

JavaScript Back End Developer Test

CHALLENGE

Context: We would like you to build a small API to test your knowledge of Back End Development and related technologies.

Every hour, the server should automatically initiate a request to the [Contentful API](#) to fetch data for 'Product' entries. This scheduled task ensures that the server retrieves the latest Product data from Contentful at regular one-hour intervals. It should insert the data from this API into a database and also define a REST API which the client (e.g. Postman) will be used to retrieve the data.

The service should provide a public module that returns paginated results with a maximum of 5 items per page and should be able to be filtered by product attributes (e.g., name, category, price range). The service should also allow users to remove items and these ones should not reappear when the app is restarted.

The service should also provide a private module of reports where you can obtain the following information:

1. Percentage of deleted products.
2. Percentage of non-deleted products with the following parameters:
 - a. With or without price.
 - b. With a custom date range.
3. A report of your choice.

In order to access the endpoints of the private module, an authorization parameter with a JWT must be sent in the headers. Please take note that there should be private and public modules.

STACK

You must use the following technologies to build the app:

- Active LTS version of Node.js + NestJS.
- Database: MongoDB or PostgreSQL.

- ORM: Mongoose or TypeORM.
- API Doc: Swagger, should be exposed at /api/docs.
- Docker

CONSIDERATIONS

- Node.js version active LTS
- The server component should be coded in TypeScript.
- At least 30% test coverage (statements) for the server component.
- The artifacts (server API) must be Dockerized.
- To start the project there should be a docker-compose that uses at least the server and database image.
- Tests and linters should run on GitHub Actions.
- Use conventional commit and gitflow.

Other than that, you are free to use any suitable npm or other libraries.

Send us your completed code as a public GitHub repo URL emailed to the recruiter who sent you this test.

When you finish, write an email to <recruiter email> to let us know.

Include a *README* file which explains anything we need to do to run the demo app, for example: setting up a database, forcing a data refresh to populate the DB for the first time, assumptions, choices, etc.

If you have any questions about the task, please let us know.

Once again, thank you for your time and we are looking forward to seeing the results!

Environment Variables

To access the Contentful API, use the following environment variables in your application:

```
CONTENTFUL_SPACE_ID=9xs161319f7v
CONTENTFUL_ACCESS_TOKEN=I-ThsT55eE_B3sCUWEQyDT4VqV03x__20ufuie9usns
CONTENTFUL_ENVIRONMENT=master
CONTENTFUL_CONTENT_TYPE=product
```