Supply Side Documentation (2.14)

General Information

- As a publisher, the integration between your system or between your site and Yieldlab is quite simple through the use of established adtags
- You receive an adtag from Yieldlab for integrating into your system (e.g.: adserver, CMS, website)
- We don’t count clicks in our system due to the fact that our RTB platform measures performance based on ad impressions. This means that the integration of a click parameter in our adtags is not necessary or supported.
- Replacing adtag parameters dynamically via passbacks is supported in some cases.
  If necessary, you can speak to your Yieldlab contact person.
- Yieldlab adtags are implemented using JavaScript (via document.write) by default
- The following pages describe how you can integrate Yieldlab adtags into your site

Adtag

Example of a standard adtag and explanation of the relevant parameters
NOTE: This adtag is for example use only

Javascript

Example Code

```
<script type="text/javascript" language="JavaScript" src="https://ad.yieldlab.net/d/${adslot_id}/${check_id}?
ts=${timestamp}&consent=${gdpr_consent_string}&schain=${supply_chain_object}&pubref=${url_encoded_referrer}"><
/script>
```

Request parameters

Parameter	Name	Required	Description
adslot ID	Placement ID	X	The unique adtag placement ID in the Yieldlab platform. This is provided by Yieldlab and should not be changed
check ID	Checknumber	X	A verification ID specific to the adtag or publisher. This is provided by Yieldlab and should not be changed
ts	Cachebuster	X	The TS must be replaced with a unique, random number (e.g.: timestamp or random number). The timestamp is necessary to avoid impression discrepancies due to browser caching
id	External ID		This parameter is optional. It can be an ID (e.g.: 123) for each unique website in the publishers network and can be used for reporting purposes. Please note: If you are using the Adition-Yieldlab Singe-Request-Integration, this parameter will be filled automatically with Adition's Content Unit, Campaign ID and Creative ID.
pubref pubappname	Publisher referrer URL/app name		This parameter allows the publisher to override the referrer URL for the placement. Use "pubref" for web adslots, and "pubappname" for in-app adslots. When providing either "pubref" or "pubappname", make sure that they are URL-encoded! Please note: • If you are using referrer groups, the parameter "pubref" is required! • "pubref" should be a valid URI (including protocol)
pubbundlename	Publisher bundle name		*Overrides the bundle name configured on adslot level. E.g. com.yieldlab.ssp*
pubstoreurl	Publisher store URL		Overrides the store url configured on adslot level. String needs to be URLEncoded.
consent	Consent string		Base64Url encoded vendor consent string as per the GDPR Transparency and Consent Framework
t	Targeting		Sample: interest=auto,sport&gender=male (encodiert: interest%3Dauto%2Csport%26gender%3Dmale%20) • Key-Values are separated by ampersand "&" • Multiple Values are separated by comma "," String needs to be URLEncoded.
schain	Supply chain		SupplyChainObject with allowed and ordered properties, which must be separated by a comma (“,”) *and each node needs to be headed by a bang ("!"): "ver,complete!asi,sid,hp,rid,name,domain,ext".* Optional parameters (*rid, name,domain,ext)* can be left empty, but commas in between stay necessary. Example: schain=1.0,1!exchange1.com,1234,1,bid-request-1,publisher,publisher.com schain is not needed when using the Yieldprobe to initiate auctions (see page 10). Read more about the SupplyChainObject here: https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/supplychainobject.md
ids	Extended IDs		An array of Extended IDs that will be sent selectively to demand technologies in the bid request. The single IDs within the *ids* parameter are separated by commas, and each id name and its value are separated by a colon. Currently supported Extended IDs: • unifiedId - 'adserver.org' • identityLink - 'liveramp.com' • criteo - 'criteo.com' • NetId - 'netid.de' • id5Id - 'id5-sync.com' • pubCommonId - 'pubcid.org' • Adition Id - 'adition.com' Example: http://ad.yieldlab.net/d/1234567/9876/100x100?ts=1589966746344&ids=netid.de:fH5A3n2O8_CZZyPoJVD-eabc6ECb7jhxCicsds7qSg,sample.com:1111-111-1111-111

gp	Google providers		true or false (alternative: 1 or 0) Append this GET parameter to force sending your Google provider whitelist in bid requests. Your Google provider whitelist can be found in YRD (Settings/Google DV360 Provider List, see https://my.yieldlab.net/settings/provider-whitelist).

iab_content	Content metadata		An object containing information about the content that will surround the placement. This information will be passed to DSPs in the OpenRTB *site.content* and *app.content* objects.
			
			For more information about the supported properties refer to the IAB content object below.
			
			Please note:
			
			• *All* string values must be URL encoded. This applies to individual string values as well as string values inside a string array.
			  ◦ e.g. `iab_content=[...],title:Game%20of%20Thrones,[...]`
			• String array values must be concatenated with a pipe symbol.
			  ◦ **please note:** this only applies to *cat* and *keywords* fields
			  ◦ e.g. `iab_content=[...],producer.cat:IAB1-7|IAB26-1,[...]`
			• Object array values must be denoted by their index
			  ◦ **please note:** array values inside that structure have to be denoted by their indices as well
			  ◦ e.g. `iab_content=[...],data.segment.0.name:one,data.segment.0.value:two,data.segment.0.ext.arbit.prop:erty-value,data.segment.0.ext.strarr.0:value1,data.segment.0.ext.strarr.1:value2,[...]`
			• Property names and their values must be separated by a colon
			  ◦ e.g. `iab_content=id:idvalue`
			• Properties themselves must be concatenated with a comma
			  ◦ e.g. `iab_content=id:idvalue,episode:1`

