

Validierung von Eingabefeldern

Ab der Version 1.0 des LoxBerry gibt es eine neue Validierungsfunktion für Eingabefelder.

Dafür gibt es das Script `/opt/loxberry/webfrontend/html/system/scripts/validate.js` von [Christian Wörstenfeld](#).

Die [alte Version](#) wird für eine Übergangszeit noch parallel benutzbar sein und später aus dem LoxBerry Core entfallen. Sollte sie weiter benötigt werden, muss sie vom Plugin eingebunden werden.

- [1\) Voraussetzungen für die Validierung](#)
- [2\) Besonderheit weitere Formatierungen](#)
- [3\) Beispiele für reguläre Ausdrücke](#)
- [4\) Standard-Validierungen \(Alias special:\)](#)

1) Voraussetzungen für die Validierung

Die Validierung benutzt ein klassisches Eingabefeld, und Prüft die Werte während der Eingabe.

Ein Standardwert kann über das Template System übergeben werden. Hier im Beispiel ist es `<TMPL_VAR DEFAULT_FROM_BACKEND>`

```
<input id='eingabefeld' type='hidden' value='<TMPL_VAR DEFAULT_FROM_BACKEND>' data-validation-error-msg="
<TMPL_VAR ERR_FROM_BACKEND>" data-validation-rule="special:alphanumeric">
```

Wichtig sind aber die folgenden Attribute:

- `data-validation-error-msg` - Dieses Attribut enthält die Fehlermeldung bei falschen Eingaben und sollte vom Backend aus der Sprachdatei befüllt werden. z.B. `<TMPL_VAR ERR_FROM_BACKEND>`
- `data-validation-rule` - Dieses Attribut enthält einen regulären Ausdruck oder eine Standardregel welche mit `special:` beginnen muss, gegen den die Eingabe geprüft wird. Einige Beispiele findet ihr weiter unten.
- `data-validation-or` - Optional: Wenn zwei Felder ODER-verknüpft werden sollen ist im Attribut `data-validation-or` das jeweilige Partnerobjekt einzutragen. Also zum Beispiel in Feld1: `data-validation-or="#Feld2"`
- `data-validation-coloring` - Optional: Wenn eine Validierung aber keine Einfärbung erfolgen soll, das Attribut `data-validation-coloring="off"` setzen.
-

Der Ausdruck im Beispiel erlaubt die 26 großen und kleinen Buchstaben des Alphabets und alle Ziffern von 0 bis 9.

Weitere Infos und Tests gibt es bei <https://regex101.com>

Um die Validierung dann einzuschalten, muss sie in einem JavaScript-Block aktiviert werden.

Dies sollte zur Sicherheit erst erfolgen, wenn die Seite geladen ist. Deshalb innerhalb der `$(document).ready` Funktion.

```
<script>
    $( document ).ready(function()
    {
        validate_enable('#eingabefeld');
    });
</script>
```

Da das INPUT Objekt mit seiner `id` angesprochen wird, muss vor die `id` Bezeichnung eine `#` eingefügt werden. Aus `id='eingabefeld'` wird also `#eingabefeld'`.

Mit `validate_enable` wird nur die Prüfung des Feldes eingeschaltet. Dies verhindert nicht das Absenden des Formulars mit ungewollten Werten. Um das zu Verhindern, muss zusätzlich die Funktion `validate_chk_object` aufgerufen werden.

Der Funktion `validate_chk_object` wird ein Array der beim Absenden des Formulars zu prüfenden Objekte übergeben. (Mehrere Felder werden durch Komma getrennt)

```
// Ein Feld
validate_chk_object(['#eingabefeld']);

// oder mehrere Felder
validate_chk_object(['#eingabefeld1', '#eingabefeld2']);
```

Sollen Eingabefelder durch veränderte Bedingungen auf der Webseite nicht mehr verhindern, dass das Formular abgesendet wird, können sie durch den Aufruf der Funktion `validate_clean_objects` wieder freigegeben werden. Die Überprüfung und Einfärbung bleibt aktiviert aber bereits eingegebene Werte in diesen Objekten werden gelöscht.

```
// Ein Feld
validate_clean_objects(['#eingabefeld']);
// oder mehrere Felder
validate_clean_objects(['#eingabefeld1', '#eingabefeld2']);
```

Das ist alles was für eine Basis-Validierung benötigt wird.

