UFC 5.0

Operator Integration Instructions

Please note that the code snippets displayed in this article are intended to be used for assessment and testing purposes only. Prior to going live, please reach out to integrations@imgarena.com.

Integration Library

The Front Row Seat integration library is distributed with an ES module build and a UMD build, the examples below use the UMD syntax for clarity and to show the simplest implementation.

The package is distributed using npm, where both ES module and UMD builds are available: https://www.npmjs.com/package/@img-arena/front-row-seat

The UMD build is available on unpkg: https://unpkg.com/@img-arena/front-row-seat

For more information on how to use unpkg please see their docs: https://unpkg.com

Usage

Integrating the IMG Arena UFC Event Centre into your website can be achieved by following the below steps:

1. Add a "***container***" HTML DOM element for where the Event Centre should appear.
2. Include the integration library script.
3. Initialise the integration library.

```
<!DOCTYPE html>
<html>
  <head>
    <title>IMG Arena Event Centre</title>
    <meta charset="UTF-8" />
    <link rel="stylesheet" href="./styles.css" />
  </head>

  <body>
    <div id="img-arena-event-centre"></div>
    <script src="https://unpkg.com/@img-arena/front-row-seat"></script>
    <script>
      const { MessageTopics } = frontRowSeat.eventCentreUtils;
      const eventCentreInstance = frontRowSeat.eventCentre({
        operator: "OPERATOR-NAME",
        sport: "ufc",
        targetModule: "full",
        eventId: "EventID",
        version: "version number",
        targetElementSelector: "#img-arena-event-centre",
        language: "en"
      });

      eventCentreInstance.on(MessageTopics.CONTEXT_UPDATE, msg => {
        console.log(msg);
      });

      eventCentreInstance.on(MessageTopics.HANDSHAKE_FAILED, () => {
        console.log("Handshake failed");
      });
    </script>
  </body>
</html>
```

Target Element

There may be occasions where it is preferable to pass in an element node instead of a string for the target destination of the event centre.

The `targetElementSelector` property can be either a string for the selection of a DOM element, or an element node itself.

```
const targetElement = document.querySelector('#img-arena-event-centre');
const eventCentreInstance = frontRowSeat.eventCentre({
  operator: "[OPERATOR-NAME]",
  sport: "ufc",
  version: "latest",
  eventId: "705",
  language: "en",
  targetModule: "full",
  targetElementSelector: targetElement
});
```

Authorisation

Each integrated customer receives their own unique IMG Arena Event Centre URL. To prevent potential web scraping, we frequently monitor for the unauthorised use of this URL and have the ability to shut down any suspicious usage.

Initialisation

To handle the synchronising of the UFC Event Centre with your site there are two methods exposed on the return value from the `eventCentre` initialiser: `on` and `emit`. Both these methods follow the pub/sub pattern for asynchronous message exchange. The `on` method is for receiving messages from the Event Centre, whereas the `emit` method is for sending messages to the Event Centre.

Subscribing

```
.on(MESSAGE_TOPIC_NAME, (messageData) => {
  // operate on received messageData, see below for returned fields for e
})
```

The `on` method is available on the return value from the `eventCentre` initialiser, it is used to subscribe to messages emitted from the Event Centre. The supported message topics are exposed on the `eventCentreUtils` namespace,

`eventCentreUtils.MessageTopics`, and are detailed below. The callback receives a single value which is an object containing the respective fields to the message topic subscribed to.

Emitting

```
// See below the fields to include on object passed for each topic
.emit(MESSAGE_TOPIC_NAME, messageData)
```

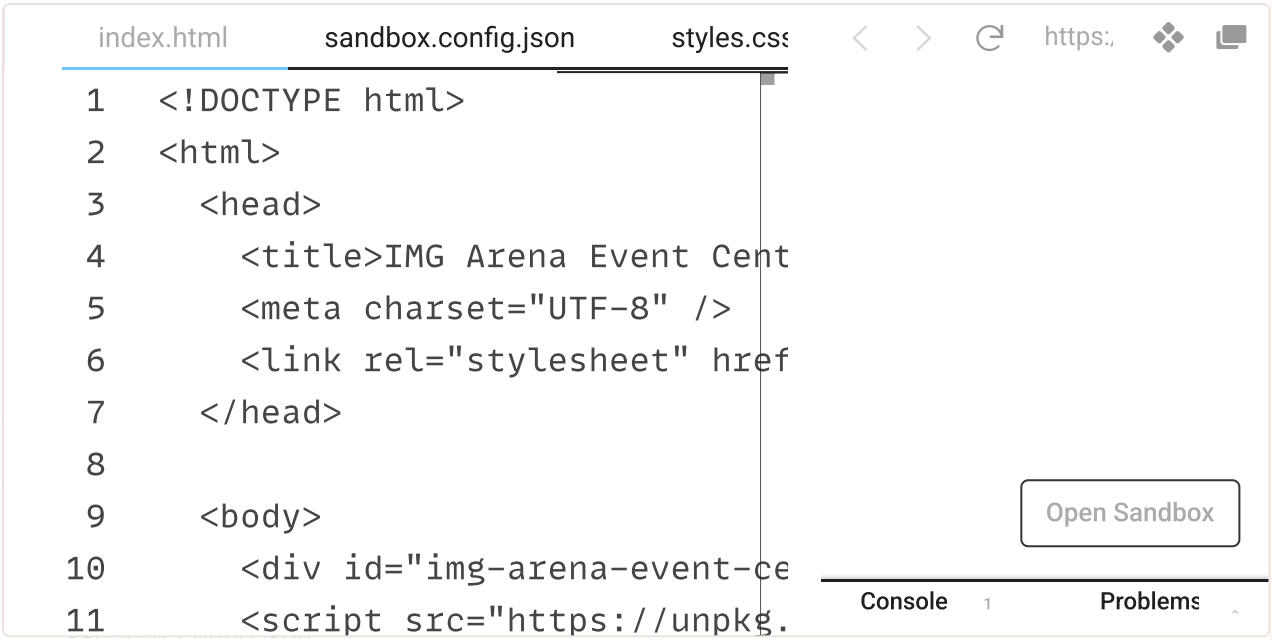
The `emit` method is available on the return value from the `eventCentre` initialiser, it is used to send messages to the Event Centre. Only supported message topics will be passed to the Event Centre.

Languages

As can be seen from the examples, 'language' can be set as a variable. Please see: Language Translations Available

Theme

The theme of the event centre can be configured by passing in a value for the `theme` parameter. `theme: "dark"` will display the dark-themed version of the event centre. Note that if you do not pass the `theme` parameter, the light version of the event centre will be rendered.



Options

Some additional options can be specified in the `options` field.

Option name	Description	Default	Example
`videoPlaybackEnabled`	Enables viewing broadcasts when they are available	false	true or false
`disablePeopleImages`	Disables images of people, showing placeholders instead	false	true or false
`userId`	Used for google analytics tracking	undefined	"123e4567"

SDK Integration

Android SDK

This sample app shows how to integrate the IMGA front row seat SDK into an Android application.

Supported IDEs:

- Android Studio

Supported build systems:

- Gradle

Adding the library to the project

Using Github Packages Maven repository

Add the repository to your build.gradle file

```
repositories {
     maven {
            url = uri("https://maven.pkg.github.com/IMGARENA/front-row-se
            credentials {
                username = "username"
                password = "password"
            }
        }
}
```

Add the package dependencies to your build.gradle file

```
dependencies {

...

implementation 'com.imgarena:eventcentre:1.0.2'
}
```

Adding the library locally to the project

Copy the eventcentre.aar to the libs folder of your application:

```
flatDir {
   dirs 'libs'
}
```

Then include the event centre in the dependencies section of your application:

```
dependencies {
   implementation(name:'eventcentre', ext:'aar')
}
```

Usage

Initialiasing SDK

Configuration parameters

```
var params: EventCentreParams = EventCentreParams(
    operator = "[OPERATOR-NAME]",
    sport = "ufc",
    version = "5.x",
    eventId = "667",
    targetModule = "full",
    language = "en"
)

// Optionally initialise a context
params.initialContext = mapOf(
    "view" to "Fight",
    "fightId" to "Insert fight ID",
)
```

Creating the Event Center instance

```
var imga = EventCentre(container, params, listener)
```

Event Centre: Emitting Messages

The **emitMessage** method is available on the instance of the eventCentre, it is used to send messages to the Event Centre. Only supported message topics will be passed to the Event Centre.

```
fun emitMessage(topic: EventCentreMessageTopics, message: EventCentreMess
```

The **EventCentreMessageTopics** enum contains the following values:

- CONTEXT_UPDATE: Topic for covering general UI state updates, for example navigation changes or the user selecting a player in the UI.
- SELECTION_UPDATE: Dedicated topic for handling user selection updates. The selected field within the message data handles both selecting and deselecting updates.

The **EventCentreMessage** type is an alias of a HashMap<String, Any> class. Keys must be Strings and the Values can be any type.

