

FRONTEND MĀJASLAPAS IZSTRĀDE

8. lekcija - formu validācija



LATVIJAS UNIVERSITĀTE
**BIZNESA, VADĪBAS
UN EKONOMIKAS
FAKULTĀTE**

VUMC VADĪBAS UN
UZŅĒMĒJDARBĪBAS
MĀCĪBU CENTRS

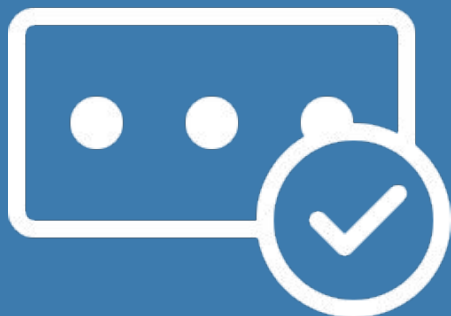
NACIONĀLAIS
ATTĪSTĪBAS
PLĀNS 2020



EIROPAS SAVIENĪBA
Eiropas Sociālais
fonds

IEGULDĪJUMS TAVĀ NĀKOTNĒ

ESF projekts Nr. 8.4.1.0/16/I/001 "Nodarbināto personu profesionālās kompetences pilnveide"



Datu validācija ar HTML

Formas validācija

Pirms nosūtām datus no formas(submit), tos nepieciešams pārbaudīt - vai tie atbilst paredzētajiem vērtības formātam.

Formu validāciju veic gan pārlūkprogrammā, gan servera pusē.

Veicot validāciju, mēs varam nodrošināt labāku lietotāja pieredzi mājaslapā. Lietotājs var izlabot kļūdains datus, negaidot atbildi no servera, ja veicam datu validāciju pirms tie tiek nosūtīti.

Ar datu validāciju pārlūkprogramā varam nodrošināt arī neliela mēra lietotāja un servera datu drošību.

Datu validācija arī servera pusē ir obligāti nepieciešama, jo tikai tā varam parūpēties ar servera drošību.

Neatbilstošu datu formāta ievadišanas gadījumā, varam informēt lietotāju par kļūdu ar kļūdas paziņojumu.

Sign Up

Username:

Username must be between 3 and 25 characters.

Email:

Password:

Password must has at least 8 characters that include at least 1 lowercase character, 1 uppercase characters, 1 number, and 1 special character in (!@#\$%^&*)

Confirm Password:

Please enter the password again

SIGN UP

Validācija ar HTML


Kopš HTML5 ir pieejami rīki, kas nodrošina formas datu validāciju arī bez JavaScript izmantošanas. Vienkāršās aplikācijas ar to var būt gana, lai nodrošinātu labu lietotāja pieredzi.

HTML formas validāciju panākam ar HTML formas elementu atribūtiem:

required - formas lauka vērtība ir obligāta
minlength un **maxlength**: teksta(string) vērtības minimālais un maksimālais simbolu skaits.
min un **max**: skaitlisku vērtību(number) minimālais un maksimālais lielums.
type - ievades laukā sagaidāmais datu formāts(skaitlis, teksts, epasts u.c.).
pattern - specifiska formāta vērtība, ko definējam ar RegEx izteiksmi.

Text Input

Submit

 Please fill out this field.

HTML formas validācijas stāvokļi

Ja formas datu validācija ir izdevusies, tie ir pareizi, izpildās sekojošie nosacījumi:

- Formas ievades elements satur CSS pseido klasi `:valid`. To izmantojot, varam piešķirt formas elementam vizuālu noformējumu.
- Dati tiks nosūtīti uz serveri, ja vien kas cits to neaizkavē, piem., papildus JS validācija.

Ja formas datu validācija nav izdevusies, dati ir nepareizi, izpildās šie nosacījumi:

- Formas ievades elements satur CSS pseido klasi `:invalid`. To izmantojot, varam piešķirt formas elementam vizuālu noformējumu.
- Dati netiks nosūtīti uz serveri, tiks parādīts kļūdas paziņojums.



RegEx

RegEx (regular expressions)

RegEx(regular expressions) ir simbolu izteiksmes ar kurām meklējam sakritības tekstā.

Plaši izmantots rīks dažādās programmēšanas valodās. Tas mums palīdz atrast meklētas sakritības ar norādīto simbolu izteiksmi un atgriezt vērtību pārbaudīt, validēt, aizvietot vai veikt jebkuru citu iespējamu darbību ar tekstu jeb string tipa vērtību.

[Ievads RegEx \(interaktīvs\)](#)

[Ievads RegEx \(teksts\)](#)

[MDN RegEx resurss](#)

Reg[ular]
Ex[pression]

RegExp deklarēšana

RegExp izteiksmi iespējams deklarēt JavaScript divos veidos:

```
let regEx = /abc/;
```

Šis risinājums ir viēglāk pierakstāms, bet derīgs tikai statiskai RegExp izteiksmei, kas nesatur mainīgo vērtību.

```
let regEx = new RegExp('abc');
```

Inicializējot jaunu RegExp klases instanci, iespējams izmantot dinamiskas izteiksmes, piem, ja izteiksme tiks mainīta aplikācijas izpildes laikā vai veidota, balstoties uz ievades lauka vērtību.

