

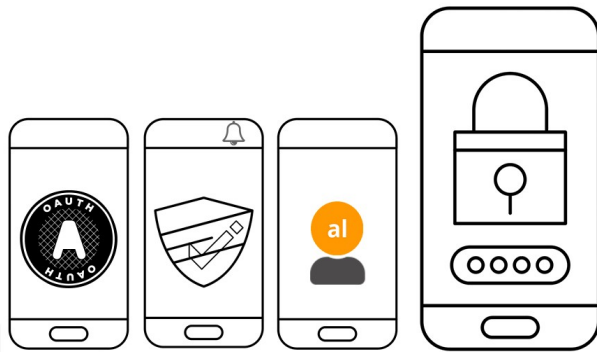
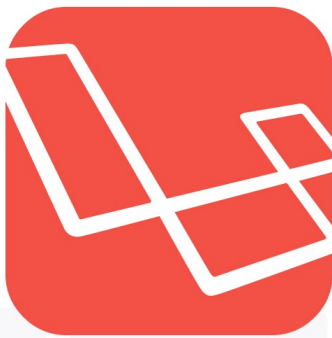
API Rest with Laravel 5.6 Passport Authentication — Reset Password (Part 4)



Alfredo Barron

Aug 2, 2018 · 6 min read

We will learn to create a password reset system



API Rest with Laravel 5.6 Passport
Authentication-Reset Password
(Part 4)

<https://www.vecteezy.com>

Step 1. Update migration

In first step, we require to update the migration file

database/migrations/xxx_create_password_resets_table.php like bellow code:

```
public function up()
{
    Schema::create('password_resets', function (Blueprint $table) {
        $table->increments('id');
    });
}
```

```

        $table->string('email')->index();
        $table->string('token');
        $table->timestamps();
    });
}

```

Step 2. Create PasswordReset model

Open your terminal or command prompt and run bellow command:

```
php artisan make:model PasswordReset
```

This command will create `app/PasswordReset.php` file, in this file set fillable inputs.

```

class PasswordReset extends Model
{
    protected $fillable = [
        'email', 'token'
    ];
}

```

Step 3. Create Notifications

We create two notifications `PasswordResetRequest` and `PasswordResetSuccess`, in your terminal or command prompt run bellow commands:

```
php artisan make:notification PasswordResetRequest
```

```
php artisan make:notification PasswordResetSuccess
```

This command will create `app/Notifications/PasswordResetRequest.php` and `app/Notifications/PasswordResetSuccess.php` files.

In the `PasswordResetRequest.php` file add the next code:

```

<?php

namespace App\Notifications;

```

```

use Illuminate\Bus\Queueable;
use Illuminate\Notifications\Notification;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Notifications\Messages\MailMessage;

class PasswordResetRequest extends Notification implements
ShouldQueue
{
    use Queueable;

    protected $token;

    /**
     * Create a new notification instance.
     *
     * @return void
     */
    public function __construct($token)
    {
        $this->token = $token;
    }

    /**
     * Get the notification's delivery channels.
     *
     * @param mixed $notifiable
     * @return array
     */
    public function via($notifiable)
    {
        return ['mail'];
    }

    /**
     * Get the mail representation of the notification.
     *
     * @param mixed $notifiable
     * @return \Illuminate\Notifications\Messages\MailMessage
     */
    public function toMail($notifiable)
    {
        $url = url('/api/password/find/' . $this->token);

        return (new MailMessage)
            ->line('You are receiving this email because we
received a password reset request for your account.')
            ->action('Reset Password', url($url))
            ->line('If you did not request a password reset, no
further action is required.');
    }

    /**
     * Get the array representation of the notification.
     *
     * @param mixed $notifiable
     * @return array
     */

```

```

*/
public function toArray($notifiable)
{
    return [
        //
    ];
}
}

