

primaTime frontend úkol – ComboBox

Cílem úkolu je ověřit vaše znalosti TypeScriptu, Reactu a schopnost implementace z některých knihoven na základě dokumentace (pokud je neznáte). Vaším úkolem bude vytvořit ComboBox komponentu dle zadání.

Jaké technologie musíte použít?

Musíte použít vyžadované technologie a knihovny/balíčky, nicméně nejste omezeni tím, že použijete nějakou další knihovnu, kterou znáte. Níže je seznam vyžadovaných technologií:

- [TypeScript](#) – správné typování – nechceme, abyste použili TypeScript a všude typovali pomocí any
- [React Hook Form](#) nebo podobná knihovna na správu formulářů
- [Axios](#)
- [TanStack Query](#)

Zadání

Krok č. 1 – základní textový input

Pro vytvoření samotného ComboBoxu budeme nejprve potřebovat vytvořit komponentu s obyčejným textovým inputem. Taková komponenta musí mít rozhraní, které půjde napojit na React Hook Form (nebo obdobnou knihovnu, viz výše), komponenta musí být ale zároveň použitelná i mimo formulářovou knihovnu (tedy mimo kontext formuláře, jako samostatná entita). Počítejte s tím, že tato komponenta bude poté součástí další formulářové komponenty (ComboBoxu). Grafický design je čistě na vás, nicméně komponenta musí splňovat následující schéma:



Tato komponenta zároveň musí mít stavy disabled a error. V disabled stavu ji uživatel sice uvidí, nebude ale moci upravit uživatelský vstup a zároveň komponenta “zešedne”, aby bylo jasné, že je disabled. V error stavu komponenta zčervená a pod ní se zobrazí chybová hláška. Jak již bylo zmíněno, tak takto musí být možnost komponentu napojit na React Hook Form (nebo obdobnou knihovnu na formuláře), aby zobrazovala chybový stav z formulářové knihovny, zároveň ale komponenta musí být použitelná i samostatně a to včetně nastavení chybového a disabled stavu. Náповěda jak toto zařídit může být vytvoření další komponenty, která poskytne rozhraní mezi formulářovou knihovnou a touto komponentou, tedy pokud budeme chtít textové pole použít ve formuláře knihovně, tak jej ještě obalíme do další komponenty.

Krok č. 2 – získání dat pro ComboBox

Ještě předtím, než budeme vytvářet samotný ComboBox, tak použijeme Axios a TanStack Query knihovnu, abychom si připravili získání dat pro ComboBox. Využijte tedy tyto 2 knihovny (Axios a TanStack Query) k tomu, abyste získaly potřebná data z endpointu, který získá seznam univerzit. Endpoint má následující tvar:

[http://universities.hipolabs.com/search?country=Czech+Republic&name=\[name\]](http://universities.hipolabs.com/search?country=Czech+Republic&name=[name])

Kde [name] je mezinárodní název univerzity (v angličtině), ze kterého chceme vyhledat seznam univerzit. Více o tomto endpointu naleznete na adrese <https://github.com/Hipo/university-domains-list>. Parametr country nechme na Czech Republic, budeme používat pouze univerzity v České republice.

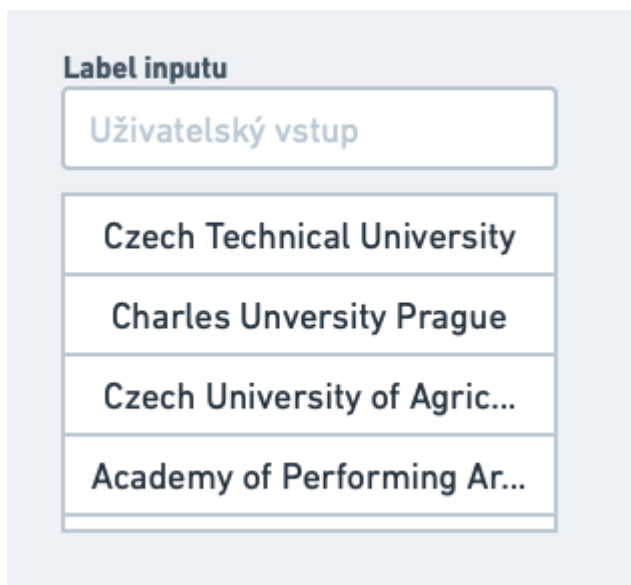
Příklad použití (pro získání ČVUT (Czech Technical University) stačí použít v parametru pouze Czech Technical):

<http://universities.hipolabs.com/search?country=Czech+Republic&name=Czech+Technical>

Připravte si to tak, abyste parametr [name] mohli dynamicky měnit na základě vstupu uživatele.

Krok č. 3 – ComboBox

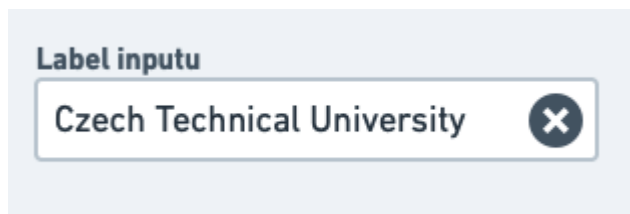
Nyní využijte připravený textový input a připravené získávání dat pro to, abyste vytvořili ComboBox komponentu. Základem této komponenty bude textový input, do kterého pokud začne uživatel psát, tak se jeho uživatelský vstup použije jako [name] parametr v endpointu pro získání univerzit (krok č. 2). Získaná data se použijí k tomu, že se u inputu zobrazí panel (např. pomocí knihovny [Floating UI](#) nebo jiné), který zobrazí seznam výsledných univerzit na základě uživatelského vstupu. Jelikož se data získávají pomocí dotazu na server, tak nezapomeňte zobrazit i načítání. Panel bude obsahovat seznam položek, opět nezáleží na grafickém designu, ale mělo by to kopírovat následující schéma:



The diagram illustrates a ComboBox component. It features a text input field at the top with the placeholder text "Uživatelský vstup". Below the input field is a dropdown menu that is currently open, displaying a list of university names. The names listed are "Czech Technical University", "Charles University Prague", "Czech University of Agric...", and "Academy of Performing Ar...". The dropdown menu has a light blue background and a thin border.

