Password Reset

Password Reset

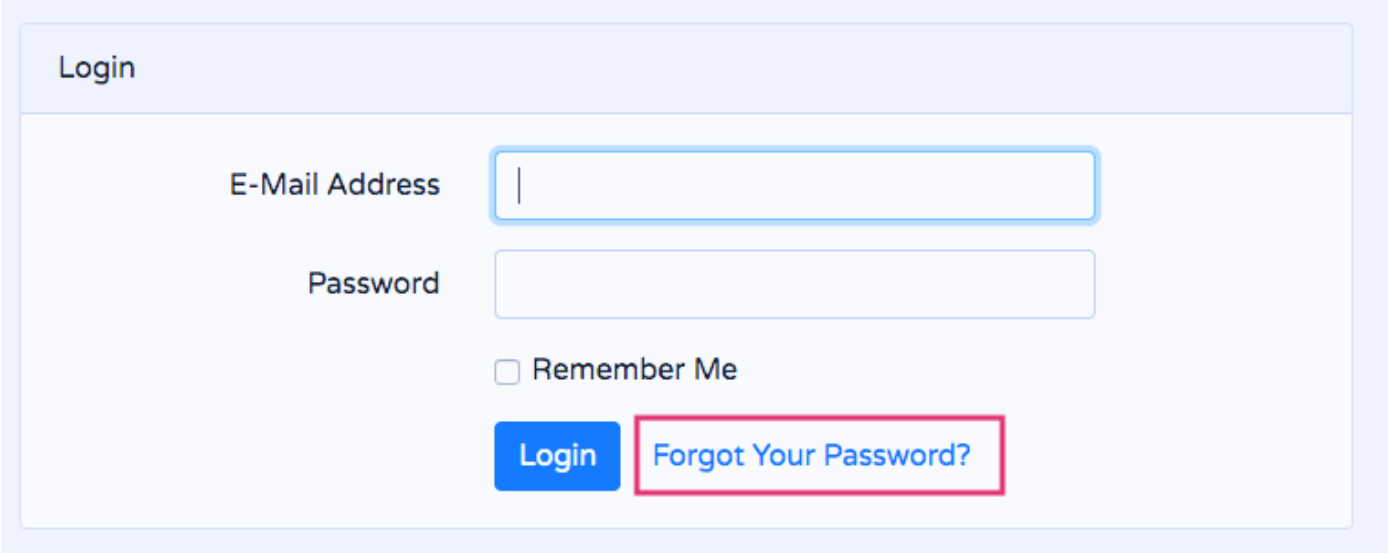
1. Introduction
2. Fixing Swift_TransportException error

1. Introduction

In this lesson, we're going to learn the flow of password reset in Laravel authentication. There are a couple of actions that need to take place:

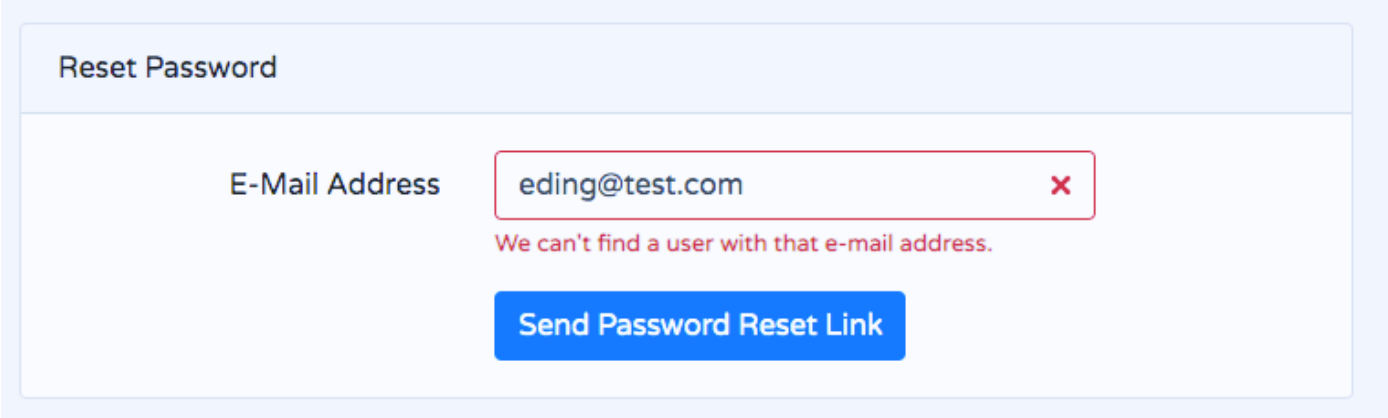
- You request a password reset
- You enter your email. If the email you entered is valid, a password reset link will be sent to your email.
- You Access the reset link
- If the link that you accessed is valid, you can enter your new password.

