Upgrading to brainCloud from PlayFab

Leaderboards and Statistics

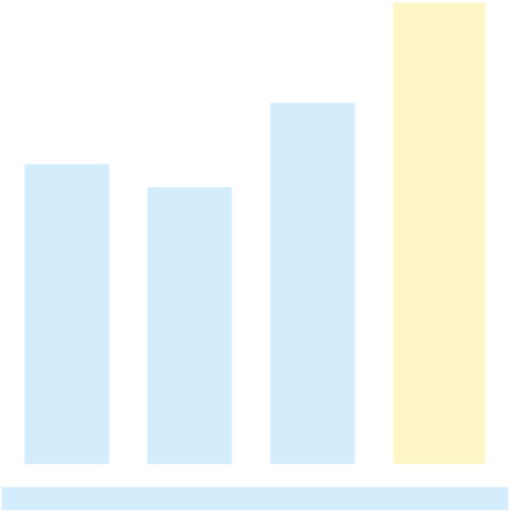


Table of Contents

Introduction

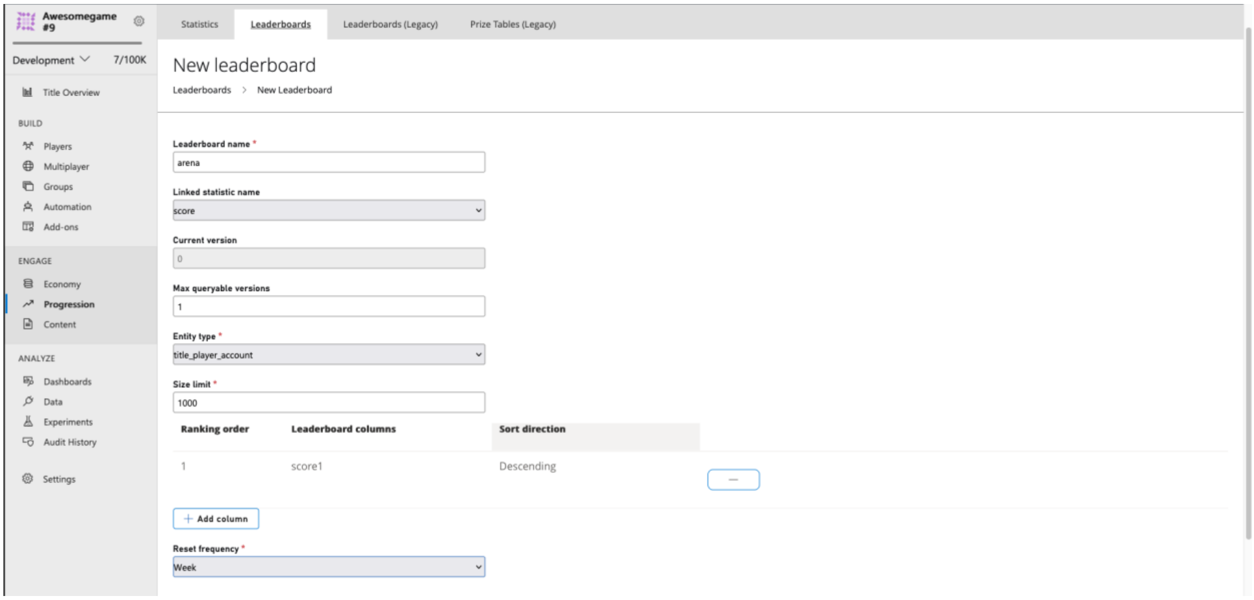
Both PlayFab and brainCloud support leaderboards and statistics, with much of the functionality being similar across both platforms. This guide explains how to set up leaderboards in brainCloud and migrate your existing PlayFab implementation. We'll also highlight additional brainCloud features related to leaderboards and statistics that could ease your migration or support post-migration plans.

Note that PlayFab offers two versions of leaderboards & statistics: the current version updated after the Economy v2 release, and legacy leaderboards. Where possible we will show examples of both versions in comparison to brainCloud.

brainCloud Leaderboards

A key difference between PlayFab and brainCloud leaderboards is their use of statistics. While brainCloud supports statistics as an independent feature (covered later), they are not required to set up leaderboards.

Let's look at a simple leaderboard configuration in PlayFab as an example.



