

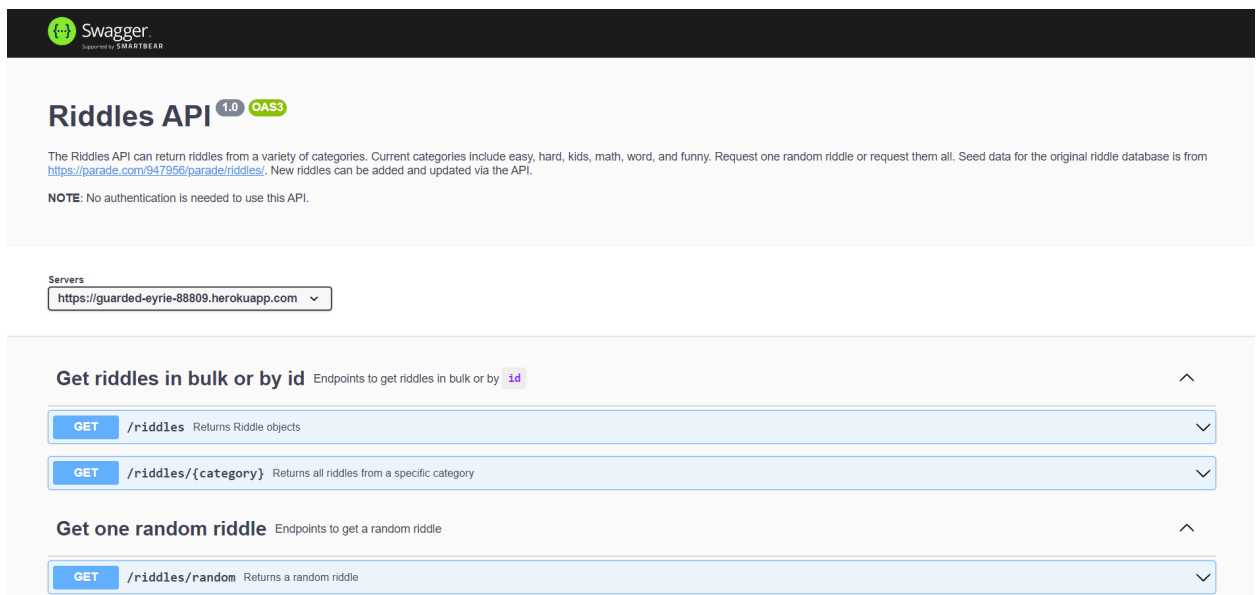
# Introduction

I created this simple API to experiment with API documentation tools. This project was created with Node.js with Express and MongoDB with Mongoose.

I am using [Heroku](#) to host a live version of the app so that the interactive tools are usable.

For this project, I've chosen to document the API with three different tools: [Swagger UI](#), [Stoplight](#), and [Markdown](#). I created a yaml file that follows the [OpenAPI Specification Version 3.0.3](#) and used both Swagger UI and Stoplight to render the documentation. I also created the documentation using Markdown, which makes up the main part of this README file. Select the links below to check out the results of each.

- [API documentation using Swagger UI](#)



The screenshot shows the Swagger UI for the 'Riddles API'. At the top, there's a Swagger logo and the text 'powered by SWAGGER'. Below that, the title 'Riddles API' is followed by '1.0' and 'OAS3' in green badges. A descriptive paragraph states: 'The Riddles API can return riddles from a variety of categories. Current categories include easy, hard, kids, math, word, and funny. Request one random riddle or request them all. Seed data for the original riddle database is from <https://parade.com/947956/parade/riddles/>. New riddles can be added and updated via the API.' A note below says: 'NOTE: No authentication is needed to use this API.' Under the 'Servers' section, a dropdown menu shows 'https://guarded-eyrie-88809.herokuapp.com'. The main content area lists two groups of endpoints. The first group, 'Get riddles in bulk or by id', has two endpoints: 'GET /riddles Returns Riddle objects' and 'GET /riddles/{category} Returns all riddles from a specific category'. The second group, 'Get one random riddle', has one endpoint: 'GET /riddles/random Returns a random riddle'. Each endpoint is in a blue box with a 'GET' method label, a dropdown arrow on the right, and a small 'id' or 'category' label in a purple box.

