

Septembre 2023

# Documentation de Projet

Jeu mogettes

Clients : Jean-Pierre et Robin Guedon

Développeur : Tanguy Serrand

# Introduction

Bienvenue dans la documentation de ce projet. Cette ressource vise à vous fournir toutes les informations nécessaires pour comprendre, installer, configurer et utiliser ce projet avec succès.

## Objectif de la Documentation

Cette documentation a été créée dans le but de :

**Faciliter l'Installation :** Vous guider à travers le processus d'installation de ce projet, quel que soit votre environnement.

**Fournir des Instructions d'Utilisation :** Vous montrer comment utiliser ce projet de manière efficace

**Expliquer la Configuration :** Détaillez les différentes options de configuration disponibles pour personnaliser ce projet selon vos besoins.

## Public Cible

Cette documentation s'adresse principalement aux :

**Développeurs :** Ceux qui souhaitent contribuer au développement, à l'amélioration ou à la personnalisation de ce projet.

**Autres contributeurs :** Toute personne qui prend part au projet

## Organisation de la Documentation

La documentation est organisée en sections thématiques pour faciliter la navigation. Vous pouvez vous référer à la table des matières pour trouver rapidement les informations dont vous avez besoin.

<b>Introduction.....</b>	<b>2</b>
Objectif de la Documentation.....	2
Public Cible.....	2
Organisation de la Documentation.....	2
<b>Backend.....</b>	<b>4</b>
À propos du Backend.....	4
Prérequis.....	4
Configuration et lancement.....	4
Installation des Dépendances.....	4
Lancement.....	4
Configuration de la Base de Données.....	4
Utilisation de l'API.....	5
Gestion des Utilisateurs.....	5
Gestion des Départements.....	5
Gestion des Régions.....	6
Gestion des Énigmes.....	6
Gestion des Niveaux.....	6
Niveaux Mathématiques (Math Levels).....	6
Niveaux en Informatique (CS Levels).....	6
Niveaux Régionaux (Région Levels).....	6
<b>Frontend.....</b>	<b>8</b>
Présentation Générale.....	8
Installation.....	8
Structure du Projet.....	9
Authentification.....	9
Connexion (Login).....	9
Inscription (Register).....	10
Fonctionnalités Principales.....	11
Carte (Map).....	11
Départements.....	11
Dialogues des départements (departments dialog).....	12

# Backend

## À propos du Backend

Le backend de cette application est construit avec Flask, un framework web minimaliste pour Python. Il offre une API pour gérer les données de l'application, y compris les utilisateurs, les niveaux, les énigmes, les départements et les régions.

## Prérequis

Avant de commencer à utiliser ce backend, assurez-vous d'avoir les éléments suivants :

Python3  
SQLite3 ou un autre système de gestion de base de données (si nécessaire)  
pip (outil de gestion des packages Python)  
Une compréhension de base de Flask et des APIs REST.

## Configuration et lancement

### Installation des Dépendances

Pour installer les dépendances requises, exécutez la commande suivante :

```
pip install -r requirements.txt
```

### Lancement

```
bash run_serv_debug.sh
```

## Configuration de la Base de Données

Le backend utilise SQLite par défaut. Le schéma de la base de données est défini dans le fichier schema.sql. Vous pouvez personnaliser la configuration de la base de données en modifiant DATABASE dans le fichier server.py.

Ce schéma peut également être modifier, dans ce cas il faut utiliser :

```
bash init_db.sh
```

Ce script initialise le serveur avec le schéma modifié.

## Utilisation de l'API

Cette section détaille l'utilisation de chaque endpoint de l'API. Assurez-vous de respecter les méthodes HTTP appropriées (GET, POST, DELETE, etc.) pour chaque endpoint. On peut évidemment l'utiliser à partir du front mais aussi via un outil tiers tel que postman.

### Gestion des Utilisateurs

GET /users: Récupérer la liste de tous les utilisateurs.

POST /users: Ajouter un nouvel utilisateur.

DELETE /users/<user\_id>: Supprimer un utilisateur par ID.