New leaderboard

Leaderboards > New Leaderboard

Leaderboard name *
arena

Linked statistic name
score

Current version
0

Max queryable versions
1

Entity type *
title_player_account

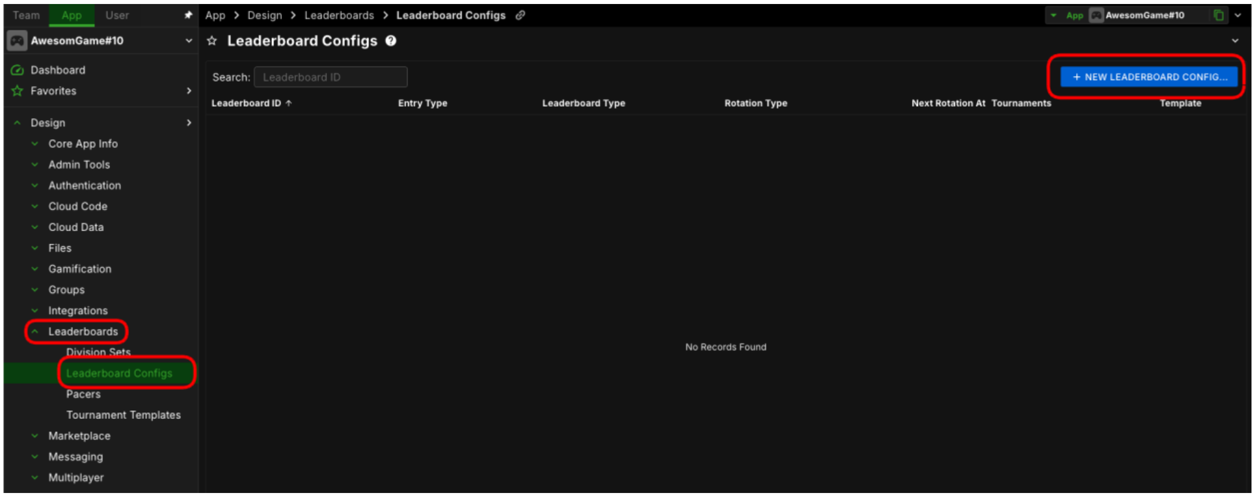
Size limit *
1000

Ranking order	Leaderboard columns	Sort direction
1	score1	Descending

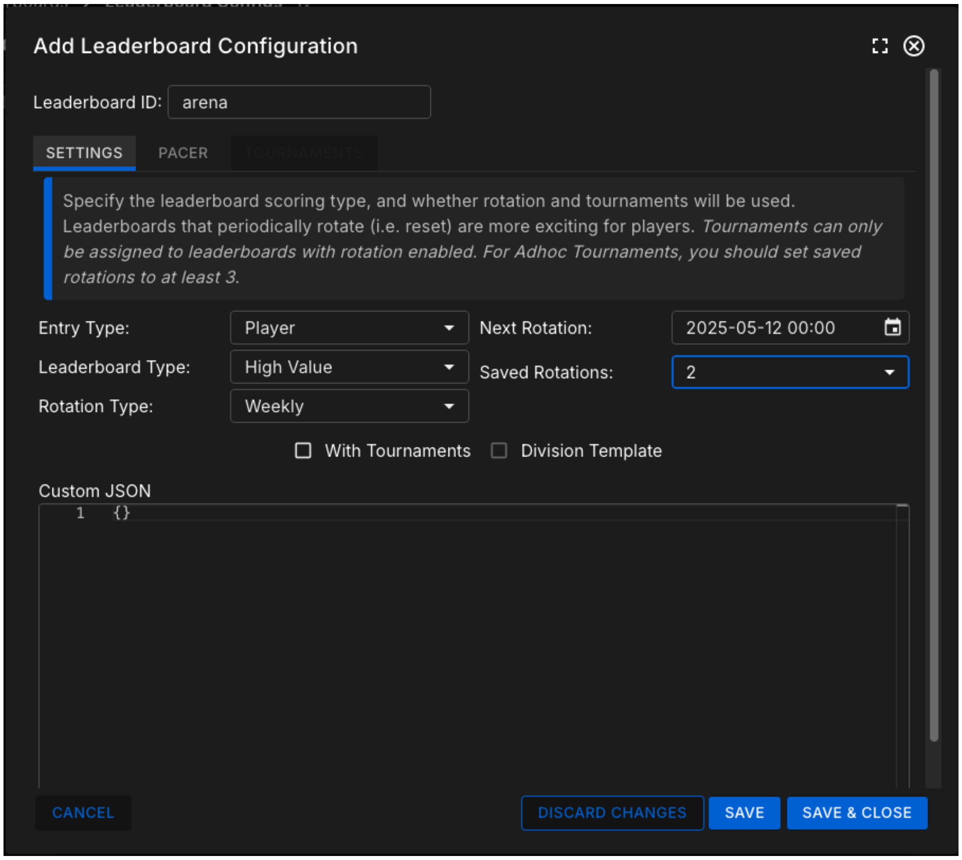
Reset frequency *
Week

This leaderboard is linked to a statistic called “score” with a *Max* aggregation. It’s tied to a player account, uses descending order, and resets automatically every week. Let’s see how to replicate this in brainCloud.

To create a new leaderboard in brainCloud, go to **Design → Leaderboards → Leaderboard Configs** and click **New Leaderboard Config**.



You’ll notice that many options are similar to those in PlayFab’s leaderboard configuration.



Add Leaderboard Configuration

Leaderboard ID: arena

Specify the leaderboard scoring type, and whether rotation and tournaments will be used. Leaderboards that periodically rotate (i.e. reset) are more exciting for players. *Tournaments can only be assigned to leaderboards with rotation enabled. For Adhoc Tournaments, you should set saved rotations to at least 3.*

Entry Type: Player

Leaderboard Type: High Value

Rotation Type: Weekly

Next Rotation: 2025-05-12 00:00

Saved Rotations: 2

☐ With Tournaments ☐ Division Template

Custom JSON

```
{}
```

Let’s take a look at some of these settings:

- **Entry Type**
  This can be set to either the player or a group, similar to PlayFab.

- **Leaderboard Type**
  This matches the aggregation you set on your statistic in PlayFab. In our example, aggregation was set to *Max*, meaning only the highest score is recorded.

- **Rotation Type**
  Options include manual/never, daily, weekly, monthly, or yearly. With brainCloud, you can also specify exactly when the next rotation occurs, allowing resets on chosen weekdays. You can also set the number of Saved Rotations, letting players review previous leaderboard records if needed.

After configuring your leaderboard, click **Save & Close**. For more examples of different leaderboard settings, see the additional configuration options provided here, which may be more specific to your game.

Additional Leaderboard Settings

brainCloud offers extra leaderboard features that may be useful.

Custom JSON

This allows you to add custom configuration to your leaderboard config. You can access and modify this data from the client, providing additional context before posting or listing leaderboard scores.

Pacers

Pacers are helpful when creating or resetting leaderboards. They let you pre-populate leaderboards with preset scores, preventing them from appearing empty at launch or after a reset.

Note

This feature is especially useful during migration. In a passive migration, there may be no existing leaderboard data, which could upset early migrating players if ranks are wiped. With Pacers, you can use an ETL tool to migrate existing scores and populate the initial brainCloud leaderboards, ensuring they don't start empty during migration.

Posting Scores

After recreating your leaderboard in brainCloud, you're ready to post scores.

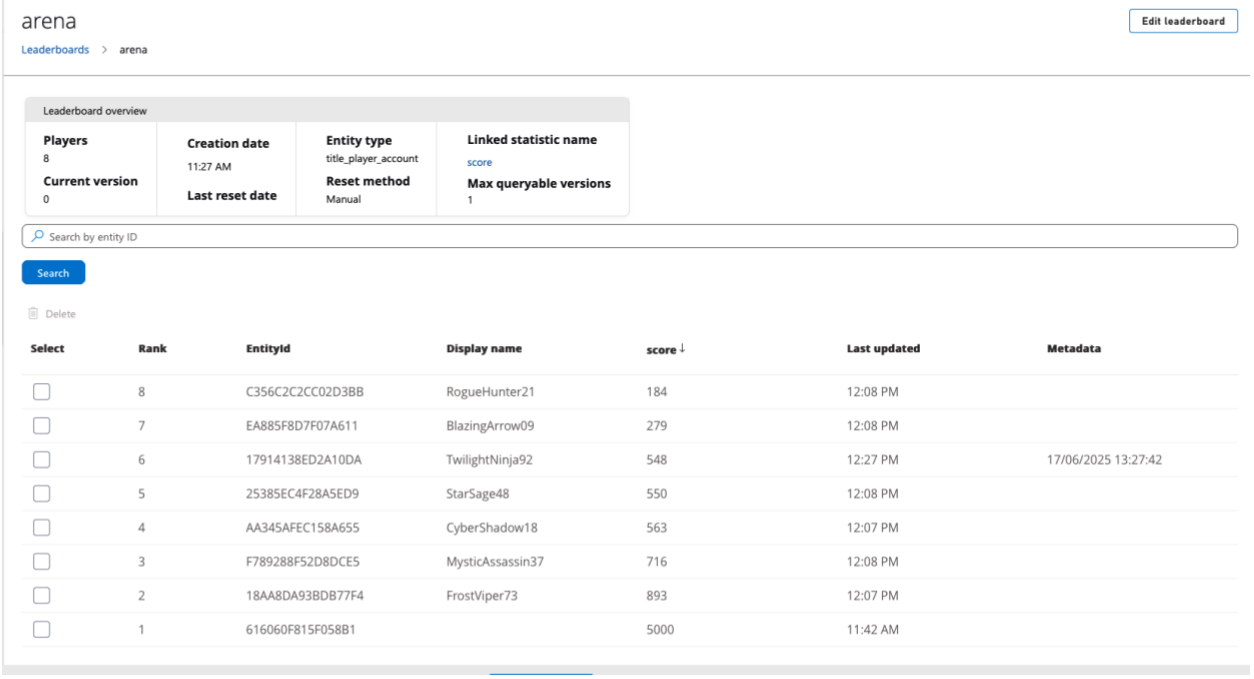
In PlayFab, to update a player's rank on a leaderboard, you must update the connected stat. For v2 leaderboards, this must be done via the "trusted backend," while v1 leaderboards allow updates from the client using the UpdatePlayerStatisticsRequest.

With brainCloud, you can post directly to the leaderboard from the client. The simplest and most common method is the PostScoreToLeaderboard request, which requires only the *leaderboardId* and the *score* to post. Additionally, brainCloud lets players submit a data string with their score, allowing you to attach extra information - similar to the *Metadata* parameter in PlayFab's UpdateStatistics API.

There are several other variations of these requests for posting to leaderboards, which you can review here.

Inspecting Ranks

Just like PlayFab, brainCloud lets you view leaderboard ranks directly from the portal.



