

FOLLOW ME










READ ME

Benjamin NOEL – Sakr LIMEM

Contents

I.	Technologies utilisés	3
I.	Présentation de l'application	4
II.	Fonctionnalités	4
III.	Architecture de l'application (Vue sous Eclipse)	6
IV.	Interface Graphique	11

I. Technologies utilisés

Environnement de développement	
Serveur web	 Apache Tomcat
Plateforme de développement Web (WampServer)	
Base de données	
Outil pour la gestion et l'automatisation de production	
Framework	
Langage web	
Object Relational Mapping Library	 HIBERNATE
Service d'hébergement et de gestion de développement	

I. Présentation de l'application

Follow Me est une petite application web qui permet de créer une activité sportive, artistique, ou autres, de l'ajouter sur une google map. Cette activité sera visible par tous les utilisateurs, et ils pourront y participer si le créateur de cette activité les accepte.

Cette application est développée selon les règles de l'art du Java J2EE. Nous avons mis en place une architecture flexible et scalable grâce à l'externalisation de tous les fichiers de paramétrages, de configurations, de messages, mais aussi des imports Java Script CSS et JQuery.

Nous n'avons pas pu implémenter toutes les fonctionnalités sur Follow Me.

Nous n'avons pas eu non plus beaucoup de temps pour tester cette version de l'application, et gérer les anomalies qu'il y'a dessus.

L'application est développée selon une architecture MVC, qui vise à être légère, mais étant donné que l'application est développée en Java, pour obtenir un temps de réponse raisonnable, l'application doit tourner sur un bon serveur.

II. Fonctionnalités

Authentification

L'utilisateur accède à l'URL de l'application

L'application Web se réfère à web.xml

Le web.xml correspond au modèle d'URL

Le contrôle sera redirigé vers DispatcherServlet dans le framework Spring

Le Framework Spring constate que tous les URL sont sécurisés et affiche donc la page de connexion à l'utilisateur

L'utilisateur entre son login et mot de passe

Spring Security vérifie l'authentification et l'autorisation avec la base de données et si les informations d'identification sont valides, il ouvre la session et redirige vers l'URL consultée initialement.

Autorisations

En fonction de son rôle ou autrement dit de son profil chaque utilisateur peut avoir accès à des fonctionnalités et d'autres pas, cet aspect est aussi géré grâce à Spring Security. Nous avons mis en place ce mécanisme mais étant donné que nous n'avons pas assez de temps pour implémenter toutes les fonctionnalités et que l'objectif du projet est de réaliser une petite application, nous nous sommes limités aux fonctionnalités de base dans cette version de l'application.

Validation du compte par mail

Lorsqu'un utilisateur crée un compte, nous lui envoyons un mail avec un lien d'activation, contenant un numéro unique stocké auparavant sur la base de données. Ce lien lui permettra de valider le mail que nous lui transmettons, d'activer son compte, et de pouvoir l'utiliser, autrement le compte restera bloqué.

Internationalisation (Anglais/Français)

Nous utilisons Spring pour gérer l'internationalisation. Nous indexons les messages dans deux fichiers de propriétés différents en deux langues différentes, et nous les appelons dans les pages par leurs index pour pouvoir recharger tous les messages lorsque l'utilisateur change la langue.

Enregistrement de la latitude et de la longitude de l'utilisateur sur la Base de données

Lorsqu'un utilisateur crée un compte et qu'il saisit son adresse, nous utilisons un HttpClient d'Apache pour récupérer sa longitude et sa latitude, les enregistrer sur la base, et les réutiliser par la suite.

Gestions des notifications

Lorsqu'une personne demande de participer à un événement, un mail est envoyé à l'organisateur, et une notification sur sa page. Et lorsque l'organisateur accepte un participant ce dernier est prévenu aussi par mail.

