

# BanK データ送信仕様書

2025/08/01 第 1.13 版

株式会社バンダイナムコエンターテインメント

## 更新履歴

更新日	更新者	変更箇所	変更内容	版数
2017/06/23	BNE 汐崎		初版	第 1 版
2017/08/10	BNE 汐崎		Appendix を別ファイルに分離。文言の修正等	第 1.1 版
2017/09/05	BNE 汐崎	7. データ送信テスト	誤記の修正、および Composer がインストール出来ない場合の対応方法を追記	第 1.2 版
2017/12/26	BNE 汐崎	4.5 国内版と海外版との違い	外貨金額の取得方法として getSkuDetails メソッドの参照フィールドを「price」から「price_amount_micros」に変更	第 1.3 版
		7.5.3 通貨コード	説明文言を一部変更	
		7.5.4 プラットフォーム別ユーザー識別子	プラットフォーム別ユーザー識別子 (IDFV/Android ID) の取得方法を追加	
2018/3/7	BNE 汐崎	7.5.3 通貨コード	ユーザー行動履歴とユーザー課金履歴における通貨コード取得方法の違いが分かるように、決済時の通貨コード取得方法を「4.5 国内版と海外版との違い」から移動し並べて記載	第 1.4 版
2018/4/6	BNE 塙	3.2 f001 のデータテーブル定義 7.5.3 通貨コード	f001(ユーザー行動履歴)の通貨コードは、仕様の簡略化を目的として仕様から削除	第 1.5 版
		4.2 f002 のデータテーブル定義 5.2 f003 のデータテーブル定義	直接課金・定期購入についての仕様を追加	
		4.8 定期購入の課金タイミングについて	定期購入の課金タイミングについての記載を追加	
		3.2.5 時刻データ 8. ユーザーデータの引継ぎについて	ユーザーデータの引継ぎ対応についての記載を追加	
		8. ユーザーデータの引継ぎ	データ引継ぎの対応内容を変更	
2018/11/05	BNE 塙			第 1.6 版
2021/06/14	BNE 汐崎	6. fluentd のインストールと設定 7. データ送信テスト	データ受信方法を fluentd 方式から S3 方式へ変更、左記 2 つの大項目を仕様書から削除し、ドキュメント全体の記述を修正	第 1.7 版
		3.2 f001 のデータテーブル定義	sim_id, mac_address の記述を削除し、BNIDv3 から追加された user_id を追記	
2021/10/8	BNE 高橋	3.2 f001 のデータテーブル定義	BNIDv2 及び BNIDv3 での設定項目の必須記載を修正	第 1.8 版
		7.1 引継ぎ方法について	プラットフォームを跨ぐ場合の app_user_id 及び app_system_prefix の設定内容変更を修正	
2023/12/01	SCN 佐川	3.2 f001 のデータテーブル定義 3.2.4 プラットフォームを特定する情報 4.2.5 プラットフォームを特定する情報	Steam、PS4/PS5、Xbox SeriesX S/Xbox One、Nintendo Switch の場合を追記	第 1.9 版
		6.2 国コード 6.3 通貨コード 6.4 プラットフォーム別ユーザー識別子	スーパーマルチタイトルに関する注記を追記	
		8. スーパーマルチタイトルについて	スーパーマルチタイトルについて追記	
		3.2.4 プラットフォームを特定する情報 4.2.5 プラットフォーム	Platform_id = win を追加	
2024/01/12	SCN 佐川			第 1.10 版

		ムを特定する情報		
2024/02/01	AIQ 佐川	4. 2. 5 プラットフォームを特定する情報	platform_id にアソビストア (ios_asb、and_asb) を追加	
2024/02/29	AIQ 佐川	9. アソビストアについて	アソビストアについて追記	
2024/04/05	AIQ 佐川	4. 2. 5 プラットフォームを特定する情報	platform_id にアソビストア (dmm_asb) を追加	
2024/04/16	AIQ 佐川	4. 2. 5 プラットフォームを特定する情報	dmm_asb を削除	第 1.11 版
2024/06/13	AIQ 佐川	4. 2. 5 プラットフォームを特定する情報	platform_id にアソビストア (asb) を追加	
2024/06/13	AIQ 佐川	9. アソビストアについて	アソビストアの platform_id 定義について追記	
2024/11/27	AIQ 佐川	全般	アソビストアを WebStore/アソビストアに変更	第 1.12 版
2025/05/23	AIQ 佐川	5. f003 データ送信の注意点	窓口 ML を変更	
2025/08/01	IRT 田中	5. 2 f003 のデータテーブル定義	2 種目の有償通貨に関する項目の記述を追加	第 1.13 版
2025/08/01	AIQ 佐川	9. 1 platform_id について	マルチログインに関する注記を追加	第 1.13 版
2025/08/01	AIQ 佐川	10. 2 種類の有償通貨について	2 種類の有償通貨について追加	第 1.13 版

## 目次

1	ドキュメント	7
1.1	ドキュメントの目的	7
1.2	対象者	7
2	送信仕様	7
2.1	送信仕様概要	7
2.2	送信データの種類	7
①	f001 データ（ユーザー行動履歴）	7
②	f002 データ（ユーザー課金履歴）	7
③	f003 データ（サービス売上情報）	8
2.3	BanK サーバ構成	9
2.3.1	検証環境と本番環境のサーバ構成	9
2.3.2	アップロードする S3 のパス	10
2.3.3	ファイルフォーマット	10
2.3.4	データのアップロード間隔	11
2.4	認証	11
2.4.1	S3 バケットへの接続	11
2.4.2	「アプリケーション ID(サービス ID)」「クライアント ID」「クライアントシークレット」	11
3	ユーザー行動履歴に関するデータ送信 (f001)	12
3.1	f001 概要	12
3.2	f001 のデータテーブル定義	12
詳細については、3.2.4 プラットフォームを特定する情報を参照	13	
3.2.1	データ送信元を特定する情報	13
3.2.2	イベント ID とイベント発生日時 (event_id と event_time)	13
3.2.3	ユーザーを特定する情報	13
3.2.4	プラットフォームを特定する情報	14
3.2.5	時刻データ	14
3.2.6	国コードと通貨コード	15
3.2.7	広告 ID	15
3.3	BanK における集計項目との関連性	15
3.4	ユーザーの新規作成	17
3.5	ログインの通知	17
3.6	リセットマラソン（リセマラ）への対応	17
4	ユーザー課金履歴に関するデータ送信 (f002)	19
4.1	f002 概要	19

4.2 f002 のデータテーブル定義 .....	19
4.2.1 データ送信元を特定する情報 .....	20
4.2.2 ユーザーを特定する情報 .....	20
4.2.3 購入または消費のデータ .....	20
4.2.4 時刻データ .....	20
4.2.5 プラットフォームを特定する情報 .....	21
4.2.6 国コードと通貨コード .....	21
4.3 有償通貨における課金のタイミング .....	21
4.3.1 コインや魔法石などの有償通貨を購入したとき .....	21
4.3.2 コインや魔法石などを消費したとき .....	21
4.3.3 無償でコインや魔法石などを提供する場合 .....	22
4.3.4 購入時の単価が違う場合の消費金額 .....	22
4.4 BanK における集計項目との関連性 .....	23
4.5 国内版と海外版との違い .....	24
4.6 コインや魔法石などの有償通貨を購入したときの設定例 .....	24
4.7 コインや魔法石などの有償通貨を消費したときの設定例 .....	24
4.8 定期購入の課金タイミングについて .....	24
5 サービス売上情報に関するデータ送信 (f003) .....	26
5.1 f003 概要 .....	26
5.2 f003 のデータテーブル定義 .....	27
5.3 集計対象 .....	28
5.4 データ送信 .....	30
5.4.1 日次送信 .....	30
5.4.2 月次送信 .....	30
5.5 f003 データ送信の注意点 .....	31
6 データに関する共通の注意 .....	32
6.1 日時 .....	32
6.2 国コード .....	32
6.3 通貨コード .....	33
6.4 プラットフォーム別ユーザー識別子 (IDFV/Android ID) .....	34
7 ユーザーデータの引継ぎ .....	35
7.1 引継ぎ方法について .....	35
7.1.1 同一プラットフォームの場合 .....	35

---

7.1.2 プラットフォームを跨ぐ場合 .....	35
7.1.3 引継ぎ時の課金ログについて .....	35
<b>8     スーパーマルチタイトルについて .....</b>	<b>36</b>
8.1 国内・海外での扱いについて .....	36
8.2 国コードの取得方法について .....	36
<b>9     WebStore/アソビストアについて .....</b>	<b>37</b>
9.1 platform_id について .....	37
9.2 f001 で送信するデータについて .....	37
9.3 f002、f003 で送信するデータについて .....	37
<b>10    2 種類の有償通貨について .....</b>	<b>39</b>

## 1 ドキュメント

本ドキュメントは、BanK システムに対してデータを送信する際の仕様を定義したものです。

### 1.1 ドキュメントの目的

本ドキュメントは、正しい情報を提供することで、BanK システムに正しい形式・正しい手順でデータを送信して頂き、正しい集計結果となることを目的とします。

### 1.2 対象者

本ドキュメントの対象者は、BanK システムにデータを送信するサーバプログラム開発者です。

## 2 送信仕様

本項では、送信仕様を説明します。

### 2.1 送信仕様概要

BanK システムは、S3 バケットからデータを取り込みます。

ゲームシステムでは、本仕様で規定する書式の JSON 形式データを作り、それをひとつの gzip 形式ファイル (.gz) にまとめて、S3 バケットに配置する処理を実装してください。

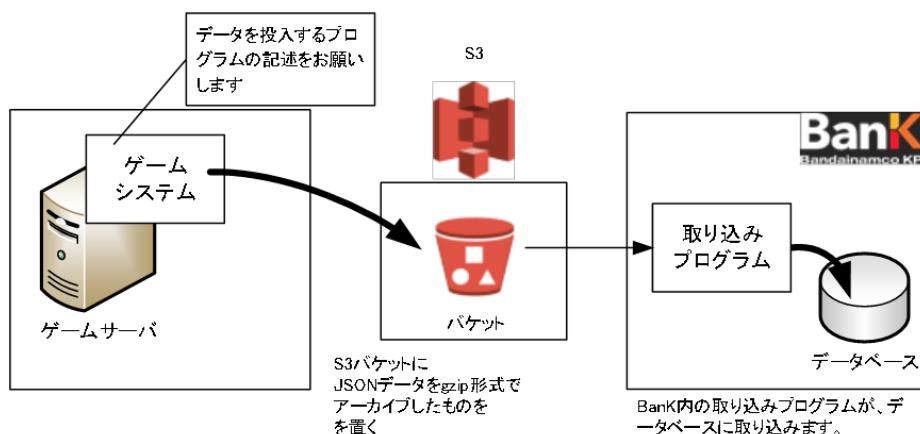


図 2-1 S3 バケットに JSON データを配置すると、BanK システムに取り込まれる

### 2.2 送信データの種類

ゲームシステムから BanK に送信するデータは、次の 3 種類があり、「f001」「f002」「f003」と呼びます。

3 種のデータのうち「f001」と「f002」は、ユーザー単位のアクション（新規登録、課金など）データ、「f003」は売上計上、資金決済法への対応を目的とした、ゲームサービス単位の有償通貨の購入、消費データです。

#### ①f001 データ（ユーザー行動履歴）

新規ユーザーの登録や HAU (Hourly Active Users)、DAU (Daily Active Users)、MAU (Monthly Active Users) などのアクティブラグユーザーを集計するための基データです。

