

Codex tesztelési irányelvek – Alapdokumentáció precíz fejlesztéshez (Flutter + Firebase környezet)

Ez a dokumentum rögzíti, hogy a Codex AI-agent milyen teszteléseket végezzen a fejlesztési ciklus során, milyen formában, milyen lefedettséggel, milyen környezetben – különösen a TippmixApp projekt struktúráját követve.

Tesztelési cél

- Minden fejlesztés során biztosítani kell, hogy a kód helyesen működik, nem tör el meglévő funkciókat, megfelel az architekturális és stílusbeli szabályoknak.
- A tesztelés célja:
 - regressziók elkerülése
 - stabil működés biztosítása
 - folyamatos minőségellenőrzés automatizált módon

Codex által futtatható teszt típusok

Teszt típus	Célja	Codex támogatja?
Unit test	Business logic ellenőrzése	✓ teljes mértékben
Widget test	UI viselkedés validálása (mock háttérrel)	✓ mockolt környezetben
Golden test	Vizuális regresszió megelőzése	⚠ csak headless módban

Futási környezet és technikai elvárások

1. Futási parancsok

```
flutter pub get
flutter analyze --fatal-infos --fatal-warnings
flutter test --coverage
```

2. Tesztkörnyezet

- VM-alapú, headless környezet (nincs mobil emulátor)
- Internetelérés engedélyezve
- Firebase-hívások mockolva (`firebase_coremocks`, `firebase_authmocks`, stb.)
- Szolgáltatásfüggőségek `locator`-ral vagy `FakeService`-ekkel kezelve

3. Elvárt struktúra

- `test/` → unit és widget tesztek

Lefedettségi elvárások

- Minden `flutter test --coverage` futtatás után:
- min. 90% coverage legyen
- a coverage-t `.lcov` vagy HTML formátumban generálja (pl. CI-hez)

Lefedés:

- Minden service class → legalább 1 unit test
 - Minden képernyő → legalább 1 widget test
 - Kritikus UI → opcionálisan golden test
-

Kódminőség-ellenőrzések a tesztek előtt

- `flutter analyze` ne adjon hibát vagy warningot
 - Ne legyen `print`, `TODO`, vagy kommentben maradt hibás kódrészlet
 - Minden fájl legyen `null-safe`
 - `dart format .` módosítás nélküli legyen (idempotens)
-

CI integrációra előkészítés

Codex által biztosított lépések: - Teszt parancsokat előkészít (parancssor) - `coverage/` vagy `lcov.info` generálás bekapcsolva - Tesztelési log exportálása (stdout, fájl) - Git diff vagy patch készítése a tesztelés során javított fájlokról

Kimeneti elvárások Codex futtatás után

- Részletes `flutter test` output (PASS/FAIL jelzéssel)
 - Lefedettségi arány százalékban
 - Hibák vagy tesztelési hiányosságok listázása
 - Git patch a módosított fájlokról
-

Összefoglalás

Codex minden fejlesztés után az alábbiakat hajtsa végre: 1. Formázás: `dart format .` 2. Statisztikai elemzés: `flutter analyze` 3. Tesztelés: - `flutter test --coverage` 4. Kimenet: - teszteredmények - lefedettségi százalék - patch/javítások

Minden platform channel hívást mockolni kell. A Codex kizárólag olyan teszteket futtasson, amelyek VM-safe, mockolt, és stabil módon CI-ben is újrafuttathatók.