

9 Opakování: LeviExpress, první část

Všechno, co jsme se v Reactu naučili, si procvičíme na projektu takzvaně „ze života“.

< Předchozí
React Router

Následující >
Opakování: LeviExpress, druhá část

2 Výběr města

Na konci tohoto cvičení si uživatel bude schopen vybrat startovní a cílové město ze seznamu měst stažených z API. Vytvoříte komponentu `CityOptions`, která dostane na vstupu pole se seznamem destinací a vytvoří z něj elementy `<option>` do `select` ů pro výběr výchozího a cílového města. Seznam měst se bude stahovat v komponentě `JourneyPicker` z API endpointu [/api/cities](#). Prohlédněte si strukturu dat, která endpoint vrací.

1. Přímo v souboru `index.jsx` komponenty `JourneyPicker` si vytvořte novou komponentu `CityOptions`. Přesuňte do ní všechny `<option>` ze `select` ů pro výběr výchozího města. Komponentu `CityOptions` použijte v obou `select` ech pro výběr města. Zkontrolujte, že v prohlížeči vše funguje stejně.
2. V komponentě `JourneyPicker` si vytvořte pomocí `useState` další stav `cities`. V tomto stavu bude seznam měst, mezi kterými lze cestovat. Pro otestování nastavte do seznamu dvě města:

```
[
  { name: 'Praha', code: 'CZ-PRG' },
  { name: 'Brno', code: 'CZ-BRQ' },
];
```

3. Komponenta `CityOptions` bude v `props` přijímat `cities`. V něm bude pole – seznam měst, které může uživatel vybrat. Upravte tedy příslušným způsobem hlavičku komponenty `CityOptions` a získanou property `cities` si vypíšte do konzole.

4. Na obou místech, kde máte komponentu `CityOptions` použitou, nastavte property/atribut `cities` na hodnotu stavu `cities`. V prohlížeči se teď při zobrazení stránky vypíše dvakrát do konzole seznam měst – Praha a Brno.
5. Upravte komponentu `CityOptions` tak, aby se na základě seznamu měst v property `cities` vypsaly jednotlivé `<option>`. Pro text `option` se použije název destinace, jako hodnota (atribut `value`) a také jako klíč pro React (`key`) se použije kód destinace.
6. Nechceme, aby při zobrazení stránky byla rovnou vybraná první města ze seznamu. Proto před elementy `option` vytvořené z pole ještě ručně vložíme jeden `option` s textem „Vyberte“. Atribut `value` tohoto `option` bude prázdný.
7. Odstraňte z komponenty `CityOptions` výpis do konzole a zkontrolujte v prohlížeči, že se v seznamu výchozích i cílových měst zobrazuje Praha a Brno a že stále správně funguje výběr měst.
8. Do hlavní komponenty `JourneyPicker` přidejte `useEffect`, který se bude volat při prvním zobrazení komponenty. Přesuňte do něj nastavení stavu `cities` – naše dvě testovací města. Výchozí stav pro `cities` tedy bude prázdné pole, teprve `useEffect` nastaví seznam měst na Prahu a Brno. Ověřte v prohlížeči, že se v `select` ech stále zobrazují obě města. Dejte pozor na to, aby se efekt volal opravdu jen při prvním zobrazení komponenty – můžete si to ověřit pomocným výpisem do konzole prohlížeče, který se musí objevit jen jednou – když budete překlikávat na jiná města, výpis už se nebude opakovat.
9. Upravte `useEffect` tak, že bude seznam měst získávat z API. Endpoint je na adrese <https://apps.kodim.cz/daweb/leviexpress/api/cities> a vrací seznam měst jako JSON ve formátu, který jsme použili výše. Získaná data použijte místo Prahy a Brna ve stavu `cities`. Ověřte v prohlížeči, že se v seznamu měst objeví i další města.
10. Commitněte změny.

Verze 2.0.0-beta.7