

LeadIQ Technical Challenge

Intro

Your challenge is to build an [Imgur](#) image uploading service that exposes a REST API.

Clients of this service:

- Submit image upload jobs
 - Each upload job is an array of image URLs. The service takes each URL, downloads the content, and uploads it to Imgur.
- Get the status of an image upload job
 - Every job returns a jobId which can be used to check if the job is completed yet.
 - Returns the lists of image URLs that are still waiting to be processed, were successfully uploaded, and those that failed.
- Get a list of all images uploaded
 - Return a list of Imgur links to all of the images uploaded

The service should be Dockerized.

Requirements

- Implement the API outlined in the next section
- Do not use Imgur's upload by link functionality - the service needs to download the image at the given URL and re-upload to Imgur.
- Use your choice of language/platform
- OAuth client ID and secret can be embedded into the service config or injected at runtime.
- Do not use any external datastore - use in-memory data structures.
- `GET /v1/images` should return the links of all images
- `POST /v1/images/upload`
 - Should return immediately. The uploading happens asynchronously.
 - Should log the reason for failed image uploads.
- Write unit tests to test your functions
- Delete print() & test code that you used for testing
- Finally, please add a Dockerfile
- If anything is unclear, you can come up with your own assumptions. Please also state them in the README together with instructions on how to run the service and any documentation that you think is necessary

API

Submit image URLs for upload

Submits a request to upload a set of image URLs to Imgur. The images will be uploaded to the configured Imgur, viewable by anyone with the link.

Request

`POST /v1/images/upload`

The request is just a JSON body, no query parameters.

Request body

Attributes:

- `urls`: An array of URLs to images that will be uploaded. Duplicates should be stripped out.

Example request body:

```
{
  "urls": [
    "https://farm3.staticflickr.com/2879/11234651086_681b3c2c00_b_d.jpg",
    "https://farm4.staticflickr.com/3790/11244125445_3c2f32cd83_k_d.jpg"
  ]
}
```

Response

On success, returns immediately with an appropriate status code with the id of the job.

Response body

Attributes:

- `jobId`: The id of the upload job that was just submitted.

Example response body:

```
{
  "jobId": "55355b7c-9b86-4a1a-b32e-6cdd6db07183",
}
```

Get upload job status

Gets the status of an upload images job.

Request

```
GET /v1/images/upload/:jobId
```

The request has no body and no query parameters. :jobId is an ID returned from the POST upload images API.

Request body

None

Response

On success, returns immediately with an appropriate status code with the id of the job.

Response body

Attributes:

- id: The id of the upload job.
- created: When job was created. In ISO8601 format (YYYY-MM-DDTHH:mm:ss.sssZ) for GMT.
- finished: When job was completed. In same format as created. Is null if status is not complete.
- status: The status of the entire upload job. Is one of:
 - pending: indicates job has not started processing.
 - in-progress: job has started processing.
 - complete: job is complete.
- uploaded: An object of arrays containing the set of URLs submitted, in several arrays indicating the status of that image URL upload (pending, complete, failed).

Example response body:

```
{
  "id": "55355b7c-9b86-4a1a-b32e-6cdd6db07183",
  "created": "2017-12-22T16:48:29+00:00",
  "finished": null,
  "status": "in-progress",
```

```
{
  "uploaded": {
    "pending": [
      "https://www.factslices.com/imgs/black-cat.jpg",
    ],
    "complete": [
      "https://i.imgur.com/gAGub9k.jpg",
      "https://i.imgur.com/skSp0.jpg"
    ],
    "failed": [
    ]
  }
}
```

Get list of all uploaded image links

Gets the links of all images uploaded to Imgur. These links will be accessible by anyone.

Request

```
GET /v1/images
```

The request has no body and no query parameters.

Request body

None

Response

On success, return an array of the Imgur links to the successfully uploaded images. The links should be public.

Response body

Attributes:

- **uploaded**: An array of the Imgur links to the uploaded images.

Example response body:

```
{
  "uploaded": [
    "https://i.imgur.com/gAGub9k.jpg",
    "https://i.imgur.com/skSp0.jpg"
  ]
}
```