GAME TESTER

LIONBRIDGE GAMES STUDIOS

GAME TESTER
LIONBRIDGE GAMES STUDIOS

API DOCUMENTATION V1.6D

CONTENTS

support@gametester.co
www.gametester.gg

GameTester Pty Ltd
3 Dock Road
Waterway House
Cape Town
8000
South Africa

INTRODUCTION

This document details the usage of the Game Tester API, used by game developers to send analytical data during tests.

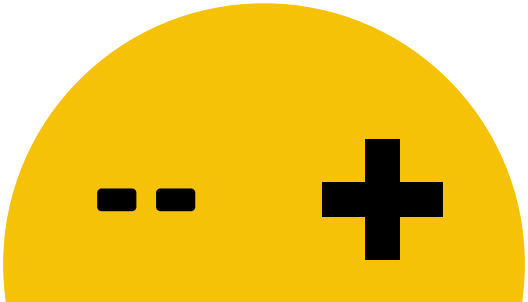
All server responses will have an HTTP status code of 200 and a result code in the response that can either be a success or error. If the HTTP status code is not 200 then it is likely a network related problem and should be handled by the developer.

All response data will have this format {code: integer, message: string}. Success messages will have a code of -1. All others will have specific error numbers that should be dealt with differently. Each endpoint will list the error codes that can be expected.

If your game is in closed development and you don't want to have the test's game play be recorded or distributed, we suggest you add a watermark in the game that contains the tester's token. This way it will be easy to track which tester shared his game play and should also dissuade recording or sharing from the start.

CHANGELOG

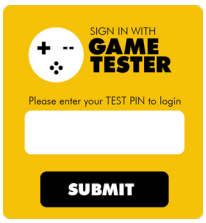
V1.6d - Removed the need to supply a redirect URL when requesting a connect token.
This URL should be supplied during test setup.
V1.6b - Added error code 14 which occurs when a player tries to /auth when
their test has already been unlocked.
V1.6a - Changed dynamic login url query parameters to 'gt_pin' and 'gt_user_id'.
V1.6 - Auth endpoint now returns a playerToken that should be used in subsequent calls.
- Renamed old playerToken to connectToken to avoid name clash.
- Moved unlock function to its own URL (/unlock)
- Some error codes have changed.
V1.5 - Changed success code from 0 to -1 to prevent false positives in the event of a response parsing error.

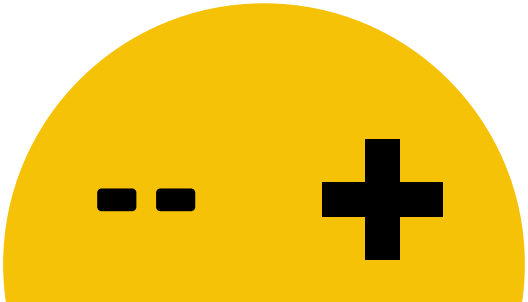


DOWNLOADABLE ASSETS

You can also view a **quick tutorial on the Game Tester API here:**

We have a downloadable asset pack available with images, buttons and forms that you can use for the integration. **Download it here.**





CONNECTION URLs

PRODUCTION

https://server.gametester.gg/dev-api/v1
When communicating with the production URL you should use your test's token for authentication. When you are ready to validate your API implementation for your test you can find a special validation PIN on the validation tab of the last step of the test's setup wizard. Use this PIN to play your game as a special tester before the test is live.

SANDBOX

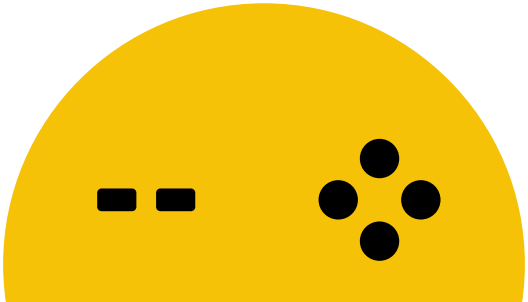
https://server.gametester.gg/dev-api/v1/sandbox
Use the sandbox url to test out different calls to the API without having to implement it in your game.
The sandbox has specific values that you can pass to simulate how the API responds to a variety of scenarios.
You can use tools such as Postman to connect to the sandbox and get familiar with the API.

Note: Do not use your test's token in sandbox mode. Only use the values that are described in the Sandbox section further down.

PLUGINS & EXAMPLES

We provide the following plugins to make it easier for you implement the API in your game. Over time this list will grow to include more game engines. If your engine is not here, you just need to check it's documentation on how to send HTTP POST messages.

UNITY
Plugin: https://github.com/Game-Tester/unity-api-wrapper
Example: https://github.com/Game-Tester/unity-example



ENDPOINTS

/AUTH {{POST}}

This endpoint is used for validating the developer and players. This endpoint requires the test's developerToken and either the playerPin or connectToken. It returns a playerToken if successful. This player token should be used for all subsequent calls to the API.