このデータの詳細は、「3. ユーザーアクションに関するデータ送信 (f001)」で説明します。

#### ②f002 データ（ユーザー課金履歴）

コイン等の有償通貨の購入や消費、直接課金による購入や消費を集計するデータです。

このデータの詳細は、「4. 課金アクションに関するデータ送信 (f002)」で説明します。

### ③f003 データ（サービス売上情報）

日次および月次の売上集計データです。有償通貨の日次および月次売上集計データです。

国内版のアプリでのみ利用可能です。

このデータの詳細は、「5. 売上情報に関するデータ送信（f003）」で説明します。

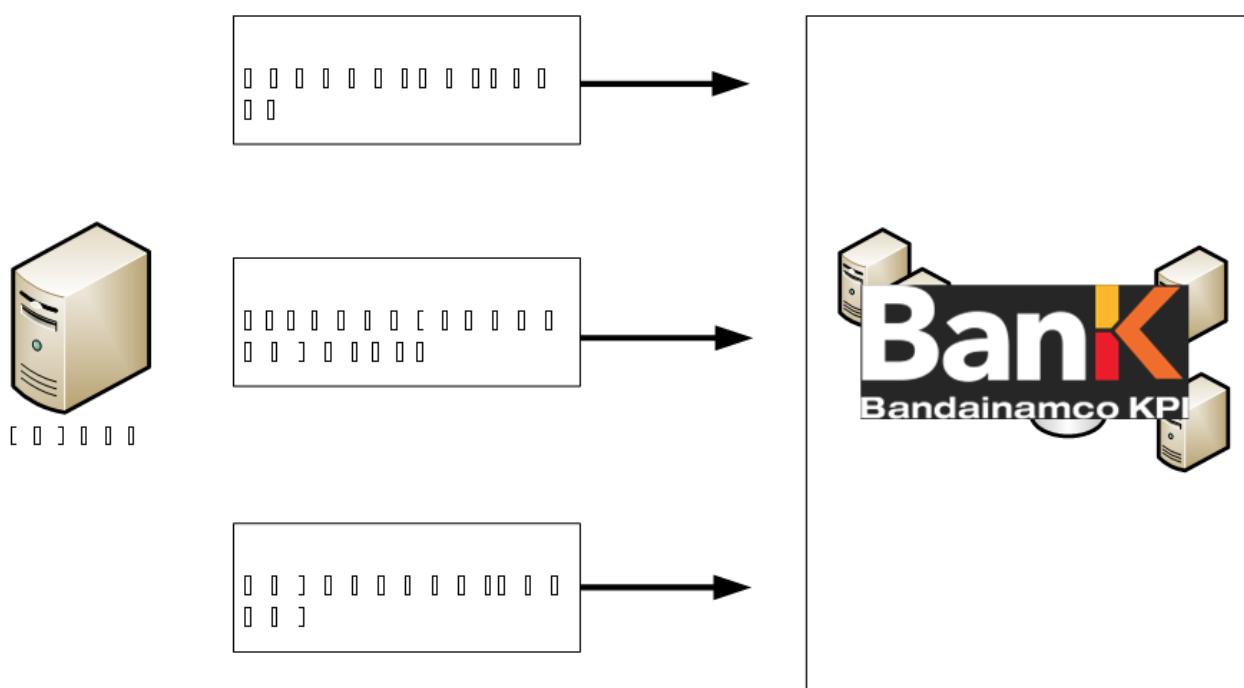


図 2-2 BanK に送信すべき 3 種類のデータ

## 2.3 BanK サーバ構成

BanK システムには、「本番環境 (prd)」と「検証環境 (stg)」の 2 つが提供され、それぞれ、図 2-3 のように構成されています。

それぞれ別のデータ転送用バケットが用意されています。

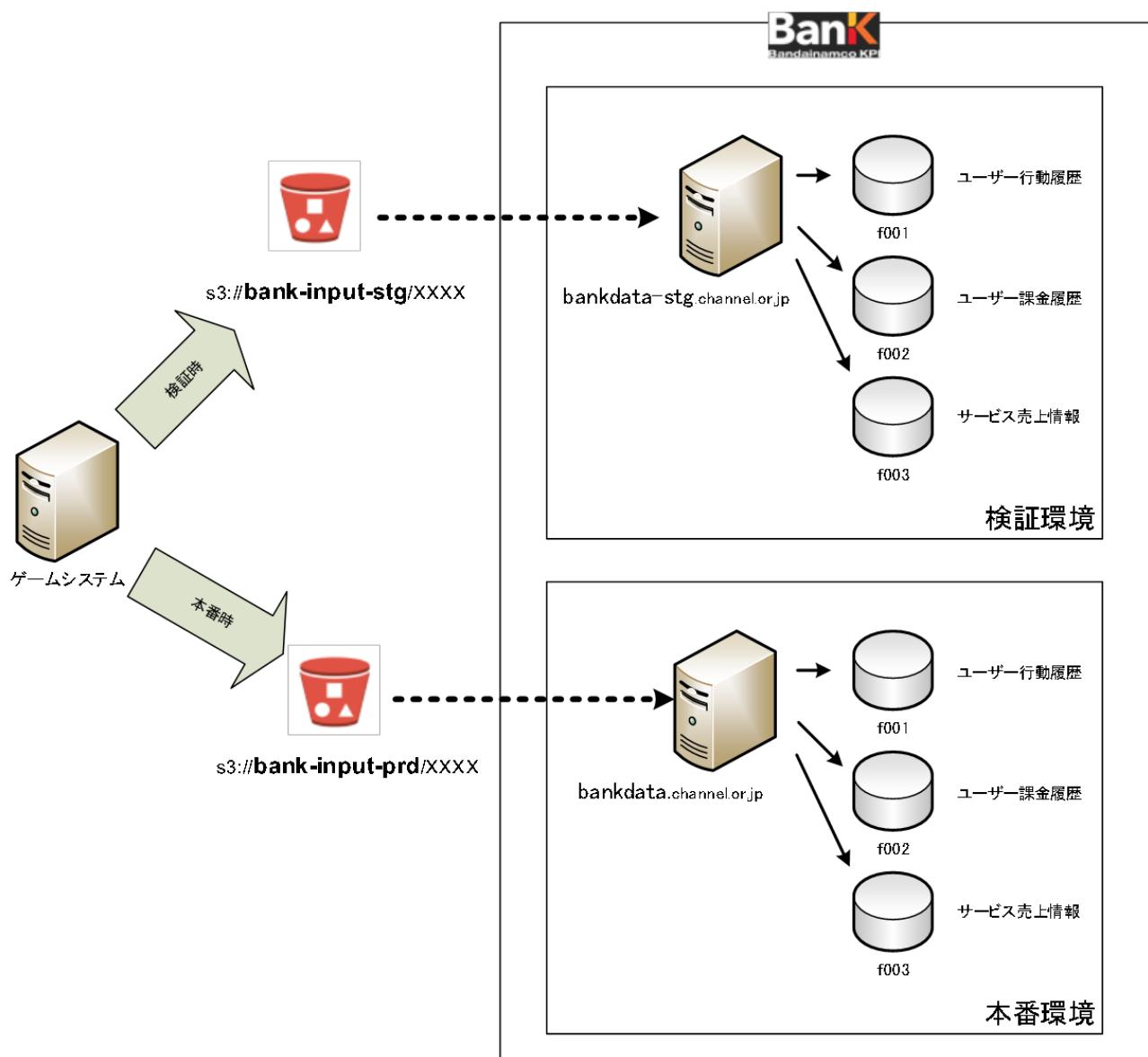


図 2-3 BanK システムの送信仕様

### 2.3.1 検証環境と本番環境のサーバ構成

BanK システムには、「検証環境」と「本番環境」の 2 系統があります。

#### 【検証環境】

S3 バケット	<code>s3://bank-input-stg/</code>
---------	-----------------------------------

#### 【本番環境】

S3 バケット	<code>s3://bank-input-prd/</code>
---------	-----------------------------------

### 2.3.2 アップロードする S3 のパス

アップロードするパスは、「sa」「年」「月」「日」「時」「フォーマット (f001、f002、f003)」から構成される、下記のパスとします。「年月日時」は UTC 時刻とします。

※送信データの中身は JST としていますが、アップロードパスは UTC で作成ください。

#### 【検証環境】

```
s3://bank-input-stg/data/{app_id}/{YYYY}/{MM}/{DD}/{HH}/{format_id}/ファイル名
```

#### 【本番環境】

```
s3://bank-input-prd/data/{app_id}/{YYYY}/{MM}/{DD}/{HH}/{format_id}/ファイル名
```

{app\_id} = アプリケーション ID

{YYYY} = 年、{MM} = 月、{DD} = 日、{HH} = 時

たとえばアプリケーション ID が「12345」のサービスにおいて、2021 年 1 月 3 日 17 時にユーザー行動履歴 (f001) をアップロードする場合は、下記のパスに配置します。下記の例では、ファイル名を「server1-prd.2017040417\_0.gz」としていますが、ファイル名は任意です。

```
s3://bank-input-prd/data/12345/2021/01/03/17/f001/server1-prd.2021010317_0.gz
```

### 2.3.3 ファイルフォーマット

転送するファイルは、1 行を 1 レコードとして構成します。1 レコードは、タブ文字で区切られた「①gentime」「②fluentd\_tag」「③データ」の 3 つの項目で構成されます。

gentime (TAB) fluentd_tag (TAB) データ
-------------------------------------

データ例を以下に示します。

#### 【データ例】

2021-02-28T15:15:00Z	bng.kpi.gs.prd.12345.f001	{"client_id": "xxxx", "client_secret": "xxxx", ...}
2021-02-28T15:15:01Z	bng.kpi.gs.prd.12345.f001	{"client_id": "xxxx", "client_secret": "xxxx", ...}
2021-02-28T15:15:01Z	bng.kpi.gs.prd.12345.f001	{"client_id": "xxxx", "client_secret": "xxxx", ...}
2021-02-28T15:15:02Z	bng.kpi.gs.prd.12345.f001	{"client_id": "xxxx", "client_secret": "xxxx", ...}

#### ①gentime

レコードの出力日時を示します。フォーマットは、次の通りです。

年-月-日 T 時:分:秒 Z 例) 2021-02-28T15:15:00Z
--

この日時は、あくまでもレコード出力日時であり、イベントの発生日時ではありません。

※リアルタイムにレコードを出力する場合は、イベントの発生日時と近くなるはずです。

※バッチ方式で翌日以降にまとめて出力する場合は、バッチが実際に実行される日時（例えば翌日以降のバッチ開始日時）です。

#### ②fluentd\_tag

レコードの属性（「検証環境」か「本番環境」か、および、「f001」「f002」「f003」のいずれか）を示す項目です。フォーマットは、以下の通りです。

**【検証環境の場合】**

bng.kpi.gs.stg. アプリケーション ID. f001  
bng.kpi.gs.stg. アプリケーション ID. f002  
bng.kpi.gs.stg. アプリケーション ID. f003

**【本番環境の場合】**

bng.kpi.gs.prd. アプリケーション ID. f001  
bng.kpi.gs.prd. アプリケーション ID. f002  
bng.kpi.gs.prd. アプリケーション ID. f003

**③データ**

本ドキュメントで解説している、実際に送受信するデータです。JSON 形式文字列として記述します。

JSON 形式としてパースできない行は、無視されます（たとえば行中に TSV 形式や CSV 形式のデータが含まれていてもスキップされます）

#### 2.3.4 データのアップロード間隔

データ送信は S3 バケットへのデータアップロードで行いますが、よりリアルタイムにグラフを確認出来るよう日次のデータアップロードではなく、リアルタイムや 1 時間単位でデータを送信ください。

