

Javaslati lista (bónusz)

1. Swagger

- A requestId mező jelenleg numerikus értéként érkezik a backendtől, miközben az OpenAPI definícióban stringként van megadva. Ez típuseltéréshez vezethet, ami zavaró lehet az automatikus validáció vagy generált kliensek szempontjából.
 - Az API osztályok generálásakor minden végpont DefaultApi név alá kerül, mivel nincs használva tags alapú csoportosítás. Az info szekción kívül elhelyezett tags objektum lehetőséget ad részletes leírásokra és különálló osztályok generálására (pl. HighwayVignetteApi).
-

2. UI optimalizálás

- Az AppButton komponens jelenleg fix stílust használ. A ThemeExtension használata lehetővé teszi egy globális téma definiálását, amely bármely widget tree-ben lokálisan felülírható. Ez egységes megjelenést biztosít, miközben rugalmasan alakítható marad.
 - A szövegek jelenleg közvetlenül a widget tree-ben szerepelnek (pl. Text('Hiba történt:')), ami megnehezíti a karbantartást és módosítást. Egy lokalizációs szolgáltatás – még egynyelvű esetben is – átláthatóbbá tenné a szövegkezelést, és megkönnyítené a későbbi bővítést vagy egységesítést.
-

3. Backend hívások

- A /v1/highway/info végpont több, egymástól független adatot (járműkategóriák, matricák, vármegyék) ad vissza egyetlen válaszban. Azok számára, akik nem igényelnek vármegyei matricát, a counties adat felesleges lekérésnek számít.
 - A highwayVignettes lista országos és vármegyei matricákat egyaránt tartalmaz, amely minden esetben visszatérésre kerül. Az országos típusok (DAY, MONTH, YEAR) fixen ismertek, míg a vármegyeiek (YEAR_XX) bővíthetnek, így különválasztásuk logikusabb struktúrát eredményezhet.
 - A vignetteType jelenlegi tömbös felépítése összetettebbé teszi az egyértelmű azonosítást, különösen vármegyei esetekben. Egyedi vignetteId használatával a counties struktúra könnyebben kapcsolható lenne a megfelelő díjtételekhez.
 - A /v1/highway/order végpont category mezője név szerint egyezik a vehicleCategories.category mezővel, azonban a jelenlévő vignetteCategory (pl. D1) megtévesztő lehet. A Swagger dokumentáció hiánya esetén könnyen félreérthető, hogy melyik értéket kell használni, miközben a backend 200-as státuszkóddal válaszol akkor is, ha nem megfelelő az input.
-

4. API kezelés / hálózati réteg

- A hibakezelés minden egyes hívásnál lokálisan try-catch blokkban történik, ami ismétlődő kódhoz és fragmentált hibakezelési logikához vezet. Egy egységes interceptor képes lehet központilag kezelni a hálózati és backend hibákat, miközben naplózás vagy lokalizált hibaüzenetek is társíthatók.
 - Az Interactor jelenleg többféle adatforrást kezel közvetlenül, ami a felelősségi körök elmosódásához vezethet. A Repository réteg bevezetésével az adatforrások kezelése elválasztható a konkrét üzleti logikától, ami letisztultabb és könnyebben tesztelhető architektúrát eredményez.
-

5. UX

- A vármegyei kiválasztó nézeten csak visszanyíl található a fejlécben, míg a vásárlás megerősítő nézet már „Mégsem” gombot is kínál. Ez a különbség eltérő elvárásokat kelthet a felhasználóban, különösen, ha a visszanyíl hatása nem egyértelmű (pl. a teljes folyamat megszakad-e, vagy csak egy lépéssel lépünk vissza).
 - Az SVG alapú vármegyei térkép jelenleg statikus, de interaktív funkciókkal – mint például kattintható megyék, vizuális kijelölés, visszajelzések – tovább fokozható lenne a felhasználói élmény.
-

6. Kód tisztaság

- Az oldalankénti BLoC-ok használata követhető struktúrát eredményez. Ahogy a projekt mérete nő, ezek csoportosítása egy dedikált logic/ vagy application/ könyvtárba segítheti a rendezettebb fájlszerkezet kialakítását, és gyorsabb tájékozódást biztosíthat új fejlesztők számára.