2) Besonderheit weitere Formatierungen

Soll eine andere Hintergrundfarbe benutzt werden, kann dies über JavaScript erfolgen:

```
<script>
    // Andere Hintergrundfarbe
    $('#eingabefeld').css('background-color', '#FFFC0');

    // Soll eine Validierung erfolgen, jedoch keine Einfärbung, kann das Attribut data-validation-coloring="
off" gesetzt werden
    <INPUT ... data-validation-coloring="off" ...>
</script>
```

3) Beispiele für reguläre Ausdrücke

Hier einige Beispiele für reguläre Ausdrücke. Diese werden ohne führenden und abschließenden / eingegeben. Die Syntax ist hierbei die Syntax für JavaScript.

```
1) Alle Buchstaben und Zahlen - Länge zwischen 4 und 10 Zeichen:
    ^([A-Za-z0-9]){4,10}$"
2) Hostnamen nach RFC 1123 aber nicht PAUL
    ^(?!(PAUL$)(?![0-9]+$)(?!.*-$)(?!-[a-zA-Z0-9-]){1,63})$
3) IP-Adresse (IPv4)
    ^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.((25[0-5]|2[0-4][0-9]|
[01]?[0-9][0-9]?)\.((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$
4) IP Network Mask
    ^(((255\.){3}(254|252|248|240|224|192|128|0+))|((255\.){2}(255|254|252|248|240|224|192|128|0+)\.0)|
((255\.)(255|254|252|248|240|224|192|128|0+)\.0+)\{2})|((255|252|248|240|224|192|128|0+)\.0+)\{3}))$
5) WLAN SSID
    ^
([\w\u0020\u0024\u0040\u005e\u0060\u002c\u007c\u0025\u003b\u002e\u007e\u0028\u0029\u002f\u005c\u007b\u007d\u003a
\u003f\u005b\u005d\u003d\u002d\u002b\u00f5\u0023\u0021]{1,32})$
6) WLAN WPA Key
    ^([\u0020-\u007e\u00a0-\u00ff]{8,64})$
7) Passwort mit 5 bis 20 Zeichen und mindestens 1 Großbuchstabe, 2 Kleinbuchstaben, 1 Ziffer und eins der
folgenden Sonderzeichen: .!@#$$&*-_ und ohne Umlaute
    ^(?=.*[A-Z])(?=.*[!\@\#\$\&*\.\-\_])(?=.*[0-9])(?=.*[a-z].*[a-z]).{5,20}$
8) 4-8 Ziffern-PIN oder nichts
    ^(((0-9){4,8})|())$
9) IP-Port Bereich
    ^(((1-9){1}[[1-9][0-9]{1,3})|1-5[0-9]{4}|6[0-4][0-9]{3}|65[0-4][0-9]{2}|655[0-2][0-9]|6553[0-5])$
10) Special Rules
    special:<keyword_siehe_weiter_unten>
```

4) Standard-Validierungen (Alias special:)