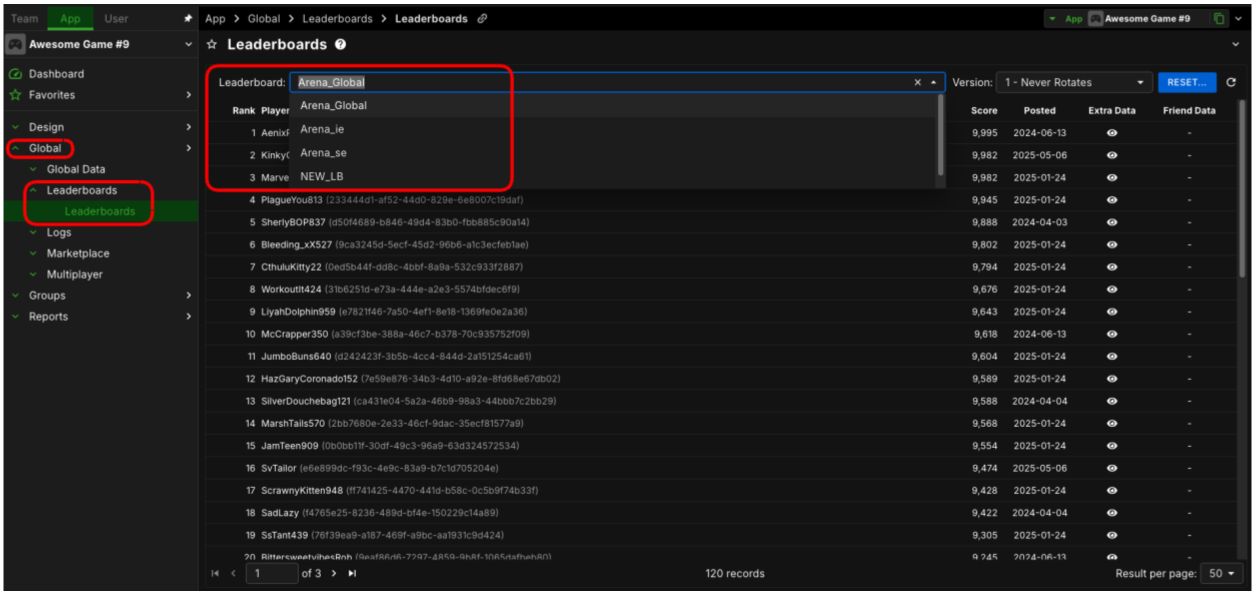
arena

Leaderboard overview

Players 8	**Creation date** 11:27 AM	**Entity type** title_player_account	**Linked statistic name** score
Current version 0	**Last reset date**	**Reset method** Manual	**Max queryable versions** 1

Select	Rank	EntityId	Display name	score ↓	Last updated	Metadata
□	8	C356C2C2CC02D3BB	RogueHunter21	184	12:08 PM	
□	7	EA885F8D7F07A611	BlazingArrow09	279	12:08 PM	
□	6	17914138ED2A10DA	TwilightNinja92	548	12:27 PM	17/06/2025 13:27:42
□	5	25385EC4F28A5ED9	StarSage48	550	12:08 PM	
□	4	AA345AFEC158A655	CyberShadow18	563	12:07 PM	
□	3	F789288F52D8DCE5	MysticAssassin37	716	12:08 PM	
□	2	18AA8DA93BDB77F4	FrostViper73	893	12:07 PM	
□	1	616060F815F058B1		5000	11:42 AM	

In brainCloud, go to **Global → Leaderboards**. At the top of the page, use the drop-down menu to select the leaderboard you want to view.



Leaderboard: Arena_Global

Version: 1 - Never Rotates

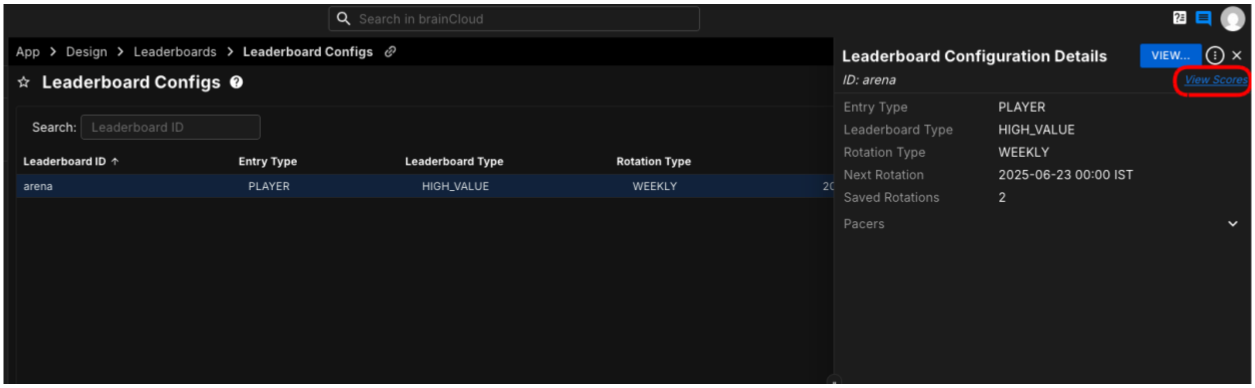
Rank	Player	Score	Posted	Extra Data	Friend Data
1	AenixF	9,995	2024-06-13	👁	-
2	Kinky(	9,982	2025-05-06	👁	-
3	Marve	9,982	2025-01-24	👁	-
4	PlagueYou813 (233444d1-af52-44d0-829e-6e8007c19daf)	9,945	2025-01-24	👁	-
5	SherlyBOP837 (d50f4689-b846-49d4-83b0-fbb885c90a14)	9,888	2024-04-03	👁	-
6	Bleeding_xX527 (9ca3245d-5ecf-45d2-96b6-a1c3ecfeb1ae)	9,802	2025-01-24	👁	-
7	CthuluKitty22 (0ed5b44f-dd8c-4bbf-8a9a-532c933f2887)	9,794	2025-01-24	👁	-
8	WorkoutIt424 (31b6251d-e73a-444e-a2e3-5574bfdec6f9)	9,676	2025-01-24	👁	-
9	LiyahDolphin959 (e7821f46-7a50-4ef1-8e18-1369fe0e2a36)	9,643	2025-01-24	👁	-
10	McCrapper350 (a39cf3be-388a-46c7-b378-70c935752f09)	9,618	2024-06-13	👁	-
11	JumboBuns640 (d242423f-3b5b-4cc4-844d-2a151254ca61)	9,604	2025-01-24	👁	-
12	HazGaryCoronado152 (7e59e876-34b3-4d10-a92e-8fd68e67db02)	9,589	2025-01-24	👁	-
13	SilverDouchebag121 (ca431e04-5a2a-46b9-98a3-44bbb7c2bb29)	9,588	2024-04-04	👁	-
14	MarshTails570 (2bb7680e-2e33-46cf-9dac-35ecf81577a9)	9,568	2025-01-24	👁	-
15	JamTeen909 (0b0bb11f-30df-49c3-96a9-63d324572534)	9,554	2025-01-24	👁	-
16	SvTailor (e6e899dc-f93c-4e9c-83a9-b7c1d705204e)	9,474	2025-05-06	👁	-
17	ScrawnyKitten948 (ff741425-4470-441d-b58c-0c5b9f74b33f)	9,428	2025-01-24	👁	-
18	SadLazy (f4765e25-8236-489d-bf4e-150229c14a89)	9,422	2024-04-04	👁	-
19	SsTant439 (76f39ea9-a187-469f-a9bc-aa1931c9d424)	9,305	2025-01-24	👁	-
20	BittersweetvibesRob (9eaf86d6-7297-4859-9b8f-1065dafbeb80)	9,245	2024-06-13	👁	-

1 of 3

120 records

Result per page: 50

You can also get to this dashboard by clicking on your Leaderboard configuration. You will see a text-link "View Scores" in the top corner of the popup window.



