

## Fullstack Lab1: Murwan Eisa

The project is a basic REST API that handles a music album collection. Node.js, Express.js, and MongoDB are used to build the API. The API's endpoints are as follows:

GET /api/albums to retrieve all albums

POST /api/albums to create a new album.

GET /api/albums/:title - returns an album based on its title.

PUT /api/albums/:id - to update an album based on its id.

DELETE /api/albums/:id - deletes an album based on its id.

Mongoose is used to define the music album structure, with the 'year' information saved as a Date object. When a new album is created, the year is formatted using the toISOString() method, which returns only the year, month, and day.

The biggest difficulty encountered was in implementing the PUT request. I attempted to update the album using the findByIdAndUpdate() method at first, but it did not work as planned. Although the corrected album was returned, the year field had not been updated. After some troubleshooting, I discovered that the problem was with how the year was given in the request body.

To fix this, I switched to the findOneAndUpdate() method and made sure to format the year field in the request body before sending it to the database. This resolved the problem, and the PUT request functioned as expected.

I used the Express Router to specify the endpoints and separate them into their own module, so routing was relatively simple. This made managing the many APIs and keeping the code organized much easier.

### Conclusion

Finally, the endeavor was a success. The API performed the required CRUD processes, and the PUT request issue was fixed. Overall, the project provided me with valuable hands-on experience with Node.js, Express.js, and MongoDB, and I learned a lot about designing REST APIs.