Technical Partner Onboarding

Introduction

Gambling.com Group (GDC) partnership AdTech integrations can utilize one of three different implementation options to provide operator lists, exit pages and tracking tags for partner sites.

1. Client Side JS Simple Include
2. Client Side JS Advanced Includes
3. Server Side JSON Feed

If you are a partner where GDC develops and maintains a partner microsite that requires reverse proxy please review the details for Proxying a GDC Microsite.

Each partner should review performance considerations as part of your integration planning to ensure your site is fast.

1. Client Side JS Simple Include

Simple AdTech integration by the partner into their existing site using GDC client-side JS integration to inject CTAs specific to a partner that is also tracked by GDC AdTech.

GDC Group ads can be displayed on partner sites by adding the following to relevant pages:

1. A javascript library that takes care of initializing the adtech and injecting the relevant ads in the page.
2. One or more HTML snippets throughout the page which serve as mounting points for the ads.

Adtech Javascript Library

Each partner site has a dedicated javascript library. The up-to-date URL for each library can be found on the AdTech dashboard for each site. The partner AdTech URL should use the partner's domain (i.e. bet.partnerdomain.com) so that ads aren't blocked by AdBlockers.

The example library URL is visible at the top of each page:

```
https://<partner-adtech-domain>/js/app.js
```

That URL can be added as follows at the bottom of the <body> tag of each page that is supposed to display an ad:

```
<script type="text/javascript" src="https://<partner-adtech-domain>/js/app.js"></script>
```

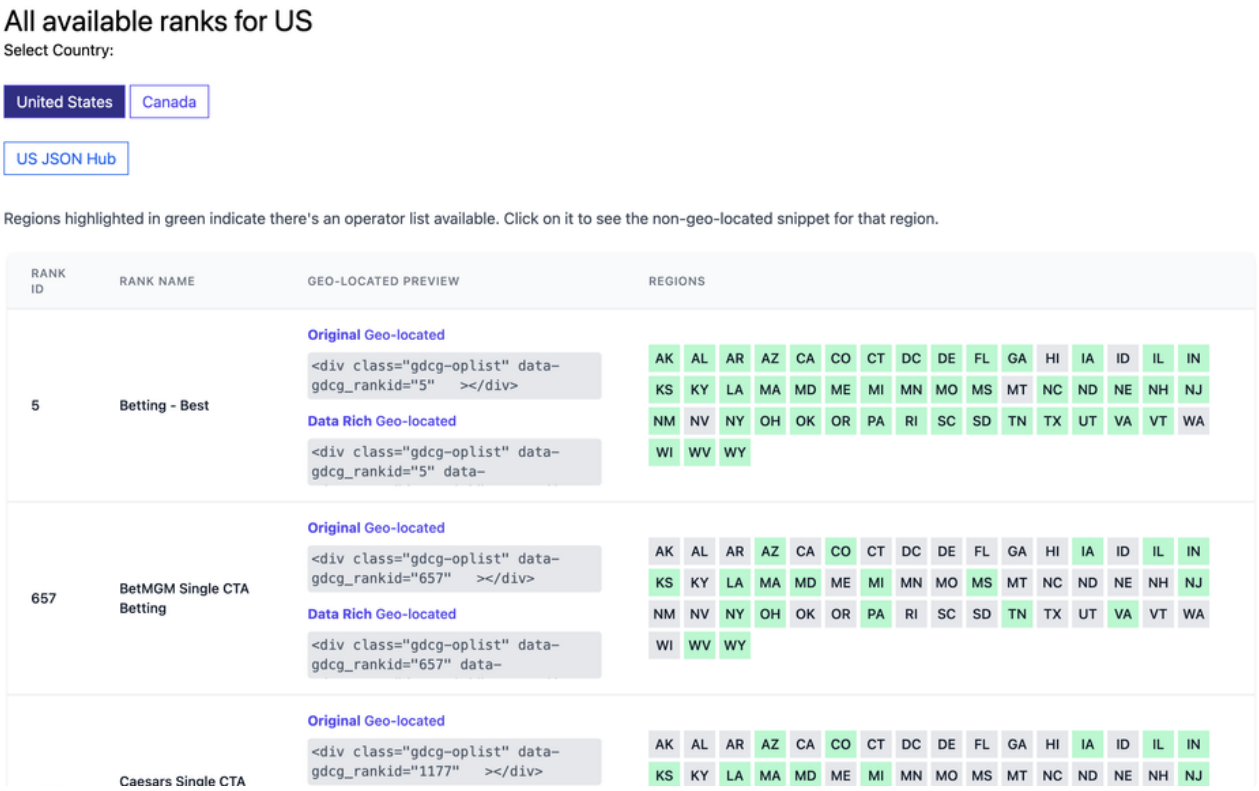
HTML snippets

HTML snippets serve as mounting points for the several ads that can be added to a page.

