Fidbek SDK Rehberi

Fidbek mobil SDK ailesi icin teknik, urunsel ve operasyonel basvuru rehberi

Tarih: 10 Mart 2026

Bu dokuman, Fidbek mobil SDK ekosistemini teknik ve urunsel acidan tek yerde toplamak icin hazirlandi. Amac, hem ekip ici referans hem de yeni entegrasyon yapacak gelistiriciler icin ciddi bir basvuru kaynagi olusturmaktir.

1. Yonetici Ozeti

Fidbek, mobil uygulamalarda kullanicidan bug, iyilestirme talebi ve genel geribildirim toplamak icin gelistirilmis bir SDK ailesidir. Cekirdek deneyim kullanicinin uygulama icinden geri bildirim formu acmasi, aciklamasini yazmasi, gerekirse ekran goruntusu veya ek medya eklemesi ve bu raporun Fidbek backend'ine gonderilmesi uzerine kuruludur.

Bugun sistemin en guclu tarafi sunlardir:

- Native iOS ve Android cekirdek SDK'lar
- Flutter ve React Native wrapper'lari
- Kimlik esleme (`identify`) ve kalici `installation_id`
- Dashboard tarafinda report, audience ve installation takibi
- Slack, Webhook ve Jira entegrasyonlari
- Cok dilli yerlesik geri bildirim arayuzu

Bugun sistemin en onemli yapisal gercegi sunudur:

- Android native binary hattimiz `0.3.0` seviyesindedir.
- iOS native binary hattimiz `0.3.5` seviyesindedir.
- Wrapper'lar iOS tarafinda `0.3.5` binary'sini tasir.

Bu fark teknik olarak yonetilebilir durumdadir, ancak roadmap tarafinda Android native hattinin da ayni release standardina getirilmesi gerekir.

2. Rehberin Kapsami

Bu rehber su konulari kapsar:

- SDK ailesinin mimarisi
- Guncel surumler
- Public API yuzeyi
- Kimliklendirme ve installation tracking
- Report payload ve toplanan veriler
- Lokalizasyonlar
- Dashboard ile iliski
- Wrapper politikasi
- Privacy ve veri minimizasyonu
- Entegrasyon onerileri
- Troubleshooting
- Release sureci

Bu rehberin disinda kalan konular:

- Backend schema'nin tam satir satir aciklamasi
- Dashboard ekranlarinin pixel-perfect UI dokumantasyonu
- Fiyatlandirma ve ticari kosullar

3. Repo ve Bilesen Haritasi

Fidbek SDK sistemi tek bir repo degil, birden fazla kaynagin birlikte calistigi bir yapidir:

- `source-android`

Android native source-of-truth

- `source-ios`

iOS native source-of-truth

- `android/fidbek-android`

  Android binary / maven distribution repo

- `ios/fidbek-ios`

  iOS binary Swift Package distribution repo

- `fidbek-flutter`

  Flutter wrapper

- `fidbek-react-native`

  React Native wrapper

Pratik kural:

- Native davranis degisiyorsa kaynak repo `source-android` veya `source-ios` tarafindan baslar.
- Wrapper API veya paketleme degisiyorsa Flutter / React Native repo'lari ayrica guncellenir.
- Binary dagitim repo'lari wrapper'lara giden artefact'larin resmi dagitim noktalaridir.

4. Guncel Surum Matrisi

10 Mart 2026 itibariyla guncel durum:

Bilesen	Guncel Surum	Not
Android native source	0.3.0 hattinda	Wrapper'larda Android binary 0.3.0
iOS native source	0.3.5	Scriptli XCFramework pipeline aktif
iOS binary repo	0.3.5	Swift Package binary dagitimi
Flutter wrapper	0.3.5	iOS binary 0.3.5, Android binary 0.3.0
React Native wrapper	0.3.5	iOS binary 0.3.5, Android binary 0.3.0

Bu matristeki en kritik not:

- Wrapper versiyonu ile native Android binary versiyonu bire bir ayni olmak zorunda degildir.
- Bugun wrapper'larin ana patch release'leri son donemde iOS binary ve iOS packaging duzeltmeleri uzerinden ilerlemistir.

