Sistema de Autenticación API Rest con Laravel 5.6 — Parte 4

Claudio Vallejo
Dec 26, 2018 · 7 min read

* Restablecer Contraseña *

Seguimos la parte 4 del Sistema de Autenticación API Rest con Laravel 5.6 con el sistema para **restablecer la contraseña.** Esta es una traducción al español, con algunas adiciones y complementaciones, más algunos pequeños ajustes del artículo escrito por Alfredo Barron (a quién recomiendo seguir) y no corresponde a un tutorial de mi autoría.

. . .

1. Actualizar la migración

El primer paso requerirá una actualización de la tabla `password_resets` , para ello deberemos crear una nueva migración que incorpore los cambios que necesitamos adicionar a dicha tabla.

Debemos escribir en la terminal el siguiente comando que nos generará una nueva migración:

```
php artisan make:migration alter_create_password_resets_table --
table=password_resets
```

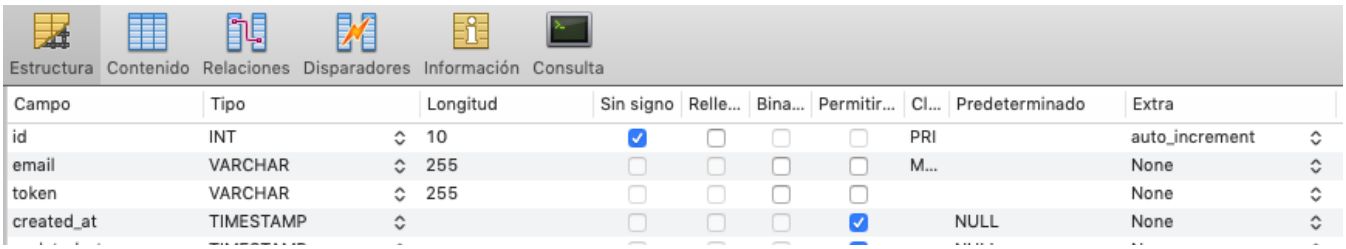
Luego, deberemos agregar lo siguiente en el archivo que acabamos de crear `database/migrations/xxxx_alter_create_password_resets_table.php`

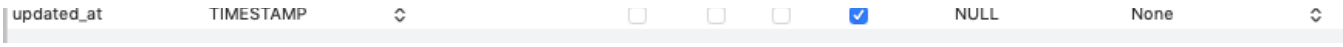
```
public function up()
{
    Schema::table('password_resets', function (Blueprint $table){
       $table->increments('id')->first();
       $table->timestamp('updated_at')->nullable();
    });
}

public function down()
{
    Schema::table('password_resets', function (Blueprint $table) {
        $table->dropColumn('id');
        $table->dropColumn('updated_at');
    });
}
```

Por último, debemos correr la migración en nuestra base de datos con el comando:

```
php artisan migrate
```

En la base de datos la nueva estructura debiera ser así:





2. Crear el modelo PasswordReset

Abrir la terminar y ejecutar el siguiente comando para crear el modelo:

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

Este comando creará un archivo en `app/PasswordReset.php` donde debemos incluir los campos que permitiremos sean escritos *'fillable'* en nuestra base de datos

```
...

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

3. Crear las Notificaciones

Debemos crear dos notificaciones, *'PaswordResetRequest'* y *'PasswordResetSuccess'*. Para ello hay que escribir en la terminal los siguientes comandos

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

Estos comandos crearan los archivos `app/Notifications/PasswordResetRequest.php` y `app/Notifications/PasswordResetSuccess.php`.

En el archivo `PasswordResetRequest.php` debemos agregar el siguiente código:

```
<?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 [
                //
            ];
        }
    }
```

Y en el archivo `PasswordResetSuccess.php` , hay que agregar el siguiente código:

```
<?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 succeful.')
            ->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 [
            //
        ];
    }
}
```

4. Crear las rutas API

Debemos crear las rutas API; para ello Laravel nos provee el archivo `routes/api.php` que nos permite escribir las rutas para los servicios web. Agreguemos entonces la siguiente ruta a nuestro archivo.