Updating the Context

There is a convinience method to update the context:

```
fun updateContext ( context: EventCentreContext) {
```

This method will update the context of the widget, and receives only the parameter context of type **EventCentreContext**. This type is a subclass of **EventCentreMessage** and contains the following properties:

Name.	Type
fightId	integer

Event Centre: Receiving Messages

In order to subscribe to different messages emitted by the SDK, the **EventCentreListener** interface must be implemented.

```
class TestActivity : AppCompatActivity(), EventCentreListener {
```

Subscribing to status changes

The SDK will send every change in its status using the following Listener method:

```
override fun onStatusChanged(status: EventCentreStatus) {
    Log.d("eventcentretest", "Transition to status: ${status.name}")

    Toast.makeText(this, status.toString(), Toast.LENGTH_SHORT).show()
}
```

The **EventCentreStatus** enum contains the following values:

- INITIALISING: the SDK is initialising and not fully loaded.
- INITIALIZED: the SDK is ready to interact.
- ACTIVE: All the views are fully loaded.
- EMITTING_MESSAGE: the SDK is emitting a message.
- INACTIVE: the SDK is closed and can't receive any message.

Subscribing to new messages

To receive context or selection updates from the SDK, the host app must subscribe to the Listener using this method:

```
override fun onMessageReceived(topic: EventCentreMessageTopics, message:
    Log.d("eventcentretest", "Message received: $message")
    Toast.makeText(this, message.toString(2), Toast.LENGTH_SHORT).show()
}
```

The **EventCentreMessageTopics** enum contains the following values:

- HANDSHAKE_ATTEMPT:

- HANDSHAKE_FAILED: Dedicated topic to signify the handshake between Event Centre and your site failed to complete. This topic is emitted by the integration library, it should only be subscribed to, not emitted.
- HANDSHAKE_REPLY:
- APP_LOADED: Topic sent when all the views are fully loaded.
- CLEAR_SELECTIONS:
- CONTEXT_UPDATE: Topic for covering general UI state updates, for example navigation changes or the user selecting a player in the UI.
- SELECTION_UPDATE: Dedicated topic for handling user selection updates. The selected field within the message data handles both selecting and deselecting updates.
- OPTIONS_UPDATE:
- ERROR: Topic sent if the SDK had any problem to be initialized. This topic is emitted by the integration library, it should only be subscribed to, not emitted.

Stop the sdk

```
// Instance the sdk
var imga = EventCentre(container, params, listener )

// Close the instance
imga.close()
```

Credits

IMGA SDK is owned and maintained by the IMGA Development team.

License

Private license

iOS SDK

MGA SDK for iOS is library for iOS providing a powerful high-level webview abstractions to interact with the IMGA web app.

Installation

IMGA SDK supports multiple methods for installing the library in a project.

Installation with CocoaPods

To integrate IMGA SDK into your Xcode project using CocoaPods, specify it in your `Podfile`:

```
pod 'IMGASDK', :git => 'git@github.com:IMGARENA/front-row-seat-ios-cocoap
```

Installation with SPM

To integrate IMGA SDK into your Swift project using SPM, specify it in your Package.swift

```
dependencies: [
    .package(url: "https://github.com/IMGARENA/front-row-seat-ios-cocoapo
],
```

Requirements

IMGA SDK requires iOS 14 and xCode 14.1

Usage

Creating an instance

```
let imga = IMGASDK(logLevel: .debug)
```

Initialiasing de SDK (launch)

Launching the SDK with raw data

```
let eventCentreData:IMGAData = [
            "operator": "[OPERATOR-NAME]",
            "sport": "ufc",
            "targetModule": "full",
            "eventId": "667",
            "language": "en",
            "version": "5.x",
            "targetElementSelector": "#img-arena-event-centre",
            "options":["videoPlaybackEnabled": "true"]
        ]
//Launch with the event centre data
try? imga.launch(with: eventCentreData) { imgaView in
    if let newView = imgaView {
        //Do something with the imgaView
    }
}
```

Launching the SDK with IMGAEventCentreParams object

```
let initialContex:IMGAData = [
          view: "Fight",
          fightId: "fightID"]

let eventCentreDataParams = IMGAEventCentreParams(operatorName: "operator
                                          sport: "sport",
                                          version: "version",
                                          eventId: "eventId",
                                          language: "language",
                                          theme: "theme",
                                          targetModule: "targetModule",
                                          initialContext: initialContex,
                                          options: nil,
                                          targetElementSelector: "targetEl

//Launch with the event centre data
try? imga.launch(with: eventCentreDataParams) { imgaView in
    if let newView = imgaView {
        //Do something with the imgaView
    }
}
```

Launching the SDK and append the IMGA View to a container

```
let initialContex:IMGAData = [
view: "Fight",
fightId: "fightID"]

let eventCentreDataParams = IMGAEventCentreParams(operatorName: "operator
                        sport: "sport",
                        version: "version",
                        eventId: "eventId",
                        language: "language",
                        theme: "theme",
                        targetModule: "targetModule",
                        initialContext: initialContex,
                        options: nil,
                        targetElementSelector: "targetElementSelector")

//Container where to add the IMGA view
let myContainer = UIView()

//Launch with the event centre data
try? imga.launch(andAddTo: myContainer, with: eventTest1)
```

Subscribing to status changes

```
//Instance the sdk
let imga = IMGASDK(logLevel: .debug)
//Subscribe to the sdk status changes
imga.onStatusChange = {status, error in
    print("Status:\(status)")
    if let e = error {
        print("Error on status:\(e)")
    }
}
```

Subscribing to new messages

There are several types of messages/topics

- `CONTEXT_UPDATE`: Topic for covering general UI state updates, for example navigation changes or the user selecting a player in the UI.
- `HANDSHAKE_FAILED`: Topic to signify the handshake between Event Centre and your site failed to complete.

```
//Instance the sdk
let imga = IMGASDK(logLevel: .debug)
imga.onNewMessage = {name, message, error in
    print("New Message:\(name)-\(message)")
    if let e = error {
        print("Error on message:\(e.description)")
    }
}
```

Stop the sdk

```
//Instance the sdk
let imga = IMGASDK(logLevel: .debug)
//Stop the instance
imga.stop()
```

Video Streaming Authentication

Please reference the doc here.

Credits

IMGA SDK is owned and maintained by the IMGA Development team.

License

IMG Media Limited

UFC EC Modules

The Event Centre supports being initialised with a specific UI state. This is achieved by passing in a "Context Update" message object, when initialising the Event Centre.

Modularisation

The UFC Event Centre supports being initialised in 7 different modular configurations. This is done by providing the module name within the "targetModule" field.

Module name	Description
full	Access to the full UFC Event Centre
fight	single fight
fightCard	Only displays the fightcard - allows no navigation
striking	Only displays the striking view - allows no navigation
stats	Only displays the stats view - allows no navigation
matchup	Only displays the matchup view - allows no navigation
tape	Only displays the tape view - allows no navigation

Message Topics

All the tables below detail the data fields present in the message data objects that are **sent and received** for each message topic.

Handshake Failed

Dedicated topic to signify the handshake between Event Centre and your site failed to complete. This topic is emitted by the integration library, it **should only be subscribed to, not emitted**.

Field	Type	Description	Required
timestamp	number	unix timestamp from time of failure	✅

Context Update

Topic for covering general UI state updates via the initialContext object, for example: navigating to a specific fight card or fight

Field	Description
view	Defines the view the event centre should load on. Possible values are fightcard / Fight / striking / stats / tape / matchup
eventId	Event ID as defined by the DDE.
fightId	Fight ID as defined by DDE. Available for view = fight/ striking / stats / tape / matchup

Full Fight Card

To embed the full UFC Event Centre, you must specify the following fields specific to your build.

Field	Value
operator	your unique operator Id name

sport	"ufc"
eventId	Event Id of the fight card
version	The version of UFC Event Centre.
theme	default = light version "light" = light version "dark" = dark version

- 'targetModule' specifies whether the app should display the full fight card, single fight, fightcard, striking, matchup, stats or tape.
- 'eventId' refers to the UFC fightcard/event that you wish to display.

Full Event Centre

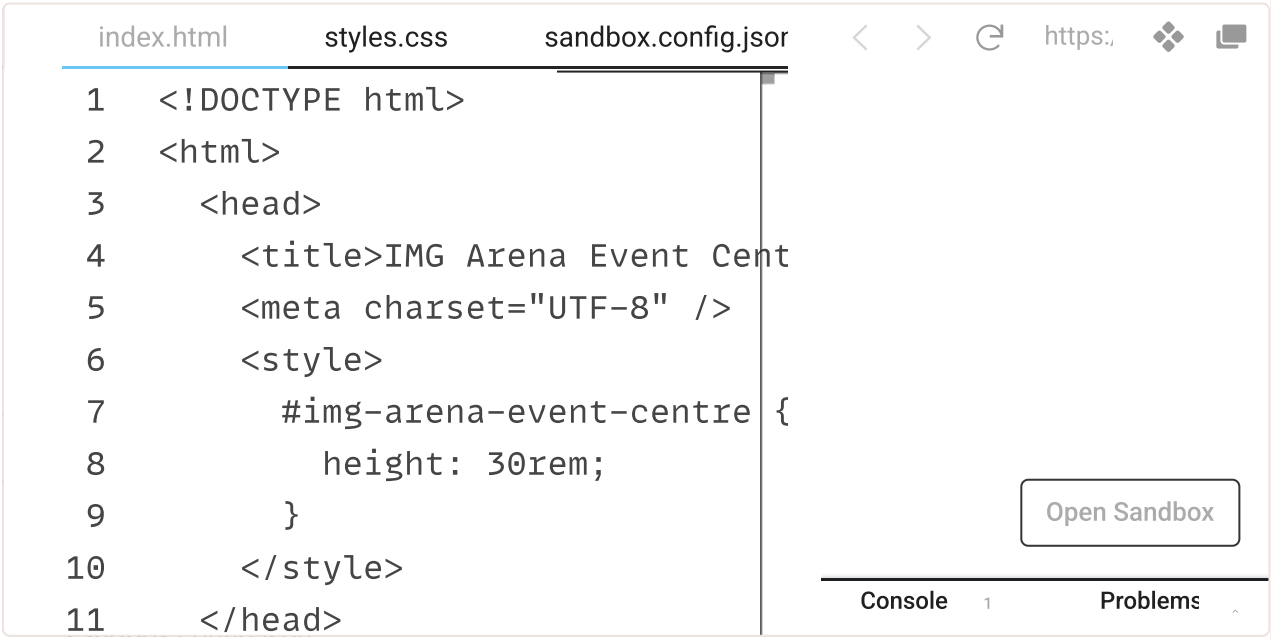
The following provides steps on how to implement an Event Centre with all features.

