

Documentation des Routes API

Routes Artists (/api/artists)

Méthode	Endpoint	Description	Paramètres d'entrée
GET	/api/artists	Récupère tous les artistes avec pagination	Query: <code>page</code> (numéro de page), <code>limit</code> (nombre par page), <code>search</code> (recherche par nom)
GET	/api/artists/filter	Filtre les artistes selon plusieurs critères	Query: <code>nameContains</code> (nom contient), <code>nameStartsWith</code> (nom commence par), <code>ids</code> (liste d'IDs séparés par virgule)
GET	/api/artists/top/:limit	Récupère les meilleurs artistes par nombre d'albums	Param: <code>limit</code> (nombre d'artistes à retourner)
GET	/api/artists/:id	Récupère un artiste par ID avec ses albums	Param: <code>id</code> (ID de l'artiste)
GET	/api/artists/:id/complete	Récupère un artiste avec albums et pistes complètes	Param: <code>id</code> (ID de l'artiste)
GET	/api/artists/:id/stats	Récupère les statistiques d'un artiste (total albums, pistes, durée, prix moyen)	Param: <code>id</code> (ID de l'artiste)
POST	/api/artists	Crée un nouvel artiste	Body: <code>name</code> (nom de l'artiste)
PUT	/api/artists/:id	Met à jour un artiste existant	Param: <code>id</code> (ID de l'artiste), Body: <code>name</code> (nouveau nom)
DELETE	/api/artists/:id	Supprime un artiste	Param: <code>id</code> (ID de l'artiste)

Routes Albums (/api/albums)

Méthode	Endpoint	Description	Paramètres d'entrée
GET	/api/albums	Récupère tous les albums avec leurs artistes	Aucun
GET	/api/albums/search	Recherche des albums par titre	Query: <code>q</code> (texte de recherche)
GET	/api/albums/artist/:artistId	Récupère tous les albums d'un artiste spécifique	Param: <code>artistId</code> (ID de l'artiste)
GET	/api/albums/:id	Récupère un album par ID avec son artiste	Param: <code>id</code> (ID de l'album)
GET	/api/albums/:id/tracks	Récupère un album avec toutes ses pistes	Param: <code>id</code> (ID de l'album)
GET	/api/albums/:id/stats	Récupère les statistiques d'un album (total pistes, durée, prix moyen/min/max)	Param: <code>id</code> (ID de l'album)
POST	/api/albums	Crée un nouvel album	Body: <code>title</code> (titre), <code>artistId</code> (ID de l'artiste)
PUT	/api/albums/:id	Met à jour un album	Param: <code>id</code> (ID de l'album), Body: <code>title</code> (titre), <code>artistId</code> (ID de l'artiste)
DELETE	/api/albums/:id	Supprime un album	Param: <code>id</code> (ID de l'album)

Routes Tracks (/api/tracks)

Méthode	Endpoint	Description	Paramètres d'entrée
GET	/api/tracks	Récupère toutes les pistes avec pagination, filtrage et tri	Query: page, limit, sortBy, sortOrder, search, minPrice, maxPrice, minDuration, maxDuration
GET	/api/tracks/search/advanced	Recherche avancée de pistes par plusieurs critères	Query: name (nom), composer (compositeur), albumTitle (titre d'album), artistName (nom d'artiste)
GET	/api/tracks/expensive/:limit	Récupère les pistes les plus chères	Param: limit (nombre de pistes)
GET	/api/tracks/longest/:limit	Récupère les pistes les plus longues	Param: limit (nombre de pistes)
GET	/api/tracks/stats	Récupère les statistiques globales des pistes (total, durée moyenne/min/max, prix moyen/total)	Aucun
GET	/api/tracks/price-range	Récupère les pistes groupées par gamme de prix	Aucun
GET	/api/tracks/:id	Récupère une piste par ID avec ses playlists	Param: id (ID de la piste)
GET	/api/tracks/:id/full	Récupère une piste avec détails complets (playlists, album, artiste)	Param: id (ID de la piste)
POST	/api/tracks	Crée une nouvelle piste	Body: name, albumId, mediaTypeId, genreId, composer, milliseconds, bytes, unitPrice
PUT	/api/tracks/:id	Met à jour une piste	Param: id (ID de la piste), Body: champs à mettre à jour
DELETE	/api/tracks/:id	Supprime une piste	Param: id (ID de la piste)

Routes Playlists (/api/playlists)

Méthode	Endpoint	Description	Paramètres d'entrée
GET	/api/playlists	Récupère toutes les playlists avec filtrage optionnel	Query: q (recherche par nom)
GET	/api/playlists/search/by-track	Trouve les playlists contenant une piste spécifique	Query: trackName (nom de la piste)
GET	/api/playlists/search/by-artist	Trouve les playlists contenant des pistes d'un artiste	Query: artistName (nom de l'artiste)
GET	/api/playlists/longest/:limit	Récupère les playlists avec le plus de pistes	Param: limit (nombre de playlists)
GET	/api/playlists/expensive/:limit	Récupère les playlists les plus chères (total des prix)	Param: limit (nombre de playlists)
GET	/api/playlists/:id	Récupère une playlist par ID avec ses pistes	Param: id (ID de la playlist)
GET	/api/playlists/:id/complete	Récupère une playlist avec détails complets (pistes, albums, artistes)	Param: id (ID de la playlist)
GET	/api/playlists/:id/stats	Récupère les statistiques d'une playlist (total pistes, durée, prix moyen/total)	Param: id (ID de la playlist)
GET	/api/playlists/:id/tracks/filter	Récupère les pistes d'une playlist avec filtres	Param: id (ID de la playlist), Query: minPrice, maxPrice, composer

Méthode	Endpoint	Description	Paramètres d'entrée
POST	/api/playlists	Crée une nouvelle playlist	Body: name (nom de la playlist)
POST	/api/playlists/:id/tracks	Ajoute des pistes à une playlist	Param: id (ID de la playlist), Body: trackIds (tableau d'IDs de pistes)
POST	/api/playlists/:id/tracks/bulk	Ajoute des pistes en masse selon des critères	Param: id (ID de la playlist), Body: artistId, albumId, minPrice, maxPrice
PUT	/api/playlists/:id	Met à jour le nom d'une playlist	Param: id (ID de la playlist), Body: name (nouveau nom)
DELETE	/api/playlists/:id	Supprime une playlist	Param: id (ID de la playlist)
DELETE	/api/playlists/:playlistId/tracks/:trackId	Retire une piste d'une playlist	Params: playlistId (ID de la playlist), trackId (ID de la piste)

Notes importantes

Pagination

- Les routes qui supportent la pagination utilisent les paramètres page et limit
- Par défaut: page=1, limit=20

Filtrage

- Les filtres de texte utilisent généralement LIKE ou iLike (insensible à la casse)
- Les filtres numériques supportent min et max pour définir des plages

Codes de statut HTTP

- 200 - Succès (GET, PUT)
- 201 - Créé avec succès (POST)
- 204 - Succès sans contenu (DELETE)
- 400 - Mauvaise requête
- 404 - Ressource non trouvée
- 422 - Erreur de validation
- 500 - Erreur serveur interne

Format de réponse

Toutes les réponses sont en format JSON. Les réponses d'erreur incluent un objet { error: "message" } .