

# Schemas

**Photo**, to be used in the ‘**photo**’ collection:

Attribute	Type	Conditions	Values
name	String	required:true	Any valid string
tags	List	–	[Any valid strings]
location	String	–	Any valid string
image_file	Image	required:true	Any valid image file
albums	List of <b>Album</b>	–	[albumIds]

**Album**, to be used in the ‘**album**’ collection:

Attribute	Type	Conditions	Values
name	String	required:true unique: true	Any valid string
description	String	–	Any valid string

# Operations

## Note

All GET requests with a body should use the query string to pass the parameters.

**Photo**:

Request	Type	Route	Request Body	Response Body	Response Status Code
Create	POST	/listPhoto	Photo (**)	{ message: 'Photo successfully created', id: <b>photo_id</b> }	201
Read	GET	/listPhoto/{photo_id}	–	Single database object { name: <b>name</b> , tags: <b>[tags]</b> , location: <b>location</b> , albums: <b>[albums]</b> , file: <b>image_file</b> }	200

Request	Type	Route	Request Body	Response Body	Response Status Code
Read	GET	/listPhotos	tag (*)	Multiple database objects [{name: name, location: location, file: image_file}]	200
Read	GET	/listPhotos	albumName (*)	Multiple database objects [{name: name, location: location, file: image_file}]	200
Update	PUT	/listPhoto/{photo_id}	Photo (~)	{ message: 'Photo successfully updated', id: photo_id }	200
Delete	DELETE	/listPhoto/{photo_id}	—	{ message: 'Photo successfully deleted', id: photo_id }	200

(\*) This is a query string as in the note above.

(\*\*) The request body is a multi-part form, i.e., **file** (the image) as **request.files** and the associated json data as **request.form**

(~) **image\_file** is not sent as part of request body. The remaining **Photo** is sent as json.

All the images by default are placed in an **Album** with the name **Default**. When any **Photo** entry is created the POST method should check if the default album exists and create it if it does not. Each image belongs to the default Album as well as to any additional albums to which it is added to. The albumName in GET /listPhotos is optional, thus, it could be empty. In the latter case, the default **Album** should be used.

The **image\_file** returned in the three GET requests discussed in the above table should be encoded using **base64** encoding and then decoded using **utf-8**. This has been illustrated in the scaffolding code provided at the end of requirements section. No encoding/decoding is required while saving to the database.

## Note

All `/listAlbum` requests use JSON in their request body.

Album:

Request	Type	Route	Request Body	Response Body	Response Status Code
Create	POST	/listAlbum	Album	{ message: 'Album successfully created', id: <code>album_id</code> }	201
Read	GET	/listAlbum/{album_id}	–	Single database object { id: <code>album_id</code> , name: <code>name</code> }	200
Update	PUT	/listAlbum/{album_id}	Album	{ message: 'Album successfully updated', id: <code>album_id</code> }	200
Delete	DELETE	/listAlbum/{album_id}	–	{ message: 'Album successfully deleted', id: <code>album_id</code> }	200