Wie man E-Mail-Adressen in einem Mautic-Formular mit JavaScript validiert

Validierung ist eine Methode zur ständigen Überprüfung, Verifizierung, Authentifizierung und zum Nachweis der Richtigkeit von etwas. E-Mail-Validierung ist eine Methode zur Überprüfung, Authentifizierung und zum Nachweis der Richtigkeit und Gültigkeit von E-Mails. E-Mails sind eines der bequemsten und beliebtesten Kommunikationsmittel, das von vielen Menschen genutzt wird - die Notwendigkeit, E-Mails zu validieren, ist daher sehr wichtig:

1. Sie verbessern die Qualität der Daten: Durch die Validierung von E-Mails können Sie die Genauigkeit der Daten sicherstellen, da nur korrekte und gültige E-Mails gesammelt werden; eine große Anzahl ungültiger E-Mail-Adressen bedeutet, dass es eine Menge zu verbessern gibt.
2. Verbessert das E-Mail-Marketing: Die Validierung von E-Mails verbessert das E-Mail-Marketing, da beim Versand von E-Mails nur zustellfähige E-Mails berücksichtigt werden.
3. Führt zu einer erhöhten Zustellbarkeit: Wenn E-Mails validiert werden, müssen Sie sich nicht mit Problemen herumschlagen, die beim Versand falscher oder ungültiger E-Mails auftreten, wie z. B. eine hohe Spam-Beschwerdequote.
4. Verhindert E-Mail-Bounce: E-Mail-Bounces treten auf, wenn die E-Mail-Adresse des Empfängers nicht korrekt ist oder die E-Mail nicht existiert. Die E-Mail-Validierung bereinigt die Liste der E-Mails und entfernt ungültige oder falsche E-Mails.
5. Verbessert den Ruf des Absenders: Durch die Validierung von E-Mails werden Hindernisse wie fehlerhafte oder falsch eingegebene E-Mails beseitigt, die den guten Ruf des Absenders beeinträchtigen können.
6. Spart Geld: Das Versenden von E-Mails kostet Geld. Durch die E-Mail-Validierung werden unerwünschte E-Mails eliminiert, so dass nur relevante E-Mails übrig bleiben, wodurch die Kosten gesenkt werden.

E-Mails können mit verschiedenen Methoden validiert werden, z. B. durch den Versand von E-Mails mit Java, die Verwendung von Mautic Campaign zur Validierung von E-Mail-Adressen usw. In diesem Artikel wollen wir uns

ansehen, wie wir Javascript einrichten können, um E-Mail-Adressen auf der Client-Seite zu validieren. Es ist sehr schnell und wird von den meisten Menschen bevorzugt, und wir werden sehen, wie dies in den folgenden Abschnitten geschehen kann.

Was sind die Richtlinien, die bei der E-Mail-Validierung beachtet werden sollten?

Bei der Validierung von E-Mails ist es sehr wichtig, dass Sie die Struktur der E-Mail berücksichtigen. Dies ist eine wichtige Regel, die beachtet werden sollte. E-Mails bestehen aus zwei Teilen, die durch dieses _@_-Symbol getrennt sind. Die beiden Teile werden als lokaler Teil und Domänenteil bezeichnet. Der erste Teil, der lokale Teil, besteht normalerweise aus:

- Großbuchstaben (A-Z) und Kleinbuchstaben (a-z)
- Ziffern (0-9)
- Einige Sonderzeichen, die sind:! # $ % & ' * + - / = ? ^ _ ` { | ~
- . (period, dot or full stop), die nicht das erste oder letzte Zeichen sein dürfen und nicht wiederholt werden können

Der zweite Teil, der Domänenteil, besteht aus:

- Großbuchstaben (A-Z) und Kleinbuchstaben (a-z)
- Ziffern (0-9)
- "." und "-" Punkt und Bindestriche

Wie man Regex verwendet

Bei der Validierung von E-Mails auf der Client-Seite mit Javascript ist die bevorzugte Methode die Verwendung von regulären Ausdrücken, auch bekannt als Regex.

Hinweis: Es gibt keine universellen oder allgemein anerkannten Regex-Ausdruck, der für E-Mail-Validierung verwendet wird. Die meisten Leute

verwenden verschiedene Regex-Ausdrücke, die verschiedene Anwendungsfälle bei der Validierung von E-Mails erfüllen. Bevor wir eine Regex verwenden, müssen wir sie mit einigen (richtigen und falschen) E-Mail-Adressen testen, um sicher zu sein, dass sie einen Fehler auslöst, wenn eine gefälschte oder eine falsche Adresse verwendet wird. Das folgende Regex-Beispiel wurde getestet, um bis zu 99 % der E-Mail-Adressen zu überprüfen.

```
/^(([^<>()[]\.,;:s@"]+(.[^<>()[]\.,;:s@"]+)*)|(".+"))@((([0-9]{1,3}.[
0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}])|(([a-zA-Z-0-9]+.)+[a-zA-Z]{2,}))$/
```

Um unseren Regex-Ausdruck schnell mit einer E-Mail-Adresse zu testen, würden wir diese Funktion verwenden

```
function testRegex(email) {return /^(([^<>()[]\.,;:s@"]+(.[^<>()[]\.,
;:s@"]+)*)|(".+"))@((([0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}])|(
([a-zA-Z-0-9]+.)+[a-zA-Z]{2,}))$/.test(email) }
```

Hier ist die Ausgabe beim Test mit diesen beiden E-Mail-Adressen.

