Reminder
1. Starting August 2, 2023, all new applications must use the Billing Library version 5 or newer. Starting November 1, 2023, all updates for existing applications must use Version 5 Billing Library or newer. Learn more (https://developer.android.com/google/play/billing/deprecation-faq?hl=id).
2. If your application targets Android 14 or newer, you must update to PBL 5.2.1 (https://developer.android.com/google/play/billing/release-notes?hl=id#5-2-1) or PBL 6.0.1 (https://developer.android.com/google/play/billing/release-notes?hl=id#6-0-1) or higher.

AIDL Reference Google Play Billing Service

Warning: AIDL is no longer used and will be deleted in future releases. To apply billing feature for Google Play, use Google Play Billing Service Library (https://developer.android.com/google/play/billing?hl=id).

This documentation presents technical reference information related to the use of AIDL Google Play Billing Service.

Server Response Code

The following table contains all server response codes sent from Google Play to your application. Google Play sends a response code simultaneously as an integer mapped to the key `RESPONSE_CODE` in `Bundle` response. Your application must handle all this response

code.

Seasonal subscriptions are no longer used starting June 15, 2019. Google Play now display **`BILLING_RESPONSE_RESULT_DEVELOPER_ERROR`** for subscription purchases new season.

Table 1. Summary of response codes for AIDL calls Google Play Billing Services.

Response Code	ValueDescription	
`BILLING_RESPONSE_RESULT_OK`	0	Success
`BILLING_RESPONSE_RESULT_USER_CANCELED`	1	The user presses back or cancels the dialog
`BILLING_RESPONSE_RESULT_SERVICE_UNAVAILABLE`	2	Weakening network connection
`BILLING_RESPONSE_RESULT_BILLING_UNAVAILABLE`	3	AIDL version of Google Play Billing Service is not supported for the type requested
`BILLING_RESPONSE_RESULT_ITEM_UNAVAILABLE`	4	The requested product is not available for purchase

`BILLING_RESPONSE_RESULT_DEVELOPER_ERROR`	5	Invalid arguments are given to the API. This error can also indicate that the application not signed correctly or not prepared properly for billing, or do not have zin needed in the manifest.
`BILLING_RESPONSE_RESULT_ERROR`	6	Error fatal during API action
`BILLING_RESPONSE_RESULT_ITEM_ALREADY_OWNED`	7	Failure to buy because the item is already owned
`BILLING_RESPONSE_RESULT_ITEM_NOT_OWNED`	8	Failure to use because the item is not owned

AIDL Reference Google Play Billing Service - Support

This section explains methods for obtaining information about types of support billing available for your application.

Method isBillingSupported()

This method indicates whether:

- The version of the API provided is supported for your application.
- Google Play supports billing in user countries.
- The Google Play billing system is activated in your application package.

- Your application can use the types of items provided for billing purposes.

Table 2. Parameter `isBillingSupported()`.

Key	Type	Description
`apiVersion`	`int`	AIDL version number Google Play Billing Service that your application uses.
`packageName`	`String`	The name of the application package that calls this method.
`type`	`String`	Value must be `inapp` for products in applications or `subs` for subscriptions.

This method is available in AIDL Google Play Billing Service version 3 and newer.

Method isBillingSupportedExtraParams()

This method is identical to `isBillingSupported()` (#isBillingSupported), except You can continue the fourth argument, `Bundle` (https://developer.android.com/reference/android/os/Bundle?hl=id), which can contains extra parameters.

Table 3. Parameter `isBillingSupportedExtraParams()`.

Key	Type	Description
`apiVersion`	`int`	AIDL version number Google Play Billing Service that your application uses.

`packageName`	`String`	The name of the application package that calls this method.
`type`	`String`	Value must be **`inapp`** for products in applications or **`subs`** for subscriptions.
`extraParams`	Bundle (https://developer.android.com/reference/android/os/Bundle?hl=id)	A collection of additional parameters that further determine the type of system Google Play billing supported by the application. This package can contain an optional key called **`vr`** which has a boolean value. This sign determines whether this application supports plot purchase virtual reality (VR) (https://developers.google.com/vr/android/?hl=id).

This method is available in AIDL Google Play Billing Service version 7 and newer.

AIDL Reference Google Play Billing Service - Details

AIDL Google Play Billing Service is specified in the file `IInAppBillingService.aidl`, which is included in sample application (https://developer.android.com/training/google-play-billing/preparing-iab-app?hl=id#GetSample) Version 3.

Method getSkuDetails ()

Use method `getSkuDetails()` this is to get product details from the list Appropriate product ID.

Table 4. Parameter `GetSkuDetails()`.

