Case A - Backend

Technical Test - Case A - Backend

Welcome to the KOJI recruitment process, the objective of this test is to understand your technical level and your ability to resonate with a minimal level of information.

The estimated time for this test is around 1-2 hours

General information

- This test is to be done independently and should not include copy pasted elements from any source. (aka Github or StackOverflow)
- Using code you don't own apart from an external library will result in direct disqualification.
- You can make assumptions to complete the test. Don't forget to include them in your report.
- Your code must be provided by creating a pull request to this repository.
- Your code must be runnable only by following your README.md (the project will be run under the latest version of OSX)
- If you have any questions regarding this test, you should email them to the person in charge of your recruitment.
- You can use any external libraries you want.
- Any feature not described below will be considered as a bonus, you can customize anything if the requested features are present

General information

- This test is to be done independently and should not include copy pasted elements from any source. (aka Github or StackOverflow)
- Using code you don't own apart from an external library will result in direct disqualification.
- You can make assumptions to complete the test. Don't forget to include them in your report.
- Your code must be provided by creating a pull request to this repository.
- Your code must be runnable only by following your README.md (the project will be run under the latest version of OSX)

- If you have any questions regarding this test, you should email them to the person in charge of your recruitment.
- You can use any external libraries you want.
- The design is up to you, but keep in mind that this app must be easy to use.
- Any feature not described below will be considered as a bonus, you can customize anything if the requested features are present

Statement:

- You must create a NestJS API allowing the user to transform a Markdown document to a PDF.
- You can use the NestJS CLI to set up your API.
- The user must be able to provide a Markdown document and a list of key/values couples.
- The server will parse the Markdown document to replace the provided keys with their related values before the PDF transformation and return the resulted file to the user.
- The API must only accept a multi-part body and the file content must not be provided in plain text format.
- The multi-part key for the couples of key/value must be `replace`.
- The multi-part key for the file must be `file`.
- You don't have to handle the following edge cases:
  - Missing file permissions
  - Invalid file format

User Stories

- As a user, I want to send a Markdown file using the API so that I can the transformed PDF document.
- As a user, I want to provide a list of keys/values with my Markdown document so that they will be replaced in the Markdown document before its transformation to a PDF file.

Examples of `replace` key

JSON Format

```
JavaScript
```

```
{
  "Key1": "Value1",
  "FIRST_NAME": "Tristan",
  "LAST_NAME": "De Oliveira"
}
```

Comma separator format

JavaScript

```
KeyA:ValueA,FIRST_NAME:Tristan,LAST_NAME:De Oliveira
```

You could use `\` to escape the `,` separator if needed.