- testRegex(jane@.example.com) //true
- testRegex(jane..smith@example.com) //false

Verwendung von Javascript mit Regex

Das Code-Beispiel unten zeigt Regex-Implementierung in Javascript, der Code ist nicht schwer zu verstehen, aus dem Code, wenn die E-Mail-Wert mit dem Regex dann gibt es (Dies ist eine gültige E-Mail-Adresse) Pop-up-Alarm, wenn es nicht übereinstimmt, gibt es (Dies ist nicht eine gültige E-Mail-Adresse) Pop-up-Alarm.

index.js

```
function ValidateEmail(inputText){var mailformat = /^(([^<>()[]\.,;:s
@"]+(.[^<>()[]\.,;:s@"]+)*)|(".+"))@(([[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,
3}.[0-9]{1,3}])|(([a-zA-Z-0-9]+.)+[a-zA-Z]{2,}))$/;if(inputText.value
.match(mailformat)) {alert("Valid email address!");document.form1.tex
t1.focus();return true;}else {alert("You have entered an invalid emai
l address!");document.form1.text1.focus();return false;}}
```

Nun wollen wir diesen Javascript-Code auf ein Formular anwenden.

Validierung von E-Mails in einem Formular mit Javascript

Wir werden ein Formular erstellen, das eine E-Mail-Eingabe hat, und Javascript verwenden, um die in das Formular eingegebenen E-Mails zu validieren, wenn sie übermittelt werden. Schauen wir uns an, wie das funktioniert.

Zunächst erstellen wir eine Datei mit dem Namen index.html und fügen den unten stehenden Code ein. Die index.html wird den zuvor erstellten Javascript-Code verwenden, um die E-Mails auf der Client-Seite zu validieren.

index.html

```
<!DOCTYPE html><html lang="en"><head><meta charset="<dfn class="the
saurus">utf-8</dfn>"><title>JavaScript <dfn class="thesaurus">form<
/dfn> validation with regex</title><link rel='stylesheet' href='ind
ex.css' type='text/css' />      </head><body onload='document.form1
.text1.focus()'><div class="mail"><h2>Input an email and Submit to
validate</h2><form name="form1" action="#"> <ul><li><input type='te
xt' name='text1'/></li><li> </li><li class="submit"><input type="su
bmit" name="submit" value="Submit" onclick="ValidateEmail(document.
form1.text1)"/></li><li> </li></ul></form></div><script src="index.
js"></script></body></html>
```

Als Nächstes erstellen wir eine Datei mit dem Namen index.css und platzieren darin den unten stehenden Code, der das Stylesheet zur Verschönerung des Formulars darstellt.

index.css

```
li {     list-style-type: none;     font-size: 16pt; } .mail {
margin: auto;     padding-top: 10px;     padding-bottom: 10px;
width: 400px;     border: 1px soild  silver; } .mail h2 {     margi
n-left: 38px; } input {     font-size: 20pt; } input:focus, textare
a:focus{     background-color: #2ef8cf; } input submit {     font-s
ize: 12pt; } .rq {     color:  #FF0000;     font-size: 10pt;     }
```

Führen Sie den Code aus und sehen Sie sich die Ausgabe in Ihrem Browser an. Hier ist die erwartete Ausgabe des Codes.

Nun wollen wir unseren Code mit einigen E-Mails testen - der Test mit dieser E-Mail jane.doe@gmail.com wird funktionieren und ein Pop-up-Fenster mit der Meldung "Gültige E-Mail-Adresse" anzeigen, wie unten zu sehen.

Die Eingabe dieser E-Mail jane..smith@example.com funktioniert nicht und führt zu der Meldung "You have entered an invalid email address! Hier ist die Ausgabe unten.

Sie können versuchen, mit mehreren E-Mails zu testen, um zu sehen, welche gültig oder ungültig ist

- jane@.example.com
- Jane.example.com
- john12@gmail.com
- smith+consult@hotmail.com
- peter@example.com
- jake@example.co.in
- peter.wash@example.com
- john_smith@example.com
- jane@example.company.in
- .jake@example.com
- peter@example.com.
- jane@example.c
- smith@example.company

Andere Überlegungen

Die Validierung von E-Mail-Adressen auf der Client-Seite mit JavaScript unter Verwendung von Regex macht Spaß, wie die obigen Beispiele zeigen, obwohl nicht jeder das Konzept von Regex begreifen kann, da es manchmal etwas schwierig zu verstehen ist und es keine allgemeine oder universelle Regex für die Validierung von E-Mails gibt; die Leute neigen dazu, die Regex nach ihren Bedürfnissen zu schreiben und einige Regex-Ausdrücke können einige gültige E-Mail-Adressen ausschließen.

Um die Verwendung einiger schwieriger Regex-Ausdrücke bei der Validierung von E-Mails zu vermeiden und auch Ihre E-Mails zu testen und zu debuggen, um Probleme zu beheben, bevor Sie sie versenden, können Sie Mailtrap verwenden.

Mit Mailtrap können Sie überprüfen, ob Ihre E-Mails richtig formatiert sind und nicht im Spam-Ordner landen. Außerdem werden Sie keine gültigen E-Mail-Adressen los, die bei der Verwendung von Regex eliminiert werden könnten.

Viel Spaß beim Versenden von E-Mails!

Online URL: https://kb.mautic.org/article/wie-man-e-mail-adressen-in-einem-mautic-formular-mit-javascript-validiert.html

SyntaxHighlighter.config.stripBrs=true; SyntaxHighlighter.all();