iGameMedia

ONEPlatform Integration

Hybrids Supplement

Contents

Introduction

Welcome to the guide for the API of ONEPlatform. If you're here, it's likely that you're interested in integrating our product with your own software or service. ONEPlatform is a powerful tool that enables you to access a wide range of video content through a single RESTful API. With this guide, you'll learn how to connect to our API and start using our services in your own applications.

Our API is designed to be easy to use and flexible, allowing you to access the full range of features and services that ONEPlatform has to offer. Whether you're building a new application from scratch or integrating with an existing system, our API makes it easy to get started.

Throughout this guide, we'll walk you through the process of getting started with the ONEPlatform API, from setting up your account and obtaining your API keys, to making your first API call and accessing the full range of content available to you. So, let's get started!

Content Access

In order to integrate with iGameMedia's streaming services, iGame requires each client to pass the correct access details to our ONEPlatform API.

The below details should be used for accessing event info and streaming content:

Partner Name

CustomerUID

Whitelisted IP's

Note: Please do not disclose your Customer GUID to any third party. Disclosure of this information could jeopardize the security of your integration with iGameMedia.

API Developer Guide

Getting Started

This document will familiarise you with the APIs and associated methods required to integrate live iGame Media video streaming content into your website.

Note: This API does not deliver the video stream itself - it only tells you what the URL for a given stream will be. You will need to connect to the video stream separately or via our player.

(Please discuss with us, if your intention is to use our player)

What you will need before you start

- A link to swagger UI
- Your CustomerUID for the service – This is a unique GUID used for authorisation on our APIs.
- A list of the IP addresses you intend to call our services from so we can whitelist them.
- A list of response codes found in section 7 of this document.

Please contact support@igamemedia.com if you have not yet received this information.

API Domain Name

For Test

https://stream-services-test.igamemedia.com/api

For Production

https://stream-service.igamemedia.com/api

Date Format

All Date objects are returned in the UTC time zone, using the ISO 8601 combined format. For more information about ISO 8601, please see https://en.wikipedia.org/wiki/ISO_86015.

THEO Live Player

Installation

The package to install is available at: https://www.npmjs.com/package/@theolive/player.

Install using your favorite package manager for Node (such as npm or yarn):

```
npm install @theolive/player
```

Using NPM means that you will need to host the .js files yourself, and periodically update these to the latest version.

File

THEOLive.js

This is the main Javascript file containing the logic to create a player and load a channel. This version contains the THEOlive UI.

THEOLive.sw.js

This is a Service Worker that allows playback of THEOlive streams on iOS Safari. It is important that this file is hosted on the same domain as the webpage over https.

Note: In order to use THEOlive on any iOS Browser, we need the THEOlive serviceworker to be active. This implies the following rules:

- HTTPS is required
- The THEOLive.sw.js file needs to be on the same domain and path as the page initiating the player

Customise THEO Live

First you should load the THEOlive player in your HTML page:

```
<script type=”text/javascript” src=node_modules/@theolive/player/THEOLive.js></script>
```

Note: If the file is hosted on a different location,
make sure to update the src to the correct path.

First you should load the THEOlive player in your HTML page:

```
<script type=”text/javascript”>
    if (THEOLive.requiresServiceWorker()) {
        navigator.serviceWorker.register(“THEOLive.sw.js”)
    }
</script>
```

Add an html element to pass to the player

`<div id=”player”></div>`

Finally, create the player and load the desired channel

```
<script type=”text/javascript”>
    const player = new THEOLive.Player(document.getElementById(‘player’));
    player
        .loadChannel(‘<your-channel-id>’)
        .then(() => {
            player.play(); // not required if your channel is configured to autoplay
        }).catch(console.error);

</script>
```

Full Example

```
<!DOCTYPE html>
<html lang=”en”>
<head>
    <title>Test page</title>
    <link href=node_modules/@theolive/player/THEOLive.css rel=”stylesheet”>
        <script type=”text/javascript” src=node_modules/@theolive/player/THEOLive.js></script>
    <style>
        #player {
            width: 80vw;
            margin: auto;
        }
    </style>
</head>
<body>
<h1>Testing</h1>
<div id=”player”></div>

<script type=”text/javascript”>
    if (THEOLive.requiresServiceWorker()) {
        navigator.serviceWorker.register(“THEOLive.sw.js”)
    }
    const player = new THEOLive.Player(document.getElementById(‘player’));
    player
        .loadChannel(‘<your-channel-id>’)
        .then(() => {
            player.play(); // not required if your channel is configured as autoplay
        }).catch(console.error);

</script>
</body>
</html>
```

Make sure to replace <your-channel-id> with your actual channel-id and that you have completed the steps in the getting started tutorial and your channel is up and running.