- API documentation using Stoplight

rmcknight

← Home

Riddles API

RIDDLES API

Overview

Get riddles in bulk or by id

Returns Riddle objects

Returns all riddles from a specific category

Get one random riddle

Returns a random riddle

Returns a random riddle from the catego...

Create or delete a riddle

Modify a riddle

Riddle Object

Sign in

powered by Stoplight

Search Riddles API...

GET

/riddles

Use the optional `id` query parameter to have the corresponding Riddle object returned. Omit the query parameter to have an array of all the Riddle objects in the database returned.

Query Parameters

`id` string optional

Examples: `ee8b0dfaf5c7f8a32f16887c`

Optional: If the database id for a riddle is known, use this parameter to get that specific Riddle object. **Omit for all riddles to be returned.**

Responses

200

OK: A successful server response. The requested content will be returned.

404

500

Schema

Example-Individual

Example-Array

Returned Riddle Object (5) The JSON object returned from a "GET" request.

`_id` string The database id for the Riddle object.

`riddle` string The riddle's question.

Try It Out

GET

https://guarded-eyrie-88809

/riddles

Query

Headers

Body

Path

Code Generation

id

Value

Key

Value

Send

- API documentation using Markdown

☰ README.md

🔗 Endpoints for getting one random riddle

📌 To select a random riddle from the entire database:

HTTP Method	Endpoint	Summary
GET	/riddles/random	Returns a random riddle

Make a `GET` request. A random Riddle object from the entire database is selected and returned.

Example curl request:

```
curl -X GET 'https://guarded-eyrie-88809.herokuapp.com/riddles/random'
```

A successful response will return an HTTP status code of `200`. The Riddle object returned uses this schema:

Field	Type	Description
<code>_id</code>	string	The database id for the Riddle object
<code>riddle</code>	string	The riddle's question
<code>answer</code>	string	The riddle's answer
<code>category</code>	string	A classification of the riddle. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
<code>source</code>	string	The source of the riddle

Example successful response:

# Riddles API Documentation

## Contents

---

- [Overview](#)
- [Authentication](#)
- [Usage](#)
  - [Get riddles in bulk or by id](#)
  - [Get a random riddle](#)
  - [Create or delete a riddle](#)
  - [Modify a riddle](#)

## Overview

---

Are you looking for a way to add riddles to your application? Rather than build the feature yourself, you can use this pre-built API. The Riddles API can return riddles from a variety of categories. Current categories include easy, hard, kids, math, word, and funny. Request one random riddle or request them all! Riddles in the original database are from <https://parade.com/947956/parade/riddles/>. New riddles can be added and updated via the API.

**Riddles API Server:** <https://guarded-eyrie-88809.herokuapp.com>

## Authentication

---

No authentication is needed to use this API.

## Usage

---

The API's endpoints and usage are listed and described in the following sections:

- Get riddles in bulk or by id:
  - Get all riddles or an individual riddle
  - Get all riddles from a specific category
- Get one random riddle:
  - Select a random riddle from the database
  - Select a random riddle from a specific category
- Create or delete a riddle:
  - Add a riddle to the database
  - Delete a riddle from the database
- Modify a riddle:
  - Update an existing riddle by overwriting
  - Update a field of an existing riddle

## Getting riddles in bulk or by id

 To get all riddles or an individual riddle

HTTP Method	Endpoint	Summary
GET	/riddles	Returns Riddle objects

Use the optional `id` query parameter to have the corresponding Riddle object returned. Omit the query parameter to have an array of all the Riddle objects in the database returned.