Steps to implement full EC:

- Add your unique operator ID name.
- Set targetModule as "full"
- Enter the eventId, note to use the unique ID for the fightcard, i.e UFC 254, event ID is "625".

```
const { MessageTopics } = frontRowSeat.eventCentreUtils;
const eventCentreInstance = frontRowSeat.eventCentre({
  operator: "operatorIDName",
  sport: "ufc",
  targetModule: "full",
  eventId: "eventID",
  version: "latest",
  targetElementSelector: "#img-arena-event-centre",
  language: "en"
});
```

This will grant full access to UFC Fightcard

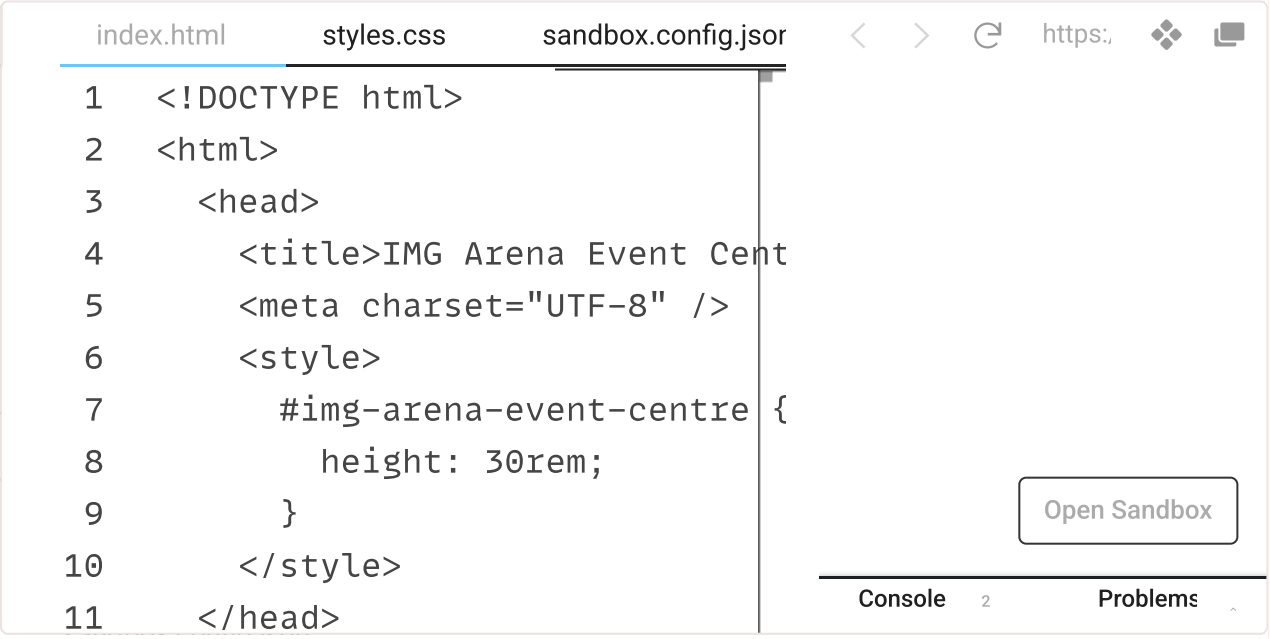


Fight Event Centre

The following provides steps on how to implement an Event Centre to a specific fight with or without navigation.

- Add your unique operator ID name.
- Set targetModule: "iframe".
- Enter the eventId, note to use the unique ID for the fightcard, i.e UFC FIGHT NIGHT, event ID is "707".
- Find the unique fightID that you would like to navigate to when the event centre is initially loaded, i.e on this instance fightID is "6667"
- Add the initialContext which consists of "fightId"
  - 'View' must be: "Fight" to allow navigation to full fight card
  - 'fightId' of a given fight.

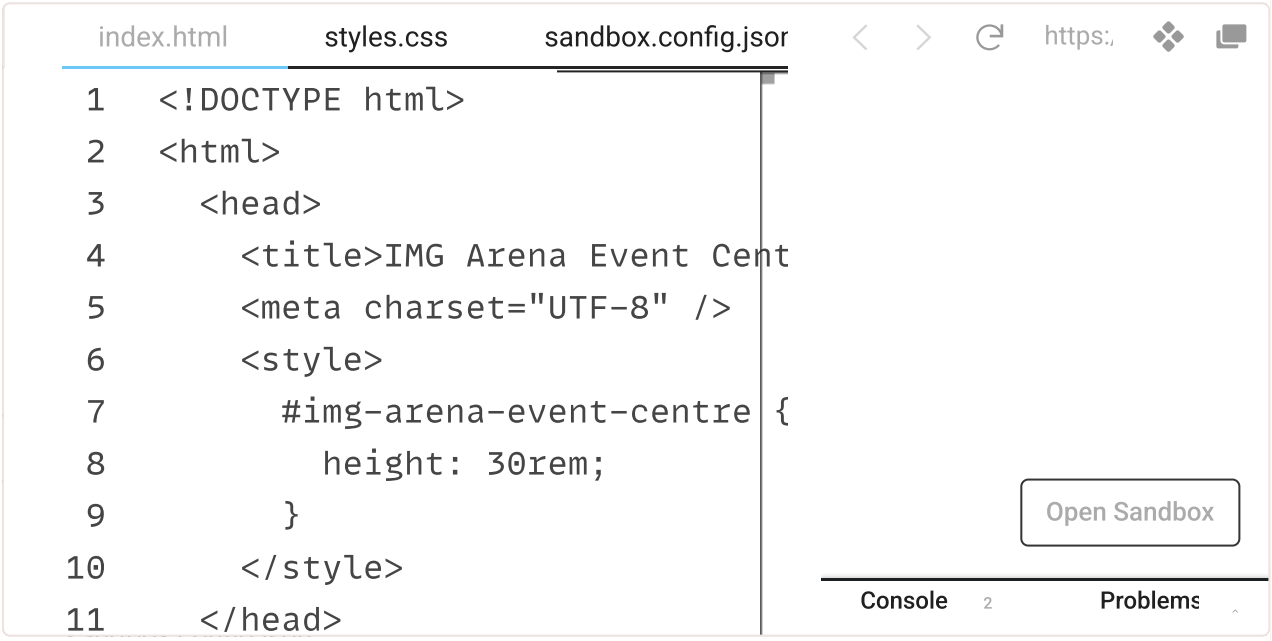
```
initialContext: {
          view: "Fight",
          fightId: "fightID"
        }
```



Sample code:

```
const { MessageTopics } = frontRowSeat.eventCentreUtils;
const eventCentreInstance = frontRowSeat.eventCentre({
        operator: "operatorIDName",
        sport: "ufc",
        targetModule: "full",
        eventId: "eventID",
        version: "version number",
        targetElementSelector: "#img-arena-event-centre",
        language: "en",
        initialContext: {
          view: "Fight",
          fightId: "fightID"
        }
      });
```

Note that if you do not want to allow navigation and want to limit view just for a particular fight, this can be done by passing "targetModule:" as "fight".