In order to include THEO Live player in a React application please refer to: https://github.com/THEOplayer/theolive-react-sample

Events

Get Event

This method call will return all events for a particular date.

Request Parameters

Please review the section “Getting Started” for more information.

customerUid: Your company’s identifier. This is provided to you by iGame.

eventDate: Date of the event. Date passed with format “DD-MM-YYYY”.

Response Model

```
{
    “events”: [
        {
            “eventId”: “”,
            “locationCode”: “”,
            “locationName”: “”,
            “sportCode”: “”,
            “eventStatusCode”: “”,
            “eventNumber”: “”,
            “title”: “”,
            “startTime”: “”,
            “estimatedEndTime”: ““,
            “geoAllow”: “”,
            “geoBlock”: “”
        }
    ]
}
```

Response Example

```
curl -XGET -H “Content-type: application/json” ‘https://stream-services-test.igamemedia.com/api/event/all/{customerUid}/{eventDate}’
```

```
{
    “events”: [
            {
                “eventId”: “013WOFT640-32325”,
                “locationCode”: “013WOFT640”,
                “locationName”: “Dutch Eredivisie 2015”,
                “sportCode”: “FT”,
                “eventStatusCode”: “P”,
                “eventNumber”: “32325”,
                “title”: “NEC v FC Twente”,
                “startTime”: “2020-11-09T:00:00:00”
                “estimatedEndTime”: “2020-11-09T:02:00:00”,
                “geoAllow”: “GB,IE”,
                “geoBlock”: “”
            }
        ]
}
```

Get Open Event

This method call will return all events for a particular date that are open. You can then use one of these event ID’s to generate a token by calling a token service.

Request Parameters

Please review the section “Getting Started” for more information.

customerUid: Your company’s identifier. This is provided to you by iGame.

Response Model

```
{
    “events”: [
        {
            “eventId”: “”,
            “locationCode”: “”,
            “locationName”: “”,
            “sportCode”: “”,
            “eventStatusCode”: “”,
            “eventNumber”: “”,
            “title”: “”,
            “startTime”: “”,
            “estimatedEndTime”: ““,
            “geoAllow”: “”,
            “geoBlock”: “”
        }
    ]
}
```

Response Example

```
curl -XGET -H “Content-type: application/json” ‘https://stream-services-test.igamemedia.com/api/event/open/{customerUid}’
```

```
{
    “events”: [
        {
            “eventId”: “013WOFT640-32325”,
            “locationCode”: “013WOFT640”,
            “locationName”: “Dutch Eredivisie 2015”,
            “sportCode”: “FT”,
            “eventStatusCode”: “P”,
            “eventNumber”: “32325”,
            “title”: “NEC v FC Twente”,
            “startTime”: “2020-11-09T:00:00:00”
            “estimatedEndTime”: “2020-11-09T:02:00:00”,
            “geoAllow”: “GB,IE”,
            “geoBlock”: “”
        }
    ]
}
```

Get Events Hybrid

This method will return all events for a particular date along with their mapped event creation IDs.

Request Parameters

Please review the section “Getting Started” for more information.

customerUID: Your company’s identifier. This is provided to you by iGame.

eventDate: Date of the event. Date passed with format “DD-MM-YYYY”

Response Model

```
{
    “events”: [
        {
            “eventId”: “013WOFT640-32325”,
            “locationCode”: “013WOFT640”,
            “locationName”: “Dutch Eredivisie 2015”,
            “sportCode”: “FT”,
            “eventStatusCode”: “P”,
            “eventNumber”: “32325”,
            “title”: “NEC v FC Twente”,
            “startTime”: “2020-11-09T:00:00:00”
            “estimatedEndTime”: “2020-11-09T:02:00:00”,
            “geoAllow”: “GB,IE”,
            “geoBlock”: “”
            “btrId”:“”,
            “btgId”:“”
        }
    ]
}
```

