04/11/2018

Timothé Coger

Projet test Pokemon

Table des matières

[Pré requis 2](#_Toc529131612)

[Problèmes connus 2](#_Toc529131613)

[Structure du projet 2](#_Toc529131614)

[Solution BackEnd 3](#_Toc529131615)

[Solution FrontEnd 4](#_Toc529131616)

[Démarrage du projet 5](#_Toc529131617)

[Démarrage du BackEnd 5](#_Toc529131618)

[Démarrage du FrontEnd 8](#_Toc529131619)

[Utilisation de l’interface 10](#_Toc529131620)

[Saisie d’un nom de Personnage 10](#_Toc529131621)

[Saisie d’une lettre ou d’un nom partiel 11](#_Toc529131622)

[Utilisation de la liste « Top Ranking » 12](#_Toc529131623)

[Utilisation de la fonction de Vote 13](#_Toc529131624)

# Pré requis

La liste des pré requis est la suivante :

* Visual Studio 2017 (Community utilisé, installé avec options par défaut)
* SDK Framework .NET 4.7.1/4.7.2
* Présence des dossiers packages au niveau des projets (pour les dépendances NuGet)

# Problèmes connus

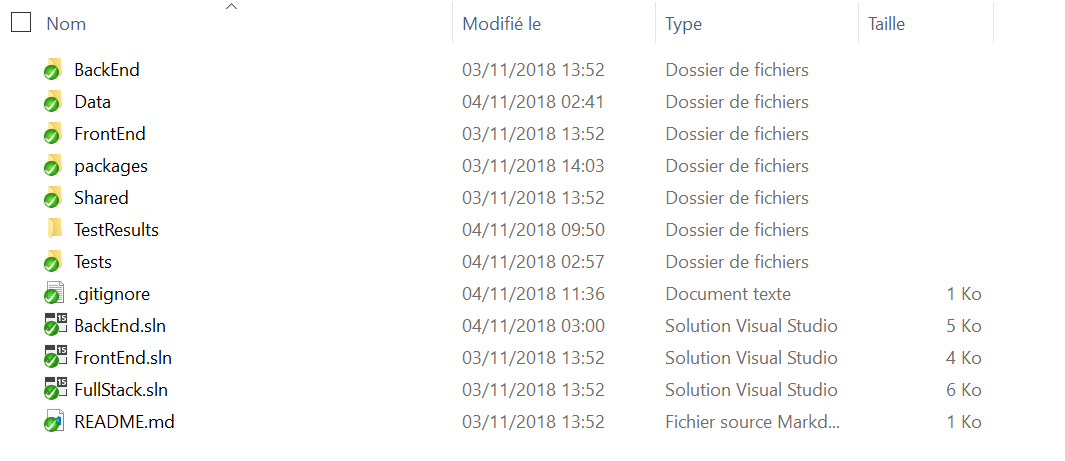
La connexion pour le mapping entre EF6 et la base SQLite ne fonctionne pas pour le moment. Le clone du projet devra se faire si possible sur C:\DEV\ pour un chemin d’extraction C:\DEV\test pour la racine.

# Structure du projet

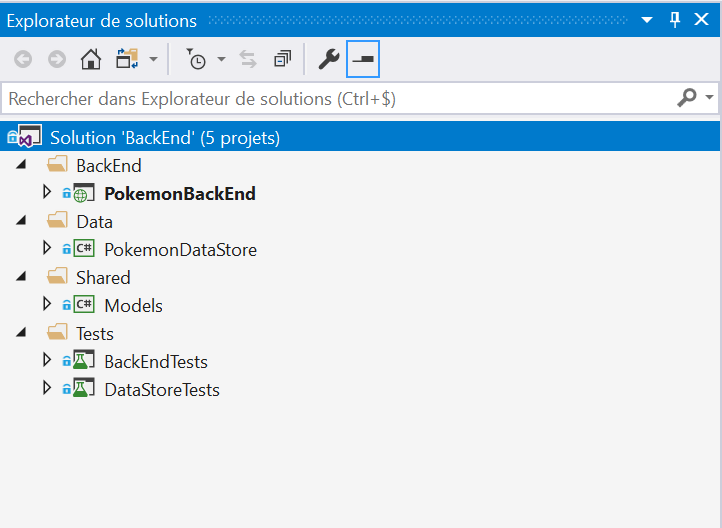
Dans le cas des illustrations, la racine du projet est « C:\DEV\test\ ».

Le projet comprend 3 solutions :

* Une solution FullStack pour avoir une vision globale des projets
* Une solution BackEnd pour lancer la partie BackEnd
* Une solution FrontEnd pour lancer la partie FrontEnd



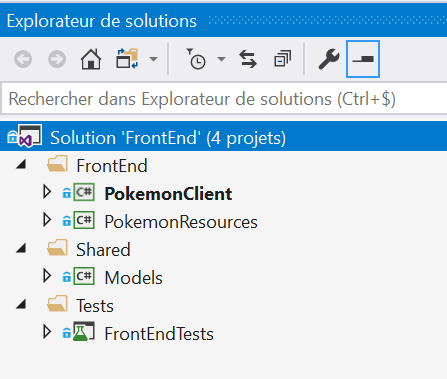
## Solution BackEnd



La solution BackEnd contient 5 projets :

* Le projet PokemonBackEnd qui est le projet de démarrage de l’application
* Le projet PokemonDataStore qui contient le modèle de données et mapping relationnel (EF + SQLite)
* Le projet Shared qui contient les objets Models utilisés par le FrontEnd également
* Les projets de test du BackEnd et du DataStore.