```

In the `PasswordResetSuccess.php` file add the next code:

```

<?php

namespace App\Notifications;

use Illuminate\Bus\Queueable;
use Illuminate\Notifications\Notification;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Notifications\Messages\MailMessage;

class PasswordResetSuccess extends Notification implements
ShouldQueue
{
    use Queueable;

    /**
     * Create a new notification instance.
     *
     * @return void
     */
    public function __construct()
    {
        //
    }

    /**
     * Get the notification's delivery channels.
     *
     * @param mixed $notifiable
     * @return array
     */
    public function via($notifiable)
    {
        return ['mail'];
    }

    /**
     * Get the mail representation of the notification.
     *
     * @param mixed $notifiable
     * @return \Illuminate\Notifications\Messages\MailMessage
     */
    public function toMail($notifiable)

```

```

    {
        return (new MailMessage)
            ->line('You are changed your password successful.')
            ->line('If you did change password, no further action is
required. ');
            ->line('If you did not change password, protect your
account. ');
    }

/**
 * Get the array representation of the notification.
 *
 * @param mixed $notifiable
 * @return array
 */
public function toArray($notifiable)
{
    return [
        //
    ];
}
}

```

Step 4. Create API Routes

We will create api routes. Laravel provide `routes/api.php` file for write web services route. So, let's add new route on that file.

```

<?php

use Illuminate\Http\Request;

...

Route::group([
    'namespace' => 'Auth',
    'middleware' => 'api',
    'prefix' => 'password'
], function () {
    Route::post('create', 'PasswordResetController@create');
    Route::get('find/{token}', 'PasswordResetController@find');
    Route::post('reset', 'PasswordResetController@reset');
});

```

Step 5: Install Dependencies

We use Carbon package to help with dates, in your terminal run bellow command:

```
composer require nesbot/carbon
```

Step 6: Create Controller

In this step we have to create new controller and three api method. So let's create

`PasswordResetController` and put bellow code:

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Carbon\Carbon;
use App\Notifications>PasswordResetRequest;
use App\Notifications>PasswordResetSuccess;
use App\User;
use App>PasswordReset;

class PasswordResetController extends Controller
{
    /**
     * Create token password reset
     *
     * @param [string] email
     * @return [string] message
     */
    public function create(Request $request)
    {
        $request->validate([
            'email' => 'required|string|email',
        ]);

        $user = User::where('email', $request->email)->first();

        if (!$user)
            return response()->json([
                'message' => 'We can't find a user with that e-mail
address.'
            ], 404);

        $passwordReset = PasswordReset::updateOrCreate(
            ['email' => $user->email],
            [
                'email' => $user->email,
                'token' => str_random(60)
            ]
        );
    }
}
```

```

    if ($user && $passwordReset)
        $user->notify(
            new PasswordResetRequest($passwordReset->token)
        );

    return response()->json([
        'message' => 'We have e-mailed your password reset
link!'
    ]);
}

/**
 * Find token password reset
 *
 * @param [string] $token
 * @return [string] message
 * @return [json] passwordReset object
 */
public function find($token)
{
    $passwordReset = PasswordReset::where('token', $token)
        ->first();

    if (!$passwordReset)
        return response()->json([
            'message' => 'This password reset token is invalid.'
        ], 404);

    if (Carbon::parse($passwordReset->updated_at)-
>addMinutes(720)->isPast()) {
        $passwordReset->delete();
        return response()->json([
            'message' => 'This password reset token is invalid.'
        ], 404);
    }

    return response()->json($passwordReset);
}

/**
 * Reset password
 *
 * @param [string] email
 * @param [string] password
 * @param [string] password_confirmation
 * @param [string] token
 * @return [string] message
 * @return [json] user object
 */
public function reset(Request $request)
{
    $request->validate([
        'email' => 'required|string|email',
        'password' => 'required|string|confirmed',
        'token' => 'required|string'
    ]);
}

```