Note that there is no back button on the above event centre.

sample code:

```
const { MessageTopics } = frontRowSeat.eventCentreUtils;
const eventCentreInstance = frontRowSeat.eventCentre({
        operator: "operatorIDName",
        sport: "ufc",
        targetModule: "fight",
        eventId: "681",
        version: "dev/beac4ea5",
        targetElementSelector: "#img-arena-event-centre",
        language: "en",
        initialContext: {
          view: "Fight",
          fightId: "6667"
        }
      });
```

Fightcard Event Centre

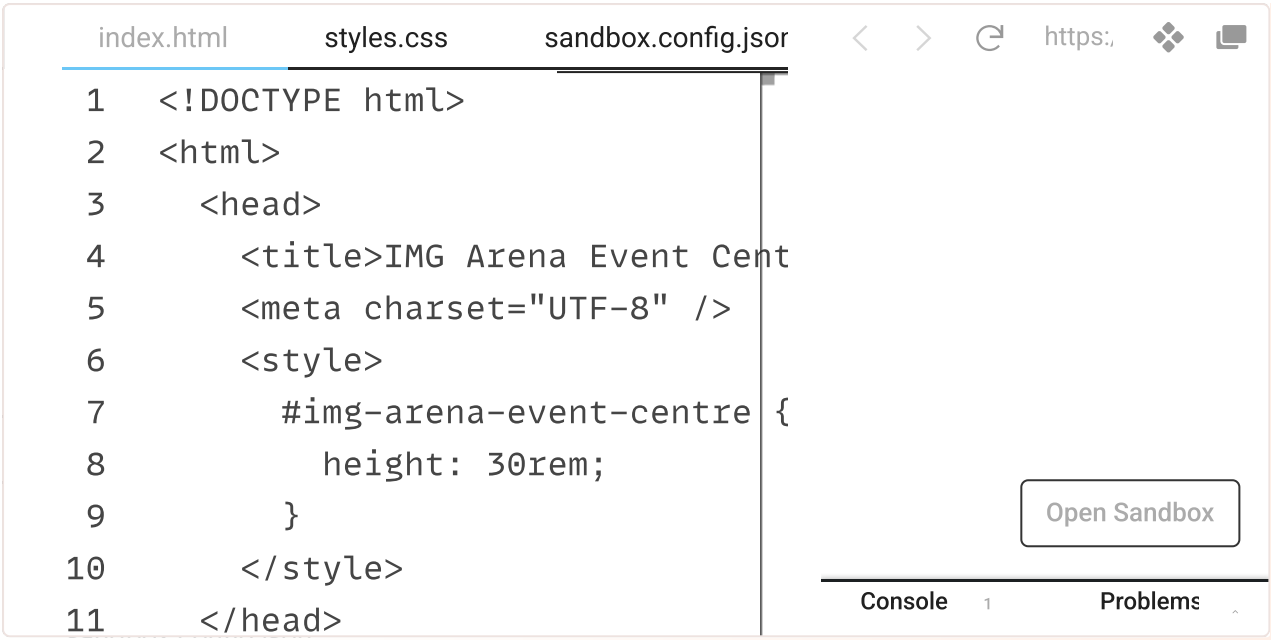
The following provides steps to embed just the fight card without navigation.

Steps to embed just the fight card - no navigation

- Add your unique operator ID.
- Set targetModule: "fightCard" (case sensitive).
- Enter the eventId, note to use the unique ID for the fightCard, i.e UFC Fight Night, event ID is "707".

sample code:

```
const { MessageTopics } = frontRowSeat.eventCentreUtils;
const eventCentreInstance = frontRowSeat.eventCentre({
        operator: "operatorIDName",
        sport: "ufc",
        targetModule: "fightCard",
        eventId: "eventID",
        version: "version number",
        targetElementSelector: "#img-arena-event-centre",
        language: "en"
      });
```


index.html
styles.css
sandbox.config.jsor
https:/
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>IMG Arena Event Cent
5 <meta charset="UTF-8" />
6 <style>
7 #img-arena-event-centre {
8 height: 30rem;
9 }
10 </style>
11 </head>
Open Sandbox
Console 1
Problems

Matchup Event Centre

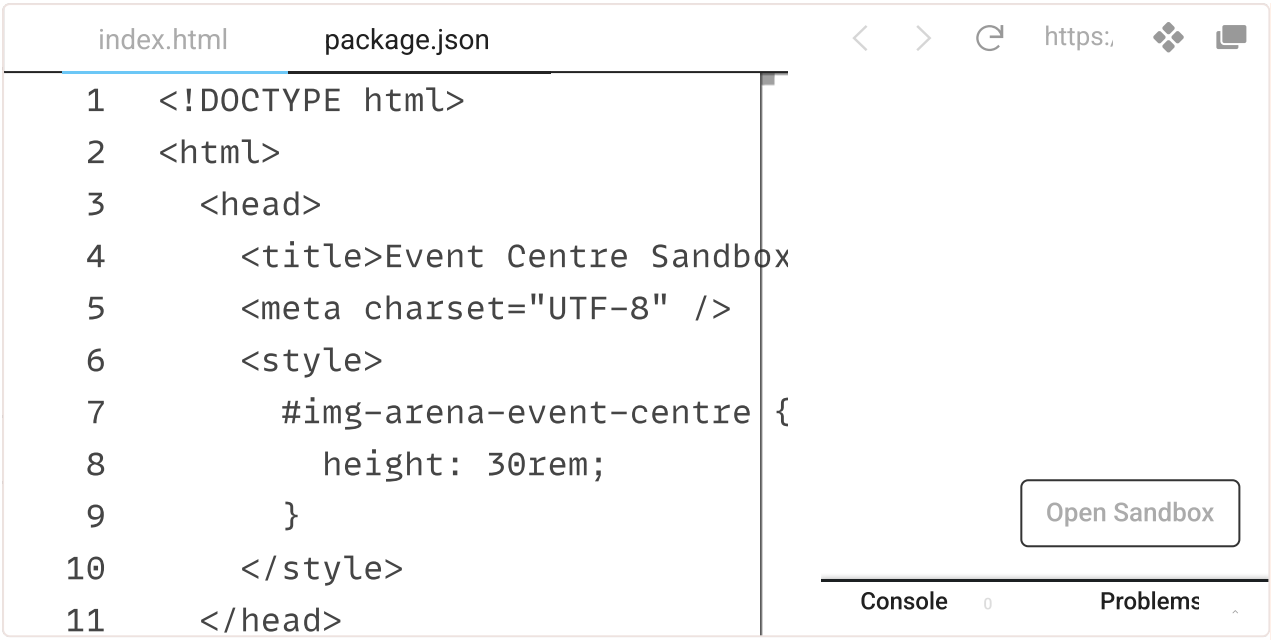
The following provides steps on how to implement an Event Centre to a specific matchup view without navigation.

Steps to render Matchup view - No navigation

- Add your unique operator ID.
- Set your targetModule: "matchup" (case sensitive).
- Enter the eventId, note to use the unique ID for the fightcard
- Find the unique fightID that you would like to navigate to when the event centre is initially rendered, e.g. 7121
- Add the initialContext which consists of "view" & "fightId"
  - 'View' must be: "matchup" to render the fightcard in matchup view
  - 'fightId' of a given fight.

Sample Code:

```
const { MessageTopics } = frontRowSeat.eventCentreUtils;
const eventCentreInstance = frontRowSeat.eventCentre({
        operator: "operatorIDName",
        sport: "ufc",
        targetModule: "matchup",
        eventId: "eventID",
        version: "version number",
        targetElementSelector: "#img-arena-event-centre",
        language: "en",
        initialContext: {
          view: "matchup",
          fightId: "fightID"
        }
      });
```


index.html
package.json
https:/
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Event Centre Sandbox
5 <meta charset="UTF-8" />
6 <style>
7 #img-arena-event-centre {
8 height: 30rem;
9 }
10 </style>
11 </head>
Open Sandbox
Console
Problems

Tape Event Centre

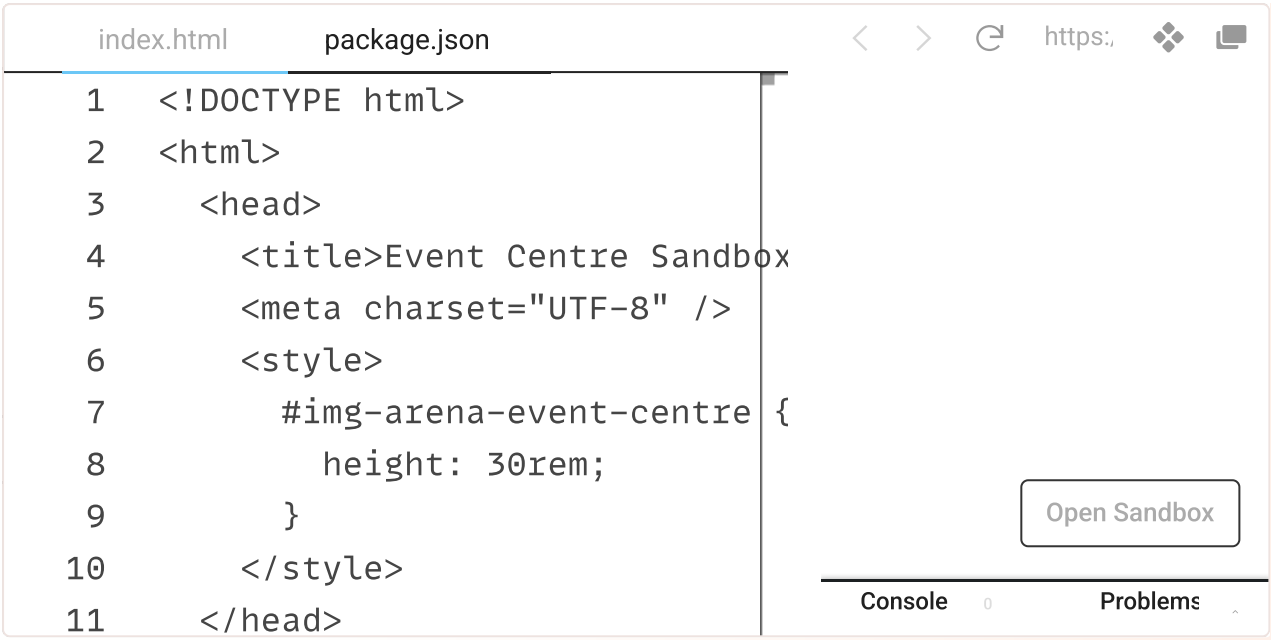
The following provides steps on how to implement an Event Centre to a specific tape view without navigation.

