

## Критерії приймання

- Створено репозиторій `goit-js-hw-10`.
- Домашня робота містить два посилання: на вихідні файли і робочу сторінку на `GitHub Pages`.
- В консолі відсутні помилки і попередження під час відкриття живої сторінки завдання.
- Проект зібраний за допомогою [parcel-project-template] (<https://github.com/goitacademy/parcel-project-template>).
- Код відформатований за допомогою `Prettier`.

## ## Стартові файли

[Завантажуй стартові файли] (<https://minhaskamal.github.io/DownGit/#/home?url=https://github.com/goitacademy/javascript-homework/tree/main/v2/10/src>)

з базовою розміткою та стилями завдання. Скопіюй їх собі в проект, повністю замінивши папку `src` у

[parcel-project-template] (<https://github.com/goitacademy/parcel-project-template>).

## ## Завдання - пошук країн

Створи фронтенд частину програми пошуку даних про країну за її частковою або повною назвою. Подивися демо-відео роботи програми.

<https://user-images.githubusercontent.com/17479434/131147741-7700e8c5-8744-4eea-8a8e-1c3d4635248a.mp4>

## ### HTTP-запит

Використовуй публічний API [Rest Countries v2] (<https://restcountries.com/>), а саме [ресурс name] (<https://restcountries.com/#api-endpoints-v3-name>), який повертає масив об'єктів країн, що задовольнили критерій пошуку. Додай мінімальне оформлення елементів інтерфейсу.

Напиши функцію `fetchCountries(name)`, яка робить HTTP-запит на [ресурс name] (<https://restcountries.com/#api-endpoints-v3-name>) і повертає проміс з масивом країн - результатом запиту. Винеси її в окремий файл `fetchCountries.js` і зроби іменований експорт.

## ### Фільтрація полів

У відповіді від бекенду повертаються об'єкти, велика частина властивостей яких, тобі не знадобиться. Щоб скоротити обсяг переданих даних, додай рядок параметрів запиту - таким чином цей бекенд реалізує фільтрацію полів. Ознайомся з [документацією синтаксису фільтрів] (<https://restcountries.com/#filter-response>).

Тобі потрібні тільки наступні властивості:

- `name.official` - повна назва країни
- `capital` - столиця
- `population` - населення
- `flags.svg` - посилання на зображення прапора
- `languages` - масив мов

### ### Поле пошуку

Назву країни для пошуку користувач вводить у текстове поле `input#search-box`. HTTP-запити виконуються при введенні назви країни, тобто на події `input`. Але робити запит з кожним натисканням клавіші не можна, оскільки одночасно буде багато запитів і вони будуть виконуватися в непередбачуваному порядку.

Необхідно застосувати прийом `Debounce` на обробнику події і робити HTTP-запит через `300мс` після того, як користувач перестав вводити текст. Використовуй пакет `[lodash.debounce]` (<https://www.npmjs.com/package/lodash.debounce>).

Якщо користувач повністю очищає поле пошуку, то HTTP-запит не виконується, а розмітка списку країн або інформації про країну зникає.

Виконай санітизацію введеного рядка методом `trim()`, це вирішить проблему, коли в полі введення тільки пробіли, або вони є на початку і в кінці рядка.

### ### Інтерфейс

Якщо у відповіді бекенд повернув більше ніж 10 країн, в інтерфейсі з'являється повідомлення про те, що назва повинна бути специфічнішою. Для повідомлень використовуй [бібліотеку `notiflix`] (<https://github.com/notiflix/Notiflix#readme>) і виводь такий рядок

```
"Too many matches found. Please enter a more specific name."
```

![Too many matches alert] (<https://raw.githubusercontent.com/goitacademy/javascript-homework/main/v2/10/preview/too-many-matches.png>)

Якщо бекенд повернув від 2-х до 10-и країн, під тестовим полем відображається список знайдених країн. Кожен елемент списку складається з прапора та назви країни.

![Country list UI] (<https://raw.githubusercontent.com/goitacademy/javascript-homework/main/v2/10/preview/country-list.png>)

Якщо результат запиту - це масив з однією країною, в інтерфейсі відображається розмітка картки з даними про країну: прапор, назва, столиця, населення і мови.

![Country info UI] (<https://raw.githubusercontent.com/goitacademy/javascript-homework/main/v2/10/preview/country-info.png>)

:::caution Увага

Достатньо, щоб застосунок працював для більшості країн. Деякі країни, як-от `Sudan`, можуть створювати проблеми, оскільки назва країни є частиною назви іншої країни - `South Sudan`. Не потрібно турбуватися про ці винятки.

:::

### ### Обробка помилки

Якщо користувач ввів назву країни, якої не існує, бекенд поверне не порожній масив, а помилку зі статус кодом `404` - не знайдено. Якщо це не обробити, то користувач ніколи не дізнається про те, що пошук не дав результатів. Додай повідомлення `"Oops, there is no country with that name"` у разі помилки, використовуючи

[бібліотеку notiflix] (<https://github.com/notiflix/Notiflix#readme>).

![Error alert] (<https://raw.githubusercontent.com/goitacademy/javascript-homework/main/v2/10/preview/error-alert.png>)

:::caution Увага

Не забувай про те, що `fetch` не вважає 404 помилкою, тому необхідно явно відхилити проміс, щоб можна було зловити і обробити помилку.

:::