Document

# Client

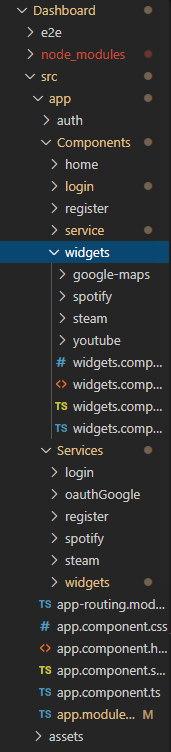
Technologie : AngularJS

Port : 4200

Méthode de communication avec le back : Routes / API rest

Méthode d’authentification : OAuth0

Architecture :

Dashboard -> Racine du projet

Node\_modules -> dependences avec toutes les librairies

App -> Dossier avec notre application

Components -> Dossier avec l’enssemble de nos services / widgets

Services -> Fonctions membres des composents

Assets -> images, gif,…

Les widgets avec authentication :

1. Service RSS :
   1. Récupérer une liste d’articles
   2. Pouvoir cliquer dessus et être redirigé vers la page de l’article
2. Youtube:
   1. Affiche la dernière vidéo de Nota Bene
   2. Affiche ces statistiques et des commentaires
3. Spotify :
   1. Affiche les tracks d’un album
   2. Jouer une track
   3. S’abonner à l’artiste

Les widgets sans authentication :

1. Widget Météo
2. Widget pense-bête
3. Widget google map
4. Widget youtube

Le Timer :

Toutes les X minutes, choisi par l’utilisateur, le client, les widgets se mettent à jour.

# Serveur

Technologie : NodeJS

Port : 8080

Méthode de communication avec le front : Routes / API rest

Méthode authentification : Token ( username / password )

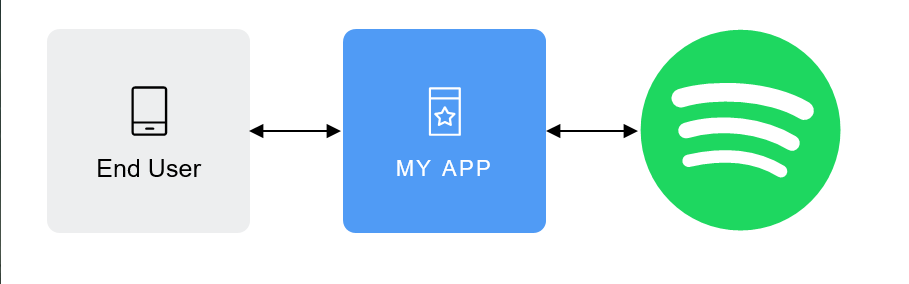
Liste des routes :

* ***Route :*** 
  + ***Methode*** 
    - ***Paramètre***
  + ***Réponse renvoyé***
* about.json :
  + GET
* Login :
  + POST :
    - Username
    - Password
    - Réponse :
      * token ( sauvegardé en cookie )
      * Erreur 400 ( login invalide )
  + DELETE /account :
    - Token
    - Réponse :
      * 300, OK
      * 400, NOK
* Logout :
  + POST :
    - Token
    - Réponse :
      * 300, OK
      * 400, NOK
* Register :
  + POST :
    - Username
    - Password
    - Réponse :
      * 300 , utilisateur créé
      * 400 , utilisateur déjà existant
* Services & Widget :
  + POST :
    - Id de service
    - Paramètres optionnels
    - Réponse :
      * 300, OK + id de widget
      * 400, NOK
  + GET :
    - Token
    - Réponses :
    - Widgets avec leurs id ect…
  + UPDATE :
    - Token
    - Id\_widget
    - Paramètres
    - Réponse :
      * 300, OK
      * 400, NOK
  + DELETE :
    - Token
    - Id\_widget
    - Réponse :
      * 300, OK
      * 400, NOK

Architecture du projet :