Key	Type	Description
`apiVersion`	`int`	AIDL version number Google Play Billing Service that your application uses.
`packageName`	`String`	The name of the application package that calls this method.
`type`	`String`	Item types in the application ("inapp "for one-time purchases and" subs "for subscriptions).
`skusBundle`	`Bundle`	Package that contains `StringArrayList` SKU with key . `ITEM_ID_LIST`

If method `getSkuDetails()` successful, then Google Play will send `Bundle` response. Query results are stored at `Bundle` in String `ArrayList` mapped to the key `DETAILS_LIST`. Each String in the detailed list contains product details for one product in JSON format. Column in JSON string with product details summarized in table 5.

Table 5. Description of the JSON column with product item details displayed from the request `getSkuDetails()`.

Key	Description
`productId`	Product ID for the product.
`type`	Value must be `inapp` for products in applications or `subs` for subscriptions.
`price`	Price of item format, including currency markings. Price not including tax.

`price_amount_micros`	Prices in micro units, with 1,000,000 micro units, are equivalent to one unit currency. For example, if `price` is , then is . This value symbolizes the price that has been localized and rounded for a particular currency.`"€7.99"price_amount_micros"7990000"`
`price_currency_code`	ISO 4217 currency code (https://en.wikipedia.org/wiki/ISO_4217#Active_codes) for `price`. For example, if `price` determined in British pound sterling, is .`price_currency_code"GBP"`
`title`	Product title.
`description`	Product Description.
`subscriptionPeriod`	Subscription period, specified in the format ISO 8601 (https://en.wikipedia.org/wiki/ISO_8601). For example, `P1W` equal to one week, `P1M` equal to one month, `P3M` equal to three months `P6M` equal to six months, and `P1Y` same as one year. ★ **Note**: Shown only for subscriptions.
`freeTrialPeriod`	The trial period is configured in the Google Play Console, specified in format ISO 8601 (https://en.wikipedia.org/wiki/ISO_8601). For example `P7D` equal to seven days. To learn about the feasibility of further free trials, see Subscription in the Application (https://developer.android.com/google/play/billing/billing_subscriptions.html?hl=id#trials). ★ **Note**: Shown only for subscriptions with configured trial periods.

`introductoryPrice`	Formatted introductory prices for subscriptions, including marks currency, like . Prices do not include tax. `€3.99`
	★ **Note**: Shown only for subscriptions with periods pre-configured introduction.
`introductoryPriceAmountMicros`	Introductory price in micro units. The currency is the same as . `price_currency_code`
	★ **Note**: Shown only for subscriptions with periods pre-configured introduction.
`introductoryPricePeriod`	Introductory price billing period, specified in format ISO 8601 (https://en.wikipedia.org/wiki/ISO_8601).
	★ **Note:** Returned only for subscriptions with preface period that has been configured.
`introductoryPriceCycles`	Number of subscription billing periods that provide introductory prices to users, like 3.
	★ **Note**: Shown only for subscriptions with periods pre-configured introduction.

Method getBuyIntent()

This method displays the response code integer mapped to the key `RESPONSE_CODE`, and `PendingIntent` (https://developer.android.com/reference/android/app/PendingIntent?hl=id) to launch the item purchase path in the application mapped to the key `BUY_INTENT`, as explained in Implement Google Play Billing Service (https://developer.android.com/google/play/billing/billing_integrate?hl=id#purchase). When receiving `PendingIntent` (https://developer.android.com/reference/android/app/PendingIntent?hl=id), Google Play send response `Intent` along with data for the purchase order. Data displayed in `Intent` responses are summarized in table 6.

Note: Instead of using this method, you should use **`getBuyIntentExtraParams()`** (#getBuyIntentExtraParams), that provide additional functions.

Table 6. Response data from Google Play purchase requests.

Key	Description
`RESPONSE_CODE`	Value is **`0`** if the purchase is successful, error if not.
`INAPP_PURCHASE_DATA`	String in JSON format that contains details about purchase orders. See table 7 for a description of the JSON column.
`INAPP_DATA_SIGNATURE`	A string containing signed purchase data with a developer private key. Data signatures use schemas RSASSA-PKCS1-v1_5.

Table 7 describes the JSON column returned in the response data for purchase order.

Table 7. JSON column description for `INAPP_PURCHASE_DATA`.

