

# POLchatka - Instrukcja instalacji backendu PHP

## Wymagania

- **PHP 7.4+** (zalecane PHP 8.0+)
- **MySQL 5.7+** lub **MariaDB 10.2+**
- **Apache** lub **Nginx** z mod\_rewrite
- Rozszerzenia PHP: `pdo`, `pdo_mysql`, `mbstring`, `json`

## Instalacja

### 1. Klonowanie repozytorium

```
bash

git clone https://github.com/mikusz3/POLchatka.git
cd POLchatka
```

### 2. Konfiguracja bazy danych

#### A. Utwórz bazę danych

```
sql

CREATE DATABASE polchatka_db CHARACTER SET utf8mb4 COLLATE utf8mb4_polish_ci;
```

#### B. Zaimportuj strukturę

```
bash

mysql -u root -p polchatka_db < php/init.sql
```

Lub w phpMyAdmin zaimportuj plik `php/init.sql`

### 3. Konfiguracja PHP

#### A. Edytuj `php/config.php`

```
php

$host = 'localhost';      // Adres serwera MySQL
$dbname = 'polchatka_db'; // Nazwa bazy danych
$username = 'twoja_nazwa'; // Nazwa użytkownika MySQL
$password = 'twoje_haslo'; // Hasło MySQL
```

## B. Ustaw uprawnienia (Linux/Mac)

```
bash

chmod 755 php/
chmod 644 php/*.php
```

## 4. Konfiguracja serwera web

### Apache (.htaccess)

```
apache

RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^api/(.*)$ php/api.php?endpoint=$1 [QSA,L]

# Zabezpieczenia
<Files "config.php">
    Require all denied
</Files>

<Files "*.sql">
    Require all denied
</Files>
```

### Nginx

```
nginx

location /api/ {
    try_files $uri $uri/ /php/api.php?endpoint=$uri&$args;
}

location ~ \.php$ {
    fastcgi_pass unix:/var/run/php/php8.0-fpm.sock;
    fastcgi_index index.php;
    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}

# Zabezpieczenia
location ~ \.(sql|log)$ {
    deny all;
}
```

## Struktura plików PHP

```
php/
├── config.php      # Konfiguracja bazy danych i funkcje pomocnicze
├── init.sql        # Struktura bazy danych z przykładowymi danymi
├── register.php    # Rejestracja użytkowników
├── session_login.php # Logowanie użytkowników
├── logout.php      # Wylogowanie
├── profile.php     # Zarządzanie profilami
├── admin_panel.php # Panel administratora
└── api.php         # API do różnych funkcji
```

## Endpointy API

### Publiczne

- `GET /api/check_username?username=test` - Sprawdź dostępność nazwy użytkownika
- `GET /api/check_email?email=test@example.com` - Sprawdź dostępność emaila
- `GET /api/online_users` - Lista użytkowników online
- `GET /api/site_stats` - Statystyki strony
- `GET /api/csrf_token` - Pobranie tokenu CSRF

### Wymagające logowania

- `POST /api/heartbeat` - Aktualizacja aktywności użytkownika
- `POST /api/report_content` - Zgłaszanie treści

### Bezpośrednie pliki PHP

- `POST /php/register.php` - Rejestracja
- `POST /php/session_login.php` - Logowanie
- `POST /php/logout.php` - Wylogowanie
- `GET|POST /php/profile.php` - Zarządzanie profilem
- `GET|POST /php/admin_panel.php` - Panel administracyjny

## Konta testowe

Po zaimportowaniu `init.sql` dostępne są konta:

Login	Hasło	Typ
admin	admin123	Administrator
user	user123	Użytkownik
testuser	user123	Użytkownik

## Bezpieczeństwo

### Implementowane zabezpieczenia:

- **Hashowanie haseł** - bcrypt z solą
- **Ochrona CSRF** - tokeny w formularzach
- **Prepared statements** - ochrona przed SQL injection
- **Sanityzacja danych** - oczyszczanie inputów
- **Sesje bezpieczne** - regeneracja ID, httpOnly cookies
- **Logowanie aktywności** - śledzenie działań użytkowników
- **Rate limiting** - opóźnienia przy niepoprawnych logowaniach

### Dodatkowe zabezpieczenia (produkcja):

php

// W config.php dla HTTPS:

```
ini_set('session.cookie_secure', 1);
```

// Dodatkowe nagłówki bezpieczeństwa:

```
header('X-Content-Type-Options: nosniff');
```

```
header('X-Frame-Options: DENY');
```

```
header('X-XSS-Protection: 1; mode=block');
```

## Tabele bazy danych

- **users** - Dane użytkowników
- **user\_sessions** - Sesje "zapamiętaj mnie"
- **activity\_logs** - Logi aktywności
- **groups** - Grupy użytkowników
- **wall\_posts** - Posty na ścianie

## Debugowanie

### Włączenie logów błędów:

php

```
// Na początku config.php (tylko development!)
error_reporting(E_ALL);
ini_set('display_errors', 1);
ini_set('log_errors', 1);
ini_set('error_log', 'php_errors.log');
```

## Sprawdzenie połączenia z bazą:

bash

```
php -r "require 'php/config.php'; echo 'Połączenie OK!';"
```

## Logi aktywności:

Wszystkie ważne działania są logowane w tabeli `activity_logs`



## Uruchomienie

### XAMPP/WAMP

1. Skopiuj pliki do `htdocs/polchatka/`
2. Uruchom Apache i MySQL
3. Zaimportuj bazę danych
4. Skonfiguruj `config.php`
5. Otwórz `http://localhost/polchatka/content/index.html`

### Laragon

1. Skopiuj do `www/polchatka/`
2. Konfiguracja automatyczna
3. Dostęp: `http://polchatka.test`

### Serwer produkcyjny

1. Upload plików przez FTP/SFTP
2. Utwórz bazę danych w panelu hosta
3. Skonfiguruj `config.php` z danymi hosta
4. Ustaw uprawnienia katalogów (755) i plików (644)



## Rozszerzenia

Backend jest przygotowany na przyszłe funkcje:

- Upload plików (awatary, zdjęcia)

- System wiadomości prywatnych
- Rozbudowane grupy z członkostwem
- System ocen i komentarzy
- Integracja z social media
- API mobilne

## Troubleshooting

### Błąd połączenia z bazą:

- Sprawdź dane w `config.php`
- Upewnij się że MySQL działa
- Sprawdź uprawnienia użytkownika

### Błędy sesji:

- Sprawdź uprawnienia do katalogu `/tmp`
- Wyczyść cookies przeglądarki

### Błędy 500:

- Sprawdź logi błędów serwera
- Upewnij się że wszystkie pliki PHP są poprawne
- Sprawdź konfigurację Apache/Nginx

Gotowy backend obsługuje pełny cykl życia użytkownika od rejestracji po zaawansowane funkcje administracyjne! 🎉