

Criar uma API usando um framework PHP, que atenda o seguinte formato disponível em <http://bit.ly/moblytestex> e <https://bit.ly/moblytest>, lembre-se de utilizar os status HTTP.

Framework laravel 5.8.*

Dev PhpStorm 2018.3

Postman

Utilizado o serve do laravel *'php artisan serve'*

cicero@cicero-3490:/var/www/html/laravel\$ php artisan serve

Laravel development server started: <<http://127.0.0.1:8000>>

Crud

Post Create User

<http://localhost:8000/api/v1/users>

Get List Users

<http://localhost:8000/api/v1/users>

Get List One User Ex:

<http://localhost:8000/api/v1/users/35>

Put Update User Ex:

<http://localhost:8000/api/v1/users/35>

Del Delete User

<http://localhost:8000/api/v1/users/11>

Post Create User

The screenshot shows the Postman application interface. The top bar displays the time as 19:35 and the user 'pt'. The main workspace is titled 'My Workspace' and includes an 'Invite' button. The left sidebar shows a collection named 'Mobly Test' with 5 requests, including 'Create User', 'List Users', 'List One User', 'Update User', and 'Delete User'. The 'Create User' request is selected, showing a POST method to the URL 'http://localhost:8000/api/v1/users'. The request body is a JSON object:

```
{  "username": "Antonio",  "first_name": "Vieira",  "last_name": "Silva",  "password": "abc123456"}
```

. The response is also shown in the bottom panel, indicating a successful status of 201 Created, with a time of 22 ms and a size of 338 B. The response body is a JSON object:

```
{  "success": true,  "data": 35}
```

.

Activities Postman

ter 19:35

Postman

File Edit View Help

New Import Runner

My Workspace Invite

No Environment

Filter

History Collections

Trash

Mobly Test
5 requests

Rest

POST Create User

GET List Users

GET List One User

PUT Update User

DEL Delete User

POST Create User

GET List Users

GET List One User

PUT Update User

DEL Delete User

POST

http://localhost:8000/api/v1/users

Send Save

Params Authorization Headers (1) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary JSON (application/json)

Beautify

1 {
2 "username": "Antonio",
3 "first_name": "Vieira",
4 "last_name": "Silva",
5 "password": "abc123456",
6 }

Body Cookies Headers (9) Test Results (1/3)

Status: 201 Created Time: 22 ms Size: 338 B Save Download

Pretty Raw Preview JSON

1 {
2 "success": true,
3 "data": 35
4 }

Learn

Get List Users

The screenshot shows the Postman application interface. The top bar includes the 'Activities' menu, the 'Postman' title, and system status icons. The main workspace is titled 'My Workspace' and contains a list of requests: 'POST Create User', 'GET List Users' (selected), 'GET List One User', 'PUT Update User', and 'DEL Delete User'. The 'GET List Users' request is configured with the URL 'http://localhost:8000/api/v1/users' and a 'GET' method. The response is displayed in the 'Body' tab, showing a JSON object with a 'success' flag and a list of four users. The status bar at the bottom indicates a '200 OK' response with a time of 23 ms and a size of 3.41 KB.

Activities Postman ter 19:40 pt

File Edit View Help

New Import Runner

My Workspace Invite

No Environment

Filter

History Collections

Trash

Mobly Test
5 requests

Rest

POST Create User

GET List Users

GET List One User

PUT Update User

DEL Delete User

POST Create User GET List Users GET List One User PUT Update User DEL Delete User

GET http://localhost:8000/api/v1/users

Send Save

Body Cookies Headers (9) Test Results (2/2) Status: 200 OK Time: 23 ms Size: 3.41 KB Save Download