Response Example

curl -XGET -H “Content-type: application/json” ‘https://stream-services-test.igamemedia.com/api/event/hybrid/all/{customerUID}/{EventDate}’

```
{
    “events”: [
            {
                “eventId”: “013WOFT640-32325”,
                “locationCode”: “013WOFT640”,
                “locationName”: “Dutch Eredivisie 2015”,
                “sportCode”: “FT”,
                “eventStatusCode”: “P”,
                “eventNumber”: “32325”,
                “title”: “NEC v FC Twente”,
                “startTime”: “2020-11-09T:00:00:00”
                “estimatedEndTime”: “2020-11-09T:02:00:00”,
                “geoAllow”: “GB,IE”,
                “geoBlock”: “”
                “btrId”:”83884351”,
                “btgId”:”13411129”
            }
        ]
}
```

Get Open Events Hybrid

This method call will return all events for a particular date that are open along with their mapped event creation IDs. You can then use one of these event id's to generate a token by calling a token service.

Request Parameters

Please review the section "Getting Started" for more information.

customerUID: Your company's identifier.
This is provided to you by iGame.

Response Model

```
{
    "events": [
        {
            "eventId": "",
            "locationCode": "",
            "locationName": "",
            "sportCode": "",
            "eventStatusCode": "",
            "eventNumber": "",
            "title": "",
            "startTime": "",
            "estimatedEndTime": null,
            "geoAllow": "",
            "geoBlock": ""
            "btrId":"",
            "btgId":""
        }
    ]
}
```

Response Example

curl -XGET -H "Content-type: application/json" 'https://stream-services-test.igamemedia.com/api/event/hybrid/open/{customerUID}/{EventDate}'

```
{
    "events": [
        {
            "eventId": "013WOFT640-32325",
            "locationCode": "013WOFT640",
            "locationName": "Dutch Eredivisie 2015",
            "sportCode": "FT",
            "eventStatusCode": "P",
            "eventNumber": "32325",
            "title": "NEC v FC Twente",
            "startTime": "2020-11-09T:00:00:00"
            "estimatedEndTime": "2020-11-09T:02:00:00",
            "geoAllow": "GB,IE",
            "geoBlock": ""
            "btrId":"83884351",
            "btgId":"13411129"
        }
    ]
}
```

Streaming

Platform management providers have only one integration to manage. Our fully dedicated iGame team with decades of experience is always ready to troubleshoot and manage day to day schedules, code changes and event mapping.

Get Streams for Events

This method will return all available streams for a specific event. We provide HLS streams for all content but iGame Media also provides the industry’s first HESP ultra-low latency streams for select content.

-HLS
-ULL

Request Parameters

Please review the section “Getting Started” for more information.

customerUid: Your company’s identifier. This is provided to you by iGame.

eventId: ID of specific event you wish to stream.

Response Model

```
{
    “eventStreams”: [
        {
            “uniqueStreamName”: “”,
            “streamType”: “”,
            “description”: “”,
            “geoCountryAllow”: “”,
            “geoCountryBlock”: “”,
            “playerWidth”: ,
            “playerHeight”: ,
            “fixedSize”:,
            “controls”:,
            “fullscreen”:,
            “videoBitRateKbps”:,
            “audioBitRateKbps”: ,
            “streamStretching”: “”,
            “streamWidth”:,
            “streamHeight”:
        }
    ]
}
```

Response Example (HLS Only)

curl -XGET -H “Content-type: application/json” ‘https://stream-services-test.igamemedia.com/api/stream/{customerUid}/{eventId}’

```
{
    “EventStreams”: [
        {
            “uniqueStreamName”: “HLS”,
            “streamType”: “HLS”,
            “description”: “PRM HLS HIGH”,
            “geoCountryAllow”: “UK, US”,
            “geoCountryBlock”: “GB, AT”,
            “playerWidth”: 640,
            “playerHeight”: 360,
            “fixedSize”: true,
            “controls”: false,
            “fullscreen”: false,
            “videoBitRateKbps”: 500,
            “audioBitRateKbps”: 22,
            “streamStretching”: “exactfit”,
            “streamWidth”: 640,
            “streamHeight”: 360
        }
    ]
}
```

Generate Event View

Return Raw Link

This method is used to return the raw link as provided to us by our data providers. This allows you the flexibility to use this raw link within your own player and native apps.

Request Parameters

Please review the section “Getting Started” for more information.

customerUid: Your company’s identifier. This is provided to you by iGame.

userId: Unique identifier for end-user.

userIp: IP address of end-user.