### 2.4 認証

#### 2.4.1 S3 バケットへの接続

S3 バケットへの接続は、IAM 認証を用います。接続に必要な情報については、担当プロデューサ、もしくは、BanK 担当にご確認ください。

#### 2.4.2 「アプリケーション ID(サービス ID)」「クライアント ID」「クライアントシークレット」

BNE ではゲームタイトルごとに「アプリケーション ID (app\_id)」「クライアント ID (client\_id)」と「クライアントシークレット (client\_secret)」を発行します。

JSON データの各データ列に「app\_id」「client\_id」「client\_secret」の属性値として、これらを含めてください。合致しない場合、BanK システムはそのデータを拒否します。

「アプリケーション ID(サービス ID)」「クライアント ID」「クライアントシークレット」は、担当プロデューサが以下「BanK Moderator」のタイトル情報で確認出来ます。

※セキュリティ上、BNE 担当者以外は「クライアント ID」「クライアントシークレット」は表示されません。

【BanK Moderator】 <https://bankadm.channel.or.jp/moderator/>

### 3 ユーザー行動履歴に関するデータ送信 (f001)

この項では、新規ユーザーの記録やユーザーがプレイを継続している旨の通知など、ユーザーアクションに対して BanK に送信するデータについて説明します。

#### 3.1 f001 概要

f001 は、新規登録者数や累計登録者数、アクティブユーザー、アクティブ率、継続率などを集計する際に使われる基データとなります。

f001 の役割は、主に 2 つあります。

##### ①新規ユーザーを記録する

新規ユーザーの加入を BanK システムに送信します。これにより、新規登録者数と累計登録者数を計測できます。

##### ②ユーザーがプレイを継続していることを記録する

ユーザーがプレイを継続していることを BanK システムに送信します。これにより、アクティブユーザー数やアクティブ率、継続率などを計測できます。

BanK システムにおけるアクティブユーザー計測の最小単位は「HAU (1 時間当たりのアクティブユーザー数)」です。

※1 時間の間であれば、定期間隔である必要はありません。たとえばユーザー A がプレイしている通知を 8 時 40 分に送信して、9 時台も引き続き遊んでいるようであれば、9:00~9:59 の間のいずれかのタイミングで、ユーザー A がプレイしている旨を 1 回送信すれば大丈夫です。

#### 3.2 f001 のデータテーブル定義

f001 のデータテーブル定義は、表 3-1 の通りです。本項では、必須となる重要な部分のみ説明します。すべての項目は、Appendix に示します。

表 3-1 f001 のデータテーブル定義

列名	必須 (国内)	必須 (海外)	解説
version			バージョン番号。現在は「1.0」固定。省略可
app_id	○	○	アプリケーション ID (SVID)。BNE が発行。“1241”など
client_id	○	○	クライアント ID。BNE が発行
client_secret	○	○	クライアントシークレット。BNE が発行
event_id	○	○	イベント ID。表 3-2 に示す「100」「200」「300」のいずれか
event_time	○	○	イベントが発生した日時。JST。YYYY-MM-DD HH:MI:SS の書式。「2016-01-20 07:50:32」など
app_user_id	○	○	ユーザー ID。アプリにおけるユーザーを区別する任意の ID。最大 128 文字以内の文字列
app_system_prefix			app_user_id と併せてユーザーをユニークに識別するための補助識別子。 app_user_id のみでユニーク性が保証されている場合は不要。最大 10 文字
platform_user_id	○	○	プラットフォーム上のユーザーの識別番号。 各プラットフォームのユーザー識別番号は、iOS:IDFV、Android:ANDROID_ID、Steam:SteamID、PS4/PS5:アカウント ID、Xbox SeriesX S/XboxOne:XUID、Nintendo switch:ネットワークサービスアカウントの内部 ID を設定してください。 「6.4 プラットフォーム別ユーザー識別子 (IDFV/Android ID)」を参照 「8.4 プラットフォーム別ユーザー識別子について」を参照

user_id	※BNID 連携時は必須	※BNID 連携時は必須	バンダイナムコ ID の user ID の値。バンダイナムコ ID のユーザー情報と結びつき、年代や性別のデータを表示できる。“001”など 注) タイトルが V3 サービスの場合に必須設定する
person_id	※BNID 連携時は必須	※BNID 連携時は必須	バンダイナムコ ID の PersonID の値。バンダイナムコ ID のユーザー情報と結びつき、年代や性別のデータを表示できる。“pi00.EhyONJNFP2B5Ij_eZBHT_w--”など 注) タイトルが V2 サービスの場合に必須設定する
mbid	※BNID 連携時は必須	※BNID 連携時は必須	バンダイナムコ ID の MBID。“107725418”など 注) タイトルが V2 サービスの場合に必須設定する
ktid	※BNID 連携時は必須	※BNID 連携時は必須	バンダイナムコ ID の「かんたん ID」。 “hz8udrtocchqyaqp4xrm8bwbolifypnsl5w6mn5kddz1vf”など 注) タイトルが V2 サービスの場合に任意設定する
platform_id	○	○	BNE 運用チームが定めたプラットフォーム ID。 詳細については、3.2.4 プラットフォームを特定する情報を参照
platform_version			プラットフォームの OS バージョン。Android なら「2.3」「4.2」など。iOS なら「7.1」「8.0」など。最大 10 文字まで
user_agent			ブラウザの User-Agent と同じもの。最大 350 文字まで。 “Mozilla/5.0 (Linux; U; Android 4.1.2; ja-jp; S0-04E Build/10.1.1.D.2.31) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30”など
created_time	○	○	ユーザーを新規作成した日時。継続率集計に利用。YYYY-MM-DD HH:MI:SS の書式。「2016-01-20 07:50:32」など
last_access_time			該当ユーザーの最終アクセス時間。JST。YYYY-MM-DD HH:MI:SS の書式。「2016-01-20 07:50:32」など
old_last_access_time			該当ユーザーの前回の最終アクセス時間。JST。YYYY-MM-DD HH:MI:SS の書式。「2016-01-20 07:50:32」など
country_code		○	国コード。国内配信アプリは「JP」固定、海外配信アプリは指定必須。「7.5.2 国コード」を参照
ad_id	○	○	広告識別子。iOS の場合は「IDFA」、Android の場合は「GAID」を設定してください 広告 ID が取得できない場合や、コンシューマタイトルにおいては、下記の桁数のコードを入れて送信してください。 00000000-0000-0000-0000-000000000000 ※8 桁-4 桁-4 桁-4 桁-12 桁

### 3.2.1 データ送信元を特定する情報

「app\_id」はゲームアプリ（ゲームタイトル）ごとに発行されるユニークキーです。また「client\_id」「client\_secret」は、「app\_id」と結びつけ、BanK システムへ送信されたデータを受信するときの認証に利用されます。詳しくは「7-4 データ送信のサンプルプログラム」を参照してください。

### 3.2.2 イベント ID とイベント発生日時 (event\_id と event\_time)

event\_id は、ユーザーイベントの種類を示します。「100」「200」「300」のいずれかです（表 3-2）。event\_time には、そのイベントが発生した日時（JST）を設定します。

表 3-2 イベント ID

値	意味
100	ユーザー登録
200	ユーザー無効（リセマラ対策用）
300	プレイ継続中（ログイン）

### 3.2.3 ユーザーを特定する情報

BanK でユーザーを集計するときは、次の項目で、ユーザーを区別します。

## (a) app\_user\_id と app\_system\_prefix

app\_user\_id は、アプリでユーザーを識別する任意の番号です。app\_system\_prefix は、管理上などの理由で app\_user\_id の前に付けたい接頭辞で、必要なければ省略できます。

「app\_user\_id」と「app\_system\_prefix」は、BanK システム上で、「同一ユーザーかどうか」を判定する際に使われる、重要なデータです。

※リセマラユーザーと判断して、「ユーザー無効 (event\_id=200)」を送信した場合、そのユーザー (app\_user\_id) は削除されたものとして扱われます。そのため、再度アプリをインストールしてプレイを開始したユーザー（リセマラユーザー）は、別ユーザーとして判別できるよう、app\_user\_id を新たに発行して頂くか、app\_system\_prefix を付与して頂く必要があります。

## (b) OS や端末の区別 (platform\_user\_id)

platform\_user\_id は、プラットフォーム上でユーザーを識別する番号です。iOS の場合は「IDFV」を、Android の場合は「ANDROID\_ID」を設定してください。

## (c) バンダイナムコ ID との連携

バンダイナムコ ID と連携したいときは、下記の値を設定してください。バンダイナムコ ID のユーザー情報と結びつき、ユーザーの年代や性別のデータを表示できます。

user\_id バンダイナムコ ID の「userID」の値

person\_id バンダイナムコ ID の「PersonID」の値

mbid バンダイナムコ ID の「MBID」の値

## 3.2.4 プラットフォームを特定する情報

platform\_id は、プラットフォームを特定する情報を示します（表 3-2）。

また BanK 運用チーム（BNE）が定めた値を指定することもあります。

表 3-2 プラットフォーム ID 定義

プラットフォーム	設定値
iOS	ios
android, androidPC(GooglePlay Games)	android
DMM	dmm
Steam	steam
PS4/PS5	ps4, ps5
Xbox SeriesX S /Xbox One	xsx
Nintendo Switch	nsw
Windows	win

※f002、f003 では上記のほかに「アソビストア」がプラットフォームとしてありますが、「アソビストア」は、購入のみ可能な為、f001 では取り扱いません。また、受け付けませんのでご注意ください。

platform\_version は、プラットフォームのバージョンです。iOS なら「7.1」や「8.2」など、Android なら「2.3」「4.2」など、最大 10 文字までの任意のバージョン名を設定できます。

user-agent は、ブラウザの User-Agent と同じものです。最大 350 文字まで任意の文字列を設定できます。

## 3.2.5 時刻データ

created\_time は、ユーザーの新規登録日時を識別する値です。ユーザーの継続利用集計に利用します。次のように設定してください。

## ① event\_id 「100」のとき

ユーザーが新規登録した日時、を設定してください。これは event\_time と同じ値です。

※ユーザーデータ引継ぎ時も新規登録とし、引継ぎ後の日時を設定してください。

## ②event\_id 「200」 および 「300」 のとき

該当ユーザーを event\_id 「100」 で新規登録したときに設定した created\_time と同じ値を指定してください。

last\_access\_time は該当ユーザーの最終アクセス時間、old\_last\_access\_time は last\_access\_time よりひとつ前の最終アクセス時間を、それぞれ示します。必要なければ、設定を省略してかまいません。

すべての日時は「JST」です（「6 データに関する共通の注意」を参照）。

※継続率は created\_time を基に集計しています。

### 3.2.6 国コードと通貨コード

国コードを country\_code、通貨コードを currency\_code に設定します。海外配信アプリでは必ず指定する必要があります。

書式は「6 データに関する共通の注意」を参照してください。国内配信アプリは、それぞれ「JP」、「JPY」に固定されます。

### 3.2.7 広告 ID

広告 ID を ad\_id に設定します。iOS の場合は「Identification For Advertisers (IDFA)」、Android の場合は「Google Advertising ID (GAID)」を設定してください。

## 3.3 BanK における集計項目との関連性

f001 で送信したデータは、BanK システムの下記の項目に関連します。

### ◆アクティブユーザー数

時次：年月日時におけるアクティブなユニークユーザー数 (event\_id : 100、300)

日次：年月日におけるアクティブなユニークユーザー数 (event\_id : 100、300) — 同日のリセマラユーザー数 (event\_id : 200)

月次：年月におけるアクティブなユニークユーザー数 (event\_id : 100、300) — 同月のリセマラユーザー数 (event\_id : 200)

### ◆アクティブユーザー数（平均）

アクティブユーザー数のデイリー平均（※）

※日次：月初からの日次アクティブユーザー数合計を日数で割ったもの。月次は日次アクティブユーザー数の平均。

### ◆登録ユーザー数（新規）

時次：年月日時における新規ユニークユーザー数 (event\_id : 100)

