

In-App Purchases in React Native

Ramazan Arslan
Co-founder / CTO



Ramazan Arslan

Co-founder / CTO



Boğaziçi Üniversitesi
Bilgisayar Mühendisliği

ramazan-arslan 

ramazanarslan 

arslan1845 



ex-

intertech

Collegefind me

WesterOps

PARAŞÜT



Started in 2020



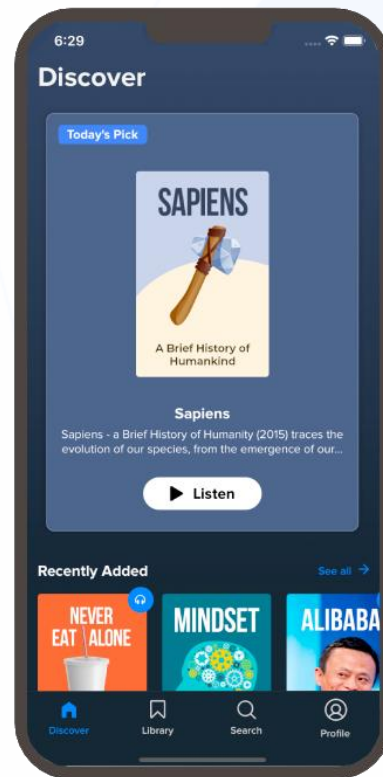
More than 1 Million + Download



4.7 Rate Global

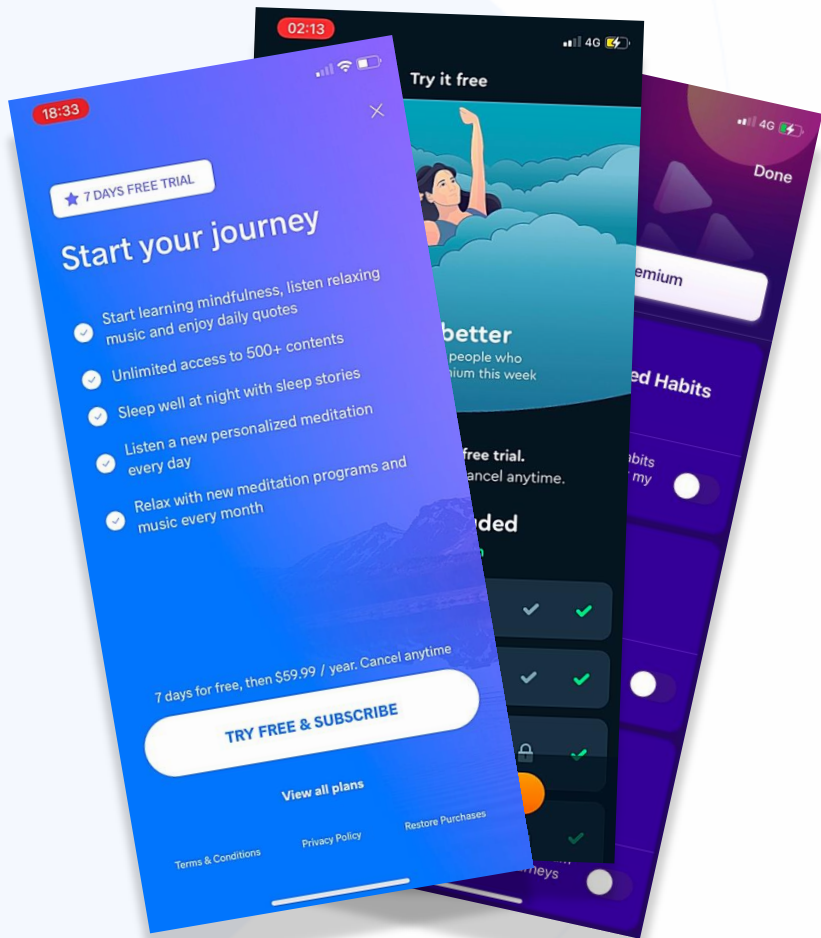


Turkish,
Spanish,
Portuguese,
English,
Russian



Topics

- What is In-App Purchase?
- In-App Purchase Types
- Purchase flow with In-App Purchase
- Subscription Lifecycle
- Testing In-App Purchases with Sandbox