Just in case you forgot your password when you enter your credentials in the **Login** form, you can hit the **Forgot Your Password** link.



If you hit that link it'll take you to the **Reset Password** form. In that form, you can request a reset by entering the email that you entered when registering a new user.

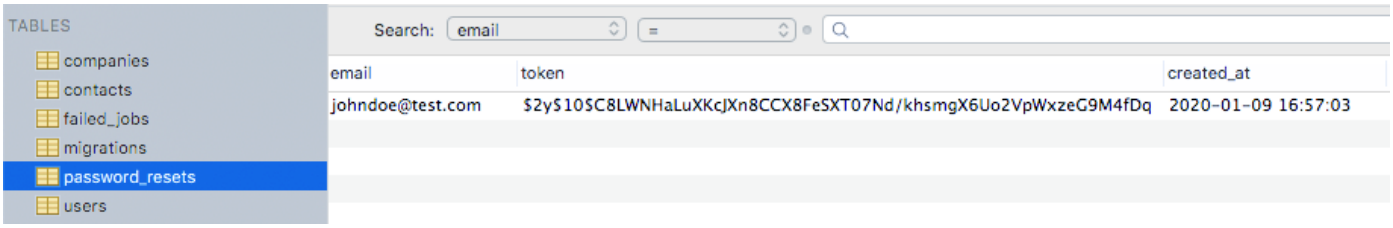
Once you hit the **Send Password Reset Link** button, under the hood it will find in the Users table the email that you provided. If the email you've entered did not found in the database, you'll get a validation error message like this..



But, if the email that you entered was valid (found in the database), you'll get `Swift_TransportException` error like this.

If you read the error, you'll realize that Laravel tried to send a message to your email. Since you didn't configure the email settings, that's why you got that error.

But If you see the `password_resets` table, you'll find a new record inserted.



2. Fixing Swift_TransportException error

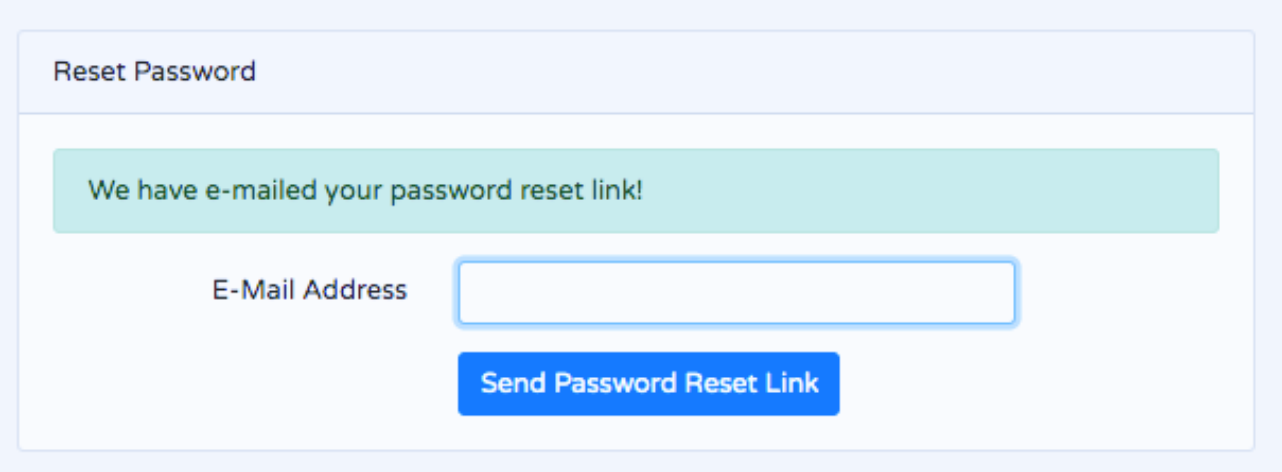
The simple way to fix that error is to use `log` instead of `smtp`. Let's open the `.env` file. Then change the `MAIL_MAILER` from `smtp` to `log`.

```
MAIL_MAILER=log
```