日次：年月日における新規ユニークユーザー数 (event\_id : 100) — 同日のリセマラユーザー数 (event\_id : 200)

月次：年月における新規ユニークユーザー数 (event\_id : 100) — 同月のリセマラユーザー数 (event\_id : 200)

### ◆登録ユーザー数（累計）※リセマラ含む

時次：年月日時における新規ユニークユーザー数 (event\_id : 100) 累計

日次：年月日における新規ユニークユーザー数 (event\_id : 100) 累計

月次：年月における新規ユニークユーザー数 (event\_id : 100) 累計

### ◆登録ユーザー数（累計）※リセマラ除外

時次：年月日時までの新規ユニークユーザー数 (event\_id : 100) 累計

日次：年月日までの新規ユニークユーザー数 (event\_id : 100) 累計 — 同日までのリセマラユーザー数 (event\_id : 200) 累計

月次：年月までの新規ユニークユーザー数 (event\_id : 100) 累計 — 同月までリセマラユーザー数 (event\_id : 200) 累計

## ◆アクティブ率

アクティブユーザー数／登録ユーザー数（累計）※リセマラ除外

## ◆アクティブ率（平均）

アクティブ率のデイリー平均（※）

※日次：月初からの日次アクティブ率合計を日数で割ったもの。月次は日次アクティブ率の平均。

## ◆ARPU

課金額 ÷ アクティブユーザー数

## ◆ARPU（平均）

指定期間が月別の場合はデイリー値の平均。指定期間が日別の場合は、毎月 1 日～該当日までの平均値

## ◆継続率(特定時点)N 日後 (N=1, 3, 7, 14, 30, 60)

M 日に新規登録したユーザーが N 日後にアクティブになっている割合

N 日後のアクティブなユニークユーザー数 ÷ M 日の登録ユーザー数（新規）

## ◆継続率(ログイン割合)N 日後 (N=1, 3, 7, 14, 30, 60)

M 日に新規登録したユーザーが N 日後までアクティブになった頻度

(N 日後までにアクティブになったユニークユーザー数日数 ÷ N) ÷ M 日の登録ユーザー数（新規）

## ◆5 日間連続性

直近 5 日間におけるユーザーのアクティブ状況

- ・1 日アクティブ（新規登録）
- ・1 日アクティブ（新規登録以外）
- ・2 日アクティブ
- ・3 日アクティブ
- ・4 日アクティブ
- ・5 日アクティブ

## ◆30 日間連続性 × 1 日頻度

直近 30 日間におけるユーザーのアクティブ状況

- ・1 日アクティブ
- ・2~14 日低頻度アクティブ
- ・2~14 日中頻度アクティブ
- ・2~14 日高頻度アクティブ
- ・15~28 日低頻度アクティブ
- ・15~28 日中頻度アクティブ
- ・15~28 日高頻度アクティブ
- ・29~日低頻度アクティブ
- ・29~日中頻度アクティブ
- ・29~日高頻度アクティブ

※低頻度： 1 日平均 HAU≤2

※中頻度： 2<1 日平均 HAU<6

※高頻度： 6≤1 日平均 HAU

## ◆年代別性別

mbid もしくは person\_id で紐づく BNID の年代別・性別ユーザー数

※自ゲームで mbid／person\_id を設定していない場合でも、他ゲームが mbid／person\_id を

設定していて platform\_user\_id により同一端末（＝同一利用者）と判定された場合、

BNID の年代別・性別ユーザー数を表示します

◆アプリ利用者の性別

mbid もしくは person\_id で紐づく BNID の性別ユーザー数  
 ※自ゲームで mbid／person\_id を設定していない場合でも、他ゲームが mbid／person\_id を  
 設定していて platform\_user\_id により同一端末（＝同一利用者）と判定された場合、  
 BNID の性別ユーザー数を表示します

◆重複利用状況 全ユーザー

自ゲーム／他ゲームの全期間中で同一の platform\_user\_id を持つユーザー数

◆重複利用状況 アクティブユーザー1

自ゲーム／他ゲームの直近 30 日間で同一の platform\_user\_id を持つユーザー数

◆重複利用状況 アクティブユーザー2

自ゲームの直近 30 日間／他ゲームの全期間で同一の platform\_user\_id を持つユーザー数

### 3.4 ユーザーの新規作成

ユーザーが新規作成されたときには、event\_id に「100」を設定して、BanK に向けてデータを送信します。  
 データの例は、リスト 3-1 の通りです。

新規ユーザー登録は、ユーザー記録作成時にサーバ時刻を event\_time と created\_time に設定して送る  
 ようにしてください。

リスト 3-1 ユーザーの新規作成

```
{"app_id": "アプリケーション ID", "client_id": "クライアント ID", "client_secret": "クライアントシークレット",
"app_user_id": 1234567890, "app_system_prefix": "", "event_id": 100, "event_time": "2021-02-04 12:13:00",
"platform_id": "android", "platform_user_id": "8900000000000006156351606050141232",
"created_time": "2021-02-04 12:13:00", "ad_id": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"}
```

※新規アプリケーションユーザーIDとして「1234567890」を割り当てた場合。

### 3.5 ログインの通知

ユーザーがプレイ継続中であるときは、最低 1 時間に 1 回、イベント ID に「300」を設定したデータを、BanK に向けてデータを送信してください。たとえば、リスト 3-2 のようにします。

このデータは、HAU (Hourly Active Users) を計測するのに使われます。

送信するタイミングとしては、「ゲームをアクティブにしてプレイし始めたとき」「ゲームが再アクティブになったとき」などが考えられます。

BanK のアクティブユーザー表示の最小単位は HAU となります。ですので、ログイン情報は 1 時間単位で送信してください。

※1 時間内に複数回ログインした場合も、1 回分のログイン情報だけ送信して頂けますと BanK 側の負荷が軽くなります。

リスト 3-2 プレイ継続中の通知の例

```
{"app_id": "アプリケーション ID", "client_id": "クライアント ID", "client_secret": "クライアントシークレット",
"app_user_id": 1234567890, "app_system_prefix": "", "event_id": 300, "event_time": "2021-02-05 09:15:00",
"platform_id": "android", "platform_user_id": "8900000000000006156351606050141232",
"created_time": "2021-02-04 12:13:00"}
```

### 3.6 リセットマラソン（リセマラ）への対応

ユーザーがアプリの初回利用で希望のアイテムが出なかったときに、希望アイテムが出るまでインストール／再インストールを繰り返す「リセットマラソン（リセマラ）」が行われると、実際には利用していないユーザーIDが BanK に登録されていき、データがにぎってしまいます。

リセマラユーザーをカウントしないようにするために、以下 2 つのどちらかの対応を実施願います。

①ゲームサーバ側でリセマラユーザーを除外（1 日のユニークユーザーを確定）してから、BanK へデータを送信する

②BanK へ新規登録データを送信後、同一デバイス ID で新規登録があった場合（リセマラ）、同日最新以外の登録はリセマラユーザーと見なし BanK に「evnet\_id = 200」を送信する。

※ユーザー削除「evnet\_id = 200」の送信は、後日送信でも問題ありません。

なお、無効にした後の復帰機能はありません。「evnet\_id = 200」を送信したユーザーは、同じユーザーID／同じ event\_time（年月日時）で「evnet\_id = 100」で登録しても無視されたままとなるので注意してください。

※リセマラユーザーと判断して、「ユーザー無効（event\_id = 200）」を送信した場合、そのユーザー（app\_user\_id）は削除されたものとして扱われます。そのため、再度アプリをインストールしてプレイを開始したユーザー（リセマラユーザー）は、別ユーザーとして判別できるよう、app\_user\_id を新たに発行して頂くか、app\_system\_prefix を付与して頂く必要があります。

### リスト 3-3 ユーザーの無効の例

```
{"app_id": "アプリケーション ID", "client_id": "クライアント ID", "client_secret": "クライアントシークレット",
"app_user_id": 1234567890, "app_system_prefix": "", "event_id": 200, "event_time": "2021-02-05 16:32:00",
"platform_id": "android", "platform_user_id": "8900000000000006156351606050141232",
"created_time": "2021-02-04 12:13:00"]
```

## 4 ユーザー課金履歴に関するデータ送信 (f002)

この項では、ユーザーがコイン等の有償通貨を購入・消費したときに、BanK に送信しなければならないデータについて説明します。

### 4.1 f002 概要

f002 は、ユーザーごとの課金データです。コイン等の有償通貨の購入や消費に加え、アイテムの直接購入や定期購入が発生したときにも用いるデータです。購入・消費金額や課金 UU、課金率、ARPU や ARPPU の基データとなります。

※f003 も同じ課金データですが、f002 はユーザー単位の課金情報という点が異なります。

### 4.2 f002 のデータテーブル定義

f002 のデータテーブル定義は、表 4-1 の通りです。本項では、必須となる重要な部分のみ説明します。すべての項目は Appendix に示します。

表 4-1 f002 のデータテーブル定義

列名	必須 (国内)	必須 (海外)	解説
version			バージョン番号。現在は「1.0」固定。省略可
app_id	○	○	アプリケーション ID (SVID)。BNE が発行。“1241”など
client_id	○	○	クライアント ID。BNE が発行
client_secret	○	○	クライアントシークレット。BNE が発行
app_user_id	○	○	ユーザーID。アプリにおけるユーザーを区別する任意の ID。最大 128 文字以内の文字列
app_system_prefix			app_user_id と併せてユーザーをユニークに識別するための補助識別子。 app_user_id のみでユニーク性が保証されている場合は不要。最大 10 文字
buy_coin			購入コイン数。購入したコインや魔法石などの有償通貨の数を設定します。なお、buy_coin の設定時は buy_amount にも設定が必要です。省略したときは「0」が設定されます
buy_amount	※1	※1	購入コイン金額。購入したコインや魔法石などの有償通貨の購入金額を設定してください。国内のアプリでは「円」換算の金額を指定する必要があります。DOUBLE 型なので小数以下の設定も可能です。プレゼントなど、無償で提供した場合は、buy_coin の設定時は「0」を設定してください。なお、buy_amount の設定時は buy_coin にも設定が必要です
pay_coin			消費コイン数。消費したコインや魔法石などの数を設定します。なお、pay_coin の設定時は pay_amount にも設定が必要です。省略したときは「0」が設定されます
pay_amount	※1		消費コイン金額。コインや魔法石などの消費金額を設定してください。国内のアプリでは「円」換算の金額を指定する必要があります。DOUBLE 型なので小数以下の設定も可能です。プレゼントなど、無償で提供した場合は「0」を設定してください。pay_amount の設定時は pay_coin にも設定が必要です
direct_amount	※1	※1	直接課金金額。有償通貨を使用せずに直接アイテム購入した場合に、こちらに直接課金の金額を設定してください。
subscription_amount	※1	※1	定期購入の場合に、こちらの金額を設定してください。

item_id			アイテム ID。有償通貨および直接課金での商品購入、商品消費発生時に設定します。 アイテム ID は最大 50 文字までの任意の文字列で、どのように利用するのかは任意です
list_price			定価。版元とのロイヤリティ契約上、割引販売前の価格が必要な場合に設定します
platform_id	○	○	BNE 運用チームが定めたプラットフォーム ID。 詳細については、4.2.5 プラットフォームを特定する情報を参照
insert_time	○	○	有償通貨を購入または消費した日時。JST。YYYY-MM-DD HH:MI:SS の書式。「2016-01-20 07:50:32」など
country_code		○	国コード。国内配信アプリは「JP」固定、海外版。「6 データに関する共通の注意」を参照
currency_code		○	通貨コード。国内配信アプリは「JPY」固定。「6 データに関する共通の注意」を参照