### Gestion des Départements

Les départements sont des entités géographiques importantes de cette application. Vous pouvez gérer les départements via l'API comme suit :

POST /departments: Ajouter une liste de nouveaux départements en spécifiant les détails nécessaires dans un objet JSON comme le suivant (voir figure 1).

```
[
  {
    "id": 44,
    "name": "Loire-Atlantique",
    "coordinates": "m 128.15408,...,-1.06 -0.70,-1.21 z",
    "region": "Pays de la Loire",
    "zones": [
      {
        "x": 345,
        "y": 970,
        "radius": 254,
        "type": "m"
      }, {
        "x": 1009,
        "y": 970,
        "radius": 254,
        "type": "pdl"
      }
    ]
  }
]
```

*figure 1 - Exemple de département sous JSON (voir \_departments.json)*

GET /departments: Récupérer la liste de tous les départements enregistrés.  
GET /departments-by-region?region=<nom\_de\_la\_région>: Récupérer la liste des départements d'une région spécifique. Assurez-vous de spécifier le nom de la région dans la requête.

## Gestion des Régions

Les régions représentent des zones géographiques plus larges dans cette application. Vous pouvez les gérer via l'API de la manière suivante :

POST /regions: Ajouter une liste de nouvelles régions en fournissant les informations nécessaires sous forme d'objet JSON.  
GET /regions: Récupérer la liste de toutes les régions enregistrées.

## Gestion des Énigmes

Les énigmes sont un élément clé de votre application. Vous pouvez les gérer via l'API comme suit :

POST /enigmes: Ajouter de nouvelles énigmes.  
GET /enigmes/<id>: Récupérer une énigme par son ID.

## Gestion des Niveaux

La gestion des niveaux est un élément crucial de votre application, car elle permet aux utilisateurs de progresser et de relever des défis.

### Niveaux Mathématiques (Math Levels)

POST /users/<user\_id>/mlevels: Ajouter un nouveau niveau mathématique réussi à un utilisateur. Cela augmente le score mathématique de l'utilisateur.  
GET /users/<user\_id>/mathscore: Récupérer le score mathématique actuel d'un utilisateur.

### Niveaux en Informatique (CS Levels)

POST /users/<user\_id>/cslevels: Ajouter un nouveau niveau en informatique réussi à un utilisateur. Cela augmente le score en informatique de l'utilisateur.  
GET /users/<user\_id>/csscore: Récupérer le score en informatique actuel d'un utilisateur.

### Niveaux Régionaux (Region Levels)

POST /users/<user\_id>/rlevels: Ajouter un nouveau niveau régional réussi à un utilisateur. Cela lui permet de progresser dans la découverte de votre application.

GET /users/<user\_id>/regionprogress: Récupérer la progression régionale actuelle d'un utilisateur.

# Frontend

## Présentation Générale

Ce document présente une documentation complète du Frontend de l'application. Le Frontend est développé en utilisant Angular, un framework JavaScript populaire pour la création d'interfaces utilisateur modernes et réactives. Il s'agit de la partie de l'application avec laquelle les utilisateurs interagissent directement.

## Installation

Avant de commencer à travailler sur le Frontend, assurez-vous d'avoir Node.js et npm (Node Package Manager) installés sur votre système. Si ce n'est pas déjà le cas, vous pouvez les télécharger et les installer à partir du site officiel de Node.js : <https://nodejs.org/>.

Une fois que Node.js et npm sont installés, suivez ces étapes pour installer et exécuter le Frontend de l'application :

Accès au Répertoire Frontend : Naviguez vers le répertoire du Frontend.

```
cd game
```

Installation des Dépendances : Utilisez npm pour installer toutes les dépendances nécessaires répertoriées dans le fichier package.json.

```
npm install
```

Exécution de l'Application : Une fois les dépendances installées et la configuration terminée, vous pouvez exécuter l'application en utilisant la commande suivante :

```
ng serve
```

L'application sera disponible à l'adresse <http://localhost:4200/> par défaut. Vous pouvez y accéder depuis votre navigateur web.

Développement Local : Pour le développement, Angular propose un serveur de développement avec rechargement automatique. À chaque modification de code, l'application sera rechargée automatiquement dans le navigateur.

Production : Pour créer une version de production de l'application, utilisez la commande suivante :



ng build --prod

Cela générera les fichiers de production dans le répertoire dist/.