The available ads are:

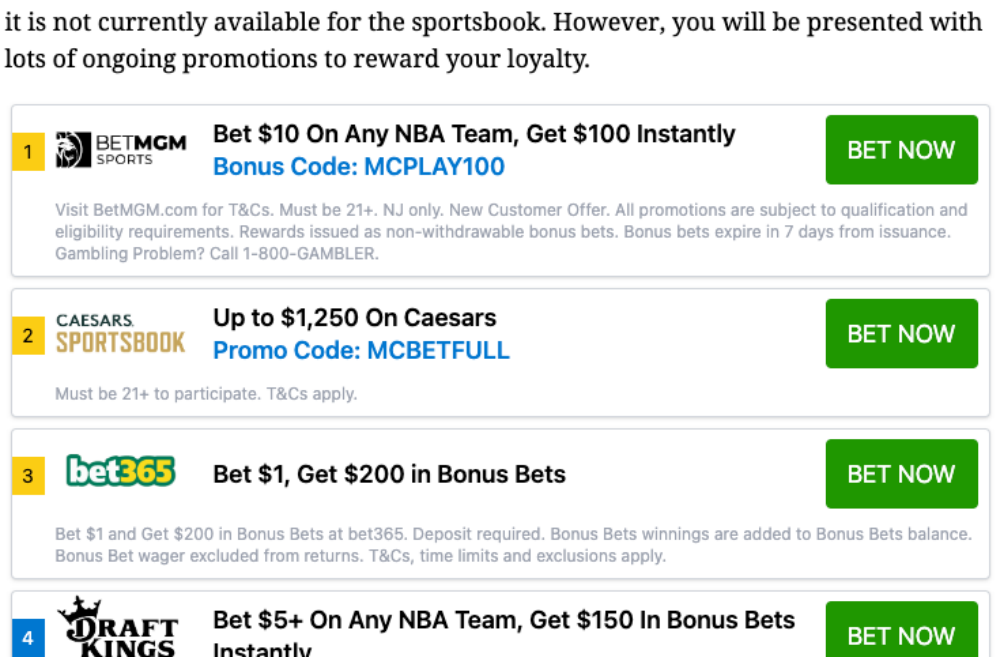
- Geo-located operator list
- Non-geo-located operator list

All snippets available for each site are visible in the corresponding Adtech dashboard. In the image below, inside the first gray text areas, are the HTML snippets for geo-located operator lists:



For an oplist to appear on a page it is enough to copy/paste a snippet within a page's HTML.

A sample oplist is visible below.



Looking at the source code of the page it's possible to see the adtech snippet:

```
<div class="gdcg-oplist" data-gdcg_rankid="5"></div>
```

The snippet is just an empty DIV tag with a class and a few data attributes.

When the widget is initialized, the adtech endpoint returns the HTML for each DIVs mounting point. Such HTML is injected and becomes the inner HTML for each of the DIVs.

We currently have two types of operator list:

- Original
- Data Rich

They are defined using the `data-gdcg_type` attribute. The only type that currently needs to be defined is Data Rich: `data-gdcg_type="data-rich"`. When no type is defined it will render the Original oplist as default.

All data attributes starting with `data-gdcg_` define the parameters for the ad that should be displayed. The ones currently available are:

- data-gdcg_rankid - integer identifying an oplist cluster
- data-gdcg_country - 2 characters country code
- data-gdcg_region - 2 characters region code (US states or Canada regions)

Mandatory parameters:

- `class="gdcg-oplist"` - used by the javascript code to find oplists mounting point on the pages
- `data-gdcg_rankid` - used to define the rank of the oplist

Optional parameters:

- `data-gdcg_type` - oplist frontend styling
- `data-gdcg_country` and `data-gdcg_region` - used to define if an oplist is geo-dynamic or static. If there are none specified, the oplist is geo-dynamic.