5. Urun Davranisi: Fidbek SDK Tam Olarak Ne Yapar?

Son kullanici tarafinda temel ak■■ soyledir:

1. Uygulama Fidbek SDK'yi token ile configure eder.
2. Kullanici manual olarak veya shake ile Fidbek formunu acar.
3. Ilk dialogda kategori secilir.
4. Kullanici mesajini girer.
5. Bug ise occurrence frequency secilebilir.
6. Kullanici izin verirse iletisim e-postasi gonderir.
7. Screenshot veya ek medya secilebilir.
8. Rapor backend'e iletilir.
9. Dashboard, Slack, Webhook ve Jira taraflari bu rapor uzerinden calisir.

Dashboard tarafinda bu report bir entity olarak gorulur. Ayrica device ping akisi sayesinde installation / audience gorunumu de olusur.

6. Public API Yuzeyi

6.1 Native Android

Android ana entrypoint `Fidbek` objesidir. Tavsiye edilen cekirdek yuzey:

- `initialize(application, config)`
- `initialize(application, token, shakeToOpenEnabled)`
- `open()`
- `identify(userId?, name?, email?)`
- `clearIdentity()`
- `shutdown()`

Not:

- Programmatic attachment staging methodlari native source'ta backward compatibility icin tutulur.
- Ancak bu methodlar deprecated durumdadir.
- Wrapper'lara expose edilmez.

6.2 Native iOS

iOS ana entrypoint `Fidbek.shared` uzerindendir. Tavsiye edilen cekirdek yuzey:

- `configure(_ configuration)`
- `configure(token:shakeToOpenEnabled:)`
- `present()`
- `identify(userId:name:email:)`
- `clearIdentity()`
- `stop()`

Not:

- iOS tarafinda da attachment staging helper'lari vardir.
- Bunlar deprecated durumdadir.
- Tavsiye edilen entegrasyon yuzeyinin parcasi degildir.

6.3 Flutter Wrapper

Flutter wrapper yuzeyi bilincli olarak dar tutulur:

- `FidbekFlutter.configure`
- `FidbekFlutter.open`
- `FidbekFlutter.identify`
- `FidbekFlutter.clearIdentity`
- `FidbekFlutter.shutdown`

Wrapper politikasi:

- Son kullaniciya acik experience, SDK icindeki yerlesik akistir.
- Native attachment staging methodlari Flutter tarafina expose edilmez.

6.4 React Native Wrapper

React Native wrapper yuzeyi de ayni sekilde cekirdek 5 method ile sinirlidir:

- `configure`
- `open`
- `identify`
- `clearIdentity`
- `shutdown`

React Native wrapper New Architecture odaklidir. Dokumantasyonda eski bridge fallback tavsiye edilmez.

7. Konfigurasyon Modeli

Bugunku konfigurasyon modeli sade tutulmustur:

- `token`
- `shakeToOpenEnabled`

Bu sadelik bilincli bir tercihtir. Ancak gelecekte asagidaki alanlarin eklenmesi muhtemeldir:

- `baseUrl`
- `enabled`
- `theme / branding`
- `category policy`
- `autoScreenshot`
- `logLevel`

Bugun bunlar public config yuzeyinde yoktur.

8. Identity ve Installation Tracking

Fidbek'in son donemdeki en onemli urunsel kabiliyetlerinden biri `identify` ve `installation_id` modelidir.

8.1 Installation ID

Her kurulum icin kalici bir `installation_id` uretilir ve saklanir.

- Android: `SharedPreferences`
- iOS: `UserDefaults`

Bu alan:

- uygulama her acildiginda sifirlanmaz
- app silinirse veya app data temizlenirse degisebilir
- audience / installation inventory ekraninin temel anahtari olarak kullanilir

8.2 Identify

Host app, kurulumun kendi kullanici sistemiyle eslenmesi icin `identify` cagirir.