Steps to render Tape view - No navigation

- Add your unique operator ID.
- Set your targetModule: "tape" (case sensitive).
- Enter the eventId, note to use the unique ID for the fightcard
- Find the unique fightID that you would like to navigate to when the event centre is initially rendered, e.g. 7121
- Add the initialContext which consists of "view" & "fightId"
  - 'View' must be: "tape" to render the fightcard in tape view
  - 'fightId' of a given fight.

Sample Code:

```
const { MessageTopics } = frontRowSeat.eventCentreUtils;
const eventCentreInstance = frontRowSeat.eventCentre({
        operator: "operatorIDName",
        sport: "ufc",
        targetModule: "tape",
        eventId: "eventID",
        version: "version number",
        targetElementSelector: "#img-arena-event-centre",
        language: "en",
        initialContext: {
          view: "tape",
          fightId: "fightID"
        }
      });
```


index.html
package.json
https:/
<!DOCTYPE html>
<html>
<head>
<title>Event Centre Sandbox
<meta charset="UTF-8" />
<style>
#img-arena-event-centre {
height: 30rem;
}
</style>
</head>
Open Sandbox
Console
Problems

Stats Event Centre

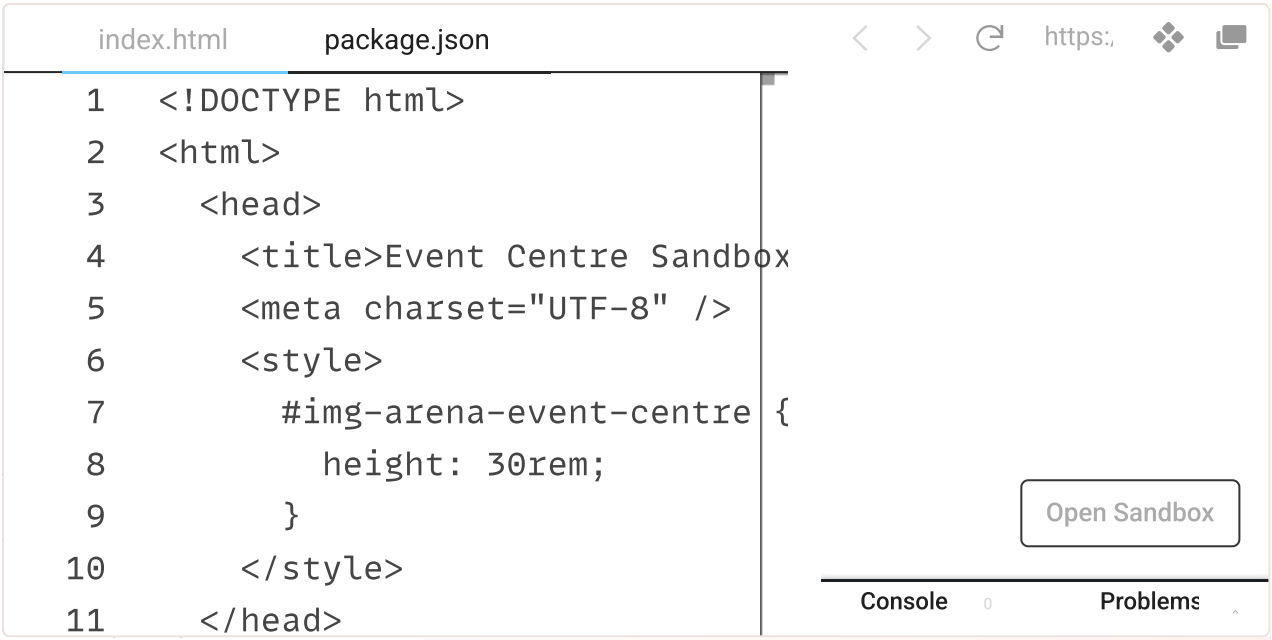
The following provides steps on how to implement an Event Centre to a specific tape view without navigation.

Steps to render Stats view - No navigation

- Add your unique operator ID.
- Set your targetModule: "stats" (case sensitive).
- Enter the eventId, note to use the unique ID for the fightcard
- Find the unique fightID that you would like to navigate to when the event centre is initially rendered, e.g. 7121
- Add the initialContext which consists of "view" & "fightId"
  - 'View' must be: "stats" to render the fightcard in stats view
  - 'fightId' of a given fight.

Sample Code:

```
const { MessageTopics } = frontRowSeat.eventCentreUtils;
const eventCentreInstance = frontRowSeat.eventCentre({
        operator: "operatorIDName",
        sport: "ufc",
        targetModule: "stats",
        eventId: "eventID",
        version: "version number",
        targetElementSelector: "#img-arena-event-centre",
        language: "en",
        initialContext: {
          view: "stats",
          fightId: "fightID"
        }
      });
```


index.html
package.json
https:/
<!DOCTYPE html>
<html>
<head>
<title>Event Centre Sandbox
<meta charset="UTF-8" />
<style>
#img-arena-event-centre {
height: 30rem;
}
</style>
</head>
Open Sandbox
Console
Problems

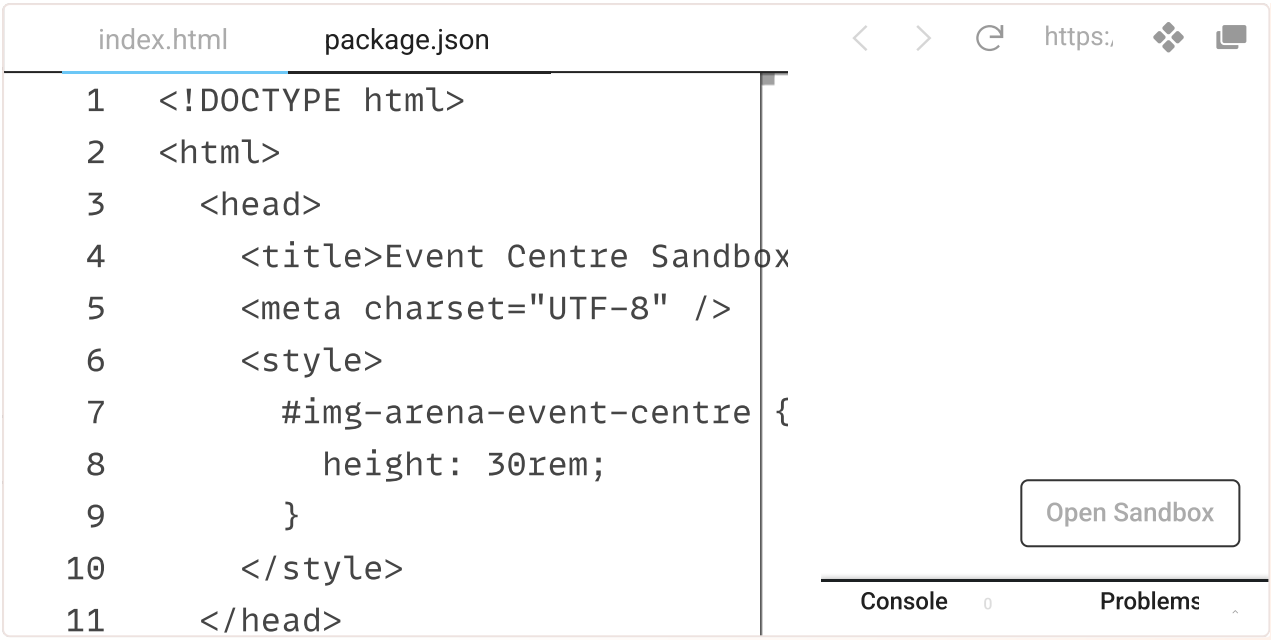
Striking Event Centre

The following provides steps on how to implement an Event Centre to a specific striking view without navigation.

- Add your unique operator ID.
- Set your targetModule: "striking" (case sensitive).
- Enter the eventId, note to use the unique ID for the fightcard
- Find the unique fightID that you would like to navigate to when the event centre is initially rendered, e.g. 7121
- Add the initialContext which consists of "view" & "fightId"
  - 'View' must be: "striking" to render the fightcard in striking view
  - 'fightId' of a given fight.