## Solution FrontEnd



La solution FrontEnd contient 4 projets :

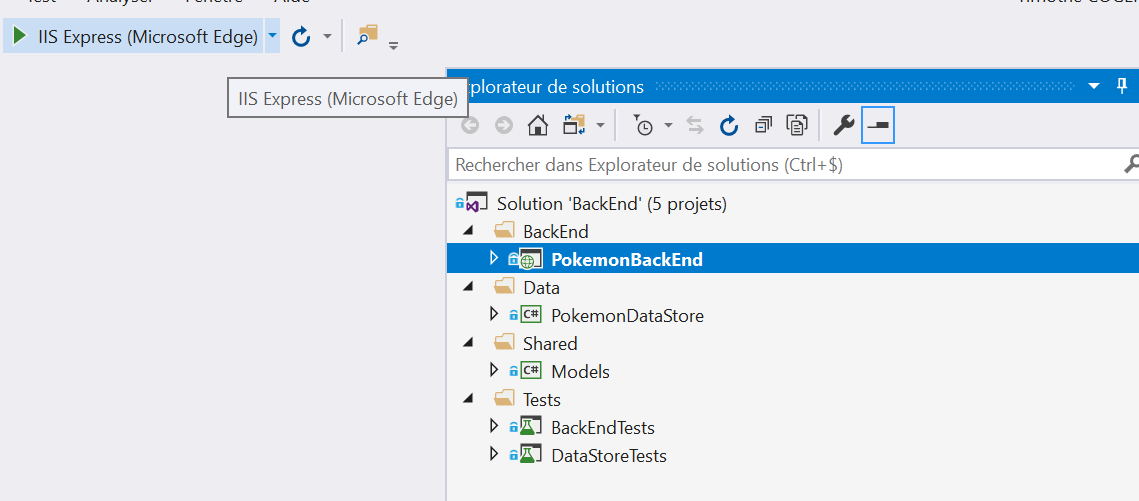
* Le projet de test FrontEndTests
* Le projet partagé avec le BackEnd qui contient les classes Models utilisés par le BackEnd également
* Le projet PokemonResources qui contient les resources pour le projet Client (images, libellisation et internationalisation (non développée))
* Le projet PokemonClient qui contient la logique client et l’interface (WPF + MVVM) et constitue le démarrage de la partie cliente.

# Démarrage du projet

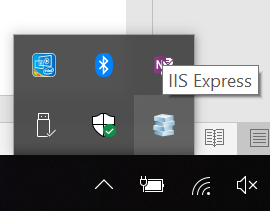
Pour démarrer le projet, il faudra ouvrir les 2 solutions BackEnd et FrontEnd dans 2 instances de Visual Studio 2017 différentes.

## Démarrage du BackEnd

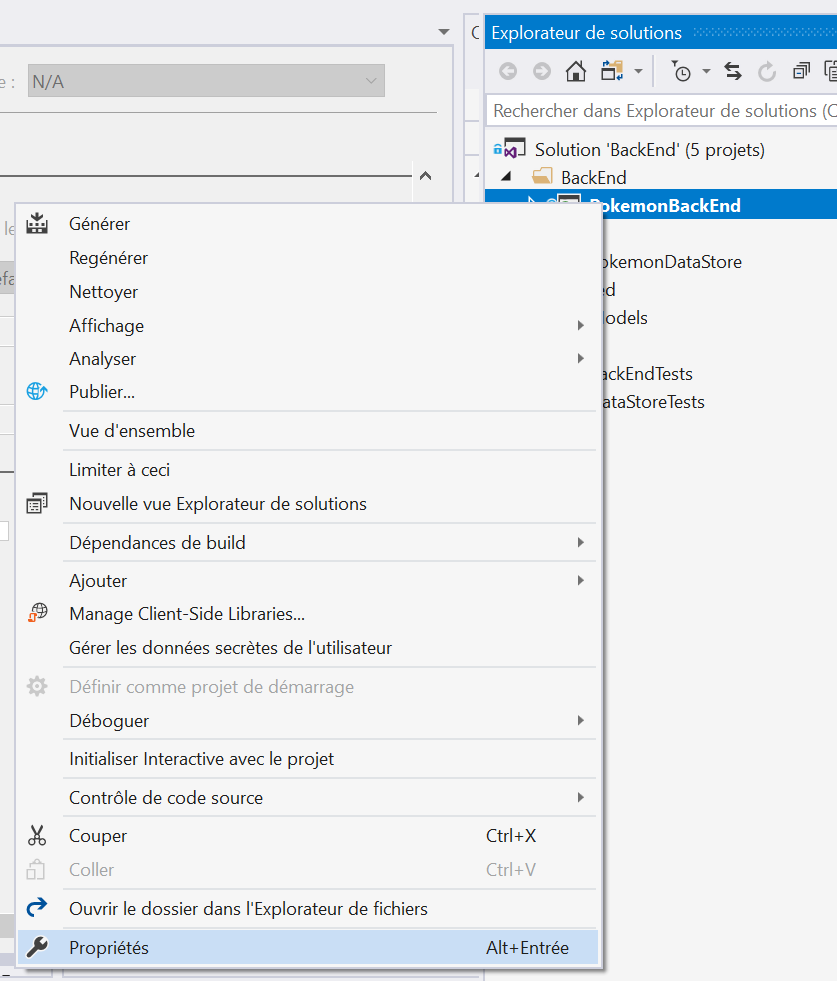
Le projet PokemonBackEnd doit être sélectionné comme projet de démarrage et utiliser le bouton de démarrage du projet dans la première instance de Visual Studio que vous aurez ouvert.



