FelElek fejlesztői dokumentáció

# Általános információk

A FelElek nevű applikáció a [Dart](https://dart.dev/) programozási nyelvben készült, és a [Flutter](https://flutter.dev/) framework-öt használja a crossplatformos applikáció fejlesztésre. A program leépítéséhez android operációs rendszerre elég a Flutter fejlesztői eszközök telepítése (<https://flutter.dev/docs/get-started/install>), viszont IOS mobil operációs rendszerre fordításhoz szükséges MacOS használata.

# Fordítási útmutató

A program lefordítható integrált fejlesztő környezet segítségével (ajánlott), vagy használható a Flutter parancssoros eszköze a következő módon:

flutter pub get

flutter build apk

Az első parancs letölti, és lefrissíti a pubspec.yml-ben megadott könyvtárakat, és a második parancs létrehoz számunkra egy telepíthető .APK fájlt.

# Kód formázási elvek

A kód univerzális szerkeszthetősége végett a ’Clean code’ elvein túl, a dartfmt programot használjuk a nyelv saját formázási szabályainak betartására.

<https://dart.dev/guides/language/effective-dart/style#formatting>

Fontos egyediség a Dart programozási nyelvben: A paraméterek után, úgy nevezett lengő vesszők használata, hogy a komponens alapú GUI építés ne menjen túlságosan a jobb oldalra.

# Fájl elhelyezkedések

A fájlok elhelyezkedése az alapértelmezett Flutter projekt szabványait követik.

## Beállítások

A felhasznált könyvtárak a pubspec.yml fájlban találhatóak.

Az android applikáció fordításához a beállítások a /android mappában találhatóak.

Az IOS mobil applikáció fordításához a beállítások a /ios mappában találhatóak

## Kód

A projektben a kódok, és az őket tartalmazó fájlok, a mapparendszerben szerepük szerint vannak elhelyezve a /lib mappában.

## Kellékek

Minden a kódon túl a /assets mappában találhatóak meg (Pl: nyelvi fájlok).

# Flutter

## Alkalmazás állapot kezelés

A komponenseken megjelenő adatokat – amikor nem eldönthető az adat tulajdonos – az ún. BLoC mintában tároljuk, amire a flutter-bloc könyvtár ad segítő komponenseket a könnyű elérésért. Ez lehetővé teszi, hogy majdnem az összes komponens ún. Stateless widget legyen, ami nagyban segít a teljesítményen, és csökkenti a program komplexitását.

## Megjelenítés kezelés

A jelenleg megjelenő oldal kezeléséhez, és a vissza gomb kontextusához egy Navigátort használunk, ami URL-szerűség alapján dönti el, hogy éppen melyik komponens fa legyen a memóriában és megjelenítve.

Ezek az elérési utak megtalálhatóak a /lib/navigation/route\_generator.dart fájlban.

## Adattárolás

A minimális kulcs-érték tárolásra a Flutter által kínált SharedPreferencies-t használjuk, a nagyobb adatok tárolásához a Dart beépített fájlkezelőjét az applikációnak kijelölt mappában.