

Motivation

To define the technical design that is used for the implementation of the Mini Image Processor Service as a requirement to apply for the position at Ubersnap.

Product Requirement

- User should be able to convert image files from PNG to JPEG
- User should be able to resize an image according to specified dimensions.
 - This is regardless of whether or not the image will look squished.
- User should be able to compress images to reduce file size while maintaining reasonable quality.

Solution Description

1. We will create three APIs to fulfill all the requirements:
 - a. /convert
 - b. /resize
 - c. /compress
2. All three of these APIs will accept a PNG file, but some APIs will accept other additional parameters in the form data.
3. We will return either the file if the operation is a success or a JSON if it fails.

Edge Cases

1. If the image is too large, REST API cannot limit the size of the image. I think the size limit feature could be added on the front end side.

Architecture Decisions

1. Tech Stack
 - a. Golang 1.19
 - b. Echo
 - c. GoCV

System Design

API Design

API Name	Description	Endpoint	Authentication	Request Schema	Request Sample	Response Schema	Response Sample	Expected Caller	Expected QPS
Convert	Given a file, check if it is a valid PNG file. If it is a valid PNG file convert it to JPEG.	/convert POST	None	Form data	Form data: File: file.png	File/JSON	File.jpg Or { "error": "missing file parameter" }	Front End	<=1
Resize	Given a file and some parameters, check if it is a valid PNG file. Resize according to the parameters if it is.	/resize	None	Form Data	height:200 width:300 Interpolation_flag:0 File: file.png	File/JSON	File_resized.png Or { "error": "missing file parameter" }	Front End	<=1
Compress	Given a file, check if it is a valid PNG file. Compress it with compression level 9 if it is.	/compress/	None	Form Data	Form data: File: file.png	File/JSON	File_compressed.png Or { "error": "missing file parameter" }	Front End	<=1