Search: Leaderboard ID

Leaderboard ID ↑	Entry Type	Leaderboard Type	Rotation Type
arena	PLAYER	HIGH_VALUE	WEEKLY

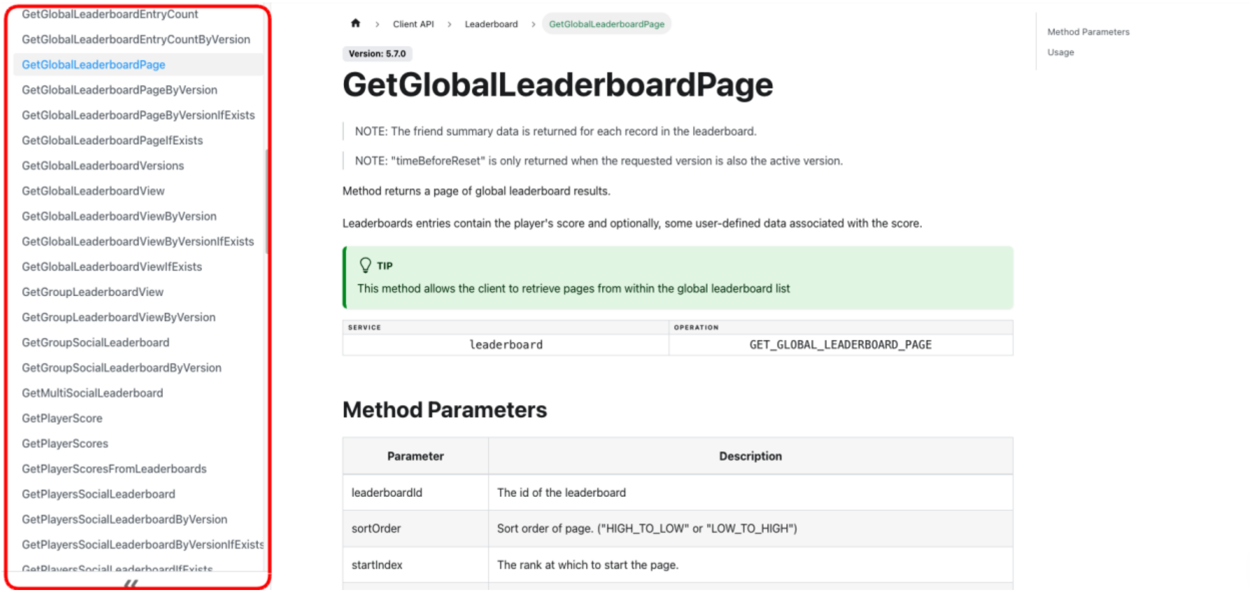
Leaderboard Configuration Details
ID: arena

Entry Type	PLAYER
Leaderboard Type	HIGH_VALUE
Rotation Type	WEEKLY
Next Rotation	2025-06-23 00:00 IST
Saved Rotations	2
Pacers	

Listing Ranks

Next, let's look at how to retrieve leaderboard ranks from the client. In PlayFab, you can use REST calls like GetLeaderboardAroundEntity (for v2) or the GetLeaderboard or GetLeaderboardAroundPlayerRequest. brainCloud provides similar client APIs. The main difference is that brainCloud uses the *leaderboardId* instead of the stat name, similar to PlayFab's v2 APIs.

To replace GetLeaderboardRequest, use GetLeaderboardPage in brainCloud. For GetLeaderboadAroundPlayerRequest, use GetGlobalLeaderboardView request. However, brainCloud also offers a variety of options for retrieving ranks and leaderboard data, depending on your game's needs. You can explore these in the documentation links above.



GetGlobalLeaderboardEntryCount
GetGlobalLeaderboardEntryCountByVersion
GetGlobalLeaderboardPage
GetGlobalLeaderboardPageByVersion
GetGlobalLeaderboardPageByVersionIfExists
GetGlobalLeaderboardPageIfExists
GetGlobalLeaderboardVersions
GetGlobalLeaderboardView
GetGlobalLeaderboardViewByVersion
GetGlobalLeaderboardViewByVersionIfExists
GetGlobalLeaderboardViewIfExists
GetGroupLeaderboardView
GetGroupLeaderboardViewByVersion
GetGroupSocialLeaderboard
GetGroupSocialLeaderboardByVersion
GetMultiSocialLeaderboard
GetPlayerScore
GetPlayerScores
GetPlayerScoresFromLeaderboards
GetPlayersSocialLeaderboard
GetPlayersSocialLeaderboardByVersion
GetPlayersSocialLeaderboardByVersionIfExists
GetPlayersSocialLeaderboardIfExists

Version: 5.7.0

GetGlobalLeaderboardPage

> NOTE: The friend summary data is returned for each record in the leaderboard.

> NOTE: "timeBeforeReset" is only returned when the requested version is also the active version.

Method returns a page of global leaderboard results.

Leaderboards entries contain the player's score and optionally, some user-defined data associated with the score.

TIP
This method allows the client to retrieve pages from within the global leaderboard list

SERVICE	OPERATION
leaderboard	GET_GLOBAL_LEADERBOARD_PAGE

Method Parameters

Parameter	Description
leaderboardId	The id of the leaderboard
sortOrder	Sort order of page. ("HIGH_TO_LOW" or "LOW_TO_HIGH")
startIndex	The rank at which to start the page.

Group Leaderboards

If your game requires migrating group leaderboards, the functionality is similar to what was previously described. Group leaderboards are available only for v2 leaderboards and use the entity ID and *type* parameters to distinguish between player and group entities.

In brainCloud, separate client requests are available for posting and listing group leaderboard ranks. PostScoreToGroupLeaderboard takes a groupId but otherwise works like the previous example. Similarly, GetGroupLeaderboardView can be used to list ranks for group leaderboards.

Friends Leaderboards

While neither PlayFab nor brainCloud offers specific "friends" leaderboards, both platforms let you retrieve ranks for only a player's friends on a given leaderboard. brainCloud's equivalent of GetFriendLeaderboardForEntity is GetSocialLeaderboard.

Tournaments

PlayFab does not have a dedicated Tournament feature, but you can create tournaments using a combination of leaderboards and custom cloud code.

In contrast, brainCloud offers a comprehensive Tournament system that integrates directly with existing leaderboards. The brainCloud Tournament system includes the following components:

Feature	PlayFab (Built-In)	PlayFab (Custom-Built)	brainCloud
Leaderboards	✓	✓	✓
Entry Price	✗	✓ (Custom logic using purchase, VCs, items, etc.)	✓
Bracket Logic	✗	✓ (Possible, but dev must build all logic)	✓
Reward Delivery	✗	✓ (Custom via CloudScript or scheduled tasks)	✓ (Built-in rewards by rank)
Tournament UI	✗	✗ (Requires your own UI, can use APIs as backend)	✓

If your PlayFab title has a custom-built Tournament system that needs migration, consider using brainCloud's built-in features and contact brainCloud support for guidance on migrating your implementation.

Statistics

Statistics serve various purposes on both platforms. Developers typically use them to:

- **Display progress**
  UI (show XP, wins, kills, etc)

- **Check for achievements or rewards**
  (e.g., 100 wins unlocks a badge)

- **To drive matchmaking**
  (e.g., MMR or Elo as a statistic)

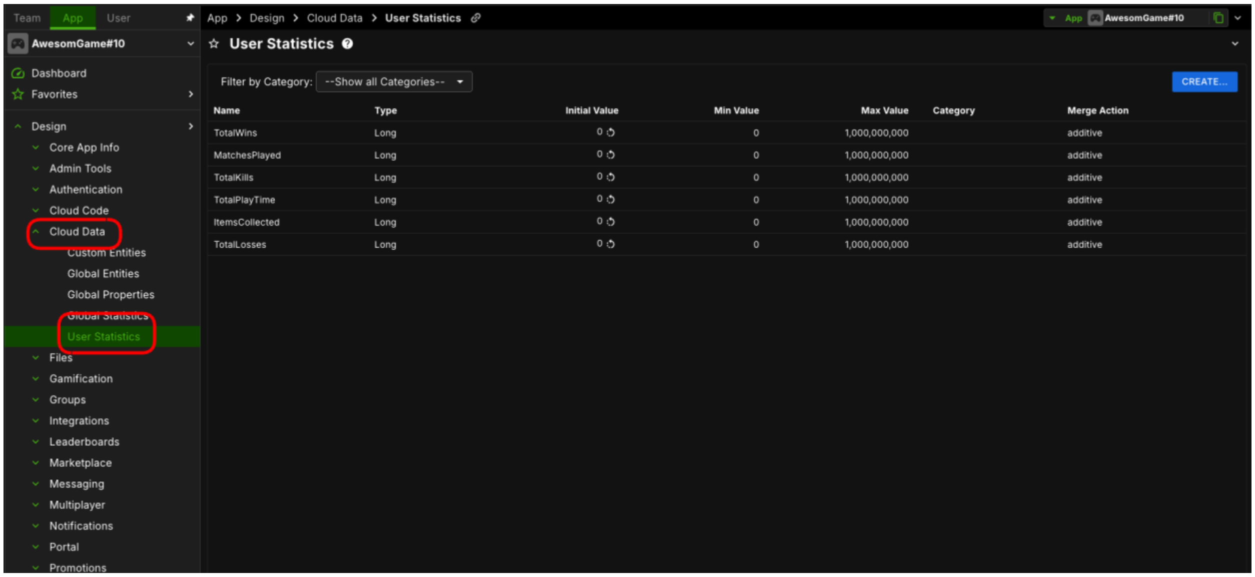
- **Leaderboards**

The main difference between brainCloud and PlayFab is that brainCloud leaderboards do not require statistics, as discussed earlier. Otherwise, statistics operate similarly on both platforms.