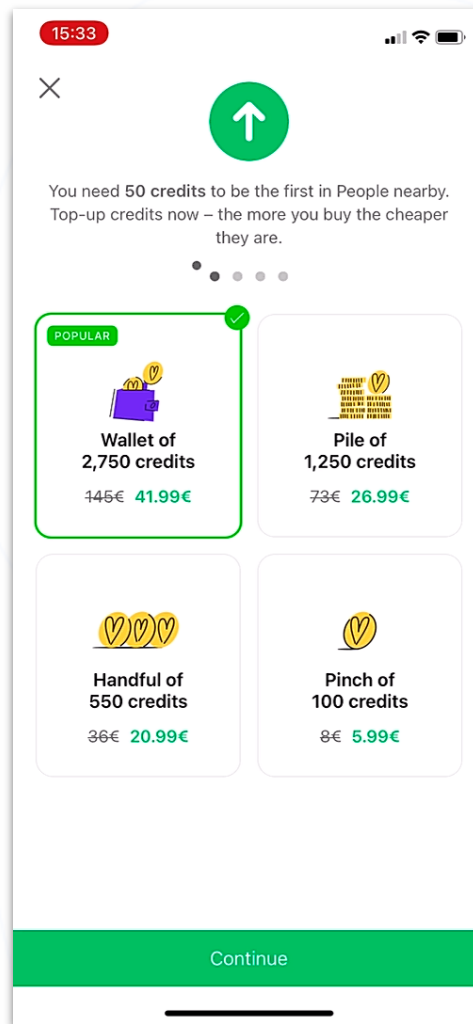
In App Purchases (IAP)