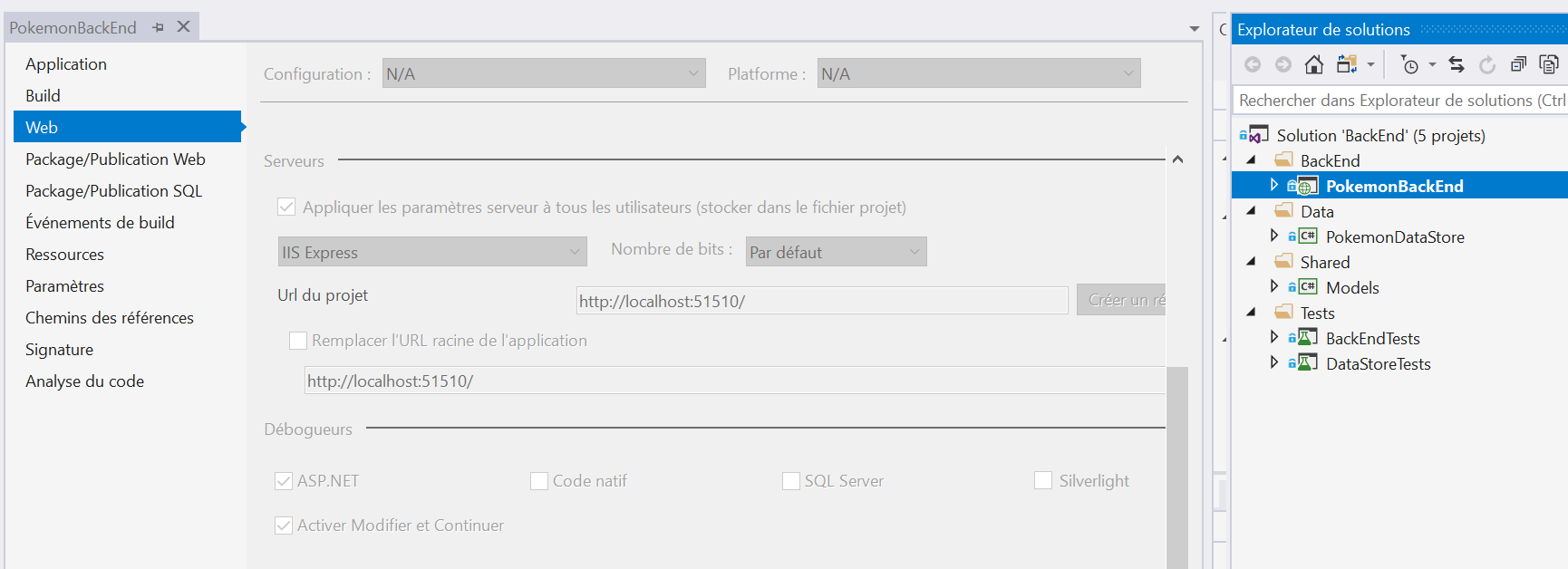
IIS express va se lancer et le navigateur va s’afficher pour une communication sur l’URL localhost :51510



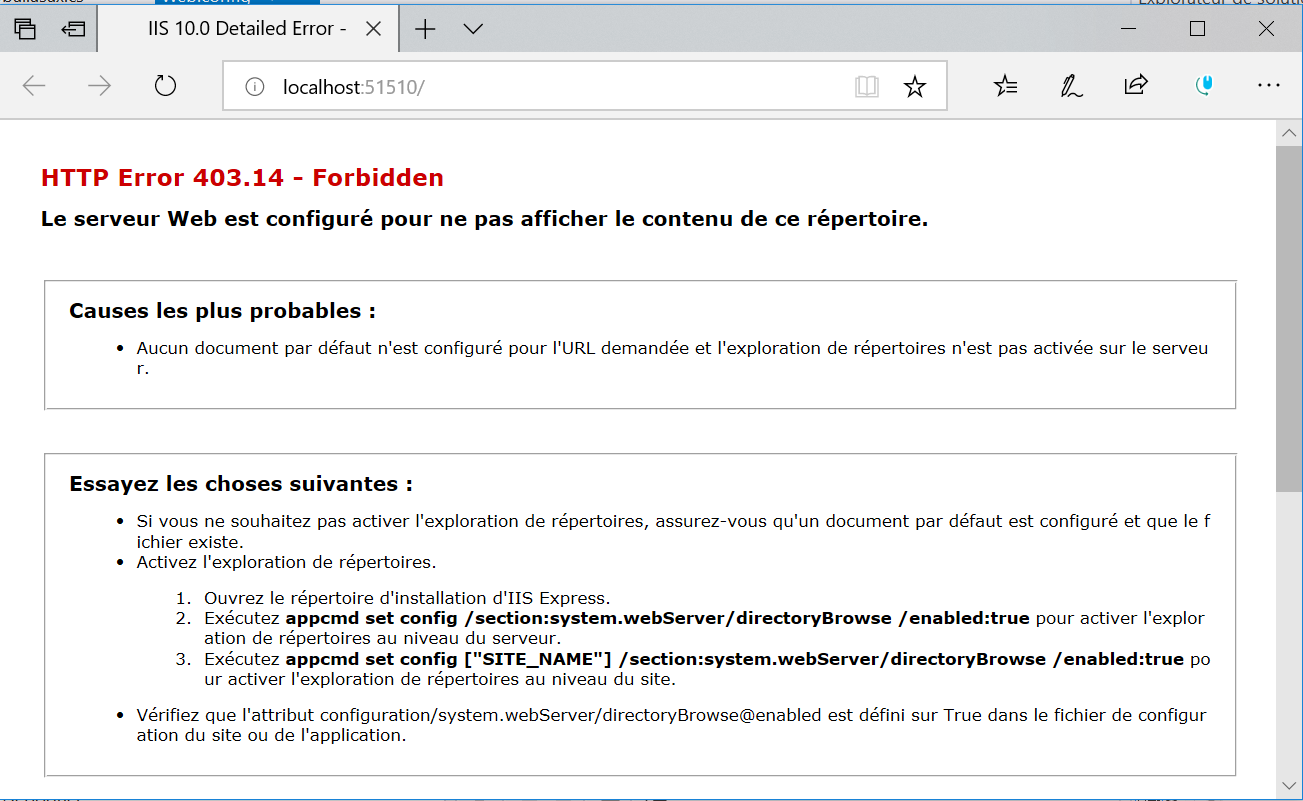
Cette configuration correspond à celle précisée dans les propriétés du projet auxquelles vous pouvez accéder par un clic droit puis l’utilisation du menu « Propriétés ».



