Gameflex Universal Launcher (GUL)
Integration Guide

Client:	Iforium
Date:	17th June 2021
Revision:	1.8
Location:	GameflexUniversalLauncher-IntegrationGuide1.8.docx
Security Classification	Confidential
Distribution List:	Gameflex Operators

Iforium Ltd
The Old Courthouse
2nd Floor, Athol Street
Douglas
Isle of Man IM1 1LD
Tel: +44 (0) 1624 822933
Email: info@iforium.com
Web: www.iforium.com

Change History

Version	Date	Author	Summary of Changes
1.0	7th August 2018	Matthew Keggen	Initial Release
1.1	9th October 2018	Matthew Keggen	Added origin to handshake
1.2	4th September 2019	Matthew Keggen	Support added for mobile.
1.3	12th February 2020	Matthew Keggen	Added autoplay command
1.4	9th July 2020	Matthew Keggen	Added scale change and iframe features.
1.5	31st July 2020	Luke Babich Matthew Keggen	Add scroll top event for provider scaling game providers.
1.6	16th September 2020	Matthew Keggen	Tidy up launch parameters.
1.7	24th September 2020	Matthew Keggen	Add missing Table ID for Live Casino launches.
1.8	17th June 2021	Luke Babich	Added in reload balance command

Table of Contents

1 Preface

This document describes how Operators can launch Gameflex Game Content using the Gameflex Universal Launcher.

3 Background

Gameflex Universal Launcher providers Operators with a single Url to launch any game within the Gameflex Platform.

When choosing to integrate Operators may choose to provide their own Regulatory Dialogs or use the built-in functionality provided by Gameflex. This decision will be agreed between Iforium and the Operator before the integration commences.

All Game Provider Content can be loaded in iframes on both desktop and mobile and the integration steps for both are the same.

Operators using the Gameflex Express Library (GEL) will continue to use the Gameflex Engage Object to handle this logic.

4 Data Mapping

4.1 Brands

A brand per market will be configured (e.g. UK/Malta), however in some circumstances there could be multiple UK brands.

Usually, a player belongs to a single brand and on game launch, the **casinoid** parameter will identify the players brand.

Iforium will define the **casinoid** values which are needed when launching games.

4.2 Game IDs

Every Game within the Gameflex Platform is given a unique identifier based on its Desktop/Mobile variant.

The Operator will be required to store and send the Gameflex Game ID.

4.3 Table IDs

When using a live casino product, it may also be necessary to pass a table id on the launch. This will allow the player to launch direct to a table instead of a live casino lobby.

4.4 ISO Standard Codes

Gameflex uses ISO Standard Identifiers for Language, Currency and Country codes.

4.5 Device Channels

Device Channels are used by Gameflex to identify the platform where the game is being loaded. Gameflex will apply additional logic and requirements for certain platforms to ensure the best experience is delivered to the player.

Value	Usage
web	Use when launching games on desktop or mobile web browsers such as Chrome/Safari.
iosWeb	Use when launching games within an iOS Native App where the game assets are loaded from the remote server.
iosApp	Use when launching games within an iOS Native App where the game assets are bundled locally within the App. *N.B. Additional launch parameters are required for this approach and only certain game providers currently support game bundling.*
androidWeb	Use when launching games within a Native Android App where the game assets are loaded from the remote server.
androidApp	Use when launching games within a Native Android App where the game assets are loaded from the remote server. *N.B. Additional launch parameters are required for this approach and only certain game providers currently support game bundling.*

5 Integration Approaches

5.1 Desktop

5.1.1 Iframe

Generally, the best player experience on desktop is to embed an iframe to the game on the operator website.

Using this approach:

- Game lifecycle must be implemented.
- Operator website will be responsible for removing the game and/or redirecting the player back to the lobby when receiving the close game event.

5.1.2 New Window

Should the operator website require launching the game into a new window this should be discussed with Iforium so that the correct configuration is setup prior to integration.