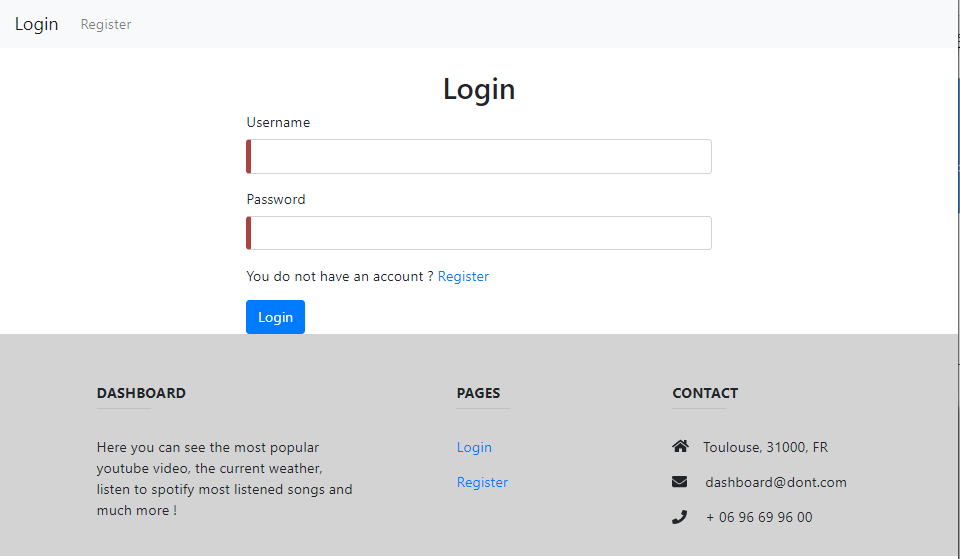
* Class Main
* Class Account
* Class Widgets

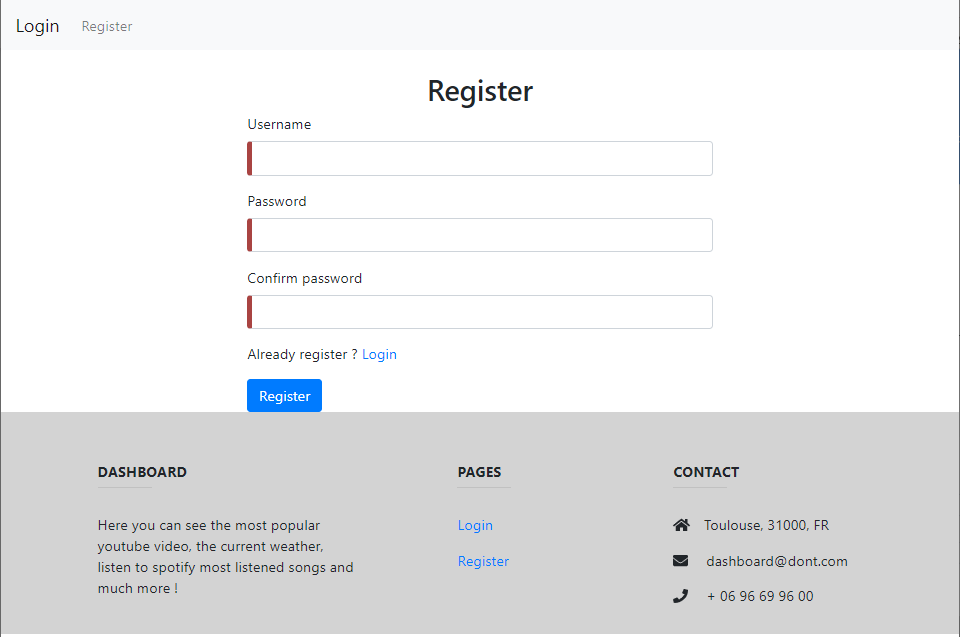
## Oauth2.0 avec Spotify :