```
<?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');
});
```

5. Crear el Controlador

Debemos crear un nuevo controlador y tres métodos API. Creemos el controlador `PasswordResetController` y adicionemos el siguiente código:

a. Crear controlador en la ruta `app/Http/Controllers/Auth/` :

```
php artisan make:controller Auth\\PasswordResetController
```

b. Adicionar el siguiente código:

```
<?php

namespace App\Http\Controllers\Auth;

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

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', $request-
>token)->first();
            if (!$passwordReset)
                return response()->json([
                    'message' => 'This password reset token is invalid.'
                ], 404);
            $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);
            $user->password = bcrypt($request->password);
            $user->save();
            $passwordReset->delete();
```

```
            $user->notify(new PasswordResetSuccess($passwordReset));
            return response()->json($user);
        }
    }
```

> *Desde este punto en adelante, tu aplicación está plenamente funcional y operativa. Solo necesitas correr tu sitio en el ambiente local que tengas (homestead, valet, single server o cualquier otro entorno). En el caso de un servidor simple o directo, basta con escribir el comando:*

```
php artisan serve
```

. . .

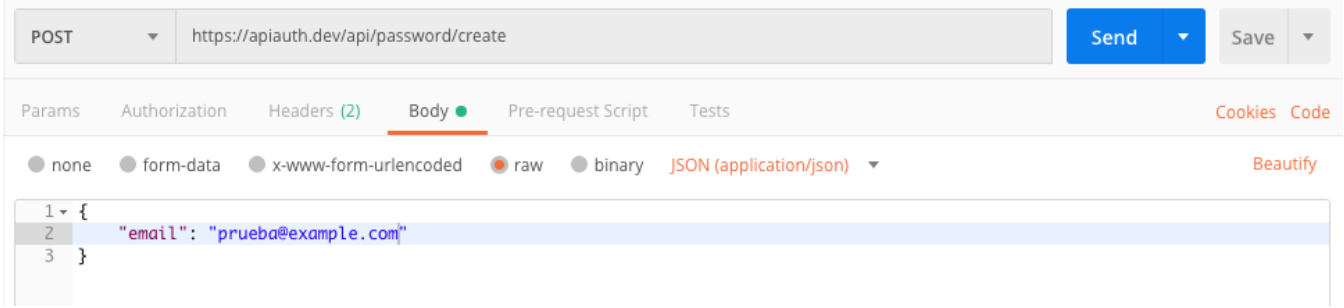
6. Pruebas

Para las pruebas, utilicé Postman (Postman)

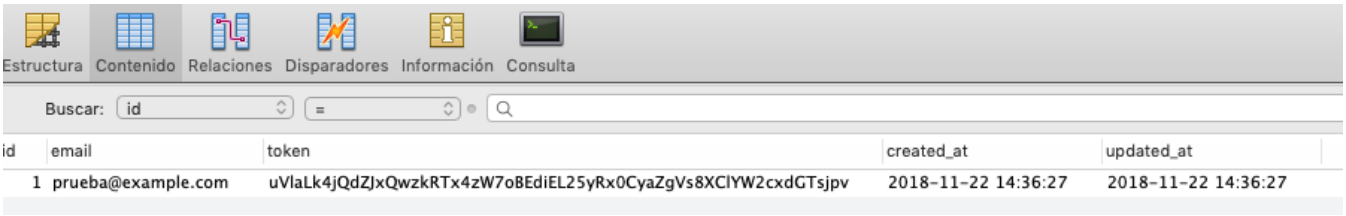
Para la correcta utilización, hay que configurar las siguientes dos cabeceras:

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

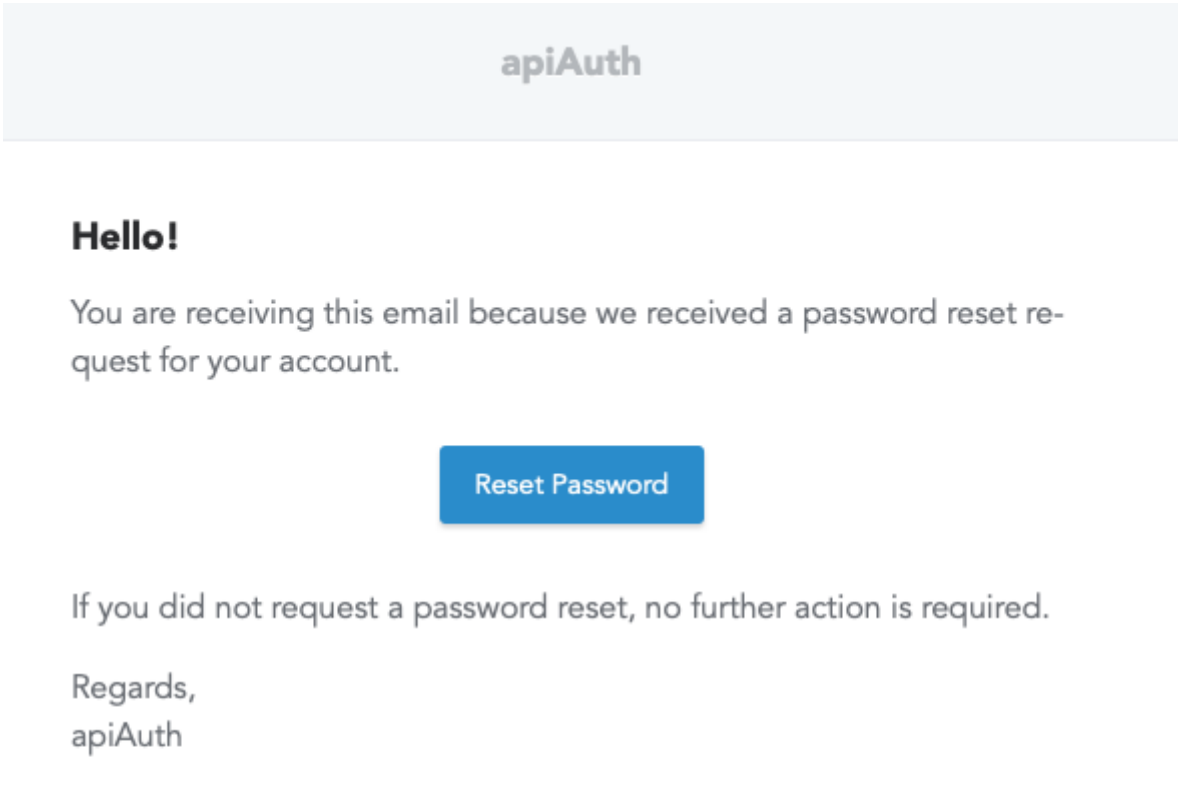
Petición de Restablecer Contraseña



Cuando la petición de restablecer contraseña se solicita, es creado el token en la base de datos



