Technical Design

Data Types

> These data types are used to to design the database schema. These will not necessarily be the data types used in the responses of the MagentaM3 API

Name	Type
User	`{ id: id, display_name: string, country: string (ISO 3166-1 alpha-2 code), email: string, playlists: [Playlist], images: [Image], uri: string (link) }`
Playlist	`{ id: id, collaborative: boolean, description: string, images: [Image], name: string, owner: User, tracks: [PlaylistTrack] snapshot_id: string, uri: string (link) }`
PlaylistTrack	`{ id: id, added_at: date, added_by: User, track: Track, playlist: Playlist }`
Track	`{ id: id album: Album, artists: [Artist], duration_ms: integer, disc_number: integer explicit: boolean, name: name, popularity: integer from 0 to 100, preview_url: string, track_number: integer, uri: string (link) // no images yet }`

Album

```
{
  id: id
  album_type: string one of ("album", "single", "compilation"),
  total_tracks: integer,
  images: [Image],
  name: string,
  release_date: date,
  release_date_precision: string one of ( "year", "month", "day"),
  uri: string (link),
  artists: [Artist]
}
```

Artist

```
{
  id: id,
  name: string,
  uri: string (link),
}
```

Image

```
{
  id: id,
  url: string (link),
  height: integer,
  width: integer
}
```

Interface

Method	tRPC Route	Description
Mutation	user.syncSpotifyData	Synchronises an MM3 user with Spotify data
Query	playlist.getPlaylists	Provides a list of playlists owned by the user
Query	playlist.getPlaylist	Returns information about a playlist including its tracks
Mutation	playlist.syncPlaylist	Synchronises an MM3 playlist with Spotify data
Mutation	playlist.updatePlaylist	Updates a playlist across MM3 and Spotify

Limitations

- No handling of market logic, content restrictions, explicit content information, copyrights
- Doesn’t support local files
- Doesn’t have external ids in entities
- No logic regarding genres for artists/albums or audio analysis for songs
- Only one-way relationship for album and tracks: can get album from track but not tracks from album
- Can’t handle episodes, only tracks
- No UserTrack entity that stores information about the track per user e.g. tags the user has used