

# photomanager

REST API project in Django

made by Robert Rachubiński

e-mail: robert.rachubinski@gmail.com

## EXAMPLE OF REQUESTS USING POSTMAN

### 1) POST new photo from local file

The screenshot shows a Postman interface for a POST request. The URL is `http://127.0.0.1:8000/photos`. The request body is a JSON object with the following fields: `"title": "strawberries", "album_id": 1, "url": "/home/robert/media_examples/strawberries.jpg"`. The response is a 201 status code, created in 1908 ms, with a body size of 491 B. The response body is a JSON object with the following fields: `"id": 1, "title": "strawberries", "album_id": 1, "width": 1350, "height": 898, "color_dominant": "d01c23", "url": "/home/robert/media_examples/strawberries.jpg"`.

```
POST http://127.0.0.1:8000/photos
```

Params Auth Headers (8) **Body** Pre-req. Tests Settings Cookies

raw Text

```
1 {
2   ... "title": "strawberries",
3   ... "album_id": 1,
4   ... "url": "/home/robert/media_examples/strawberries.jpg"
5 }
```

Body 201 Created 1908 ms 491 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "title": "strawberries",
4   "album_id": 1,
5   "width": 1350,
6   "height": 898,
7   "color_dominant": "d01c23",
8   "url": "/home/robert/media_examples/strawberries.jpg"
9 }
```

## 2) POST new photo from external API

POST

http://127.0.0.1:8000/photos

Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

raw Text

```
1 {
2   "url": "https://jsonplaceholder.typicode.com/photos/3"
3 }
```

Body 201 Created 228 ms 518 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "title": "officia porro iure quia iusto qui ipsa ut modi",
4   "album_id": 1,
5   "width": 600,
6   "height": 600,
7   "color_dominant": "24f355",
8   "url": "https://via.placeholder.com/600/24f355"
9 }
```

## 3) POST new photo from JSON file

POST

http://127.0.0.1:8000/photos

Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

raw Text

```
1 {
2   "url": "/home/robert/photomanager/manager/example_photos/photo_json.json"
3 }
```

Body 201 Created 17 ms 484 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 3,
3   "title": "et qui rerum",
4   "album_id": 2,
5   "width": 600,
6   "height": 600,
7   "color_dominant": "14ba42",
8   "url": "https://via.placeholder.com/600/14ba42"
9 }
```

#### 4) GET all photos

GET ⌵ http://127.0.0.1:8000/photos Send ⌵

Params Auth Headers (8) Body ● Pre-req. Tests Settings Cookies

Body ⌵ 🌐 200 OK 13 ms 836 B Save Response ⌵

Pretty Raw Preview Visualize JSON ⌵ ≡ 📋 🔍

```
1  [
2    {
3      "id": 1,
4      "title": "strawberries",
5      "album_id": 1,
6      "width": 1350,
7      "height": 898,
8      "color_dominant": "d01c23",
9      "url": "/home/robert/media_examples/strawberries.jpg"
10   },
11   {
12     "id": 2,
13     "title": "officia porro iure quia iusto qui ipsa ut modi",
14     "album_id": 1,
15     "width": 600,
16     "height": 600,
17     "color_dominant": "24f355",
18     "url": "https://via.placeholder.com/600/24f355"
19   },
20   {
21     "id": 3,
22     "title": "et qui rerum",
23     "album_id": 2,
24     "width": 600,
25     "height": 600,
26     "color_dominant": "14ba42",
27     "url": "https://via.placeholder.com/600/14ba42"
28   }
29 ]
```

#### 5) GET photo with defined id=3

GET ⌵ http://127.0.0.1:8000/photos/3 Send ⌵

Params Auth Headers (8) Body ● Pre-req. Tests Settings Cookies

Body ⌵ 🌐 200 OK 6 ms 486 B Save Response ⌵

Pretty Raw Preview Visualize JSON ⌵ ≡ 📋 🔍

```
1  {
2    "id": 3,
3    "title": "et qui rerum",
4    "album_id": 2,
5    "width": 600,
6    "height": 600,
7    "color_dominant": "14ba42",
8    "url": "https://via.placeholder.com/600/14ba42"
9  }
```

## 6) PUT photo with defined id=3

(example of changing the title from "et qui rerum" into "NEW EDITED TITLE")

PUT

http://127.0.0.1:8000/photos/3

Send

ParamsAuthHeaders (8)BodyPre-req. TestsSettingsCookies

rawText