※1 buy\_amount、pay\_amount、direct\_amount、subscription\_amount の何れかが必須設定。有償通貨の購入時にはbuy\_amount、消費時はpay\_amount、直接課金時にはdirect\_amount、定期購入時にはsubscription\_amountを指定する

#### 4.2.1 データ送信元を特定する情報

「app\_id」はゲームアプリ(ゲームタイトル)ごとに発行されるユニークキーです。また「client\_id」「client\_secret」は、「app\_id」と結びつけ、BanK システムへ送信されたデータを受信するときの認証に利用されます。詳しくは「7-4 データ送信のサンプルプログラム」を参照してください。

#### 4.2.2 ユーザーを特定する情報

app\_user\_id と app\_system\_prefix は、コインや魔法石などの購入や消費をしたユーザーを特定する ID です。  
「3.2.3 ユーザーを特定する情報」を参照してください。

#### 4.2.3 購入または消費のデータ

購入または消費のデータは、buy\_coin、buy\_amount、pay\_coin、pay\_amount の 4 つの項目で指定します。

「購入したとき」と「消費したとき」は、それぞれ、下記のデータを設定します。より詳しくは、「4.3 有償通貨における課金のタイミング」で説明します。

##### 【有償通貨で購入したとき】

buy\_coin にコインや魔法石などの数を、buy\_amount に購入した金額を設定します。pay\_coin、pay\_amount は指定しません。無償で提供した場合は、buy\_amount に「0」を設定します。

item\_id には、購入したコインや魔法石、アイテムを識別するための任意の ID を指定できます。

この指定は、アプリごとに任意です。

##### 【有償通貨で購入したものを消費したとき】

pay\_coin にコインや魔法石などの数を、pay\_amount に、消費した相当金額を設定します。buy\_coin、buy\_amount は指定しません。

item\_id には、消費したコインや魔法石、アイテムを識別するための任意の ID を指定できます。

#### 4.2.4 時刻データ

購入または消費の日時を、insert\_time に設定します。

#### 4.2.5 プラットフォームを特定する情報

`platform_id` は、プラットフォームを特定する情報を示します（表 4-2）。

また BanK 運用チーム（BNE）が定めた値を指定することもあります。

表 4-2 プラットフォーム ID 定義

プラットフォーム	設定値
iOS	ios
android, androidPC(GooglePlay Games)	android
DMM	dmm
Steam	steam
PS4/PS5	ps4, ps5
Xbox SeriesX S /Xbox One	xsx
Nintendo Switch	nsw
Windows	win
WebStore/アソビストア(iOS) ※	ios_asb
WebStore/アソビストア(android) ※	and_asb
WebStore/アソビストア ※	asb

※WebStore/アソビストアについては「9. WebStore/アソビストアについて」を参照

#### 4.2.6 国コードと通貨コード

国コードを `country_code`、通貨コードを `currency_code` に設定します。海外配信アプリでは必ず指定する必要があります。

書式は「6 データに関する共通の注意」を参照してください。国内配信アプリはそれぞれ「JP」、「JPY」に固定されます。

### 4.3 有償通貨における課金のタイミング

f002 のデータを BanK システムに送信するタイミングは、「コインや魔法石などの有償通貨を購入したとき」「消費したとき」の 2 つのタイミングがあります。

#### 4.3.1 コインや魔法石などの有償通貨を購入したとき

たとえば、¥1,000 でコイン 50 個を買ったとします。この場合、`buy_coin`、`buy_amount` を次のように構成した f002 データを送信します。

【コインを¥1,000 で 50 個買ったとき】

`buy_coin = 50`      `buy_amount = 1000`

`buy_coin` はコインの数です。`buy_amount` は、この取引で支払った金額です。

また、このとき「コインの種類」を区別するために、`item_id` に任意の ID を指定することもできます。ID は最大 50 文字までの任意の文字列です。

#### 4.3.2 コインや魔法石などを消費したとき

¥1,000 で購入した 50 個のコインを持っているとき、そのうちの 10 個のコインを消費して、何かアイテムを購入したとします。このとき、`pay_coin`、`pay_amount` を次のように構成した f002 データを送信します。

【コイン 10 個を消費して、アイテムを購入したとき】

`pay_coin = 10`      `pay_amount = 200`    `item_id = アイテムの ID`

`pay_amount` は、消費するコインに相当する金額です。

前の手順のように、¥1,000 で 50 個買っているのであれば、コイン 1 個は「¥1,000 ÷ 50 個 = ¥20」です。よって 10 個のコインの価値は、「¥20 × 10 = ¥200」なので、これを `pay_amount` として設定します。

`item_id`には、購入したアイテムのアイテム ID を格納します。ID は最大 50 文字までの任意の文字列です。

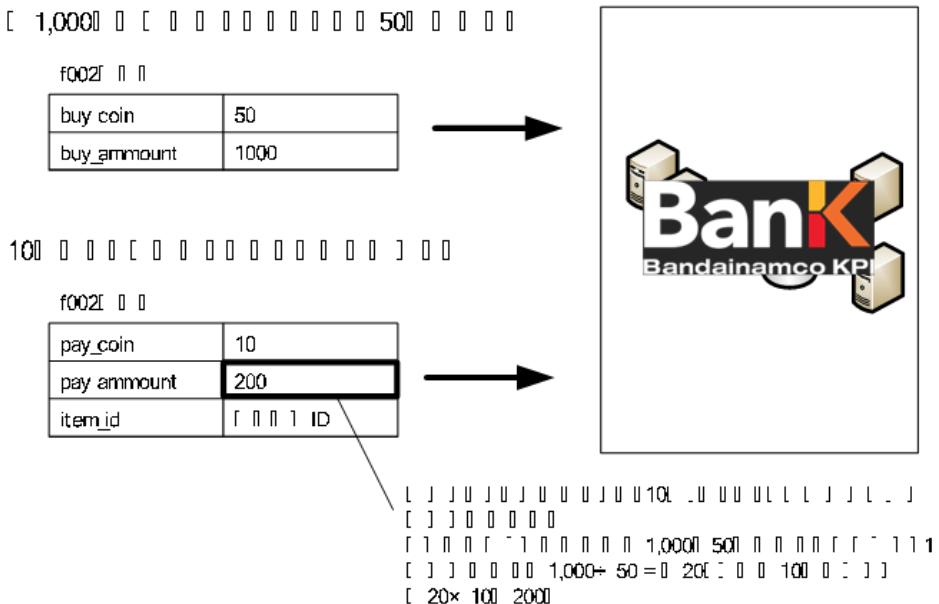


図 4-1 f002 データ送信の基本

#### 4.3.3 無償でコインや魔法石などを提供する場合

ボーナスプレゼントなどでコインや魔法石などを無償配布するときは、`buy_coin` に配布した数、`buy_amount` に 0 を設定してください。つまり、「ユーザーは¥0でそれらを購入した」というデータを作ります。

たとえば、コイン 5 個を無償配布したときは、次のように構成した f002 データを送信します。

【コイン5個を無償配布したとき】

buy\_coin = 5                  buy\_amount = 0

#### 4.3.4 購入時の単価が違う場合の消費金額

`pay_amount` には、消費するコインに相当する価格を設定する必要があります。

有償通貨の販売単位における、1個当たりの単価を割り出し、消費したコイン数の積算から求められた値を設定してください。

単価は販売される有償通貨（コイン）の単位によって異なります。

たとえば、「50コインなら¥1,000」「110コインなら¥2,000」「300コインなら¥5,000」を、有償通貨として購入した場合はそれぞれ、

- ①buy\_coin = 50, buy\_amount = 1000
  - ②buy\_coin = 110, buy\_amount = 2000
  - ③buy\_coin = 300, buy\_amount = 5000

という f002 データを送信します。それぞれの単価は、「¥20」「約¥18.18」「約¥16.66」と異なります。

これらの購入されたコインから「コイン 60 個を消費してアイテムを購入する」場合、  
pay\_amount=60 に違いありませんが、pay\_amount は、組み合わせによって換算する必要があります。たとえば、下記に示す組み合わせが考えられます。

### 【パターンA ①と②のケース】

①が 50 枚で ¥1,000、②が 残り 10 枚なので  $¥18.18 \times 10 = ¥181.8$ 。よって、 $¥1,000 + ¥181.8 = ¥1,181.8$

### 【パターンB ①と③のケース】

①が50枚で¥1,000、③が残り10枚なので $\text{¥}16.66 \times 10 = \text{¥}166.6$ 。よって、 $\text{¥}1,000 + \text{¥}166.6 = \text{¥}1,166.6$

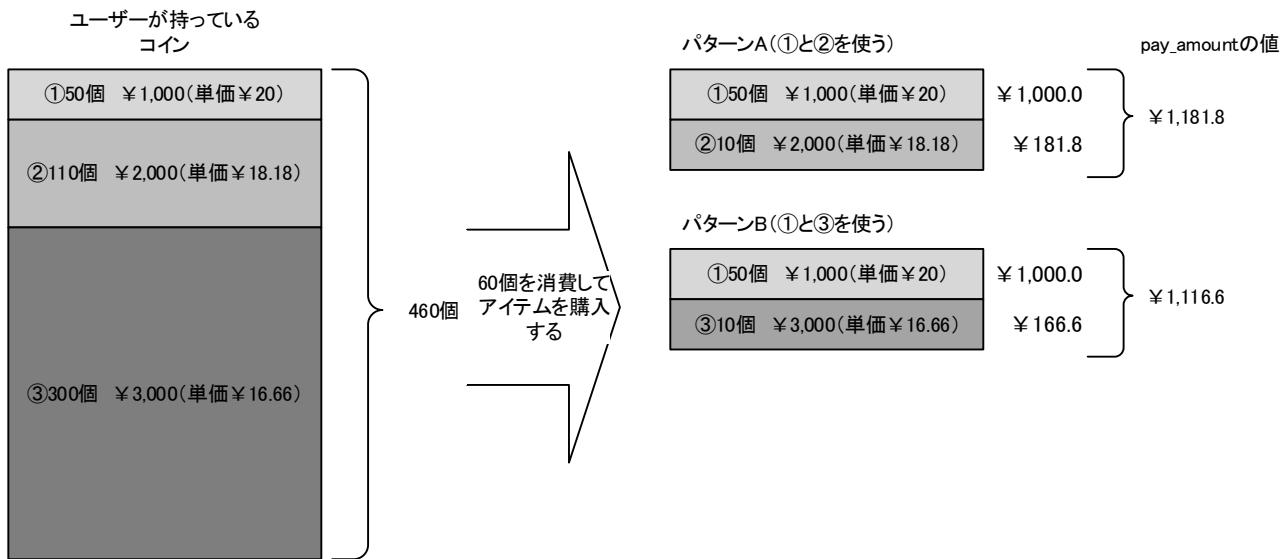


図 4-2 購入時の単価が違うコインを組み合わせて支払うケース

有償通貨（コイン）を消費する順番は、先入れ先出しを基本とします。

単価の異なるコインの消費データの送信する場合、上記のように合算して、送信してください。これは課金回数のデータを正しく表示するためです。

#### 4.4 BanK における集計項目との関連性

f002 で送信したデータは、BanK システムの下記の項目に関連します。

##### ◆ 購入額・消費額

国内：コインや魔法石などを“消費した”金額  
海外：コインや魔法石などを“購入した”金額

##### ◆ 課金率

課金UU(Unique User)数 ÷ アクティブユーザー数 × 100%

##### ◆ 課金率（平均）

指定期間が月別の場合はデイリー値の平均。指定期間が日別の場合は、毎月 1 日～該当までの平均値

##### ◆ 課金UU

国内：コインや魔法石などを“消費した”ユーザーのユニーク数  
海外：コインや魔法石などを“購入した”ユーザーのユニーク数

##### ◆ 課金回数

コインや魔法石などを“消費した”回数

##### ◆ ARPU (Average Revenue Per User)

課金額 ÷ アクティブユーザー数

##### ◆ ARPU（平均）

指定期間が月別の場合はデイリー値の平均。指定期間が日別の場合は、毎月 1 日～該当までの平均値

##### ◆ ARPPU (Average Revenue Per Payment User)

課金額 ÷ 課金UU

◆<重複利用>課金ユーザー

当該アプリにて課金をしたことがあるユーザーを対象として、重複利用しているアプリを表示

#### 4.5 国内版と海外版との違い

BanK では、f002 で送信されたデータを基に購入額・消費額を集計します。集計仕様は国内配信アプリと海外配信アプリで異なります。

国内配信アプリは、有償通貨の消費データを、海外配信アプリは有償通貨の購入データを、それぞれ売上として表示します。

消費データとは pay\_coin／pay\_amount に消費情報を含むもの、購入データとは buy\_coin／buy\_amount に購入情報を含むものを指します。