5.2 Mobile

5.2.1 Redirect

The Operator Website may choose to redirect the player directly to the Gameflex GUL URL. As there is no wrapper around GUL the game lifecycle cannot be used.

Using this approach:

- Game lifecycle cannot be used.
- The Operator Website must supply the **lobbyurl** parameter which will be used to return the player to the lobby when the game has ended.
- Gameflex built-in support for full screen tap or swipe to play is enabled.

5.2.2 Iframe

The Operator Website may choose to embed Gameflex GUL URL within an iframe on the mobile lobby.

Using this approach:

- Game lifecycle must be implemented.
- Operator website will be responsible for removing the game and/or redirecting the player back to the lobby when receiving the close game event.
- Gameflex built-in support for full screen tap or swipe to play are disabled. It is up to the Operator Website to handle this functionality in their wrapper.

5.3 Regulated Markets

Gameflex supports a wide range of Regulated Markets and these features are automatically enabled for operators.

Should the operator website take responsibility for displaying regulated market requirements, the iframe approach will be required on both desktop and mobile games. The game life cycle must also be implemented.

To disable the built-in regulation engine the parameter **regulationsenabled=false** should be added to the launch parameters.

6 Universal Launcher

The operator will be provided with a **BaseUrl** to launch any Desktop or Mobile game.

When a game is launched it can be sent as a HTTP GET or POST.

6.1 Game Launch Parameters

6.1.1 Standard Parameters

Parameter names sent to Gameflex are always done in **lowercase** to avoid any casing issues.

Name	Type	Example	Comments
casinoid	String	S009-IFO-20	Identifier supplied by Iforium to represent a single brand within the Operator Website
sessiontoken	String		Should be a single-use token (e.g. UUID) This token is redeemed on the wallet to identify the player. Refer to the API Integration Guide for further information. *N.B. This parameter can be omitted when launching for **demo**.*
gameid	String	723	See section 4.2 Game IDs
languagecode	String	en	See section 4.4 ISO Standard Codes
playmode	String	real | fun | demo	When using **real/fun** require sessiontoken to handshake player account whilst launching the game. When using **demo** an anonymous launch occurs where no Player data is transferred.
channelid	String	desktop | mobile	
devicechannel	String	web	See section 4.5 Device Channels
lobbyurl	Url	https://www.operator.com/lobby	URL Gameflex will redirect the player to when leaving the Game. *N.B. This parameter is always sent even if the game life cycle is being used. This is used as a fallback in case of any unknown errors during launch.*

6.1.2 Optional

Name	Type	Example	Comments
tableid	String	abc123	See section 4.3 Table IDs.
cashierurl	Url	https://www.operator.com/cashier	URL Gameflex will redirect the player to should they leave the Game to go the Banking/Cashier/Account area. *N.B. This parameter is optional and if not specified will default to the **lobbyurl** value.* *This parameter can be omitted when launching for **demo**.*
currencycode	String	EUR	ISO Standard Codes. *N.B. Only required when performing a **demo** launch and will set the Game currency for the demo mode.*
regulationsenabled	Boolean	false	Can be used to disable Gameflex built-in Regulation Engine.

6.2 Regulated Market Parameters

Iforium will confirm additional parameters required for each market.

7 Game Display

The operator website is responsible for the position and display of the iframe used to load the Gameflex game.

Using the data returned in the initialise or ready events the operator can determine the scaling mode for the game.

7.1 Auto Scaling (desktop or mobile)

Operator website can set their iframe to any size and the game will automatically fit that area.

```
dimensions: {
  'scalingMode': 'auto-scaling'
}
```

N.B. All games on mobile use this approach. On desktop, only Inspired Virtual Sports uses this scaling mode.

7.2 GEL Scaling (desktop only)

On desktop games many Operators choose to overlay the game in a floating div on their lobby. When taking this approach, it is important the iframe/containers to the game are sized based on the aspect ratio of the game. This avoids dead-space and other scaling issues.