Note: For server-side calls please specify end-user IP in ‘X-Forwarded-For’ header

eventId: ID of specific event you wish to stream.

streamName: This value can be found by getting the list of streams for event.

Note: Please append “-RAW” to end of UniqueStreamName e.g. “HLS-RAW”

redirectUrl: This will be the page the user is redirected to when the event is over.

Request Model

```
{
    “customerUid”: “”,
    “userId”: “”,
    “eventId”: “”,
    “userIp”: “”,
    “streamName”: “”,
    “redirectUrl”: “”
}
```

Response Model

```
{
    “streamLink”: “”,
    “streamType”: “”,
    “provider”: “”
}
```

Response Example

```
curl -XPOST -H “Content-type: application/json” ‘https://stream-services-test.igamemedia. com/api/stream/link?customerUID={customerUid}&userId={userId}&userIp={userIp}&eventID={eventID}&streamName={streamName}-RAW&redirectURL={redirectURL}’
```

```
{
    “streamLink”:”NjA7YzliODYyMTFmMWYyNTFjYmY0NTdkMDEzMzVkN2E0OTliYjkzZmY
                   3NDg5Y2JjYzhlMDJmOTE1NDk3ZGQ2MmEwN2ZlMTdmYjI3Y2Q0MGI5M2Y=”,
    “streamType”: “HLS”,
    “provider”: “IGM”
}
```

Using Encoded Transaction Reference to Retrieve Raw Stream URL

The encoded transaction reference is returned in streamLink parameter

Response Model

```
{
    “streamLink”: “”,
    “streamType”: “”,
    “provider”: “”
}
```

Response Example

`curl -XPOST -H “Content-type: application/json” ‘https://player-test.igamemedia.com/api/liveStream?encodedTransactionRef={EncodedStreamLink}”`

```
{
    “streamLink”: “https://stream.test.com/streammobile.m3u8?referer=
    client&stream=demo_mbr&t=1602541578473&h=fd500b8f9f54225f0e135e61
    36125724&forceformat=ios&hd=1”,
    “streamType”: 2,
    “contentDeliveryNetwork”: “AAA”,
    “transactionId”: 61,
    “streamToken”: “c9b86211f1f251cbf457d01335d7a499bb93ff
    7489cbcc8e1aa9c4dfc666ac78a32f99b4f7b140ef”,
    “eventId”: “IGMRCN-DEMO”
}
```

Note: When returning a raw stream link for an available ULL event, the relevant Theo channel id will be returned in the streamLink field

Using RAW Stream Link

Pass the stream URL to your player (see the documentation of the player you are using). For example, JWPlayer can be configured with the following code:

```
var streamUrl = “https://bitdash-a.akamaihd.net/content/sintel/hls/playlist.m3u8”;

var playerInstance = jwplayer(“myElement”);

playerInstance.setup({ file: streamUrl });
```

Generate Event View

Return Player Link

This method returns an iGame Player link that can be used on your sites.

Request Parameters

Please review the section “Getting Started” for more information.

customerUid: Your company’s identifier. This is provided to you by iGame.

userId: Unique identifier for end-user.

userIp: IP address of end-user.

Note: For server-side calls please specify end-user IP in ‘X-Forwarded-For’ header

eventId: ID of specific event you wish to stream.

streamName: This value can be found by getting the list of streams for event.

Request Model

```
{
    “customerUid”: “”,
    “userId”: “”,
    “eventId”: “”,
    “userIp”: “”,
    “streamName”: “”,
    “redirectUrl”: “”
}
```

Response Model

```
{
    “streamLink”: “”,
    “streamType”: “”,
    “provider”: “”
}
```

Response Example

curl -XPOST -H “Content-type: application/json” ‘https://stream-services-test.igamemedia.com/api/stream/link?customerUid={customerUid}&userId={userId}&userIp={userIp}&eventId={eventId}&streamName={streamName}&redirectURL={redirectURL}’

```
{
    “streamLink”: “https://player-test.igamemedia.com/
liveplayer?
    tid=10001&tk=2bde432b26f1734a1b787146e4adf735b76e
    ab2eefd36a732c5c270d701d7ea08699”,
    “streamType”: “HLS”,
    “provider”: “IGM”
}
```

Note: When returning an available ULL stream, the relevant Theo channel id will be returned in the streamLink field.

