OmniAuth

# Qu’est-ce que c’est OmniAuth ?

Aujourd’hui nous avons plein de sites à s’inscrire. Ce n’est vraiment pas possible de mémoire tous les noms d’utilisateur et mots de passe. Donc le site offert une autre façon de s’inscrire en utilisant un compte d’un autre site comme Google, Facebook, twitter etc. En ce moment-là, OmniAuth joue le rôle d’obtenir l’authentication d’autre site (authentication fournisseur) pour le nouveau compte.

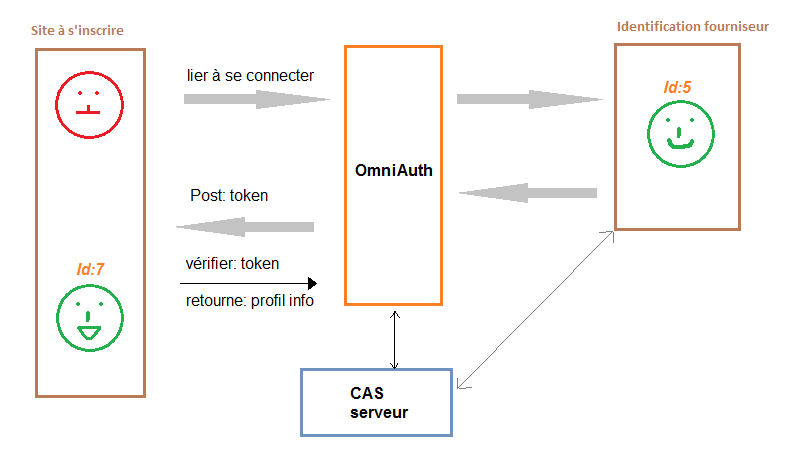
« *OmniAuth est une librairie qui standardise multifournisseur authentication pour web les applications. Il est une Ruby authentication framework qui sert à simplifier de*  »

# Qu’est-ce que c’est Authentication ?

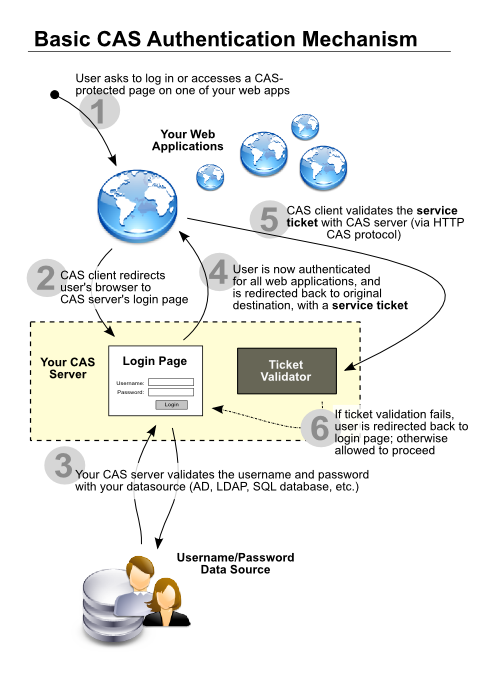
Authentication est un identification token/ticket.

Un site peut avoir plusieurs authentication fournisseurs. Un utilisateur peut avoir plusieurs authentications.

# Comment ça fonctionne ?



OmniAuth manie l’autorisation pour la 3ème partie. Par exemple, si tu veux ton application à intégrer avec Twitter. Tu veux accéder quelque 3ème partie service ou ressource comme un utilisateur sans mdp. L’application demande Twitter pour l’accès. L’utilisateur autorise (sur Twitter) puis l’app a l’accès.

CAS Authentication Mécanisme

Source lien : <https://rubycas.github.io/>

CAS centralise authentication qui permet de s’inscrire un serveur unique(toutes les applications sont configurées à un serveur unique). Mais il ne manie pas l’autorisation. OmniAuth manie l’autorisation pour la 3ème partie.

Ruby on Rails

## Modèle MVC

**Dispatcher**

**Routes**

**Web Server**

**Browser**

**Controller**

**View**

**Model**

**MySql**

HTTP

HTML

Html template est rempli de données de controller

Html rempli est retourné au Browser

Figure 1: Modèle MVC du Ruby on Rails

*Ruby on Rails* est un MVC Framework complet qui peut séparer efficacement la manière de transfert de ressource et la base de données.

* **Controller** : le *Controlller* reçoit le résultat que le Route analyse, détermine laquelle controller va être appliquée et mettre le controller à exécuter certain action. C’est-à-dire, un « Http Request » est envoi par l’utilisateur(Browser) à « Routing ». Un fichier « route.rb »(Routes) assigne certain « Action Controller ». Après l’action du controller va être appelée. En le même temps, le controller exchange des données avec *Model* et *View.*
* **View**: View est la page HTML qui est remplie des données via controller.
* **Model**: il fournit une interface de sauvegarder et interroger la base de données.

## Mapping URLs du RailsP:\Jing\rails-flow-mapping.png

Figure : Mapping flow du Rails

L’url cartographie à ses actions respectivement comme le fig. 2 . Chaque url est chargé par une action du controller. L’action est une méthode qui manie les urls, les interactions avec la base de données et rendre une HTML page.

Dans l’exemple du fig.2, on a des actions *index, add, mark, unmark, et delete* du controller