N.B. All slot games use this scaling approach.

```
dimensions: {
  'scalingMode': 'gel-scaling',
  'aspectRatio': 1.6623376623376624,
  'defaultHeight': 770,
  'defaultWidth': 1280
}
```

7.3 Provider Scaling (desktop only)

Provider Scaling is mainly used by virtual Sports Book products where the height of the content is dynamic and dependent on the sport/markets the player is viewing. Operator website can set the width of the iframe/container to any size that suits the website and should listen for the Scale Change Event to adjust the height dynamically.

```
dimensions: {
  'scalingMode': 'provider-scaling'
}
```

8 Game Launch Lifecycle

The game launch lifecycle must be implemented when hosting games within an iframe on desktop or mobile.

The Operator Website will generate the Gameflex Universal Launcher URL. This URL is then loaded within an Iframe embedded on the operator website.

All games are loaded using the Gameflex Game Wrapper which uses our Gameflex Express Library (GEL) to control communication to and from the game.

This wrapper provides a consistent approach for all games and will take care of any game provider specific integration requirements giving Operators a single launch and integration point.

8.1 Game Launch Handshake

A handshake is required from GEL to the Operator Website which will secure future communication to and from the game.

8.1.1 Initialise Event

GEL will send this message to the Operator Website as soon as it has loaded. This message will instruct the Operator Website that GEL has initialised and is ready to continue the game launch.

The message includes the **origin** of the Gameflex Game Window. The operator website can use this to secure future postMessage events to GEL.

```
{
  method: 'gel.initialise',
  data: {
    origin: 'http://www.gameflexorigin.com'
  }
}
```

8.1.2 Operator Ready Command

This event should be sent when the Operator Website is ready to continue the game launch process and should only be sent after receiving the Initialise Request.

The Operator Website should also send its own origin with the event so that future calls from GEL can be secured.

```
gameflexIframeWindow.postMessage({
  method: 'operator.ready',
  data: {
    origin: 'https://www.operatorwebsite.com'
  }
}, gameflexOrigin);
```

8.2 Game Events & Commands

Following a successful launch of the game the operator website will start receiving events from the game and will be able to send commands to control features within the game.

Events are emitted from GEL to the operator website. Commands are sent from the operator website to GEL.

8.2.1 Game Loading

Every game will emit an event which indicates it has started loading. This event will also include the dimensions of the game which can be used to correctly scale the iframe on the operator website.

8.2.1.1 Example

```
{
  method: 'gel.loading',
  data: {
    game: {
      'gameID': 123,
      'gameProviderID': 9,
      'title': 'Thunderstruck II'
    },
    dimensions: {
      'scalingMode': 'gel-scaling',
      'aspectRatio': 1.6623376623376624,
      'defaultHeight': 770,
      'defaultWidth': 1280,
      'maximumHeight': null,
      'maximumWidth': null,
      'minimumHeight': null,
      'minimumWidth': null
    }
  }
}
```

8.2.2 Game Ready

Once the game has completed loading the game ready event is sent. Some providers send this during the load phase of the game however, most providers send this when the game has reached 100% loading. This event will also include the dimensions of the game which can be used to correctly scale the iframe on the operator website.

8.2.2.1 Example

```
{
  method: 'gel.ready',
  data: {
    game: {
      'gameID': 123,
      'gameProviderID': 9,
      'title': 'Thunderstruck II'
    },
    dimensions: {
      'scalingMode': 'gel-scaling',
      'aspectRatio': 1.6623376623376624,
      'defaultHeight': 770,
      'defaultWidth': 1280,
      'maximumHeight': null,
      'maximumWidth': null,
      'minimumHeight': null,
      'minimumWidth': null
    }
  }
}
```

8.2.3 Game Dimension Events

8.2.3.1 Scale Change

This event is emitted on games using the **provider-scaling** mode. The height should be used on the operator website to adjust the iframe/containers.

```
{
  method: 'gel.scale.change',
  data: {
      'height': 1540,
    }
  }
}
```

8.2.4 Spin Events/Commands

8.2.4.1 Spin Start Event

When a player clicks spin or deal in a table game a spin start event is emitted.