## Structure du Projet

Le Frontend de l'application est organisé en utilisant une structure de projet Angular standard. Voici un aperçu de la structure des fichiers et répertoires les plus importants :

**src/app**: Le répertoire principal de l'application, contenant tous les composants, modules et services.

**\_models**: Ce répertoire contient les modèles de données utilisés dans l'application.

**\_services**: Les services Angular pour la communication avec le Backend et d'autres fonctionnalités.

**auth**: Module pour la gestion de l'authentification et des utilisateurs.

**login et register**: Composants pour la connexion et l'inscription des utilisateurs.

**auth-routing.module.ts** : Configuration des itinéraires liés à l'authentification.

**auth.module.ts**: Module principal pour l'authentification.

**game**: Module principal pour la partie jeu de l'application.

**department-dialog**: Composant pour la gestion des départements.

**map**: Composant pour l'affichage de la carte de jeu.

**app-routing.module.ts**: Configuration des itinéraires globaux de l'application.

**app.module.ts**: Module principal de l'application.

**src/assets**: Ce répertoire contient les fichiers statiques tels que les images

Cette structure permet une organisation claire du code de l'application, ce qui facilite le développement, la maintenance et l'extension de l'application.

## Authentification

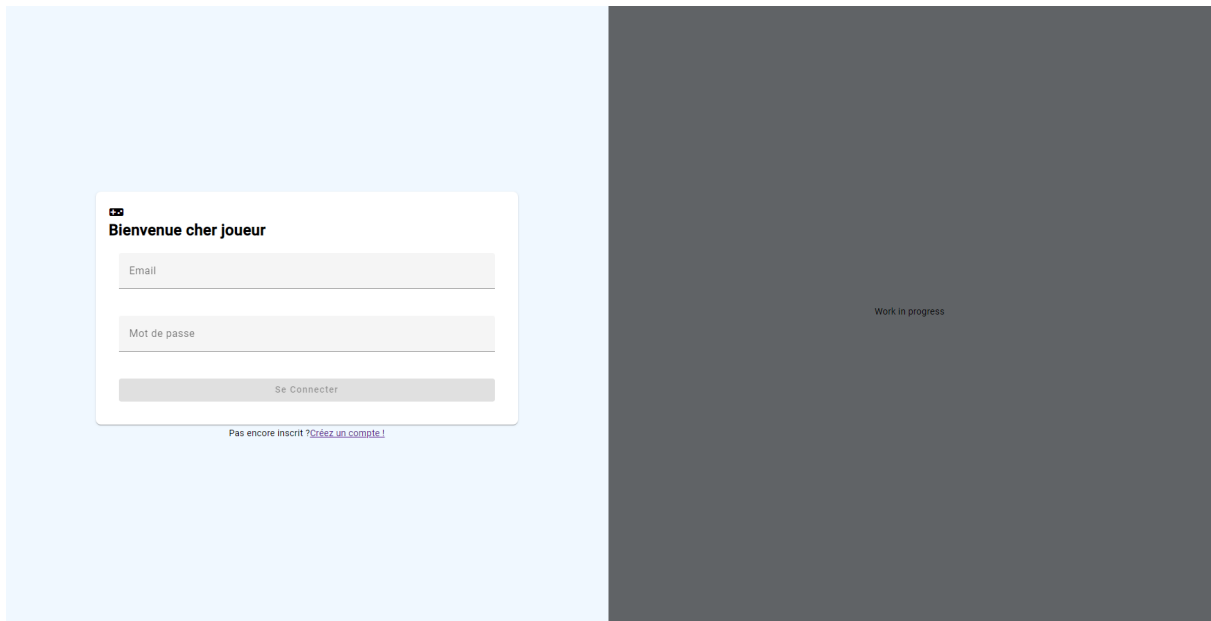
### Connexion (Login)

L'authentification est une partie essentielle de notre application, permettant aux utilisateurs de se connecter pour accéder à leurs fonctionnalités (voir figure 2). Pour vous connecter, suivez ces étapes :

Accédez à la page de connexion en allant à /login dans la barre de navigation.

Saisissez votre adresse e-mail et votre mot de passe.

Cliquez sur "Se connecter" pour accéder à votre compte.