Wordpress Shortcode

If you use Wordpress as a CMS, you can use Wordpress Shortcodes instead of HTML snippets. This option needs to be enabled by GDC upon partner request. Shortcodes have the exact same parameters as HTML snippets, syntax is the only difference:

```
[gambcom-region rankid="644" country="us" region="ca"]
```

Adtech Javascript Library - How it works

The Adtech Javascript library has a small footprint - at the time of writing 3.6kb gzipped - but it loads extra resources needed for the adtech to work. When loaded it does the following:

1. Loads ktag.js our own internal oplist analytics library
2. Loads the css file needed for oplists to display correctly
3. Scans the page's html searching for ads mounting points
   a. For each DIV mounting point, it collects all parameters from the data attributes included in the HTML snippet
   b. Sends a GET request to the adtech endpoint including all relevant parameters. The endpoint is implemented through a Cloudflare Worker which can geo-locate the request and use that information to select the best response for the user.
   c. Injects the HTML received by the endpoint inside the corresponding DIV
   d. Runs ktag.reRender() to make sure all CTAs included in the oplist will fire click events that can be tracked by ktag

2. Client Side JS Advanced Includes

AdTech integration by the partner using a combination of JS includes. The partner site will not use the single app.js file that implements the logic but rather the 3 files below which can be added by the partner as preferred in their source code.

- ktag.js - enables tracking
- app.css - styles oplists

- app-native.js - enables simple behavior on oplists such as click-to-copy for promo codes on data-rich oplists and polyfill for container queries.

Adtech snippets

The advanced JS integration will use the same snippet currently available. For example:

```
<div class="gdcg-oplist" data-gdcg_rankid="5"></div>
```

That same snippet will not be parsed by javascript, but by partner site logic in their UI or backend. Using a snippet in a different format will be possible, if it can make easier for the backend to parse it.

The important part is that the following parameters are preserved:

- rankid - integer value
- type - string (currently either empty or “data-rich“)
- country - 2 chars code
- region - 2 chars code

Rendering a snippet

The partner system will be responsible for implementing the following logic.

Build the URL to retrieve the oplist that corresponds to the snippet. The URL can be built as follows:

```
https://<partner-adtech-domain>/private/${country}/${region}/rank/${rankId}/${type}
```

If `country` and `region` are not provided in the snippet, they must be retrieved by any geo-location service used by the backend system. The response from the oplist URL is a fragment of HTML representing the oplist requested. The fragment can be injected into the page so will be rendered in the browser for the user.

Fallback oplist

Oplists may not exist for all combinations of `country` and `region`. When an oplist is missing, the above URL would return 404. In this case, the backend system must implement fallback logic to display an alternative oplist. Currently, the fallback is to display a US/NJ oplist which is always available for all ranks. So the above URL can become:

```
https://<partner-domain>/private/us/nj/rank/${rankId}/${type}
```

Constraints

The HTML returned by the URLs above is fixed: the partner system has no control over it. This means that responsibility for look and feel, and new oplist styles still lives with the GDC Adtech team. On the other hand, it makes the integration very straightforward. Ktag tracking and exit pages are still handled by the GDC Adtech system and the partner doesn’t need to re-implement all that logic.

3. Server-Side JSON Feed

AdTech integration by the partner using server-side CTA JSON feeds where the partner can parse required data elements and create custom CTA HTML output that utilizes defined exit pages and tracking. This approach gives the ability for the partner to style oplists as needed and avoid page shift (CLS) that affects SEO and experience.

To ensure proper tracking, ktag.js will have to be added by the partner to their source code.

The partner can parse through a JSON feed categorized by Country

/us/json-hub

/ca/json-hub

/uk/json-hub

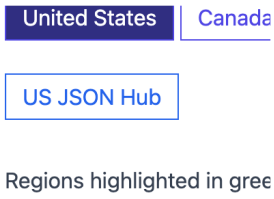
or by individual oplist separated out by region and rank ID.

/private/us/al/rank/5/json

/private/us/ct/rank/2791/json

/private/us/tn/rank/1922/json

To see these feeds there is a JSON Hub button on the homepage



This hub lists out all single endpoints sorted by region for easy access.

