
event-manager

Mar 01, 2020

1	Getting Started with Event Manager	3
1.1	Introduction	3
1.2	Installation	3
1.3	Connect to Database	4
1.4	Make The Project Online	4
2	Authentication	5
2.1	Register	5
2.2	Login	5
3	User	7
3.1	User Data	7
3.2	Edit User	7
4	Events	9
4.1	Create Event	9
4.2	Update Event	9
4.3	Delete Event	10
4.4	Events List	10
4.5	Show Event	10
4.6	Show Event Invitations	11
5	Invitations	13
5.1	Send Invitation	13
5.2	Sended Invitations	14
5.3	Received Invitations	14
5.4	Answer an Invitation	14
6	Tests	17
6.1	Run Tests	17
7	Work With Models	19
7.1	User	19
7.2	Events	19
7.3	Invitation	20

Event Manager is powerful System for Manage Your Events

Getting Started with Event Manager

this is a documentation of Event Manager System. it's help you to work with Event Manager system. first you need to know what's this system start from Introduction

1.1 Introduction

this a Laravel Project for using this project you should have knowledge of laravel applications

Event Manager is a api base system , it's help you to build and manage your events and your guests

1.2 Installation

for install this project in your system you should install PHP , Mysql and Composer in your system.

first you should clone this project in your computer , use this command:

```
git clone https://github.com/mohammad76/event-manager.git
```

then you need go to the project folder:

```
cd event-manager
```

now you should install composer packages with this command:

```
composer install
```

in the next step you need to generate app key with this command:

```
php artisan key:generate
```

now we need jwt secret key you can build that with this command:

```
php artisan jwt:secret
```

ok, project is installed successfully but it's need a couple of steps to get ready.

1.3 Connect to Database

Event Manager system work with mysql database and you should config that , here we have some help to config that first you should build database in your mysql and put that in `.env` file in project directory, like this:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=event-manager
DB_USERNAME=root
DB_PASSWORD=123
```

Note:

- `DB_PASSWORD` can be null
 - `DB_CONNECTION` and `DB_HOST` and `DB_PORT` usually don't need to change
-

after that you need to migrate database and the default data , you can use this command:

```
php artisan migrate --seed
```

ok now Event Manager database is ready to use.

1.4 Make The Project Online

if you complete previous steps correctly project should be ready to get online with this command:

```
php artisan serv
```

you can see other sections of documentation to get more familiar with project.

CHAPTER 2

Authentication

Authentication is a first step to use Event Manager Api , if you want to use any section of this system you should first register or login to the system. Event Manager system use JWT as Authentication Method , we Believe it's a better way to Authenticate Users of our system.

2.1 Register

for register in Event Manager system you can use this route:

```
[POST] http://site.test/api/v1/register
```

as you know you should pass some data to this route, this is what you should pass to the route.

```
{
  "name" : "mohammad aliabadi",
  "mobile" : "09303030300",
  "email" : "mohammad@mail.com",
  "password" : "123456789"
}
```

Note: all data are required and you should pass them.

Warning: in data you pass unique mobile , email and minimum length of password is 6 character.

2.2 Login

for login to the Event Manager system you should have registered before, after registering to the system you can login. for login you can use this route:

```
[POST] http://site.test/api/v1/login
```

of course for login to the system you should have username and password.

username field can be a email or mobile it depends on you with what you want to login , at the end it doesn't different what you choose.

```
{  
  "username" : "09303030300",  
  "password" : "123456789"  
}
```

Note: all data are required and you should pass them.

Warning: in username you should pass a valid email address or mobile number.

Users are base of our api system, because our system only work for logged user.
simply people should login or register to work with Event Manager system.

3.1 User Data

if you want see logged in user data you can use this route:

```
[GET] http://site.test/api/v1/user
```

3.2 Edit User

for updating user data you can use this route:

```
[POST] http://site.test/api/v1/user
```

update user route need some data you should pass to , this is data body

```
{
  "_method" : "PUT",
  "name" : "New Name",
  "password" : "123456789",
  "avatar" : "file",
}
```

Note:

- `_method` use for change request method, we can not upload avatar image thorough PUT method for this problem we should manipulation our system

event-manager

- for `avatar` field you should pass a image file
-

Warning: all fields are optionals and you can pass just what you want to update

CHAPTER 4

Events

for using Events Api first you should be logged in with a user for login you can see [Login Documentation](#)

4.1 Create Event

for creating Event you can use this route:

```
[POST] http://site.test/api/v1/events
```

for creating you should pass these arguments to the route

```
{
  "name": "my birthday",
  "description": "this is my 24 birthday"
}
```

Note: name and description both required and if you don't pass api give you a validation error

4.2 Update Event

for update an Event you can use this route:

```
[PUT] http://site.test/api/v1/events/{{event_id}}
```

Note: instead of {{event_id}} you should put event id you want to edit

Warning: just know you only can update own events and if you want to request for update other events api get a 403 error

as you know for update an event you should pass some data to the api this is what you should pass that

```
{
  "name": "New Name",
  "description": "New Description"
}
```

Note: name and description not required and you can pass just one of them

4.3 Delete Event

if you want delete a event you can use this route

```
[DELETE] http://site.test/api/v1/events/{{event_id}}
```

Note: instead of {{event_id}} you should put event id you want to delete

Warning: just know you only can delete own events and if you want to request for delete other events api get a 403 error

4.4 Events List

for viewing events list you can use this route:

```
[GET] http://site.test/api/v1/events?type=all
```

Note:

- you only see events if you are creator of that or you are a guest of that events
 - type parameter not required you can delete that, default type is all
 - instead of all you can use creator and guest or you can delete type parameter
-

4.5 Show Event

for viewing a single event you can use this route:

```
[GET] http://site.test/api/v1/events/{{event_id}}
```

Note: instead of `{{event_id}}` you should put event id you want to see

4.6 Show Event Invitations

if you want to see invitations related to a event you should use this route:

```
[GET] http://site.test/api/v1/events/{{event_id}}/invitations
```

Note: instead of `{{event_id}}` you should put event id you want to see

CHAPTER 5

Invitations

for using Invitations Api first you should be logged in with a user for login you can see [Login Documentation](#)

5.1 Send Invitation

for send invitation to other users you can use this route:

```
[POST] http://site.test/api/v1/{{event_id}}/send-invitations
```

Note: instead of {{event_id}} you should put event id you want invite to.

Send Invitation need some data to invite other users , you can invite other users to gather, like this:

```
{
  "invitations": ["mohammad_m69@yahoo.com", "09352864812" , "ali@yahoo.com" , "09303030300
  ↪" , "09332114546" , "09332131456" , "hasan@yahoo.com" , "sd@sadsad.com"]
}
```

Note:

- you only can invite other who is one of our users.
 - attention invitation to not one of our user ignored and we will show you these are in this api response.
 - you can invite other users by they email or mobile
 - attention you only can invite other user who they aren't one of your event member or they aaren't rejected the last invitation.
-

Warning: the `invitations` field required and you should pass an array to that

5.2 Sended Invitations

for see Sended Invitations before in Event Manager system you can use this route:

```
[GET] http://site.test/api/v1/sended-invitations?status=all
```

Note:

- `status` parameter not required you can delete that, default status is `all`
 - instead of `all` you can use `accepted`, `rejected`, `pending` or you can delete status parameter
-

5.3 Received Invitations

for see Received Invitations from other users in Event Manager system you can use this route:

```
[GET] http://site.test/api/v1/received-invitations?status=all
```

Note:

- `status` parameter not required you can delete that, default status is `all`
 - instead of `all` you can use `accepted`, `rejected`, `pending` or you can delete status parameter
-

5.4 Answer an Invitation

for answer a received invitation you can use this route:

```
[PATCH] http://site.test/api/v1/invitations/{{invitation_id}}
```

Note: instead of `{{invitation_id}}` you should put invitation id you want answer to that.

you should define data to answer an invitation , like this:

```
{  
  "status": "accepted",  
}
```

Note: you only can accept or reject an invitation, for that you can put `accepted` or `rejected` in status field.

Warning: the `status` field is required and you should put one of `accepted` or `rejected` value in there.

we build some tests for our project to get safe and workable api for you.

6.1 Run Tests

for run this tests you can use this command:

```
vendor\bin\phpunit
```


CHAPTER 7

Work With Models

we build some method for you to make your job easy , lets go to see these

7.1 User

First of all our models in `app/Models` directory, it's better for develop.

get events who I created:

```
$user->myEvents();
```

get events who I invited:

```
$user->invitedEvents();
```

get invitations who get from other users:

```
$user->received_invitations();
```

get invitations you send to other users:

```
$user->send_invitations();
```

7.2 Events

get if giving user created this event

```
$event->isCreator($user);
```

get if giving user invited to this event

```
$event->isInvited($user);
```

get if giving user member of this event

```
$event->isMember($user);
```

get event sended invitations

```
$event->invitations();
```

get event members (they are accepted a invitation)

```
$event->members();
```

7.3 Invitation

get invitation related event

```
$event->event();
```

get invitation invitor user

```
$event->invitor();
```

get invitation invited user

```
$event->invited();
```