This is a repository for my image repository application written using Erlang/OTP22.

Instructions to install the application:

1. Install Erlang in your system.
2. Copy all the contents under the *'lib\my_image_repository-1.0.0'* directory. For my case: *'C:\Program Files\erl10.7\lib\my_image_repository-1.0.0'*

Start the application by double clicking on the *start_shell.bat* file

Once you are into the shell execute the below command to create a record structure within the Erlang shell:

rd(image_data, {id, image_name, b64_image, raw_image_size, b64_image_size}).

- Here image_data is the name of the structure. This is a disc copy withing the application directory.
- Id is automatically incrementing integer value from 1. The table is an ordered set which gets ordered based on this id.
- Image_name is the name of the image which should have '.jpg' or '.png' at the end. (can be extended)
- B64_image is the base 64 encoded image. Initially while inserting an image, the absolute path of the image should be given as input. Application reads the image and encodes it in base64 and stores it in the Mnesia database.
- Raw_image_size is the size of the image without base64 encoding.
- B64_image_size is the size of the image without base64 encoding.

COMMANDS

INSERTING IMAGE DATA:

SINGLE IMAGE eg:

my_image_repository_utils:insert_images(#image_data{image_name = "Img 1.jpg", b64_image="c:/users/amalr/desktop/mypic.jpg"}). // single image_data record

MULTIPLE IMAGES eg:

Image1 = #image_data{image_name = "Img 1.jpg", b64_image="c:/users/amalr/desktop/mypic1.jpg"}.

Image2 = #image_data{image_name = "Img 2.jpg", b64_image="c:/users/amalr/desktop/mypic2.jpg"}.

my_image_repository_utils:insert_images([Image1, Image2, ….]). // list of image_data records

DELETING IMAGE DATA:

SINGLE IMAGE eg:

my_image_repository_utils:delete_image(1). //the Id of the Image

MULTIPLE IMAGE eg:

my_image_repository_utils:delete_image([1,2,…]). // list of Ids

DISPLAYING ALL IMAGES:

my_image_repository_utils:filter_image(select_all).

FILTERING IMAGES:

Based on size:

my_image_repository_utils:filter_image({raw_image_size, <size>}).

my_image_repository_utils:filter_image({b64_image_size, <size>}).

<size> : integer // in bytes

Based on name:

my_image_repository_utils:filter_image({image_name, <name>}).

<name> : string or binary string // eg: “name” or <<”name”>>

Based on Id:

my_image_repository_utils:filter_image({id, <id>}).

<id> : integer