Column	Description
`autoRenewing`	Duplicate whether the subscription is automatically extended. If `true`, active subscription, and will automatically renewed on the next billing date. If `false`, indicates that the user has canceled the subscription. User has access to subscription content until the next billing date and will loss of access at that time, unless the user reactivates the automatic renewal (or manually extend, as explained in Manual Extension (https://developer.android.com/google/play/billing/billing_subscriptions?hl=id#manual-renewal)). If you offer grace period (https://developer.android.com/google/play/billing/billing_subscriptions?hl=id#grace-period), this value is still set to `true`for all subscription, as long as the grace period is not over. Billing date the next is dynamically extended every day until the end of the period grace or until the user improves the payment method.
`orderId`	Unique order ID for transactions. This ID according to Google's payment order ID.
`packageName`	Application package where to buy.
`productId`	Item product code. Each item has a product ID, which you must specify on the application product list in the Google Console Play.
`purchaseTime`	When the product is purchased, in milliseconds since the reference time (1 Jan 1970).
`purchaseState`	Order purchase status. This status always returns `0` (bought).
`developerPayload`	A developer-specified string containing additional information about orders. You can specify values for this column when doing Request `getBuyIntent`.
`purchaseToken`	Tokens that uniquely identify purchases for item pairs and users certain.

Method consumePurchase ()

Use a purchase that matches the purchase token provided. This method will cause it to be deleted items of all next response to `getPurchases()` (#getPurchases) and possible repurchase items with the same SKU.

Table 8. Parameter `consumePurchase()`.

Key	Type	Description
`apiVersion`	`int`	AIDL version number Google Play Billing Service that your application uses.
`packageName`	`String`	The name of the application package that calls this method.
`purchaseToken`	`String`	Tokens in JSON purchase information that identifies the purchase to be used.

This method displays `RESULT_OK(0)` from successful usage, and code appropriate response if a failure occurs.

Method getBuyIntentToReplaceSkus()

This method is used to upgrade or downgrade subscription purchases. Method this is similar to `getBuyIntent()` (#getBuyIntent), but this method takes a list with exactly one SKU that has been purchased which will be replaced with SKU being purchased. When a user completes a purchase, Google Play swaps old SKU and gives credit to users for value the time of the subscription that is not used prorata. Google Play applies this credit to a new subscription, and will start charging users for new subscription after credit has been used up.

Note: Instead of using this method, you should use **`getBuyIntentExtraParams()`** (#getBuyIntentExtraParams), that provide additional functions.

This method was added to the AIDL Google Play Billing Service version 5. To verify that this method is reported, send a request AIDL `isBillingSupported`.

Note: You can only use this method for purchasing subscriptions. If parameter **`type`** which is passed on not "**`subs`**", this method will display . In addition, the forwarded SKU may not include SKU for seasonal subscriptions. **`BILLING_RESPONSE_RESULT_DEVELOPER_ERROR`** (#billing-codes)

This method displays the response code integers mapped to key `RESPONSE_CODE`, and `PendingIntent` (https://developer.android.com/reference/android/app/PendingIntent?hl=id) to launch a purchase plot for subscriptions in applications that are mapped to keys `BUY_INTENT`. When receiving `PendingIntent` (https://developer.android.com/reference/android/app/PendingIntent?hl=id), Google Play send response `Intent` along with data for the purchase order. Data displayed in response `Intent` summarized in table 9.

Table 9. Response data from purchase request AIDL Google Play Version 5 Billing Service.

Key	Description
`RESPONSE_CODE`	Value is `0` if the purchase is successful. If the purchase fails, an error code will appear.
`INAPP_PURCHASE_DATA`	String in JSON format that contains details about purchase orders. See table 6 (#purchase-data-table) for description JSON column.

`INAPP_DATA_SIGNATURE`	A string containing the signature of the purchase data signed by the developer with his private key. Data signatures use schemas RSASSA-PKCS1-v1_5.

Method getBuyIntentExtraParams ()

This method starts a purchase request. This method is a variant of the method `getBuyIntent()` (#getBuyIntent), and take parameters `extraParams` additional. This parameter is `Bundle` (https://developer.android.com/reference/android/os/Bundle?hl=id) optional keys and values that affect method operation as shown in table 10.

Table 10. Parameter additional `getBuyIntentExtraParams()`.

Key	Type	Description
`skusToReplace`	`List<String>`	Optional list with one SKU to be upgraded or downgraded by the user. Forward this column if the purchase upgrades or downgrades the subscription existing one. The specified SKU is replaced by the specified SKU is being purchased by a user. Google Play replaces the SKU specified at the start next billing cycle.
`replaceSkusProration`	`boolean`	Determine whether users should be given credit for time unused subscriptions on SKUs that are being upgraded or downgraded. If you assign this column to true, Google Play swaps the old SKU and gives credit to users with the time value of their unused subscription prorata. Google Play applies this credit to new subscriptions, and will start charging users for new subscriptions after credit used up. If you set this column to false, the user does not receive credit for unused subscription time, and repetition date do not change. The default value is true. Ignored if you do not continue **`skusToReplace`**.