The optional query parameter is defined as:

Parameter	Type	Required	Description
id	string	Optional	If the database id for a riddle is known, use this parameter to get that specific Riddle object. <b>Omit for all riddles to be returned.</b>

## Example request

A curl request to get *all* riddles:

```
curl -X GET 'https://guarded-eyrie-88809.herokuapp.com/riddles'
```

## Example response

A successful response has an HTTP status code of **200** and is an array of Riddle objects:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

[
  {
    "_id": "60bc3ffb5dcf8ac52f16a7fd",
    "riddle": "What has to be broken before you can use it?",
    "answer": "An egg",
    "category": "easy",
    "source": "https://parade.com/947956/parade/riddles/"
  },
  {
    "_id": "60bc3ffb5dcf8ac52f16a800",
    "riddle": "I'm tall when I'm young, and I'm short when I'm old. What am I?",
    "answer": "A candle",
    "category": "easy",
    "source": "https://parade.com/947956/parade/riddles/"
  }
]
```

You can use the same endpoint with the **id** query parameter to get an individual riddle.

## Example request

A curl request to get *an individual* riddle:

```
curl -X GET  
'https://guarded-eyrie-88809.herokuapp.com/riddles?id=60bd0708d7dcc31bd9376abe'
```

## Example response

A successful response has an HTTP status code of **200** and is an individual Riddle object:

```
HTTP/1.1 200 OK  
Content-Type: application/json; charset=utf-8  
  
{  
  "_id": "60bd0708d7dcc31bd9376abe",  
  "riddle": "I'm tall when I'm young, and I'm short when I'm old. What am I?",  
  "answer": "A candle",  
  "category": "easy",  
  "source": "https://parade.com/947956/parade/riddles/"  
}
```

The Riddle object returned uses this schema:

Field	Type	Description
_id	string	The database id for the Riddle object
riddle	string	The riddle's question
answer	string	The riddle's answer
category	string	A classification of the riddle. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
source	string	The source of the riddle

## Possible error responses

A **GET** request to the /riddles endpoint may return the following errors:

Error code	Description
404	<b>Not Found:</b> This error will occur if the requested riddle <code>id</code> is not found in the database. It could also occur if the database is empty.
500	<b>Internal Server Error:</b> An unexpected error occurred on the server.

[Back to usage list](#)

### To get all riddles from a specific category

HTTP Method	Endpoint	Summary
<code>GET</code>	<code>/riddles/{category}</code>	Returns all riddles from a specific category

Pass the desired category as a path parameter in the request. An array of Riddle objects matching the specific category is returned.

The path parameter is defined as:

Parameter	Type	Required	Description
category	string	Required	A riddle category. The seed data (the initial riddles added to the database) include the following categories: easy, hard, funny, kids, math, and word.

### Example request

A curl request to get all riddles from the `easy` category:

```
curl -X GET 'https://guarded-eyrie-88809.herokuapp.com/riddles/easy'
```

### Example response

A successful response is an array of Riddle objects:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
[
  {
    "_id": "60bd0708d7dcc31bd9376abe",
    "riddle": "I'm tall when I'm young, and I'm short when I'm old. What am I?",
    "answer": "A candle",
    "category": "easy",
    "source": "https://parade.com/947956/parade/riddles/"
  }
]
```

The Riddle objects returned use this schema:

Field	Type	Description
_id	string	The database id for the Riddle object
riddle	string	The riddle's question
answer	string	The riddle's answer
category	string	A classification of the riddle. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
source	string	The source of the riddle

### Possible error responses

A **GET** request to the `/riddles/{category}` endpoint may return the following errors:

Error code	Description
404	<b>Not Found</b> : This error will be returned if there are no riddles with the requested <b>category</b> in the database.
500	<b>Internal Server Error</b> : An unexpected error occurred on the server.



[Back to usage list](#)

---

## Getting one random riddle

---

 To select a random riddle from the entire database:

HTTP Method	Endpoint	Summary
GET	/riddles/random	Returns a random riddle

Make a GET request. A random Riddle object from the entire database is selected and returned.

