SPLIT THE POT

Integration documentation

1. Contents

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

2. Version history

Version	Date	Author	Comment
1.0.0	2021-09-17	C. Dargahi J. Redborg	Initial version
1.1.0	2021-09-29	C. Dargahi	Clarified game URL and its parts
1.1.1	2021-10-21	C. Dargahi	Clarified account ID
1.2.0	2021-10-30	C. Dargahi	Added isFreeRound to debit Added gameKind to debit and credit
1.3.0	2021-11-17	C. Dargahi	More detailed description of the user authentication token
1.4.0	2021-11-19	C. Dargahi	Added Dynamic Game Listing API Added 429 status code to APIs supported codes Changed wordings for clarity
1.4.1	2021-12-07	J. Redborg	Corrected 429 status code typos
1.4.2	2021-12-13	J. Redborg	Corrected updatedOn typos
1.5.0	2022-04-11	C. Dargahi	Updated SplitThePot logo Clarified token length and valid characters Clarified requirements for using DGL API Clarified hosting and location
1.5.1	2022-05-04	C. Dargahi	Added a new field description in Dynamic Game Listing API
1.6.0	2022-07-18	C. Dargahi	Dynamic Game Listing API support for games in different sections
1.6.1	2022-08-23	C. Dargahi	Clarification regarding Universal FreeRound transactions
1.7.0	2023-03-12	J. Redborg	Adding Free to Play mode documentation
1.8.0	2023-03-17	C. Dargahi	Iframe requirements
1.8.1	2023-06-13	C. Dargahi	FreeRound conditions added to request payload for debit and credit

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

3. General

This document describe the requirements SplitThePot has for a successful Seamless Wallet API and game client integration where SplitThePot can provide its games and services to the operator and its platform.

The operator is to provide an endpoint for testing and an endpoint for production together with needed API keys for each environment. If the operator has the requirement to white list any IP addresses, this can be accommodated.

All SplitThePot's games are designed mobile first and are aimed to work on low end devices running on limited bandwidth networks as good as on the high-end devices and fast networks.

Further in this documentation SplitThePot may be referred to as STP, interchangeably.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

4. Client-side Game Integration

SplitThePot will provide URLs for each game that is offered to the operator. The operator's platform is responsible to redirect the player to STP's games when requested by the player by providing an authentication token for the players' session.

The operator has two options for launching the games:

1. (**Preferred**) as a new browser tab or window, which would provide best compatibility and out most user experience for any user device.
2. (*Advise against*) inside an iframe on the operator's site which the operator is then responsible for, making sure that width and height of the iframe are adequate for a good user experience. Requirements for this option is stated in the next section. *Again, this option is not preferred due to inconsistent iframe support by browsers and operating systems, especially on mobile devices*.

4.1 Requirements for iframe solution

The following are requirements that must be met when integrating the game launch with an iframe solution.

General

The iframe HTML element is required to have the following attributes:

- `allow="fullscreen"`, to enable the game inside the iframe to go full screen on the players device, for the best experience.

Mobile devices

1. The game needs to have at least `95lvh`[1]
2. The game needs to have `100lvw`
3. *Again, avoid iframe on mobile devices*

Desktop and Tablet devices

1. The game needs to have at least `95svh`[2]
2. The game needs to have at least `80svw`

4.2 Authentication token

The operator is responsible for handling, validating and maintaining the authentication token for the session's entirety. The minimum duration for which the token is valid should be at least 2 hours.

A better option than fixed duration, is to have a sliding window duration if the operator's platform supports it.

The token must be sent as a query parameter in the URL used to launch games, as described in coming sections.

The authentication token provided should be unique and must be non-deterministic, for e.g. not the ID of the player. It must be at most 32 characters long, and contain only URL friendly characters, to put the least amount of weight on the payloads sent back and forth between clients and backend APIs, for e.g. avoid using JWT tokens. The token must be reusable for the whole duration of its

[1] CSS Values and Units Module Level 4 (w3.org)
[2] CSS Values and Units Module Level 4 (w3.org)

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

validity. It will be used for multiple requests send from STP's APIs or a client reloading a game when opened outside of an iframe.

