

Cahier des Charges

API REST Zenless Zone Zero

DEWEERDT Simon - M2 CYBER

Sommaire

1 - Présentation du projet.....	2
1.1 - Contexte.....	2
1.2 - Objectifs.....	2
1.3 - Public cible.....	2
2 - Périmètre du projet	3
2.1 - Fonctionnalités principales	3
2.2 - Contraintes.....	3
3 - Architecture technique	4
3.1 - Schéma général	4
3.2 - Structure des endpoints	4
3.3 - Modélisation des données.....	5
4 - Organisation du projet.....	6
4.1 - Phases de développement	6
4.2 - Suivi et validation.....	6
5 - Livrables	6
6 - Conclusion	6

1 - Présentation du projet

1.1 - Contexte

Zenless Zone Zero est un jeu développé par HoYoverse, intégrant des personnages appelés agents, des créatures appelées bangboo, ainsi que divers équipements influençant le gameplay, tels que les moteurs amplis et les disques de poussée.

L'objectif de ce projet est de développer une API permettant d'accéder aux données détaillées de ces éléments, en s'appuyant sur du Web Scraping afin de maintenir à jour la base de données.

1.2 - Objectifs

- Développer une API avec FastAPI permettant de récupérer des données sur les agents, bangboo, moteurs amplis et disques de poussée.
- Extraire et stocker les données via la technique de Web Scraping.
- Fournir une documentation claire et détaillée sur l'utilisation de l'API.

1.3 - Public cible

- Les joueurs cherchant des informations sur les personnages et équipements
- Les développeurs souhaitant créer des outils ou applications basés sur l'univers de Zenless Zone Zero

2 - Périmètre du projet

2.1 - Fonctionnalités principales

- **Web Scraping** : Extraction automatique des données depuis des sources officielles ou fiables.
- **Stockage** : Structuration des informations dans une base de données PostgreSQL.
- **API REST** : Mise en place d'une API permettant d'accéder aux données via des endpoints spécifiques.
- **Filtres et recherche** : Permettre aux utilisateurs de rechercher et de filtrer les données par catégorie.
- **Mise à jour automatique** : Planification de mises à jour régulières des données.

2.2 - Contraintes

Contraintes techniques :

- Langage principal : Python (FastAPI pour l'API, librairies Requests et Selectolax pour le scraping)
- Base de données : PostgreSQL
- Utilisation de Docker pour le déploiement
- L'API doit respecter les principes **RESTful**
- Utilisation du format **JSON** pour les requêtes et les réponses
- Authentification via **API Key**
- Documentation via **Swagger**
- Scraping pour alimenter la base de données :
 - <https://zzz.honeyhunterworld.com/>
 - <https://wiki.hoyolab.com/pc/zzz/home>
 - https://zenless-zone-zero.fandom.com/wiki/Zenless_Zone_Zero_Wiki

Contraintes de performances :

- Optimisation du script de Web Scraping pour éviter une surcharge des serveurs sources
- Réponses rapides aux requêtes API

