# Výsledky hlasování o nejlepší kávu

Po měsíci hlasování se k nám konečně dostaly výsledky ankety. Celkem nám přišlo 9 hlasů, ze kterých vyplívá, že nejoblíbenějším horkým nápojem z automatu je **CAFÉ+CO** spolu s Latte Macchiato a Cappucinem na druhém a třetím místě. Děkujeme všem, co hlasovali, a pokud se do topky nedostal váš oblíbený nápoj, nezoufejte – tohle není naposled, co jsme tuhle anketu dělali.

//img1 (nechat i v pdf), udělat nějaký screen tohodle článku a vytisknout ho samostatně → dát na nástěnku u kanceláře

IT doupě

# Jak na… Databázi v Minecraft pluginu

V minulém čísle se nám do pluginu podařilo dostat příkazy, které spolu s eventy vytvořily funkční základ celého projektu. Dnes se posuneme trochu mimo Bukkit API a podíváme se na databáze, do kterých budeme ukládat hráčská data.

Prvním krokem je výběr vhodné databáze. Ve většině projektech používám MongoDB, která je objektová a běží v cloudu. Problém ale nastává, když si chce někdo nezkušený v počítačích váš plugin stáhnout. Jsou k němu totiž potřeba speciální ovladače a často i registrace do samotného Monga. Java má naštěstí v základu SQLite, databázi, která se dá uložit přímo v aplikaci a používat bez dodatečných ovladačů nebo registračních údajů.

Vytvoříme si jednoduchou funkci, která bude počítat, kolikrát se hráč připojil. Když se připojí, plugin zkontroluje, jestli už není zapsaný v databázi. Pokud ano, přidá mu do databáze +1, v opačném případě pro něj vytvoří nový řádek s daty.

Nejdříve musíme vytvořit databázi. Vytvořte .db soubor, a přes jakýkoliv SQL editor do něj vložte novou tabulku. Ta naše bude obsahovat jméno hráče (*name*) a číslo reprezentující kolikrát se připojil (*joinCount*). V javě na soubor odkážete přes tzv URI, které reprezentuje cestu a zároveň způsob otevření. Bude vypadat asi nějak takto: *jdbc:sqlite:[cesta k .db].*

Teď můžeme začít přidávat logiku. Aby jsme se k databázi vůbec připojili, vytvoříme funkci *Connect()*, která vrátí Connection a odpojí stará připojení. To uložíme do globální proměnné *connection*, se kterou budeme do databáze přistupovat. Pro poslání příkazu nám teď stačí jen zavolat *connection.createStatement().executeQuery().* Takže pokud chceme zjistit, jestli už má hráč v databázi záznam, stačí nám do tohoto příkazu vložit **SELECT COUNT(\*) FROM [jméno tabulky] WHERE name=event.getPlayer().getName()**. Pokud neholdujete SQL jazyku, ve zkratce to znamená „Vyber počet řádků v tabulce [jméno tabulky], která obsahují *name* [jméno hráče]“. Funkce *getInt(1)*, kterou zapíšeme za *executeQuery(),* nám zajistí, aby databáze vrátila číslo. Pokud je rovno nule, pak hráč v databázi žádná data nemá. V tom případě mu pomocí příkazu INSERT nějaká vytvoříme. Posledním krokem je přičíst číslo k joinCount, o což se postará příkaz UPDATE. Vezme hodnotu z databáze, přepíše ji a vrátí ji zpět.

Tím bychom měli databázi hotovou. Pokud k ní chcete přidat něco navíc, rozhodně doporučuji naučit se SQL. Je to jazyk, který by měl každý programátor alespoň trochu ovládat a na rozdíl od NonSQL databází má všude stejnou syntax. V příštím díle se podíváme na vlastní konfigurační soubory a jak s nimi pracovat.

//img2