

# Používateľská príručka na inštaláciu a spustenie aplikácie

Táto príručka popisuje postup, ako si stiahnuť, nastaviť a spustiť aplikáciu Echoo z GitHub repozitára.

## 1. Predpoklady

Pred samotným spustením aplikácie musíte mať nainštalované:

- Node.js
- npm
- PostgreSQL

## 2. Stiahnutie projektu

Projekt si stiahnite z GitHubu: <https://github.com/Zsofkaaaa/VPWA>  
Následne sa presuňte do koreňového priečinka projektu: Echoo

## 3. Inštalácia závislostí

Aplikácia je rozdelená na frontend a backend časť. Závislosti je potrebné nainštalovať osobitne.

### 3.1 Koreňový priečinok (Echoo)

V koreňovom priečinku projektu nainštalujte balík pre súčasné spúšťanie aplikácií:  
*npm install --save-dev concurrently*

### 3.2 Frontend

Prejdite do priečinka frontend a nainštalujte potrebné balíky:

*npm install -D @types/node* (TypeScript typové definície pre Node.js)  
*npm install socket.io-client* (klientská knižnica pre real-time komunikáciu pomocou WebSocketov)

### 3.3 Backend

Prejdite do priečinka backend a postupne nainštalujte nasledovné balíky:

*npm install node-cron* (na plánovanie a vykonávanie automatizovaných úloh)  
*npm install axios* (na vykonávanie HTTP požiadaviek)  
*npm install -D @types/axios* (typové definície pre knižnicu Axios)

```
npm install @adonisjs/lucid (ORM nástroj frameworku AdonisJS)  
node ace configure @adonisjs/lucid (Konfiguračný príkaz pre ORM Lucid. Pri konfigurácii je potrebné zvoliť databázu PostgreSQL a potvrdiť inštaláciu doplnkových balíkov.)  
npm install --save-dev @types/pg (TypeScript typové definície pre PostgreSQL klienta)  
npm install argon2 (hashovací algoritmus používaný na ochranu používateľských hesiel)  
npm install @adonisjs/hash argon2 (hashovacia knižnica frameworku AdonisJS)  
npm install socket.io (serverová knižnica pre real-time komunikáciu pomocou websocketov)  
npm install --save-dev @types/socket.io (typeScript typové definície pre Socket.IO)  
npm install --save-dev ts-node typescript @types/node (typové definície pre Node.js)
```

## 4. Konfigurácia prostredia

V priečinku backend vytvorte súbor .env a nastavte ho podľa vzoru .env.example.

API kľúč viete vygenerovať v backende pomocou príkazu:

```
node ace generate:key
```

Tento kľúč vložte do súboru .env.

## 5. Práca s databázou

### 5.1 Vytvorenie databázy

Databázu môžete vytvoriť pomocou vlastného skriptu v backende:

```
npm run create:db
```

### 5.2 Vytvorenie tabuľiek a naplnenie dátami

Na vytvorenie databázových tabuľiek a spustenie seedov použíte príkazy:

```
node ace migration:run
```

```
node ace db:seed
```

### 5.3 Ďalšie databázové príkazy (voliteľné)

Tieto príkazy nie sú potrebné pre bežné používanie aplikácie, boli však využívané počas vývoja a testovania:

```
node ace migration:fresh (vymaže všetky existujúce tabuľky v databáze a znova ich vytvorí na základe migrácií)
```

```
node ace migration:reset (vráti databázu do pôvodného stavu zrušením všetkých spustených migrácií)
```

```
node ace migration:fresh --seed (vymaže všetky tabuľky, znova ich vytvorí a následne naplní databázu testovacími dátami)
```

## 6. Spustenie aplikácie

Po úspešnej inštalácii a konfigurácii môžete aplikáciu spustiť z koreňového priečinka projektu (priečinok Echoo) pomocou príkazu:

*npm run dev*

Tento príkaz spustí frontend aj backend súčasne. Počkajte, kým frontend [1] vypíše dostupné IP adresy, a následne kliknite na localhost, aby sa aplikácia zobrazila.