

[App Store Server Notific...](#) / `notification_type` **Deprecated**

Type

notification_type **Deprecated**

The type that describes the in-app purchase event for which the App Store sends the version 1 notification.

App Store Server Notifications 1.0–2.8 **Deprecated**

```
string notification_type
```

Deprecated

Implement the [App Store Server Notifications V2](#) endpoint on your server to receive version 2 notifications instead.

Possible Values

CANCEL

Indicates that Apple Support canceled the auto-renewable subscription and the customer received a refund as of the timestamp in `cancellation_date_ms`.

CONSUMPTION_REQUEST

Indicates that the customer initiated a refund request for a consumable in-app purchase, and the App Store is requesting that you provide consumption data. For more information, see [Send Consumption Information](#).

DID_CHANGE_RENEWAL_PREF

Indicates that the customer made a change in their subscription plan that takes effect at the next renewal. The currently active plan isn't affected. Check the `auto_renew_product_id`

field in `unified_receipt.Pending_renewal_info` to retrieve the product identifier for the product the customer's subscription renews.

DID_CHANGE_RENEWAL_STATUS

Indicates a change in the subscription renewal status. In the JSON response, check `auto_renew_status_change_date_ms` to retrieve the date and time of the last status update. Check `auto_renew_status` to get the current renewal status.

DID_FAIL_TO_RENEW

Indicates a subscription that failed to renew due to a billing issue. Check `is_in_billing_retry_period` to retrieve the current retry status of the subscription. Check `grace_period_expires_date` to get the new service expiration date if the subscription is in a billing grace period.

DID_RECOVER

Indicates a successful automatic renewal of an expired subscription that failed to renew in the past. Check `expires_date` to determine the next renewal date and time.

DID_RENEW

Indicates that a customer's subscription has successfully auto-renewed for a new transaction period. Provide the customer with access to the subscription's content or service.

INITIAL_BUY

Occurs at the user's initial purchase of the subscription. Store `latest_receipt` on your server as a token to verify the user's subscription status at any time by validating it with the App Store.

INTERACTIVE_RENEWAL

Indicates the customer renewed a subscription interactively, either by using your app's interface, or on the App Store in the account's Subscriptions settings. Make service available immediately.

PRICE_INCREASE_CONSENT

Indicates that the App Store has started asking the customer to consent to your app's auto-renewable subscription price increase that requires consent. In the `unified_receipt.Pending_renewal_info` object, the `price_consent_status` value is 0 to indicate that the user hasn't yet responded to the price increase. The App Store server sets the `price_consent_status` to 1 when the customer consents to the price increase. Check the latest price consent status by calling the `Get All Subscription Statuses` endpoint in the App Store Server API. Check the `priceIncreaseStatus` field in the `JWSRenewalInfoDecodedPayload`. You can also call `verifyReceipt` to view the updated price consent status. For more information about how StoreKit calls your app before it displays

the price consent sheet for subscription price increases that require customer consent, see [paymentQueueShouldShowPriceConsent\(_:\)](#). For more information about managing subscription prices, see [Managing Prices](#).

REFUND

Indicates that the App Store successfully refunded a transaction for a consumable in-app purchase, a non-consumable in-app purchase, or a non-renewing subscription. The `cancellation_date_ms` contains the timestamp of the refunded transaction. The `original_transaction_id` and `product_id` identify the original transaction and product. The `cancellation_reason` contains the reason.

REVOKE

Indicates that an in-app purchase the user was entitled to through Family Sharing is no longer available through sharing. StoreKit sends this notification when a purchaser disabled Family Sharing for a product, the purchaser (or family member) left the family group, or the purchaser asked for and received a refund. Your app will also receive a [paymentQueue\(_:didRevokeEntitlementsForProductIdentifiers:\)](#) call. For more information about Family Sharing, see Supporting [Supporting Family Sharing in your app](#).

RENEWAL (DEPRECATED)

As of March 10, 2021 this notification is no longer sent in production and sandbox environments. Update your existing code to rely on the `DID_RECOVER` notification type instead.

Mentioned in

 [App Store Server Notifications changelog](#)

 [Receiving App Store Server Notifications](#)

Discussion

You receive and can react to server notifications in real time for the subscription and refund events that these notification type values describe. The `notification_type` appears in the [responseBodyV1](#).

Note

If you're receiving App Store Server Notifications Version 2, see [responseBodyV2](#) and [notificationType](#) instead.