In-App Purchase Types

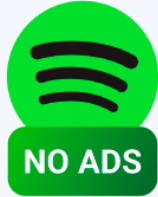
In-App Purchase Types



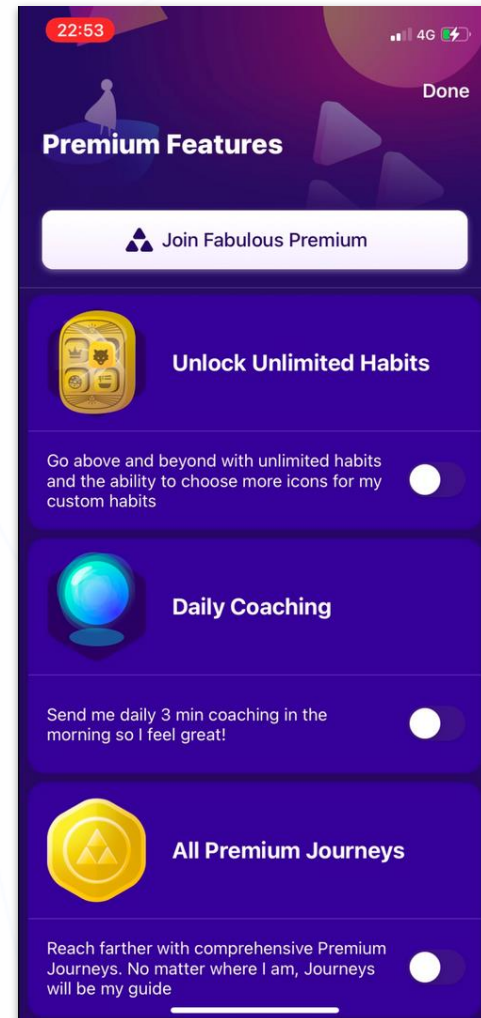
Consumable



In-App Purchase Types



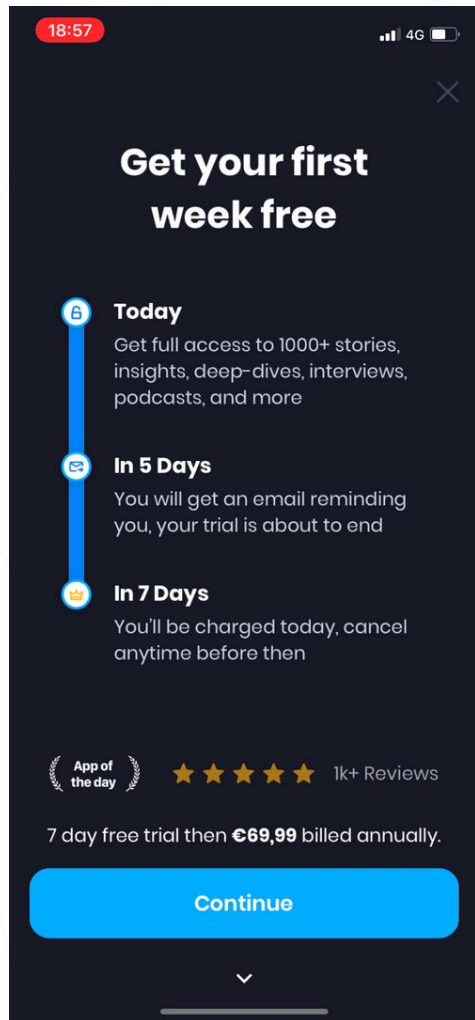
Non-consumable



In-App Purchase Types



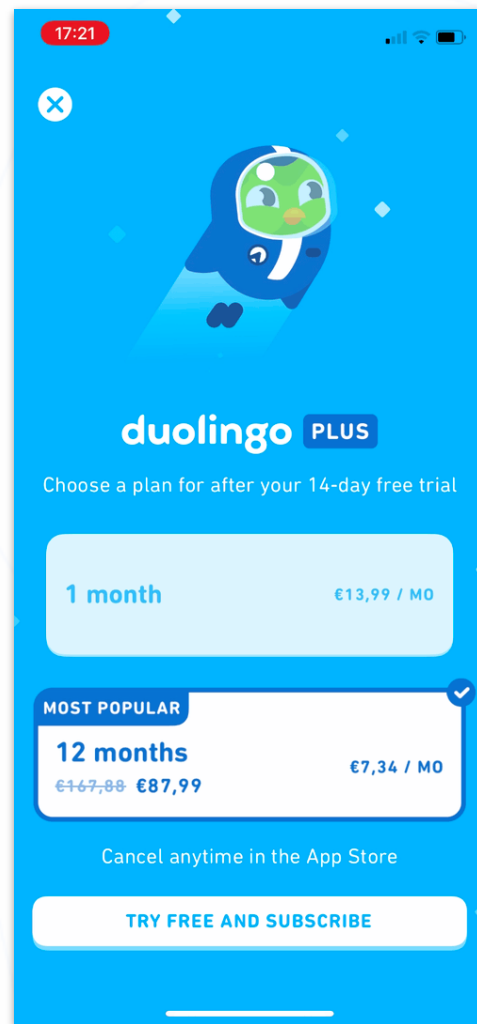
Non-renewing
subscriptions



In-App Purchase Types



Auto-renewable
subscriptions



Purchase Flow with IAP

Accept the Paid Applications Agreement



The screenshot shows the Apple Developer portal interface. At the top, the 'Ocean Journal' logo is visible, along with navigation links for 'App Store', 'Services', 'TestFlight', and 'Xcode Cloud'. The left sidebar contains a menu with sections: 'iOS App' (with a status '1.0 Ready for Sale'), 'Add macOS App', 'Add tvOS App', 'General' (with sub-items like 'App Information', 'Pricing and Availability', 'App Privacy', 'Ratings and Reviews', 'Version History', 'App Review'), and 'Features' (with sub-items like 'In-App Purchases', 'Subscriptions', 'App Store Promotions', 'Custom Product Pages', 'In-App Events', 'Product Page Optimization'). The main content area is titled 'In-App Purchases' and shows a 'Drafts (1)' section. A modal dialog box titled 'Create an In-App Purchase' is open in the center. It contains a description: 'An in-app purchase allows customers to buy content, features, or services from within your app by accessing the App Store and securely processing payments from the user. [Learn More](#)'. Below the description are three input fields: 'Type' (a dropdown menu with 'Select...' and a downward arrow), 'Reference Name' (a text input field), and 'Product ID' (a text input field). To the right of the 'Product ID' field, the text '64' is displayed. At the bottom right of the dialog are two buttons: 'Cancel' and 'Create'. In the background, a yellow warning icon and the text 'Missing Metadata' are visible.

Create an In-App Purchase

An in-app purchase allows customers to buy content, features, or services from within your app by accessing the App Store and securely processing payments from the user. [Learn More](#)

Type ?
Select... ▼

Reference Name ?

Product ID ?
 64

Cancel Create

Missing Metadata

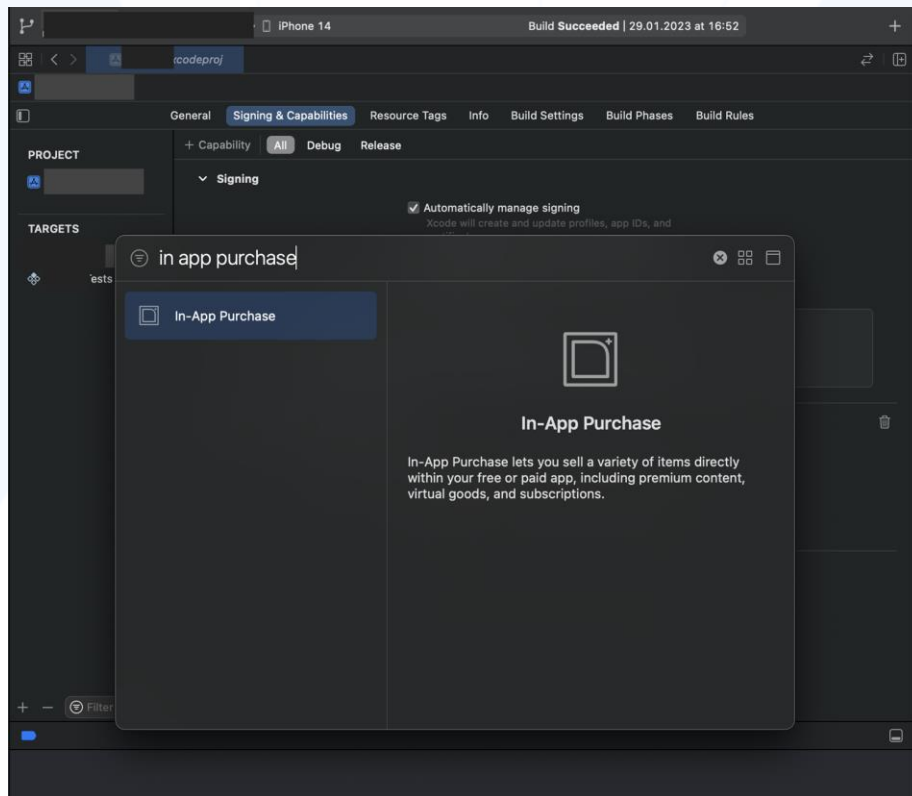
Subscription Prices +[All Prices and Currencies](#)

