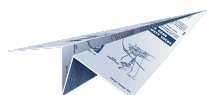




NodeJS Assessment

Flights CRUD Application



AVIOBOOK
a Thales Group Company

Firstly welcome and congratulations in getting to this stage in the process. This exercise is to test your NodeJS knowledge and development process. An important thing to note is we are interested in how you complete the task rather than the outcome itself. So be prepared to justify any decisions you make during the development.

Exercise

Overview

The exercise is to create a RESTful CRUD API that will be performing actions on a Flight entity. You should have been provided with an OpenAPI specification document and postman collection along with this document. These documents will provide you with the endpoints that will need to be created as well as example payloads.

Functional Requirements

Develop a RESTful API that adheres to the provided OpenAPI specification document. There are a total of 6 endpoints to implement:

- Generate a token that can be used to authenticate subsequent requests. Can be JWT or an opaque token if preferred.
- Schedule a new flight
- List all flights
- Retrieve an individual flight
- Update an existing flight
- Cancel a flight

Entities are provided in the specification as well as whether a property is required or not.

Technical Requirements

The interface should be developed adhering to RESTful principles and best practices. Choose whichever web framework you're familiar with but be prepared to justify why you used that framework. Please ensure that the application has:

- Authentication
 - Depending on your method of token generation, the application must be able to identify which user is making the request, as well as authentication the validity of the token.
- Validation
 - Validate the input parameters (example input is provided in the specification document)
 - Free to choose whichever framework you are most familiar
- Error Handling
 - Implement the error response codes in the specification document.
- Persistence
 - Use a MongoDB database to store the flights.
- Swagger UI endpoints
 - Integrate Swagger UI with the provided API specification so users can explore and test the endpoints.
- Extensive testing
 - Free to choose whichever testing framework

Delivery

You will have 7 days to complete the assessment, at which point you can then send us an email with a link to the GIT repository. The solution should be shared through a publicly available GIT repository (e.g. GitHub, Bitbucket, ...). We will then assess the code, and schedule an in person meeting to discuss the application you have created.

What gets evaluated

- Conformance to functional and technical requirements
- Code quality (we value good design guided by tests)
- Checkout-and-run convenience