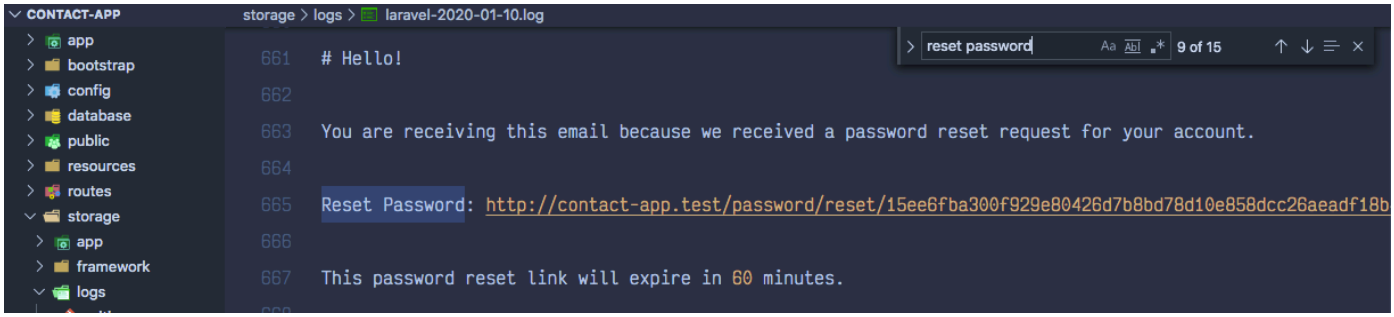
Also you need to adjust the `APP_NAME` to your project's url. This will make sure you that you'll get the proper password reset link on the sent email.

```
APP_URL=http://contact-app.test
```

Now if go back to **Reset Password** form, enter your valid email then hit the **Send Password Reset Link** you will get a success message telling you that the password reset link has been sent to your email.



Ideally you will go to your inbox, open the email, then hit the password reset link to actually reset your password. But since we used `log`, what we need to do is to navigate to `storage/logs` folder. Then open the last `.log` file. You can then simply find the reset password link, then copy to the clipboard.



If you take a look at the link it basically contains a token (original token) and an email.

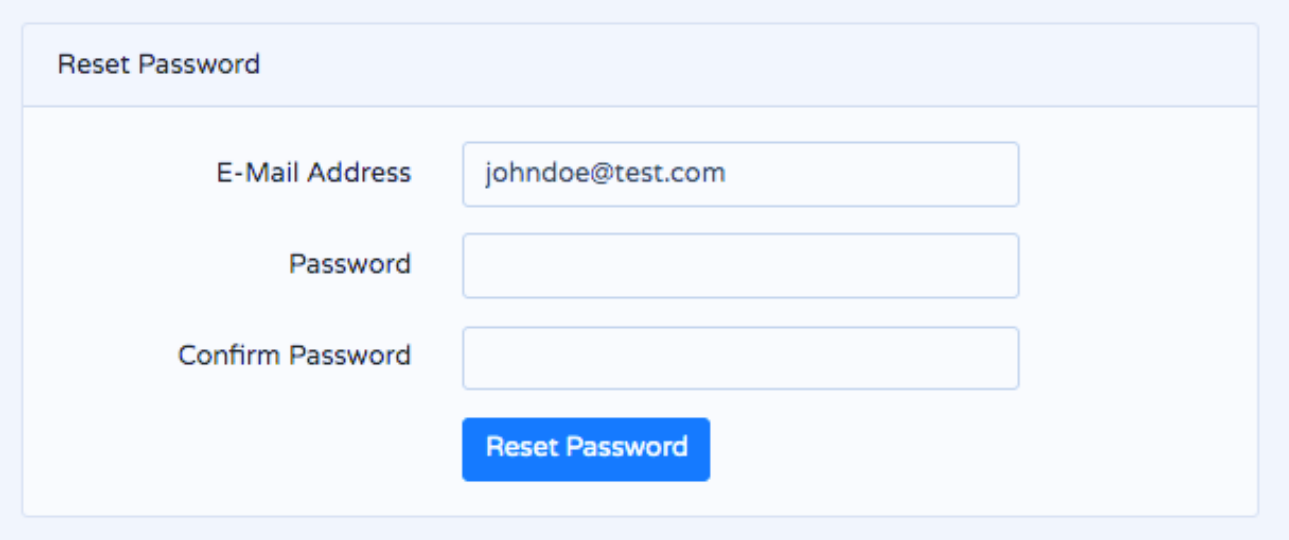
```
.../15ee6fba300f929e80426d7b8bd78d10e858dcc26aeadf18b4da0cff14e6bae0?
email=johndoe%40test.com
```