RegEx izteiksmju darbība un sintakse

Izteiksme `/abc/` atradīs sakritību tekstos "Hi, do you know your **abc**'s?" un "The latest airplane designs evolved from **slabc**rft.", bet ne tekstā "Grab **crab**".

Tiks atrastas tikai strikti izteiksmei identiskas sakritības.

Ar dažādu ne alfabētisko simbolu palīdzību varam veidot izteiksmes plašākam meklējamo vērtību lokam.

Piemēram, lai atrastu tekstā tieši vienu a burtu, kam seko neierobežots skaits ar burtu b un tam seko tieši viens burts c, varētu izmantot izteiksmi `/ab*c/`. Tekstā

"**cbbabbbbbc**debc" būtu viena sakritība ar šo izteiksmi.

Izteiksme `/waz{3,5}up/` norādīts, ka meklējam sakritību, kas satur 3-5 secīgus z burtus.

Teksti "**wazzzzzup**" un "**wazzzup**" pilnībā atbilst meklētajam, bet ne teksts "wazup".

Varam meklēt arī skaitļu un teksta kombinācijas. Izteiksme `/[A-Z]{2}\s[0-9]{7}/` meklēs 2 lielos burtus `[A-Z]{2}`, kam seko atstarpe `\s` un septiņi cipari no 0 līdz 9 `[0-9]{7}`. Šādi mēs varētu pārbaudīt kādas valsts pasta indeksu un atrast sakritību, piem., "**CA 6782344**".

RegEx izteiksmes var kļūt diez gan sarēģītas un grūti lasāmas, piem., izteiksme epasta adreses pārbaudei:

`/^[a-zA-Z0-9.!#$%&'*/+=? ^_` {}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/`

[Rīks RegEx izteiksmju darbības pārbaudei](#)

RegEx metodes

exec()

Atgriež masīvu vai null, ja neviena nav atrasta.

test()

Atgriež true vai false.

match()

Atgriež masīvu ar visām sakritībām vai sakritību grupām vai null, ja neviena nav atrasta.

matchAll()

Atgriež iterator objektu ar visām sakritībām un sakritību grupām.

search()

Atgrieš atrastās sakritības indeksu tekstā vai -1, ja tāda nav atrasta.

replace()

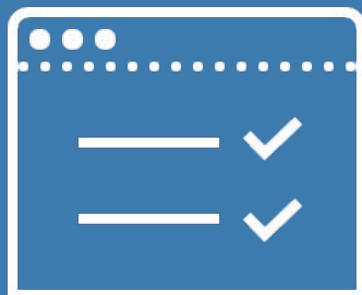
Meklē sakritību tekstā un aizvieto to ar norādīto vērtību.

replaceAll()

Meklē vairākas sakritības tekstā un aizvieto tās ar norādīto vērtību.

split()

Sadala tekstu masīvā, daloto masīva elementos pēc norādītās RegEx izteiksmes sakritības.



Datu validācija ar JavaScript

JavaScript rīki validācijai

Sarežģītākās formās var papildināt HTML validāciju vai pilnībā to aizvietot ar JavaScript. Tas mums dod lielāku kontroli pār notikumiem un datiem, apstrādājot ievades lauku vērtības.

Datu validācijai varam izmantot bibliotēkas, veidot paši savas funkcijas vai izmantot JS integrētās metodes šīm darbībām jeb [Constraint Validation API](#).



Integrētās validācijas metodes

Dotajiem HTML elementiem **<button>**, **<fieldset>**, **<input>**, **<output>**, **<select>**, **<textarea>** ir pieejams sekojošās īpašības:

validationMessage - atgriež paziņojumu par nesakritību datu formātā ar nepieciešamo vai tukšu string vērtību, ja dati atbilst kritērijiem.

validity - atgriež objektu ar detalizētu apskatu par datu atbilstību nepieciešamajiem kritērijiem.

willValidate - atgriež **true**, ja lauks tiks validēts pirms vērtību nosūtīšanas serveri vai **false**, ja tas netiks darīts.

Un vairāks metodes validācijas darbībām:

checkValidity() - atgriež **true**, ja dati atbilst uzstādītajiem kritērijiem vai **false**, ja neatbilst un izpilda "invalid" notikumu attiecīgajiem elementiem

reportValidity() - izpilda kļūdas paziņojumu par formas ievades laukiem, kuru dati neatbilst kritērijiem.

setCustomValidity(message) - uzstāda pielāgotu kļūdas paziņojumu attiecīgajam ievades laukam, ja dati neatbilst kritērijiem.

VUMC

VADĪBAS UN
UZŅĒMĒJDARBĪBAS
MĀCĪBU CENTRS

NACIONĀLAIS
ATTĪSTĪBAS
PLĀNS 2020



EIROPAS SAVIENĪBA
Eiropas Sociālais
fonds

I E G U L D Ī J U M S T A V Ā N Ā K O T N Ē

ESF projekts Nr. 8.4.1.0/16/l/001 "Nodarbināto personu profesionālās kompetences pilnveide"

VUMC

VADĪBAS UN
UZŅĒMĒJDARBĪBAS
MĀCĪBU CENTRS

Programmas nosaukums