

Back-End Development



Nodemailer

#8b Email Delivering System

How to Create a Simple Email Delivering System Using Nodemailer & Gmail API



Nodemailer

Nodemailer is a module for Node.js applications to allow easy as cake email sending. The project got started back in 2010 when there was no sane option to send email messages, today it is the solution most Node.js users turn to by default.
(See nodemailer.com).

■ Installation

```
$ npm install nodemailer --save
```

xoauth2

XOAUTH2 is modern authentication platform of e-mail protocol. It continues to be the preferred mechanism for authenticating to Gmail IMAP. XOAUTH2 is standard OAuth 2.0 plus a nonstandard SASL binding for OAuth. (See npmjs.com/package/xoauth2).

■ Installation

```
$ npm install xoauth2 --save
```

Nodemailer & Gmail

#part 1

```
const nodemailer = require('nodemailer');
const xoauth2 = require('xoauth2');

var transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    type: 'OAuth2',
    user: 'alamatemailku@gmail.com',
    clientId: 'ambildarigooogleapi',
    clientSecret: 'ambildarigooogleapi',
    refreshToken: 'ambildarigooogleapi'
  }
});
```

Nodemailer & Gmail

#part 2

```
var mailOptions = {
  from: 'Tes Node JS <wkwk@haha.com>',
  to: 'alamat1@lin.com, alamat2@tang.com',
  subject: 'Tes Email NodeJS',
  text: 'Halo Dunia!',
  html: '<h1><i>Ini Email ya gaes!</i></h1>'
}

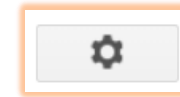
transporter.sendMail(mailOptions, (err, res)=>{
  if(err){console.log('Error gan!');}
  else{console.log('Email sukses terkirim!');}
})
```

How to Get Google Client ID

- Create a project on console.developers.google.com.
- Search: **Gmail API** then click **Enable**.
- Go to **Credentials**, click **OAuth Consent Screen** tab.
- Type the product name, then **save**.
- Create credentials, choose **OAuth Client ID**.
- On Application type, choose **Web Application**.
- On Authorized redirect URI type (must be **exactly same**):
<https://developers.google.com/oauthplayground>
- Click Create, then enjoy your **ClientId** & **ClientSecret**!

How to Get Refresh token

- Go to <https://developers.google.com/oauthplayground/>.
- Click **OAuth2 configuration** (gear icon).
- Check **Use your own OAuth credentials**.
- Input your **ClientId** & **ClientSecret**.
- On textbox scopes, input <https://mail.google.com/>.
- Click **Authorize App** then allow it.
- Click **Exchange authorization code for token**.
- Check **Auto-refresh the token before it expires**.
- Enjoy your **Refresh Token**!



Express & Nodemailer

express



Express, Nodemailer & Gmail

#part 1

```
const nodemailer = require('nodemailer');
const xoauth2 = require('xoauth2');
const express = require('express');
const app = express();

var transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    type: 'OAuth2',
    user: 'alamatemailku@gmail.com',
    clientId: 'ambildarigoogleapi',
    clientSecret: 'ambildarigoogleapi',
    refreshToken: 'ambildarigoogleapi'
  }
});
```

Express, Nodemailer & Gmail

#part 2

```
var mailOptions = {  
  from: 'Tes Node JS <wkwk@haha.com>',  
  to: 'alamat1@lin.com, alamat2@tang.com',  
  subject: 'Tes Email NodeJS',  
  text: 'Halo Dunia!',  
  html: '<h1><i>Ini Email ya gaes!</i></h1>'  
}
```

Express, Nodemailer & Gmail

#part 3

```
app.get('/', (req,res)=>{
  transporter.sendMail(mailOptions, (err,res2)=>{
    if(err){
      console.log('Error gan!');
      res.send('Error gan!');
    }else{
      console.log('Email sukses terkirim!');
      res.send('Email sukses terkirim!');
    }
  })
})

app.listen(3210, ()=>{
  console.log('Run @3210')
});
```

Using Attachments

```
var mailOptions = {
  from: 'Email Attachments <wkwk@haha.com>',
  to: 'lin@tang.com, lin@tang.id',
  subject: 'Tes Email Attachments',
  text: 'Halo Dunia!',
  html: '<h1><i>Ada Lampirannya!</i></h1>',
  attachments: [
    {
      filename: 'text1.txt',
      content: 'Halo Dunia!'
    },
    {
      filename: 'lintang.png',
      path: 'https://lintang.com/Barcelona.png'
    }
  ]
}
```

Back-End Development



Nodemailer

#8b Email Delivering System