This mechanism is almost similar when you login an account. You provided email and original password. Once submitted, Laravel will encrypt the original password into a hash. Then match the email and hashed password with database record.

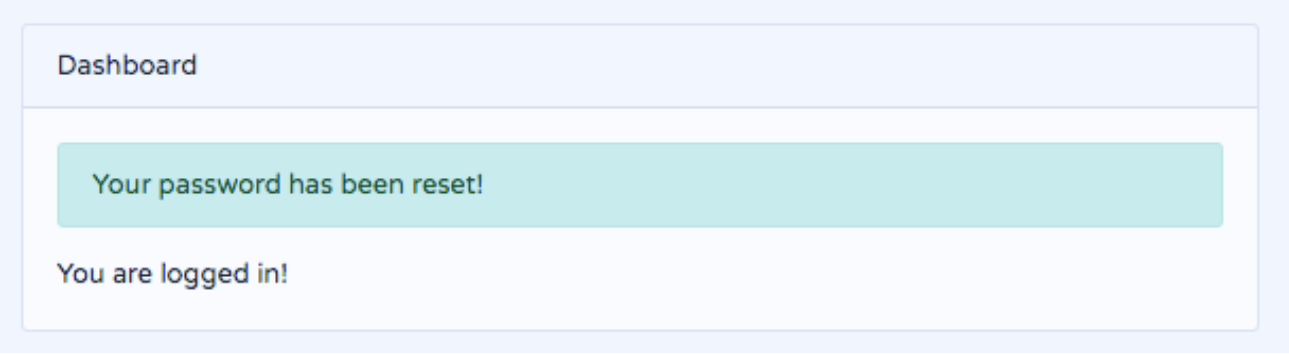
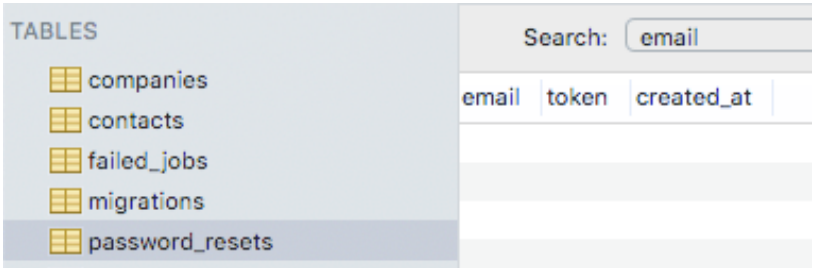
This way will ensure that only the right users can reset their password. Also the reset link only valid for `60` minutes. If for some reasons you want to increase or decrease the expire time, you can open `auth.php` file inside `config` folder. Then set the `expire` in `password` section with any number you prefer.

```
'passwords' => [
  'users' => [
    'provider' => 'users',
    'table' => 'password_resets',
    'expire' => 120,
    'throttle' => 60,
  ],
],
```

Now if you enter the link in your browser you will get a **Confirm Reset Password** form like this.



In this form the email input will populated automatically with email from the query string. You can then enter new password in the password and confirm password inputs, then hit **Reset Password** button to proceed. Once it done you'll be taken to `dashboard` page.



If you take a look at `password_resets` table again, the record now gone.