`accountId`	`String`	The optional obfuscation string is uniquely attributed to the account user in your application. If you pass this value, Google Play can use it for detecting unnatural activities, such as many devices that make purchases at same account in a short time. This column is similar to **`developerId`** because it represents one users, but note that if you have multiple applications, the same user can have **`accountId`** different for each application, but **`developerId`** must identify uniquely one user in all your applications. Don't use the Google Play Console developer ID or the user's Google ID for this column. Additionally, this column cannot contain user IDs in cleartext. We recommend using a one-way hash to create strings from User ID, and save the hashed string in this column.
`developerId`	`String`	Optional obfuscation strings that are uniquely attributed to user accounts throughout your application. This column is similar to **`accountId`** because it represents one user. However, this column must be the same in all your applications for the same user, while **`accountId`** might be unique for each application, even for the same user. Don't use the Google Play Console developer ID or the user's Google ID for this column. Additionally, this column cannot contain user IDs in cleartext. We recommend using a one-way hash to create strings from User ID, and save the hashed string in this column.
`vr`	**`boolean`**	Determine whether the given intent symbolizes the beginning of the purchase path virtual reality (VR) (https://developers.google.com/vr/android/?hl=id). ★ **Note:** So that these additional parameters affect the application You, you must use AIDL Google Play Version 7 Billing Service, or API newer.

This method is available in AIDL Google Play Billing Service version 6 and newer.

Method getPurchases()

This method returns products that are not currently used, which are owned by users, including items purchased and items obtained by exchanging promo codes. Table 11 lists the response data displayed at `Bundle` (https://developer.android.com/reference/android/os/Bundle?hl=id).

Table 11. Response data from Request `getPurchases`.

Key	Description
`RESPONSE_CODE`	Value is `0` if the request is successful, and error if not.
`INAPP_PURCHASE_ITEM_LIST`	`StringArrayList`which lists the product ID purchase from the application this.
`INAPP_PURCHASE_DATA_LIST`	`StringArrayList`which contains details for purchases from this application. See table 13 for a detailed list of information stored in each item in list.
`INAPP_DATA_SIGNATURE_LIST`	`StringArrayList`which contains a purchase signature from this application.
`INAPP_CONTINUATION_TOKEN`	A string containing a continuation token to retrieve a collection of products in the application next owned by the user. This string is only set by service Google Play if the number of products owned by users is very large. If the continuation token is in response, you

have to make another call to `getPurchases` and forward in your continuation token receive. Call `getPurchases` next displays the purchase other and maybe another continuation token.

Method getPurchaseHistory ()

This method displays the latest purchases made by users for each SKU, even if the purchase has ended, canceled or used. Table 12 lists response data displayed at `Bundle` (https://developer.android.com/reference/android/os/Bundle?hl=id):

Table 12. Response data from request `getPurchaseHistory`.

Key	Description
`RESPONSE_CODE`	Value is `0` if the request is successful, and error if not.
`INAPP_PURCHASE_ITEM_LIST`	`StringArrayList`which lists the product ID purchase from this application.
`INAPP_PURCHASE_DATA_LIST`	`StringArrayList` which contains details for the latest purchase from this application. See table 6 for a list of detailed information stored in each item in the list. `INAPP_PURCHASE_DATA`
`INAPP_DATA_SIGNATURE_LIST`	`StringArrayList`which contains a purchase signature from this application.
`INAPP_CONTINUATION_TOKEN`	A string containing a continuation token to retrieve a collection of products in the application next owned by the user. This string is only set by service Google Play if the number of products owned by users is very large. If the continuation token is in response, you

	have to make another call to **`getPurchases`** and forward in your continuation token receive. Call **`getPurchases`** next displays the purchase other and maybe another continuation token.

Table 13. JSON column description for history purchases displayed by `getPurchaseHistory()`.

Column	Description
`productId`	Item product code. Each item has a product ID, which you must specify on the application product list in the Google Console Play.
`purchaseTime`	When the product is purchased, in milliseconds since the reference time (1 Jan 1970).
`developerPayload`	A developer-specified string containing additional information about orders. You can specify values for this column when doing Request **`getBuyIntent`**.
`purchaseToken`	Tokens that uniquely identify purchases for item pairs and users certain.

Note: Method **`getPurchaseHistory()`** has overhead higher than **`getPurchases()`** (#getPurchases), because requires a call to the Google Play server. We recommend using **`getPurchases()`** (#getPurchases) if you don't really need it user purchase history.

This method is available in AIDL Google Play Billing Service version 6 and newer.

Last updated on 2020-12-22 UTC.