Below is a summary of your current pricing and any upcoming changes.

▼ **Current Pricing for New Subscribers**[All Countries or Regions](#) ▼

COUNTRIES OR REGIONS (175) ^	PRICE	INTRODUCTORY OFFERS (175)	UPCOMING CHANGES (0)
Afghanistan (USD)	\$7.99	Free for the first 3 days	
Albania (USD)	\$4.79	Free for the first 3 days	
Algeria (USD)	\$7.99	Free for the first 3 days	
Angola (USD)	\$7.99	Free for the first 3 days	
Anguilla (USD)	\$7.99	Free for the first 3 days	
Antigua and Barbuda (USD)	\$7.99	Free for the first 3 days	
Argentina (USD)	\$7.99	Free for the first 3 days	
Armenia (USD)	\$9.49	Free for the first 3 days	
Australia (AUD)	\$9.99	Free for the first 3 days	
Austria (EUR)	€9.49	Free for the first 3 days	

Enable IAP in Xcode



Purchase Flow with IAP

Install the SDK

Native SDKs



StoreKit API



Google Play Billing Library



Flutter

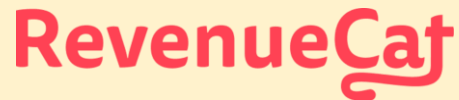
in_app_purchase



React Native

react-native-iap

Third Party SDKs



Qonversion

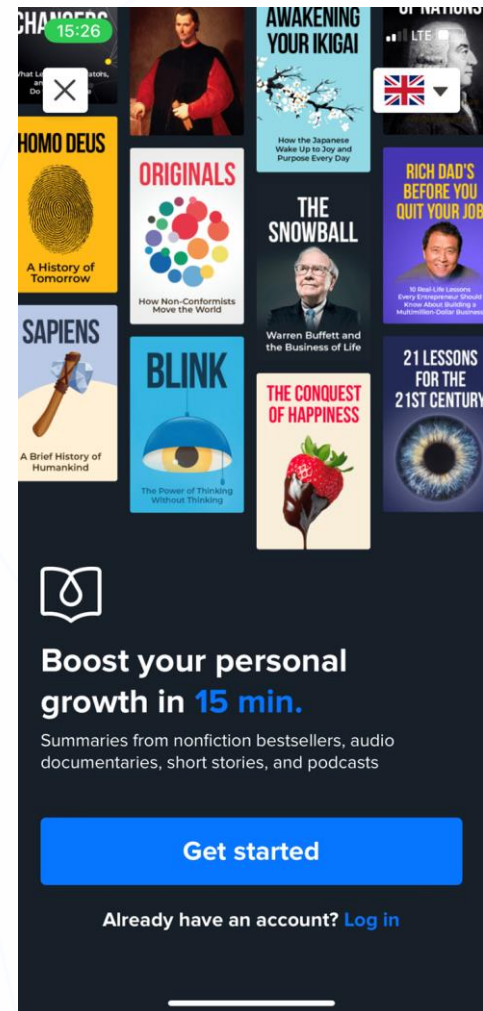


Purchase Flow with IAP

- initialize()
- getProducts()
- getPaywalls()
- openPaywall()
- makePurchase()
- restorePurchase()

Purchase Flow with IAP

- `initialize()`
- `getProducts()`
- `getPaywalls()`
- `openPaywall()`
- `makePurchase()`
- `restorePurchase()`