Pretty Raw Preview JSON

```
1 {
2   "success": true,
3   "data": [
4     {
5       "id": 1,
6       "username": "fernando",
7       "first_name": "eduardo",
8       "last_name": "camargo",
9       "password": "x12345x",
10      "created_at": "2019-05-14 14:52:02",
11      "updated_at": "2019-05-14 14:52:02"
12    },
13    {
14      "id": 2,
15      "username": "maria",
16      "first_name": "adelia",
17      "last_name": "silva",
18      "password": "bola123",
19      "created_at": "2019-05-14 14:58:22",
20      "updated_at": "2019-05-14 14:58:22"
21    },
22    {
23      "id": 3,
24      "username": "lucas",
25      "first_name": "adelia",
26      "last_name": "chivas",
27      "password": "w099999",
28      "created_at": "2019-05-14 15:37:05",
29      "updated_at": "2019-05-14 15:37:05"
30    },
31    {
32      "id": 4,
33      "username": "rodrigo",
34      "first_name": "adelia",
```

Learn

Get List One User

The screenshot displays the Postman application interface. The top bar shows the time as 19:41 and the user 'pt'. The main workspace is titled 'My Workspace' and includes an 'Invite' button. The left sidebar shows a 'Collections' view with a 'Mobly Test' collection containing 5 requests. The 'List One User' request is selected, showing a GET method to the endpoint 'http://localhost:8000/api/v1/users/35'. The request body is set to 'x-www-form-urlencoded' and contains a single key-value pair: 'Key' with value 'Value'. The response is displayed in the 'Body' tab, showing a JSON object with user details. The status is '200 OK', time is '22 ms', and size is '496 B'.

Activities Postman

File Edit View Help

New Import Runner

My Workspace Invite

No Environment

Filter

History Collections

Trash

Mobly Test
5 requests

Rest

- POST Create User
- GET List Users
- GET List One User
- PUT Update User
- DEL Delete User

POST Create User GET List Users GET List One User PUT Update User DEL Delete User

Examples (0)

GET http://localhost:8000/api/v1/users/35 Send Save

Params Authorization Headers (1) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (9) Test Results (3/7) Status: 200 OK Time: 22 ms Size: 496 B Save Download

Pretty Raw Preview JSON

```
1 {
2   "success": true,
3   "data": {
4     "id": 35,
5     "username": "Antonio",
6     "first_name": "Vieira",
7     "last_name": "Silva",
8     "password": "abc123456",
9     "created_at": "2019-05-14 22:35:50",
10    "updated_at": "2019-05-14 22:35:50",
11  }
12 }
```

Learn

Put Update User

The image shows the Postman application interface. The top bar displays the time as 19:42 and the user as 'pt'. The main workspace is titled 'My Workspace' and includes an 'Invite' button. The left sidebar shows a 'Collections' view with a 'Mobly Test' collection containing 5 requests. The 'Update User' request is selected, showing a PUT method to the endpoint 'http://localhost:8000/api/v1/users/35'. The request body is a JSON object with the following fields: 'username' (cicero vieira da silva), 'first_name' (vieira da silva), 'last_name' (araujo), and 'password' (x!21@087YY). The response is also shown in the bottom panel, indicating a successful update with a status of 200 OK, a time of 23 ms, and a size of 335 B. The response body is a JSON object with 'success' (true) and 'data' (true).

Activities Postman

File Edit View Help

New Import Runner

My Workspace Invite

No Environment

Filter

History Collections

Trash

Mobly Test
5 requests

Rest

POST Create User

GET List Users

GET List One User

PUT Update User

DEL Delete User

Update User

Examples (0)

PUT http://localhost:8000/api/v1/users/35

Send Save

Params Authorization Headers (1) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary JSON (application/json)

Beautify

```
1 {
2   "username": "cicero vieira da silva",
3   "first_name": "vieira da silva",
4   "last_name": "araujo",
5   "password": "x!21@087YY"
6 }
```

Body Cookies Headers (9) Test Results (1/2)

Status: 200 OK Time: 23 ms Size: 335 B

Save Download

Pretty Raw Preview JSON

```
1 {
2   "success": true,
3   "data": true
4 }
```

Learn

Del Delete User

The screenshot shows the Postman application interface. The top bar includes the 'Activities' menu, the 'Postman' logo, and system icons for time (19:46), network, and volume. The main interface is divided into several sections:

