

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

4 Vyhledání spoje

V tomto cvičení dokončíte komponentu pro vyhledání spojení. V komponentě už funguje výběr výchozího a cílového města a také data cesty. Nyní napojíte komponentu na API pro vyhledávání spojení.

1. Pokračujte v komponentě `JourneyPicker`. Do tlačítka „Vyhledat spoj“ přidejte atribut `disabled` tak, aby tlačítko bylo povolené pouze v případě, že jsou vybrána obě města i datum.
2. Při kliknutí na tlačítko „Vyhledat spoj“ se volá funkce `handleSubmit`, která vypíše údaje zadané uživatelem. Nyní výpis do konzole nahradíte voláním API. Bude se volat následující API endpoint
<https://apps.kodim.cz/daweb/leviexpress/api/journey?fromCity=...&toCity=...&date=...>
Vytečkovaná místa se nahradí hodnotami vybranými uživatelem, které jsou uloženy ve stavech `fromCity`, `toCity` a `date`.
3. Volání tohoto API vrací JSON s nalezenými spoji. Vypište si výstup do konzole prohlížeče.
4. Nalezená spojení budeme potřebovat zobrazit v další komponentě na stránce. Potřebujeme je tedy poslat z komponenty `JourneyPicker` jejímu rodiči – to uděláme v následujících krocích.
5. V komponentě `Home` si připravte funkci `handleJourneyChange`. Funkce `handleJourneyChange` bude očekávat jeden parametr – objekt s údaji o nalezeném spojení. Nazvěte jej třeba `journey`. Ve funkci zatím vypište tento parametr do konzole. Funkce se zatím nebude nikde volat – volání zajistíme v

následujících krocích, kde bude funkce `handleJourneyChange` sloužit jako callback předaný do komponenty `JourneyPicker`.

6. V komponentě `Home` používáte komponentu `JourneyPicker`. Této komponentě předejte property jménem `onJourneyChange`, jako hodnotu jí předejte funkci (callback) `handleJourneyChange`.
7. V komponentě `JourneyPicker` bude property `onJourneyChange`, do které rodič (`Home`) vkládá funkci, která se zavolá s údaji nalezeném spoji. Všimněte si, že v hlavičce komponenty `JourneyPicker` už je property `onJourneyChange` připravená.
8. Ve funkci `handleSubmit` v komponentě `JourneyPicker` máte nyní výpis nalezených spojení do konzole prohlížeče. Tento výpis nahraďte voláním funkce uložené v property `onJourneyChange`, které jako paramter předáte data získaná z volání API pod klíčem `results`.
9. Vraťte se do komponenty `Home`, ze které se volá komponenta `JourneyPicker`. V komponentě `Home` vytvořte pomocí `useState` nový stav `journey`, výchozí hodnota bude `null`.
10. Propojte komponentu `JourneyPicker` se stavem `journey` – když je volán callback `handleJourneyChange` s údaji o nalezeném spoji, nastaví se toto spojení do stavu `journey`.
11. Upravte komponentu `Home` tak, aby v případě, kdy ve stavu `journey` je nějaké spojení, vypsala pod vyhledávací formulář text „Nalezeno spojení s id ...“. Místo tří teček bude `journeyId` z dat o nalezeném spojení.
12. Ověřte, že funguje výběr měst a data, že po zadání všech třech údajů můžete kliknout na „Vyhledat spoj“ a že se po kliknutí vyhledá nějaké spojení a vypíše se do stránky jeho id.
13. Commitněte změny.