La page des propriétés du projet va alors s’afficher.



Le démarrage achevé va s’accompagner d’une erreur « HTTP 403.14 Forbidden ». Ceci est attendu car il n’y a pas d’élément de démarrage par défaut.



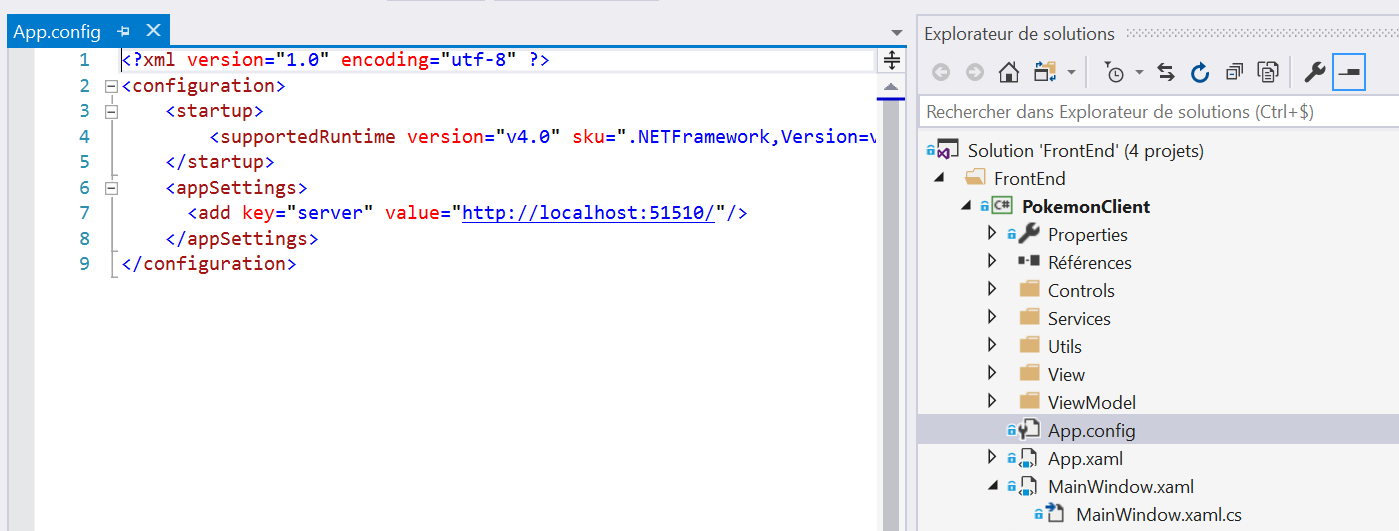
Dès lors, vous pouvez tester un appel d’un endpoint, par exemple : <http://localhost:51510/api/characters/name/Pikachu>

## Démarrage du FrontEnd

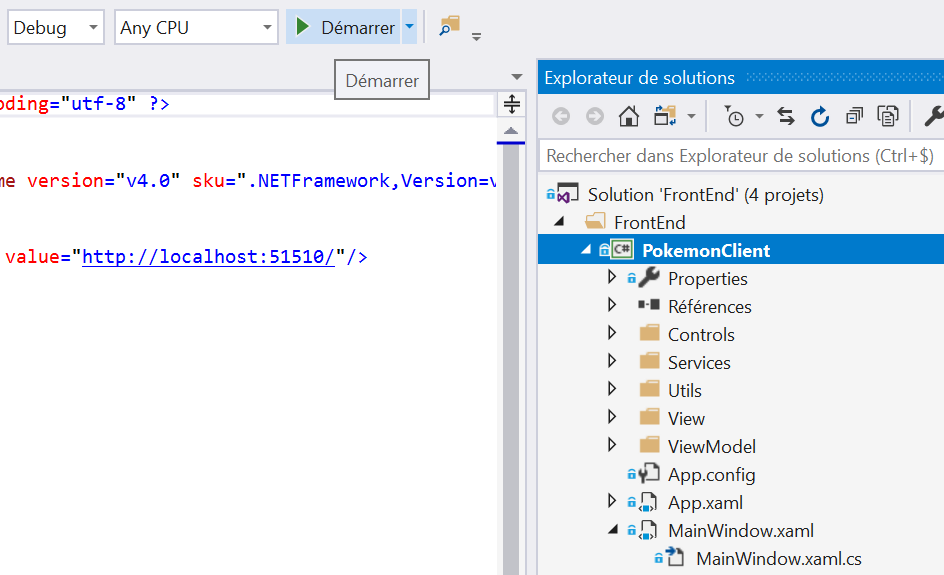
Une fois le backEnd démarré, passez à l’autre instance de Visual Studio 2017 que vous aurez ouvert sur la solution FrontEnd, il faut vérifier que le projet de démarrage du FrontEnd va communiquer avec la bonne chaine de connexion.

