Intra formatif - 4204W6

## Consignes (Pour le sommatif seulement)

* Vous avez 1h50 pour compléter l’examen.
* La seule documentation permise est celle qui accompagne cet examen.
* Si vous n’arrivez pas à compléter une fonctionnalité, faites-en le plus possible, quitte à mettre du code en commentaire pour montrer vos démarches et tentatives.
* Il n’y a aucun package npm à installer. Si un package est nécessaire, il est déjà installé.

# Lancer le serveur Web API et ouvrir les notes de cours

Ce fonctionnement sera identique pendant le sommatif, pratiquez-vous !

* Pour lancer le serveur Web API (auquel nous enverrons des requêtes), ouvrez Powershell dans …/formatif/serveurVincible et tapez les deux commandes suivantes :
  + **dotnet ef database update**
  + **dotnet run** (Gardez cette fenêtre Powershell ouverte jusqu’à la fin de l’examen)
* Pour accéder aux notes de cours, ouvrez Powershell dans .../formatif/notes/web et tapez les deux commandes suivantes :
  + **npm install**
  + **npm start** (Gardez cette fenêtre Powershell ouverte jusqu’à la fin de l’examen)
  + Les liens externes dans les notes de cours ne fonctionneront pas pendant le sommatif.

L’exécution du projet Angular (dossier **clientVincible**) se déroulera comme d’habitude.

# Projet de départ

Dans le projet de départ, il y a déjà 4 composants :

* **app** : Squelette de la page (header, nav, footer et intégration des 3 autres composants)
* **characters** : Page pour chercher des personnages par leur nom
* **superpowers** : Page pour chercher des personnages par leur superpouvoir
* **favorites** : Affiche les personnages favoris de l’utilisateur

Il y a également un service nommé **VincibleService**., **qui devra contenir toutes les requêtes http**.

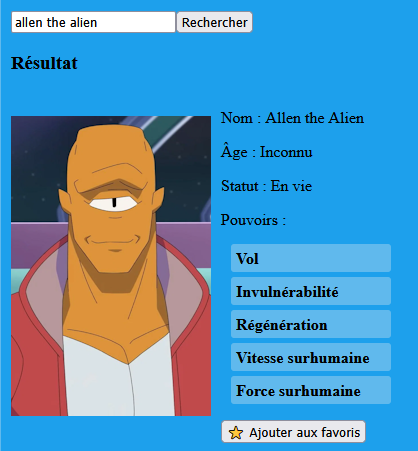


# Question 1 (5 points hypothétiques)

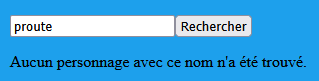
Dans le composant **characters**, rendez fonctionnels la recherche et l’affichage d’un personnage. La requête est utiliser est la suivante :

http://localhost:5254/api/Characters/GetCharacterByName/NOM\_RECHERCHÉ

Si le serveur retourne une erreur, affichez un message expliquant qu’aucun résultat n’a été trouvé.

