Documentation of code

Some important information

Comments are also added in the Code of server.js.
Please also add -o “filename” in the curl request and run it in that directory where you want to save it.

1. How code is working:

It has these parts:

a. Starting the Server
b. Getting the required Credentials
c. Getting the Required Data as JSON file
d. Modifying the JSON Data according to API Input
e. Giving Document API Input and credentials to get PDF file in the result
f. Cleaning the Already existing PDF with merged_document.pdf name
g. Storing the PDF file in local storage
h. Responding with that PDF file
i. Deleting that pdf file
j. Error handling of 400, 401, 404, 500 Error Codes

2. What was be the Problems and how they are solved?

First Problem: result.writeTostream not working for response although response is itself a writable stream.

Solution: So I used saveAsfile function.

Second Problem: Saving in local Storage results in problem in Multiple requests

Solution: I used Mutex lock to get Atomicity so that only one PDF will get formed and then deleted. And No two users response will be conflicting.

Third Problem: Private data storing for saving it in local storage, that user may question.

Solution: It will get solved once writeTostream Function starts working on Response object as well, then it will directly write to response stream and no private data storing. Although my code is going to delete it at last. But This will be the best option, which I suggested in previous lines.

Fourth Problem: Handling Multiple Requests, But Adobe API is having timeout of 10 seconds.

Solution: When there will be many users, the timeout will get reached, as I am also setting a timeout of 3 seconds for correctly saving a PDF file. I can’t change that ADOBE API timeout from here. But will definitely do if I get selected and get access to it. Really eager to work on these projects more. These are FUN :)

THANK YOU FOR REVIEWING MY WORK

TANUJ KAMBOJ

IIIT DELHI