**Sending Email After Form Submission with Node.js**

[Nodemailer](https://nodemailer.com/about/) is for sending email without the hustle of writing many lines of code. It uses the existing email services (Outlook, Hotmail, Gmail and so on) to authenticate and send messages. Authentication is done by simply adding your email credentials as a mail option. It also handles various transport protocols (SMTP as default). In this post, we will use the nodemailer module to send email once the express app receives form submission.

We will build the form submission mailer functionality on top of what we have done in the previous post ([Website Page Routing with Node.js Express and EJS](https://www.mydatahack.com/website-page-routing-with-node-js-express-and-ejs)). The major changes are:

* Adding contact form in contact.ejs.
* Adding different routers, including form post request handler and mailer.
* Adding redirect pages after sending emails.

Before we start, we should add the nodemailer module by running npm I nodemailer -SE. The rest of the dependencies are the same.

OK, let’s get started.

**Folder Structure**

Adding different routers and a few new ejs pages.

[cc lang="text" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

**Adding or Updating EJS Files**

**contact.ejs**

Simply adding the form to the existing contact.ejs. The form will be sent to contact/sent, which will be handled by the post request handler in contact.js.

[cc lang="html5" tab\_size="4" lines="-1"]

[/cc]

**contact\_send.ejs**

Redirected to this page after successful email send.

[cc lang="html5" tab\_size="4" lines="-1"]

[/cc]

**contact\_error.ejs**

Redirected to this page after unsuccessful email send.

[cc lang="html5" tab\_size="4" lines="-1"]

[/cc]

**Routers and Request Handlers**

In the index file, we will define each router and they can be imported in the app.js. Within app.js, we can do a path-based routing. When you specify the router folder to import module in app.js, it looks for index.js by default. Therefore, you only need to add the folder name.

**Index.js**

This is for organising different request handlers so that importing them in the main app.js becomes cleaner.

**contact.js**

This handles post requests from the form. The example uses Hotmail. It supports all the major email service providers. See the [documentation](https://nodemailer.com/about/) for further information about Nodemailer for your own implementation.

After successful email send, it routes to the email success page. It redirects the traffic to the get method in pageRouter.js.

**pageRouter.js**

Routing direct and redirect traffic.

**app.js**

This is very similar to the [previous example](https://www.mydatahack.com/website-page-routing-with-node-js-express-and-ejs). The main difference is to import different routers and use each router according to the different path.

All done. This is a simple demonstration on how easy it is to implement automated email functionality by Node.js Express and Nodemailer. Now it is time for you implement your own!