User Statistics

First, you need to create statistics through the portal before you can update or modify them. Here's how to create statistics in the brainCloud portal:

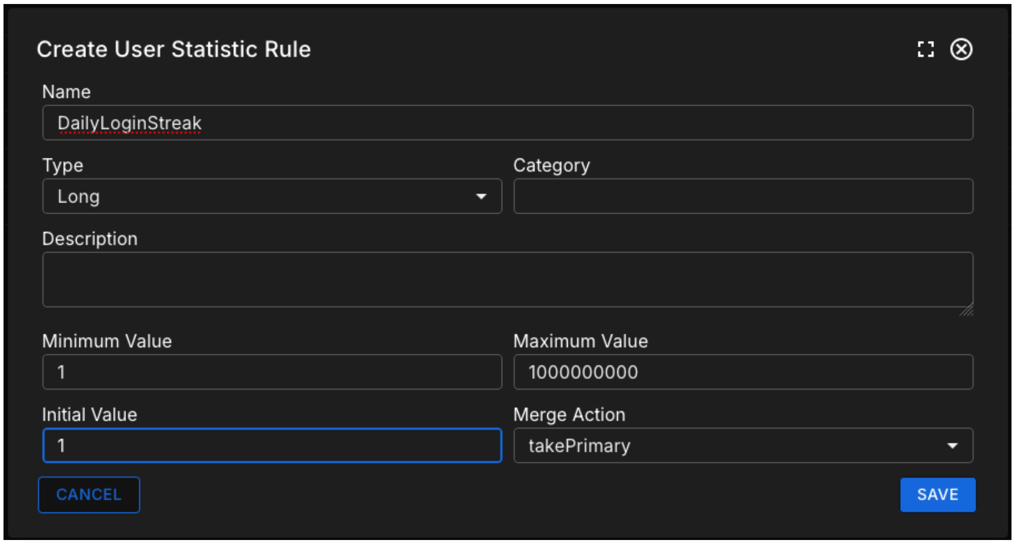
From the side menu, go to **Cloud Data → User Statistics**.



Filter by Category: --Show all Categories--

Name	Type	Initial Value	Min Value	Max Value	Category	Merge Action
TotalWins	Long	0	0	1,000,000,000		additive
MatchesPlayed	Long	0	0	1,000,000,000		additive
TotalKills	Long	0	0	1,000,000,000		additive
TotalPlayTime	Long	0	0	1,000,000,000		additive
ItemsCollected	Long	0	0	1,000,000,000		additive
TotalLosses	Long	0	0	1,000,000,000		additive

Click the **Create** button in the top-right corner of the dashboard to add a new statistic.



Create User Statistic Rule

Name: DailyLoginStreak

Type: Long

Category:

Description:

Minimum Value: 1

Maximum Value: 1000000000

Initial Value: 1

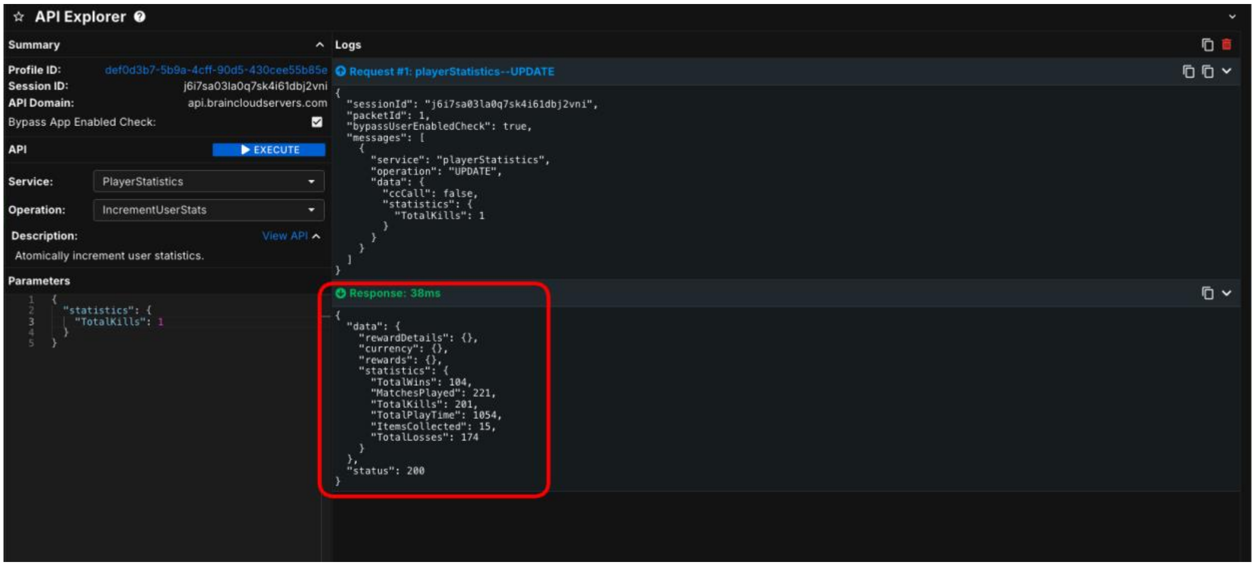
Merge Action: takePrimary

Most configuration parameters for statistics are self-explanatory (see more details here). For a basic setup, we use a *Long* number type, with both the minimum and initial value set to 1, since players will always have a login streak of at least one (the day they log in) in this example.

Reading Statistics

After updating and modifying statistics, the next step is retrieving those values on the client to update the UI. In PlayFab, you use GetPlayerStatistics (for v1) or GetStatistics (for v2). In brainCloud, you can read user statistics on the client with the ReadAllUserStats request, which returns all stats. If you only need specific values, use ReadUserStatsSubset to read selected statistics.

You may also reduce unnecessary API calls during migration by checking where statistic summaries are returned automatically. For example, with the IncrementUserStats response, even if only one statistic is updated, the current state of all statistics is returned.



☆ API Explorer

Summary

Profile ID: def0d3b7-5b9a-4cff-90d5-430cee55b85e
Session ID: j6i7sa03la0q7sk4i61dbj2vni
API Domain: api.braincloudservers.com
Bypass App Enabled Check: ☑

API ▶ EXECUTE

Service: PlayerStatistics
Operation: IncrementUserStats

Description: View API
Atomically increment user statistics.