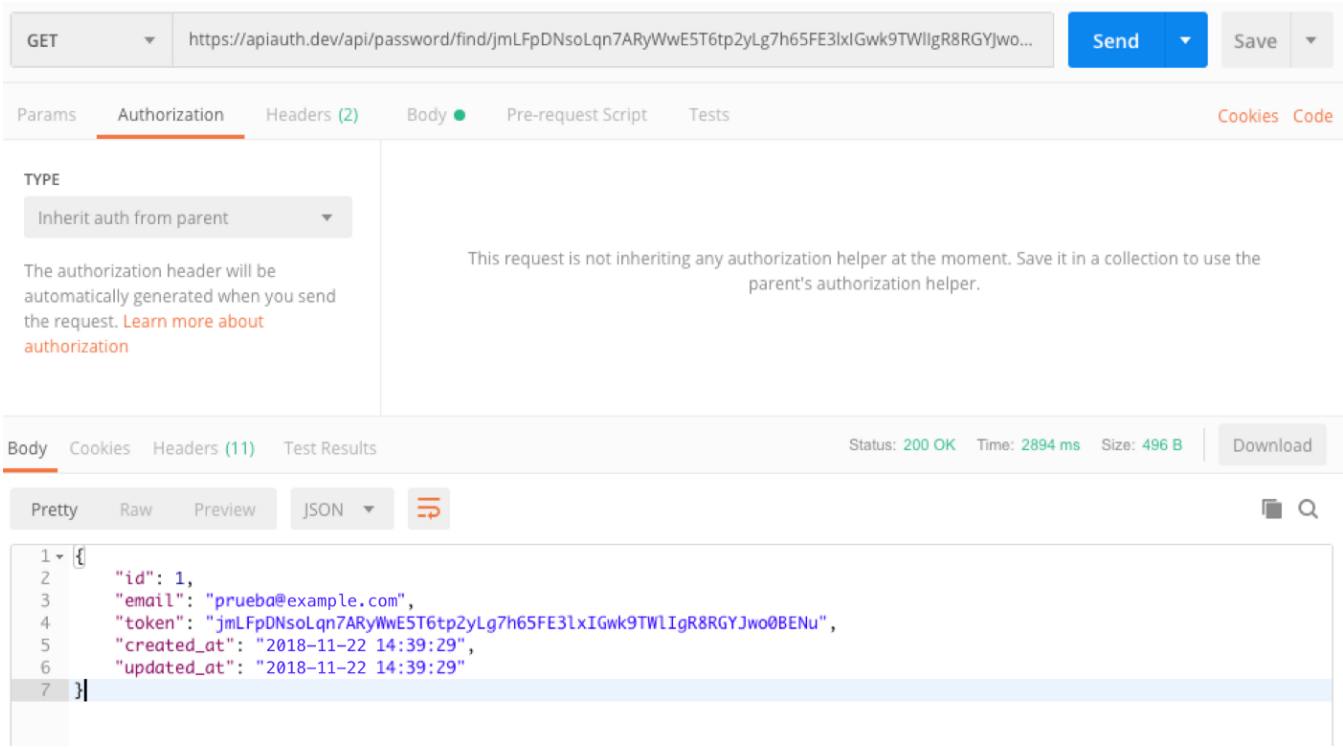
Cuando la petición de restablecer contraseña se solicita, se recibe un correo electrónico con el enlace para Restablecer Contraseña.



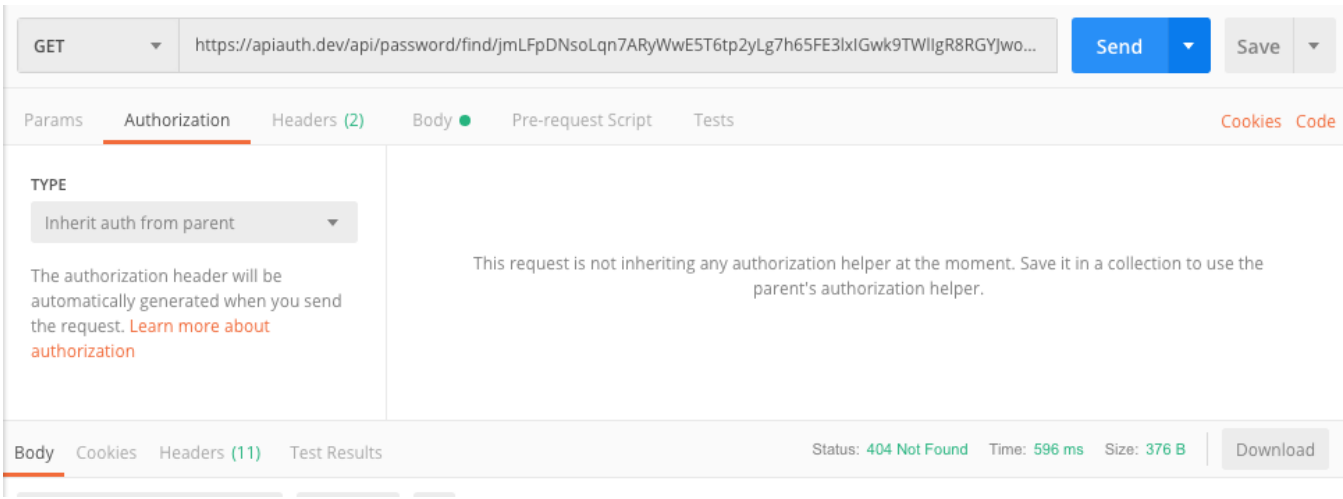
If you're having trouble clicking the "Reset Password" button, copy and paste the URL below into your web browser:
https://apiauth.dev/api/password/find/uVlaLk4jQdZJxQwzkRTx4zW7oBEdiEL25yRx0CyaZgV-s8XClYW2cxdGTsjpv

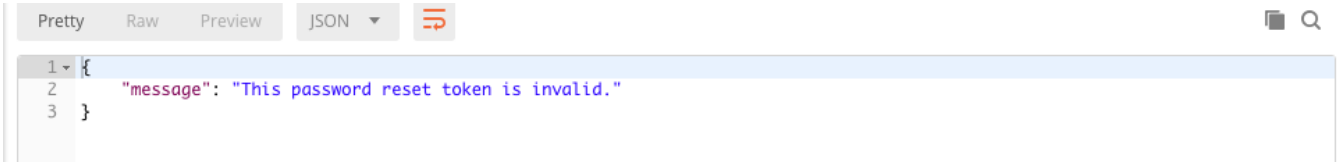
Buscar Restablecer Contraseña

- Si el token está activo, la respuesta contendrá los datos de éste.