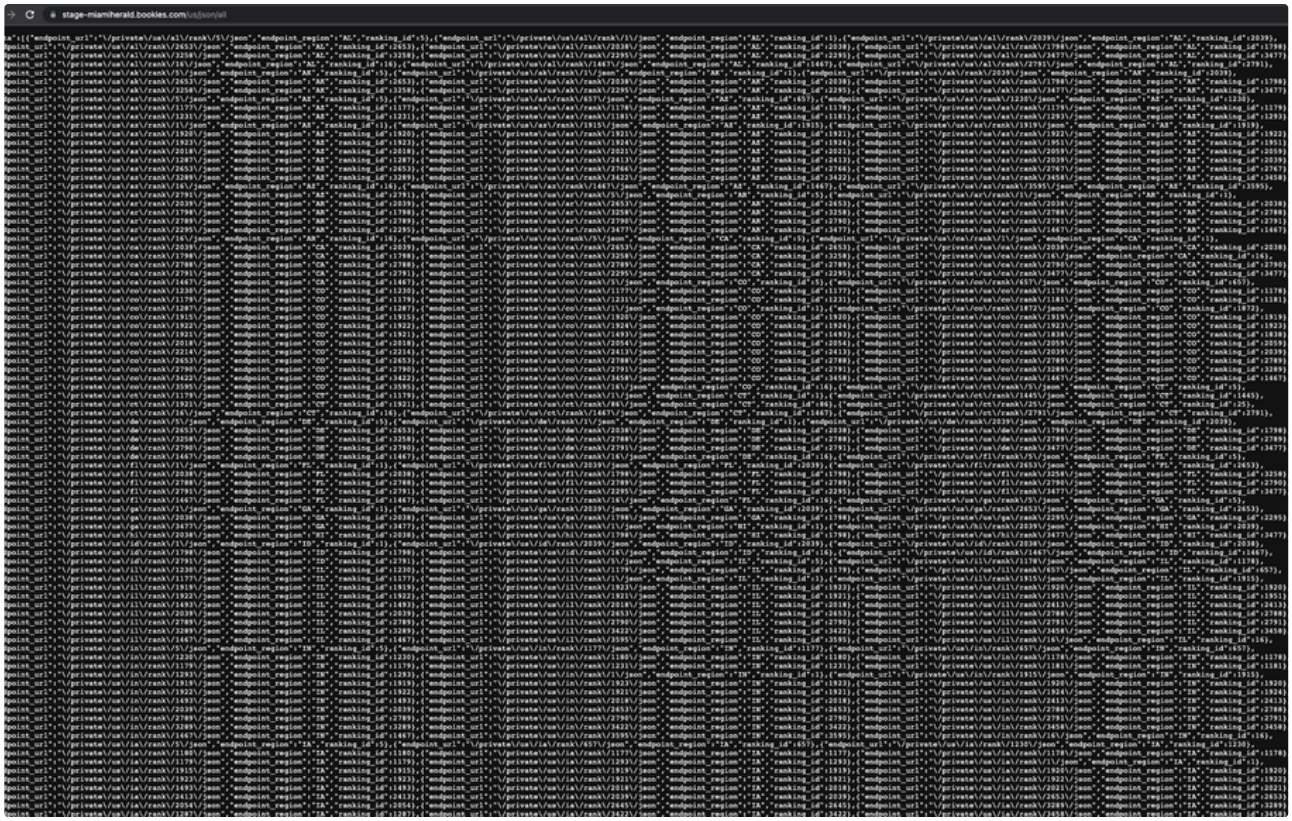
644 endpoints

	State/Region	Single Endpoint Link	Endpoint URL
1	Alabama	Link	/private/us/al/rank/5/json
2	Alabama	Link	/private/us/al/rank/2757/json
3	Alabama	Link	/private/us/al/rank/644/json
4	Alabama	Link	/private/us/al/rank/645/json
5	Alabama	Link	/private/us/al/rank/16/json
6	Alabama	Link	/private/us/al/rank/3630/json
7	Alabama	Link	/private/us/al/rank/3679/json
8	Alaska	Link	/private/us/ak/rank/5/json

On this hub page, you will also find the link to the JSON list of endpoints.

```
/{country}/json/all
```

This list has all of the endpoints in JSON format needed for the selected country.