Parameters

```
{
  "statistics": {
    "TotalKills": 1
  }
}
```

Logs

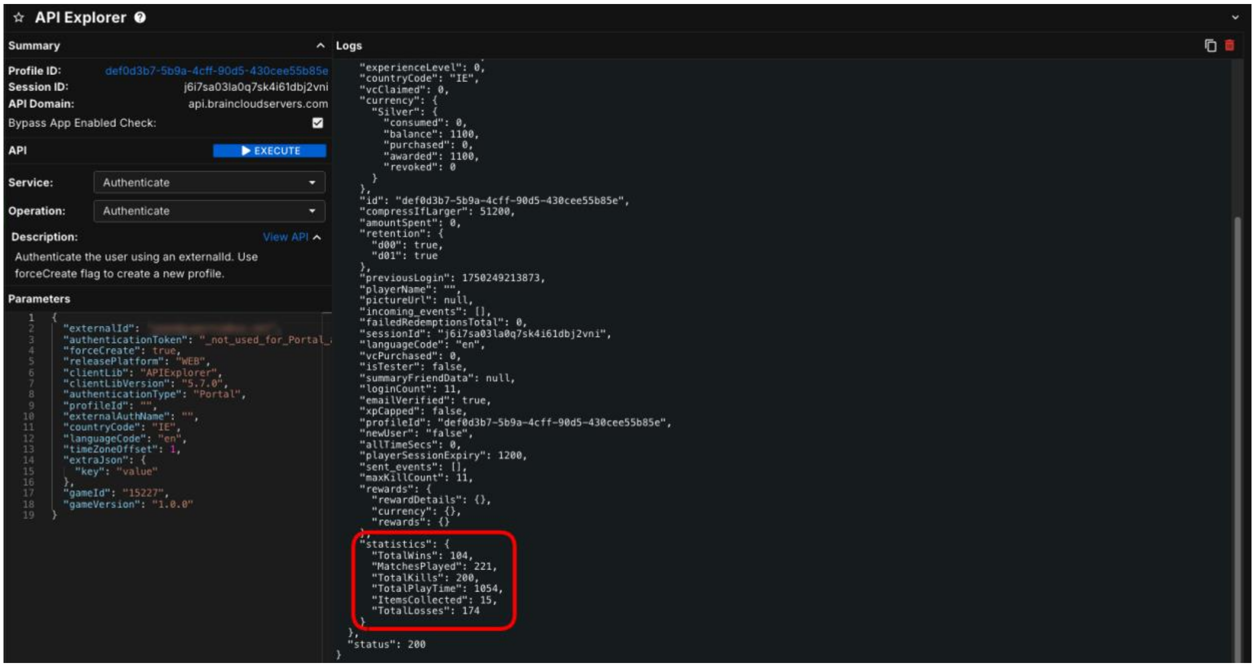
Request #1: playerStatistics--UPDATE

```
{
  "sessionId": "j6i7sa03la0q7sk4i61dbj2vni",
  "packetId": 1,
  "bypassUserEnabledCheck": true,
  "messages": [
    {
      "service": "playerStatistics",
      "operation": "UPDATE",
      "data": {
        "ccCall": false,
        "statistics": {
          "TotalKills": 1
        }
      }
    }
  ]
}
```

Response: 38ms

```
{
  "data": {
    "rewardDetails": {},
    "currency": {},
    "rewards": {},
    "statistics": {
      "TotalWins": 104,
      "MatchesPlayed": 221,
      "TotalKills": 201,
      "TotalPlayTime": 1054,
      "ItemsCollected": 15,
      "TotalLosses": 174
    }
  },
  "status": 200
}
```

Also, whenever a user authenticates, the full statistics summary is included in the authentication response. You can view this using the API Explorer (see guide here <link> for using API Explorer), for example.



☆ API Explorer

Summary

Profile ID: def0d3b7-5b9a-4cff-90d5-430cee55b85e
Session ID: j6i7sa03la0q7sk4i61dbj2vni
API Domain: api.braincloudservers.com
Bypass App Enabled Check: ☑

API ▶ EXECUTE

Service: Authenticate
Operation: Authenticate

Description: View API
Authenticate the user using an externalId. Use forceCreate flag to create a new profile.

Parameters

```
{
  "externalId": [redacted]
  "authenticationToken": "_not_used_for_Portal_
  "forceCreate": true,
  "releasePlatform": "WEB",
  "clientLib": "APIExplorer",
  "clientLibVersion": "5.7.0",
  "authenticationType": "Portal",
  "profileId": "",
  "externalAuthName": "",
  "countryCode": "IE",
  "languageCode": "en",
  "timeZoneOffset": 1,
  "extraJson": {
    "key": "value"
  },
  "gameId": "15227",
  "gameVersion": "1.0.0"
}
```

Logs

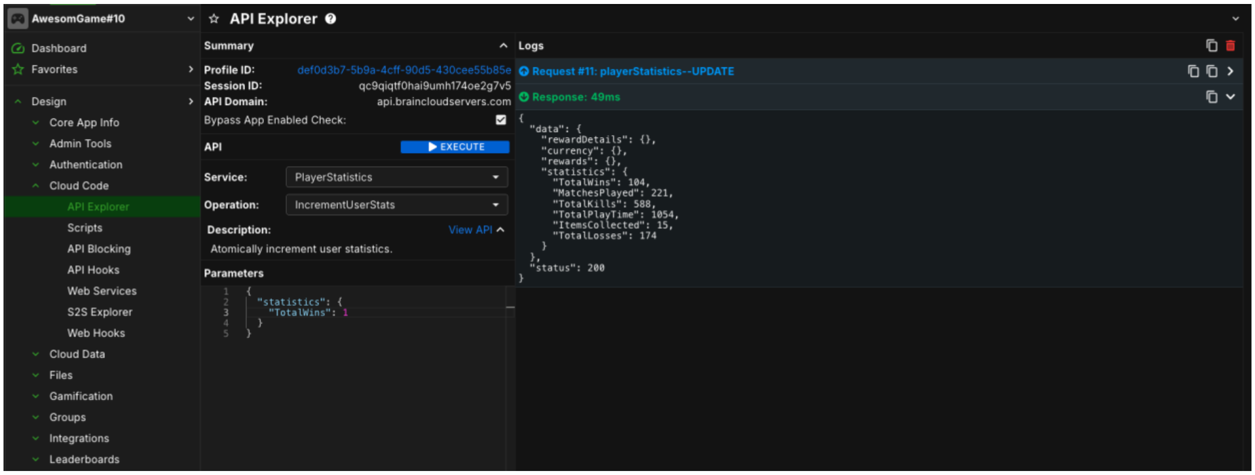
```
    "experienceLevel": 0,
    "countryCode": "IE",
    "vcClaimed": 0,
    "currency": {
      "Silver": {
        "consumed": 0,
        "balance": 1100,
        "purchased": 0,
        "awarded": 1100,
        "revoked": 0
      }
    },
    "id": "def0d3b7-5b9a-4cff-90d5-430cee55b85e",
    "compressIfLarger": 51200,
    "amountSpent": 0,
    "retention": {
      "d00": true,
      "d01": true
    },
    "previousLogin": 1750249213873,
    "playerName": "",
    "pictureUrl": null,
    "incoming_events": [],
    "failedRedemptionsTotal": 0,
    "sessionId": "j6i7sa03la0q7sk4i61dbj2vni",
    "languageCode": "en",
    "vcPurchased": 0,
    "isTester": false,
    "summaryFriendData": null,
    "loginCount": 11,
    "emailVerified": true,
    "xpCapped": false,
    "profileId": "def0d3b7-5b9a-4cff-90d5-430cee55b85e",
    "newUser": "false",
    "allTimeSecs": 0,
    "playerSessionExpiry": 1200,
    "sent_events": [],
    "maxKillCount": 11,
    "rewards": {
      "rewardDetails": {},
      "currency": {},
      "rewards": {}
    },
    "statistics": {
      "TotalWins": 104,
      "MatchesPlayed": 221,
      "TotalKills": 200,
      "TotalPlayTime": 1054,
      "ItemsCollected": 15,
      "TotalLosses": 174
    }
  },
  "status": 200
}
```

Keep in mind that brainCloud charges you per API call, so aim to minimize backend requests to help reduce your overall costs.

Updating Statistics

In brainCloud, you update or set statistic values using the IncrementUserStats request. This works similarly to PlayFab's UpdateStatistics request, but brainCloud offers more complex aggregation options such as increment, decrement, reset, limits, etc. You can learn more about statistics grammar here.

You can test this API by running the request in the API Explorer.



Profile ID: def0d3b7-5b9a-4cff-90d5-430cee55b85e
Session ID: qc9qiqtf0hai9umh174oe2g7v5
API Domain: api.braincloudservers.com
Bypass App Enabled Check: ☑

API

Service: PlayerStatistics
Operation: IncrementUserStats
Description: Atomically increment user statistics.