III. Architecture de l'application (Vue sous Eclipse)

<ul style="list-style-type: none"> follow-me [followMe master] <ul style="list-style-type: none"> Spring Elements Deployment Descriptor: follow-me JAX-WS Web Services Java Resources <ul style="list-style-type: none"> src/main/java <ul style="list-style-type: none"> fr.esiea.web.bean <ul style="list-style-type: none"> AccountFormBean.java 	<p>Les Forms Beans sont des classes java qui mappent les formulaires présents sur les pages web de l'application.</p>
<ul style="list-style-type: none"> fr.esiea.web.controller <ul style="list-style-type: none"> AccountController.java ConfigController.java HomeController.java LoginLogoutController.java MainController.java 	<p>Tous les contrôleurs héritent d'une même classe mère ConfigController qui regroupe l'ensemble des méthodes qui seront utilisés par tous les contrôleurs. Ces méthodes sont des méthodes globales telles que la méthode qui retourne les données de l'utilisateur connecté, la méthode d'accès aux fichiers de ressources pour la gestion de l'internationalisation, et la méthode pour envoyer des mails.</p>
<ul style="list-style-type: none"> fr.esiea.web.security <ul style="list-style-type: none"> CustomAuthenticationManager.java UserListFilter.java 	<p>CustomAuthenticationManager est une classe invoqué par Spring pour la gestion de l'authentification, et du login.</p>
<ul style="list-style-type: none"> fr.esiea.web.model <ul style="list-style-type: none"> ActivityCategory.java ActivityDetail.java ActivityName.java ActivityParticipants.java Adress.java User.java 	<p>Ces classes représentent les entités qui mappent la base de données grâce aux annotations d'Hibernate</p> <pre>@Id @Column(name="id") @GeneratedValue(strategy=GenerationType.IDENTITY) public int getId() { return id; }</pre> <p>Hibernate facilite la persistance et la recherche de données dans une base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant.</p>

```

fr.esiea.web.dao
├── ActivityCategoryDao.java
├── ActivityCategoryDaoImpl.java
├── ActivityDetailDao.java
├── ActivityDetailDaoImpl.java
├── ActivityNameDao.java
├── ActivityNameDaoImpl.java
├── ActivityParticipantsDao.java
├── ActivityParticipantsDaoImpl.java
├── AdressDao.java
├── AdressDaoImpl.java
├── UserDao.java
└── UserDaoImpl.java

```

```

fr.esiea.web.service
├── ActivityCategoryService.java
├── ActivityCategoryServiceImpl.java
├── ActivityDetailService.java
├── ActivityDetailServiceImpl.java
├── ActivityNameService.java
├── ActivityNameServiceImpl.java
├── ActivityParticipantsService.java
├── ActivityParticipantsServiceImpl.java
├── AdressService.java
├── AdressServiceImpl.java
├── UserService.java
└── UserServiceImpl.java

```

```

src/main/resources
├── META-INF
│   ├── create_db_followMe.sql
│   ├── log4j.xml
│   ├── messages_en.properties
│   └── messages_fr.properties
├── src/test/resources
├── Libraries
├── JavaScript Resources
├── Deployed Resources
└── src
    ├── main
    │   ├── java
    │   └── resources

```

Le gros avantage par rapport aux serveurs d'application est qu'avec Spring, les classes n'ont pas besoin d'implémenter une quelconque interface pour être prises en charge par le framework (au contraire des serveurs d'applications J2EE et des EJBs). C'est en ce sens que Spring est qualifié de conteneur « léger ».

Spring s'appuie principalement sur l'intégration de trois concepts clés :

- l'inversion de contrôle ou injection de dépendance (IoC).
- la programmation orientée aspect (AOP).
- une couche d'abstraction.

Les messages.properties (en/fr) regroupent l'ensemble de messages présents sur les pages web. L'utilisation de ces fichiers de propriétés permet la gestion de l'internationalisation grâce à la classe [org.springframework.context.support.ReloadableResourceBundleMessageSource](https://docs.spring.io/spring-framework/docs/current/javadoc-api/org.springframework.context.support.ReloadableResourceBundleMessageSource.html) du framework Spring.