Pour cela, il faut vérifier que le fichier App.config contient la bonne valeur.

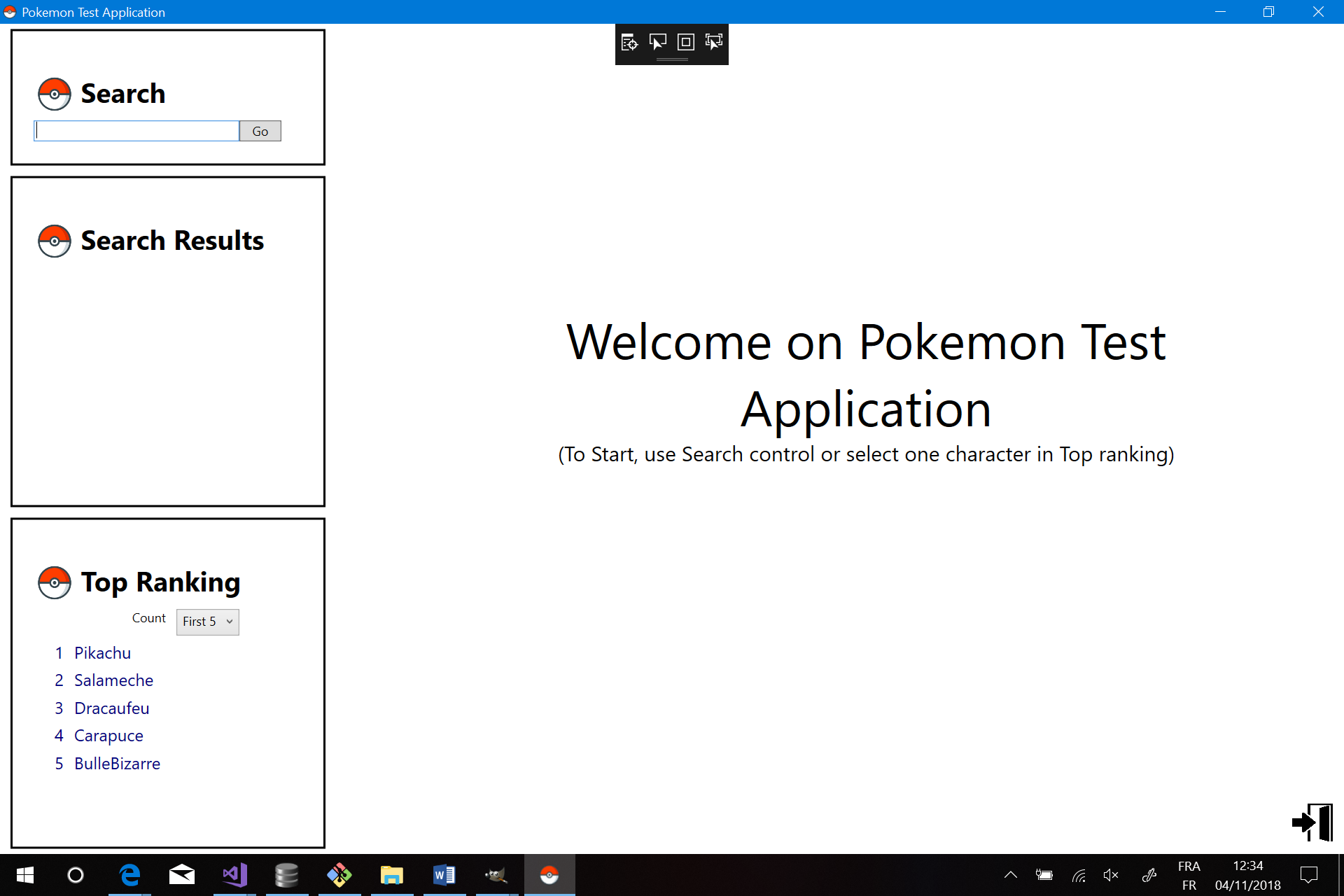
Ici, IIS Express écoute bien sur localhost et le port 51510, la valeur est correcte dans le fichier de configuration.



Vous pouvez ensuite démarrer le projet FrontEnd.



L’écran d’interface doit alors d’afficher.