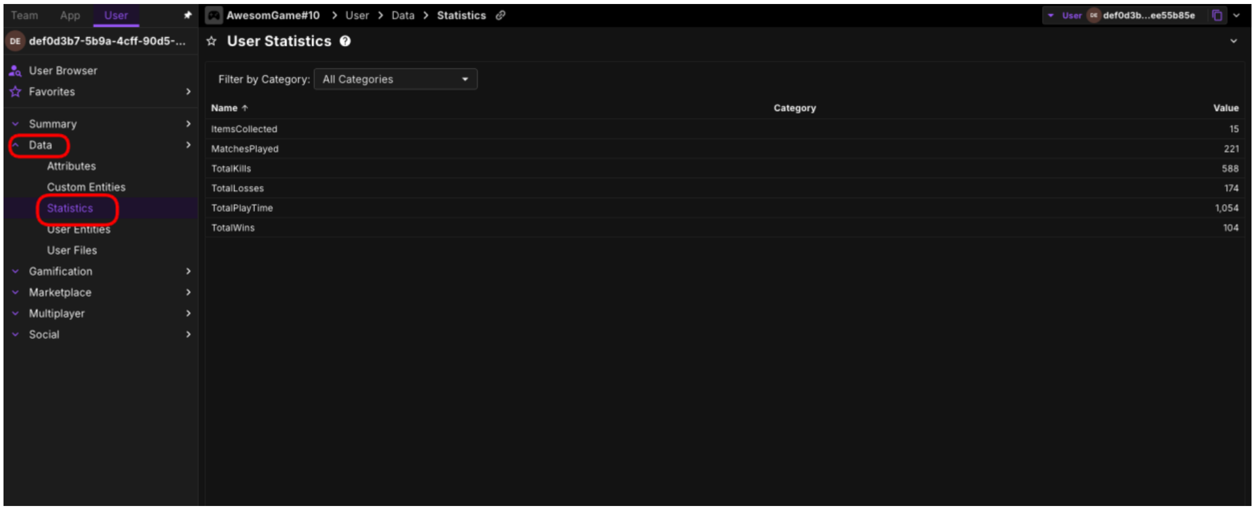
Parameters

```
{
  "statistics": {
    "TotalWins": 1
  }
}
```

Request #11: playerStatistics--UPDATE
Response: 49ms

```
{
  "data": {
    "rewardDetails": {},
    "currency": {},
    "rewards": {},
    "statistics": {
      "TotalWins": 104,
      "MatchesPlayed": 221,
      "TotalKills": 588,
      "TotalPlayTime": 1054,
      "ItemsCollected": 15,
      "TotalLosses": 174
    }
  },
  "status": 200
}
```

You can also view player statistics by selecting **Data → Statistics** when viewing a user's account in the portal.



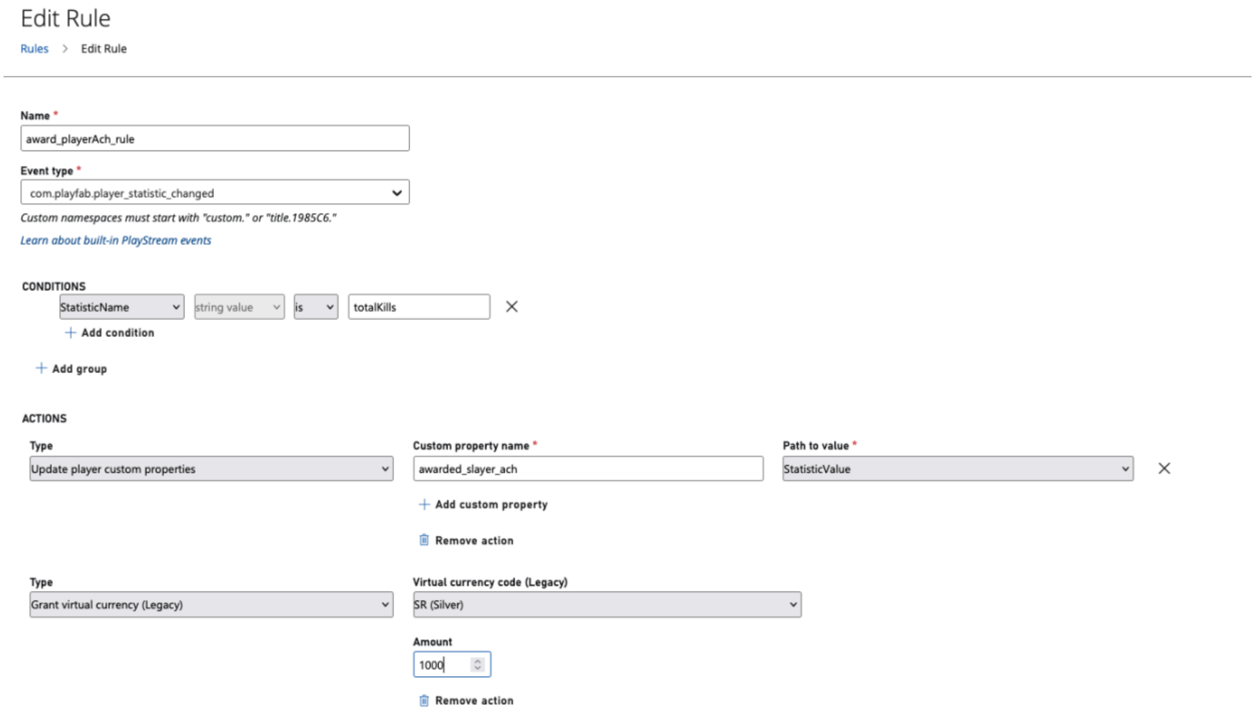
Filter by Category: All Categories

Name ↑	Category	Value
ItemsCollected		15
MatchesPlayed		221
TotalKills		588
TotalLosses		174
TotalPlayTime		1,054
TotalWins		104

Statistic Rules

PlayFab Rules offer a useful feature, letting the backend trigger events dynamically based on statistics. If your game uses this system triggered by statistics, here is how you can achieve similar functionality in brainCloud.

For example, suppose a rule triggers when the *totalKills* statistic reaches 200 and sets a custom property to track an achievement along with granting the player 1,000 silver.



Edit Rule

Rules > Edit Rule

Name *: award_playerAch_rule

Event type *: com.playfab.player_statistic_changed

Custom namespaces must start with "custom." or "title.1985C6."

CONDITIONS

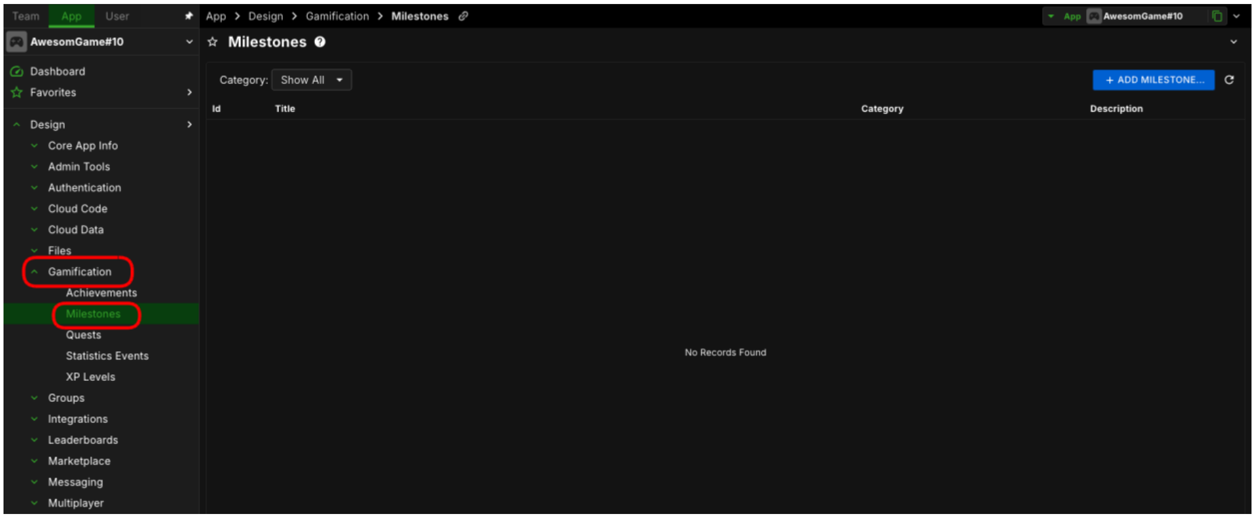
StatisticName | string value | is | totalKills

