

Documentation Technique - Projet Cosmosound

Models :

```
interface Playlist {
  id: string;
  title: string;
  dateCreation: string;
  owner: User;
  musics: Array<MusicDetails>;
}

export interface PlaylistMusic {
  id: string;
  title: string;
  playlistThumbnail: string;
  dateCreation: string;
  duration: number;
  genres: Array<Genre>;
  artist: Artist;
  catalog: Catalog;
}

enum TypeCatalog {
  SINGLE,
  ALBUM,
  EP,
}

interface Catalog {
  id: string;
  title: string;
  owner: Artist;
  dateCreation: string;
  thumbnail: string;
  type: TypeCatalog;
}

interface CatalogWithMusic extends Catalog {
  musics: Array<MusicDetails> | Array<Music>;
}

interface Music {
  id: string;
  title: string;
  dateCreation: string;
  duration: number;
  genres: Array<Genre>;
}

interface MusicDetails extends Music {
  artist: Artist;
  catalog: Catalog | CatalogWithMusic;
}

interface Genre {
  name: string;
}

interface User {
  id: number;
  name: string;
  email: string;
  dateCreation: string;
}
```

```

dateCreation: string;
pictureProfile: string;
followers: number;
}

interface UserDetails extends User {
    likedArtists: string[];
    likedMusics: string[];
    likedCatalogs: string[];
    likedGenres: string[];
    likedPlaylists: string[];
}

interface Artist extends UserDetails {
    isVerified: boolean;
    socialMedia: SocialMediaLink[];
    genre: Genre;
    artistName: string;
}

interface DetailedArtistInfo extends Artist {
    musics: MusicWithCatalog[];
    catalogs: Catalog[];
}

interface SocialMediaLink {
    media: Media;
    link: string;
}

enum Media {
    X,
    SPOTIFY,
    APPLE_MUSIC,
    YTB_MUSIC,
    INSTAGRAM,
}

```

API Endpoints :

Tous les endpoints de l'API seront préfixés par /api.

Playlist

Verbe	Endpoint	Request	Response	Response Code
GET	/playlists/:id	-	Playlist	200, 401, 404
GET	/playlists/search	?name=playlistName	Playlist[]	200, 401
POST	/playlists	{ "title": string }	{ "message": "Created", "id": string }	201, 401, 400
POST	/playlists/:id/music	{ "idMusic": string }	{ "message": "Created" }	201, 401, 400, 404, 403
DELETE	/playlists/:id	-	{ "message": "Deleted" }	200, 401, 404, 403
DELETE	/playlists/:id/music/:musicId	-	{ "message": "Music removed" }	200, 401, 404, 403

Catalogs

Verbe	Endpoint	Request	Response	Response Code
GET	/catalogs/:id	-	CatalogWithMusic	200, 401, 404
GET	/catalogs/search	?name=catalogName	CatalogWithMusic[]	200, 401
GET	/catalogs/musics/search	?name=musicName	CatalogWithMusic[]	200, 401
POST	/catalogs	CreateCatalogFormData	{ "message": "Created", "id": string }	201, 401, 400, 403
DELETE	/catalogs/:id	-	{ "message": "Deleted" }	200, 401, 404, 403
DELETE	/catalogs/:id/music/:musicId	-	{ "message": "Music removed" }	200, 401, 404, 403

```
interface CreateCatalogFormData {
  titleCatalog: string;
  thumbnailCatalog: File | string;
  musics: [
    title: string;
    duration: number;
    genres: string[];
  ];
}
```

Users

Verbe	Endpoint	Request	Response	Response Code
GET	/artists/:id	-	DetailedArtistInfo	200, 401, 404
GET	/artists/search	?name=artistName	DetailedArtistInfo[]	200, 401
PATCH	/users/:id	PatchUserData	{ "message": "Updated" }	200, 401, 400, 404, 403
PATCH	/artists/:id	PatchArtistData	{ "message": "Updated" }	200, 401, 400, 404, 403

```
interface PatchUserData {
  username?: string;
  password?: string;
  confirmPassword?: string;
  email?: string;
  image?: File;
}

interface PatchArtistData {
  artistName?: string;
  spotifyLink?: string;
  youtubeLink?: string;
  appleMusicLink?: string;
  xLink?: string;
  instagramLink?: string;
  genre?: Genre;
}
```

