

Shamzam Microservices Design

This design comprises two microservices:

- **Catalogue Service** – handles adding, removing, and listing tracks (S1, S2, S3).
- **Recognition Service** – recognizes a music fragment and maps it to a track (S4).

Each service exposes one or more REST endpoints, shown in the following diagrams. The diagrams are structured as follows:

- On the *left*, I list the HTTP **verb**, the **resource path**, and the **JSON input** (if any).
- In the *center*, I show the **microservice**.
- On the *right*, I list possible **response codes** and any **JSON output**.

1. Catalogue Service

Endpoint: Add a Music Track (S1)

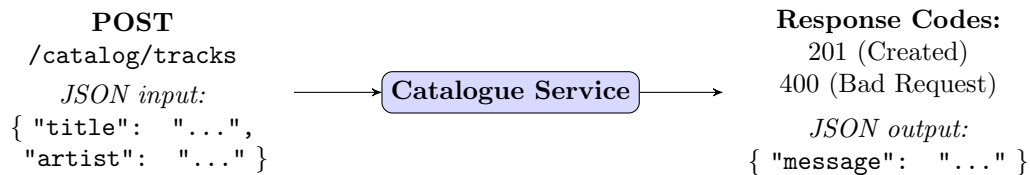


Figure 1: Adding a music track to the catalogue.

Endpoint: Remove a Music Track (S2)

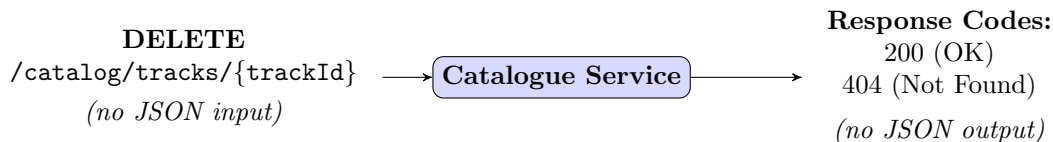


Figure 2: Removing a track from the catalogue.

Endpoint: List Music Tracks (S3)



Figure 3: Listing all tracks in the catalogue.

2. Recognition Service

Endpoint: Recognize Music Fragment (S4)



Figure 4: Converting an audio fragment to a known track.

When the client sends an audio fragment (e.g. `.wav` or `.mp3`), the Recognition Service:

- Forwards the fragment to an external API (e.g. `Audd.io`) to detect a match.
- If a match is found, returns the Shamzam `trackId` and metadata (title, artist, etc.) so the user may listen.
- If no match is found or the request is invalid, returns an appropriate error code.