# ISDB

## The Brief

Following the success of “IMDB”, the well known Internet Movie Database, we’re launching “ISDB” - the Internet Song Database. In this new world of interconnected software, our plan is to sell an API on this and leave it to customers to build the various front ends needed to use the database.

We’ve collected more than 3 000 tracks from 300+ albums covering from Metal to Classic.

As we intend to have the people pay for access, we obviously need to offer token based authentication.

The data will be provided to you as CSV files for:

- Tracks
- Albums
- Artists
- Genres

What we want is an API offering the following endpoints.

**All endpoints need to be securitized with token based auth except when mentioned.**
**All outputs should be JSON**

| URL | Method | Parameters | Expected output |
|---|---|---|---|
| **/register** | POST | Username, password | Newly created user<br>No token needed here |
| **/login** | POST | Username, password | Token if valid<br>No token needed here |
| **/tracks/:id** | GET | None | All information about the specific track, including genre and album info |
| **/genres** | GET | None | All genres (as an array) |

| | | | |
|---|---|---|---|
| **/albums/:id** | GET | None | All information about the specific album, including artist info |
| **/tracks** | POST | Name, Album id, Genre id, composer, duration, size in byte, prices | JSON message with the newly created record or proper error message if not valid<br><br>Album and genre have to exists already |
| /**artists/:id** | GET | None | All information about the specific artist |

# Deliverables

- A running Express / Mongo application deployed on heroku/mongodb with the mentioned routes and the data loaded in
- A GitHub repo that should include a README file containing explanations of the technologies used, the approach taken, installation instructions, unsolved problems, documentation assets, and any other useful information.
- A brief presentation of your work, its supporting documentation and code

# Assessment Criteria

**Score Legend:**

- 0 - Failing
- 1 - Progressing
- 2 - Performing
- 3 - Excelling

**Code Quality**

Is the code well-formatted? Are variable and function names semantic and sensible? Is the code easy to read and understand? Does it follow industry best practices?

**Technical Requirements**

How does the project stack up to the requirements? Is the developer making use of the material we've covered in a way that makes sense?

**API**

Is the application API consistent? Does it follow good practices regarding HTTP and JSON standards. Are error messages well documented?

**Functionality**
Does the application work without errors or bugs? Does it present a complete app, where every feature is fully implemented in a way that makes sense?

**Presentation**
Is there adequate documentation? Is the project repository well organized and free of clutter? Was the developer able to convey their ideas and work during the project presentation?

**Project Evaluation**
You will receive informal feedback for each of the 5 categories highlighting what you did well along with what can be improved. You will also receive highlights, growth opportunities, and overall comments for each project.
t
You need a score of at least 8 points out of a max of 15 to pass the project. Scoring 0 points ewill mean you have failed the project.

# Project Presentation

Each presentation will be **5 minute** long, in which you should answer the following questions:

- What is the project about?
- Is there any information you think might help us understand what you built?
- What features did you include?
    - Make sure to explain anything "new" (things that we didn't cover in class)
- What was your favorite part to work on?
- What was the most difficult part of the project?
- What would you like to add next?
- Demo of the website

The presentations will be recorded for elevation. Please go over the project assessment criteria to make sure you understand what is expected of you.

After each presentation, your instructors and classmates will have 5 minutes to share their questions and comments with you.

# Project Readme

In case, you are wondering what to include in your **README** file. Then checkout some of these suggestions and resources. But only when you have completed the MVP for your project.

- List technologies used
- Embedded images or links to wireframes and user stories.
- Document your planning and tell a story about your development process and problem-solving strategy
- List unsolved problems which would be fixed in future iterations.
- Describe how some of your favorite functions work

## Resources

- Markdown Cheatsheet
- How to write a good README for your GitHub project?
- Example Project Readme