

## Wymogi technologiczne:

- Użycie biblioteki Lit (dawniej lit-html/lit-element) - <https://lit.dev>
- Użycie biblioteki Lion - <https://lion-web.netlify.app>
- JavaScript w wersji ES6
- Aplikacja działa w przeglądarce Chrome

## Treść zadania:

Napisz aplikację pozwalającą na generowanie, walidowanie oraz zarządzanie widocznością pól formularza na podstawie JSONa dostarczonego z aplikacji backendowej.

Przykładowy JSON:

**GET** /api/simple-form

```
{
  "name": {
    "type": "text",
    "validators": ["required", "max-len:20", "min-len:3"],
    "visibility": "always"
  },
  "surname": {
    "type": "text",
    "validators": ["required", "max-len:20", "min-len:3"],
    "visibility": "always"
  },
  "nationality": {
    "type": "select",
    "validators": ["required"],
    "dataset": ["PL", "GB", "DE"],
    "visibility": "always"
  },
  "pesel": {
    "type": "text",
    "validators": ["required", "len:11"],
    "visibility": "nationality === PL"
  }
}
```

Po wprowadzeniu nieprawidłowej wartości, pod polem powinien wyświetlić się wynik walidacji:

imie

aa

Wartość pola imie nie osiągnęła minimalnej ilości znaków. Minimalna długość: 3

Jeśli na formularzu nie zostały wprowadzone wymagane dane, to po kliknięciu przycisku zapisz pod obowiązkowymi polami powinny wyświetlić się komunikaty:

Formularz z dwoma polami tekstowymi i przyciskiem. Pole 'imie' ma komunikat 'Wartość pola imie jest wymagana'. Pole 'nazwisko' ma komunikat 'Wartość pola nazwisko jest wymagana'. Przycisk 'Zapisz' jest szary i nieaktywny.

imie	<input type="text"/>
Wartość pola imie jest wymagana	
nazwisko	<input type="text"/>
Wartość pola nazwisko jest wymagana	
<input type="button" value="Zapisz"/>	

Jeżeli cały formularz zostanie uzupełniony poprawnie to po kliknięciu **zapisz** przycisk powinien stać się nieaktywny a aplikacja wysłać POST na backend. Po zwróceniu 200 wyświetli się komunikat z informacją o powodzeniu zapisu.

Formularz z wypełnionymi polami: imie: Jan, nazwisko: Kowalski, narodowość: PL. Przycisk 'Zapisz' jest szary. Na dole znajduje się zielony komunikat 'Zapisano formularz'.

imie	Jan
nazwisko	Kowalski
narodowość	PL ▼
<input type="button" value="Zapisz"/>	
Zapisano formularz	

Analogicznie w przypadku błędu po stronie backendowej do wyświetlenia komunikat o błędzie.

Formularz z wypełnionymi polami: imie: Jan, nazwisko: Kowalski, narodowość: PL. Przycisk 'Zapisz' jest szary. Na dole znajduje się czerwony komunikat 'Zapis nie powiódł się'.

imie	Jan
nazwisko	Kowalski
narodowość	PL ▼
<input type="button" value="Zapisz"/>	
Zapis nie powiódł się	

Obiekty w JSON'ie mogą się składać z pól:

- type – rodzaj pola
- validators – walidatory
- visibility – warunek widoczności
- dataset – dane (np. dla select, radio-group)

Typy pól, które powinien generować formularz to:

- text
- textarea
- select
- checkbox
- radio-group

Minimalna lista walidatorów, które powinny być zaimplementowane w aplikacji:

- required – wartość nie może być pusta
- min-len – ilość znaków/elementów w liście nie może być mniejsza niż...
- max-len – ilość znaków/elementów w liście nie może być większa niż...
- len – ilość znaków/elementów w liście musi być równa...

## Uproszczenia

- Wygląd aplikacji nie jest istotny. Kod powinien być czytelny oraz otwarty na rozszerzanie.
- Nie ma wymogu implementacji części backendowej (sposób mockowania dowolny).

Preferowanym sposobem dostarczenia rozwiązania jest skompresowany plik zip lub link do publicznego repozytorium.