### Example request

A curl request to get a random riddle:

```
curl -X GET 'https://guarded-eyrie-88809.herokuapp.com/riddles/random'
```

A successful response will return an HTTP status code of 200. The Riddle object returned uses this schema:

Field	Type	Description
_id	string	The database id for the Riddle object
riddle	string	The riddle's question
answer	string	The riddle's answer
category	string	A classification of the riddle. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
source	string	The source of the riddle

## Example response

A successful response is an individual Riddle object:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "_id": "60bd0708d7dcc31bd9376abe",
  "riddle": "I'm tall when I'm young, and I'm short when I'm old. What am I?",
  "answer": "A candle",
  "category": "easy",
  "source": "https://parade.com/947956/parade/riddles/"
}
```

## Possible error responses

A **GET** request to the `/riddles/random` endpoint may return the following errors:

Error code	Description
404	<b>Not Found</b> : This error will be returned if there are no riddles in the database.
500	<b>Internal Server Error</b> : An unexpected error occurred on the server.

[Back to usage list](#)



To select a random riddle from a specific category:

HTTP Method	Endpoint	Summary
<b>GET</b>	<code>/riddles/{category}/random</code>	Returns a random riddle from the category specified

Pass the desired category as a path parameter in the request. A random Riddle object from the specified category will be selected and returned.

The path parameter is defined as:

Parameter	Type	Required	Description
category	string	Required	A riddle category. The seed data (the initial riddles added to the database) include the following categories: easy, hard, funny, kids, math, and word.

### Example request

A curl request to get a random riddle from the `easy` category:

```
curl -X GET 'https://guarded-eyrie-88809.herokuapp.com/riddles/easy/random'
```

A successful response will return an HTTP status code of `200`. The Riddle object returned uses this schema:

Field	Type	Description
<code>_id</code>	string	The database id for the Riddle object
<code>riddle</code>	string	The riddle's question
<code>answer</code>	string	The riddle's answer
<code>category</code>	string	A classification of the riddle. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
<code>source</code>	string	The source of the riddle

### Example response

A successful response is an individual Riddle object:

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

```
{
  "_id": "60bd0708d7dcc31bd9376abe",
  "riddle": "I'm tall when I'm young, and I'm short when I'm old. What am I?",
  "answer": "A candle",
  "category": "easy",
  "source": "https://parade.com/947956/parade/riddles/"
}
```

## Possible error responses

A **GET** request to the `/riddles/{category}/random` endpoint may return the following errors:

Error code	Description
404	<b>Not Found</b> : This error will be returned if there are no riddles with the requested <b>category</b> in the database.
500	<b>Internal Server Error</b> : An unexpected error occurred on the server.

[Back to usage list](#)

---

## Creating or deleting a riddle

 **To add a new riddle to the database**

HTTP Method	Endpoint	Summary
<b>POST</b>	<code>/riddles</code>	Adds a new riddle to the database

Use the JSON request body to add a new Riddle object to the database.

The request body should use the following schema:

Field	Type	Required	Description
riddle	string	Required	The riddle's question
answer	string	Required	The riddle's answer
category	string	Required	A one-word classification of the riddle. It should not contain spaces. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
source	string	Optional	The source of the riddle

**Note:** The `Content-Type` header should be set to `application/json`.

### Example request

A curl request to add a new riddle:

```
curl -X POST \
  'https://guarded-eyrie-88809.herokuapp.com/riddles' \
  -H 'Content-Type: application/json' \
  -d '{
    "riddle": "There\'\'s only one word in the dictionary that\'\'s spelled wrong.
What is it?",
    "answer": "The word \'\'wrong.\'\' It\'\'s the only word that\'\'s spelled
W-R-O-N-G.",
    "category": "kids",
    "source": "https://www.prodigygame.com/main-en/blog/riddles-for-kids/"
  }'
```

A successful response will return an HTTP status code of 201 and have the following schema:

Field	Type	Description
message	string	A brief success message
content	object	Top-level containing Riddle object

Where the Riddle object has this schema:

Field	Type	Description
_id	string	The database id for the Riddle object
riddle	string	The riddle's question
answer	string	The riddle's answer
category	string	A classification of the riddle. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
source	string	The source of the riddle
__v	number	An internal versioning number used by Mongoose (the Object Data Model library used to connect to the MongoDB database).

## Example response

A successful response to a `POST` request is an object like below:

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8

{
  "message": "Successfully added new riddle",
  "content": {
    "_id": "60bd0708d7dcc31bd9376abe",
    "riddle": "I'm tall when I'm young, and I'm short when I'm old. What am I?",
    "answer": "A candle",
    "category": "easy",
    "source": "https://parade.com/947956/parade/riddles/",
    "__v": 0
  }
}
```

## Possible error responses

A `POST` request to the `/riddles` endpoint may return the following errors.