Purchase Flow with IAP

- initialize()
- **getProducts()**
- getPaywalls()
- openPaywall()
- makePurchase()
- restorePurchase()

```
[
  {
    productId: "YearlyPremium",
    localizedTitle: "Yıllık Abonelik",
    localizedSubscriptionPeriod: "1 Yıl",
    localizedPrice: "₺119,99",
    currencySymbol: "₺",
    price: 119.99,
    //OTHER RELATED FIELDS
  },
  {
    productId: "YearlyPremium",
    localizedTitle: "Yıllık Abonelik",
    localizedSubscriptionPeriod: "1 Yıl",
    localizedPrice: "₺119,99",
    currencySymbol: "₺",
    price: 119.99,
    //OTHER RELATED FIELDS
  },
];
```

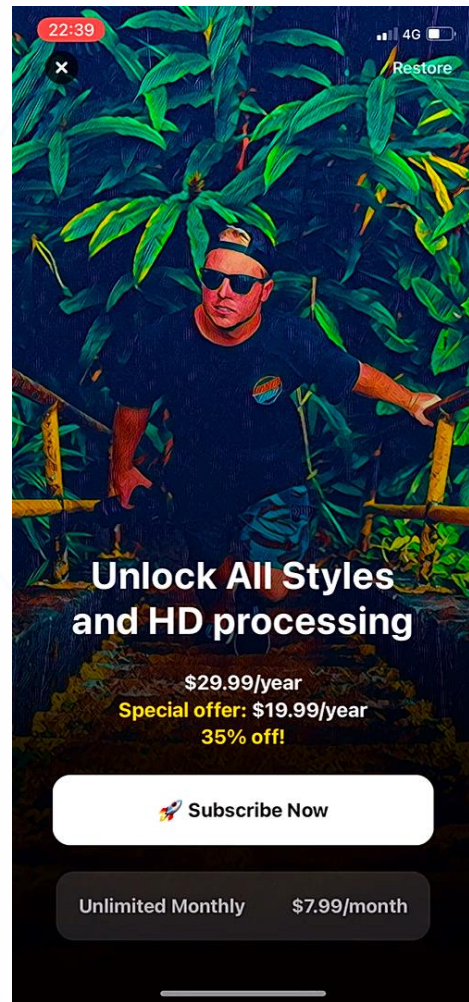
Purchase Flow with IAP

- initialize()
- getProducts()
- **getPaywalls()**
- openPaywall()
- makePurchase()
- restorePurchase()

```
{
  title: "Try premium, get an unlimited access
  closeButtonDelay: 1000,
  purchaseButton: {
    i18n: { tr: "DEVAM ET", en: "CONTINUE"},
    color: "blue",
  },
}
```

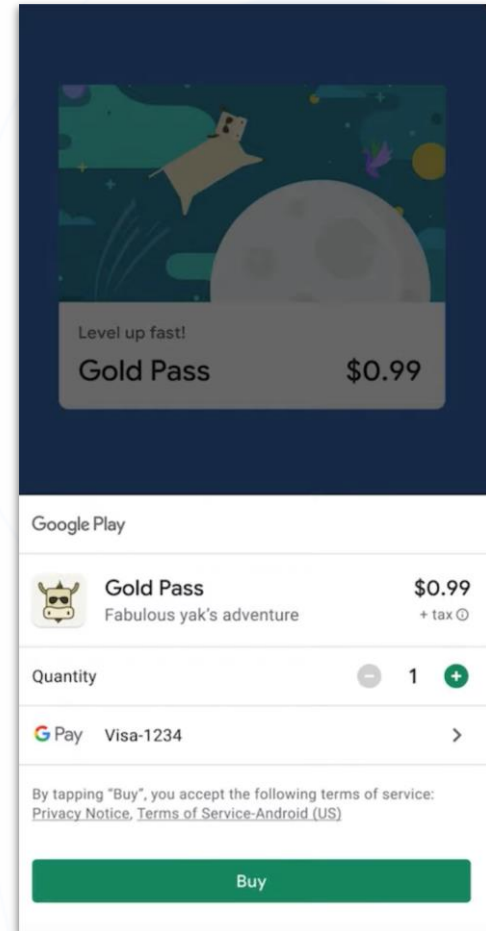
Purchase Flow with IAP

- initialize()
- getProducts()
- getPaywalls()
- **openPaywall()**
- makePurchase()
- restorePurchase()



Purchase Flow with IAP

- initialize()
- getProducts()
- getPaywalls()
- openPaywall()
- **makePurchase()**
- restorePurchase()