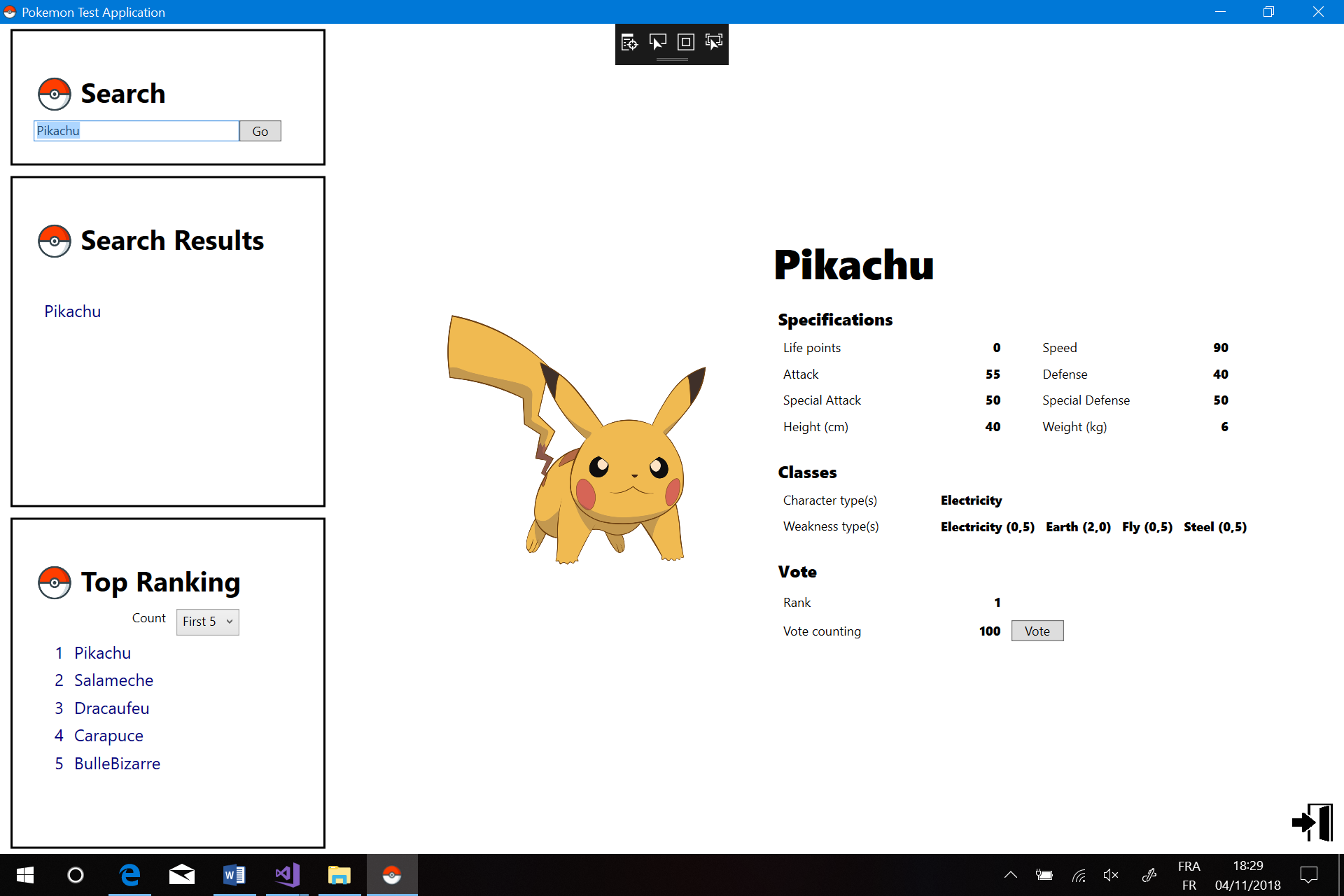
# Utilisation de l’interface

## Saisie d’un nom de Personnage

Saisie d’une chaine de charactère dans la recherche, Zone de saisie teste dans l’encart « Search ».

La validation s’effectue par la touche Entrée ou le bouton « Go » situé à droite de cette zone.

