

# Síťová Aplikace

Matěj Halmich

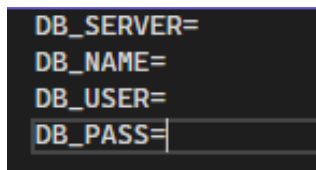
[Matejhalmich06@gmail.com](mailto:Matejhalmich06@gmail.com)

14.01.2025

Střední průmyslová škola elektrotechnická, Praha 2, Ječná 30

Tato dokumentace je součástí školního projektu

# Konfigurační soubor .env



```
DB_SERVER=  
DB_NAME=  
DB_USER=  
DB_PASS=
```

Aplikace využívá konfigurační soubor .env do kterého uživatel vyplní údaje do svého Microsoft SQL Serveru. (Správné vyplnění je nezbytné pro správné fungování aplikace.)

**DB\_SERVER=** ip adresa uživatelova serveru

**DB\_NAME=** název databáze do které uživatel nainportoval databázi

**DB\_USER=** název uživatele který se k databázi připojuje

**DB\_PASS=** heslo které uživatel používá na svém Microsoft SQL Serveru

## Upozornění:

Pokud zadáte nesprávné údaje aplikace se sice spustí, ale nebude možné pracovat s databází.

Pro účel školního projektu jsou údaje již vyplněné.

# Funkce aplikace + ovládání

Aplikace představuje decentralizovaný bankovní uzel, který komunikuje přes TCP/IP a funguje jako peer-to-peer systém. Každý uzel spravuje bankovní účty a umožňuje provádět různé operace, jako je:

- Vrácení kódu banky (vrací se moje aktuální ip adresa) (BC)
  - BC
  - BC 10.1.2.3
- Vytvoření účtu (vygeneruje se náhodný účet)(AC)
  - AC
  - 10001/10.1.2.3
- Vklad peněz na účet (AD)
  - AD 10001/10.1.2.3 3000
  - AD
  - Pokud uživatel zadá špatný formát zprávy např. Číslo vypíše se také error: ER Špatný formát čísla částky (musí být nezáporné celé číslo).
- Výběr peněz z účtu (AW)
  - AW 10001/10.1.2.3 2000
  - AW
  - Pokud chci vybrat více než mám a účtě vypíše se: "ER Není dostatek finančních prostředků."
- Vrácení zůstatku na účtu (AB)
  - AB 10001/10.1.2.3
  - AB 1000
- Smazání účtu (AR)
  - AR 10001/10.1.2.3
  - AR
  - Pokud chci smazat účet na které jsou finance vypíše se: "ER Nelze smazat bankovní účet na kterém jsou finance."
- Zjištění celkového zůstatku banky (BA)
  - BA
  - BA 1000
- Zjištění počtu klientů (BN)
  - BN
  - 1

Pokud uživatel zadá jiný příkaz tak se vypíše “ER Neznámý příkaz”.

Aplikace má timeout pro odpovědi:

- Pokud server neodpoví do 5ti sekund vyhodnotí to program jako chybu.
- Pokud uživatel nezadá žádný příkaz do 60ti sekund také to program vyhodnotí jako chybu.

## Vysvětlivky k příkazům

- BC- Bank code
  - Příkaz vrací kód banky. Kód banky je aktuální ip adresa
- AC - Account create
  - Příkaz vytvoří v bance nový bankovní účet a vrátí zpět tomu, kdo ho spustil jako odpověď číslo tohoto účtu. Číslo účtu je v rozmezí 10000 až 99999
- AD - Account deposit
  - Příkaz umožňuje vložit na bankovní účet peníze. Syntaxe je složená z kódu příkazu, který následuje mezera, pak číslo účtu, lomítko a kód banky. Dále mezera a částka.
- AW - Account withdrawal
  - Příkaz AW umožňuje vybrat z bankovního účtu peníze. Syntaxe i chování je stejné jako v u příkazu AD.
- AB - Account balance
  - Vrací aktuální zůstatek na účtu.
- AR - Account remove
  - Smaže bankovní účet, ale pouze za předpokladu že je zůstatek 0.
- BA - Bank (total) amount
  - Příkaz zjistí celkový součet všech finančních prostředků ve bance (Součet všech zůstatků na účtech)
- BN - Bank number (of clients)
  - Příkaz vypíše počet klientů, kteří mají v bance aktuálně bankovní účet.