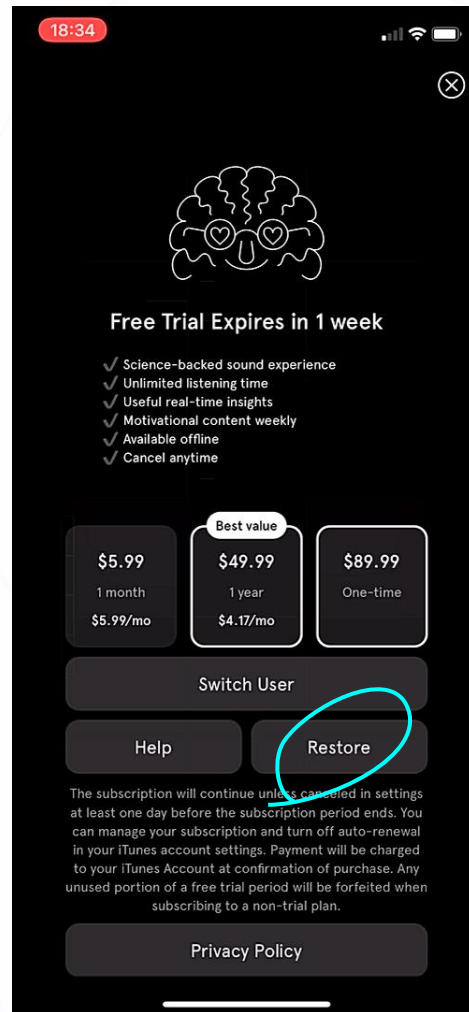
Purchase Flow with IAP

- initialize()
- getProducts()
- getPaywalls()
- openPaywall()
- **makePurchase()**
- restorePurchase()

```
{  
  productId: "YearlyPremium",  
  originalTransactionId: "123213123213",  
  transactionId: "123213123213",  
  expiresAt: "2023-09-05T14:01:32",  
  startsAt: "2022-09-05T14:01:32",  
  isSandbox: true,  
  isRefund: false,  
}
```

Purchase Flow with IAP

- initialize()
- getProducts()
- getPaywalls()
- openPaywall()
- makePurchase()
- **restorePurchase()**



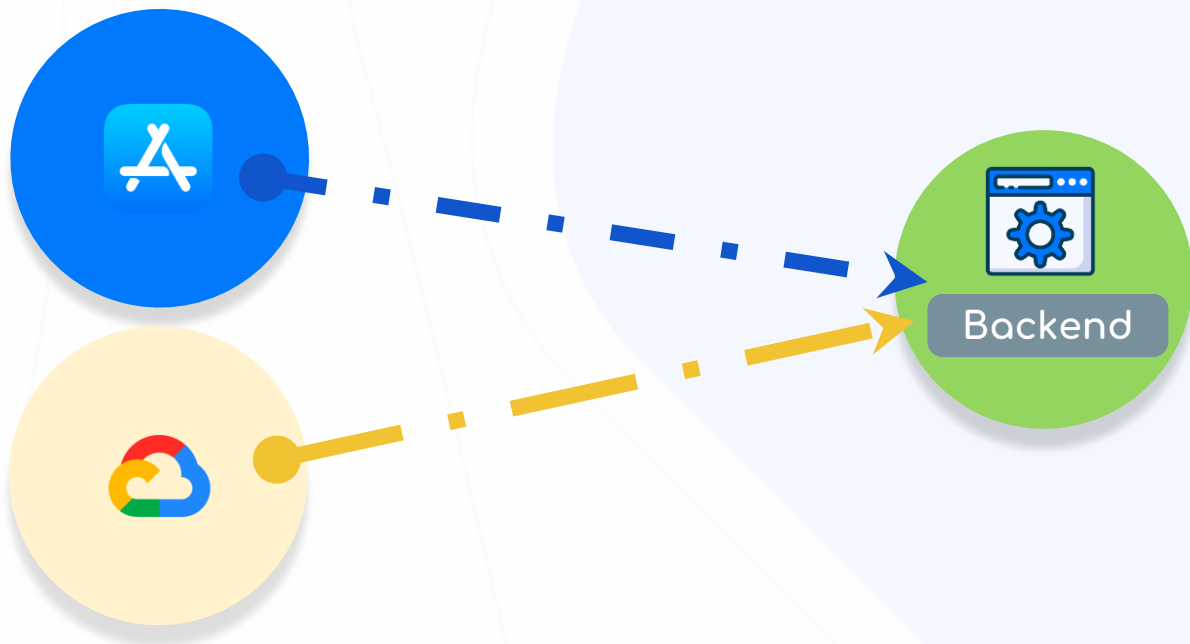
Purchase Flow



Events

- Subscription started
- Subscription renewed
- Subscription cancelled
- Trial started
- Trial converted
- Trial cancelled
- Trial reactivated
- Entered grace period
- Refunded

Server to Server (S2S)





App Store Connect S2S

iOS App

✓ 4.51 Ready for Sale

[Add macOS App](#)

[Add tvOS App](#)

General

App Information

Pricing and Availability

App Privacy

Promotional Artwork

Ratings and Reviews

Version History

In-App Purchases

Manage

Subscription Groups

App Store Promotions

App Information

This information is used for all platforms of this app. Any changes will be released with your next app version.