`identify` alanlari:

- `userId`
- `name`
- `email`

Kural:

- Ucun de ayni anda zorunlu degildir
- En az birinin dolu olmasi gerekir
- `clearIdentity()` logout veya user switch aninda cagrilmalidir

8.3 Kimlik Verisinin Yasam Dongusu

- `identify(...)` cagrildiginda state kalici olarak yazilir
- Sonraki report'larda ayni identity tekrar kullanilir
- `clearIdentity()` cagrilmadikca otomatik sifirlanmaz
- `installation_id` temizlenmez, identity temizlenir

Bu model audience ekraninda su tip sorulari cevaplayabilir hale getirir:

- Bu uygulama hangi kullanicilarda kurulu?
- Bu kullanicinin kac cihazi var?
- Son gorulen surum ne?
- Son 24 saatte aktif mi?

9. Report Akisi

9.1 Report Kategorileri

Temel kategori modeli:

- `bug`
- `improvement` / `feature request`
- `feedback`

Bug kategorisinde occurrence frequency secimi de payload'a eklenebilir.

9.2 Form Akisi

Temel UI adimlari:

- kategori secimi
- mesaj giri■i
- bug ise frequency secimi
- e-posta ve iletisim izni
- screenshot / media
- gonderim

Son donemde eklenen onemli davranis:

- Kullanici `identify` ile e-posta bazli eslenmisse, formdaki email alanina bu e-posta otomatik prefill olur
- Bu kullanicinin yerine checkbox otomatik isaretlenmez; kullanici riza karari hala kendisine aittir

10. Toplanan ve Gonderilen Veriler

Bu bolum urunsel ve hukuki acidan kritiktir. SDK'nin ne gonderdigi acikca bilinmelidir.

10.1 Report Payload

Report submit sirasinda giden ana alanlar:

- `token`
- `category`
- `message`
- `occurrence_frequency` (bug ise)
- `steps` (modelde var, bugun fiilen dolu gelmiyor)
- `installation_id`
- `external_user_id`
- `external_user_name`
- `external_user_email`
- `email` (kullanici formda verdiyse)
- `allow_contact`
- `screenshot_upload_id`
- `attachment_upload_ids`
- `device`
- `app`
- `occurred_at`

10.2 Device Bilgisi

`device` altinda giden alanlar:

- `model`
- `machine`
- `os_name`
- `os_version`
- `locale`

- `time_zone`
- `screen_width`
- `screen_height`
- `interface_style`
- `identifier_for_vendor`

Not:

- Android tarafinda `identifier_for_vendor` fiilen `ANDROID_ID` kaynaklidir
- iOS tarafinda `identifierForVendor` kullanilir

10.3 App Bilgisi

app altinda giden alanlar:

- `bundle_id`
- `version`
- `build`

10.4 Device Ping

SDK configure veya open sonrasinda device ping atilir. Bu akista giden alanlar:

- `token`
- `platform`
- `bundle_id`
- `device_id`
- `installation_id`
- `external_user_id`
- `external_user_name`
- `external_user_email`
- opsiyonel `device`
- opsiyonel `app`

Bu veri report'tan ayridir. Audience ve installation inventory mantigi icin kullanilir.

10.5 Upload Akisi

Screenshot veya medya varsa upload 3 asamada ilerler:

1. `uploads/init`
2. signed upload URL'ye ham binary upload
3. `uploads/complete`

10.6 Toplanmayan Veriler

Kod bazli bugunku durumda asagidakiler gorunmemektedir:

- konum
- rehber
- sms / call log
- installed apps listesi
- clipboard polling
- reklam ID'si
- analytics event stream

Bu ayrim, privacy tarafinda kritik avantajdir.

