

REST API Documentation for MOOC Project

Andrew Lalis
andrewlalisofficial@gmail.com

Last modified on
April 8, 2018

Contents

1	Introduction	2
2	Documentation Template	2
2.1	Title	2
3	Endpoints	4
3.1	Title	4

1 Introduction

This document lists in detail each of the requests that can be made to the MOOC Projectserver. The API relies on the REST principles, and all responses will be in the form of an HTTP result code, such as 200 `SUCCESS` and 404 `NOT FOUND`, and if applicable, an accompanying JSON object containing the response data.

2 Documentation Template

Each possible command will be structured in the following manner:

2.1 Title

Additional information about the API call. Use verbs which match the request type, and proper plurality (one vs. multiple).

2.1.1 URL

The URL structure (path only, no root url). Any variable endpoints or values are denoted with a `:`.

2.1.2 Method

The request type GET — POST — DELETE — PUT

2.1.3 URL Params

If there are URL parameters, name them as they appear in the URL section, denoting any constraints, and separating optional and required parameters.

Required:

`id=[integer]`

Optional:

`photo_id=[alphanumeric]`

2.1.4 Data Params

If a post request is sent, what should the payload body look like? Below is an example payload body.

```
{
  "id": 12345,
  "name": "Andrew"
}
```

2.1.5 Success Response

What should the status code be on success, and what, if any, data is returned? Give some example content.

Code: 200 **Content:**

```
{
  "id": 12345,
  "count": 5
}
```

2.1.6 Error Response

What gets returned when things go wrong? All possible errors should be listed here, even if it seems repetitive to do so.

Code: 401 UNAUTHORIZED **Content:**

```
{  
    "error": "Log in"  
}
```

OR

Code: 404 NOT FOUND **Content:** *none*

2.1.7 Sample Call

*A sample call to the endpoint in a runnable format (\$.ajax call or curl request).
This makes it easier for someone to implement your request.*

2.1.8 Notes

All commentary, discussion, or miscellaneous info should be shared here.

3 Endpoints

3.1 Title

Additional information about the API call. Use verbs which match the request type, and proper plurality (one vs. multiple).

3.1.1 URL

The URL structure (path only, no root url). Any variable endpoints or values are denoted with a `:`.

3.1.2 Method

The request type GET — POST — DELETE — PUT

3.1.3 URL Params

If there are URL parameters, name them as they appear in the URL section, denoting any constraints, and separating optional and required parameters.

Required:

`id=[integer]`

Optional:

`photo_id=[alphanumeric]`

3.1.4 Data Params

If a post request is sent, what should the payload body look like? Below is an example payload body.

```
{
  "id": 12345,
  "name": "Andrew"
}
```

3.1.5 Success Response

What should the status code be on success, and what, if any, data is returned? Give some example content.

Code: 200 **Content:**

```
{
  "id": 12345,
```

```
        "count": 5
    }
```

3.1.6 Error Response

What gets returned when things go wrong? All possible errors should be listed here, even if it seems repetitive to do so.

Code: 401 UNAUTHORIZED **Content:**

```
{
    "error": "Log in"
}
```

OR

Code: 404 NOT FOUND **Content:** *none*

3.1.7 Sample Call

A sample call to the endpoint in a runnable format (\$.ajax call or curl request). This makes it easier for someone to implement your request.

3.1.8 Notes

All commentary, discussion, or miscellaneous info should be shared here.