Handle use cases for notification events

When events occur that affect the customer’s product and subscription life-cycle, your server receives notifications from the App Store. Here are some examples of product events and the server notifications you can expect to receive:

Subscription or in-app purchase event	Notification types
Customer completed an initial purchase of a subscription	INITIAL_BUY
Subscription is active; customer upgraded to another SKU	DID_CHANGE_RENEWAL_STATUS, INTERACTIVE_RENEWAL
Subscription is active; customer downgraded to another SKU	DID_CHANGE_RENEWAL_PREF
Subscription has expired; customer resubscribed to the same SKU	DID_CHANGE_RENEWAL_STATUS
Subscription has expired; customer resubscribed to another SKU (upgrade or downgrade)	INTERACTIVE_RENEWAL, DID_CHANGE_RENEWAL_STATUS
Customer canceled the subscription from the App Store Subscriptions settings page. Their subscription will not auto-renew and will expire on the expires_date	DID_CHANGE_RENEWAL_STATUS
Customer previously canceled the subscription, but now resubscribed to same product before the subscription expired. The subscription will auto-renew on the expires_date	DID_CHANGE_RENEWAL_STATUS
AppleCare refunded a subscription	CANCEL, DID_CHANGE_RENEWAL_STATUS
Subscription failed to renew because of a billing issue	DID_FAIL_TO_RENEW
Expired subscription recovered by App Store through a billing retry	DID_RECOVER
Subscription churned after failed billing retry attempts	DID_CHANGE_RENEWAL_STATUS

Subscription or in-app purchase event	Notification types
AppleCare successfully refunded the transaction for a consumable, non-consumable, or a non-renewing subscription	REFUND
You've increased the price of an auto-renewable subscription and the price increase requires customer consent before the subscription auto-renews	PRICE_INCREASE_CONSENT
Subscription successfully auto-renewed	DID_RENEW
A purchaser disabled Family Sharing for a product, the purchaser (or family member) left the family group, or the purchaser asked for and received a refund	REVOKE
The customer initiated a refund request for a consumable in-app purchase	CONSUMPTION_REQUEST

Receive notifications for the purchaser and family members

The following table identifies the notifications you receive for the purchaser and for their family members who share products through Family Sharing. To determine if a notification is for the purchaser or a family member, check the value of the `in_app_ownership_type` field, which appears in the `unified_receipt.latest_receipt_info` of the `responseBody` object. For more information about Family Sharing, see [Supporting Family Sharing in your app](#).

Notification type	Received for Purchaser	Received for Family Members
CANCEL	YES	NO
CONSUMPTION_REQUEST	YES	N/A
DID_CHANGE_RENEWAL_PREF	YES	YES
DID_CHANGE_RENEWAL_STATUS	YES	YES
DID_FAIL_TO_RENEW	YES	YES
DID_RECOVER	YES	YES
DID_RENEW	YES	YES

Notification type	Received for Purchaser	Received for Family Members
INITIAL_BUY	YES	NO
INTERACTIVE_RENEWAL	YES	YES
PRICE_INCREASE_CONSENT	YES	NO
REFUND	YES	NO
REVOKE	NO	YES
RENEWAL (Deprecated)	N/A	N/A

The CONSUMPTION_REQUEST notification applies to consumable in-app purchases, which aren't eligible for Family Sharing.

Test notification events with sandbox

Your development-signed apps use the sandbox environment when you sign in to App Store using a Sandbox Apple Account. To create a Sandbox Apple Account or test account in App Store Connect, see [Create a sandbox tester account](#).

If you enabled App Store Server Notifications, test your logic for transactions in the sandbox environment. To determine if a notification for a subscription event occurred in the test environment, check whether the value of the `environment` field in the JSON `responseBodyV1` object equals `Sandbox`.

The following notification types are available in sandbox: `INITIAL_BUY`, `DID_CHANGE_RENEWAL_PREF`, `DID_CHANGE_RENEWAL_STATUS`, `DID_RENEW`, `INTERACTIVE_RENEWAL`, `CANCEL`, and `REFUND`. Notifications in the sandbox environment are for the purchaser only, and have `in_app_ownership_type` equal to `PURCHASED`. For more information about testing in-app purchases, see [Testing In-App Purchases with sandbox](#).

See Also

Version 1 notifications

~~App Store Server Notifications V1~~

Specify your secure server's URL in App Store Connect to receive version 1 notifications.

Deprecated

~~object responseBodyV1~~

The response body containing JSON data that the App Store sends in a version 1 server notification.

Deprecated