

Upgrading to brainCloud from PlayFab

Player Authentication



 Player Authentication	Version: 1.0
	Issue Date: 2026-02-19

Table of Contents

- Introduction..... 3
- Basic Authentication 3
 - Anonymous & Device Authentication 3
 - Email or Username Authentication..... 4
 - Email & Password Settings 4
- Social Authentication 5
- Custom/Advanced Authentication..... 5
 - Authentication Payloads 5
- External Authentication 6
- Authentication API Reference Guide 6

 brainCloud Player Authentication	Version: 1.0
	Issue Date: 2026-02-19

Introduction

Once you have integrated the brainCloud SDK into your project, the next step in your migration is to recreate the authentication/login steps of your game with brainCloud. This guide covers how authentication works on brainCloud, what differs, and how you can meet your game's current requirements.

At the end of this guide, you'll find a table illustrating each PlayFab authentication type and its equivalent request on brainCloud for quick reference.

Basic Authentication

Typically, there are two basic forms of authentication most games use:

- 1. Anonymous Authentication**

Also known as Device Authentication or Frictionless Authentication, this usually requires no input or action from the player. It can be done with a custom ID or a device ID.

- 2. Username & Password Authentication**

A common route is prompting the player to identify themselves, either with a username alone or with a password. *Email* and password combinations are also very common.

Let's first look at how these authentication routes compare between brainCloud and PlayFab.

Anonymous & Device Authentication

With PlayFab, anonymous or device authentication is typically achieved using the `LoginWithCustomID` API, which lets you add any unique identifier for authentication.

This is also possible with brainCloud; however, an extra step is required before calling the auth-request. With brainCloud, you must first call [InitializeIdentity](#). This sets up the authentication service in the SDK with the player's profile ID and the anonymous ID.

Note

When migrating the player's profile, use the same anonymous/device ID as the previous profile, if possible, to facilitate post-authentication data migration.

 Player Authentication	Version: 1.0
	Issue Date: 2026-02-19

You can then use the [AuthenticateAnonymous](#) request to authenticate the player. This request has one parameter, **forceCreate**, which automatically creates a new player profile if one is not found. The profile ID is returned in the authentication response and should be cached on the device, as it will be required for future authentication when **InitialzeIdentity** is called.

This covers most anonymous authentication patterns. However, PlayFab also offers two device authentication routes: **LoginWithIOSDeviceID** and **LoginWithAndroidDeviceID**. Both use the device ID along with additional data, but the critical part is the ID, so this example still works. If you need to send additional data during authentication, check the section on Advanced Authentication Payloads below.

Email or Username Authentication

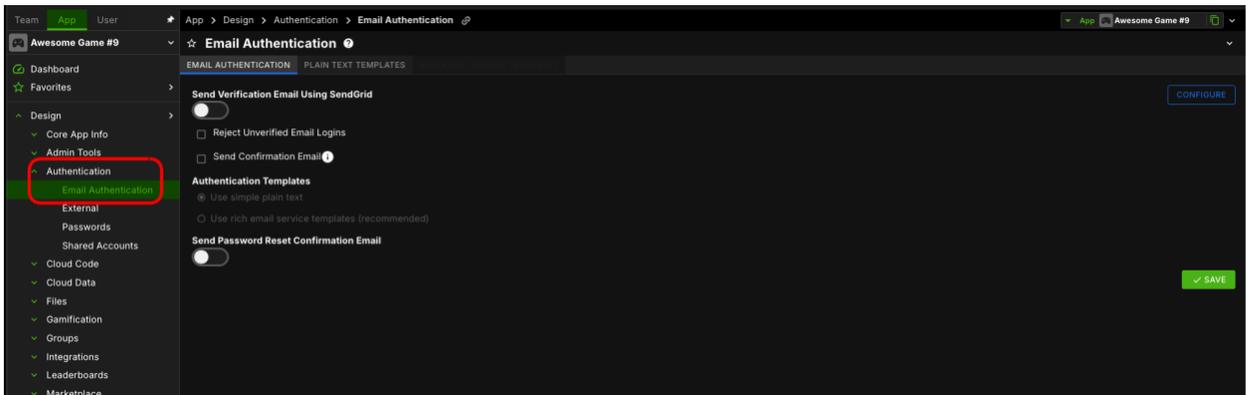
Another common authentication route uses an email and password or a variation, such as a user-name. For email authentication, use [AuthenticateEmailPassword](#), and for user-names, use [AuthenticateUniversal](#).

With both options, registration is not required before calling these requests. Instead, use the **forceCreate** parameter to determine whether to create a new user with the given credentials if one does not exist.