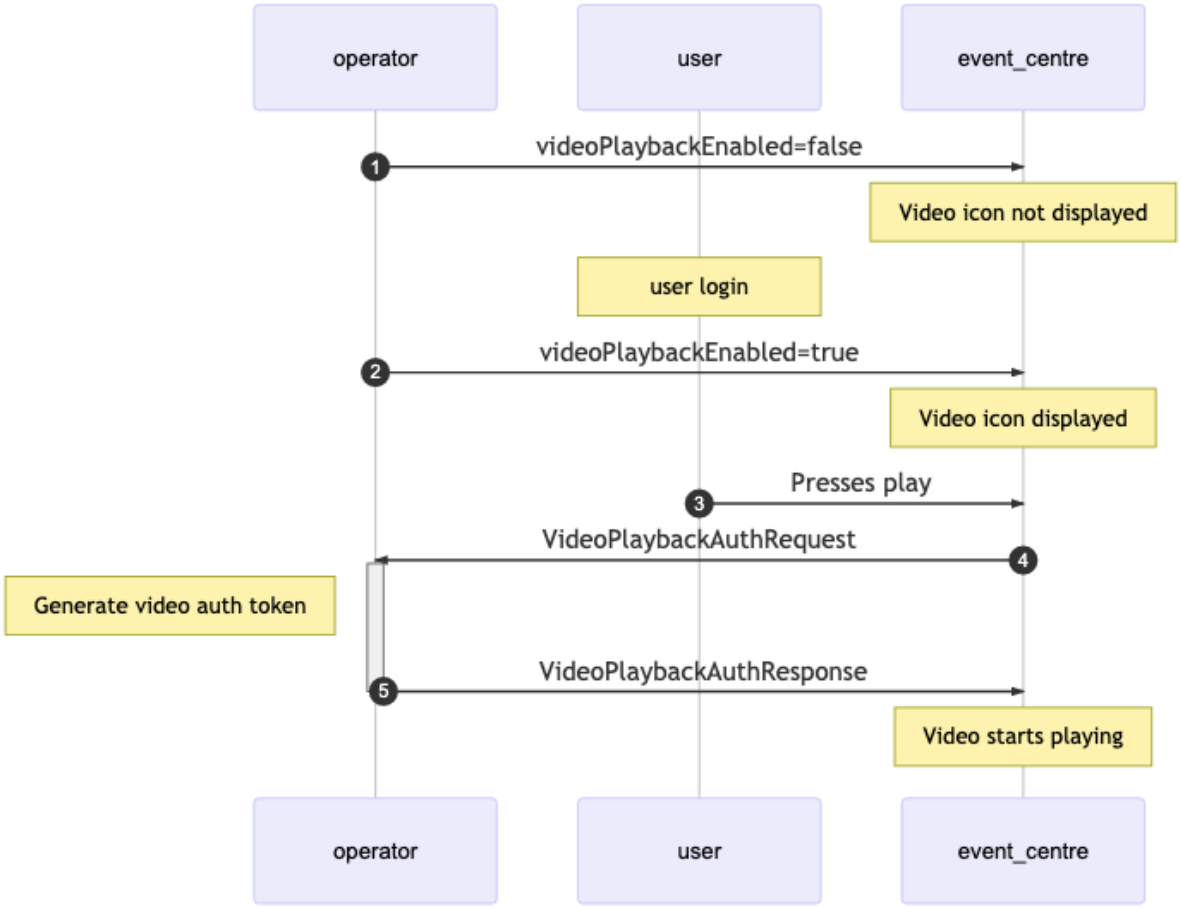
Sample Code:

```
const { MessageTopics } = frontRowSeat.eventCentreUtils;
const eventCentreInstance = frontRowSeat.eventCentre({
        operator: "operatorIDName",
        sport: "ufc",
        targetModule: "striking",
        eventId: "eventID",
        version: "version number",
        targetElementSelector: "#img-arena-event-centre",
        language: "en",
        initialContext: {
          view: "striking",
          fightId: "fightID"
        }
      });
```


index.html
package.json
https:/
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Event Centre Sandbox
5 <meta charset="UTF-8" />
6 <style>
7 #img-arena-event-centre {
8 height: 30rem;
9 }
10 </style>
11 </head>
Open Sandbox
Console
Problems

Video Stream Authentication

Overview of video authentication flow



An operator can set parameters when initialising the event centre to enable or disable video in the UI. The message displayed to users can be configured in advance.

```
frontRowSeat.eventCentre({
  ..
  options: {
    videoPlaybackEnabled: true/false
  }
)
```

it would also be possible to update this value at runtime

```
.emit(OptionsUpdate: {
  videoPlaybackEnabled: true/false
})
```

When a user presses play the event-centre will emit a message requesting the operator to authenticate the video playback.

```
.on(VideoPlaybackAuthRequest, {
  messageType: 'VideoPlaybackAuthRequest'
})
```

The operator should respond with the auth token using the existing guidelines described in IMG ALC Streaming documentation. It is important that the operatorId, and timestamp match those used to create the token. For those integratiing in the USA please reference IMG ALC US Streaming documentation.

```
.emit(VideoPlaybackAuthResponse, {
  messageType: 'VideoPlaybackAuthResponse',
  operatorId: '<OPERATOR-ALC-ID>'.
  auth: '<OPERATOR-ALC-TOKEN>',
  timestamp: 1576339325918
})
```

Video Stream Restrictions

Please note that streaming restrictions are present for the following territories

- Afghanistan
- Algeria
- Australia
- Bahrain
- Bangladesh
- Bhutan
- Brazil
- Canada
- Chad
- China
- Djibouti
- Egypt
- Georgia
- India
- Iran
- Iraq
- Jordan
- Kuwait
- Lebanon
- Libya
- Maldives
- Mauritania
- Morocco
- Myanmar [Burma]
- Nepal
- Netherlands
- New Zealand

- Oman
- Pakistan
- Palestine
- Qatar
- Saudi Arabia
- Somalia
- South Korea
- South Sudan
- Spain
- Sri Lanka
- Sudan
- Syria
- Tunisia
- United Arab Emirates
- United States
- Vietnam
- Yemen

Device/Browser Support Matrix

Please find the link below to our Device Browser Support Matrix

https://docs.imgarena.com/event-centres-common-docs/event-centres-common-docs/device-browser-support-matrix

Static Data

You will first need to create a database or gather a list of "fightcards" (a list of events) and the fights associated to the fightcard/event.

Required Headers to use our endpoints:

Key	Value
Accept	application/vnd.imggaming.dde.api+json;version=1
Content-Type	application/json
Authorization	Bearer eyvhaoudfgpdfgo*

In order retrieve a list of "fightcards" or events, please hit the IMG UFC fightcard endpoint:

`https://dde-api.data.imgarena.com/mma/fightcards`

Note the Key values needed from this endpoint:

Endpoint	Description	Link to Sandbox
fightcardName	the name of the fightcard	-
identifier	the identifier of the fightcard	EventId

If you want to look at just the information for a particular fightcard, you can pass the event ID i.e:

`https://dde-api.data.imgarena.com/mma/fightcards/{id}`

To return "fightcards" for a given date range, this can be done by passing the dateFrom and dateTo parameters. Both parameters should specify a date in a `YYYY-mm-dd` format.

In addition you will need to get a list of fights associated to the fightcard/event, this can be done by passing the EventId for a fightcard via the IMG Arena UFC "Schedule" endpoint.

`https://dde-api.data.imgarena.com/mma/fightcards/{id}/schedule`

Note the Key values needed from endpoint:

Endpoint	Description	Link to sandbox
redTeam	An object containing information on the fighter of the red team	-
blueTeam	An object containing information on the fighter of the blue team	-
fightId	The ID number of the fight	fightId

Fightcards Endpoint

This endpoint returns information on all fight cards, past and present.

Endpoint URLs

For all tournaments:

`https://dde-api.data.imgarena.com/mma/fightcards`

If you want to look at just information for a particular fightcard:

`https://dde-api.data.imgarena.com/mma/fightcards/{id}`

Required Headers

Key	Value
Accept	application/vnd.imggaming.dde.api+json;version=1
Content-Type	application/json
Authorization	Bearer eyvhaoudfgpdfgo*

Request Parameters

This endpoint can be requested using dateFrom and dateTo parameters together, to return just the fightcards within a certain date range. Both parameters should specify a date in a `YYYY-mm-dd` format.

Response Model

Tournament Object

Field Name	Type	Description
fightcardName	string	the name of the fightcard
identifier	integer	the identifier of the fightcard
countryCode	string	the country code of the country where the fightcard is taking place
location	string	the country in which the fightcard is taking place
state	string	the state/province in which the fightcard is taking place
city	string	The city in which the fightcard is taking place
venue	string	The name of the venue where the fightcard is taking place
venueId	integer	The ID of the venue where the fightcard is taking place
startDate	date	The start date for the fightcard
endDate	date	the end date for the fightcard
startTime	time	the time in which the fightcard starts, in local time + UTC offset format
utcOffset	integer	the UTC offset of the timezone in which the fightcard is taking place
year	integer	the year in which the fightcard is taking place

sport	string	the sport associated with the fightcard, in this case MMA
numberOfFights	integer	the number of fights on the fightcard
scheduleResource	string	a reference to the API URL for the schedule of this fightcard
status	string	the status of this fightcard. Potential values are: active / canceled.
booking Status	object	information on whether or not a fightcard is booked
comments	array	the latest comments on the status of the fightcard
numberOfFightsBookedToday	integer	Number of fights booked on the DDE as per your license agreement
eventSummary	object	Summary of preliminary and main cards i.e. start and number of fights
attendance	integer	Total fans in attendance
....		

Fights Summary Object

Each fightcard will be broken into three cards; main, prelims1 and prelims 2. This object, will indicate the date of the fightcard, and the three cards.