# Spuštění Aplikace

**Důležité upozornění:** Ke spuštění aplikace je nutné mít nainstalovaný .net

<https://dotnet.microsoft.com/en-us/>

## Spouštění ve Visual studio:

- Zapněte svůj Ms SQL Server a vytvořte v něm databázi
- Do databáze nakopírujte tabulku BankAccountDB.sql ze složky Database\_import
- Celý skript executněte
- Do .env konfiguračního souboru vyplňte údaje k vašemu Ms SQL Serveru

```
DB_SERVER=  
DB_NAME=  
DB_USER=  
DB_PASS=
```

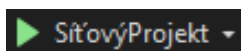
**DB\_SERVER=** ip adresa uživatelského serveru

**DB\_NAME=** název databáze do které uživatel naimportoval databázi

**DB\_USER=** název uživatele který se k databázi připojuje

**DB\_PASS=** heslo které uživatel používá na svém Microsoft SQL Serveru

- Klikněte ve Visual studio na spuštění aplikace



## Uživatelské spuštění exe souboru:

- Otevřete složku SíťovýProjekt
- Přejděte do složky bin\Debug\net8.0-windows
- Spusťte program SíťovýProjekt.exe

SíťovýProjekt\SíťovýProjekt\SíťovýProjekt\bin\Debug\net8.0

**Důležité upozornění:** Pro odevzdání jako školní projekt jsou data vyplněná (V release verzi kterou bych chtěl veřejně někam nahrát údaje vymažu a nechám uživatele aby používal vlastní db).

## Zdroje

<https://dev.to/luca1iu/using-the-logger-class-in-python-for-effective-logging-4ghc>

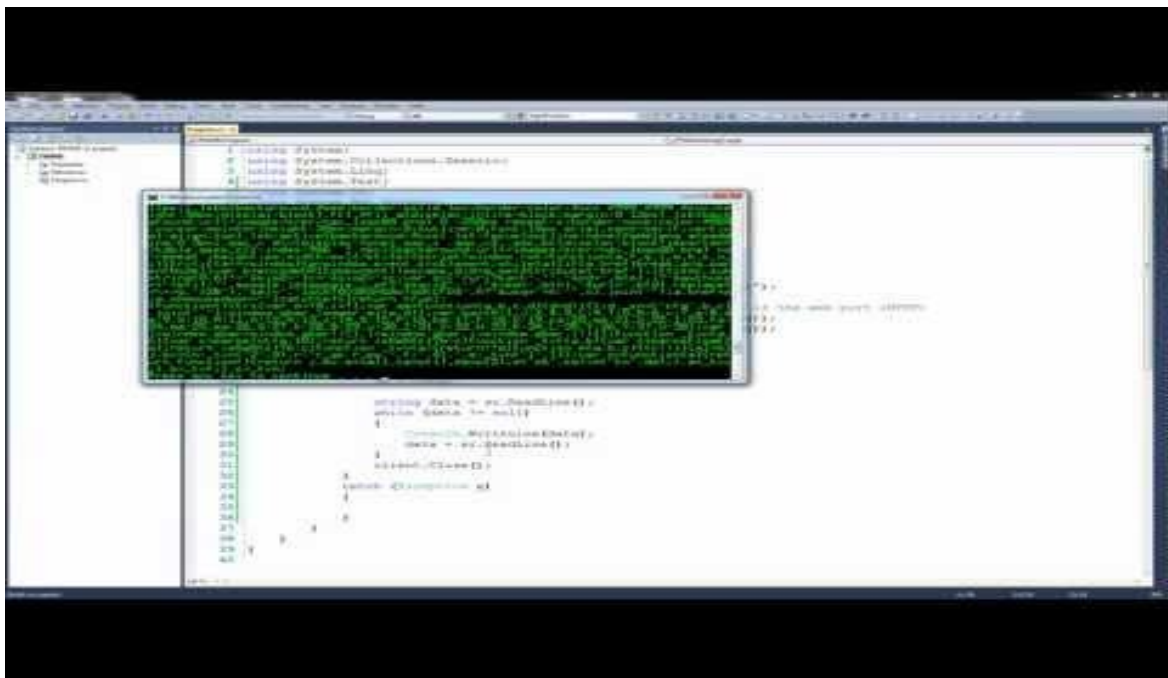
<https://learn.microsoft.com/en-us/dotnet/fundamentals/networking/sockets/tcp-classes>

