

Opis aplikacji do powiadomień

Dokumentacja skrócona

20 listopada 2025

1 Przeznaczenie

Aplikacja **Insert Powiadomienia** to wewnętrzny kanał komunikacji do dystrybucji ogłoszeń na urządzenia mobilne. Łączy backend FastAPI (MySQL) z frontendem Flutter i zapewnia jednolite doświadczenie na Androidzie oraz iOS. Powiadomienia trafiają przez Firebase Cloud Messaging (FCM) na topic `all`, więc każdy zalogowany klient dostaje je równocześnie, niezależnie od urządzenia. Dostęp kontroluje logowanie Google oparte na liście dozwolonych e-maili w bazie, co ułatwia zarządzanie odbiorcami bez dodatkowych kont aplikacyjnych. Po zalogowaniu użytkownik może szybko przeszukiwać i rozwijać komunikaty, brać udział w ankietach i dodawać komentarze; w foregroundzie pojawiają się lokalne banery dla nowych pushy, a w tle powiadomienia systemowe.

2 Architektura

- **Backend (backend/)**: FastAPI z APSchedulerem i MySQL. Wystawia REST dla powiadomień, logowanie Google oraz moduł ankiet i komentarzy. Harmonogram działa w tle, wysyłając FCM z konta serwisowego Firebase.
- **Frontend (frontend/)**: Flutter z Firebase Messaging, lokalnymi powiadomieniami, logowaniem Google (`google_sign_in`), zarządzaniem stanem (`provider`) i zapisem preferencji w `SharedPreferences`. Interfejs utrzymuje kolorystykę InsERT i obsługuje tryb jasny/ciemny.
- **Integracje**: FCM HTTP v1 (klucz serwisowy + `PROJECT_ID`) oraz OAuth Google (WEB CLIENT ID w zmiennej `GOOGLE_CLIENT_ID`). Klient subskrybuje topic `all`, a backend weryfikuje *audience* tokena przed wpuszczeniem użytkownika.

3 Backend – działanie

3.1 Notyfikacje

Końcówka `/notifications` zwraca wysłane wpisy zaplanowane na przeszłość, sortowane od najnowszych, z limitem dla płynności i stałą strukturą pól (ID, tytuł, treść, skrót, miniatura, czas zaplanowania, status wysyłki). Dodawanie przyjmuje tytuł, treść, opcjonalny obraz oraz datę publikacji; automatycznie tworzy skrót (pierwsze zdanie) i zapisuje rekord ze statusem `sent=0`, który czeka na wysyłkę. Dzięki temu operator może wprowadzać ogłoszenia z wyprzedzeniem, a harmonogram zajmie się ich dystrybucją.

3.2 Harmonogram wysyłki

Scheduler co minutę szuka rekordów `sent=0` z przeszłą datą. Dla każdego pobiera token dostępu z konta serwisowego Firebase, buduje komunikat (tytuł, treść, opcjonalny obraz) i wysyła go na topic `all`. Przy udanym statusie HTTP oznacza wpis jako wysłany, by uniknąć duplikatów, a w przypadku błędu zostawia status `sent=0`, co pozwala na ponowną próbę w kolejnej iteracji. Logi zawierają statusy i ewentualne błędy, dzięki czemu widać opóźnienia, niedostępność FCM lub problemy z kluczem serwisowym.

3.3 Autoryzacja Google

Endpoint `/auth/google` przyjmuje idToken, weryfikuje podpis u Google, sprawdza *audience* względem `GOOGLE_CLIENT_ID`, a następnie potwierdza, że e-mail jest zweryfikowany i znajduje się w tabeli `users`. Sukces zwracany jest tylko wtedy; w innym przypadku pojawia się 401/403. Mechanizm zapewnia prostą kontrolę dostępu opartą na whiteliście i eliminuje konieczność przechowywania haseł, a jednocześnie uniemożliwia dostęp z tokenami wystawionymi dla innych aplikacji.

3.4 Ankiety i komentarze

Moduł `/notifications/...` pozwala tworzyć ankiety przy powiadomieniach, oddawać i nadpisywać głosy (po jednym na adres) oraz pobierać wyniki z procentami i sumą głosów. Dodatkowo umożliwia dodawanie komentarzy, wyświetlanie ich z licznikiem plusów/minusów i głosowanie w obu kierunkach. Każdy wpis zawiera autora (z e-maila lub podanej nazwy) i czas dodania, co pozwala śledzić aktywność użytkowników i wyłapywać najbardziej wartościowe odpowiedzi.

4 Frontend – działanie

4.1 Start aplikacji

Po uruchomieniu wczytywany jest plik `.env`, inicjalizowane jest Firebase, rejestrowany jest handler wiadomości w tle, a na Androidzie tworzony jest kanał lokalnych powiadomień o wysokim priorytecie. Aplikacja prosi o zgodę na powiadomienia (Android 13+/iOS), subskrybuje topic `all`, ustawia alerty w foregroundzie i wypisuje token FCM do logów diagnostycznych. Dzięki temu pierwsze uruchomienie przygotowuje aplikację do odbioru pushy bez dodatkowych ekranów konfiguracji.

4.2 Logowanie

Ekran startowy oferuje „Zaloguj się przez Google”. Na podstawie platformy wybierany jest odpowiedni identyfikator klienta. Po wybraniu konta aplikacja pobiera `idToken`, wywołuje `/auth/google` i: przy statusie 200 przechodzi do ekranu głównego z e-mailem użytkownika, w przeciwnym razie informuje o braku dostępu i czyści stan logowania. Poprzedni stan Google Sign-In jest przed próbą resetowany, a cichy sign-in przyspiesza ponowne wejście.

4.3 Ekran główny

Lista powiadomień pochodzi z `/notifications`. Użytkownik może wyszukiwać po tytule i skrócie, odświeżać listę gestem pull-to-refresh, przełączać tryb jasny/ciemny (zapamiętywany lokalnie) i wylogować się. Każda karta zawiera miniaturę, tytuł, skrót, względny czas oraz rozwijaną treść z obsługą klikalnych linków; długie przytrzymanie otwiera modal z pełnym widokiem i powiększonym obrazem. W foregroundzie nowe push'e są dodatkowo prezentowane banerem lokalnym, a kliknięcie powiadomienia z tła przenosi do aplikacji, co zapewnia spójny przepływ od powiadomienia do szczegółu.

4.4 Ankiety i komentarze

Gdy powiadomienie ma aktywną ankietę, sekcja pokazuje pytanie, opcje, paski procentowe i przycisk „Zagłosuj”; po głosie dane są odświeżane i pojawia się komunikat `SnackBar`. Sekcja komentarzy pozwala dodać wpis podpisany nazwą z e-maila lub podaną nazwą, przeglądać wszystkie komentarze z liczbą głosów i datą oraz głosować w góre/dół; przy każdej akcji lista jest aktualizowana.