- Left Sidebar:** Contains a 'File Edit View Help' menu, a 'New' button, and a 'My Workspace' dropdown. Below this is a 'Collections' tab with a search filter and a list of requests under 'Mobly Test' (5 requests). The 'Delete User' request is selected.
- Request Editor:** Shows the 'DELETE' method and the URL 'http://localhost:8000/api/v1/users/11'. The 'Body' tab is active, showing 'x-www-form-urlencoded' as the content type. A table with 'KEY' and 'VALUE' columns is visible, with 'Key' and 'Value' as the first entry.
- Response Section:** Displays the status '200 OK', time '24 ms', and size '323 B'. The response body is shown in 'Pretty' format as JSON:

```
{  "success": true}
```
- Bottom Bar:** Includes a 'Learn' link and a help icon.

Not Found

The screenshot shows the Postman application interface. The top bar includes the 'Activities' menu, the 'Postman' logo, and system status icons. The main menu bar contains 'File', 'Edit', 'View', and 'Help'. Below this is a toolbar with 'New', 'Import', 'Runner', and 'My Workspace' (with an 'Invite' button). The left sidebar shows a 'Collections' view with a 'Trash' section and a 'Mobly Test' collection containing 5 requests: 'Create User', 'List Users', 'List One User' (selected), 'Update User', and 'Delete User'. The main workspace displays the details for the 'List One User' request, which is a GET request to 'http://localhost:8000/api/v1/users/11'. The request is configured with 'x-www-form-urlencoded' body type and a 'Key' header with value 'Value'. The response status is '404 Not Found' with a time of 56 ms and size of 354 B. The response body is shown in JSON format:

```
{  "success": false,  "message": "Not Found!"}
```

Activities Postman ter 19:47

Postman

File Edit View Help

New Import Runner My Workspace Invite

Filter

History Collections

Trash

Mobly Test
5 requests

Rest

- POST Create User
- GET List Users
- GET List One User
- PUT Update User
- DEL Delete User

POST Create User GET List Users GET List One User PUT Update User DEL Delete User

List One User

GET http://localhost:8000/api/v1/users/11

Send Save

Params Authorization Headers (1) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (9) Test Results (0/7) Status: 404 Not Found Time: 56 ms Size: 354 B Save Download

Pretty Raw Preview JSON

```
1 {
2   "success": false,
3   "message": "Not Found!"
4 }
```

Learn

Database

Activities MySQL Workbench ter 19:52 MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Management Schemas

SCHEMAS

Filter objects

- creditoapp
- mt4
- sys
- usuarios**
- voyager

Query 2 usuarios

1 • SELECT * FROM usuarios.usuarios;

Result Grid

#	id	username	first_name	last_name	password	created_at	updated_at
9	9	lucasvieixxr	adelia	chivas	w0999xx666x99	2019-05-14 18:56:17	2019-05-14 21:25:43
10	10	lrfafasfgasf	adelia	chivas	w0999xx666x99	2019-05-14 18:56:58	2019-05-14 22:21:46
11	12	adriana	lima	chivas	345435	2019-05-14 20:09:34	2019-05-14 20:09:34
12	13	laura	adelia	silva	553511	2019-05-14 20:21:29	2019-05-14 20:21:29
13	14	tereza	ferreira	chivas	534645	2019-05-14 20:57:21	2019-05-14 20:57:21
14	15	laura	adelia	souza	4234yy	2019-05-14 21:04:03	2019-05-14 21:04:03
15	16	laura212	aparecida	silva	4234yy	2019-05-14 21:20:40	2019-05-14 21:20:40
16	33	rogerio	andrade	chivas	8685223	2019-05-14 22:22:02	2019-05-14 22:22:02
17	34	cicero vieira da silva	adelia	chivas	1234567890	2019-05-14 22:23:52	2019-05-14 22:24:25
18	35	cicero vieira da silva	vieira da silva	araujo	x121@087YY	2019-05-14 22:35:50	2019-05-14 22:42:30
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

usuarios 7

Object Info Session

Schema: usuarios

Active schema changed to usuarios

Apply Revert