Error code	Description
400	<b>Bad Request</b> : This error will be returned if a required field in the request body is missing or if the <code>category</code> field contains a space.
500	<b>Internal Server Error</b> : An unexpected error occurred on the server.

[Back to usage list](#)

---

## To delete a riddle from the database

HTTP Method	Endpoint	Summary
DELETE	/riddles	Deletes a riddle from the database

Select a Riddle object to delete by setting the query parameter to its `id`. Currently, deleting seed data (the initial riddles added to the database) is not allowed.

The query parameter is defined as:

Parameter	Type	Required	Description
id	string	Required	The database <code>id</code> for the riddle. This is needed for <code>PUT</code> , <code>PATCH</code> , or <code>DELETE</code> requests which are operations performed on an existing riddle.

### Example request

A curl request to delete a riddle:

```
curl -X DELETE  
'https://guarded-eyrie-88809.herokuapp.com/riddles?id=60bc3adb1e6946b94ca7a70a'
```

A successful response will return an HTTP status code of `200` and have the following schema:

Field	Type	Description
message	string	A brief success message

### Example response

A successful response to a `DELETE` request:



```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "message": Successfully deleted riddle"
}
```

## Possible error responses

A **DELETE** request to the /riddles endpoint may return the following errors:

Error code	Description
403	<b>Forbidden</b> : This error will be returned if you try to delete the seed data (the initial riddles added to the database).
404	<b>Not Found</b> : This error will occur if the requested riddle <b>id</b> is not found in the database.
500	<b>Internal Server Error</b> : An unexpected error occurred on the server.

[Back to usage list](#)

## Modifying a riddle

### To update an existing riddle by overwriting

HTTP Method	Endpoint	Summary
<b>PUT</b>	/riddles	Updates (by overwriting) a riddle in the database

Select a Riddle object to update by setting the query parameter to its **id**. Use the JSON request body to set new values. Fields set in the request body will overwrite all existing values. Omitting a non-required field from the request body sets that field to null.

Currently, updating seed data (the initial riddles added to the database) is not allowed.

The query parameter is defined as:

Parameter	Type	Required	Description
id	string	Required	The database <code>id</code> for the riddle. This is needed for <code>PUT</code> , <code>PATCH</code> , or <code>DELETE</code> requests which are operations performed on an existing riddle.

The request body should use the following schema:

Field	Type	Required	Description
riddle	string	Required	The riddle's question
answer	string	Required	The riddle's answer
category	string	Required	A one-word classification of the riddle. It should not contain spaces. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
source	string	Optional	The source of the riddle

**Note:** The `Content-Type` header should be set to `application/json`.

### Example request

A curl request to update (overwrite) a riddle:

```
curl -X PUT \
  'http://guarded-eyrie-88809.herokuapp.com/riddles?id=60bd0708d7dcc31bd9376abe' \
  -H 'Content-Type: application/json' \
  -d '{
    "riddle": "There\'\'s only one word in the dictionary that\'\'s spelled wrong.
What is it?",
    "answer": "The word \'\'wrong.\'\' It\'\'s the only word that\'\'s spelled
W-R-O-N-G.",
    "category": "kids",
    "source": "https://www.prodigygame.com/main-en/blog/riddles-for-kids/"
  }'
```

A successful response will return an HTTP status code of **200** and have the following schema:

Field	Type	Description
message	string	A brief success message
content	object	Top-level containing Riddle object

Where the Riddle object has this schema:

Field	Type	Description
_id	string	The database id for the Riddle object
riddle	string	The riddle's question
answer	string	The riddle's answer
category	string	A classification of the riddle. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
source	string	The source of the riddle