```

$passwordReset = PasswordReset::where([
    ['token', $request->token],
    ['email', $request->email]
])->first();

if (!$passwordReset)
    return response()->json([
        'message' => 'This password reset token is invalid.'
    ], 404);

$user = User::where('email', $passwordReset->email)-
>first();

if (!$user)
    return response()->json([
        'message' => 'We can't find a user with that e-mail
address.'
    ], 404);

$user->password = bcrypt($request->password);
$user->save();

$passwordReset->delete();

$user->notify(new PasswordResetSuccess($passwordReset));

return response()->json($user);
}
}

```

Now we are ready to run our example so run bellow command to quick run:

```
php artisan serve
```

. . .

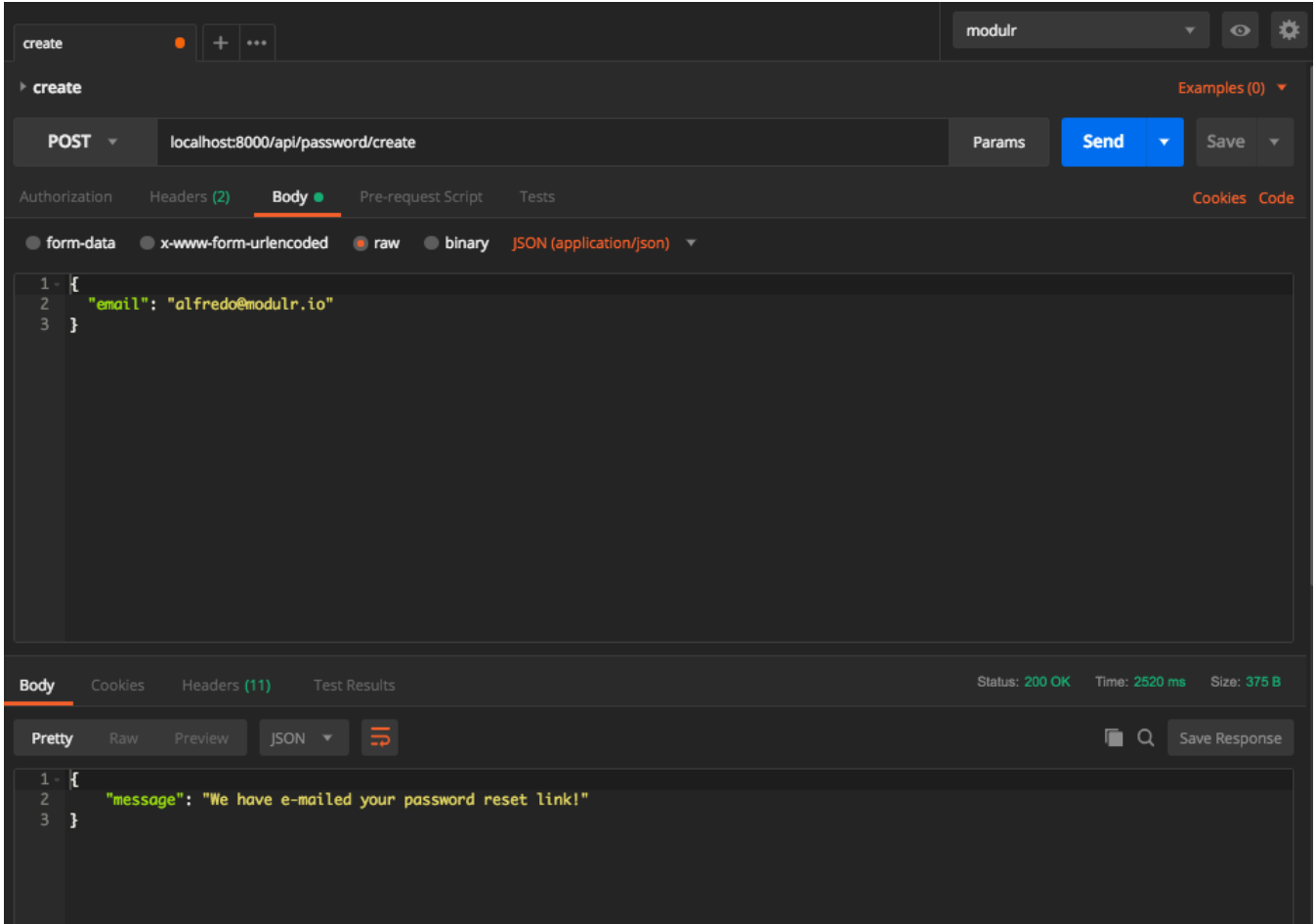
Tests

Now, we can simple test by rest client tools (Postman), So I test it and you can see below screenshots.

In this api you have to set two header as listed below:

```
Content-Type: application/json  
X-Requested-With: XMLHttpRequest
```

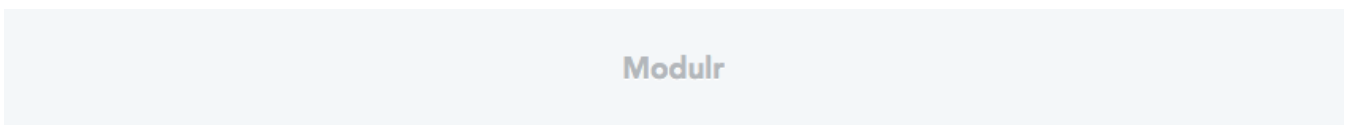
Request Password Reset



When request password reset will created token in data base

id	email	token	created_at	updated_at
1	alfredo@modulr.io	Fhjd5anbinF9SvBa1FOofwpRRR7e3VPSnt6pSjk1fjUEbitcVkrQj8Mh97w4	2018-08-02 04:05:20	2018-08-02 04:05:20

When request password reset will receive a email with the link to Reset Password.



Hello!

You are receiving this email because we received a password reset re-

quest for your account.

Reset Password

If you did not request a password reset, no further action is required.

Regards,
Modulr

If you're having trouble clicking the "Reset Password" button, copy and paste the URL below into your web browser: <http://api-laravel-passport.test/api/password/find/FhJd5anbInF9SvBa1FOofwpRRR7e3VPSNt6pSjk1fJUEb1tcVkJ8Mh97w4>

© 2018 Modulr. All rights reserved.

Find Password Reset

If the token is active response with token data

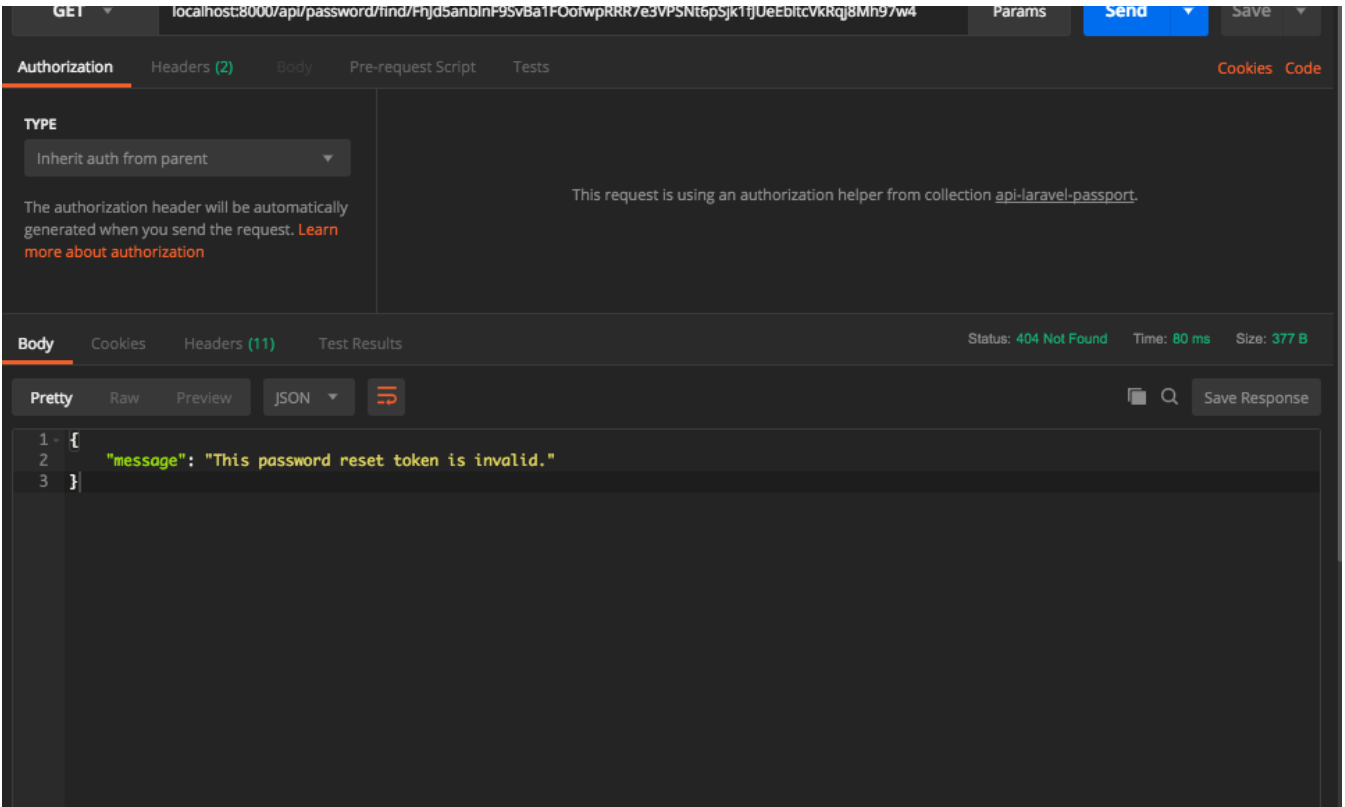
The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/password/find/FhJd5anbInF9SvBa1FOofwpRRR7e3VPSNt6pSjk1fJUEb1tcVkJ8Mh97w4
- Status:** 200 OK
- Time:** 110 ms
- Size:** 496 B
- Body (JSON):**