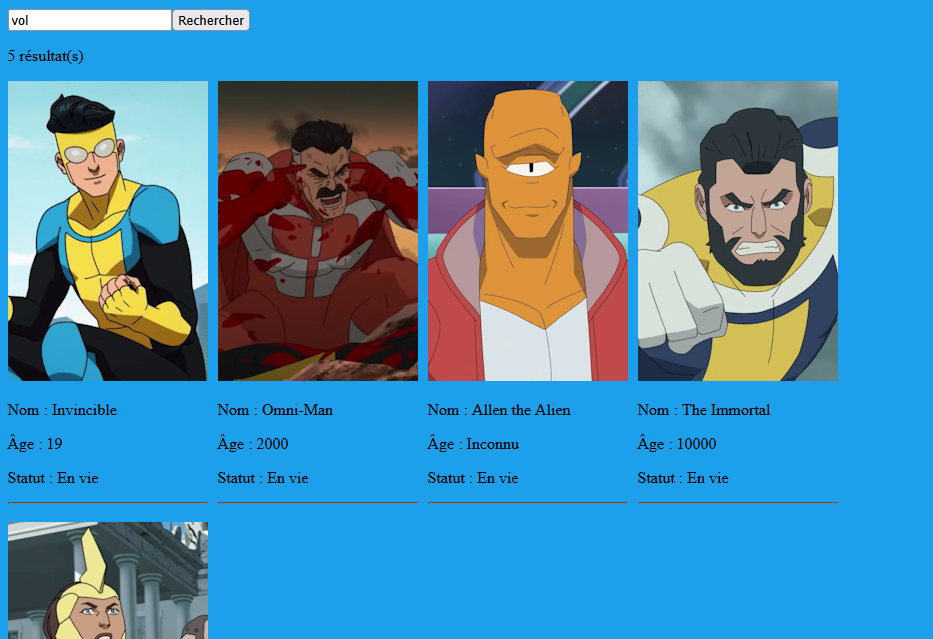


*Regardez comme Allen a plus de rizz que vous.*



# Question 2 (2 points hypothétiques)

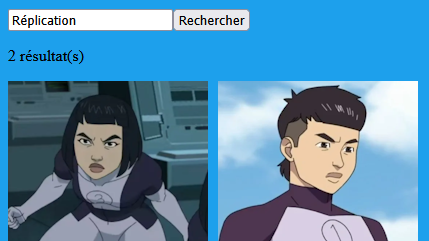
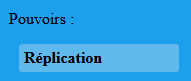
La requête de recherche de personnages par leur superpouvoir est déjà en partie implémentée dans le composant **superpowers**. Il vous reste à remplacer l’affichage *hardcodé* par **le tableau de personnages** retourné par la fonction qui contient la requête.



# Question 3 (2 points hypothétiques)

Dans le composant **characters**, nous aimerions que cliquer sur un superpouvoir (parmi la liste des superpouvoirs d’un personnage qu’on vient de rechercher) ait l’effet suivant :

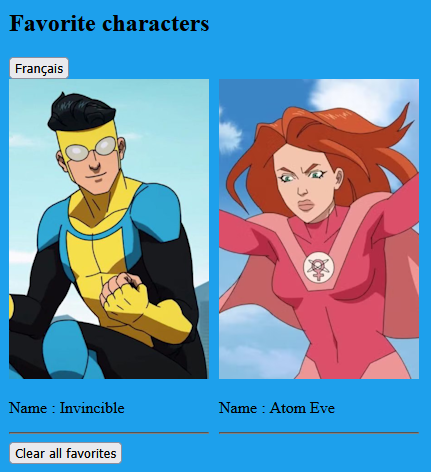
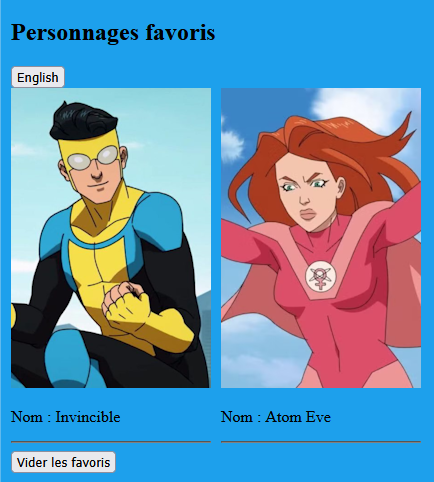
* On se déplace vers le composant **superpowers** grâce au routage.
* Le **nom du superpouvoir cliqué** est transmis en paramètre de route.
* Une fois dans le composant **superpowers** :
  + Le nom du superpouvoir reçu en paramètre de route est automatiquement affiché dans l’input de recherche. (Sans qu’on ait à l’écrire manuellement)
  + La recherche est automatiquement lancée avec le superpouvoir en question.



# Question 4 (3 points hypothétiques)

Internationalisez le composant **favorites** en français et en anglais et ajoutez un bouton ou un menu permettant d’alterner entre les deux langues quelque part dans la page.

(Si bouton : il doit afficher la



# Question 5 (3 points hypothétiques)

Rendez fonctionnel le bouton « ⭐ **Ajouter aux favoris** » dans le composant **characters**. Remplacez les personnages hardcodés dans le composant **favorites** par les personnages ayant été ajoutés en favoris par l’utilisateur. Pas besoin d’empêcher les doublons.

Les favoris doivent **persister** même si on réactualise la page.

Rendez également fonctionnel le bouton « **Vider les favoris** ». Son effet doit être immédiat lorsque utilisé.

Le formatif n’est pas à remettre. Bonne révision ! 😐