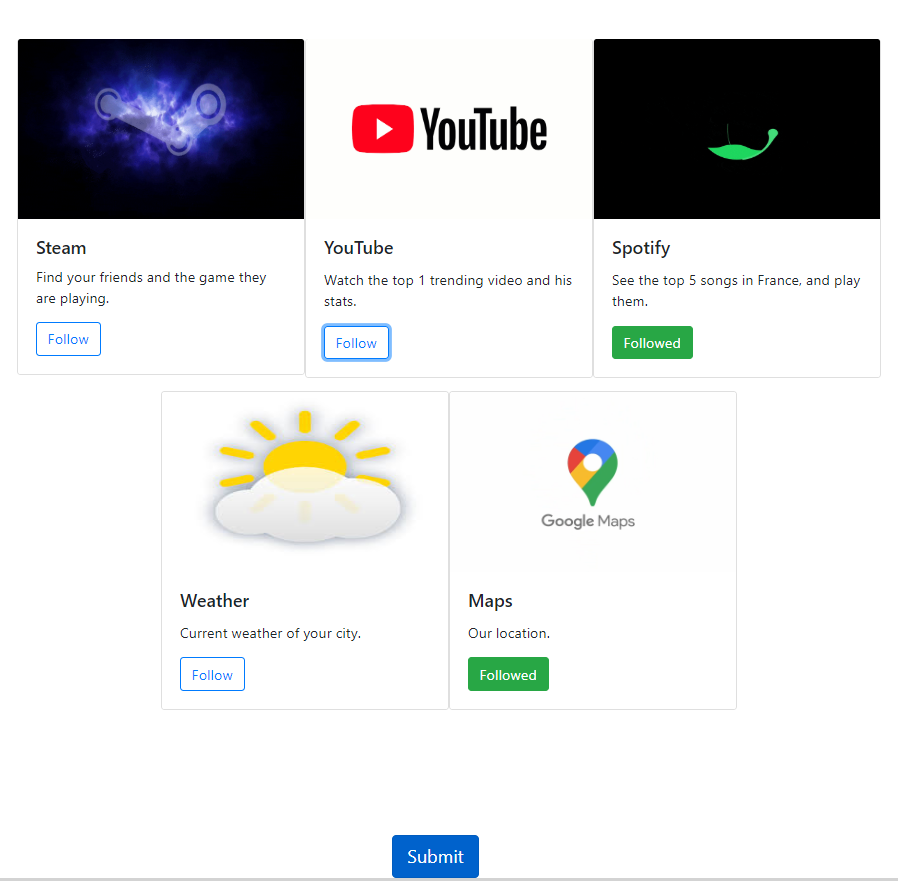
# MOCKS-UP

Page Login/Register :



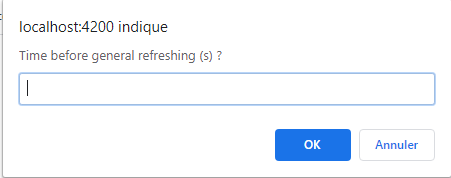


Une fois le client Log, celui-ci est redirigé vers la pages des services :

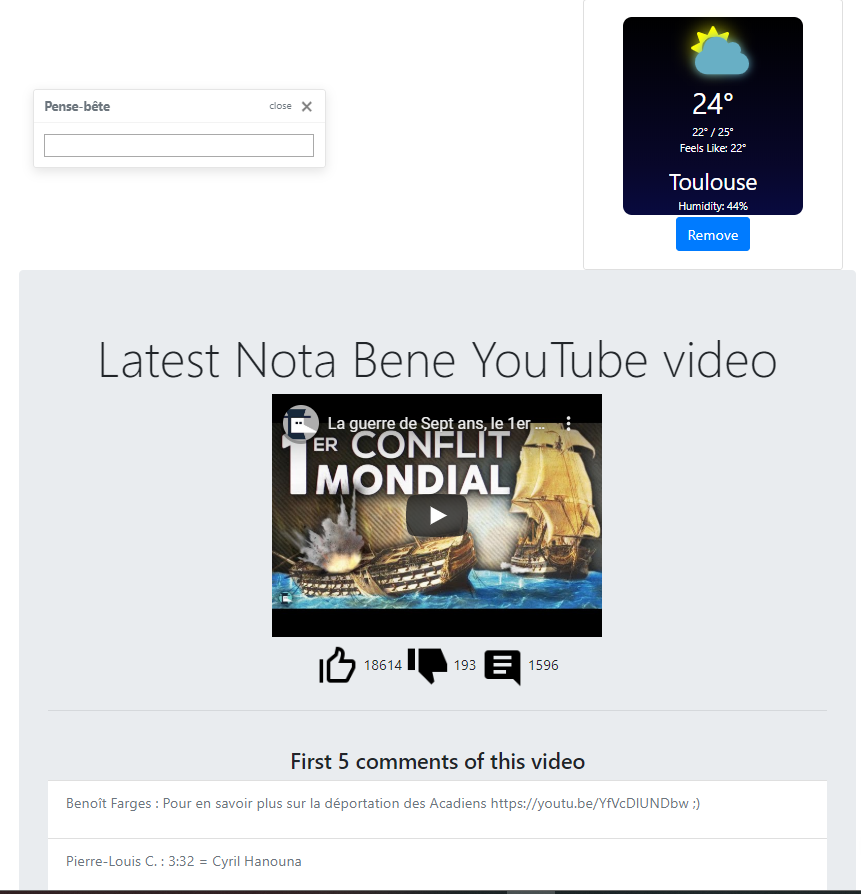


Une fois les services choisis, le client les sauvegardes et est redirigé vers la page Home :

* Avant de pouvoir accéder au dashboard, il doit saisir un temps de rafraîchissement :



* Et peut ensuite voir les widgets :



S’il le souhaite il peux accéder aux services unique tel que Youtube par exemple, où il ne verra que ce services et non les autres  :

