

Imagine that you need to create a backend **REST API** for a movie ticket booking portal.

1. General info:

This is a backend for a web application for managing movie tickets.

2. Functionality:

- a. Application should implement **Create/Read/Update/Delete** for the ticket model. Authorization is required. But no need to include user model, just imitate it by sending an encoded authorization token with a request and verify it on the backend, it can be some constant string.
- b. The ticket model should include but not limited to:
 - Creation date
 - Customer name
 - Movie title
 - Movie time
 - Ticket price

3. Application needs some analytics:

- a. How much money was earned by movie between 2 dates with division by months example response: [{month: 'September', summaryProfit: 8000}, {month: 'October', summaryProfit: 6000}, ...]
- b. How many people visited movie between 2 dates with division by months example response: [{month: 'September', summaryVisits: 800}, {month: 'October', summaryVisits: 600}, ...]
- c. Do the analytics in 2 ways –
 - DB aggregation
 - By using JS algorithms.
 - Define some query param to determine which method are used
For example, /analytics/visited?method=db-aggregation

4. Mandatory Deliverables

- a. The APIs should be protected i.e., only a logged-in with right privilege should be allowed to access that API.
- b. All the code should be well-styled with proper naming's. We pay a lot of attention to code organization (modularity, dependencies between modules, etc).
- c. Use Git for version control, and host the project in a public Github repository. Share the Github link with us.
- d. Readme file with instructions on how to run the project.
- e. Postman collection/ Swagger document with all the apis.

5. Notes:

You can use any framework. Koa/Express is preferred and as for the DB use Postgres. You can use any node packages that you need.

Earn extra brownie points for doing the following -

- All the above deliverables
- Using Typescript with good type definitions
- Add tests
- Host the application on a service like Heroku/Vercel
- Dockerize the service.
- Use a logging framework
- Articulate the architecture and the rationale behind your design choice

Please add README with a description of how to run the project

After finishing, please share the GitHub link to the project.