Path: https://server.gametester.gg/dev-api/v1/auth

It expects the following JSON data structure:

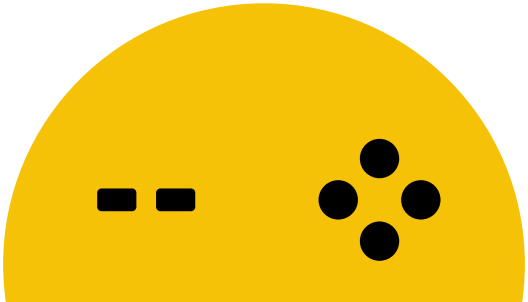
```
{
"developerToken": string, required,
"connectToken": string, required if not using playerPin,
"playerPin": string, required if not using connectToken
}
```

developerToken: This is the developer token that identifies the developer and test. It can be found on the test summary for a test that has been created.

connectToken: This is a token that can be used to authenticate the player. To get this token a player must be navigated to this URL: https://app.gametester.gg/auth/connect/{{testId}} where testId is the id of the test that can be found on the test setup page. The URL can also be found in the API Info page of the test setup wizard. The tester will be redirected to the URL that you provided during setup with a query string of &token={{connectToken}}. You can then read the connect token and use it to call the /auth endpoint to get the playerToken for subsequent API calls.

playerPin: An alternate authentication method is using a playerPin. When a player accepts a test a pin will be generated for that player. It is up to the developer to have an input for the player to type this pin to validate the player.

Optionally during test creation the developer can also choose to set up a dynamic login URL. The player will be sent to this URL to start the test and we will attach the player's pin as a query string using 'gt_pin' as the field name and the player's id as 'gt_user_id' if you need it. Example: https://mygame.com?gt_user_id=123456&gt_pin=abc123. You can now read gt_user_id and gt_pin to get the testers details without having to create a form to capture the information. This is useful if you already have a webpage that the tester can be directed to.



/ {{POST}}

This is the main endpoint for sending data points. E.g. https://server.gametester.gg/dev-api/v1
It expects the following JSON data structure:

```
{
“developerToken”: string, required,
“playerToken”: string, required,
“datapointId”: Integer, required,
}
```

datapointId: An integer that represents the endpoint that data is being sent for. These endpoints are created during test creation and their ids can be found on the test summary.

/UNLOCK {{POST}}

Once the test requirements have been met it is required to send a call to this endpoint to inform us the player completed the test. If successful a message should be shown to the player informing him that the test is complete. Check the asset pack for examples. E.g. https://server.gametester.gg/dev-api/v1/unlock
It expects the following JSON data structure:

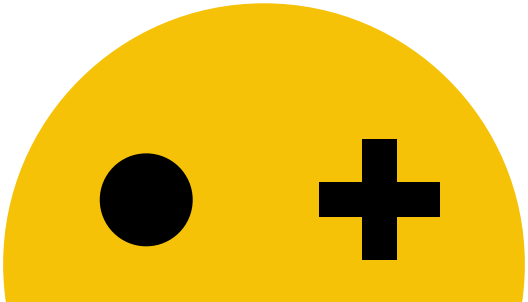
```
{
“developerToken”: string, required,
“playerToken”: string, required,
}
```

ERROR RESPONSES

Calls to the endpoints can result in the following errors being returned:

- { code: 1, message: ‘Missing developer token’ }
- { code: 2, message: ‘Missing player auth’ }
- { code: 3, message: ‘Invalid developer token’ }
- { code: 4, message: ‘Invalid player connect token’ }
- { code: 5, message: ‘Invalid player pin’ }
- { code: 6, message: ‘Missing “datapointId” property’}
- { code: 7, message: ‘Datapoint does not exist’ }
- { code: 8, message: ‘Test is not currently running’ }
- { code: 9, message: ‘Invalid player for this test’ }
- { code: 10, message: ‘Test already unlocked’ }
- { code: 11, message: ‘Test is not in setup state’ }
- { code: 12, message: ‘Missing player token’ }
- { code: 13, message: ‘Invalid player token’ }
- { code: 14, message: ‘Player already finished the test’ }

It is up to the developer to handle these potential errors in a graceful manner.