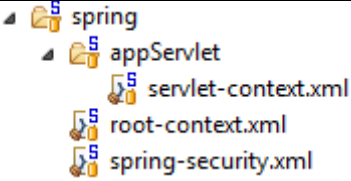
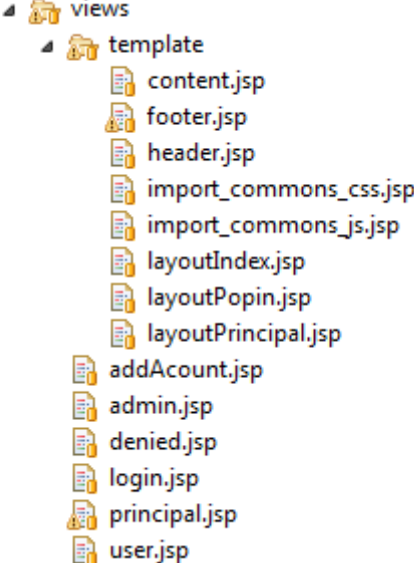
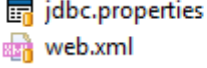
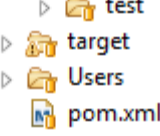
Le fichier create_db_followMe.sql contient le script de création de la base de données ainsi que de ses tables.

<ul style="list-style-type: none"> webapp <ul style="list-style-type: none"> resources <ul style="list-style-type: none"> css <ul style="list-style-type: none"> datepicker.css ie.css jquery.fancybox-1.3.4.css principal_popin.css principale.css table.css font <ul style="list-style-type: none"> Montserrat 	<p>Les css sont tous organisé dans ce même dossier, et importés grâce à Tiles Apache en fonction de la template dans laquelle s'ouvre la page. Ils seront tous importé d'une façon factorisée dans un même fichier import_common_css.jsp</p>
---	--

<ul style="list-style-type: none"> images <ul style="list-style-type: none"> date_picker fancybox jquery-ui activityResume.png adresse.png amateur.png badmington.png basket.png birth.png blank.png bowling.png close.png detente.png 	<p>Les images utilisées par les pages.</p>
---	--

<ul style="list-style-type: none"> jquery <ul style="list-style-type: none"> jquery.js js <ul style="list-style-type: none"> lang api-map.js dataTables.js datepicker.packed.js FollowMe.js infobox.js jquery.fancybox-1.3.4.pack.js jquery.min.js pin.js scrollfix.js 	<p>Les scripts et librairies JavaScript et JQuery sont tous organisé dans ce même dossier et seront aussi tous importé d'une façon factorisée dans un même fichier import_common_js.jsp</p>
---	---

<ul style="list-style-type: none"> WEB-INF <ul style="list-style-type: none"> layouts <ul style="list-style-type: none"> layouts.xml views.xml 	<p>Afin de simplifier le développement des interfaces utilisateur de l'application nous avons utilisé Apache Tiles, qui permet de définir des fragments de pages qui seront assemblés en une seule page à l'exécution. Ces fragments, ou ces Tiles, peuvent être utilisés facilement tout en réduisant la duplication des éléments communs de la page.</p> <p>Ça permet aussi d'externaliser les imports de css ou des javascript.</p>
--	--

 <p>spring appServlet servlet-context.xml root-context.xml spring-security.xml</p>	<p>Dans ces fichiers nous ajoutons toutes les fonctionnalités du framework spring que nous allons utiliser pour développer : Gestion de la base de données, gestion des services, gestion des mails, gestions de l'internationalisation, sécurisation de l'application (Login/Logout/Authentification).</p>
 <p>views template content.jsp footer.jsp header.jsp import_commons_css.jsp import_commons_js.jsp layoutIndex.jsp layoutPopin.jsp layoutPrincipal.jsp addAcount.jsp admin.jsp denied.jsp login.jsp principal.jsp user.jsp</p>	<p>Ces pages s'ouvriront toutes dans les templates de Tiles Apache. Ce sont des pages JSP mais qui contiennent du HTML5 et du CSS3.</p>
 <p>jdbc.properties web.xml</p>	<p>Nous avons externalisé les données de configuration dans le fichier jdbc.properties de l'accès à la base de données pour pouvoir les changer facilement. Le web.xml est le cœur d'une application spring il permet de charger les contextes et lancer l'application.</p>
 <p>test target Users pom.xml</p>	<p>Le pom.xml est le fichier nécessaire à maven pour importer les dépendances et créer le war pour le déploiement de l'application sur Tomcat Apache Voici la liste des dépendances dont nous avons eu besoin</p> <p>Spring dependency :</p> <ul style="list-style-type: none"> • spring-context • spring-webmvc • spring-web • spring-context • spring-tx • spring-core • spring-orm • spring-security-core • spring-security-config • spring-security-web • spring-context-support

Sql dependency:

- mysql-connector-java
- commons-dbcp
- commons-digester3
- commons-beanutils

Hibernate dependency:

- hibernate-core
- hibernate-entitymanager

Mail dependency :

- javax-mail

Tiles Apache dependency

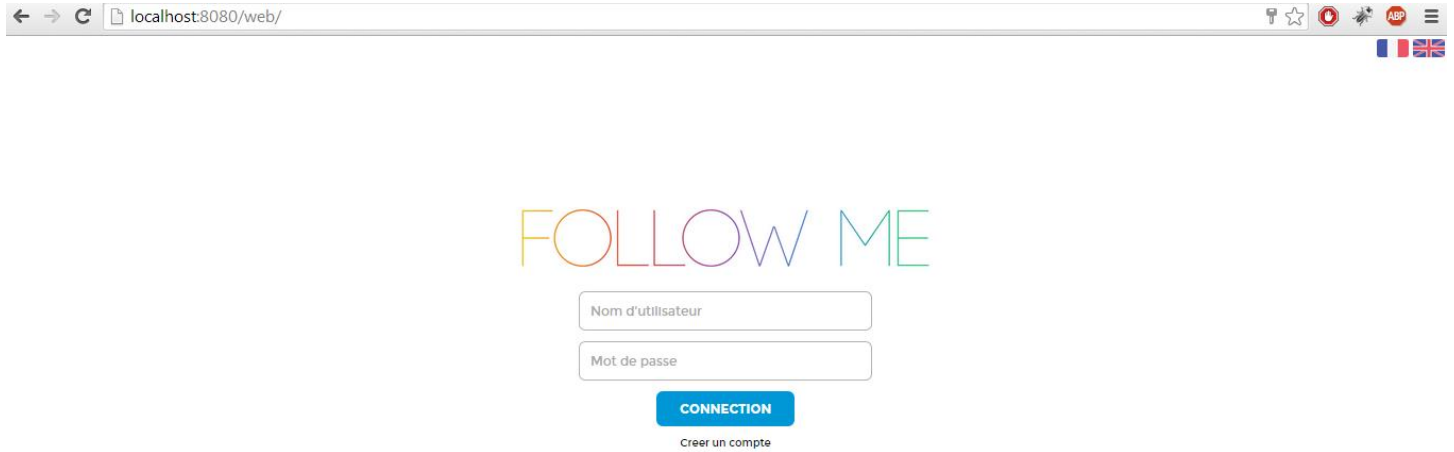
- tiles-extras
- tiles-jsp
- tiles-core
- tiles-servlet

Logger, Test, and Other dependency

- xercesImpl
- javax.inject
- servlet-api
- jsp-api
- slf4j-api
- jcl-over-slf4j
- slf4j-log4j12
- log4j
- junit