*figure 2 - Écran de connexion*

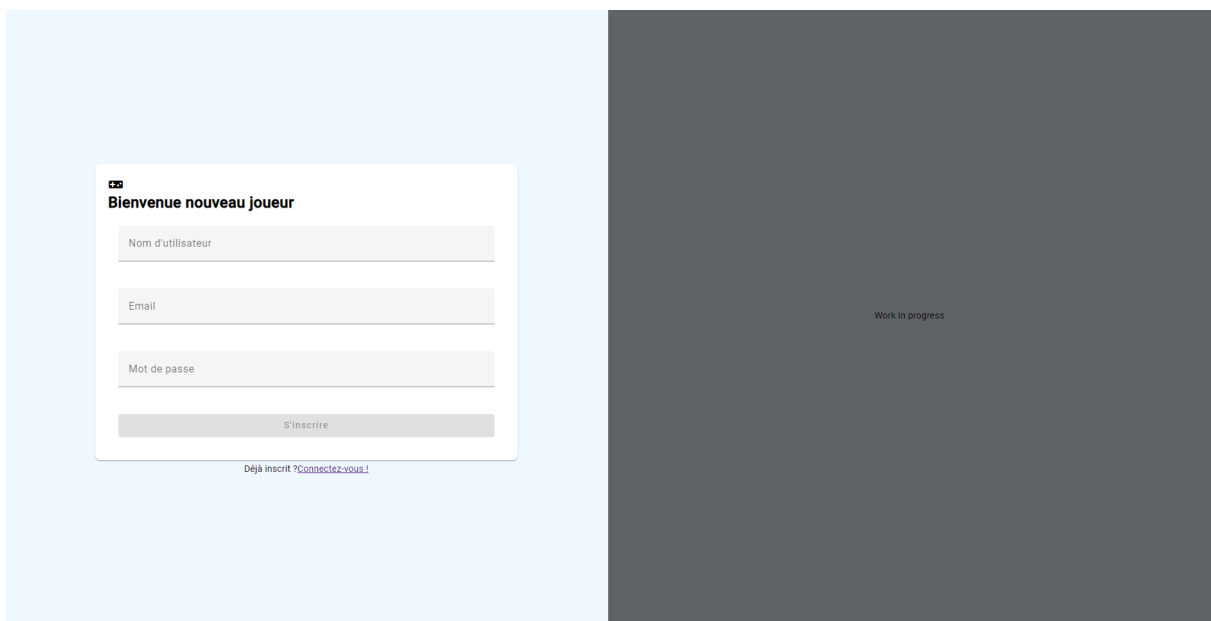
## Inscription (Register)

Si vous n'avez pas encore de compte, vous pouvez vous inscrire en suivant ces étapes (voir figure 3) :

Accédez à la page d'inscription en allant à /register dans la barre de navigation.

Remplissez le formulaire d'inscription en fournissant votre nom d'utilisateur, votre adresse e-mail et un mot de passe.

Cliquez sur "S'inscrire" pour créer un compte.



*figure 3 - Écran d'inscription*

## Fonctionnalités Principales

### Carte (Map)

La carte est la fonctionnalité centrale de notre application. Elle permet aux utilisateurs de découvrir et d'explorer différents départements. Vous pouvez interagir avec la carte en cliquant sur les régions pour accéder aux départements (voir figure 4).



*figure 4 - Carte principale*

### Départements

Lorsque vous cliquez sur un département sur la carte (voir figure 5), une illustration s'affiche avec plusieurs boutons pour accéder aux énigmes.