●国内アプリでは円固定

国内アプリでは、金額は「JPY（円）」に固定です。currency\_code に「JPY 以外」を指定しても無視されます。ユーザーが他の通貨で購入したとしても、JPY（円）で設定している金額を送信してください。

●海外版は現地通貨

海外アプリの場合の消費額・購入額は現地通貨とし、その現地通貨の金額とともに、通貨コード「currency\_code」を送信してください。「6 データに関する共通の注意」を参照

BanK 上で表示される消費額・購入額は、消費や購入された日の換算金額（円）（※前日終値を適用）で表示されます。

#### 4.6 コインや魔法石などの有償通貨を購入したときの設定例

ここまで説明をもとに、具体的な設定例を示します。まずは、コインや魔法石などの有償通貨を購入したときです。

ユーザーが、これらの有償通貨を購入したときには、buy\_coin にその数量、buy\_amount にその価格を格納した f002 データを送信します。たとえば、「50 コインを￥1,000 で購入したとき」の処理は、リスト 4-1 のようになります。

リスト 4-1 有償通貨を購入する場合の例

```
{"app_id": "アプリケーション ID", "client_id": "クライアント ID", "client_secret": "クライアントシークレット",
"app_user_id": 1234567890, "app_system_prefix": "", "platform_id": "android", "buy_coin": 50,
"buy_amount": 1000, "insert_time": "2021-02-10 11:34:00"}
```

#### 4.7 コインや魔法石などの有償通貨を消費したときの設定例

コインや魔法石などを消費してアイテムを購入したときには、pay\_coin にコイン数、pay\_amount に相当金額、item\_id に購入したアイテム ID を設定します。

たとえば、全体で￥600 の価値がある 30 個のコインを使って、「sord01」というアイテムを購入したときは、リスト 4-2 に示すデータを送信します。

リスト 4-2 コインや魔法石などを消費する例

```
{"app_id": "アプリケーション ID", "client_id": "クライアント ID", "client_secret": "クライアントシークレット",
"app_user_id": 1234567890, "app_system_prefix": "", "platform_id": "android", "pay_coin": 30,
"pay_amount": 600, "insert_time": "2021-02-10 11:52:00"}
```

#### 4.8 定期購入の課金タイミングについて

以下に月額の定期購入における、継続時と終了時の送信例を記載

【自動更新の場合】

2017/10/01 定期購入開始 subscription\_amount=1000 を送信

~  
2017/11/01 定期購入継続 subscription\_amount=1000 を送信  
....

【自動更新を停止、または非自動更新の場合】

2017/10/01 定期購入開始 subscription\_amount=1000 を送信  
~2017/10/15 定期購入停止～  
2017/11/01 定期購入終了 送信しない

## 5 サービス売上情報に関するデータ送信 (f003)

この項では、国内配信アプリにおける、日次および月次の売上データを BanK に送信する際のデータについて説明します。

※海外アプリは送信不要です。

### 5.1 f003 概要

f003 売上情報で送信される情報は、国内配信アプリの売上計上やロイヤリティ支払に利用されます。また、財務局への有償通貨残高報告にも利用されます。

資金決済法に基づく「売上」は、コインや魔法石などの有償通貨を消費したタイミングであると定められています。そのため、この節で述べる「売上」とは、コインや魔法石などが購入された金額ではなくて、その日（または月）に、消費されたコインや魔法石などの金額を示します。

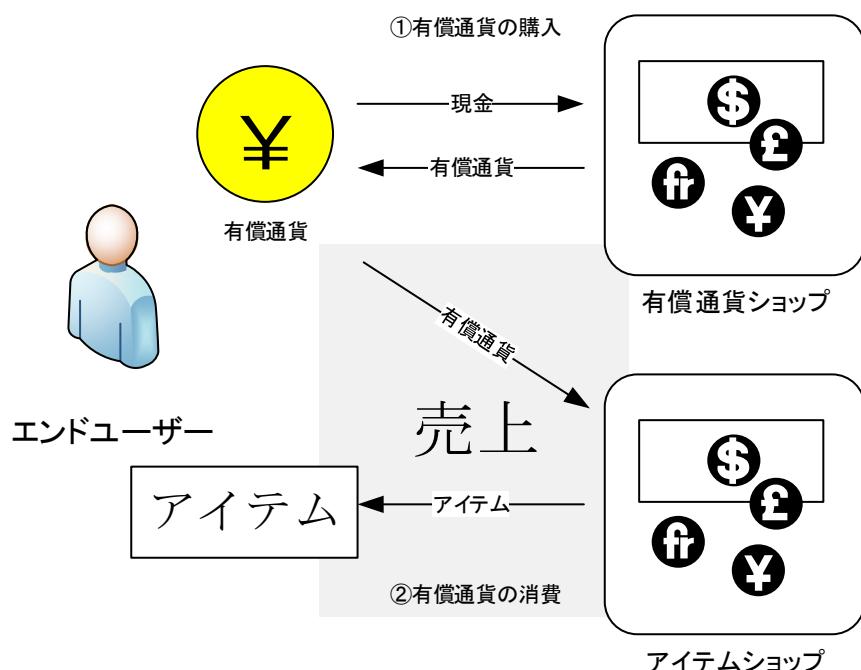


図 5-1 有償通貨における売上のタイミング

f003 で送信されたデータは財務局へ報告します。f003 へ送信する有償通貨に関する購入情報・消費情報は正確性を求められることに留意してください。

なお、送信したデータは「BanK Moderator サービス売上情報」の画面

(<https://bankadm.channel.or.jp/moderator/?p=support.service.flu.summaryPointCash>) にて確認可能です。



図 5-2 f002 と f003 との違い

## 5.2 f003 のデータテーブル定義

f003 のデータテーブル定義は、表 5-1 の通りです。本項では、必須となる重要な部分のみ説明します。すべての項目の意味は、Appendix に示します。

表 5-1 f003 のデータテーブル定義

列名	必須 (国内)	必須 (海外)	解説
app_id	○	—	アプリケーション ID (SVID)。登録申請時に取得した値
date	○	—	売上報告の年月 (YYYYMM の 6 桁)。もしくは年月日 (YYYYMMDD の 8 桁)。
total_sales	○	—	有償通貨における、date の期間内の売上合計の金額 (日本円)
platform_id	○	—	BNE 運用チームが定めたプラットフォーム ID。 詳細については、4.2.5 プラットフォームを特定する情報を参照
data	○	—	それぞれのコインや魔法石などのセットごとの明細データを JSON 形式で示した配列 (表 5-2 を参照)
direct_total_sales	※1	—	直接課金における、date で指定した月次または日次の売上を送信してください。
direct_data	※1	—	直接課金で購入したアイテムの明細データを JSON 形式で示した配列 (表 5-3 を参照)
subscription_total_sales	※2	—	定期購入における、date で指定した月次または日次の売上を送信してください。
subscription_data	※2	—	定期購入アイテムを JSON 形式で示した配列 (表 5-4 を参照)
total_sales2	※3	—	2 種目の有償通貨における、date の期間内の売上合計の金額 (日本円)
data2	※3	—	2 種目の有償通貨における、それぞれのコインや魔法石などのセットごとの明細データを JSON 形式で示した配列 (表 5-2 を参照)

※1 直接課金を行う場合は必須

※2 定期購入を行う場合は必須

※3 有償通貨を 2 種使い分ける際は必須。「total\_sales」「date」に 1 種目、「total\_sales2」「date2」に 2 種目を設定してください。2 種目の有償通貨については「[10. 2 種類の有償通貨について](#)」を参照

表 5-2 『表 5-1 f003 データテーブル定義』における、『data』『data2』列のテーブル定義

列名	必須 (国内)	必須 (海外)	解説
name	○	—	コインや魔法石セットの名称。「コイン 50 パック」「魔法石セット C」「300 ゴールド」など
coin	○	—	販売コイン数。このアイテムを買ったときに、コインがいくつ増えるか。「コイン 50 パックなら『50』」「魔法石セット C なら『12』」「300 ゴールドなら『300』」
price	○	—	name を販売するときの価格。「コイン 50 パックなら『2400』(円)」「魔法石セット C なら『3840 (円)』」「300 ゴールドなら『3300 (円)』」など
total_count	○	—	報告期間（表 5-1 の date）におけるコイン（有償・無償通貨）の発行総数
total_consumption	○	—	報告期間（表 5-1 の date）におけるコイン（有償・無償通貨）の消費総数

表 5-3 『表 5-1 f003 データテーブル定義』における、『direct\_data』列のテーブル定義

列名	必須 (国内)	必須 (海外)	解説
name	○	—	販売時の名称（スタミナ回復 10 個など）
price	○	—	販売時の値段（スタミナ回復 240 (円)）
total_count	○	—	月次または日次で指定した期間内におけるアイテムの総数 ※「スタミナ回復 10 個」も 1 としてカウント

表 5-4 『表 5-1 f003 データテーブル定義』における、『subscription\_data』列のテーブル定義

列名	必須 (国内)	必須 (海外)	解説
name	○	—	販売時の名称（プレミアムプランなど）
price	○	—	販売時の値段（プレミアムプラン 3,800 (円)）
total_count	○	—	月次または日次で指定した期間内におけるアイテムの総数
unit	○	—	1:日、2:週、3:月を選択
span	○	—	UNIT で指定した期間単位がどの期間に渡るか (例) 3 週間の場合は unit に「2」span を「3」で設定 ※BNE のレギュレーションとしては、定期購入（サブスクリプション）は最長一ヶ月となります。

### 5.3 集計対象

集計対象は、コインや魔法石などの個数と単価です。

f003 データ運用の性格上、送信データに反映されたコインや魔法石などの個数と単価には正確性が求められます。

理解を深め、正しいデータを送信するため、下記に留意すべきポイントを挙げます。

ここでは、次のケースを想定します。

例) あるアプリで「コイン 1 個」「コイン 50 パック」「コイン 100 パック」という 3 種類のコインパックがあるとし、これら集計対象を f003 データとして、2016 年 10 月 10 日に送信する場合（表 5-3）。

表 5-3 集計対象

コインパック名 (name)	パック 1 個に含まれるコインの個数 (coin)	販売価格 (price)	コイン 1 個当たりの単価 (price ÷ coin)
コイン 1 個	1	50 円	50 円
コイン 50 パック	50	2,400 円	48 円
コイン 100 パック	100	4,500 円	45 円