IV. Interface Graphique

Ecran de Login



localhost:8080/web/

FOLLOW ME

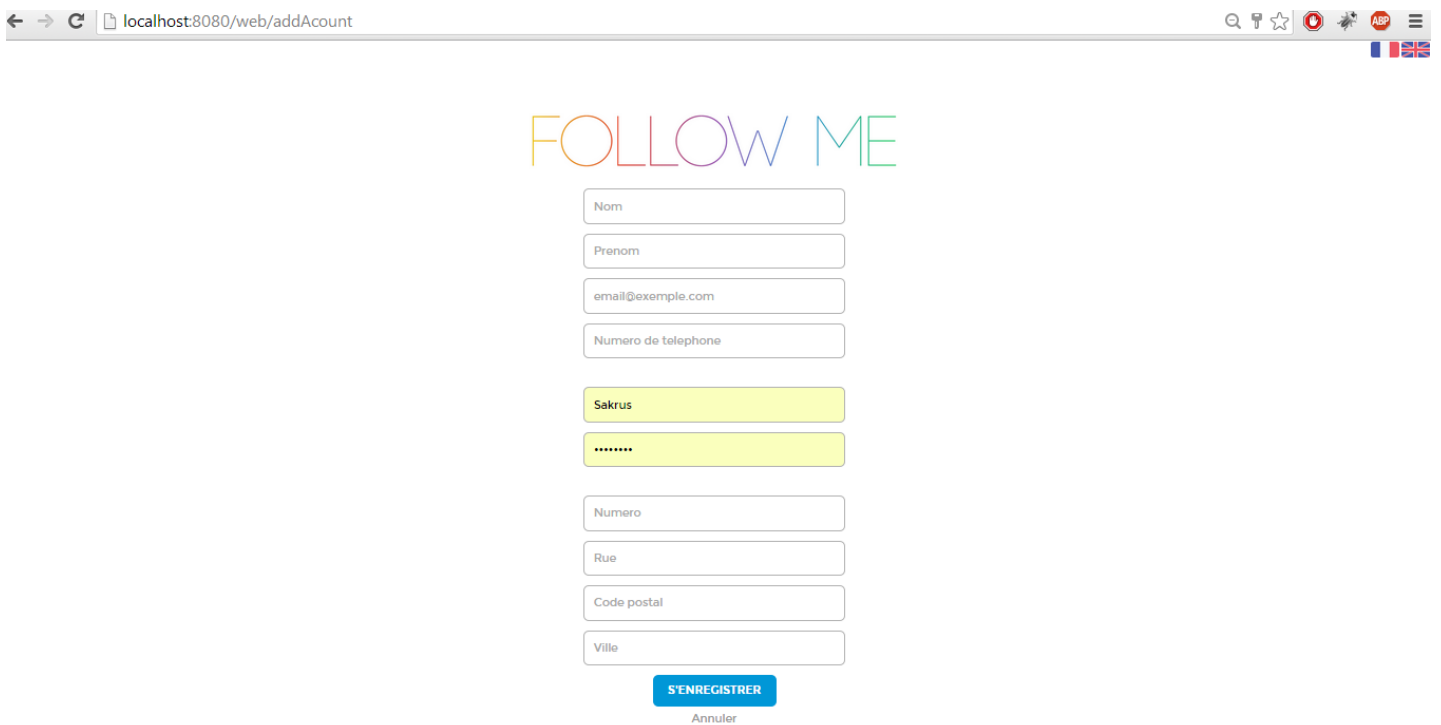
Nom d'utilisateur

Mot de passe

CONNECTION

Créer un compte

Ajout d'un compte



localhost:8080/web/addAccount

FOLLOW ME

Nom

Prenom

email@exemple.com

Numero de telephone

Sakrus

.....

Numero

Rue

Code postal

Ville

S'ENREGISTRER

Annuler

Enregistrement

FOLLOW ME

Nom

Prenom ! Veuillez renseigner ce champ.

email@exemple.com

Numero de telephone

Vérification des champs

localhost8080/web/addAccount

FOLLOW ME

Sakr

LIMEM

limem.sakr@gmail.com

0786480904

Sakrus

22

Rue des freres voisins

92130

Issy les Moulineaux

S'ENREGISTRER

Annuler

Le mail et le lien de validation

Follow Me Boîte de réception x

followmetechicalmail@gmail.com 02:18 (Il y a 1 minute) ☆

À moi ▾

anglais > français Traduire le message Désactiver pour : anglais x

Follow Me: Welcome to our platform for organizing activities. Starting today you are a member of the Follow Me app. You can organize or participate in activities. <http://localhost:8080/web/activateAccount?pin=33d2e43b-a31d-4adb-8c2b-ee86e3353c40>

Activation du compte



FOLLOW ME

CONNECTION

[Créer un compte](#)

En cas d'erreur



FOLLOW ME

CONNECTION

[Créer un compte](#)

You have entered an invalid username or password!