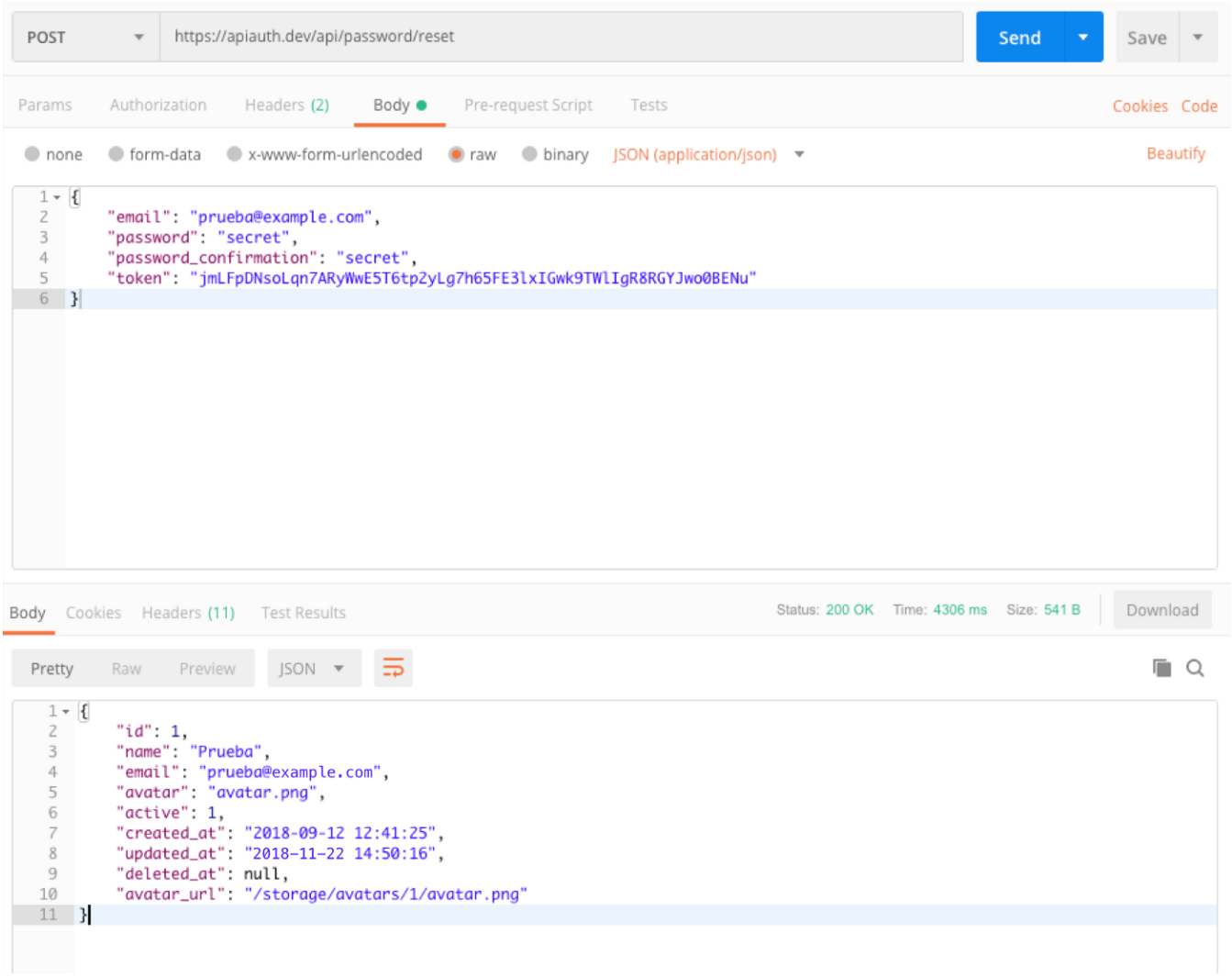
- Si el token no está activo, la respuesta es un error 404*



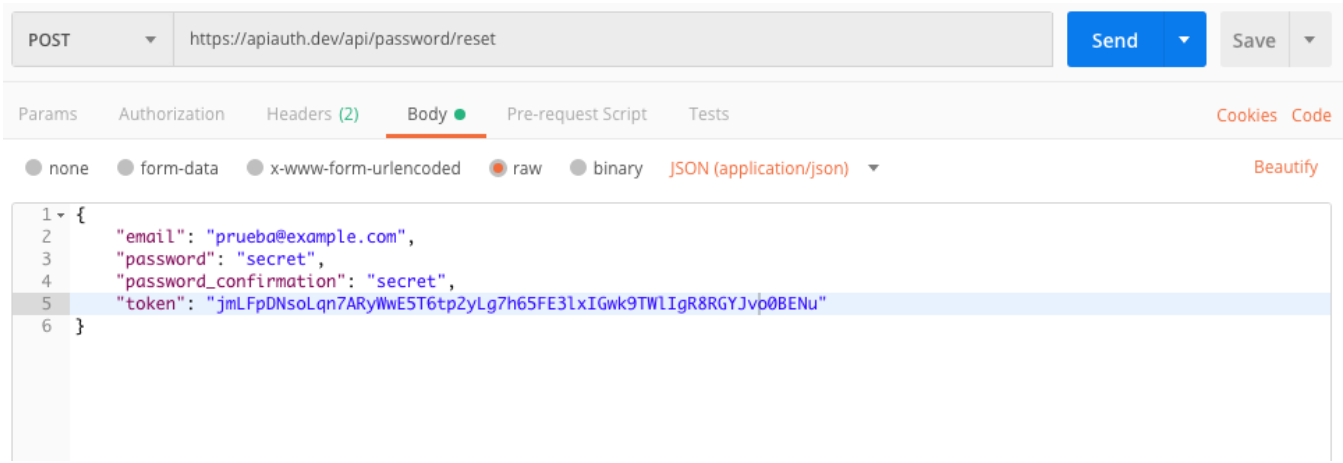


Reset Password

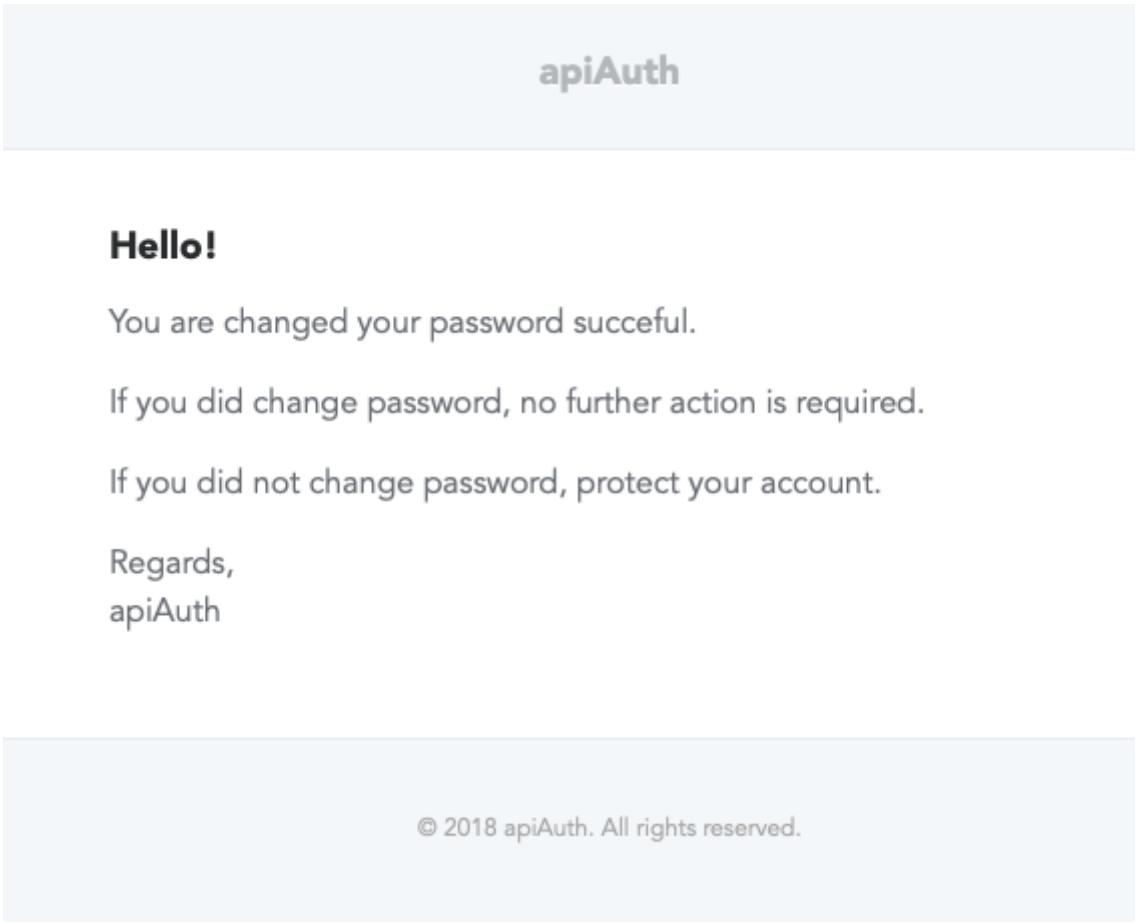
- Si el token está activo, entrega la respuesta con los datos del usuario



- Si el token no está activo, la respuesta es un error 404*



Por último, si se ha restablecido correctamente la contraseña, deberá llegar el siguiente correo:



> Con estas pruebas funcionales hemos terminado la cuarta parte correspondiente al restablecimiento de contraseñas.

. . .

Serie: Sistema de Autenticación API Rest con Laravel 5.6

- Parte 1 : Instalación y configuración

- Parte 2 : Confirmación de cuenta y notificaciones
- Parte 3 : Generar un avatar
- Parte 4 : Restablecer contraseña
- Parte 5 : Enviar notificaciones con espera en Redis

. . .

Gracias por tu lectura.

Si te ha gustado, podrías darme un aplauso y seguirme :)

. . .

Puedes compartir esta publicación en tus redes sociales

Claudio Vallejo (@cvallejo) | Twitter

The latest Tweets from Claudio Vallejo (@cvallejo). Wine, programmer, diver & photographer lover. Agronomist Engineer...

twitter.com

Laravel API Rest A Español Development

About Help Legal

Get the Medium app