In Brief

1-Generating a secret token

2-Associating generated secret token with the user account and saving in database

3-Intialize the user active status to inactive

4-Send email to the email-id of the user

5-When the user correctly submits secret token making the user status to active

Detailed

1. Secret token is generated through randomstring package which is a library to create random strings.

Syntax is randomstring.generate(n) with n representing the length of the string

1. UserSchema contains fields- activation\_key(String), active(Boolean)

When the Post request for user registration is handled, activation\_key is mapped to secrete token and the active value to boolean

1. When the user log in with a email that has not been activated , then a error message is displayed , "email is not yet activated"
2. Then a email is triggered to the user, with the content

Hi there,

Thank you for registering!

Please verify your email by typing the following token:

Token: RQ4MvZ9mFTyx84WHPTEV88oSWZLVTomU

<http://localhost:3000/users/activation>

Email is triggered using the package nodemailer

1. nodemailer is then configued as shown

const transport = nodemailer.createTransport({

service: 'Mailgun',

auth: {

user: config.MAILGUN\_USER,

pass: config.MAILGUN\_PASS

},

tls: {

rejectUnauthorized: false

}

});

1. Service used was MailGun

<http://localhost:3000/users/activation> - This link redirects the user to the activation page, where activation code has to be submitted

1. If the activation\_key received through the post request matches with the activation\_key of the user with the given email id then the active status is turned to True. This completes the email activation process.