```
1 - {
2   "id": 1,
3   "email": "alfredo@modulr.io",
4   "token": "FhJd5anbInF9SvBa1FOofwpRRR7e3VPSNt6pSjk1fJUEb1tcVkJ8Mh97w4",
5   "created_at": "2018-08-02 04:05:20",
6   "updated_at": "2018-08-02 04:05:20"
7 }
```

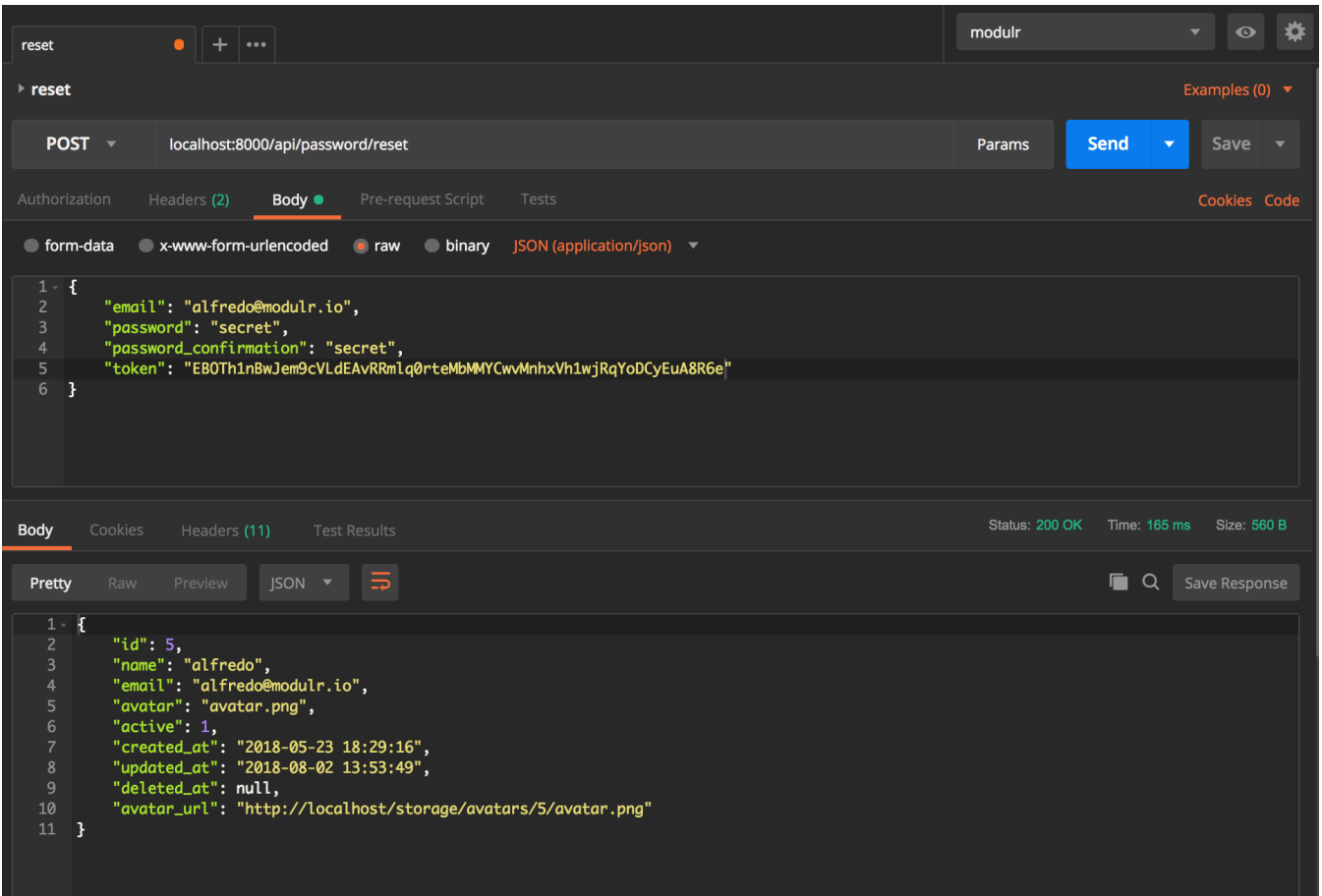
If the token is not active the response is a 404 error

The screenshot shows the top part of a REST client interface, including the method (GET) and the URL, but the response body is not visible.

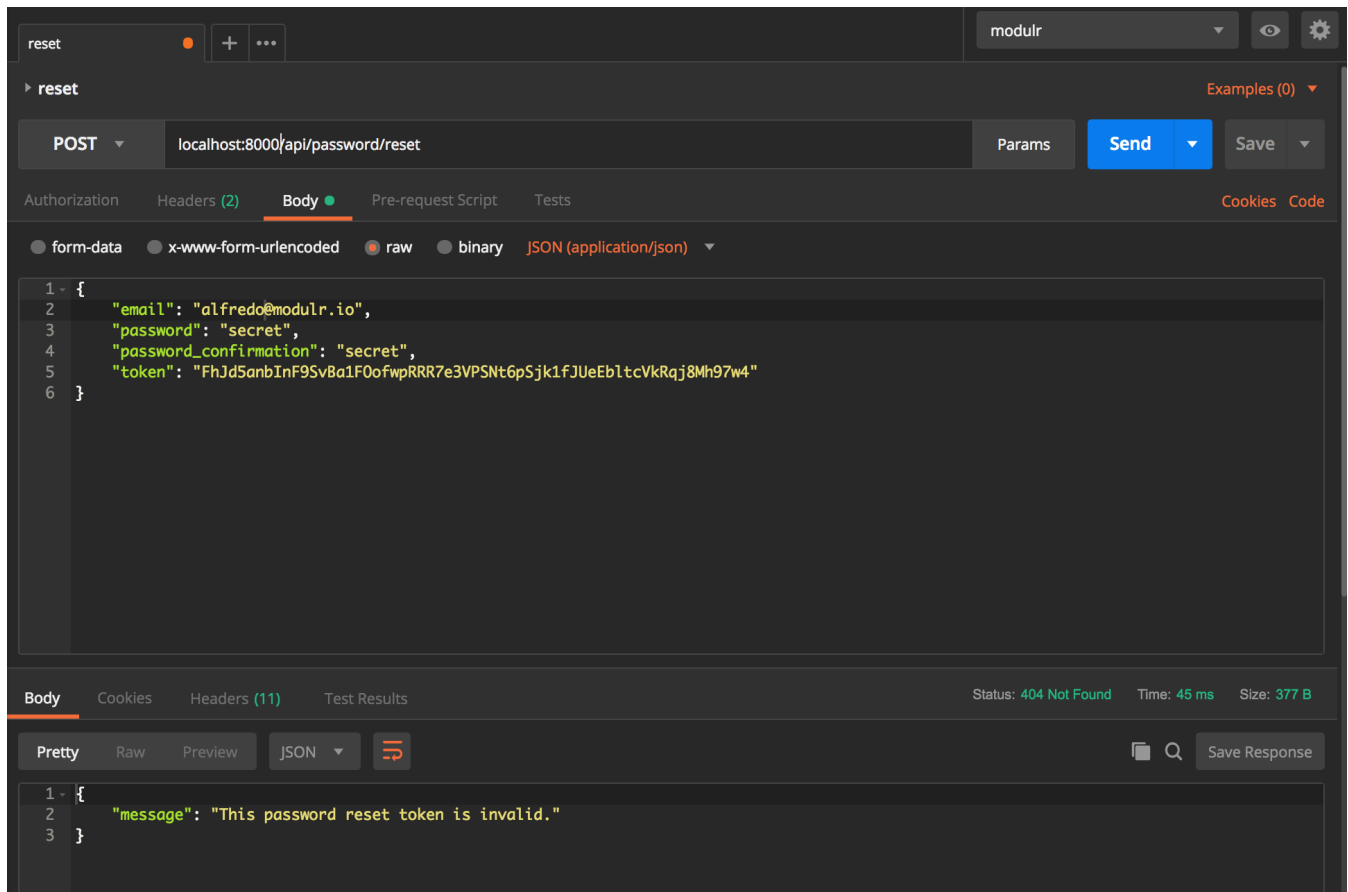


Reset Password

If the token is active response with User data



If the token is not active the response is a 404 error



The screenshot shows a Postman interface for a REST client. The request is a POST to `localhost:8000/api/password/reset`. The body is a JSON object with the following fields:

```
1 - {
2   "email": "alfredo@modulr.io",
3   "password": "secret",
4   "password_confirmation": "secret",
5   "token": "FhJd5anbInF9SvBa1F0ofwpRRR7e3VPSNt6pSjk1fJUeEb1tcVkrqj8Mh97w4"
6 }
```

The response status is `404 Not Found`. The response body is a JSON object with the following field:

```
1 - {
2   "message": "This password reset token is invalid."
3 }
```

• • •

Thanks for reading! I'm Alfredo Barrón, Feel free to connect with me via Twitter.

Part 1. Passport Authentication

Part 2. Confirm account + notifications

Part 3. Generate avatar

Part 4. Reset Password

Part 5. Send Notifications with Queues on Redis

Resources

GitHub

Postman collections

)