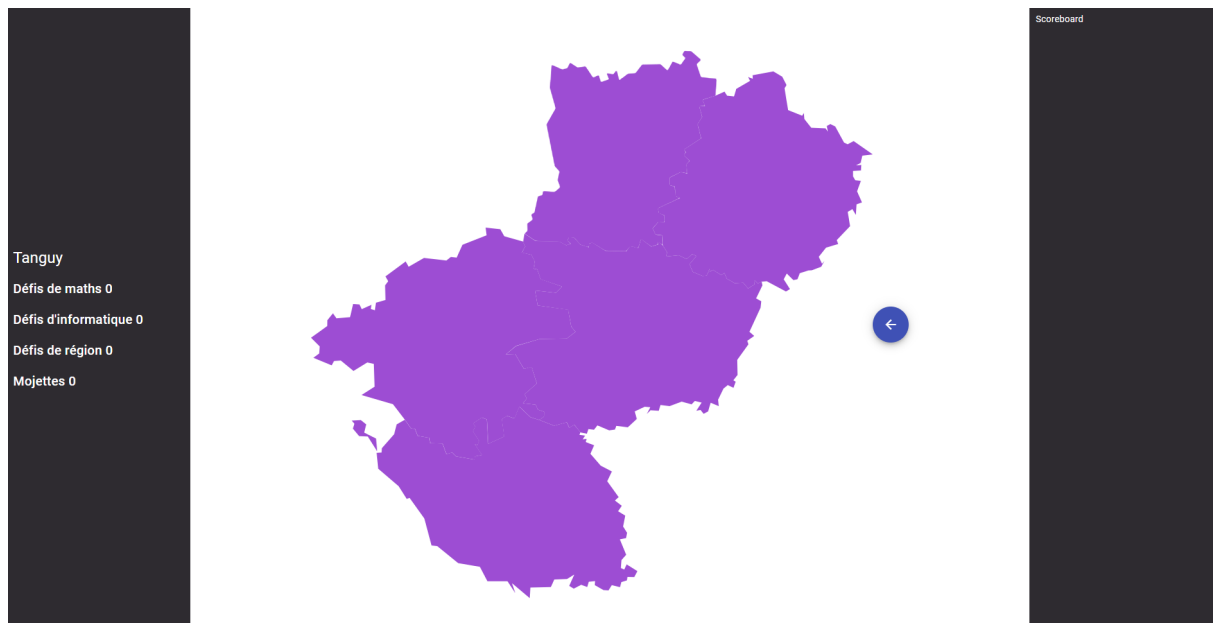


figure 5 - Carte des départements

## Dialogues des départements (departments dialog)

Au clic sur un de ces boutons, un pop-up (fenêtre de dialogue) s'affiche avec le contenu de l'énigme. L'utilisateur peut alors y répondre et envoyer sa réponse.



figure 6 - Illustration d'un département

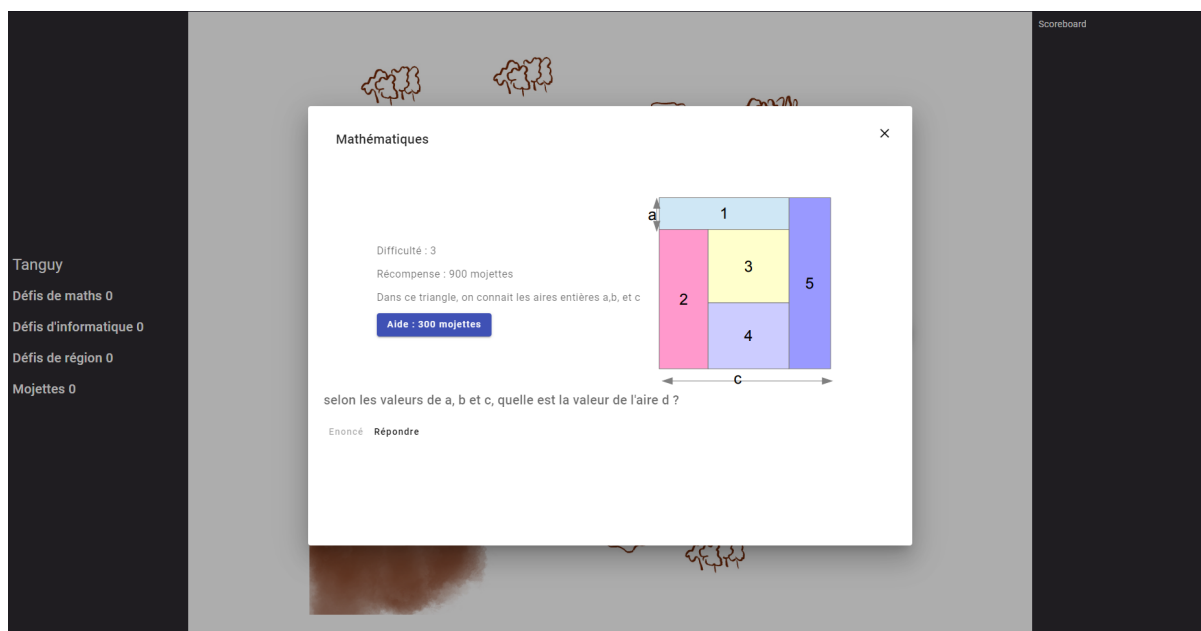


figure 7 - Fenêtre de dialogue d'un département