SANDBOX

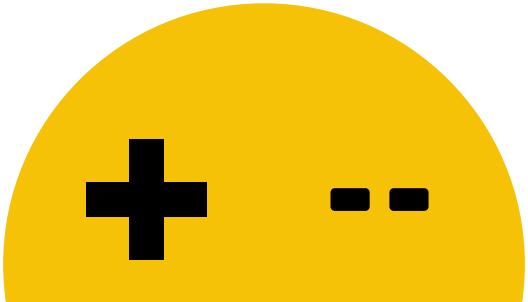
/AUTH {{POST}}

The sandbox AUTH endpoint can be used to test if your authentication structure is correct.
Path: https://server.gametester.gg/dev-api/v1/sandbox/auth
It expects the following JSON data structure:

```
{
“developerToken”: string, required,
“connectToken”: string, required if not using playerPin,
“playerPin”: string, required if not using connectToken
}
```

Use the following data for testing:

Description	Property	Value
To test for a successful developer token	developerToken	“validToken”
To test for a successful connect token	connectToken	“validToken”
To test for a successful player pin	playerPin	“validPin”
To test for valid authentication but a player that is not valid for the test	connectToken or playerPin	“invalidPlayer”
To test for a valid authentication but the test is not running	developerToken	“validTokenButNoTest”
To test /auth for a player who’s test is already unlocked	playerPin	“unlockedPlayer”



/ {{POST}}

The sandbox / endpoint can be used to test if your data point structure is correct.
Path: https://server.gametester.gg/dev-api/v1/sandbox

It expects the following JSON data structure:

```
{
“developerToken”: string, required,
“playerToken”: string, required,
“datapointId”: Integer, required,
}
```

Use the following data for testing:

Description	Property	Value
To test for a valid data point	data point Id	1
To test for an invalid data point	data point Id	2

/UNLOCK {{POST}}

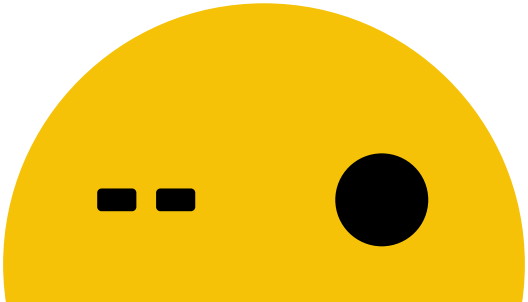
The sandbox /unlock endpoint can be used to test if your unlock call works correctly.
Path: https://server.gametester.gg/dev-api/v1/sandbox/unlock

It expects the following JSON data structure:

```
{
“developerToken”: string, required,
“playerToken”: string, required,
}
```

Use the following data for testing (any other values will result in an error):

Description	Property	Value
To test a valid unlock	developerToken	“validToken”
	playerToken	“validToken”



EXAMPLES:

AUTHENTICATING THE PLAYER

URL: https://server.gametester.gg/dev-api/v1/auth
Method: POST
Content-type: application/json
Payload:

```
{
“developerToken”: “eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9…”,
“playerPin”: “ABC123”,
}
```

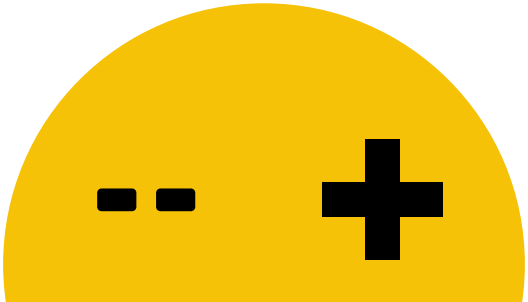
Response: { code: -1, message: ‘Success’, playerName: ‘TestPlayer’, playerToken: ‘eyJ0eXAiOiJKV1Qi...’ }

SENDING A VALID DATA POINT

URL: https://server.gametester.gg/dev-api/v1
Method: POST
Content-type: application/json
Payload:

```
{
“developerToken”: “eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9…”,
“playerToken”: “eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6I…”,
“datapointId”: 1
}
```

Response: { code: -1, message: ‘Success’ }



NOTIFYING THAT THE TEST IS UNLOCKED

URL: https://server.gametester.gg/dev-api/v1/unlock
Method: POST
Content-type: application/json
Payload:

```
{
“developerToken”: “eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9…”,
“playerToken”: “eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6I…”,
}
```

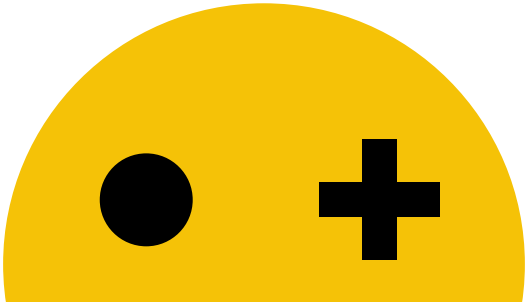
Response: { code: -1, message: Unlock completed }

SENDING AN INVALID DATA POINT

URL: https://server.gametester.gg/dev-api/v1
Method: POST
Content-type: application/json
Payload:

```
{
“developerToken”: “eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9…”,
“playerToken”: “eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6I…”,
“datapointId”: 6
}
```

Response: { code: 7, message: ‘Data point does not exist’ }



THANK YOU