You can also view the single JSON endpoint from the home page by clicking on the green button of the region and list that you want

and then clicking the JSON Endpoint link

Server-Side Dependency Notes

- Geo-location will have to be handled by a partner backend/edge system.
- ktag and exit page HTML implementation should follow the reference included above in the client side JS advanced includes option.
  - KTAG tracking with JSON Server Side CTA's must include the javascript file for tracking: `https://~SITE~.com/js/ktag.js`

    For tracking the partner must add:
    - CTA Button: `data-kt="cta"` on the 'a' tag
    - Oplist click that's not a CTA Button: `data-kt="oplistclk"` on the 'a' tag
- GDC AdTech stage (test) environment is protected by basic auth and any testing via Postman or any other API tester, will need to pass through credentials for testing. GDC will provide credentials to the partner implementation team.

GDC Matrix Widgets for Partners (Draft)

What: GDC provides various prepackaged website components to display sports data information such as league game, or player statistics or associated odds for sporting events which lives alongside editorial content on a page.

Why: GDC and MCC are requesting removal of MetaBet widgets on MCC sites in favor of a GDC powered set of widgets which as more cost effective and will allow for further functional expansion.

How: In order to implement GDC matrix widgets on an external partner site not managed by GDC CMS there is a quick integration option.

1. Include JS library in head of page templates where sports widgets are desired.
   a. This can be enabled for all pages in your existing app.js include
   b. or as a partner you can choose for it to be integrated on a case by case basis with the additional inclusion of a bookmakers.js which includes code to connect with the GDC sports data API to return near realtime sports data.
2. CMS Editor adds mounting point content div so that JS above on page load will make appropriate API calls and populate sports data in the appropriate div tag. Example below but not for use, only reference.
   a.
```
<div class="matrix-widget mw-outrights" mw-matrix-ids="e3d50f377935feefb29189a5bab47c5fd2c88d4d" mw-odds-for
```

MetaBet Replacement

- Existing MetaBet Example for MCC - NFL Week 4 Odds & Betting Lines: Point Spreads, Moneylines, Totals
- Example GDC matrix widget - https://widgets.kaxmatrix.com/demo/pages/team-vs-team?site=GDC-US-NJ&t=-240&locale=en&american=true&sport=74716bec6739607fc7a05a185f805a6769543dfe&tournament=49c10f47211438b77cc9b3af72663899b1f79d53&sportEvent=9f2b26378a508d2985b66cbc50300d2311a6425c

Questions

- Where is the MB widget JS being included currently within MCC systems?
  - 8 properties (bellville, center daily, charollote, KC, miami, Raleigh, sacramento, witchita)
- Can the MB JS include be disabled or removed on the MCC side and will that disable the MB widgets in the various MCC local sites without content editors needing to go in and edit each page to remove?
  - Action - Chase to test GDC widgets on non production site and will work with MCC dev team to test MB removal as a priority and what is impact. Do we get errors in articles when JS is removed or are the widget contents just not shown?
- If yes to above, how long will that take to do given EOM MB contract shutdown?
- If no, content team will work to remove and post completion would want the JS include removed as well.
- We have staging code to share tomorrow and will release final production widget code for MCC later this week. What do you need to get started and when do you think you can release if a code change is needed?

FAQ

- How fast are the GDC widgets?
  - GDC widget data is cached to a redis instance to ensure optimal performance
- Who maintains the mw-bookmakers.js file?
  - The GDC partnership dev team maintains the middleware code for partners
  - We use standard software dev
- Are analytics tied to the use of widgets?
  - Yes, the widgets integrate with the existing ktag.js code to track exit clicks performed from GDC matrix widget CTAs.
- What are the SEO implications with dynamic data?
  - We will follow up with static representation to be better indexed by Google.

Performance Considerations

The URLs with HTML fragments or JSON representing oplists are static files stored on GDC's Cloudflare CDN and are updated by GDC's publishing tool. The oplist response is fast as no additional processing time is required. If a page is requested frequently it may slow the site down and is the responsibility of the partner backend system to implement a caching system.

Proxying a GDC Microsite

Partners can reverse proxy a GDC microsite so they control the domain and sub-path where the site is reverse proxied. The partner will need to provide the client geo location (2-letter code for country and region) in the header for GDC to use to map to partner and geo-specific CTAs. The partner and GDC will need to coordinate on what header names are set by the partner for integration.

GDC stage (test) microsites are protected by authentication and any proxy will need to pass thru credentials for testing so the end user doesn't get a login prompt. GDC will provide those credentials for basic auth to the partner.