

15 SUPABASE a služby a databázy

pre web appky potrebuješ kombo server + klient + databázu. všetko rôzne technológie, každá vyžaduje iný programovací jazyk a má vlastné zákutia a všetky spolu musia komunikovať a je toho veľa. je to veľa roboty. preto existujú kolegovia. ale zároveň **existujú služby!** ktoré robotu robia za teba. ale za cenu.

ja som ukázal službu <https://supabase.com/>, ktorá vie byť tvoj backend. ponúka databázu, používateľov, všeličo. preštuduj si, čo ponúka. a preštuduj si, za akú cenu. netvrdím, že supabase musíš používať. ja som to napríklad na nič reálne nepoužil. len na bastlenie 😊 keď ti ukazujem veci, ber to tak, že hovorím *"aha, aj toto existuje!"* a nehovorím *"aha, toto teraz musíš používať!"* napríklad ja mám radšej svoju vlastnú databázu sám u seba 😊

ale je dosť možné, že aj v práci budeš používať podobné služby. preto považujem za dôležité ukázať, že služby existujú, s čím ti vedia pomôcť a cca ako sa používajú. a ako sa používajú? presne tak, **to vždy nájdeš v dokumentácii** 😊

na ukážku toho, ako sa služby používajú, som zvolil supabase. služby bývajú platené, ale často majú aj svoj free model. preštuduj si a možno zistíš, že tvoj projekt spadá do kategórie, že supabase bude pre teba zdarma. ale zároveň zaruč, že tvoj projekt nikdy nebude príliš úspešný! lebo potom si zaklopú na dvere a budú chcieť money.

ale hlavne vedz, že vo web svete VŽDY existujú alternatívy. napríklad ja som tomu teraz dal tak 3,5 sekundy googlenia a zistil, že existuje <https://appwrite.io/> 🤖 a všetci kopírujú googlacky firebase. ak chceš mať niečo podobné, ale mať to doma u seba, existuje <https://pocketbase.io>.

zároveň služby nie sú jediný spôsob, ako robiť backend. taký jednoduchý možno ešte spravím použitím node + sqlite, uvidíme. každopádne existujú alternatívy!

to znamená, že keď ja ti niečo ukážem, NEPOUŽÍVAJ TO 😊 keď ja ti niečo ukážem, toto by sa malo stať v tvojej hlave: *"aha, takže toto existuje. dobre vedieť. teraz si otvorím internet a pozriem si, aké existujú alternatívy a vyberiem si z nich tú, ktorá vyhovuje mne a ktorá je mne sympatická!"*

ak máš prístup k serveru (napríklad ak máš tzv. VPN), vieš tam nahráť <https://pocketbase.io/> a máš to zdarma. ak použiješ laravel, rails, django... buď tam rovno máš alebo si ľahko poriešiš backend. ak si vyhľadáš *"fullstack framework"* nájdeš kopu technológií, ktoré ti poskytnú frontend aj backend.

ale ja som **na ukážku vyrobil appku, ktorá používa službu supabase.** ak ju chceš rozbehať...

SUPABASE

musíš si vytvoriť účet na <https://supabase.com> stránke. tam vytvoríš projekt. a vyklikáš si databázu. a na konci klikania supabase vytvorí pre teba unikátne **Project URL** a **API Key**. stiahni si môj kód. v súbore **database.js** nájdeš 2 premenné a budú prázdne. **ty budeš musieť tvoj Project URL skopírovať do premennej supabaseUrl a tvoj unikátny API Key do premennej supabaseKey.**

každý z vás bude mať vlastné hodnoty a nemôžeš používať moje!! 😊 ber to jak prihlasovacie údaje. supabase ti ich vyrobí iba pre teba. ty ich skopíruješ. ale zvyšok kódu ti dám. a bude to hotová appka, ktorá

vie: **zobraziť všetky cardy, jeden konkrétny card, vytvoriť nový card, upraviť card, vymazať card**. čiže máš pokrytý celý **CRUD**. čiže všetko, čo potrebuješ, na každý reálny projekt. lebo **[C]**reate, **[U]**pdate, **[R]**ead, **[D]**elete.

pôjdeš takto nejak:

- zaregistruj sa na <https://supabase.com>
- vytvor **New project** (vymysli mu názov, Central Europe) pár minút to bude pracovať, potom by ťa to malo hodiť na stránku so sekciou **Project API**, kde nájdeš tvoje unikátne **Project URL** a **API Key**. tie neskôr budeš kopírovať do môjho kódu.

a potom:

- vľavo v menu nájdeš **Table Editor**
- **Create new table**, názov zadaj **cards**
- **pre takúto hráčarsku mini-appku vypni Enable Row Level Security (RLS)**
- **add column**, column name daj **title**, type vyber **TEXT**
- **add column**, column name daj **content**, type vyber **TEXT**
- **id** a **created_at** tam nechaj, tých sa nechytaj
- **save**

a potom:

- vľavo v menu nájdeš **Table Editor**
- vľavo (ale o niečo menej vľavo) je zoznam tabuliek
- mala by tam byť tvoja novo vytvorená **cards**, klikni
- cez **Insert**, **Insert Row** vieš pridať nové dáta
- pridaj si tam zopár testovacích
- **vpravo hore je malý, ale svojou užitočnosťou dôležitý** button **API docs** s ukázkami kódu (ako presne v javascripte získaš dáta, ako pridáš, vymažeš...)

ak máš toto, môžeš použiť môj kód 🤖 kde do **main.js** len zadáš tvoje **Project URL** do premennej **supabaseUrl** a ten tvoj unikátny **API Key** do premennej **supabaseKey**. a samozrejme ten kód je teraz tvoj. všetok môj kód je tvoj kód, takže všetko môžeš upravovať ako len chceš. a rob to! lebo práve tým sa učíš 😊

a pozor: ty toto čítaš v budúcnosti. lebo tak funguje čas. tento postup fungoval vtedy, keď ja som písal tento text. je možné, že supabase web sa odvtedy zmenil. ale určite to zvládneš. a keď nie, tak vieš, čo máš robiť. to, čo máš vždy robiť. to, čo je tvoja reálna robota: a teda **čítať dokumentáciu a skúšať kým to nezačne fungovať** 😊

ID, CREATED_AT v DATABÁZACH

dôležitý dodatok k databázam 🧐 a toto platí pre všetky databázy, s ktorými sa v živote stretneš. nie iba pre supabase. prakticky každá tabuľka v každej databáze bude mať stĺpec **id** a často aj stĺpec **created_at**. takže ak do databázy ty pridávaš novú tabuľku s názvom **cards** a ona má evidovať 2 hodnoty (**title** a **content**) tak **v skutočnosti musí evidovať 4 hodnoty (**id**, **created_at**, **title**, **content**)** 😊

supabase, keď pridávaš tabuľku, ti pripraví stĺpce (po anglicky columns) pre **id** a **created_at**. a tie neprepisuj. nechaj si ich tam. a pridaj k nim nové stĺpce pre nové hodnoty, ktoré ty chceš evidovať. o hodnoty v **id** a **created_at** sa ty vôbec nemusíš starať. supabase ich vždy vyplní automaticky. napríklad ak ty píšeš kód, ktorý do databázy pridáva nový card, ty do databázy pošleš iba nový **title** a **content**. **id** a **created_at** si databáza vytvorí automaticky.

ak robíš povedzme eshop, pridáš si tabuľky ako **products** a **users** a **invoices** a každá jedna z tých tabuliek bude mať **id** a **created_at** column 😊

id je absolútne potrebné, to tam proste musí byť, literally najdôležitejší stĺpec v databázovej tabuľke, všetko sa točí okolo neho. bez **created_at** by sme asi občas vedeli fungovať, ale o každom zázname je extrémne užitočné vedieť kedy vznikol. takže ho tam nechaj 😊❤️ a ak budeš niekedy vytvárať vlastnú databázu a vlastné tabuľky a žiadne **created_at** tam nebude... tak ho tam pridaj! neoľutuješ.

a pozri si môj kurz **SQL databázy: MySQL a SQLite**, už vôbec neoľutuješ 😊
ak chceš pokročilejšie javascriptovanie, pozri si môj kurz **JavaScript a ES6** 💪

každopádne v supabase je to pohoda: pridá sa tie stĺpce sama a o hodnoty v nich sa databáza stará sama. sú prednastavené tak, že hodnotu v **id** databáza automaticky zvyšuje s každým pridaným záznamom a do **created_at** uloží dátum a čas v momente vzniku nového záznamu. čajíček ☕

MÔJ SLABÝ SUPABASE KÓD

je #soubory. s názvom **08-vite-supabase-test.zip**. to ešte na hodine vylepšíme. tento kód spustíš tak, že stiahneš zip, rozbalíš, otvoríš vo vscode, otvoríš tam terminal, napíšeš **npm install** a potom spustíš **npm run dev**. len aby to fungovalo, **supabase vyžaduje prihlasovacie údaje. a ty nebudeš používať moje** 😊 takže aby to fungovalo, podľa textu hore si budeš musieť vytvoriť tvoj vlastný supabase účet a projekt a potom do môjho kódu vložiť **supabaseUrl** a **supabaseKey**, ktoré supabase vyrobí pre teba.