The token will be sent in the header of requests `x-stp-token`. Read more in following sections describing the API and how this token is utilized.

4.3 Game definition

SplitThePot defines its' games by these three identifiers: *Game ID*, *Game Kind*, *Game Variant*.

- Game ID: is the internal game engine ID that runs the games based on set of rules. This is also the identifier when in data sent in between APIs in integrations. *Note: do not take dependency on this value being unique amongst games, this is an engine ID and many games can share the same engine. For uniqueness, all three identifiers must be used.*
- Game Kind: is a game on the client side utilizing certain game engine (*Game ID*) to run a game with modified set of rules that conform to the engines allowed rule set.
- Game Variant: is a visual variant (or a skin) that is used upon a *Game Kind* to give the game certain visual flavor based on preferences.

The relationship between these can be explained as follows:

- *Game Variant's relationship to Game Kind: many to one.*
- *Game Kind's relationship to Game ID: many to one.*

4.4 Game URL

The URL of each game is structured this way:

https://splitthepot.games/{`operator`}/{`gameKind`}/{`gameVariant`}?token={`token`}

`operator` is unique for every operator and will be provided by STP to the operator as part of the integration process.
`gameKind` describes a kind of game, a `gameKind` always corresponds to a single `gameId` (which is internal and explained further in this document). A `gameKind` can have multiple `gameVariants`.
`gameVariant` will be unique for each kind of game, providing different visual or functional variations for the same kind of game running same fundamental logic and engine.
`token` is to be created by the operator as described previously.

4.5 Free to Play mode

The URL structure of each game is fundamentally the same as in regular mode (see 4.4), except that in Free to Play mode the token must be the predefined value as below:

https://splitthepot.games/{`operator`}/{`gameKind`}/{`gameVariant`}?token=freetoplay

`operator, gameKind, gameVariant` see 4.4

The currency used in Free to Play mode is 'FUN', a made-up currency with predefined settings and limits, which are not adjustable by the operator. Bets placed in Free to Play mode will only be tracked and accounted for in STP's system, therefore it is not a subject for the rest of the API integration described in this document. All games may not support Free to Play mode.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

4.6 Dynamic Game Listing API

The configuration of the games available to the operator must be fetched dynamically via the described API. Each operator is provided with its own `key` that must be sent as part of the request to identify the operator's configuration. The `key` will be provided to the operator or platform provider after the configuration has been setup. The operator is required to use healthy caching as described below, but more importantly, it's also required to check or re-fetch for new configuration after the cache has expired. For example, games may be reordered, game icons changed or a game may be put in maintenance. The operator is then required to update how it lists or visualizes the games and its icons as described in the following sections.

All responses from the API's endpoint will include

1. `Etag`[3] and/or `Last-Modified`[4] header, these values must be used by the operator's platform in subsequent requests (by using `If-None-Match`[5] and/or `If-Modified-Since`[6]) to avoid transferring large payloads in response bodies over network when no changes have been made to the configuration since last. It's also advised to do `HEAD` requests for checking if new version of the configuration is available before performing a `GET` request. Making calls from a client browser, will automatically honor these values.
2. `Cache-Control`[7], the value of `max-age` is provided and must be honored by the caller. Making calls from a client browser, will automatically honor these values.

Note: this API is not designed to be called frequently and the nature of the data is such that it will change infrequently. Use healthy caching, conditional headers, methods described in this whole section and apply good practice.

Note: the game URLs returned by this API, may not enable loading of the games before the backend integration has been completed.