[Apple's Standard License Agreement](#)

✓ Saved

App Store Server Notifications

App Store server notifications provide information about key events related to your in-app purchases. To test notifications before implementing them in production, you can set up a separate sandbox server URL. [Learn More](#)



Version 2 notifications cover more events in the customer lifecycle, including subscription expirations, offer redemptions, refunds, and more. To update to the latest version of the App Store server notifications, select Edit next to Production Server URL or Sandbox Server URL.

Production Server URL [Edit](#)

https://a 38f3da419db319ca96c31c3dd1/

Sandbox Server URL [Edit](#)

https://a419db319ca96c31c3dd1/

Additional Information

[View on App Store](#)

[Edit User Access](#)

[Remove App](#)



Google Cloud Platform S2S

Google Cloud

pub/sub

Pub/Sub

Pub/Sub

Topics

Subscriptions

Snapshots

Schemas

Pub/Sub Lite

Lite Reservations

Lite Topics

Lite Subscriptions

Subscription name

projects/ / kbackend

Topic name

projects/ / 4-48e2-9104-c0a66ded7317

Delivery type ?

☐ Pull

☒ Push

☐ Write to BigQuery

A variant of the push operation. Select this option if you want Pub/Sub to deliver messages directly to an existing BigQuery table. [Learn more](#)



Endpoint URL *

https://a / hook?platform=android




☐ Enable authentication [Learn more](#)



Quality

-  Ratings and reviews
-  Android vitals



Monetize

-  Products
-  Promo codes
-  Financial reports



Monetization setup

Policy

-  Policy status
-  App content

Monetization setup

Set up real-time event notifications and app licensing

Real-time developer notifications

Receive event notifications from Google Cloud Pub/Sub for updates to your in-app products and subscriptions.

[Learn more](#)

Topic name

Cloud Pub/Sub topic name in the format projects/{project_id}/topics/{topic_name} 0 / 300

Send test notification



Make sure the Cloud Pub/Sub topic exists and that you have given Google the necessary permissions to publish notifications. [Learn more](#)

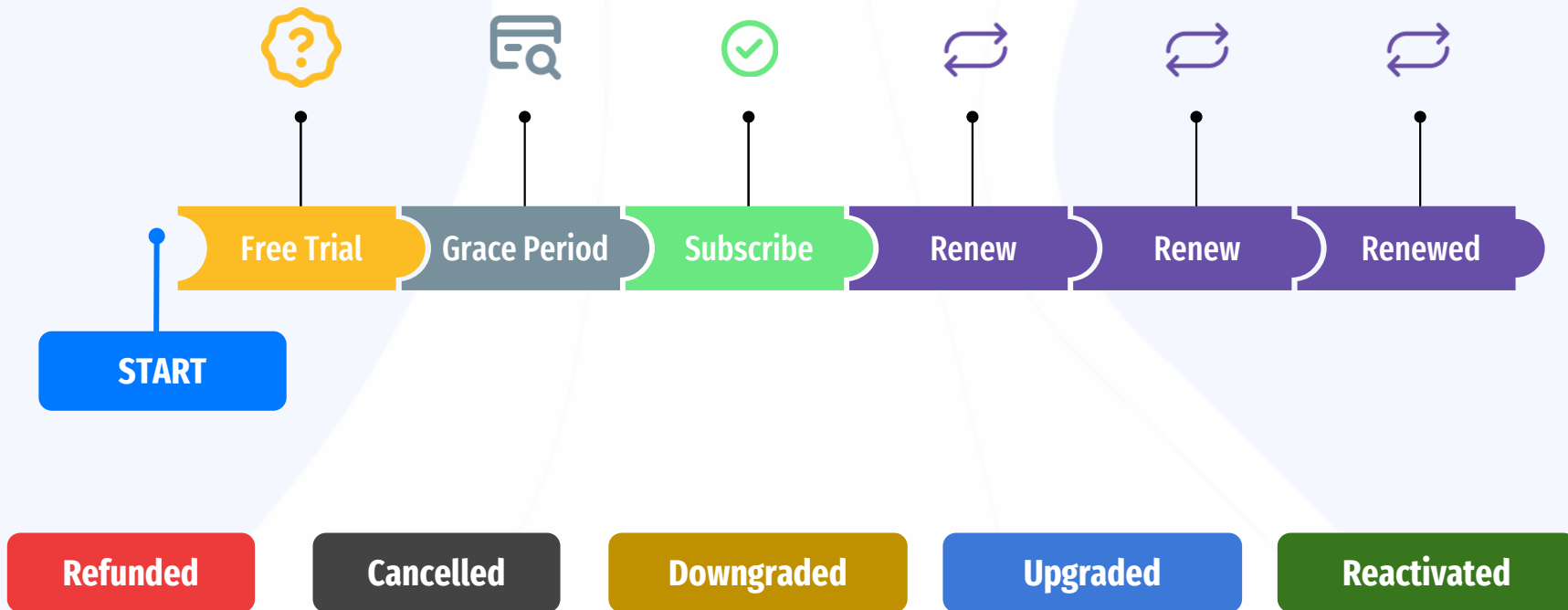
Licensing

Prevent the unauthorized distribution of your app and test in-app billing integrations. [Learn more](#)