11. Privacy ve Veri Minimizasyonu Prensipleri

Fidbek tarafinda dogru urun prensibi sunun uzerine kurulmalidir:

- ne toplayabiliyorsak degil, neye ihtiyacimiz varsa onu toplayalim

Pratik olarak 3 katman onerilir:

11.1 Core

Default ve dusuk riskli debug baglami:

- installation_id
- external_user_id
- name / email
- platform
- model
- os_version
- app_version
- locale
- time_zone
- screen context

11.2 Incident Opt-In

Kullanici aksiyonuyla giden veriler:

- screenshot
- video
- attachment
- contact email

11.3 Never by Default

Varsayilan urun politikasina girmemesi gerekenler:

- contacts
- installed apps
- clipboard
- exact location
- ad ID / fingerprinting
- sms / call log

12. Lokalizasyonlar

SDK UI bugun 10 dil hattini destekler:

- English
- Turkish
- Spanish
- French
- German
- Portuguese
- Arabic
- Hindi
- Japanese
- Simplified Chinese

Android kaynak karsiliklari `values-*` klasorlerinde, iOS ve wrapper bundle'lari ise `*.lproj` klasorleri altinda tasinir.

13. Dashboard ile Iliski

SDK yalnizca UI gosteren bir kutuphane degildir; dashboard tarafinda anlamli veri modeli uretir.

13.1 Reports

Raporlar dashboard'da listelenir ve detail ekraninda:

- kategori
- mesaj
- screenshot / medya
- identified user
- contact email
- cihaz ve app baglami

gorulebilir.

13.2 Audience

Installation ping sayesinde dashboard artik:

- installation sayisi
- identified users
- anonymous installations
- active 24h / 7d
- latest installations

gosterebilir.

13.3 Integrations

SDK report akisi dolayli olarak entegrasyonlari da tetikler:

- Slack report notifications
- Webhook report notifications
- Jira issue creation

14. Entegrasyonlarin SDK Ile Iliskisi

14.1 Slack

Yeni report olustugunda Slack webhook uzerinden mesaj gidebilir.

14.2 Webhook

Yeni report olustugunda customer backend'ine signed webhook gidebilir.

14.3 Jira

Report detail uzerinden veya otomatik policy ile Jira issue acilabilir.

Bu entegrasyonlar SDK'nin kendisinde degil, backend + dashboard tarafinda cozulur. SDK yalnizca report ve device verisini uretir.

15. Wrapper Politikasi

Wrapper'larda temel urun karari su sekildedir:

- public API yuzeyi bilincli olarak dar tutulur
- native attachment helper'lari expose edilmez
- entegratorden minimum free text ve minimum yan etki beklenir

Bu karar dogrudur. Sebepler:

- urun yuzeyi sade kalir
- support maliyeti duser
- host app'ler farkli custom media akislariyla kontrolsuz dagilmaz
- native internal API'lar gereksiz yere sozlesmeye donusmez

16. iOS Binary Pipeline

Son donemde en cok sorun cikan alan iOS binary packaging hattiydi. Bu nedenle artik resmi bir script vardir:

- [build-xcframework.sh](/Users/talha/Desktop/Fidbek/source-ios/scripts/build-xcframework.sh)

Script'in yaptigi sey:

1. iOS device release build alir
2. iOS simulator release build alir
3. generated Swift header ve module map dosyalarini toplar
4. device ve simulator framework'lerini stage eder
5. `xcodebuild -create-xcframework` ile tek XCFramework uretir
6. module dosyalarini slice'lara geri senkronlar
7. resource bundle'i kopyalar
8. version'li zip artifact uretir

Bu scriptin eklenme sebebi su iki problemi kalici hale getirmemektir:

- stale module metadata
- wrapper'larda build sirasinda slice / header kaynakli yan etkiler

17. Son Donem iOS Sorunlari ve Cikarilan Dersler

17.1 `missing required module 'CoreMotion'`

Bu hata source code seviyesinde degil, eski packaging / cache kombinasyonlari uzerinden ortaya cikti.

Bugunku dogru durum:

- source-ios tarafinda `CoreMotion` bagimliligi yok
- guncel XCFramework interface'lerinde `import CoreMotion` yok

17.2 `Multiple commands produce ... FidbekSDK-Swift.h`

Bu sorun React Native wrapper podspec'indeki asiri genis `source_files` glob'undan kaynaklandi.

Dogru cozum:

- wrapper source glob'unu sadece bridge dosyalarina daraltmak

17.3 Ana Ders

Binary dagitimi manual adimlarla degil, script ile uretilmelidir.