Folgende Validierungen können anstatt eines Regex-Ausdrucks verwendet werden. Wenn welche Fehlen, einfach in die Kommentare damit.

```
'compare-with'
//Vergleicht zwei Felder.
Beispiel: data-validation-rule="special:compare-with:#input_feld_id"
```

```

'alpha'
  // Alle Buchstaben des Alphabets a-z und A-Z
  Beispiel: data-validation-rule="special:alpha"
'alpha-uppercase'
  // Alle großen Buchstaben des Alphabets A-Z
  Beispiel: data-validation-rule="special:alpha-uppercase"
'alpha-lowercase'
  // Alle kleinen Buchstaben des Alphabets a-z
  Beispiel: data-validation-rule="special:alpha-lowercase"
'alphanumeric'
  // Alle Buchstaben des Alphabets a-z sowie A-Z und Ziffern 0-9
  Beispiel: data-validation-rule="special:alphanumeric"
'alphanumeric-lowercase'
  // Alle kleinen Buchstaben des Alphabets a-u und Ziffern 0-9
  Beispiel: data-validation-rule="special:alphanumeric-lowercase"
'alphanumeric-uppercase'
  // Alle großen Buchstaben des Alphabets A-Z und Ziffern 0-9
  Beispiel: data-validation-rule="special:alphanumeric-uppercase"
'alpha-ws', 'alpha-uppercase-ws', 'alpha-lowercase-ws', 'alphanumeric-ws', 'alphanumeric-lowercase-ws' und
'alphanumeric-uppercase-ws'
  // Wie die Regeln ohne ws jedoch auch Leerzeichen erlaubt (whitespaces)
'digits' oder 'number' oder 'numeric'
  // Ziffern
  Beispiel: data-validation-rule="special:digits"
'number-exact-digits'
  // Genau diese Zifferanzahl
  Beispiel: data-validation-rule="special:number-exact-digits:5"
  Gültig: 44556, 63451
  Ungültig: 12, 4, 45567
'number-min-digits'
  // Mindestanzahl an Ziffern
  Beispiel: data-validation-rule="special:number-min-digits:3"
  Gültig: 456, 3451
  Ungültig: 12, 4
'number-max-digits'
  // Maximalanzahl an Ziffern
  Beispiel: data-validation-rule="special:number-max-digits:6"
  Gültig: 456, 1
  Ungültig: 1275634576, 4354345
'number-min-max-digits'
  // Mindest- und Maximalanzahl an Ziffern
  Beispiel: data-validation-rule="special:number-min-max-digits:3:5"
  Gültig: 456, 7854, 24378
  Ungültig: 12, 754345
'number-exact-value'
  // Exakter Zahlenwert. (Bereich -2147483648 bis 2147483647)
  Beispiel: data-validation-rule="special:number-exact-value:3456"
  Gültig: 3456
  Ungültig: 3455, 3457, 123
'number-min-value'
  // Minimaler Zahlenwert. (Bereich -2147483648 bis 2147483647)
  Beispiel: data-validation-rule="special:number-min-value:55"
  Gültig: 67, 882
  Ungültig: 34, 12
'number-max-value'
  // Maximaler Zahlenwert. (Bereich -2147483648 bis 2147483647)
  Beispiel: data-validation-rule="special:number-max-value:55"
  Gültig: 44, 22
  Ungültig: 134, 56
'number-min-max-value'
  // Zwischen minimalem und maximalem Zahlenwert einschließlich des Wertes. (Bereich -2147483648 bis
2147483647)
  Beispiel: data-validation-rule="special:number-min-max-value:55:66"
  Gültig: 55, 60, 66
  Ungültig: 54, 67, 5654
'hostname'
  // Hostname nach RFC1123
  Beispiel: data-validation-rule="special:hostname"
  Gültig: loxberry
  Ungültig: lox-berry
'domainname_or_ipaddr'

```

```

// Hostname nach RFC1123 oder IP Adresse oder Domainname
Beispiel: data-validation-rule="special:domainname_or_ipaddr"
Gültig: loxberry, 192.168.1.10
Ungültig: lox-berry, 345.6.78.8
'hostname_or_ipaddr'
// Hostname nach RFC1123 oder IP Adresse oder Domainname
Beispiel: data-validation-rule="special:hostname_or_ipaddr"
Gültig: loxberry, loxberry.de, www.loxberry.de, 192.1.3.4
Ungültig: lox-berry, www.loxberry.invalid
'ipaddr'
// IP Adresse
Beispiel: data-validation-rule="special:ipaddr"
Gültig: 192.168.1.2
Ungültig: 345.67.8.9
'netmask' oder 'ipmask'
// IP Subnetz-Maske
Beispiel: data-validation-rule="special:netmask"
Gültig: 255.255.255.0, 255.255.255.224
Ungültig: 255.255.255.255, 255.0.255.0
'port' oder 'ipport'
// IP Port Bereich von 1-65535
Gültig: 80, 8080
Ungültig: 0, 660000
'ssid'
// WLAN SSID
'wpa'
// WLAN WPA Key
'url'
// Eine Internetadresse http / https / ftp / sftp
Beispiel: data-validation-rule="special:url"
Gültig: http://loxberry.de
'email'
// Eine eMail Adresse
'emails'
// eMail Adressen (durch Semikolon getrennt)
'alphanumeric-accented-ws'
// Für Namen wie André-Jürgen, Bjørn, Müller inkl. . _ - und Leerzeichen

```

Danke für's Lesen!