ACTIONS

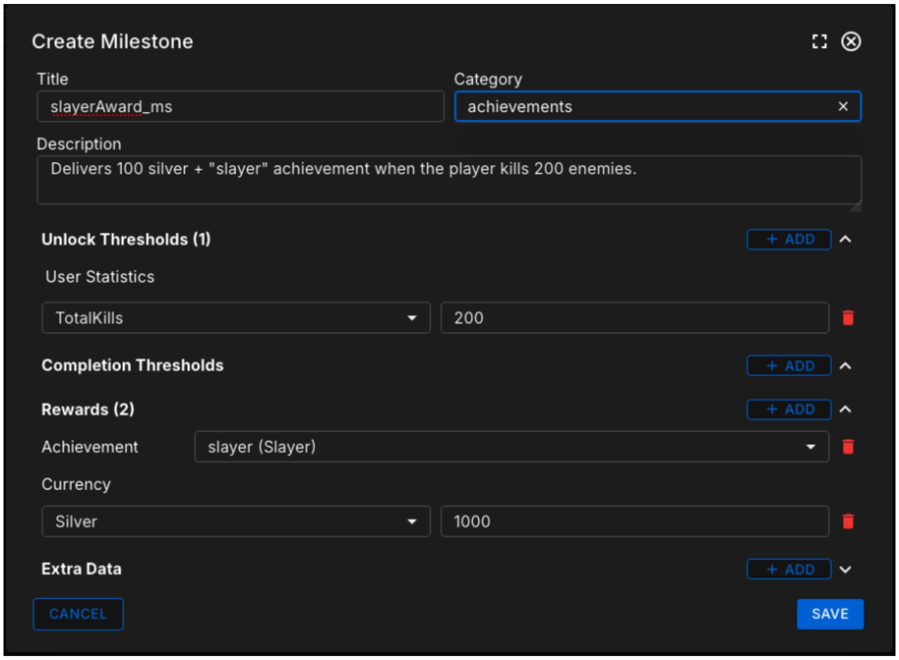
Type: Update player custom properties | Custom property name *: awarded_slayer_ach | Path to value *: StatisticValue

Type: Grant virtual currency (Legacy) | Virtual currency code (Legacy): SR (Silver) | Amount: 1000

You can easily reproduce this behavior in brainCloud using Milestones, found under **Gamification → Milestones** in the menu.



Click the **Add Milestone** button and set up the milestone as shown below. Note that in this example, I've already created an Achievement and Virtual Currency (see previous guide <link>).


Create Milestone

Title: slayerAward_ms

Category: achievements

Description: Delivers 100 silver + "slayer" achievement when the player kills 200 enemies.

Unlock Thresholds (1)

User Statistics: TotalKills 200

Completion Thresholds

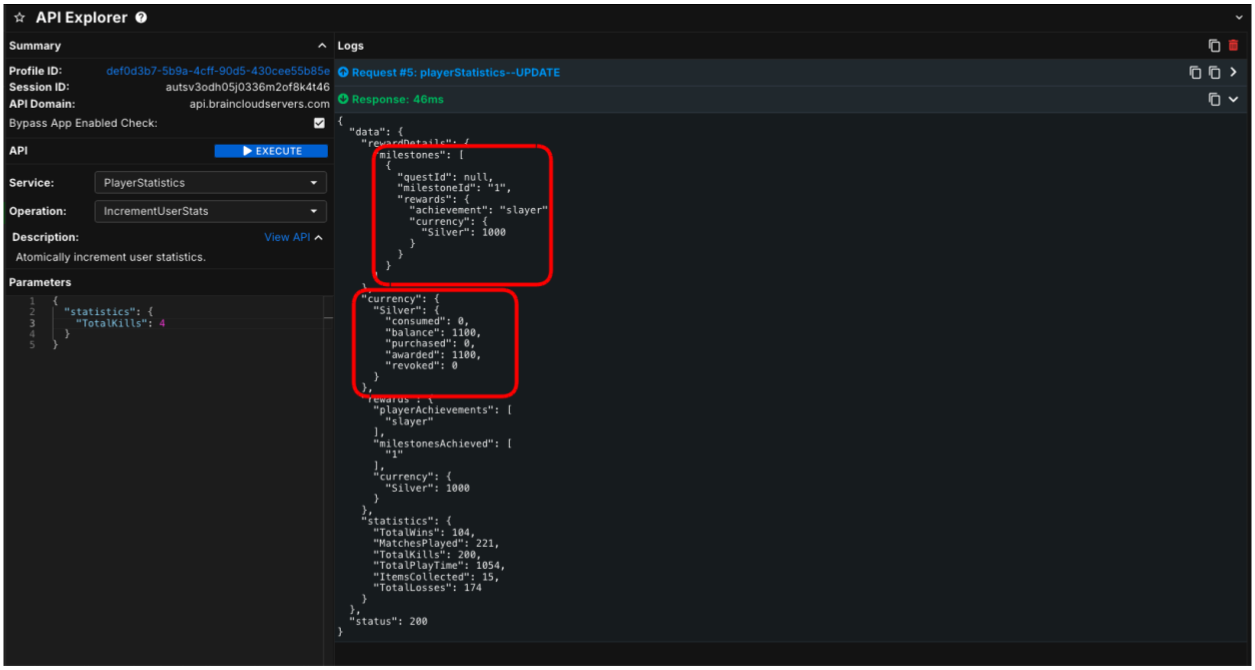
Rewards (2)

Achievement: slayer (Slayer)

Currency: Silver 1000

Extra Data

Now you can test this using the API Explorer with the IncrementUserStat request.



Profile ID: def0d3b7-5b9a-4cff-90d5-430cee55b85e
Session ID: autsv3odh05j0336m2of8k4t46
API Domain: api.braincloudservers.com
Bypass App Enabled Check: ☑

API

Service: PlayerStatistics
Operation: IncrementUserStats
Description:
Atomically increment user statistics.

Parameters

```
{
  "statistics": {
    "TotalKills": 4
  }
}
```

Request #5: playerStatistics--UPDATE
Response: 46ms

```
{
  "data": {
    "rewardDetails": {
      "milestones": [
        {
          "questId": null,
          "milestoneId": "1",
          "rewards": {
            "achievement": "slayer"
            "currency": {
              "Silver": 1000
            }
          }
        }
      ]
    },
    "currency": {
      "Silver": {
        "consumed": 0,
        "balance": 1100,
        "purchased": 0,
        "awarded": 1100,
        "revoked": 0
      }
    },
    "rewards": {
      "playerAchievements": [
        "slayer"
      ],
      "milestonesAchieved": [
        "1"
      ],
      "currency": {
        "Silver": 1000
      }
    },
    "statistics": {
      "TotalWins": 104,
      "MatchesPlayed": 221,
      "TotalKills": 200,
      "TotalPlayTime": 1054,
      "ItemsCollected": 15,
      "TotalLosses": 174
    }
  },
  "status": 200
}
```

You'll see the IncrementUserStats response now includes details on the rewards, milestones, and achievements granted when the *TotalKills* statistic reached 200.

Statistic Events

Statistics Events in brainCloud are similar to PlayFab's rules, but are triggered from the client (or cloud code) when a specific event occurs, such as a player killing an enemy. For example, to increase the TotalKills statistic by 1, you could use the IncrementUserStats request. However, if you also want to award XP or silver, Statistic Events like Milestones let developers set up these actions on the client. Designers can then manage stats and rewards without additional coding.

Global Statistics

brainCloud also offers Global Statistics which function like User Statistics but are accessible by any player in your game. Players can view and edit these stats, or they can be updated via Gamification features such as Milestones and Statistic Events.

For example, you could track both *TotalKills* per user and *TotalKills* across all players. In this case, the Statistic Event could update both the User Statistic and the Global Statistic simultaneously.