Using Player Link

To use our player link, the following example is shown using iframe:

<iframe src=”https://player-test.igamemedia.com/liveplayer?tid=10001&k=2bde432b26f1734a1b787146e4adf735b76eab2eefd36a732c5c270d7” title=”Example stream” allowfullscreen=””></iframe>

Error Codes

REST APIs use the Status-Line part of an HTTP response message to inform clients of their request's overarching result. The following are the operation codes used in the ONEPlatform API:

Status Codes	Description
200	Successful Request
400	Bad Request - Verify that query parameters are correct
500	Internal Server Error

REST Response Codes

All methods response models contain the fields:

```
{
    “message”: “string”,
    “operationCode”: “int”
}
```

The operationCode is the custom status code and message is the description of this code. The following are the custom status codes used in the ONEPlatform API:

Status Codes	Description
300	Your customer details are invalid, please contact support. This could mean your customerUid is invalid.
301	You must supply a valid IP Address for your end user. When generating a token to view an event, you must supply the end users IP address.
302	You must supply a valid User Identifier. When generating a token to view an event, you must supply the end users identifier or something that uniquely identifies the user.
303	Calls from this Server are not allowed. This means you have not sent through the IP address of the server that is calling our web services.
304	The Unique Stream Name is invalid. Each event will have a list of streams associated with it, in order to view the stream please send through the correct unique stream name. You can get a list of the stream names by calling the “GetStreamsForEvent” in the Token service.
305	The Redirect URL is invalid. A valid URL must contain the http or https protocol.
306	This event is unavailable in the requesting user’s region.
307	Unable to get stream link from external provider
308	Invalid value of encoded transaction reference

Status Codes	Description
401	Event not found. The eventId that you have sent through cannot be found.
402	Event is still pending.
403	Event is now closed.
404	Event is cancelled. You can only generate a token if the Event is Open and the Event Status Code is O.

Sports

Retrieve All Sports

You can retrieve a list of sports and their codes.

Request Parameters

customerUid: Your company’s identifier.
This is provided to you by iGame.

Response Model

```
{
    “Sports”: [
        {
            “Code”: “”,
            “Name”: “”
        }
    ]
}
```

Response Example

curl -XGET -H “Content-type: application/json” ‘https://stream-services-test.igamemedia.com/api/sport/all/{customerUid}’

```
{
    “Sports”: [
        {
            “Code”: “HR”,
            “Name”: “Horse Racing”
        }
    ]
}
```

Sports Codes

Code	Sport	Code	Sport	Code	Sport	Code	Sport
AS	Alpine Skiing	EQ	Equestrian	HI	Indoor Hockey	RG	Rugby
AF	American Football	EC	E-Sport Counter-Strike	LH	In-line Hockey	RL	Rugby League
AT	Athletics	ED	E-Sports Dota	JD	Judo	SL	Sailing
AR	Aussie Rules	EL	E-Sport League of Legends	KY	Kayak	SR	Salomon Running
BM	Badminton	ES	E-Sport	LC	Lacrosse	FW	Salomon Winterpark
BB	Basketball	FN	Fencing	MM	Mixed Martial Arts	SX	Skicross
BS	Beach Soccer	FT	Football	MS	Motorsports	SK	Skiing
BV	Beach Volleyball	FS	Futsal	MB	Mountain Bike	SO	Snooker
BD	Billiard	GF	Golf	NB	Netball	SB	Snowball
BL	Bowls	GH	Greyhound Racing	NA	None	SC	Soccer
BX	Boxing	HB	Handball	OG	Olympic Games	ST	Sonstiges
CL	Climb	HK	Hockey	PD	Padel	SQ	Squash
CK	Cricket	HR	Horse Racing	PS	Parasports	SF	Surfing
CS	Cross-Country Skiing	IH	Ice Hockey	PK	Poker	SW	Swimming
CU	Curling	IF	Indoor Football	PL	Pool	TT	Table Tennis
CC	Cycling	TS	Test Sport	RH	Rink Hockey	VB	Volleyball
DT	Darts	TR	Triathlon	RB	Roll Hockey	WB	Waterball
DV	Diving	VR	Virtual	RW	Rowing	WP	Waterpolo

iGame Media

GLOBAL HEADQUARTERS
55-59 Adelaide Street,
Belfast BT2 8FE, N.I.

igamemedia.com