UDACITY

PROJECT SPECIFICATION

Image Processing API

Setup and Architecture

CRITERIA	MEETS SPECIFICATIONS
Set up a project structure that promotes scalability	• Source code is kept separate from compiled code. • All tests should be contained in their own folder. • Separate modules are created for any processing.
Set up an npm project	• Package.json should contain both devDependencies, and dependencies. • Scripts should be created for testing, linting/prettier, starting the server, and compiling TS. • Build script should run without error.

Functionality

CRITERIA	MEETS SPECIFICATIONS
Add and use Express to a node.js project	• Start script should run without error • Provided endpoint should open in the browser with status 200

CRITERIA	MEETS SPECIFICATIONS
Folow middleware documentation to use middleware to create an API	• Accessing the provided URL with image information should successfully resize an image and save it to disk on first access, then pull from disk on subsequent access attempts. • An error message should be provided to the user when an image has failed to process or does not exist.

Code Quality

CRITERIA	MEETS SPECIFICATIONS
Write relevant unit tests with Jasmine and SuperTest to improve code quality and refactoring	• Test script runs and all tests created pass. • There is at least 1 test per endpoint and at least one test for image processing.
Utilize TypeScript to avoid errors and improve maintainability	• All code in the SRC folder should use the .ts filetype. • Functions should include typed parameters and return types and not use the `any` type. • Import and Export used for modules. • Build script should successfully compile TS to JS.
Write well-formatted	Prettier and Lint scripts should run without producing any error messages.

linted code

CRITERIA	MEETS SPECIFICATIONS

Suggestions to Make Your Project Stand Out!

1. Add additional processing to accept and output other image formats than JPG.
2. Modify the thumbnail filename to include the image size to allow for multiple sizes of the same image.
3. Further explore the options in the Sharp module and add additional processing options.
4. Add in logging to record when images are processed or accessed.
5. Create a front-end for uploading more images to the full-size directory.
6. Create a front-end that displays a thumbnail directory.
7. Create a front-end that allows for the selection of how to process a selected image.