<https://learn.microsoft.com/en-us/archive/msdn-magazine/2014/march/async-programming-asynchronous-tcp-sockets-as-an-alternative-to-wcf>

<https://learn.microsoft.com/en-us/archive/msdn-magazine/2001/february/net-p2p-writing-peer-to-peer-networked-apps-with-the-microsoft-net-framework>

<https://www.geeksforgeeks.org/what-is-p2p-peer-to-peer-process/>

<https://www.youtube.com/watch?v=K9-cuTmIK1o&t=246s>



<https://chatgpt.com/share/67a32669-782c-8005-af71-937ca48ae05b>

# ZnovuPoužitý kód

- **Třída Logger**
  - Logger jsem převzal ze svého síťového projektu který jsem dělal v pythonu akorát je předělaný do jazyka C#.
- **.env konfigurační soubor a knihovnu: DotNetEnv**
  - Pro připojení do databáze používám .env soubor. Ten se od minulého projektu nijak nezměnil.
- **Použití Knihoven pro práci s databází: EntityFrameworkCore, Microsoft.EntityFrameworkCore.SqlServer, Microsoft.Data.SqlClient**
  - V databázovém projektu jsem použil tyto knihovny pro práci s databází a pro mapování objektů na databázové tabulky. Využití zde zůstalo také stejné ale program pracuje s jinými tabulkami.
- **Třída BankContext**
  - Velmi podobná třídě CarRentalContext z databázového projektu akorát pracuje s jinými tabulkami.
- **Třída BankAccount**
  - Velmi podobná třídám které reprezentují tabulky v databázovém projektu např. Třída Customer. Akorát reprezentuje jinou tabulku.

## Použité knihovny třetích stran

V projektu byly využity následující NuGet balíčky pro zajištění funkcionality:

1. **DotNetEnv** (verze 3.1.1)
  - Knihovna pro načítání a správu konfiguračních hodnot ze souboru .env.
  - Používá se pro správu citlivých údajů, jako jsou přihlašovací údaje k databázi (např. server, uživatelské jméno, heslo).
  - <https://github.com/tonerdo/dotnet-env>
2. **Microsoft.Data.SqlClient** (verze 5.2.2)
  - Oficiální klientská knihovna pro komunikaci s Microsoft SQL Serverem.
  - Používá se pro připojení aplikace k databázi a spouštění dotazů na SQL Server.
  - <https://github.com/dotnet/SqlClient>
3. **Microsoft.EntityFrameworkCore** (verze 9.0.0)
  - Hlavní balíček pro Entity Framework Core, který zajišťuje mapování objektů na databázové tabulky (ORM).
  - Umožňuje práci s databází v objektově orientovaném stylu, bez nutnosti psát přímo SQL dotazy.
  - <https://learn.microsoft.com/en-us/ef/core/>
4. **Microsoft.EntityFrameworkCore.SqlServer** (verze 9.0.0)
  - Rozšíření Entity Framework Core, které přidává podporu pro Microsoft SQL Server.
  - Obsahuje specifické funkce pro SQL Server, jako je práce s datovými typy a optimalizace dotazů.
  - <https://learn.microsoft.com/en-us/ef/core/>