[3] ETag - HTTP | MDN (mozilla.org)
[4] Last-Modified - HTTP | MDN (mozilla.org)
[5] If-None-Match - HTTP | MDN (mozilla.org)
[6] If-Modified-Since - HTTP | MDN (mozilla.org)
[7] Cache-Control - HTTP | MDN (mozilla.org)

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

Game Icons

The game images provide a choice between a square game icon or a wide game icon based on the preference of the operator and its site, and to be able to adhere to most common icon formats. The default image for each aspect ratio is given in x3 Retina resolution. Image resolutions for x1 and x2 are also provided for the same image. It's advised to use all these versions by setting the `srcset`[8] attribute when using the images on `<image>` tags, or by using @media queries in CSS.

Some game sections may only provide game icons in the SVG format. SVG icons can be used directly on the operator's site. The operator is allowed to apply a single fill color to the SVG icon to blend it into the site's overall design language and color scheme. The size of the SVG icon is allowed to be changed for the same reason, as long as the width and height ratio of the icon is kept intact.

Game sections

There are multiple sections in the API response containing game configurations for different purposes or targets. This can include a list of games that is targeting and supporting Opera Mini Extreme mode or games that are required to be visible only in a specific section of the operator's site. The following are descriptions of the sections currently provided and their intended use and presentation:

- `games` – the games provided in this section are general games that are intended to be listed together under the same page in the site's *game lists or lobby*.
- `directGames` – the games provided in this section are required to be prominently made available to the players on the *site's main navigation* section. These games may or may not be available under the `games` section, depending on the game and its business requirements.
    - The links on the *site's main navigation* section must directly open the game
    - Using the game icon adjacent to the link title is a requirement
- `extremeGames` – the games provided in this section are specifically made to for Opera Mini Extreme mode, the games are to be listed together under a single page in the operator's site designed for Opera Mini Extreme

API Base URL

https://splitthepot.games/gamescontent

Get configuration

Endpoint

Method: GET
Endpoint: /operatorgames/{`key`}

Request Headers

`If-None-Match`: See MDN Web Docs
`If-Modified-Since`: See MDN web Docs

[8] Responsive images - Learn web development | MDN (mozilla.org)

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

Response Headers

`Etag`: See MDN Web Docs
`Last-Modified`: See MDN Web Docs
`Cache-Control`: See MDN Web Docs

Response 200 OK

```
{
    "updated": dateTime,
    "games": [
        {
            "gameId": string,
            "gameKind": string,
            "gameVariant": string,
            "title": string,
            "enabled": boolean,
            "maintenance": boolean,
            "url": string,
            "freeToPlayUrl": string,
            "bethistoryUrl": "string",
            "squareX3": {
                "url": string,
                "formats": {
                    "x1": {
                        "url": string,
                        …
                    },
                    "x2": {
                        "url": string,
                        …
                    }
                },
                …
            },
            "wideX3": {
                "url": string,
                "formats": {
                    "x1": {
                        "url": string,
                        …
                    },
                    "x2": {
                        "url": string,
                        …
                    }
                }
                …
            }
        }
    ],
    "directGames": [
        {
            "gameId": string,
            "gameKind": string,
            "gameVariant": string,
            "title": string,
            "enabled": boolean,
            "maintenance": boolean,
            "url": string,
            "freeToPlayUrl": string,
            "bethistoryUrl": "string",
            "icon": {
                "url": string,
            }
        }
    ]
    "extremeGames": [
        {
            "gameId": string,
            "gameKind": string,
            "gameVariant": string,
            "title": string,
            "enabled": boolean,
            "maintenance": boolean,
            "url": string,
            "bethistoryUrl": "string",
            "squareX3": {
                "url": string,
                "formats": {
```

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

```
                "x1": {
                    "url": string,
                    …
                },
                "x2": {
                    "url": string,
                    …
                }
            },
            …
        },
        "wideX3": {
            "url": string,
            "formats": {
                "x1": {
                    "url": string,
                    …
                },
                "x2": {
                    "url": string,
                    …
                }
            }
            …
        }
    }
  ]
}
```

