

Walidacja i wysłanie formularza

atrybuty dot. walidacji

- **required** - pole nie może być puste
- **minlength** i **maxlength** - minimalna i maksymalna liczba znaków dla tekstowych pól
- **min** i **max** - minimalna lub maksymalna wartość dla pól liczbowych i dat/czasu
- **type** - np. e-mail czy liczba
- **pattern**: wyrażenie regularne (*regular expression*)

pattern

- wyrażenie regularne
- czyli wzór - kombinacje znaków
- idealne do walidacji
- przykłady
 - https?
 - http lub https
 - [0-9]{2}-[0-9]{3}
 - kod pocztowy - dwie cyfry, myślnik, 3 cyfry
 - (?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{8,}
 - co najmniej 8 znaków, przynajmniej jedna mała litera, jedna wielka i jedna cyfra
- nie zastępuje required

Ważne sprawy związane z walidacją

- prostą walidację możemy wykonać za pomocą samego HTML-a
- bardzo ważne jest, by wszystkie wymagane pola były wypełnione i miały właściwy format przed wysłaniem formularza
- walidację przed wysłaniem formularza nazywamy walidacją po stronie klienta
- walidacja przed wysłaniem formularza jest wygodna dla użytkownika
- pomożemy użytkownikowi wypełnić prawidłowo formularz
- walidowanie formularzy po stronie klienta i po stronie serwera jest kosmicznie ważne jeśli chodzi o obronę przed potencjalnymi atakami hakerów

Wysyłamy formularz

- formularz jest po prostu wygodnym sposobem na przygotowanie żądania **HTTP**
- do serwera dane z formularza zostaną wysłane jako pary **klucz=wartość**
- np. **firstName=Krzysiek&lastName=D%C4%85browski**

metoda GET

- dane zostaną dopisane do adresu URL po znaku ?
- ciało żądania jest puste
- używamy, kiedy serwer ma coś zwrócić, a nie zapisać

metoda POST

- dane zostaną wysłane w ciele żądania HTTP (body)
- używamy np. kiedy chcemy poprosić serwer o zapisanie czegoś w bazie danych
- przeglądarka dodaje nagłówki dot. treści np. Content-Type i Content-Length

po stronie serwera

- Każdy backendowy język programowania ma swój mechanizm do odbioru danych z formularzy
- Serwer z tymi danymi może zrobić dowolne rzeczy
 - zwrócić dane
 - dodać nowe dane np. do bazy danych
 - zaktualizować dane
 - wysłać maila
 - itd.

Bezpieczeństwo

- formularze są bardzo popularnym sposobem ataku na serwery
- używaj protokołu **HTTPS**
- nigdy nie używaj **GET** do przesyłania hasła
- serwer również musi sprawdzić dane
- serwer nigdy nie powinien ufać użytkownikowi
- proś tylko o te dane, których potrzebujesz

Wysyłanie plików

- enctype="multipart/form-data"
- method="POST"
- serwery mogą mieć limity wielkości pliku lub żądania

Linki

Specyfikacja

<https://html.spec.whatwg.org/multipage/forms.html>

MDN

<https://developer.mozilla.org/en-US/docs/Learn/Forms>

Test żądania HTTP

<https://postman-echo.com/>

Test wyrażeń regularnych

<https://regexr.com/>