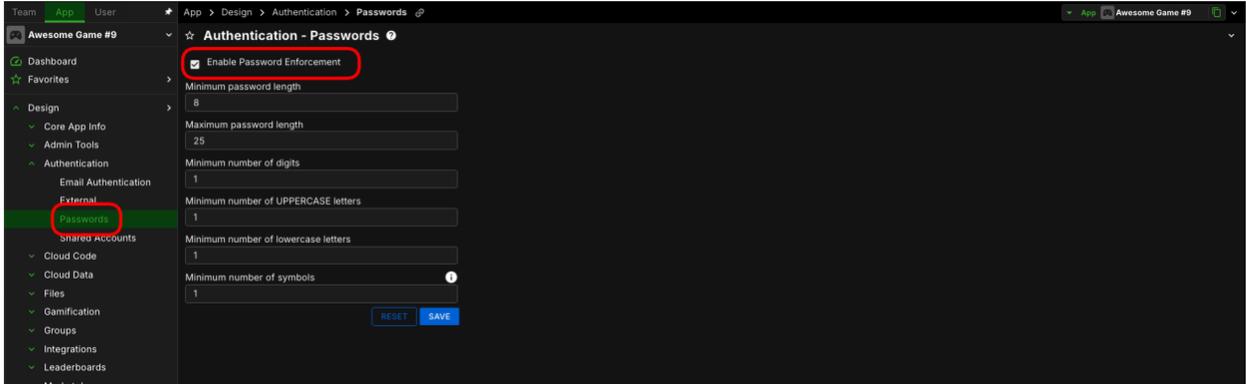
Email & Password Settings

brainCloud allows developers to configure settings for validating emails and passwords used in these requests. These are found in the **Design** → **Authentication** menus.

For emails you can optionally configure account [verification emails](#) to be automatically sent to players using the [SendGrid](#) integration.



You can also configure a custom password enforcement policy for your game if required.



Social Authentication

Social authentication typically uses an ID and/or authentication token from a third-party service to identify the player. This requires integrating an additional SDK alongside the PlayFab or brainCloud SDK to obtain those IDs and tokens. The ID and token are then passed to your backend using a specific authentication request.

The steps required for these forms of authentication are the same for both PlayFab and brainCloud, though each authentication request is different. Check the API reference table below for links on how to authenticate with the specific service you need.

Note that, as with PlayFab, some authentication routes require you to integrate the service with brainCloud via the brainCloud portal. You can find guides for this [here](#).

Custom/Advanced Authentication

In addition to the three authentication routes mentioned above, brainCloud offers another option that might be useful for your migration. [AuthenticateAdvanced](#) provides the flexibility of anonymous, email, and social authentication options in one request. The developer can choose how players identify themselves as their game requires.

Authentication Payloads

AuthenticateAdvanced allows developers to add a custom payload to their authentication request. This enables you to pass data during authentication, which can be handled by pre or post-authentication cloud-code scripts.

External Authentication

If brainCloud's existing authentication APIs do not meet your requirements, or if you perform authentication with an unlisted service before PlayFab authentication, consider using the [AuthenticateExternal](#) request.

This request allows you to perform validation against any third-party application server-side during authentication. Essentially, this is what social authentication requests do under the hood, but it allows you to design it yourself. Check out the guide [here](#) for how to set up external authentication.

Authentication API Reference Guide

As there are quite a few APIs when combining basic and social authentication requests, below is a quick reference guide with links to the brainCloud documentation for the corresponding PlayFab APIs.

PlayFab	brainCloud
LoginWithCustomID	AuthenticateAnonymous
LoginWithEmailAddress	AuthenticateEmailPassword
RegisterPlayFabUser	AuthenticateEmailPassword
LoginWithOpenIdConnect	AuthenticateGoogleOpenId
LoginWithFacebook	AuthenticateFacebook
LoginWithGoogleAccount	AuthenticateGoogle
LoginWithGooglePlayGamesServices	Custom - AuthenticateExternal
LoginWithSteam	AuthenticateSteam
LoginWithKongregate	Custom - AuthenticateExternal

 Player Authentication	Version: 1.0
	Issue Date: 2026-02-19

LoginWithAndroidDeviceID	AuthenticateAnonymous
LoginWithIOSDeviceID	AuthenticateAnonymous
LoginWithApple	AuthenticateApple
LoginWithGameCenter	AuthenticateGameCenter
LoginWithXbox	Custom - AuthenticateExternal
LoginWithPSN	AuthenticatePlaystationNetwork
LoginWithWindowsHello	Custom - AuthenticateExternal
LoginWithTwitch	Custom - AuthenticateExternal
LoginWithNintendoServiceAccount	AuthenticateNintendo
LoginWithBattleNet	Custom - AuthenticateExternal