Pokud uživatel provede focus do inputu, který je prázdný, tak by se panel měl zobrazit také, ovšem v takovém případě zobrazí všechny univerzity z České republiky. Pokud uživatel smaže svůj vstup, tak v takovém případě se také zobrazí všechny univerzity z České republiky. Panel by měl v jednotlivých položkách zobrazovat název univerzity (z příloženého endpointu v kroku č. 2). Panel by měl být stejně široký jako je šířka inputu, přičemž pokud se do něj nevejde celý název univerzity, tak se zobrazí na konci 3 tečky (...), viz obrázek výše). Panel by se měl automaticky přizpůsobovat oknu prohlížeče, tzn., že se automaticky zobrazí buď pod inputem nebo nad ním na základě volného místa (při scrollování), navíc pokud je počet položek delší, než se vejde do okna, tak by se měl v panelu zobrazit vertikální posuvník (stejně jako funguje obyčejná select komponenta).

Každá položka z panelu půjde kliknutím vybrat. Pokud bude vybraná položka, tak ji ComboBox zobrazí následovně:



Uživatel ale nemusí vybrat žádnou položku, pouze klikne mimo panel a tím se panel schová.

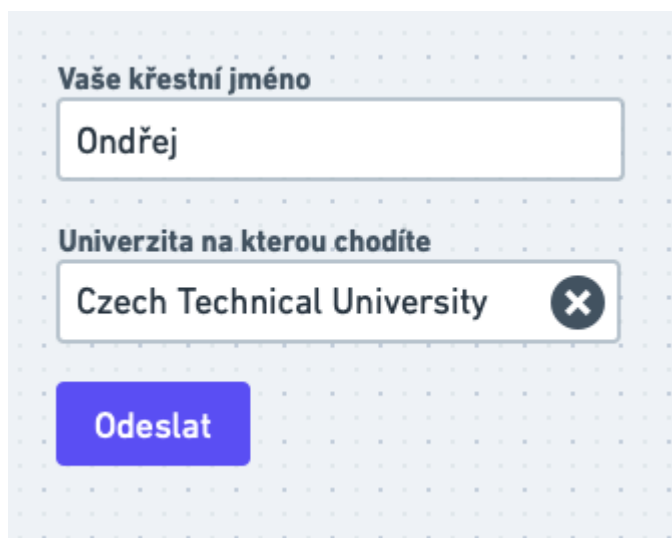
Kliknutím na křížek v pravém rohu (viz obrázek výše) půjde vybranou položku odstranit. Text nalevo (v obrázku Czech Technical University) půjde přepsat. V případě přepisování se opět zobrazí panel, který bude zobrazovat univerzity na základě uživatelského vstupu, přičemž při výběru univerzity se vybraná položka změní na nově vybranou. Pokud uživatel nevybere žádnou položku (klikne mimo), tak se do textového pole vrátí název již vybrané hodnoty.

ComboBox stejně jako textový input musí mít ještě stav disabled a error. Tedy při disabled stavu uživatel nemůže upravit hodnotu (a “zešedne”) a při error stavu ComboBox zčervená a zobrazí se pod ním chybová hláška.

Nezapomeňte na to, že stejně jako textový input z kroku č. 1 i ComboBox musí fungovat mimo formulářovou knihovnu. Zároveň si to dobře promyslete, abyste hodnotu ComboBoxu ukládali externě, například do zmíněné formulářové knihovny opět pomocí nějaké komponenty, kterou jej obalíte.

Krok č. 4 – formulář

Obě komponenty (textový input a ComboBox) použijte pro vytvoření formuláře. Formulář bude splňovat následující schéma:



Bude používat již zmíněnou formulářovou knihovnu. Přičemž po odeslání formuláře dojde nejprve k frontendové validaci (přes zvolenou formulářovou knihovnu), která zkontroluje, zda uživatel vyplnil křestní jméno (textový input). V případě, že křestní jméno nevyplnil, tak se tento textový input dostane do error stavu a pod ním se zobrazí hláška “Toto pole je povinné”. Pokud formulář projde validací, tedy uživatel vyplnil křestní jméno. Tak se zobrazí alert okno (v JavaScriptu alert funkce). A v tomto okně bude napsané: Jméno: Ondřej, Univerzita: Czech Technical University. Pokud uživatel nevyplní univerzitu, tak potom: Jméno: Ondřej, Univerzita: nevyplněno.

Závěr

Bud'te kreativní! Výslednou práci můžete okořenit nějakými dalšími věcmi, ale zase to nepřehánějte. Projekt nahrajte do nějakého veřejného repozitáře, např. GitHub a zašlete nám na něj odkaz.

ComboBox musíte sestavit podle zadání, nesmíte použít knihovnu s již hotovým (cílem je ověřit vaše myšlení a jak se k tomu postavíte).