Status Codes

200 – OK (if no conditional headers are present, or if conditional headers are matching)

304 – Configuration has not changed (only when conditional headers are present)

404 – Key is incorrect, no configuration can be found

References

`updated` timestamp of when the configuration was last updated. (See Timestamps)

`games` array of game configuration

- `gameId` (See Game definition)
- `gameKind` (See Game definition)
- `gameVariant` (See Game definition)
- `title` is the title of game that must be displayed on the operator's site adjacent to the game icon. If the operator chooses not to show any title this can be omitted, but when a title is to be displayed, it must be this value.
- `enabled` a boolean indicating if the game has been enabled. When this value is false, the game should not be displayed on the operator's site to the players.
- `maintenance` a boolean indicating if the game is undergoing maintenance. When this value is true, the game should be visually marked as being in maintenance (e.g., by adding low opacity indicating disabled), but not removed from the site. Clicks on the game icon or title should not redirect players to launch the game.
- `url` is the launch URL of the game. The operator needs to replace the `{token}` placeholder as described in previous sections (See Authentication token)
- `freeToPlayUrl` is the launch URL of the game in Free to Play mode.
- `bethistoryUrl` is the URL template to a single bet in the history of the game, tied to a single player. The operator needs to replace the `{token}` placeholder as described in previous sections (See Authentication token). The operator also needs to replace the `{gamePlayId}` placeholder that references a single bet uniquely, described in the coming sections (See API).
- `squareX3` describes the URL for fetching the x3 retina game icon image with the aspect ratio of 1:1. Under the `format` property, x1 and x2 versions of the game icon can also be found with the

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

same aspect ratio. (This object may have more properties such as width and height etc. It's advised against taking dependencies on those as they may change)

`wideX3` describes the URL for fetching the x3 retina game icon image with the aspect ratio of 16:9. Under the `format` property, x1 and x2 versions of the game icon can also be found with the same aspect ratio. (This object may have more properties such as width and height etc. It's advised against taking dependencies on those as they may change).

`directGames` array of game configuration

`gameId` (See Game definition)
`gameKind` (See Game definition)
`gameVariant` (See Game definition)
`title` is the title of game that must be displayed on the operator's site adjacent to the game icon. If the operator chooses not to show any title this can be omitted, but when a title is to be displayed, it must be this value.
`enabled` a boolean indicating if the game has been enabled. When this value is false, the game should not be displayed on the operator's site to the players.
`maintenance` a boolean indicating if the game is undergoing maintenance. When this value is true, the game should be visually marked as being in maintenance (e.g., by adding low opacity indicating disabled), but not removed from the site. Clicks on the game icon or title should not redirect players to launch the game.
`url` is the launch URL of the game. The operator needs to replace the `{token}` placeholder as described in previous sections (See Authentication token)
`freeToPlayUrl` is the launch URL of the game in Free to Play mode.
`bethistoryUrl` is the URL template to a single bet in the history of the game, tied to a single player. The operator needs to replace the `{token}` placeholder as described in previous sections (See Authentication token). The operator also needs to replace the `{gamePlayId}` placeholder that references a single bet uniquely, described in the coming sections (See API).
`icon` describes the URL for fetching the SVG game icon. (This object may have more properties such as width and height etc. It's advised against taking dependencies on those as they may change)

`extremeGames` array of game configuration
(See section for `games` property descriptions)

Check for newer version

Endpoint
Method: HEAD
Endpoint: /operatorgames/{`key`}

Request Headers
`If-None-Match`: See MDN Web Docs
`If-Modified-Since`: See MDN web Docs

Response Headers
`ETag`: See MDN Web Docs
`Last-Modified`: See MDN Web Docs
`Cache-Control`: See MDN Web Docs

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

Status Codes

200 – OK (if no conditional headers are present, or if conditional headers are matching)

304 – Configuration has not changed (only when conditional headers are present)

404 – Key is incorrect, no configuration can be found

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

5. API

All API endpoints are to be RESTFUL and will be called with JSON data payload and expect JSON response, with correct `Content-Type` heading. All endpoints are required to be HTTPS endpoints with valid signed certificates.

