Nathaniel James

Group of pictures summary

[strawhatx/Group-of-Pictures (github.com)](https://github.com/strawhatx/Group-of-Pictures)

My implementation of the Group of Pictures prompt was written using NodeJS, typescript as the programming language with Jade as the template language. I opted for typescript for its type definitions just makes the JavaScript classes cleaner. I also chose jade over EJS for its easy approach.

Once I began developing, I started out working with Ffmpeg’s CLI, I chose to work with Node’s child processes (Exec) and promises to return the standard output outside of the call back. Initially it was going well until I got to outputting the Individual clipped mp4’s. It wasn’t very easy to find a solution while staying with Ffmpeg’s CLI. So, I gave Fluent Ffmpeg a try, a wrapper library for Ffmpeg. But it wasn’t compatible with routing-controllers, so I end up having to opt for the Express router. Once I did that was able to bring together project to its current state

There were many concepts I had to learn, specifically understanding how video processing works, working with the command line in NodeJS, the basics of working with video, applying Ffmpeg and more. So, as far as future developers are concerned, they could greatly benefit by having an implementation which will help to explain how to work with Ffmpeg, and methods of working with command line tools.

Because Ffmpeg is a command line tool I had to spend most of the time learning how to interact with it. I think the app as it is, couldn’t go to production, so I don’t think it would support feature requests. I ran it a few issues when writing running the code. You have to allocate space depending on how large the file is, the current video was less than a minute, so image something like 15 minutes. Not only that, the implementation for the segmentation results in multiple requests. This solution would not be scalable. But it can be remedied with more time and a dedicated server

If this project were supporting features like dynamic cropping and filtering, I would suggest implementing a method that maintains a dictionary of Ffmpeg commands. Then have the controller route support query strings. Once a query is passed, we have a method that validates the query stings then compares the keys from both sides and appends the result to the Ffmpeg command.