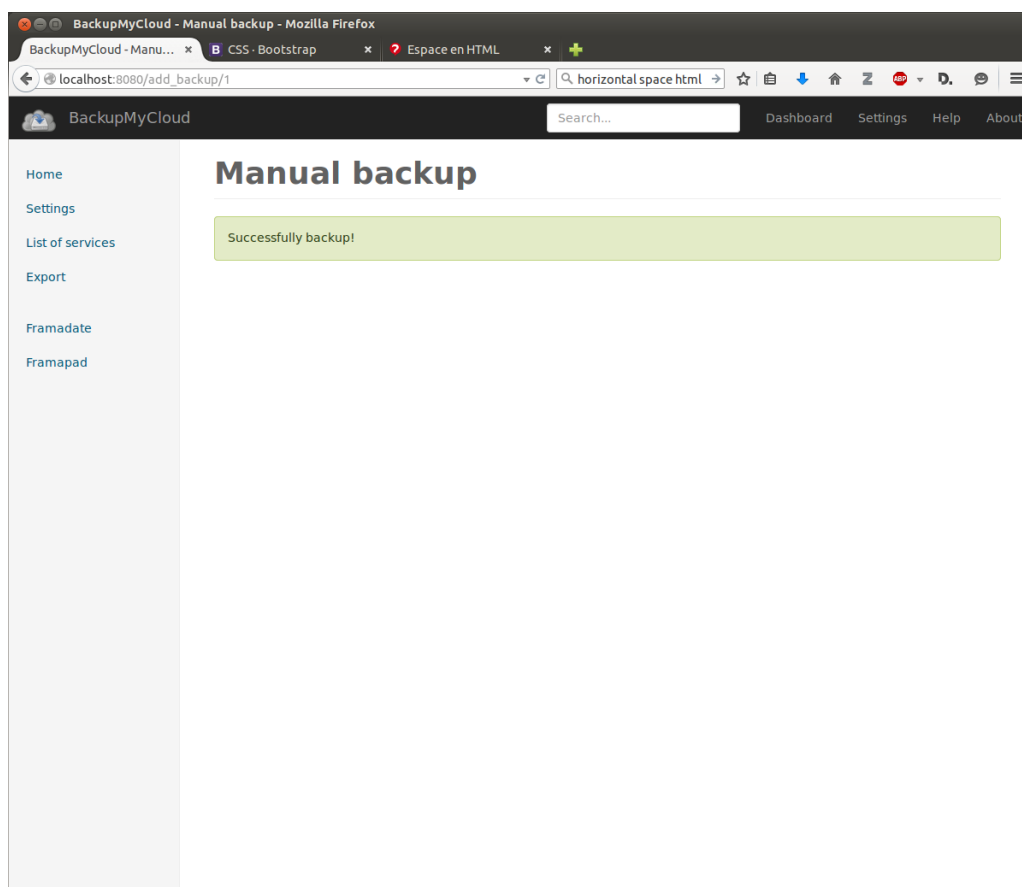


FIGURE 12 – Résultat de recherche

The screenshot shows a web browser window with the title 'BackupMyCloud - Search - Mozilla Firefox'. The address bar shows 'localhost:8080/search' and the search query 'horizontal space html'. The page header includes the 'BackupMyCloud' logo and a search bar containing 'vote important'. The main content area is titled 'Results of the search of vote important' and displays the URL 'https://framadate.org/'. Below this, there are input fields for 'Name' (containing '« Ni pigeons ni espions »') and 'Description' (containing 'Vote important'), each with an 'Enter' button. At the bottom, there are three buttons: 'Save automatically', 'Save manually', and 'Don't save (blacklist it)'. A 'Download a backup' dropdown menu is also visible, with a 'Backup Now' button next to it.

BackupMyCloud - Search - Mozilla Firefox

BackupMyCloud - Se... CSS - Bootstrap Espace en HTML

localhost:8080/search horizontal space html

BackupMyCloud vote important Dashboard Settings Help About

### Results of the search of vote important

<https://framadate.org/>

**Name** « Ni pigeons ni espions » Enter

**Description** Vote important Enter

Save automatically Save manually Don't save (blacklist it)

Download a backup Backup Now