ToiletPaper #126

Fake-SMTP-Server zum Testen von Mailversand

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

✗ Problem

Aus eurer Anwendung heraus werden jede Menge E-Mails verschickt. Jetzt würdet ihr natürlich gerne testen, dass die Mails auch tatsächlich verschickt werden und zwar am liebsten ohne dass euer eigenes Postfach zugespammt wird. Außerdem wollt ihr zum Testen verschiedene E-Mail-Adressen verwenden und nicht nur eure eigene. Wenn das ganze sogar offline funktionieren würde, wäre das besonders schick. Manchmal würde man auch gerne vorab das Layout einer E-Mail überprüfen. Gerade bei HTML-E-Mails macht es dann Sinn, diese mit verschiedenen E-Mail-Clients zu öffnen.

✓ Lösung

Es gibt diverse Tools, mit denen man einen SMTP-Server simulieren kann, sogenannte Fake-SMTP-Server. Wenn SMTP-Host und -Port in der Anwendung entsprechend konfiguriert werden, werden von der Anwendung versandte E-Mails zwar verschickt, aber vom Fake-SMTP-Server abgefangen, anstatt den eigentlichen Adressaten zu erreichen. Diese Tools eignen sich für den Einsatz in lokalen Entwicklungsumgebungen ebenso wie in Integrations-, Testing- und Staging-Umgebungen.
Hier eine kurze Übersicht über einige gängige Fake-SMTP-Server:

- fake-smtp-server
  - Vorteile: Kostenloses Tool; offline nutzbar; HTTP-Port konfigurierbar; schnell installiert und leicht nutzbar
  - Nachteil: Darstellung etwas unübersichtlich; Rendering von HTML-E-Mails nur mit erhöhtem Zusatzaufwand und weiteren Tools testbar
- Mailtrap
  - Vorteile: Professionelles Tool, das im Browser einen Mail-Client simuliert inkl. der Fähigkeit HTML-Mails darzustellen; Weiterleitung an reales E-Mail-Postfach möglich
  - Nachteile: In den meisten Einsatz-Szenarien kostenpflichtig; Rendering von HTML-Mails ist nicht völlig repräsentativ für alle Mail-Clients; nur online nutzbar
- FakeSMTP
  - Vorteile: Kostenloses Tool; offline nutzbar; HTTP-Port konfigurierbar; Speicherung von eingegangen E-Mails im EML-Format in einem konfigurierbaren Verzeichnis
  - Nachteile: letzte Version von 2015; separater Mail-Client zum Anzeigen von (HTML-)E-Mails benötigt
- weitere z. B. SMTP Bucket, DevelMail, MailSlurper, Etheral ...

➔ Beispiel

Wenn fake-smtp-server global per npm global installiert worden ist, startet es und schon lauscht es auf Port 1025, ob Mails eingehen. Bei entsprechender Konfiguration der App, werden verschickte E-Mails mit ihren Meta-Daten unter http://localhost:1080/api/emails im JSON-Format dargestellt:

```
[{  "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
},
{...},
{...}]
```

+ Weiterführende Aspekte

- 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/