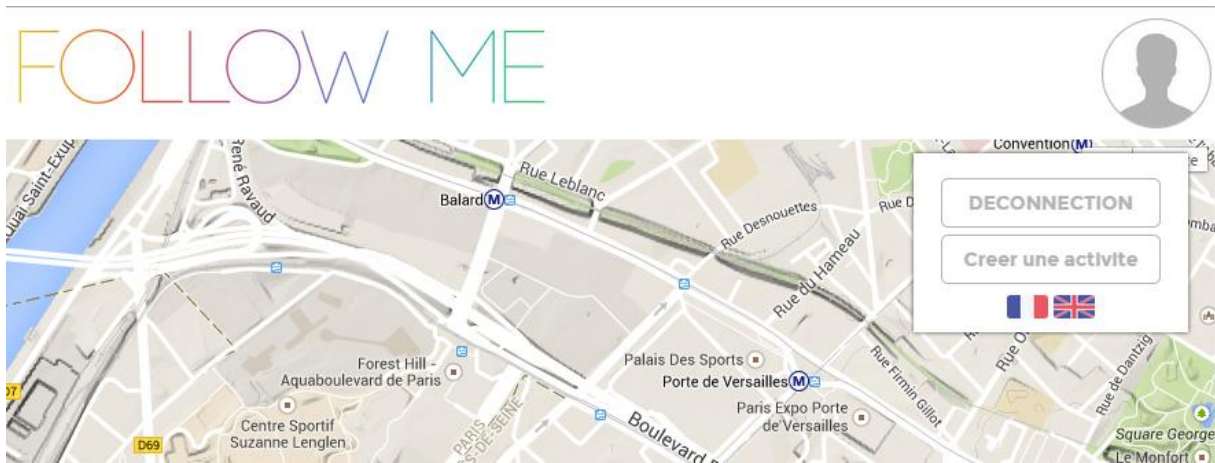
Accès à l'application



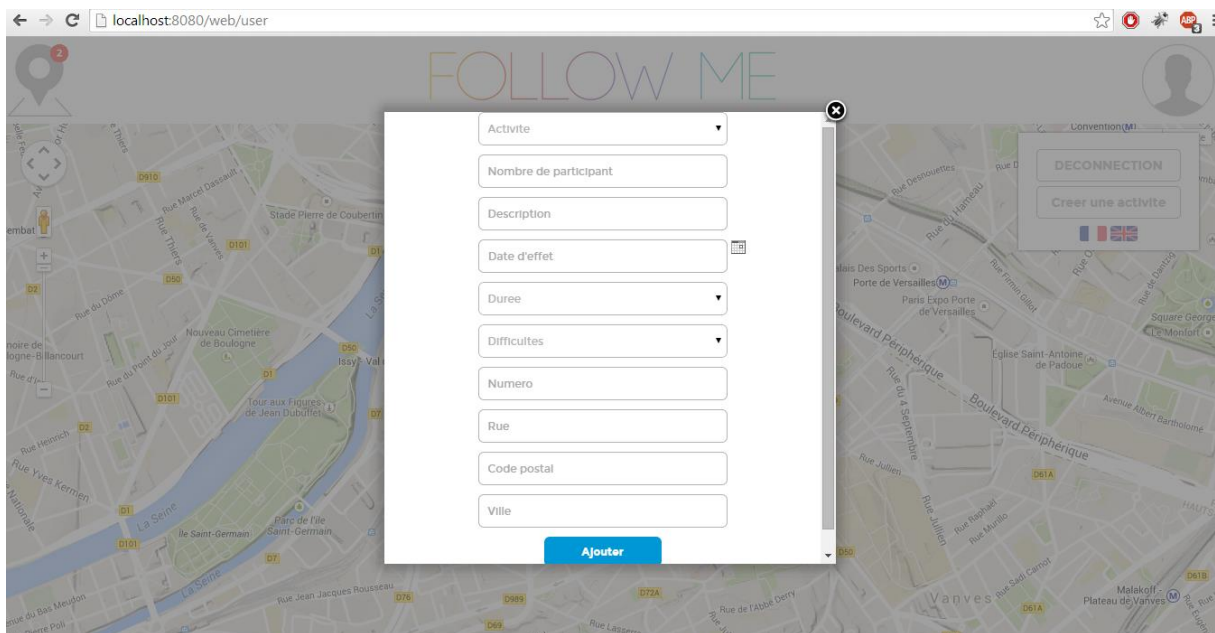
Les actions liées aux notifications



Les actions de l'utilisateur

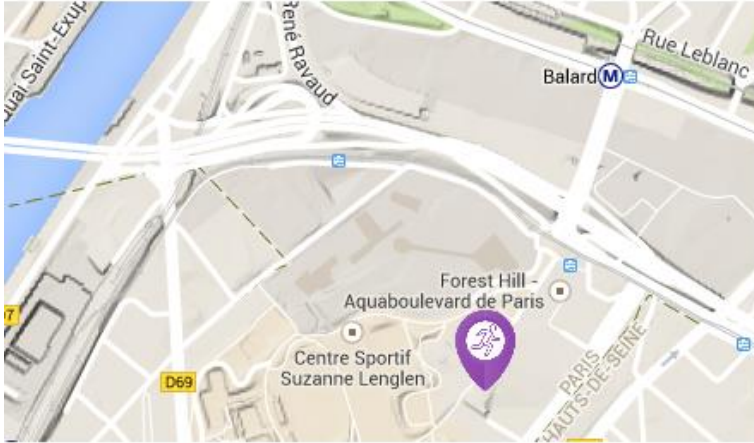


Création d'une activité (Fonctionnalité désactivé sur cette version)