Field Name	Type	Description
fights	integer	the number of fights in this particular card of the fight

Booking Status Object

Field Name	Type	Description
status	string	Current booking status of the event on the DDE

Competitions Object

Field Name	Type	Description
organisation	string	the organisation of the particular fightcard
startDate	date	The start date for the fightcard
endDate	date	the end date for the fightcard
licensingProperty	string	The DDE property that licenses this particular fightcard

Sample Response

```
{
    "city": "Las Vegas",
    "startTime": "22:00",
    "location": "USA",
    "endDate": "2022-09-10",
    "fightcardName": "UFC 279: Chimaev vs. Diaz",
    "state": "Nevada",
    "identifier": 758,
    "numberOfFightsBookedToday": 13,
    "year": 2022,
    "sport": "MMA",
    "venueId": "50",
    "scheduleResource": "/mma/fightcards/758/schedule",
    "countryCode": "USA",
    "status": "active",
    "numberOfFights": 13,
    "eventsSummary": {
        "2022-09-10": {
            "prelims1": {
                "fights": 4,
                "startTime": "00:00"
            },
            "prelims2": {
                "fights": 4,
                "startTime": "22:00"
            },
            "main": {
                "fights": 5,
                "startTime": "02:00"
            }
        }
    },
    "utcOffset": 0,
    "attendance": null,
    "startDate": "2022-09-10",
    "bookingStatus": {
        "status": "AutoBooked"
    },
    "venue": "T-Mobile Arena",
    "competitions": [
        {
            "organisation": "UFC",
            "startDate": "2022-09-10",
            "endDate": "2022-09-10"
        }
    ],
    "comments": []
}
```

Fightcards Schedule Endpoint

This endpoint returns information on the schedule of a particular fightcard.

This endpoint returns information on the schedule of a particular fightcard. Information returned includes: Detailed information on the fighters of each team, Estimated start time, Fighter Records & Weight class.

Endpoint URLs

```
https://dde-api.data.imgarena.com/mma/fightcards/{id}/schedule
```

Required Headers

Key	Value
Accept	application/vnd.imggaming.dde.api+json;version=1
Content-Type	application/json
Authorization	Bearer eyvhaoudfgpdfgo*

Request Parameters

This endpoint takes no parameters

Response Model:

Fightcard Schedule Object

Field Name	Type	Description
startTime	object	an object containing “status” – the status of the start time, e.g. EstimatedStart, and “time”, the estimated start time, in local + UTC offset format
weightClass	object	An object containing information on the weight class
referee	object	Contains ID, First and last name of the referee for the fight.
startTimeText	string	Text description of the start time. Note that the start time object contains detailed information about the start and is likely to be more useful than this field. Main values; Starts At, Follows Previous
fightOrder	integer	The fight’s order on the card. Fight number 1 is the main event, 2 is the co-main event and so on, so that the first fight of the night has the highest number. The main event is always 1
fightcardId	integer	the ID of the fightcard in which the fight is in
accolades	object	An object containing information on the belt & the name of the belt title
fightSeq	integer	The fight sequence on the overall fightcard. 1 if first, 2 if second etc.
redTeam	object	An object containing information on the fighter of

		the red team
Date	string	the date of the fight
blueTeam	object	An object containing information on the fighter of the blue team
status	string	the status of the fight. Possible values; “Not Started”, “In Progress” or “Finished”
cardSegment	string	The segment of the card that this fight belongs to. Options; main, prelims1, prelims2
fightId	string	The ID number of the fight
fightType	object	Contains information on the type of fight, total number of possible rounds
bookingStatus	object	information on whether or not a fight is booked

startTime Object

Field Name	Type	Description
status	string	Status of the current start time e.g. "EstimatedStart"
time	string	The start time of the fight + UTC offset format

Weightclass Object

Field Name	Type	Description

weight	string	the weight range, in lb, of the weightclass
description	string	description of the weightclass
id	integer	ID of the weightclass
obsolete	Boolean	Details whether the weightclass is still active within the organisation

Referee Object

Field Name	Type	Description
id	integer	the ID of the referee
firstName	string	the first name of the referee
lastName	string	the last name of the referee

Accolades Object

Field Name	Type	Description
belt	string	the type of belt
name	string	the title of the belt

Red Team Object

Field Name	Type	Description
fighter1	object	detailed information on the fighter under the red team
rank	string	the fighters current rank. (may appear null if the

		fighter is not ranked)
accolades	object	-

fighter1 Object

Field Name	Type	Description
reach	decimal	the reach, in inches, of the fighter. Will be null if unknown.
dob	date	the date of birth of the fighter
fightsOutOf	string	the city, state, country, country code of where the fighter currently fights out of
height	integer	the height, in inches, of the fighter. Will be null if unknown.
country	string	country code of the fighter
age	integer	the age of the fighter
weighInWeight	float	the weigh in weight, in lb, of the fighter. Will be null if unknown.
lastName	string	the last name of the fighter
stance	string	the stance of the fighter. Possible values; Orthodox, Southpaw, Switch, Open. Will be null if unknown.
firstName	string	the first name of the fighter
nickName	string	the nickname of the fighter
id	integer	the id of the fighter

born	string	the city, state, country, country code of where the fighter was born
record	object	win/loss/draw/no contest record of the fighter
accolades	string	will be null unless a title holder. If not null, will

Record Object

Field Name	Type	Description
wins	integer	the amount of professional wins a fighter has
losses	integer	the amount of professional losses a fighter has
draws	integer	the amount of professional draws a fighter has
noContest	integer	the number of professional contests the fighter has participated in

Blue Team Object

Field Name	Type	Description
fighter1	object	detailed information on the fighter under the blue team
rank	string	the fighters current rank. (may appear null if the fighter is not ranked)
accolades	object	appears to be null- even for champions

fightType Object

Field Name	Type	Description
possibleRounds	integer	the number of possible rounds
description	string	description on the number of rounds

bookingStatus Object

Field Name	Type	Description
status	string	-

Sample Response

```
{
    "startTime": {
        "status": "EstimatedStart",
        "time": "04:00Z"
    },
    "weightClass": {
        "weight": "156-170",
        "description": "Welterweight",
        "id": 16,
        "obsolete": false,
        "abbreviation": "WW"
    },
    "referee": null,
    "startTimeText": "Follows Previous",
    "fightOrder": 1,
    "fightcardId": 758,
    "accolades": null,
    "fightSeq": 13,
    "redTeam": {
        "fighter1": {
            "reach": 75.0,
            "dob": "1994-05-01",
            "fightsOutOf": "Chechnya, RUS",
            "height": 74,
            "country": "RUS",
            "age": 28,
            "weighInWeight": null,
            "lastName": "Chimaev",
            "stance": "ORTHODOX",
            "firstName": "Khamzat",
            "nickName": "Borz",
            "id": 3457,
            "born": "Benoy-Yurt, Chechnya, RUS",
            "record": {
                "wins": 11,
                "losses": 0,
                "draws": 0,
                "noContests": 0
            }
        },
        "rank": 3,
        "accolades": null
    },
    "date": "2022-09-10",
    "blueTeam": {
        "fighter1": {
            "reach": 76.0,
            "dob": "1985-04-16",
```

```
                "fightsOutOf": "Stockton, California, USA",
                "height": 72,
                "country": "USA",
                "age": 37,
                "weighInWeight": null,
                "lastName": "Diaz",
                "stance": "SOUTHPAW",
                "firstName": "Nate",
                "nickName": null,
                "id": 274,
                "born": "Stockton, California, USA",
                "record": {
                    "wins": 21,
                    "losses": 13,
                    "draws": 0,
                    "noContests": 0
                }
            },
            "rank": null,
            "accolades": null
        },
        "status": "Upcoming",
        "cardSegment": "main",
        "fightId": 7399,
        "fightType": {
            "possibleRounds": 5,
            "description": "5 Rnd (5-5-5-5-5)"
        },
        "bookingStatus": {
            "status": "AutoBooked"
        }
    },
```

Data Frequency for Endpoints (RESTful)

We recommend the following frequency for your REST requests to ensure that you have the most up-to-date information at all times:

Sport	REST Endpoint	Frequency
UFC	`dde-api.data.imgarena.com/mma/fightcards`	Every 12hours
UFC	`https://dde-api.data.imgarena.com/mma/fightcards/{id}/schedule`	Every hour for the upcoming fightcard

Please note that only the next 3 or 4 future events will be available via the UFC fightcards endpoint. This is due to UFC still constructing the fights taking place.

Release Management

Event Centres

Event centres are directly integrated into Sportsbook websites & apps. IMG Arena's release process is designed to reduce the amount of ongoing integration work we ask our partners to perform - whilst also retaining the option to support sportsbooks wishing to make updates part of their internal QA process.

Versioning

All releases are published using SEMVER and follow the pattern:

`<major>.<minor>.<patch>`

`major`: Contains either breaking changes or significant UI updates that may require review.

`minor`: Does not contain any breaking changes, but may include feature enhancements & refactoring that does not change the core experience

`patch`: Essential bug fixes required for the correct function of the app only

A sportsbook can embed a version of the event centre linked to a SEMVER path e.g.

