Article

Specifying prices for Advanced Commerce SKUs

Provide prices for SKUs with the supported number of decimal places, in milliunits of currency.

Discussion

Apps that use the Advanced Commerce API manage their own catalog of in-app purchases and their respective SKUs, including their prices. When supplying a price, be sure to use the supported number of decimal places, as shown in the section below. In the API, provide the price value in milliunit format.

When setting a price for your SKU, it's strongly recommended that you choose from the 900 price points that the App Store supports across 175 storefronts and 44 currencies. To view or download a .csv file that includes all the storefronts, currencies, and price points:

- In App Store Connect, log in with an App Manager role.
- Select Apps, and choose your app.
- In the left menu, select Pricing and Availability.
- Select Available Prices by Country or Region.
- Select Download All Prices and Currencies.

For price point information by currency, see App Store Pricing Update.

Determine the currency at runtime

To determine the currency to use at runtime, check the device's App Store storefront. For a list of currencies that the App Store supports for each storefront, see Financial reports regions and currencies. For information about getting the current storefront in the app, see the `current` property of `Storefront`. For more information on currency, see `currency`.

Provide prices in milliunit format in API calls

The Advanced Commerce API accepts prices in milliunit format, as noted in the documentation, such as for the `price` type.

To determine a price in milliunits, multiply the price by 1000. One unit of a currency equals 1000 milliunits. The following table shows examples of valid prices in milliunit format:

Price	Milliunits
USD 1.99	1990
KRW 3300	3300000
JPY 359	359000

Note

The payment sheet and other customer communications automatically display prices in the standard format, not in milliunits.

Use the supported number of decimal places for prices

The App Store supports either zero or two decimal places for prices, depending on the currency.

The following table shows examples of valid and invalid prices, based on the number of decimal places they use:

Currency	Supported decimal places	Valid price example (and milliunit equivalent)	Invalid price example (and milliunit equivalent)
USD	2	1.45 (1450)	$1.095 (1095)
JPY	0	310 (310000)	310.95 (310950)

Important

Don’t exceed the supported number of decimal places when you supply a price in the Advanced Commerce API. If your API call includes more decimal digits on a price than its currency supports, the system doesn’t display the payment sheet and fails with an error.

Look up the supported number of decimal places by currency code

The following currencies don’t support any decimal places:

Currency code	Decimal places
CLP	0
COP	0
DKK	0
HKD	0
HUF	0
IDR	0
INR	0
JPY	0
KRW	0
KZT	0
MXN	0
NGN	0
NOK	0
PHP	0
PKR	0

Currency code	Decimal places
RUB	0
SEK	0
THB	0
TWD	0
TZS	0
VND	0

The following currencies support two decimal places:

Currency code	Decimal places
AED	2
AUD	2
BGN	2
BRL	2
CAD	2
CHF	2
CNY	2
CZK	2
EGP	2
EUR	2
GBP	2
ILS	2
MYR	2

Currency code	Decimal places
NZD	2
PEN	2
PLN	2
QAR	2
RON	2
SAR	2
SGD	2
TRY	2
USD	2
ZAR	2

See Also

Tax codes and pricing

Choosing tax codes for your SKUs

Select a tax code for each SKU that represents a product your app offers as an in-app purchase.

Handling subscription price changes

Provide necessary customer communications to notify and gather applicable consent before you initiate a price change.