Discard changes

Save changes

Subscription Flow








Users and Access

[People](#)[Sandbox Testers](#)[Keys](#)[Shared Secret](#)[Xcode Cloud](#)

Sandbox Testers (3)


Your development-signed apps use the sandbox environment when you sign in to the App Store using a Sandbox Apple ID. To repeatedly test common in-app purchase scenarios for a sandbox purchase history. [Learn More](#)


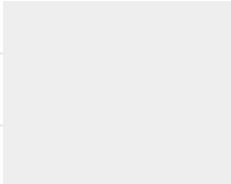
SANDBOX APPLE ID	NAME ^	APP STORE COUNTRY OR REGION	INTERRUPTED PURCHASES ?
		Turkey	
		Turkey	
		Turkey	

-  Account details
-  Developer page
-  Associated developer accounts
-  Activity log
-  **Setup**
- Email lists
- Pricing templates
- Game projects
- License testing**
- Payments profile
- API access
- Linked services
- Notifications
- Benchmarking
- App transfer

License testing

Test your licensing and in-app billing integrations. [Show more](#)

Choose license testers from the email lists 

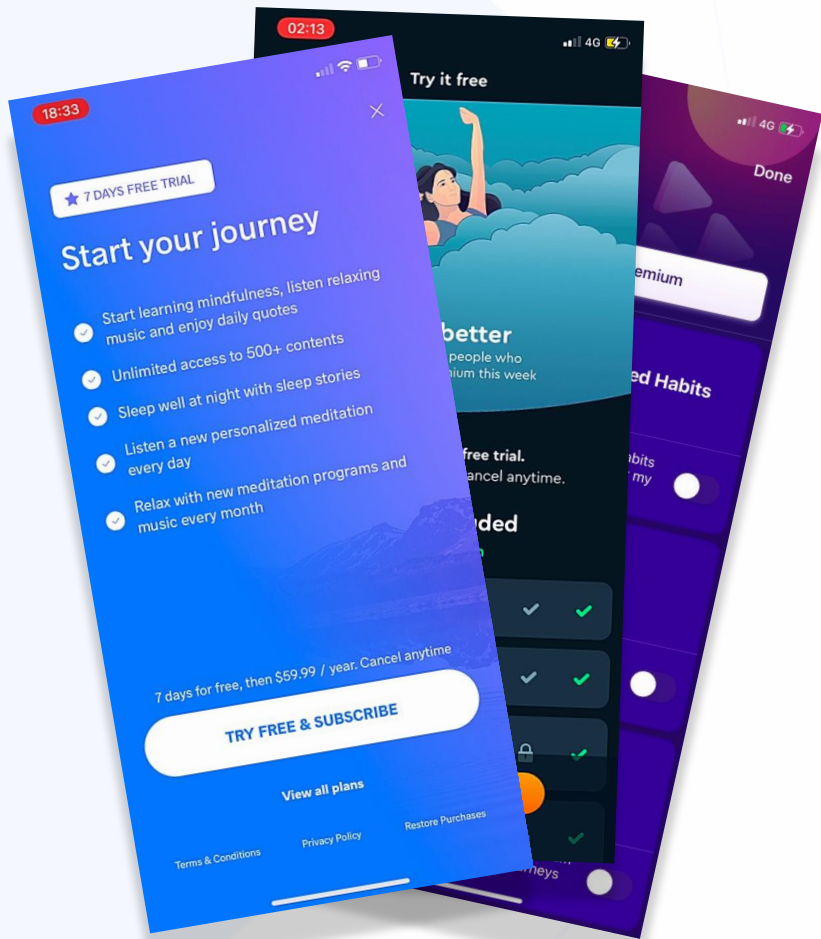
	List name	Users	
<input type="checkbox"/>		18	→
<input type="checkbox"/>		2	→
<input type="checkbox"/>		3	→
<input checked="" type="checkbox"/>	License testers (auto-generated)	18	→

[Create email list](#)

License response

RESPOND_NORMALLY 

All license testers will get this response.



In-App Purchases in React Native

Ramazan Arslan
Co-founder / CTO