All API endpoint will be called with a header called `x-stp-api-key` with a value provided by the operator, and is expected to be validated on each call for security reasons.

Account ID

Is expected to (must) be a unique identifier within the operator's platform for each player and remain static for the players entire life time within the systems. The ID will be a means to identify each single player in both the operator's and STP's platform. The Account ID will be used for identifying a player, all its bets and may be used from trouble shooting, reporting, telemetry and other system extensions and features. The ID must also not contain any personal information that could potentially be used to find the players identity outside of the systems.

Currency

All currencies are provided and expected to be in ISO 4217 code format. For example, EUR or USD.

Amount

Amounts are provided and expected in smallest denominator of the given currency. For example, in EUR the smallest denominator is cents, 1 EUR = 100 cents. Amount must be an integer and contain no decimals. To clarify if the player stakes 2.5 EUR, then the following `stake` property will be sent to the Debit (Bet) endpoint: `"stake": { "amount": 250, "currency": "EUR" }`

Idempotency

All endpoints are expected to be idempotent and only accept a successful transaction once. If a transaction with an ID fails, it's expected that if/when retried, the transaction is to be either accepted or return 304 status code.

Transaction

All transactions are expected to adhere to ACID principle. Transaction IDs are used to identify all transactions made from STP's API to the operator's platform.

Timestamps

Expected to be in ISO 8601 format, example `2021-09-17T13:05:29.678Z`

Transaction on loss

Optionally as an opt-in feature, operators can choose if the API should report losing bets by calling 5.3 Credit (Win) endpoint with `payout` and `amount` of 0. From a performance point of view, it's *recommended* to not opt-in. If the operator's platform has such requirements, then this option is available.

Universal FreeRounds transaction support

In the default setup *Universal FreeRound* bet's debit transaction is only made against STP's internal *FreeRounds Wallet*, with no API call to the operator's API.

Optionally as an opt-in feature, operators can choose if a debit call should be made to the operator's API as described in 5.2 Debit (Bet) with `isFreeRound` set to `true`. From a performance point of view, it's *recommended* to not opt-in. If the operator's platform has such requirements, then this option is

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

available. If the operator opt-in to this feature, it's expected to allow 0 stake transactions when `isFreeRound` flag is set.

If a FreeRound bet has been won and has a `payout` with `amount` greater than 0 (or if the operator has opt-in to Transaction on loss) a credit call will be made with `isFreeRound` set to `true` and the operator is expected to apply the transaction.

All debit transaction or stake values for Universal FreeRounds in the operator's system must accept and have the value 0.

For more in-depth information regarding Universal FreeRounds and integrations, advise the separate documentation.

Latency

The operators API endpoint are expected to have no more than 20ms in execution latency. Taking into account network latencies, this *should never* go beyond 150ms. Average total roundtrip latency is expected to be maximum 50ms. This point is open to some level of discussion depending on operator's server location and the network infrastructure available at that location.

Hosting

SplitThePot's services all run in Azure Cloud environment and the aim is to have its globally scalable systems deployed as near the end users and operators' location as possible, provided that Azure has available regional infrastructure available there. Currently for the African continent, services are deployed in South Africa (Johannesburg). Depending on the operator's platform location, a different service location might be used by SplitThePot to give the lowest latency and provide the best user experience.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

5.1 Authorization and player details

This endpoint will be called to authenticate the player and get the players details.