`5.2.1` locked to a patch release, no changes possible. [**not recommended**]

`5.2.x` no minor changes but will update to include essential patch releases

`5.x` no major (breaking) changes but will update to include feature enhancements and essential patch releases. [**recommended**]

There are some special tags that allow shortcuts to latest versions:

`latest` the latest stable release of the app

`beta` experimental branch used to preview an upcoming major release (unstable)

`dev` development branch (unstable)

Support Framework

We make considerable efforts not to introduce breaking changes, but from time to time this may be necessary due to underlying changes for our data suppliers for example.

All Changes will be published with comprehensive release notes.

`major` releases can be expected quarterly, at which point the previous release will enter a 3 month end-of-life period.

`minor` releases can be expected monthly, minor releases are supported for the duration of the major release.

`patch` releases are ad-hoc and should be applied to assure the continued function of the application.

Native Integration

iOS implementation

The suggested method of implementation is to create a webview with the initialisation code before implementing bridge function(s) between the webview and native code. Sample code can be found below

```
import UIKit
import SwiftUI
import PlaygroundSupport
import WebKit

class MyViewController : UIViewController, WKUIDelegate, WKScriptMessageH

    var webView: WKWebView!
    var embed = """
    <!DOCTYPE html>
      <html>
        <body>
          <div id="img-arena-event-centre" style="height:400px;width:100%

          <script src="https://unpkg.com/@img-arena/front-row-seat@0.x/di

              <script>
                const { MessageTopics } = frontRowSeat.eventCentreUtils;
                const eventCentreInstance = frontRowSeat.eventCentre({
                  operator: "operatorId",
                  sport: "ufc",
                  targetModule: "full",
                  eventId: "668",
                  version: "4.1.1",
                  targetElementSelector: "#img-arena-event-centre",
                  language: "en"
                });

                eventCentreInstance.on(MessageTopics.CONTEXT_UPDATE, func
                        // send message to native controls
                    window.webkit.messageHandlers.contextUpdate.postMessa
                });
              </script>
        </body>
      </html>
    """


    override func loadView() {
        let webConfiguration = WKWebViewConfiguration()
        webConfiguration.preferences.javaScriptEnabled = true
        webConfiguration.userContentController.add(self, name: "contextUp

        webView = WKWebView(frame: .zero, configuration: webConfiguration
        webView.uiDelegate = self
        let view = webView

        self.view = view
```

```
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        webView.loadHTMLString(embed, baseURL: nil)
    }

    func userContentController(_ userContentController:
        WKUserContentController, didReceive message: WKScriptMessage) {

        if (message.name == "contextUpdate") {
            // handle update
            print(message.body)
        }
    }
}

PlaygroundPage.current.liveView = MyViewController()
```

Language Translations Available

The list below shows the languages we currently have available. We are able to add any other language upon request.

Languages

- English (en_GB)
- Bulgarian (bg)
- Chinese Simplified (zh_CN)
- Chinese Traditional (zh_TW)
- Croatian (hr)
- Czech (cs)
- Danish (da)
- Dutch (nl)
- Estonian (et)
- French (fr)
- French Canadian (fr_CA)
- German (de)
- Greek (el)
- Hindi (hi)
- Hungarian (hu)
- Indonesian (id)
- Italian (it)
- Japanese (ja)
- Kazakh (kk)
- Korean (ko)
- Latvian (lv_LV)
- Lithuanian (lt_LT)
- Norwegian (no)
- Polish (pl)

- Portguese (pt)
- Portguese (Brazilian) (pt_BR)
- Romanian/ Moldovian (ro)
- Russian (ru)
- Serbian (Latin, Bosnia and Herzegovina) (sr_Latn_BA)
- Slovak (sk)
- Slovenian (sl)
- Spanish Castilian (es)
- Spanish Latam (es_419)
- Swedish (sv)
- Telugu (te)
- Thai (th)
- Vietnamese (vi)

FAQ

Please view the following list of frequently asked questions, hopefully it can help you integrate the UFC event centre more efficiently:

How do I start the integration?

We recommend that you first try connect to the /fightcard endpoint and retrieve "identifier", i.e eventID and the associated "fightcardName" i.e the event name to the eventID.

i.e you can connect to /fightcard endpoint and pass a date rage for the year. https://dde-api.imggaming.com/mma/fightcards?dateFrom=2022-01-01&dateTo=2022-12-31

i.e "identifier": 703 - "fightcardName": "UFC 271: Adesanya vs. Whittaker"

How do I load the UFC event centre via sandbox:

Connect to the /fightcard endpoint and it will return a list of events +/- 7 days, retrieve the "identifier" for an event that you would like to load the event centre for. Then update the eventId with the "identifier" value in the integration library,

i.e loading UFC 271: Adesanya vs. Whittaker by passing event ID: 703.

Sandbox example:

https://codesandbox.io/s/full-fight-card-forked-ow2mmu

How do I build static data?

We recommend that you first pull the /fightcard endpoint and review the data. Note to pass dateFrom and dateTo to return events against a date range. i.e: return all UFC events starting 2022 and ending 2022. i.e https://dde-api.imggaming.com/mma/fightcards?dateFrom=2022-01-01&dateTo=2022-12-31

Once you have the events Ids, you can pull the /schedule for the fightcard to get a list of matches within an event.

After this you can start mapping events.

How do I map IMG Arena ID to Internal systems?

To map at event level, we recommend that you pull the /fightcard endpoint and use "fightcardName", "startDate", "city" and apply fuzzy logic parser to map events as the name of the fightcard and start date will be universal. Note to store the "identifier" on your systems, as this ID will be used as the EventID in the integration library.

After mapping at fightcard level, you can start by mapping at fight level. This can be done by hitting the /schedule endpoint. i.e https://dde-api.imggaming.com/mma/fightcards/703/schedule
From this endpoint we recommend that you pull "fightId", "fightOrder", "fightSeq" and the "redTeam" and "blueTeam"

Which static data should I save and why?

from the /fightcard endpoint:

Parameter	Definition	Purpose
fightcardName	Name of the fight card/event	Name of the event is universal and will help with

		mapping to internal systems.
identifier	unique ID for the event	This ID will be used as the "eventId" int he integration library.
venue	name of the stadium of where the event is	useful info
startDate	start date of the tournament	you can use this for mapping the event
startTime	start time of the tournament/first fight.	To help determine when to display the event centre on your website.

→ static data to build would be:

-The name of a event,
-When it starts
-Unique IMG Arena eventID

After having this information, you can start to build the static data for a particular event:

From the /fightcard/{}/schedule endpoint

Parameter	Definition	Purpose
startTime	expected starttime of the fight.	Note that these are placeholder times.
weightClass	Weightclass of the fight	useful info to help with mapping.
fightSeq	returns the sequence of when the fight is scheduled for. i.e "fightSeq": 1 will be the first fight of the night	Helps with the ordering of the fights
cardSegment	determine which part of the fightcard the fight is under	useful info to help with mapping

redTeam	Who is in the red corner	useful info to help with mapping
blueTeam	Who is on the blue corner	useful info to help with mapping
fightId	unique fightId for the fight	This Id will be needed if you will be using the different target module when using

→ static data to build would be:

-Order of fights within an event
-The lineup for an event is, i.e RedTeam Vs blueTeam and its unique fightID

How do I know when an event starts?

You can use the /startTime returned in the /fightcard endpoint to determine when the event will start.

When can I start to display the UFC event centre?

Depending on your integration and how the other product work on your website. You can either display the event centre at the start of the week before the event, i.e on a Monday. Or if you have a in-play section, when the event starts. This is really depended on your website.

When should I stop displaying the UFC event centre?

We recommend that you stop displaying the event centre a few hours after the event. But this is depended on your website. Note that you can access an event 1

week after it has finished.

How do I know if an event has streaming:

We stream all UFC events.

When does streaming begin and end for a fight?

Currently for UFC there is 1 long stream. For the event centre, it is broken up by fight. The stream will start as players are walking out and the stream will be cut off when the winner is announced.

How to I escalate issues?

During integration; please escalate all queries to the Integration/CSM team.

Once you are live: Please escalate all queries to our support team: support@openbet.com.

Our support team are available 365 a year 24/7

What are the SLAs?

For the response time for support:

- 15 minutes for the first response
- 15 minutes for the next update
- 30 minute intervals for any further updates thereafter

What are the supported size of the iframe?

We recommend that you use a minimum height of 362px and width of 320px. Reducing the height might misalign text and buttons.

What is the quality of the stream?

Currently due to broadcasting ri-ght; all our streams run 360p (728 Kbps)

How do I do geo-blocking for streaming?

Geo-blocking for streaming is carried out at operator level on the IMG Arena side. We suggest speaking to your dedicated Customer Solution Manager for more information on this.

How should I go about testing streaming?

Unfortunately, currently you will have to test UFC streaming when the events is live.

Note that you will just have to authenticate streaming via integration library.

Can we get a list of fighters?

We have a database containing basic information on all fighters. You can access the following endpoint to retrieve this data: `https://dde-api.imggaming.com/mma/fighters`

if you require more details, please contact IMG Arena integration team.

What can I customise within the Iframe?

No we do not allow customisation for the UFC event centre as we follow the official UFC branding. We support a Dark and Light theme.