①コインパック単位でコインの発行数を算出する

コインパックごとにコインの発行数を算出してください。

これは購入金額を算出するとき、コインパック単位でコイン単価が異なるためです。

仮に、2016 年 10 月 10 日に、「1 個パックが 5 個、50 パックが 10 個、100 パックが 7 個売れた」とすると、次のようにになります。

下記において、「発行総数」が、それぞれのコインパックごとの発行数です。「1 個パックが 5 個であれば、 $1 \times 5 = 5$  コイン」、「50 パックが 10 個であれば  $50 \times 10 = 500$  コイン」、「100 パックが 7 個であれば  $100 \times 7 = 700$  コイン」を、それぞれ発行したという意味になります。

コインパック名	販売価格	発行パック数	発行総数 (total_count)
コイン 1 個	50 円	5	5
コイン 50 パック	2,400 円	10	500
コイン 100 パック	4,500 円	7	700

この「発行総数」は、データ送信時に「total\_count」の値として設定します。

### ②コインパック単位でコインの消費数を算出する

コインパックごとにコインの消費数を算出してください。

これは消費金額を算出するとき、コインパック単位でコイン単価が異なるためです。送信対象となるコインパック単位で、算出してください。

仮に、2016 年 10 月 10 日に、「1 個パックとして売られたコインが 10 個、50 パックとして売られたコインが 40 個、100 パックとして売られたコインが 30 個消費された」とすると、次のようにになります。

下記において、「消費個数」が、それぞれのコインパックごとの消費数です。

コインパック名	販売価格	消費個数 (total_consumption)
コイン 1 個	50 円	10
コイン 50 パック	2,400 円	40
コイン 100 パック	4,500 円	30

「消費個数」は、コインの消費された総数になります。

なお、「消費個数」はコインパックの数ではなく、購入済みコインパックにおける、消費されたコインの個数を指します。

この「消費個数」は、データ送信時に「total\_consumption」の値として設定します。

### ③コインパック単位で消費金額を算出する

②で求めた消費個数から、相当する消費金額を算出します。

送信対象となるコインパック単位で、算出してください。下記のようになります。

コインパック名	販売価格	消費個数 (total_consumption)	コイン 1 個当たり の単価 (price ÷ coin)	消費金額 (total_consumption × 単価)
コイン 1 個	50 円	10	50 円	500 円
コイン 50 パック	2,400 円	40	48 円	1,920 円
コイン 100 パック	4,500 円	30	45 円	1,350 円

こうして求めた「消費金額」の合計 3,770 円 (500 円 + 1,920 円 + 1,350 円) は、データ送信時に「total\_sales」の値として設定します。

上記①～③の計算をまとめると、送信するデータのイメージは、下記の通りとなります。

※データの項目名については、「表 5-1 f003 データテーブル定義」を参照してください。

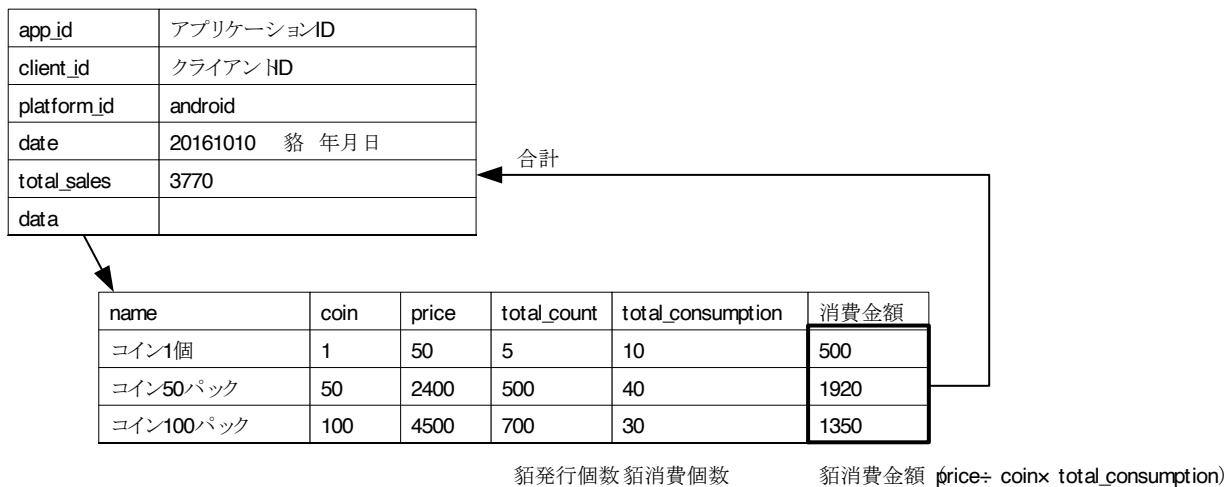


図 5-3 データテーブル定義を図示したもの

## 5.4 データ送信

f003 データの送信は、日次送信、月次送信の 2 通りです。

※日次、月次とも必須となります。

各送信概要は、表 5-4 の通りです。

表 5-4 データ送信概要

送信処理	送信単位	送信タイミング
日次送信	日単位	前日分集計後
月次送信	月単位	前月分集計後

### 5.4.1 日次送信

送信にあたり、送信プログラムを作成してください。

下記のリスト 5-1 は、日次送信プログラムの設定例です。

#### リスト 5-1 日次送信の例

```
{"app_id": "アプリケーション ID", "client_id": "クライアント ID", "client_secret": "クライアントシークレット",
"platform_id": "android", "date": "20210220", "total_sales": 3770,
"data": [{"name": "コイン1個", "coin": 1, "price": 50, "total_count": 5, "total_consumption": 10},
 {"name": "コイン50パック", "coin": 50, "price": 2400, "total_count": 500, "total_consumption": 40},
 {"name": "コイン100パック", "coin": 100, "price": 4500, "total_count": 700, "total_consumption": 30}], }"
```

### 5.4.2 月次送信

月次送信は、送信単位が月単位になります。該当する月の初日から月末日が対象です。送信にあたり、送信プログラムを作成してください。

下記のリスト 5-2 は、月次送信プログラムの設定例です。

#### リスト 5-2 月次送信の例

```
{"app_id": "アプリケーション ID", "client_id": "クライアント ID", "client_secret": "クライアントシークレット",
"platform_id": "android", "date": "202102",
※以下は日次送信と同じ構造のため省略}
```

※月次の場合、日付を指定する date 列は「YYYYMM」(YYYY=年、MM=月) の 6 術の数字で示すという点が異なります。例は、2021 年 2 月を示しています。

## 5. 5 f003 データ送信の注意点

f003 データを送信するときは、次の点に注意してください。

### ① 同一日のデータの再送信は上書きされる

date 部が同一のデータを再送した場合は、新しい値で上書きされます。

送信済みデータの修正が発生した場合は、適宜再送信をお願いします。

### ② 端数について

コインの単価の算出時に端数が発生する場合があります。その際は、小数点以下の値を極力出さないよう、販売商品ごとに、単位期間での消費数を合計し、販売個数で割った商 × 販売価格 + 余り × 販売単価で集計します。

たとえば、「コイン1個」と「コイン30パック」があり、それぞれの販売価格が120円、2000円だとします。このとき、ある日付に、前者が「1個」、後者が「41個」、それぞれ消費された場合、その日次売上は、次のように計算します。

---

### 【日次売上の計算例】

コインパック名	コイン1個	コイン30パック
コイン数	1	30
販売価格	120 円	2000 円
期間消費数	1	41
期間売上	$120 \text{ 円} \times 1 = 120 \text{ 円}$	$41 \div 30=1 \text{ 余り } 11$ $2000 \text{ 円} \times 1 + 2000 \text{ 円} \div 30 \times 11 = 2,733.33\dots \text{ 円}$

上記の計算により、日次の売上は、

120 円 + 2,733.33... 円 = 2853 円（小数以下切り捨て）

※販売商品ごとの期間売上を合算した後、小数点以下を切り捨て

---

### ③ アラートメールを参照する

送信後、正しくデータが取り込まれない場合は、事前に登録されたメールアドレスにアラートメールが送信されるので、参照願います。

※本件へのお問い合わせは弊社 BanK 担当窓口までお願いします。

メールアドレス : bank\_mgmt@m365.bandainamco.co.jp

### ④ 重複購入の課金キャンセル処理

Google Play の障害などによって重複購入が発生し、お客様から返還の問い合わせがあった場合、BNE 側で課金キャンセルを実施するケースがあります。

その場合、アプリ上で返還対象となる有償通貨を回収、または無償通貨に変換した後、「返還した日付」の有償通貨発行数を返還分だけ減らした f003 データを送信願います。

※返還の前提条件として Google Play 版であること、返還対象の有償通貨が未使用であること

※データを返還分だけ減らして送信するのは「重複購入が発生した日付」ではなく「返還した日付」である点にご注意ください。

## 6 データに関する共通の注意

データを送信するときは、下記の規定があります。

### 6.1 日時

日時は、すべて「JST（日本時刻）」です。海外版であっても、JST で送信してください。

日時の書式は「YYYY-MM-DD HH:MM:SS」です。たとえば、「2016-10-01 11:15:00」などの値です。

※「2.3.2 アップロードする S3 のパス」に記載したように、S3 バケットへデータ配置する際のディレクトリパスは UTC で作成ください。

### 6.2 国コード

データには、「国コード」を指定しなければならない項目があります（表 3-1, 4-1 の項目名 country\_code）

国コードは「JP」「KR」「NZ」「FR」などの 2 文字（大文字）です。

次のメソッドで取得した値を採用してください。国コードを省略したときは「JP」であると見なされます。

なお、国内版アプリでは、何を設定しても BanK 内で「JP」に変換されます。

※コンシューマ機については後述の「8. スーパーマルチタイトルについて」をご参照ください。

#### 【iOS】

```
[[NSLocale currentLocale] objectForKey:NSLocaleCountryCode]
```

#### 【Android】

```
java.util.Locale.getDefault().getCountry();
```

### 6.3 通貨コード

データには、「通貨コード」を指定しなければならない項目があります（表 4-1 の項目名 currency\_code）  
通貨コードは、「JPY」「KRW」「EUR」「CNY」など 3 文字（大文字）です。

次のメソッドで取得してください。通貨コードを省略したときは、「JPY（日本円）」であると見なされます。  
なお、国内版アプリにおける金額は、外貨で購入された場合も日本円の価格を送信してください。  
※コンシューマ機については後述の「8. スーパーマルチタイトルについて」をご参照ください。

#### ●ユーザー課金履歴(f002)における通貨コード取得方法

有償通貨購入時(buy\_coin、buy\_amount 記載時)は以下手段で通貨コード(および外貨価格)を取得して下さい  
【iOS】

SKProduct Class

[https://developer.apple.com/library/ios/documentation/StoreKit/Reference/SKProduct\\_Refere...](https://developer.apple.com/library/ios/documentation/StoreKit/Reference/SKProduct_Refere...)  
にて、price および priceLocale を取得

#### 【Android】

getSkuDetails

[http://developer.android.com/google/play/billing/billing\\_reference.html#...](http://developer.android.com/google/play/billing/billing_reference.html#...)  
にて、price\_amount\_micros (※) および price\_currency\_code を取得

※マイクロ単位となるため、double 型にキャストし 100 万で割った値  
(price\_amount\_micros / 1000000) を送信ください。

## 6.4 プラットフォーム別ユーザー識別子（IDFV/Android ID）

データには、「プラットフォーム別ユーザー識別子」を指定しなければならない項目があります（表 3-1 の項目名 platform\_user\_id）。

プラットフォーム毎にそれぞれ IDFV(iOS)、Android ID(Android) を送信してください。

※下記 API で取得した値を加工せず送信してください。

※コンシューマ機については後述の「8. スーパーマルチタイトルについて」をご参照ください。

### 【iOS】

```
[UIDevice currentDevice].identifierForVendor
```

※ <https://developer.apple.com/documentation/uikit/uidevice/1620059-identifierforvendor>