Endpoint

Method: GET

Endpoint: /player

Request Headers

`x-stp-api-key`: string

`x-stp-token`: string

`x-stp-version`: 1.0.0

Response 200 OK

```
{
    "accountId": string,
    "displayName": string,
    "balance": { "amount":  int, "currency": string, "updatedOn": dateTime },
}
```

Status Codes

200 – OK

401 – Token is not valid

429 – Too Many Requests (preferably with a Retry-After header)

References

`accountId` is the players account ID in the operator's platform.
`displayName` is up to the operator to format properly and is meant for visualizing the player in high-scores or in multiplayer games where it's required. Suggested format is the first name and the first letter of the last name in case the operator's platform does not already have an account field for alias or display name. Example Matthew S. **Note**: *this **must not** be an e-mail address or similar that can uniquely identify a player and/or reveals its personal information.*
`balance` is the player's current balance. `updatedOn` indicates the date and time which the balance was last updated.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

5.2 Debit (Bet)

This endpoint will be called to debit player wallet of the amount staked in bet.

Endpoint

Method: POST
Endpoint: /wallet/{`accountId`}/debit

Request Headers

`x-stp-api-key`: string
`x-stp-token`: string
`x-stp-version`: 1.0.0

Request

```
{
    "transactionId": string,
    "gameId": string,
    "gameKind": string,
    "gameVariant": string,
    "gamePlayId": string,
    "stake": { "amount": int, "currency": string},
    "isFreeRound": boolean | undefined
    "freeRoundCondition": { "campaign": string, "conversion": { "value": int, "currency": string } } | undefined
}
```

Response 200 OK

```
{
    "operatorTransactionId": string,
    "balance": { "amount": int, "currency": string, "updatedOn": dateTime },
    "timestamp": dateTime
}
```

Status Codes

200 – OK

304 – Not modified, transaction has already been applied

400 – Bad request, payload has wrong format or does not pass validation. Provide explanation in response body.

401 – Token is not valid

402 – Insufficient funds

429 – Too Many Requests (preferably with a Retry-After header)

References

`transactionId` is STP's internal transaction ID and will be unique for every bet a player places.
`gameId` is STP's internal game ID.
`gameKind` is the game game kind player is betting on.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

`gameVariant` is the game variant player is betting on.
`gamePlayId` is STP's internal unique ID which is unique across all bets and is the reference to the bet itself. For the same bet this value will be consistent across all API calls.
`stake` is the amount and currency the player has staked on this bet.
`isFreeRound` when the value is true, it indicates that the player is betting a FreeRound. *(Note: that the stake amount will be 0 when this flag is set to true)*
`freeRoundCondition` contains information about the FreeRound and is only set when `isFreeRound` value is true. The arbitrary name of the campaign it belongs to and its conversion value.

`operatorTransactionId` is the operator's internal transaction ID, and is used only for tracking and troubleshooting purposes.
`balance` is the player's balance after applying the transaction. `updatedOn` indicates the date and time which the balance was last updated.
`timestamp` is the date and time at which this transaction was performed.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

5.3 Credit (Win)

Endpoint

Method: POST

Endpoint: /wallet/{`accountId`}/credit

Request Headers

`x-stp-api-key`: string

`x-stp-version`: 1.0.0

Request

```
{
    "transactionId": string,
    "gameId": string,
    "gameKind": string,
    "gameVariant": string,
    "gamePlayId": string,
    "payout": { "amount": int, "currency": string },
    "payoutFactor": decimal,
    "isFreeRound": boolean | undefined
    "freeRoundCondition": { "campaign": string, "conversion": { "value": int, "currency": string } } | undefined
}
```

Response 200 OK

```
{
    "operatorTransactionId": string,
    "balance": { "amount": int, "currency": string, "updatedOn": dateTime },
    "timestamp": dateTime
}
```

Status Codes

200 – OK

304 – Not modified, transaction has already been applied

400 – Bad request, payload has wrong format or does not pass validation. Provide explanation in response body.