3 - Architecture technique

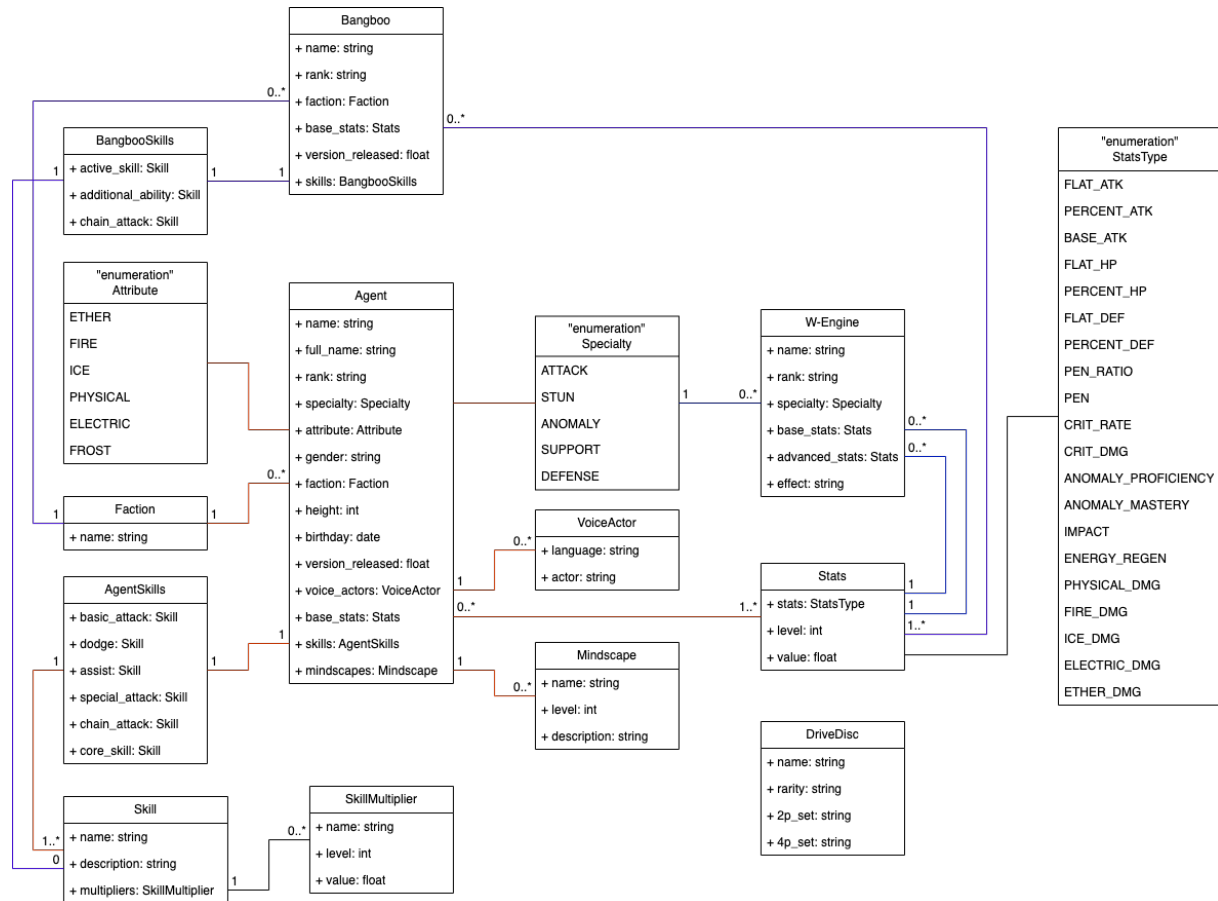
3.1 - Schéma général

- **Web Scraping** : Collecte et mise à jour des données
- **Base de données** : PostgreSQL
- **API** : Librairie Python FastAPI
- **Documentation** : Explication des endpoints et modes d'utilisation avec Swagger

3.2 - Structure des endpoints

Ressource	Méthode	URL
Récupérer les agents	GET	/agents
Récupérer un agent	GET	/agent/{name}
Ajouter un agent	POST	/agent
Modifier un agent	PUT	/agent/{name}
Supprimer un agent	DELETE	/agent/{name}
Récupérer les moteurs	GET	/w-engines
Récupérer un moteur	GET	/w-engine/{name}
Ajouter un moteur	POST	/w-engine
Modifier un moteur	PUT	/w-engine/{name}
Supprimer un moteur	DELETE	/w-engine/{name}
Récupérer les bangboo	GET	/bangboos
Récupérer un bangboo	GET	/bangboo/{name}
Ajouter un bangboo	POST	/bangboo
Modifier un bangboo	PUT	/bangboo/{name}
Supprimer un bangboo	DELETE	/bangboo/{name}
Récupérer les disques	GET	/drive-discs
Récupérer un disque	GET	/drive-disc/{name}
Ajouter un disque	POST	/drive-disc
Modifier un disque	PUT	/drive-disc/{name}
Supprimer un disque	DELETE	/drive-disc/{name}

3.3 - Modélisation des données



4 - Organisation du projet

4.1 - Phases de développement

1. **Étude et cadrage** : Définition des sources, analyse des besoins.
2. **Apprentissage** : Se former sur le Web Scraping et FastAPI
3. **Développement du Scraping** : Implémentation et stockage des données.
4. **Développement de l'API** : Mise en place des endpoints.
5. **Tests et validation** : Vérification de la robustesse et correction des erreurs.
6. **Déploiement et documentation** : Mise en ligne et finalisation du projet.

4.2 - Suivi et validation

- **GitHub** pour le versionnement du code
- **Tests unitaires et de performance** pour garantir la qualité
- **Documentation API** détaillée pour les utilisateurs

5 - Livrables

- **Code source** hébergé sur GitHub
- **Documentation API** et tutoriel d'installation
- **Rapports d'avancement** selon les livrables définis
- **Présentation finale du projet**

6 - Conclusion

Cette API servira de référence pour les joueurs et développeurs souhaitant exploiter les données du jeu Zenless Zone Zero. Une documentation détaillée sera fournie pour en faciliter l'utilisation.