Authentification

Verbe	Endpoint	Request	Response	Response Code
POST	/auth/login	{ "email": string, "password": string }	{ "token": "jwtToken" }	200, 400
POST	/auth/register	{ "name": string, "email": string, "password": string, "genre": Genre }	{ "message": "User created" }	201, 400

Genre

Verbe	Endpoint	Request	Response	Response Code
GET	/genres	-	Genre[]	200, 401
GET	/genres/search	?name=genreName	Genre[]	200, 401

MongoDB Structure :

Schéma de Catalog

Field Name	Type	Description	Required	Index
id	ObjectId	Identifiant unique du catalogue	Oui	Non
title	String	Le titre du catalogue.	Oui	Oui
owner	ObjectId (ref: Artist)	L'artiste propriétaire du catalogue.	Oui	Non
dateCreation	Date	La date de création du catalogue.	Non	Non
thumbnail	String	L'URL de la miniature du catalogue.	Oui	Non
type	String (SINGLE, ALBUM, EP)	Le type de catalogue.	Oui	Non
musics	Objects[]	La liste des morceaux de musique du catalogue.	Oui	Non
musics.id	ObjectId	Identifiant unique du morceau de musique.	Oui	Non
musics.title	String	Titre du morceau de musique.	Oui	Non
musics.playlistThumbnail	String	URL de la miniature du morceau dans la playlist.	Oui	Non
musics.dateCreation	Date	Date de création du morceau de musique.	Oui	Non
musics.duration	Number	Durée du morceau de musique (en secondes).	Oui	Non
musics.genres	String[]	Liste des genres associés au morceau.	Oui	Non
musics.artist	ObjectId (ref: Artist)	Artiste associé au morceau.	Oui	Non
musics.catalog	ObjectId (ref: Catalog)	Catalogue auquel appartient le morceau.	Oui	Non

Schéma de Playlist

Field Name	Type	Description	Required	Index
id	ObjectId	Identifiant unique de la playlist	Oui	Non

Field Name	Type	Description	Oui Required	Oui Index
owner	objectId (ref: User)	L'utilisateur qui a créé la playlist.	Oui	Non
dateCreation	Date	La date de création de la playlist.	Non	Non
musics	Objects[]	La liste des morceaux de musique de la playlist.	Oui	Non
musics.id	objectId		Oui	Non
musics.title	objectId		Oui	Non
musics.	objectId		Oui	Non
musics.catalogId	objectId	L'ID du catalogue auquel appartient la musique.	Oui	Non

Shéma de User

Field Name	Type	Description	Required	Index
id	ObjectId	Identifiant unique de l'user.	Oui	Non
name	String	Nom de l'utilisateur	Oui	Non
email	String	Email de l'utilisateur	Oui	Unique
dateCreation	Date	Date de création	Non	Non
pictureProfile	String	Photo de profil	Non	Non
followers	Number	Nombre de followers	Non	Non
likedArtists	ObjectId[] (ref: User)	Liste d'artistes likés	Non	Non
likedMusics	ObjectId[] (ref: Catalog)	Liste de musiques likées	Non	Non
likedCatalogs	ObjectId[] (ref: Catalog)	Liste de catalogues likés	Non	Non
likedGenres	String[]	Liste des genres likés	Non	Non
likedPlaylists	ObjectId[] (ref: Playlist)	Liste des playlists likées	Non	Non
role	String	Rôle de l'utilisateur (USER OU ARTIST)	Oui	Oui
isVerified	Boolean	Si l'artiste est vérifié	Non	Non
socialMedia	Object[]	Liens des réseaux sociaux	Non	Non
socialMedia.media	String	Nom de la plateforme (x , SPOTIFY , etc...)	Non	Non
socialMedia.link	String	Lien du réseau social	Non	Non
genre	String	Genre musical	Non	Oui
artistName	String	Nom d'artiste	Non	Oui