

ToiletPaper #126

Testing email exchange with fake SMTP servers

Author: Christian Förster / Senior Project Manager / Business Division Media

✘ Problem

Lots of emails are sent from your application. At one point, you would like to test if the mails are actually sent and preferably without spamming your own mailbox. You also want to use different email addresses for testing, not just your own. It would be a *nice to have* if this worked even offline. Sometimes you would also like to check the layout of an email in advance. Especially with HTML emails, it makes sense to open them with different email clients.

✔ Solution

There are various tools to simulate an SMTP server, so-called fake SMTP servers. If the SMTP host and port are configured accordingly in the application, emails sent by the application will be sent but intercepted by the fake SMTP server instead of reaching the actual addresses. These tools are suitable for use in local development environments as well as in integration, testing, and staging environments.

Here is a short overview of some common fake SMTP servers:

- [fake-smtp-server](#)
 - Advantages: freeware; can be used offline; HTTP port configurable; quickly installed and easy to use
 - Disadvantages: confusing display; rendering of HTML emails can only be tested with increased additional effort and additional tools
- [Mailtrap](#)
 - Advantages: professional tool that simulates a mail client in the browser, including the ability to display HTML mails; forwarding to a real mailbox is possible
 - Disadvantages: fee-based in most usage scenarios; rendering of HTML mails is not fully representative for all mail clients; can only be used online
- [FakeSMTP](#)
 - Advantages: freeware; can be used offline; HTTP port configurable; storage of incoming emails in EML format in a configurable directory
 - Disadvantages: last version from 2015; separate mail client required to view (HTML) emails
- Others, e.g. [SMTP Bucket](#), [DevelMail](#), [MailSlurper](#), [Ethereal](#), ...

➔ Example

After installing [fake-smtp-server](#) globally via npm global, it starts and waits on port 1025 for incoming mails. When the app is configured accordingly, sent mails with their meta data are displayed in JSON format at <http://localhost:1080/api/emails>:

```
[{ "attachments": [ ],
  "headerLines": [...],
  "text": "Welcome to my app, Max Mustermann",
  "textAsHtml": "<p>Welcome to my app, <b>Max Mustermann</b></p>",
  "subject": "Your Registration at my app was successful",
  "date": "2019-06-03T14:44:55.000Z",
  "to": {...},
  "from": {...},
  "messageId": "<1262711711.1.1559573095495>",
  "html": false
},
{...},
{...}]
```

+ Further Aspects

- <https://www.mailpoet.com/blog/email-testing-tools/>
- <https://www.softwaretestinghelp.com/email-testing-tools/>
- <https://medium.com/email-design/why-dont-email-clients-use-modern-rendering-engines-1971a0e0fda4>
- <https://www.gethighlights.co/blog/email-testing-tools/>