

SoundBond (SB) Documentation

2021-08-31

Synthèse

L'équipe

L'équipe est composée de Maguette Diop, Paul Malinvaud, Mohamed Belhaimeur ainsi que Guilhem Tissot.

Le projet

Le projet SoundBond a été pensé en analogie avec un réseau social comme Instagram, ce dernier est omniprésent aujourd'hui et il paraît tout à fait naturel d'immortaliser ses souvenirs avec une photo de son plat, de la plage ou de ses amis. En regardant les réseaux sociaux déjà existants nous nous sommes dit que personne ne jouait exclusivement sur l'aspect du son. Il a l'avantage de pouvoir offrir un nouveau regard sur les souvenirs immortalisés et à l'avenir nous aimerions bien que l'application serve à transformer chaque utilisateur en DJ du quotidien.

Les personas

Utilisateur Instagram de longue date

Musicien ou faiseur de sons

Nouvel arrivant sur les réseaux sociaux

Prévisions marketing

Nous prévoyons de pousser notre projet auprès des utilisateurs par ...

Aspect techniques

Type d'application

L'application sera d'abord une web-application jusqu'à que le backend soit entièrement fonctionnel, puis ensuite nous développerons une application mobile.

Schéma architectural

Le schéma architectural suit le modèle MERN. MERN pour MongoDB, Express, ReactJS et NodeJS. Nous avons d'abord commencé avec cette architecture classique puis nous sommes passés de MongoDB à MySQL (de l'ODM mongoose à l'ORM Sequelize) car n'appréciant pas l'aspect (et les contraintes) de MongoDB. Dans notre cas, nous avons deux modèles de persistance de données, celle de nos classes comme les utilisateurs, les sons sociaux, etc... puis les sons en tant que ressource. Ces derniers sont stockés sur ... et le chemin pour y parvenir est inscrit dans notre base

MySQL.

Plateforme technologique

Avec ReactJS en front et NodeJS en back, nous nous retrouvons à utiliser quasiment exclusivement du JavaScript. Pour les frameworks de tests nous allons utiliser mocha ? et chai pour la lib d'assertions

Plateforme opérationnelle

Gestion de versions → git avec GitHub build → d'abord npm puis gulp pour tous les plugins (dont asciidoctor) qualité de code → CI → Travis puis potentiellement GitHub Actions Déploiement → ?

Interconnection avec systèmes et API distantes

Acquisition de données et usage intelligent

Partage de données avec BirdNet ?

Modélisation

Maquette du front

Diagramme de classes global

Description API

Diagramme de séquences des interactions front/back