

## # Consignes Projet API

Projet par groupe (3 ou 4)

### ## Lexique

- API Application Programation Interface : Interface de programmation donnant accès à certaines données via des endpoints
- Endpoints : Point d'accès de l'API sous la forme d'une adresse URL

### ## 🎯 Objectif

Créer une application complète en Python avec le framework Django qui utilisera des données fournies par une **\*\*API externe\*\***. L'application devra stocker certaines données sur ses utilisateurs dans une base de données dédiée.

Gérer un projet *\_from scratch\_* en équipe, avec une répartition des tâches, un travail collaboratif, et de la communication.

Prévoir et rendre compte du travail au travers des documents rendus.

Utiliser une **\*\*API externe\*\*** et une **\*\*base de donnée\*\*** *"interne"*, et exploiter ces 2 sources de données en offrant un affichage cohérent et **\*\*agréable\*\*** à l'utilisateur

### ### 🗣️ Exemples

L'application utilise une API fournissant des animes. L'application propose aux utilisateurs de rechercher des animes, connaître des informations précises sur celui ci, si l'utilisateur est connecté il peut liker l'ainme pour l'enregistrer dans ses animes préférés.

Sur sa page compte, l'utilisateur peut voir les anime qu'il préfère et peut noter des choses dessus (le nombre d'épisodes vu / une note / ...). Il peut également rechercher d'autres utilisateurs et ainsi voir sa liste d'anime préférés.

L'application utilise une API fournissant des informations sur les différentes ligue nationales de football.

L'application permet aux utilisateurs de voir les matchs à venir et leur propose de parier sur les scores de chaque matchs. Les résultat donnent un certain nombre de points aux utilisateur, créant un classement entre tous les utilisateurs.

### ## 📌 Déroulé

#### ### 🧑 Etapes

- Choisir une API
- Imaginer les fonctionnalités de votre application
- Prévoir un planning
- Concevoir votre BDD
- Développer l'application

#### ### 📁 Livrables attendus

**\*\*En plus du code\*\*** quelques documents seront à rendre :

- Un planning
- Un cahier des charges
  - Contient toutes les fonctionnalités prévu dans votre appli
  - Ce que peuvent faire les utilisateurs non connectés / connectés / autres roles (admin / superuser / ...) si besoin
- Un diagramme de BDD (prévisionnel)
- Les outils utilisés dans le projet (pour le dev **\*\*ET\*\*** l'organisation du projet)
- Une charte graphique
  - Explication des choix graphiques (couleur, logo, experience utilisateur)
- Un readme.md complet
  - Explique comment démarrer votre application
  - Donne les fonctionnalités finalement développés et où les trouver dans le code
  - Explique le fonctionnement des appels API

Tous ces documents seront à déposer dans un repo **\*\*github\*\*** d'équipe auquel je devrais avoir accès.

### ### 📅 Planning

6 Séances (35h)

- 02/09 journée
- 17/09 journée ▶ Planning, Doc de conception, Charte graphique
- 25/11 journée
- 10/12 journée
- 16/12 demi-journée
- 18/12 demi-journée ▶ Code et readme

### ## 🗨️ API externe

Vous êtes libre de choisir n'importe quelle API disponible sur internet.

⚠️ Cependant faites très attention à l'accessibilité des API, certaines sont payante, d'autre limite le nombre d'appel par jour...

Si vous utilisez une API différente de celles proposées il faut **\*\*impérativement\*\*** la tester au plus vite. N'hésitez pas à venir me voir pour que je test la faisabilité avec vous.

2 Groupes peuvent avoir la même API mais **\*\*pas le même sujet\*\***

### ### ✅ API déjà testés

- Jinkan API 📖 : API de mangas / animes / personnages (très simple et performant)
- Open food facts 🍷 : API de produit (nourriture)

- Fonctionne mais un peu compliqué, le wrapper python fonctionne mais la doc est nulle il faut voir mon code pour comprendre, performance mauvaise un sujet peut être décider autour de ça (enregistrement de certaines données en BDD par exemple).

- API-football ⚽ : API de sports (au sens large)

- Limité à 30req/minute mais surtout 100/jours c'est peu, des solution de contournement peuvent être utilisé (peut faire partie du sujet)

- Marvel API 🦸 : Personnage et Comics

- Assez simple (besoin d'une clé et d'un hash)

- Création compte (gratuit) get API KEY & les req doivent fournir un TimeStamp et un hash (md5.fr --> (ts+privateKey+publicKey))

### ## 💡 Conseils et pièges à éviter

Faire du code **\*\*propre\*\*** vous êtes en équipe, votre code sera donc vu et lu par plusieurs personnes. Tout le monde doit être capable d'expliquer n'importe quelle partie du code (même si ce n'est pas lui qui l'a codé), si le code n'est pas lisible ce ne sera pas possible.

Donc donnez des **\*\*noms clairs\*\*** à vos variables/fonctions/classes, supprimez le code mort et gérez l'indentation avec soin. La propreté du code entre dans sa qualité et fait donc partie de la note finale.

Utilisez les différents débbugger quand vous avez des erreurs, si l'erreur vient du front utilisez **\*\*les dev tools\*\*** de votre navigateur. Si c'est du back utilisez les **\*\*\_print()\_\*\*** et le **\*\*\_debugger Python (Pdb)\_\*\***

⚠ Attention les utilisateurs géré par Django sont un peu particulier, les manipuler ou les modifier peut être difficile prenez le temps de **\*\*lire la doc\*\*** pour comprendre ce que vous faites.

**\*\*Mettez git en place au plus vite\*\***, mettez à jour votre repo local régulièrement et faites de petits commit, cela vous évitera de nombreux conflict

### ### Organisez votre dev :

- 3 axes pricipaux :

- Appel API

- Backend (Gestion utilisateur & enregistrement de données)

- Front (templates & style)

N'hésitez pas à travailler en binome (pair programming)

- Facilite le dev

- Partage la connaissance

- Moins fatigant

Pensez à **\_mock\_** les données si besoin, si vous n'arrivez pas à connecter rapidement l'API une personne peut quand même commencer le front avec de fausse données envoyer par le back.

Commencez par le plus difficile (ici l'API et le Back), en gardant cet ordre logique en tête : Donnée > Fonctionnalités > Style

Ne brulez pas les étapes, ne commencez pas les fonctionnalités avancées avant d'avoir les fonctionnalités de bases.

### **## Liens utiles**

- Appel API en python :

- [https://pythonds.linogaliana.fr/content/manipulation/04c\\_API\\_TP.html](https://pythonds.linogaliana.fr/content/manipulation/04c_API_TP.html)

- Débuter avec Django :

- <https://openclassrooms.com/fr/courses/7172076-debutez-avec-le-framework-django>

- Authentification avec Django :

- <https://docs.djangoproject.com/fr/4.2/topics/auth/default/>