



# API SPOTIFY

DAVY MARTHELLY, LAURIE JEHAM

## Introduction du Projet

Nous avons entrepris le développement d'une extension de l'API Spotify visant à améliorer l'expérience utilisateur en offrant des fonctionnalités supplémentaires telles que la gestion de groupes et la personnalisation des playlists. Utilisant JavaScript et Node.js, notre projet se distingue par une intégration poussée de SwaggerUI pour la documentation de l'API, assurant une interface claire et interactive pour les développeurs et les utilisateurs finaux.

## Décomposition du Projet

Le projet a été structuré en plusieurs phases clés, avec une emphase particulière sur la documentation dès le début.

### Documentation

Dès le départ, nous avons compris l'importance de la documentation pour assurer la maintenabilité et l'extensibilité de notre projet. Nous avons donc intégré SwaggerUI dès la phase initiale, permettant à toute personne utilisant notre API de comprendre facilement ses capacités et comment l'interfacer. La configuration de SwaggerUI a été réalisée dans `swagger.js`, et nous avons annoté nos routes dans `app.js` pour que Swagger puisse générer une documentation pertinente et à jour.

### Développement

Le cœur du développement s'est articulé autour de l'application principale (`app.js`), la gestion de l'authentification (`passport.js`), et la définition des modèles de données (`User.js`). Nous avons choisi Express pour sa simplicité et sa flexibilité dans la création d'APIs RESTful, et Mongoose pour interagir avec MongoDB, permettant une gestion fluide des utilisateurs et des groupes. La sécurité et l'authentification ont été renforcées par l'utilisation de Passport, supportant à la fois les stratégies locales et Spotify.

## Tests

BNous n'avons malheureusement pas eu le temps de mettre en place des tests autre qu'avec le service Swagger. Mais je sais qu'il existe des outils tels que Jest ou Mocha, pour garantir la fiabilité de notre code et prévenir les régressions.

## Répartition des Charges

Pour optimiser l'efficacité de notre travail en équipe, nous avons réparti les tâches en fonction de nos compétences et intérêts :

- **Documentation (laurie)** : L'intégration de SwaggerUI nécessitait une compréhension commune de notre API. Nous avons donc travaillé ensemble pour annoter nos routes et modèles de manière à générer une documentation claire et complète.
- **Développement de l'application (laurie davy) (app.js)** : Nous avons partagé le développement des fonctionnalités, veillant à implémenter une architecture solide et maintenable. Davy s'est concentré sur l'authentification et l'intégration avec l'API Spotify, tandis que Laurie a pris en charge la gestion des utilisateurs et des groupes.
- **Gestion de l'authentification (passport.js)** : Davy a configuré les stratégies d'authentification, assurant une connexion sécurisée et fluide pour nos utilisateurs.
- **Modèles de données (User.js)** : Davy travaillé sur la conception des schémas Mongoose pour les utilisateurs et les groupes, s'assurant que notre base de données soit bien structurée pour supporter les fonctionnalités de notre application.

## Conclusion

Notre projet d'extension de l'API Spotify, a réussi à offrir une solution robuste et bien documentée. En mettant un point d'honneur sur la documentation via SwaggerUI, nous avons non seulement simplifié l'utilisation de notre API pour les développeurs externes mais également posé une base solide pour les futures extensions de notre projet.