8.2.4.1.1 Example

```
{
  method: 'gel.spin.start'
}
```

8.2.4.2 Spin End Event

Spin end is usually sent after the player has seen the final animation of the spin outcome.

8.2.4.2.1 Example

```
{
  method: 'gel.spin.end'
}
```

8.2.4.3 Stop Autoplay Command

Slot games will have an autoplay feature allowing the player to spin consecutively without interruption. Sending this command will allow the Operator to break the autoplay cycle at the end of the current spin.

Operators should use this when presenting regulatory dialogs.

8.2.4.3.1 Example

```
gameflexIframeWindow.postMessage({
  method: 'operator.autoplay.stop'
}, gameflexOrigin);
```

8.2.5 Game Enable/Disable Commands

Operators may require disabling the game until a user action has been completed, for example, a reality check. Game features such as the spin button, autoplay and bet/coin sizes within the game will be disabled until re-enabled.

8.2.5.1 Disable Example

```
gameflexIframeWindow.postMessage({
  method: 'operator.game.disable'
}, gameflexOrigin);
```

8.2.5.2 Enable Example

```
gameflexIframeWindow.postMessage({
  method: 'operator.game.enable'
}, gameflexOrigin);
```

8.2.6 Game Session End

Game Session End is the most complex and important part of the game lifecycle.

When ending a game certain logic must be applied by Gameflex to correctly tear down the game session. Also, if the Operator is using Gameflex to display regulated market prompts, in the Spanish market for example, we are required to display a Session Summary when the game session ends.

All mobile games will include a home button or navigation option for the player to leave the game. The operator may still choose to include their own close button around the game and on desktop most websites will include a close or "X" button in the game container.

Should the Operator Website include any of these features it must call this command and not just remove the game from the DOM or redirect to another page.

GEL will receive this command and apply any logic to end the game session. Once complete GEL emit the ***Game Close Event*** to the Operator Website to indicate the game can be closed allowing the iframe to be removed or redirection to another page.

8.2.6.1 Example

```
gameflexIframeWindow.postMessage({
  method: 'operator.game.session.end'
}, gameflexOrigin);
```

8.2.7 Game Reload Balance Command

This event is optional.

After a game has launched, if you are wanting to update the balance within the game you can trigger a reload balance command. This sends an event down to the game to trigger a reload balance with the most up to date value. This can be implemented if the operator has implemented an in-game cashier.

8.2.7.1 Example

```
gameflexIframeWindow.postMessage({
  method: operator.game.balance.reload
}, gameflexOrigin);
```

8.2.8 Game Close

Before the Operator can remove the game from the DOM or redirect to another page it should listen for the Game Close event.

This event is sent to the Operator Website in these scenarios:

- Player has clicked on the Home button within the game
- Operator End Session Command was triggered and game can now be closed
- Gameflex Regulated Market Engine is being used and player has interacted with a dialog to close the game.
- Game has errored and must be closed

8.2.8.1 Example

```
{
  method: 'gel.close '
}
```

8.2.9 Game Scroll To Top

This event is optional.

When a game is launched using **provider-scaling** mode the game will dynamically change the height of the window. When GEL displays regulatory dialogs it will attempt to scroll the window back to the top. In some cases, depending on the Operator Website, GEL may not be able to scroll the window. The operator website should consume this event and scroll the window itself.

8.2.9.1 Example

```
{
  method: 'gel.scroll.top '
}
```

Common Issues

8.3 Game Launch Spinner Stuck

When launching a game and the spinner continually spins and the game does not launch this is generally that the operator website has not sent the **operator.ready** event to GUL.

Until that event has been received the game will not be instructed to load.