## Example response

A successful response to a `PUT` request is an object like below:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "message": "Successfully updated riddle",
  "content": {
    "_id": "60bd0708d7dcc31bd9376abe",
    "riddle": "I'm tall when I'm young, and I'm short when I'm old. What am I?",
    "answer": "A candle",
    "category": "easy",
    "source": "https://parade.com/947956/parade/riddles/",
  }
}
```

## Possible error responses

A `PUT` request to the `/riddles` endpoint may return the following errors:

Error code	Description
400	<b>Bad Request:</b> This error will be returned if a required field in the request body is missing or if the <code>category</code> field contains a space.
403	<b>Forbidden:</b> This error will be returned if you try to modify the seed data (the initial riddles added to the database).
404	<b>Not Found:</b> This error will occur if the requested riddle <code>id</code> is not found in the database.
500	<b>Internal Server Error:</b> An unexpected error occurred on the server.

[Back to usage list](#)

---

### To update a field of an existing riddle

HTTP Method	Endpoint	Summary
PATCH	/riddles	Updates individual fields of a riddle in the database

Select a Riddle object to update by setting the query parameter to its `id`. Use the JSON request body to set one or more fields to a new value. Omitted fields retain their values. Currently, updating seed data (the initial riddles added to the database) is not allowed.

The query parameter is defined as:

Parameter	Type	Required	Description
id	string	Required	The database <code>id</code> for the riddle. This is needed for <code>PUT</code> , <code>PATCH</code> , or <code>DELETE</code> requests which are operations performed on an existing riddle.

The request body should use the following schema:

Field	Type	Required	Description
riddle	string	Optional	An update to the riddle's question
answer	string	Optional	An update to the riddle's answer
category	string	Optional	An update to the classification of the riddle. It should be one-word and not contain spaces. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
source	string	Optional	An update to the source of the riddle

**Note:** The `Content-Type` header should be set to `application/json`.

## Example request

A curl request to update a field of a riddle:

```
curl -X PATCH \
  'http://guarded-eyrie-88809.herokuapp.com/riddles?id=60bd0708d7dcc31bd9376abe' \
  -H 'Content-Type: application/json' \
  -d '{
    "riddle": "There\'\'s only one word in the dictionary that\'\'s spelled wrong.
What is it?",
    "answer": "The word \'\'wrong.\'\' It\'\'s the only word that\'\'s spelled
W-R-O-N-G.",
    "category": "kids",
    "source": "https://www.prodigygame.com/main-en/blog/riddles-for-kids/"
  }'
```

A successful response will return an HTTP status code of 200 and have the following schema:

Field	Type	Description
message	string	A brief success message
content	object	Top-level containing Riddle object

Where the Riddle object has this schema:

Field	Type	Description
_id	string	The database id for the Riddle object
riddle	string	The riddle's question
answer	string	The riddle's answer
category	string	A classification of the riddle. The original database includes the categories: easy, hard, funny, kids, math, and word. This is not an enum and more can be added.
source	string	The source of the riddle

### Example response

A successful response to a `PATCH` request is an object like below:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "message": "Successfully updated riddle",
  "content": {
    "_id": "60bd0708d7dcc31bd9376abe",
    "riddle": "I'm tall when I'm young, and I'm short when I'm old. What am I?",
    "answer": "A candle",
    "category": "easy",
    "source": "https://parade.com/947956/parade/riddles/",
  }
}
```

## Possible error responses

A `PATCH` request to the `/riddles` endpoint may return the following errors:

Error code	Description
400	<code>Bad Request</code> : This error will be returned if a required field in the request body is missing or if the <code>category</code> field contains a space.
403	<code>Forbidden</code> : This error will be returned if you try to modify the seed data (the initial riddles added to the database).
404	<code>Not Found</code> : This error will occur if the requested riddle <code>id</code> is not found in the database.
500	<code>Internal Server Error</code> : An unexpected error occurred on the server.

[Top of API documentation](#)

[Top of README](#)