### 【Android】

```
Settings.Secure.getString(getApplicationContext(), Settings.System.ANDROID_ID);
```

※ [https://developer.android.com/reference/android/provider/Settings.Secure.html#ANDROID\\_ID](https://developer.android.com/reference/android/provider/Settings.Secure.html#ANDROID_ID)

## 7 ユーザーデータの引継ぎ

### 7.1 引継ぎ方法について

ユーザーデータの引継ぎについては、同一プラットフォームおよびプラットフォームを跨ぐケースで、それぞれ以下の対応をお願い致します。

#### 7.1.1 同一プラットフォームの場合

同一プラットフォームにおけるデータの引継ぎでは、BanK 上、同一ユーザーとして継続率を保つ必要があるため、行動履歴 (F001) は、以下のパラメータを引継ぎ前から変更せずに送信してください。

```
app_user_id  
app_system_prefix ※任意項目のため、設定していた場合のみ  
created_time
```

#### 7.1.2 プラットフォームを跨ぐ場合

プラットフォームを跨ぐデータ引継ぎについては、BanK 上、別ユーザーとして扱うため、行動履歴 (F001) の platform\_id のみを変更し、新規ユーザーとして event\_id=100 を送信してください。

なおプラットフォームを跨ぐ引継ぎの際、引継ぎ前のユーザーに対して event\_id=200 を送信する必要はありません。

#### 【例 : Android ⇒ iOS】

##### ▼引継ぎ前ユーザー情報

```
platform_id=android,  
app_user_id=xxx,  
app_system_prefix=yyy ※任意項目のため、設定していた場合のみ  
created_time=2021-07-31 00:00:00
```

##### ▼引継ぎ後ユーザー情報

```
platform_id=ios,  
app_user_id=xxx,  
app_system_prefix=yyy ※任意項目のため、設定していた場合のみ  
created_time=2021-08-01 00:00:00
```

※ 「app\_user\_id」及び「app\_system\_prefix」は、変更せずに引継ぎ前の値を設定して下さい。

#### 7.1.3 引継ぎ時の課金ログについて

プラットフォーム跨ぎの引継ぎに限り、引継ぎ元のユーザーID にて残っているコインをすべて消費として、課金履歴 (F002、F003) を送信してください。

## 8 スーパーマルチタイトルについて

1つのセーブデータでモバイル、コンシューマ機(PS, Switch, Xbox, Steam など)両方でプレイできる  
タイトルを「スーパーマルチタイトル」と呼んでいます。  
この項では主にコンシューマ機が、BanK に送信するデータについて説明します。  
※対応 PF がモバイル(iOS, Android)のみの場合、この章は関係ありません。

### 8.1 国内・海外での扱いについて

BanK では同じタイトルでも国内と海外で分けて管理しています。

コンシューマ機においては共通の ROM で配信する場合が主です。その為、各プラットフォームにおける  
ユーザーの国コードを参照し、BanK へ送信する際のアプリケーション ID(app\_id)を設定してください。  
※ROM が国内・海外で分かれる場合は、モバイルと同様にそれぞれのアプリケーション ID(app\_id)を  
設定してください。

### 8.2 国コードの取得方法について

各プラットフォームで用意されているライブラリや API を用いて国コード(居住国)を取得してください。

#### 【steam】

SteamWorksWebAPI GetUserInfo の response に設定されている country の値を設定してください。  
<https://partner.steamgames.com/doc/webapi/ISteamMicroTxn#GetUserInfo>

#### 【PS4/PS5】

Np ライブラリの sceNpGetAccountCountryA の pCountryCode を設定してください。  
<https://p.siedev.net/resources/documents/SDK/8.000/Np-Reference/0006.html>

#### 【Xbox SeriesX|S /Xbox One】

GetUserDefaultGeoName で取得した国コードを設定してください。  
<https://learn.microsoft.com/ja-jp/windows/win32/api/winnls/nf-winnls-getuserdefaultgeoname>  
[https://learn.microsoft.com/ja-jp/gaming/gdk/\\_content/gc/system/overviews/localization/supported\\_nls\\_apis](https://learn.microsoft.com/ja-jp/gaming/gdk/_content/gc/system/overviews/localization/supported_nls_apis)

#### 【Nintendo Switch】

購入トランザクションのレスポンスのみに国コードが含まれている為、購入前に於いて国情情報を取得できません。  
その為、ユーザー行動履歴(f001)は IP アドレスを元に変換した国コードを設定してください。  
ユーザー課金履歴(f002)、及びサービス売上情報(f003)については、購入トランザクションのレスポンスに設定されている  
country の値を設定してください。  
[https://developer.nintendo.com/html/online-docs/g1kr9vj6-ja/Packages/Docs/Guides/NintendoSwitch-ECommerce\\_Guide-ja/contents/Pages/Page\\_471905206.html](https://developer.nintendo.com/html/online-docs/g1kr9vj6-ja/Packages/Docs/Guides/NintendoSwitch-ECommerce_Guide-ja/contents/Pages/Page_471905206.html)

## 9 WebStore/アソビストアについて

WebStore (<https://jp.bandainamcoentwebstore.com/>)、およびアソビストア (<https://shop.asobistore.jp/>) は BNE が運営するコマースサイトです。

WebStore、およびアソビストア対応タイトルは、ストアにてタイトルで利用できる有償通貨やアイテム等を購入することができます。

### 9.1 platform\_id について

WebStore/アソビストアの platform\_id は「ios\_asb」、「and\_asb」、もしくは「asb」を設定します。  
どの platform\_id を設定するかは、「マルチログイン対応タイトルかどうか」で決まります。

**※ここでのマルチログインとは iOS→android 間の場合となります。**

表 9-1 アソビストア platform\_id 設定定義

マルチログイン対応有無	設定値
マルチログイン非対応 (iOS)	ios_asb
マルチログイン非対応 (android)	and_asb
マルチログイン対応 (iOS/android)	asb

### 9.2 f001 で送信するデータについて

WebStore/アソビストアで行えるのはタイトルで利用できる有償通貨やアイテム等の購入のみになる為、f001 データは取り扱いません。f001 データとして platform\_id が「ios\_asb」、「and\_asb」、「asb」が設定されている場合はエラーとなりますので、送信しないよう注意ください。

### 9.3 f002、f003 で送信するデータについて

WebStore/アソビストアで有償通貨やアイテム等の購入、また WebStore/アソビストアで購入した有償通貨を消費した際は、platform\_id に「ios\_asb」、「and\_asb」、「asb」を設定の上、f002, f003 データを送信ください。

#### 【f002 設定例】

##### マルチログイン非対応の場合 (ios\_asb/and\_asb)

① iOS 版を利用しているユーザ (app\_user\_id : aaaa) が WebStore/アソビストアで有償通貨を購入した場合  
app\_user\_id : aaaa、platform\_id : ios\_asb で buy\_amount の f002 を送信

② ①のユーザが iOS 版アプリ内で①で購入した有償通貨を消費した場合  
app\_user\_id : aaaa、platform\_id : ios\_asb で pay\_amount の f002 を送信

② android 版を利用しているユーザ (app\_user\_id : bbbb) が WebStore/アソビストアで有償通貨を購入した場合  
app\_user\_id : bbbb、platform\_id : and\_asb で buy\_amount の f002 を送信

④ ③のユーザが android 版アプリ内で③で購入した有償通貨を消費した場合  
app\_user\_id : bbbb、platform\_id : and\_asb で pay\_amount の f002 を送信

⑤ iOS 版を利用しているユーザー (app\_user\_id : aaaa) がゲーム内で有償通貨を消費する際に、iOS、WebStore/アソビストアそれぞれで購入した有償通貨をまとめて消費した場合  
消費した有償通貨の購入元に合わせて、それぞれの f002 を送信

app\_user\_id : aaaa、platform\_id : ios で pay\_amount  
app\_user\_id : aaaa、platform\_id : ios\_asb で pay\_amount

##### マルチログイン対応の場合 (asb)

① iOS 版を利用しているユーザ (app\_user\_id : aaaa) が WebStore/アソビストアで有償通貨を購入した場合  
app\_user\_id : aaaa、platform\_id : asb で buy\_amount の f002 を送信

② ①のユーザが iOS 版アプリ内で①で購入した有償通貨を消費した場合

app\_user\_id : aaaa、platform\_id : asb で pay\_amount の f002 を送信

② android 版を利用しているユーザ (app\_user\_id : bbbb) が WebStore/アソビストアで有償通貨を購入した場合

app\_user\_id : bbbb、platform\_id : asb で buy\_amount の f002 を送信

④ ③のユーザが android 版アプリ内で③で購入した有償通貨を消費した場合

app\_user\_id : bbbb、platform\_id : asb で pay\_amount の f002 を送信

⑤ iOS 版を利用しているユーザー (app\_user\_id : aaaa) がゲーム内で有償通貨を消費する際に、iOS、WebStore/アソビストアそれぞれで購入した有償通貨をまとめて消費した場合

消費した有償通貨の購入元に合わせて、それぞれの f002 を送信

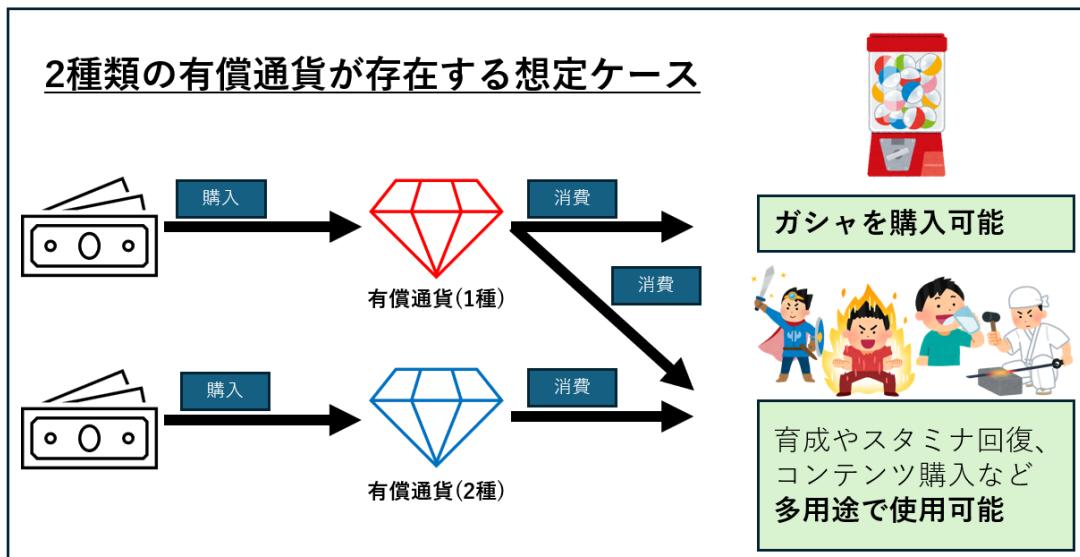
app\_user\_id : aaaa、platform\_id : ios で pay\_amount

app\_user\_id : aaaa、platform\_id : asb で pay\_amount

## 10 2 種類の有償通貨について

1つのタイトルで2種類の有償通貨を利用する場合について記載します。

※有償通貨は1種類のみというタイトルは、本項は関係しません。



f003 売上情報では、2種類の有償通貨毎に計上/報告する必要があります。

その為、前述の「5.2 f003 のデータテーブル定義」に記載の通り、有償通貨毎に分けて送信ください。

※有償通貨 A, B のどちらが total\_sales、total\_sales2 であるかは問いません。

※f002 については2種類の有償通貨は分けて扱わない為(分ける項目も無し)、種類に関係なく送信ください。