Exemple : Pikachu

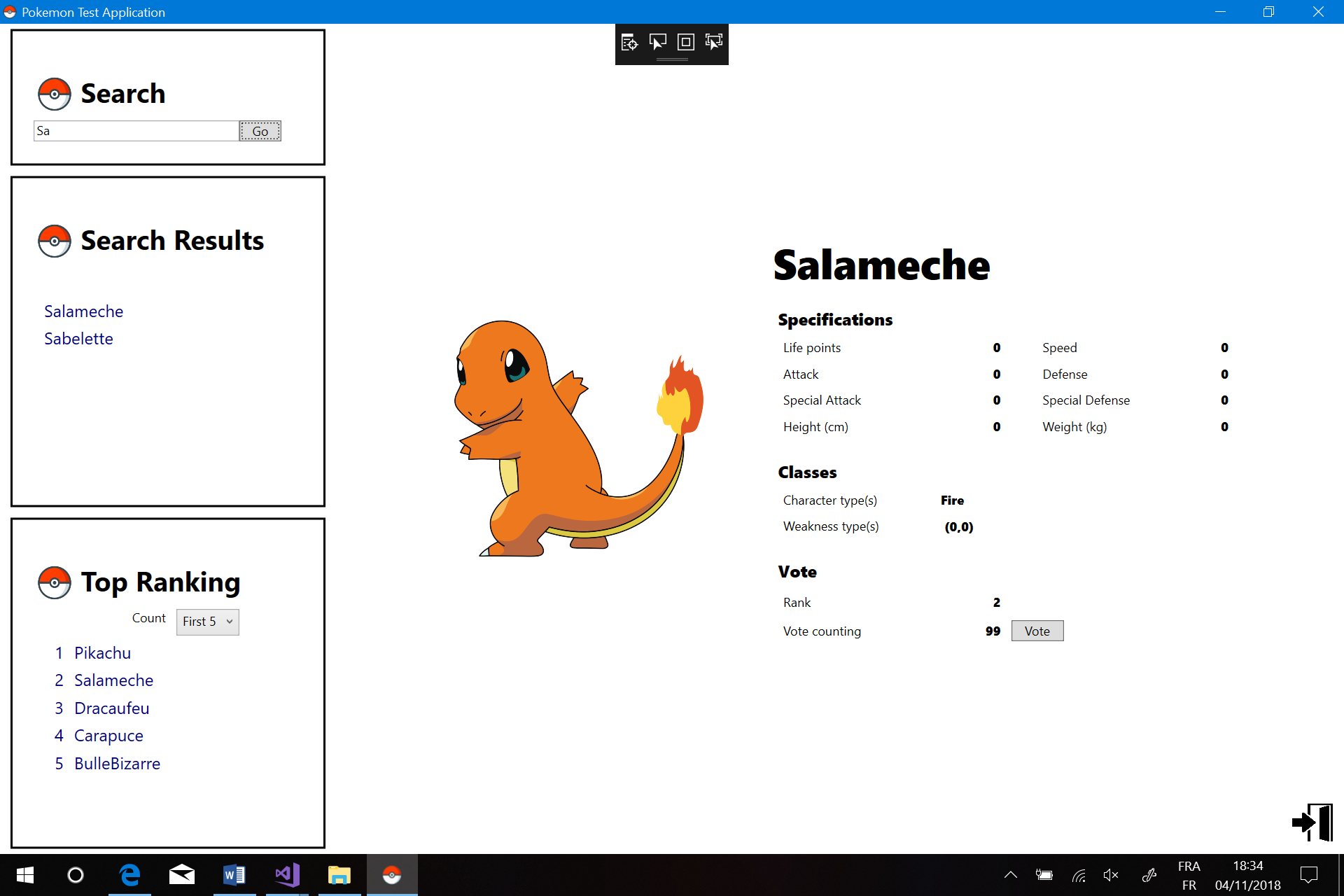


* La liste de résultats va s’afficher dans « Search Results » ;
* Le premier élément trouvé va s’afficher dans le volet de droite

## Saisie d’une lettre ou d’un nom partiel

Le scénario est le même que la saisie du nom complet.

Exemple : Saisie de la chaine de caractère « Sa »



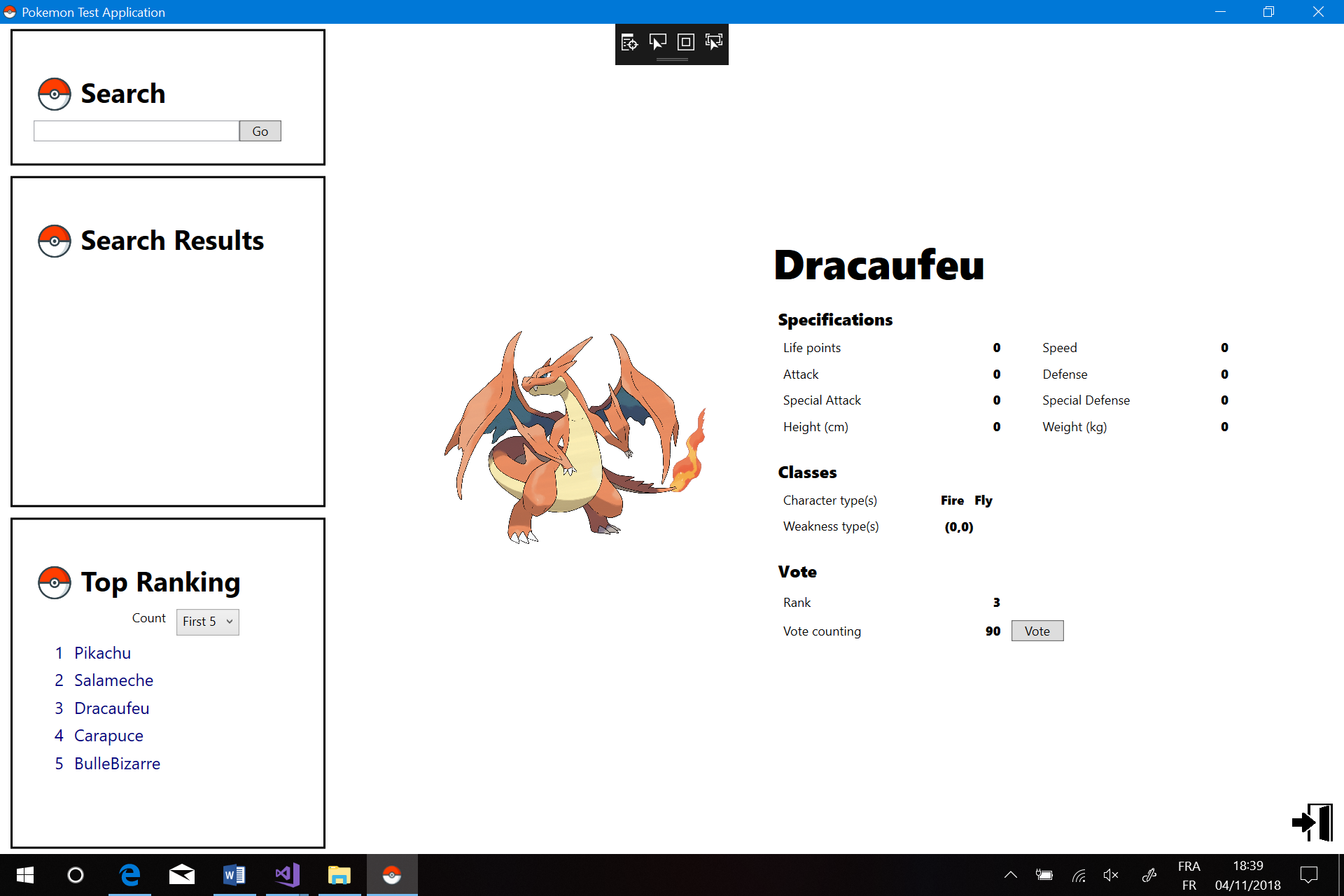
* Dans la liste des résultats s’affiche une liste de personnages contenant la chaine recherchée
* Le premier élément de cette liste s’affiche ;
* L’utilisateur peut sélectionner un autre personnage dans cette liste et l’encart de droite va s’actualiser.