18. Android Durumu

Android tarafta bugunku binary hattimiz iOS'a gore daha geride ama kullanilabilir durumdadir.

Avantajlar:

- native SDK akisi stabil
- report submit ve upload akisi oturmus
- identify ve installation tracking mevcut
- lokalizasyonlar mevcut

Eksikler:

- surum hattinin iOS ile hizalanmasi gerekir
- dokumantasyon ve release standardizasyonu iOS kadar guncele cekilmelidir

19. Entegrasyon Ornekleri

19.1 Android

```
Fidbek.initialize(application, "SDK_TOKEN")
Fidbek.identify(email = "talha@example.com")
Fidbek.open()
```

19.2 iOS

```
Fidbek.shared.configure(token: "SDK_TOKEN")
Fidbek.shared.identify(email: "talha@example.com")
Fidbek.shared.present()
```

19.3 Flutter

```
await FidbekFlutter.configure(token: 'SDK_TOKEN');
await FidbekFlutter.identify(email: 'talha@example.com');
await FidbekFlutter.open();
```

19.4 React Native

```
await Fidbek.configure({ token: 'SDK_TOKEN' });
await Fidbek.identify({ email: 'talha@example.com' });
await Fidbek.open();
```

20. En Iyi Entegrasyon Pratikleri

- SDK'yi app launch sirasinda configure edin
- login sonrasi `identify(...)` cagrin
- logout sirasinda `clearIdentity()` cagrin
- `open()` cagrilarini uygulama foreground ve UI hazir olduktan sonra yapin
- wrapper'larda native internal API yuzeyini acmaya calismayin
- report toplarken gereksiz hassas veri toplamayin

21. Troubleshooting

21.1 React Native iOS: `missing required module 'CoreMotion'`

Muhtemel neden:

- eski pod / module cache
- eski packaged XCFramework

Yapilacaklar:

- `node_modules` temiz kur
- `Pods` ve `Podfile.lock` temizle
- `DerivedData` ve `ModuleCache.noindex` temizle
- guncel wrapper surumunu kur

21.2 React Native iOS: `Multiple commands produce ... FidbekSDK-Swift.h`

Muhtemel neden:

- eski podspec glob

Yapilacaklar:

- `0.3.3+` veya ustu wrapper kullan
- pod'lari yeniden kur

21.3 Flutter publish dry-run warning

Pub tarafinda dirty git state warning'i release commit'i alinmadan once gorulebilir. Bu publish blocker olmak zorunda degildir ama temiz release disiplini icin commit alinmalidir.

22. Bilinen Tasarim Kararlari

- Wrapper API yuzeyi 5 method ile sinirli tutulur
- Attachment helper'lari native source'ta deprecated tutulur

- `installation_id` kalicidir
- `identify` verisi kalicidir; `clearIdentity()` ile temizlenir
- Dashboard audience modeli cihaz ID uretmekten ziyade installation ve host app kimligi uzerinden kurulur

23. Su Anki Dogruluk Durumu

Bu rehberin hazirlandigi anda teknik olarak dogru kabul edilen ana maddeler:

- iOS native source ve binary hattinda `CoreMotion` bagimliligi yoktur
- iOS binary packaging icin script'li pipeline vardir
- Flutter ve React Native wrapper'lar cekirdek 5 method yuzeyi ile sinirlidir
- iOS wrapper binary'leri `0.3.5` uzerindedir
- Android native binary hattimiz `0.3.0` seviyesindedir

24. Sonuc

Fidbek SDK ailesi bugun yalnizca "feedback formu gosteren bir kutuphane" degildir. Dogru ele alindiginda:

- mobil urunden veri toplar
- bu veriyi dashboard'a baglar
- installation ve audience gorunurlugu saglar
- operasyonel akislari Slack, Webhook ve Jira'ya tasir

En buyuk teknik kazan■m, iOS binary hattinin script'lenmis ve tekrar edilebilir hale gelmis olmasidir.

En buyuk urunsel kazan■m ise `identify + installation tracking + report` kombinasyonunun artik ayni veri modelinde calisiyor olmasidir.