Outils prévu pour la gestion des Marker Map

FOLLOW ME

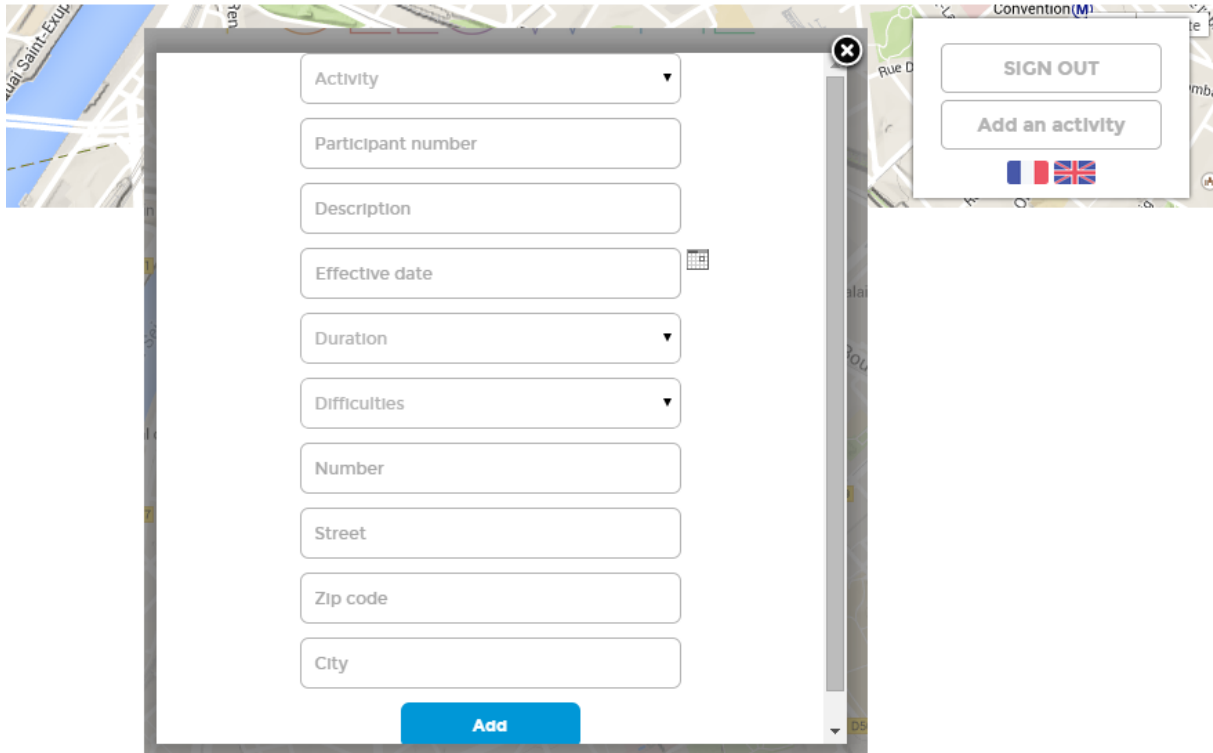


FOLLOW ME




Internationalisation

FOLLOW ME

Activity
Participant number
Description
Effective date
Duration
Difficulties
Number
Street
Zip code
City
Add

SIGN OUT
Add an activity


Vue d'ensemble



localhost:8080/web/user?locale=en

2

FOOTING
proposé par Caston

Adresse de l'activité :
Parc Frédéric Pic
92170, Vanves

40 min Détenue 03/10

PARTICIPER !

Contactez l'organisateur

Escalade : Sakr LIMEM Accept

Escalade : Benjamin NOEL Accept

SIGN OUT

Add an activity