## Utilisation de la liste « Top Ranking »

Cette liste affiche les personnages ordonnancés par leur nombre de votes.

Un choix sur liste permet d’étendre la liste à 5, 10, 20 ou 50 entrées, cette liste est initialisée au départ à 5 noms de personnages.

Exemple de scénario : L’utilisateur clique sur le troisième de la liste « Dracaufeu »

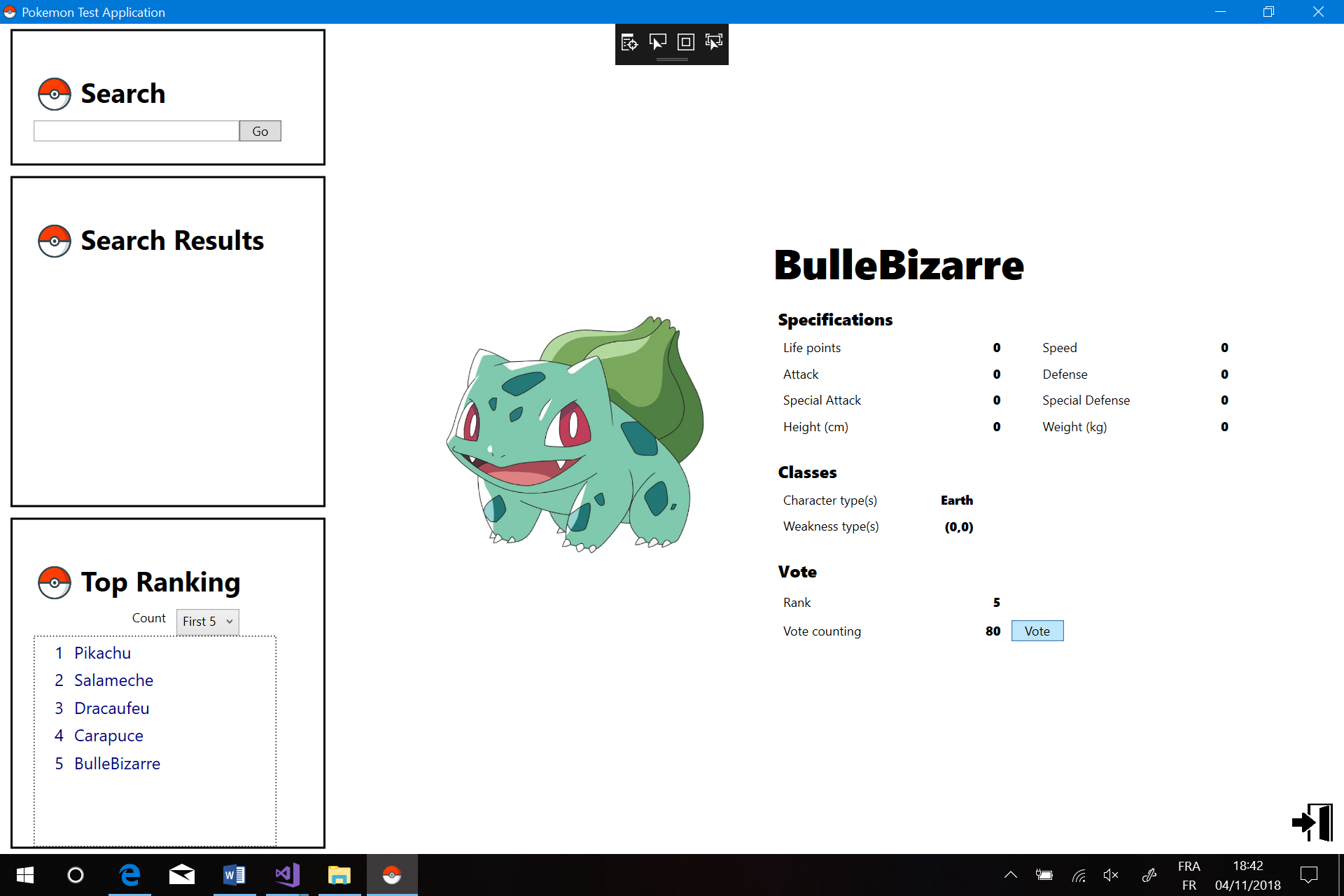


* Le personnage s’affiche à droite.

## Utilisation de la fonction de Vote

L’utilisateur ayant affiché la fiche d’un personnage peut visualiser le compte de votes et voter pour ce personnage en cliquant sur le bouton Vote.

Exemple de scénario : L’utilisateur sélectionne le personnage BulleBIzarre dans la liste « Top Ranking ». Le personnage s’affiche avec son compte de votes (80).



L’utilisateur clique sur le bouton Vote, le compte de votes passe à 81.

