OAuth2 et Authentification

Enregistrement auprès du service

Google

Créer une application avec la console Google

1. **Activer l'API Google+** pour pouvoir récupérer les informations de profil des utilisateurs voulant se connecter au site

2. Ajouter des **identifiants** (« ID client OAuth ») … « application Web »
- Url d'origines JavaScript autorisées (pour les cors) exemple « http://localhost »
- Uri de redirection (qui sera utilisée en « redirect_uri » pour récupérer le code ou tokens)

Exemple « http://localhost/demo/signin-google »

Cela peut être la même page que la page que celle contenant le bouton de connexion ou une autre page. C'est depuis cette url/page que l'on va récupérer le code/token (avec un $_GET[« code »] en php) et le token

Récupérer le **client id** et **client secret**.

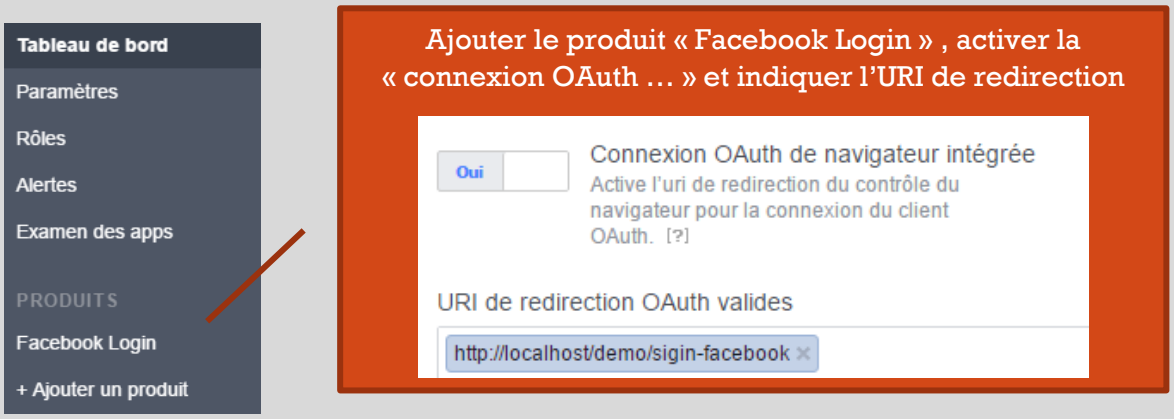
Facebook

Créer une application sur Facebook developers

1. Dans les paramètres
   - Ajouter une plate-forme (exemple « Site Web »). Indiquer l'url du site (exemple « http://localhost/demo »)
   - Définir le domaine de l'application (exemple « localhost »)

2. Dans l'onglet « Examen des apps » passer l'application en « publique »

3. Définir une « redirect_uri »



« Full serveur »

1 CODE

Demande de **code (redirection vers le service ou popup)**
GET
- https://accounts.google.com/o/oauth2/auth
- https://www.facebook.com/dialog/oauth

Paramètres d'url
- « client_id »
- « redirect_uri »
- « scope »
  - Google « https://www.googleapis.com/auth/plus.me https://www.googleapis.com/auth/userinfo.profile https://www.googleapis.com/auth/userinfo.email »
  - Facebook « public_profile,email »
- « response_type=code »

Code
Redirection du service vers la « redirect_uri » fournie
avec **paramètre d'url** le **code**

2 TOKEN

Demande de **token**
POST
- https://accounts.google.com/o/oauth2/token
- https://graph.facebook.com/oauth/access_token

En **Header**
- « Content-type: application/x-www-form-urlencoded »

En content (chaine « client_id=123&client_secret=abc… »)
- « client_id »
- « client_secret »
- « redirect_uri »
- « grant_type=authorization_code »
- code (le code que l'on vient de récupérer)

Token
JSON (google) ou chaine (Facebook) avec
- « access_token »
- « exprires_in » (Google) ou « expires » (Facebook)
- « refresh_token » (optionnel)

3 PROFIL

Demande de profil social
GET
- https://www.googleapis.com/plus/v1/people/me
- https://graph.facebook.com/me

En paramètre d'url l' « access_token »
Pour Facebook indiquer en plus les champs à récupèrer « &fields=id,name,email »
Ou POST
En content
- **« Authorization : Bearer » suivi de l' « access_token »**

Profil
JSON, les noms des champs varient selon le service :
Google : « sub,email,name,… »
Facebook (doc) : « id, emails[0]->value, displayName, …»

Application SPA (client JavaScript, serveur PHP | ASP | NODE)

1
CODE

Demande de **token (popup ou redirection)**

GET
- https://accounts.google.com/o/oauth2/auth
- https://www.facebook.com/dialog/oauth

Paramètres d'url
- « client_id »
- « redirect_uri »
- « scope »
  - Google « https://www.googleapis.com/auth/plus.me https://www.googleapis.com/auth/userinfo.profile https://www.googleapis.com/auth/userinfo.email »
  - Facebook « public_profile,email »
- « response_type=token »

Token

Redirection du service vers la « redirect_uri » (cela peut être une page « authcomplete.html » dans laquelle on extrait le token) avec **paramètre d'url** le **token**
- « access_token »
- « exprires_in » (Google) ou « expires » (Facebook)

2
PROFIL

Demande de profil social

GET
- https://www.googleapis.com/plus/v1/people/me
- https://graph.facebook.com/me

En paramètre d'url l' « access_token »
Pour Facebook indiquer en plus les champs à récupérer « &fields=id,name,email »
Ou POST
En content
- **« Authorization : Bearer » suivi de l' « access_token »**

Profil

JSON, les noms des champs varient selon le service :
Google : « sub,email,name,… »
Facebook (doc) : « id, emails[0]->value, displayName, …»

Enregistrer / connecter les utilisateurs

Une fois le profil « social » de l'utilisateur récupéré. Il faut :

- Vérifier dans la base de données du site si un utilisateur avec ces informations est déjà enregistré
- Si ce n'est pas le cas, l'enregistrer
- Connecter l'utilisateur et conserver les informations de l'utilisateur et de token. Pour cela plusieurs stratégies de stockage possibles :
    - Session
    - Base de données
    - localStorage / sessionStorage (pour applications SPA)
    - etc.

Cors

Headers à ajouter côté serveur pour les origines JavaScript

```
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: *");
header("Access-Control-Allow-Headers: *");
```

On peut filtrer plus finement sur les uri d'origines, méthodes, etc.