```
1 {
2   "id": 3,
3   "title": "NEW EDITED TITLE",
4   "album_id": 2,
5   "width": 600,
6   "height": 600,
7   "color_dominant": "14ba42",
8   "url": "https://via.placeholder.com/600/14ba42"
9 }
```

Body200 OK36 ms490 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 3,
3   "title": "NEW EDITED TITLE",
4   "album_id": 2,
5   "width": 600,
6   "height": 600,
7   "color_dominant": "14ba42",
8   "url": "https://via.placeholder.com/600/14ba42"
9 }
```

## 7) DELETE photo with defined id=3

DELETE

http://127.0.0.1:8000/photos/3

Send

ParamsAuthHeaders (8)BodyPre-req. TestsSettingsCookies

Body204 No Content17 ms389 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "message": "Photo id=3 was deleted successfully."
3 }
```

## 8) Try GET photo with defined id=3, which was deleted

The screenshot shows a REST client interface. The method is set to GET and the URL is `http://127.0.0.1:8000/photos/3`. The response status is 404 Not Found, with a response time of 8 ms and a body size of 382 B. The response body is displayed in JSON format, showing a message: `"message": "The photo id=3 does not exist!"`.

## 9) Tests (they are written in `/photomanager/manager/tests.py`)

run from terminal from `/photomanager`: `python manage.py test`

```
robort@robort-Lenovo: ~/photomanager
(photomanager) robort@robort-Lenovo:~/photomanager$ python manage.py test
Found 13 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
...log from api.view.photo_list -> request.method=='POST':
...data in post request: {'title': 'apple', 'album_id': 1, 'url': 'manager/example_photos/apple.jpg'}
...IMPORT SOURCE: local_photo_file
...response data after post: {'id': 3, 'title': 'apple', 'album_id': 1, 'width': 500, 'height': 555, 'color_dominant': '9ec23d', 'url': 'manager/example_photos/apple.jpg'}
.
...log from api.view.photo_list -> request.method=='POST':
...data in post request: {'url': 'https://jsonplaceholder.typicode.com/photos/1'}
...IMPORT SOURCE: external_api
...response data after post: {'id': 3, 'title': 'accusamus beatae ad facilis cum similique qui sunt', 'album_id': 1, 'width': 600, 'height': 600, 'color_dominant': '92c952', 'url': 'https://via.placeholder.com/600/92c952'}
.
...log from api.view.photo_list -> request.method=='POST':
...data in post request: {'url': 'manager/example_photos/photo_json.json'}
...IMPORT SOURCE: json_file
...response data after post: {'id': 3, 'title': 'et qui rerum', 'album_id': 2, 'width': 600, 'height': 600, 'color_dominant': '14ba42', 'url': 'https://via.placeholder.com/600/14ba42'}
.....
Ran 13 tests in 2.551s
OK
```

## 10) CLI script for import photos from external API or from JSON file.

run from terminal from /photomanager/manager: *python cli\_script.py*

### 10a) Import from external API:

```
robert@robert-Lenovo: ~/photomanager/manager
(photomanager) robert@robert-Lenovo:~/photomanager/manager$ python cli_script.py

Script that calls the import from 'external API' https://jsonplaceholder.typicode.com/photos/ + 'id'
or from 'JSON file' with a similar structure.

Example input url: https://jsonplaceholder.typicode.com/photos/5
Example input url: manager/example_photos/photo_json.json (or full system path)

Input source url: https://jsonplaceholder.typicode.com/photos/5
status_code = 201
CREATED: {"id": 4, "title": "natus nisi omnis corporis facere molestiae rerum in", "album_id": 1, "width":
: 600, "height": 600, "color_dominant": "f66b97", "url": "https://via.placeholder.com/600/f66b97"}
```

### 10b) Import from JSON file:

```
(photomanager) robert@robert-Lenovo:~/photomanager/manager$ python cli_script.py

Script that calls the import from 'external API' https://jsonplaceholder.typicode.com/photos/ + 'id'
or from 'JSON file' with a similar structure.

Example input url: https://jsonplaceholder.typicode.com/photos/5
Example input url: manager/example_photos/photo_json.json (or full system path)

Input source url: /home/robert/photomanager/manager/example_photos/photo_json.json
status_code = 201
CREATED: {"id": 5, "title": "et qui rerum", "album_id": 2, "width": 600, "height": 600, "color_dominant":
"14ba42", "url": "https://via.placeholder.com/600/14ba42"}
```