429 – Too Many Requests (preferably with a Retry-After header)

References

`transactionId` is STP's internal transaction ID and will be unique for every bet a player places.
`gameId` is STP's internal game ID.
`gameVariant` is the game variant player is betting on.
`gameKind` is the game game kind player is betting on.
`gamePlayId` is STP's internal unique ID which is unique across all bets and is the reference to the bet itself. For the same bet this value will be consistent across all API calls.
`payout` is the amount and currency that the player has won on the bet.
`payoutFactor` is the actual factor that the player has won, with the settings of the bet.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

`isFreeRound` when the value is true, it indicates that the player bet was a FreeRound.
`freeRoundCondition` contains information about the FreeRound and is only set when `isFreeRound` value is true. The arbitrary name of the campaign it belongs to and its conversion value.

`operatorTransactionId` is the operator's internal transaction ID, and is used only for tracking and troubleshooting purposes.
`balance` is the player's balance after applying the transaction. `updatedOn` indicates the date and time which the balance was last updated.
`timestamp` is the date and time at which this transaction was performed.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

5.4 Revert

Endpoint

Method: DELETE

Endpoint: /wallet/{accountId}/{transactionId}

Request Headers

x-stp-api-key: string

x-stp-version: 1.0.0

Response 200 OK

```
{
    "operatorTransactionId": string,
    "balance": { "amount": int, "currency": string, "updatedOn": dateTime },
    "timestamp": dateTime
}
```

Status Codes

200 – OK

304 – Not modified, transaction has already been reverted

404 – Not found, transaction ID to revert could not be found

429 – Too Many Requests (preferably with a Retry-After header)

References

transactionId is STP's internal transaction ID that should also identify the transaction to be reverted on the operator's platform.

operatorTransactionId is the operator's internal transaction ID, and is used only for tracking and troubleshooting purposes.
balance is the player's balance after applying the transaction. updatedOn indicated the date and time which the balance was last updated.
timestamp is the date and time at which this transaction was performed.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

5.5 Get transaction

Endpoint

Method: GET

Endpoint: /wallet/{accountId}/{transactionId}

Request Headers

x-stp-api-key: string

x-stp-version: 1.0.0

Response 200 OK

```
{
    "operatorTransactionId": string,
    "balance": { "amount": int, "currency": string, "updatedOn": dateTime },
    "timestamp": dateTime
}
```

Status Codes

200 – OK

404 – Not found, transaction ID could not be found

429 – Too Many Requests (preferably with a Retry-After header)

References

transactionId is STP's internal transaction ID that should identify the transaction to be retrieved from the operator's platform.

operatorTransactionId is the operator's internal transaction ID, and is used only for tracking and troubleshooting purposes.
balance is the player's current balance. updatedOn indicated the date and time which the balance was last updated.
timestamp is the date and time at which this transaction was performed.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.

5.6 Error handling and retries

In case of errors in API calls due to network, timeout or other types of transient errors, SplitThePot will retry those API calls. The retries are split in two categories explained in the following section.

On bet request pipeline

- Requests that fail while calling Debit endpoint (5.2) will be retried up to 5 times with an exponential back off. If the retries fail, the bet will simply fail and the player can attempt to place a new bet.
- Bets that fail while in a cancellable state, will be cancelled and hence the transaction is to be reverted by calling the Revert endpoint (5.4).
- Requests that fail while calling Credit endpoint (5.3) will be retried up to 5 times with an exponential back off. If the retries fail, the process is explained in the next section.
- Requests that fail while calling Revert endpoint (5.4) will be retried up to 5 times with an exponential back off. If the retries fail, the process is explained in the next section.

Off bet request pipeline

Requests that have failed and left the bets in a non-finalized state will be retried outside the bet request pipelines to enable durable and multiple retry attempts that may span hours or days. This also enables retries that puts non to little load on the service or affect the service performance.

The retries that will be re-attempted and still fail, will after reaching a dynamic threshold trigger an alert. This enables handling of those rare situations manually. *This is the main reason why Credit and Revert endpoints cannot require a token that is tied to a player session, since the retries may exceed its lifespan.*

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of SplitThePot.