dsarequired	Digital Services Act compliance required		Flag to indicate if DSA information should be made available. This will signal if the bid request belongs to an Online Platform/VLOP, such that a buyer should respond with DSA Transparency information based on the dsapubrender value. • 0 = Not required • 1 = Supported, bid responses with or without DSA object will be accepted. For the time being this value will be treated as 0 = not required. • 2 = Required, bid responses without DSA object will not be accepted • 3 = Required, bid responses without DSA object will not be accepted, Publisher is an Online Platform
dsapubrender	Publisher will render Digital Services Act information		Indicates whether the publisher intends to render DSA information in the ad • 0 = The publisher cannot render transparency info • 1 = Publisher could render depending on adrender. If DSA is required (dsarequired = 2 or 3) and no preceding Yieldprobe recommendation exists, this value is not supported. • 2 = Publisher will render. If DSA is required (dsarequired = 2 or 3) and no preceding Yieldprobe recommendation exists, this value is not supported.
dsadatatopub	Publisher requires transparency information		Indicates whether the publisher needs transparency information from the advertiser regardless of dsapubrender • 0 = The publisher does not require transparency information • 1 = Transparency information from the advertiser is optional • 2 = The advertiser must send transparency information. If DSA is required (dsarequired = 2 or 3) and no preceding Yieldprobe recommendation exists, this value is not supported.
dsadomain	Domain name of the entity that applied user parameters		The domain name of the entity that applied user parameters.
dsaparams	Digital Services Act transparency parameters		A comma separated list of numbers indicating the type of profiling used.
dsatransparency	Digital Services Act transparency parameters		String formatted Digital Services Act transparency parameters. Composed of the two items from the transparency object; the domain string and the params array. These two items are separated by a tilde “~”. Values in the params array are separated by an underscore “_”. Multiple transparency objects are separated by two tildes “~~”. Example: &dsatransparency=platform1domain.com~1~~SSP2domain.com~1_2 If this value is present it takes precedence over dsadomain and dsaparams.
segtax	Google Topics taxonomy		A string identifying the Google Topics taxonomy
segclass	Google Topics machine learning classifier		A string identifying the Google topics machine learning classifier
segments	Google Topics		An array of integers defining the Google Topics

IAB content object

<table>
<tr><th>Property</th><th>Type</th><th>Description</th></tr>
<tr><td>id</td><td>string</td><td>ID uniquely identifying the content.</td></tr>
<tr><td>episode</td><td>integer</td><td>Episode number.</td></tr>
<tr><td>title</td><td>string</td><td>Content title.</td></tr>
<tr><td>series</td><td>string</td><td>Content series.</td></tr>
<tr><td>season</td><td>string</td><td>Con tent season.</td></tr>
<tr><td>artist</td><td>string</td><td>Artist credited with the content.</td></tr>
<tr><td>genre</td><td>string</td><td>Genre that best describes the content.</td></tr>
<tr><td>album</td><td>string</td><td>Album to which the content belongs; typically for audio.</td></tr>
<tr><td>isrc</td><td>string</td><td>International Standard Recording Code conforming to ISO - 3901.</td></tr>
<tr><td>producer</td><td>object</td><td>Details about the content Producer. Refer to the IAB content producer object below.</td></tr>
<tr><td>url</td><td>string</td><td>URL of the content, for buy - side contextualization or review.</td></tr>
<tr><td>cattax</td><td>integer;
default 1</td><td>The taxonomy in use. Refer to the listing IAB category taxonomies for value below.</td></tr>
<tr><td>cat</td><td>string array</td><td>Array of IAB content categories that describe the content.

The taxonomy to be used is defined by the cattax field. If no cattax field is supplied IAB Content Category Taxonomy 1.0 is assumed.</td></tr>
<tr><td>prodq</td><td>integer</td><td>Production quality.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Unknown</td></tr><tr><td>1</td><td>Professionally Produced</td></tr><tr><td>2</td><td>Prosumer</td></tr><tr><td>3</td><td>User Generated (UGC)</td></tr></table></td></tr>
<tr><td>context</td><td>integer</td><td>Type of content (game, video, text, etc.)
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Video (i.e., video file or stream such as Internet TV broadcasts)</td></tr><tr><td>2</td><td>Game (i.e., an interactive software game)</td></tr><tr><td>3</td><td>Music (i.e., audio file or stream such as Internet radio broadcasts)</td></tr><tr><td>4</td><td>Application (i.e., an interactive software application)</td></tr><tr><td>5</td><td>Text (i.e., primarily textual document such as a web page, eBook, or news article)</td></tr><tr><td>6</td><td>Other (i.e., none of the other categories applies)</td></tr><tr><td>7</td><td>Unknown</td></tr></table></td></tr>
<tr><td>contentrating</td><td>string</td><td>Content rating (e.g., MPAA).</td></tr>
<tr><td>userrating</td><td>string</td><td>User rating of the content (e.g., number of stars, likes, etc.).</td></tr>
<tr><td>qagmediarating</td><td>integer</td><td>Media rating per IQG guidelines.
1: All Audience;
2: Everyone Over Age 12;
3: Mature Audience</td></tr>
<tr><td>keywords</td><td>string array</td><td>List of keywords describing the content</td></tr>
<tr><td>live</td><td>integer</td><td>0 = not live, 1 = content is live</td></tr>
</table>

sourcelrelationship	integer	0 = indirect, 1 = direct
len	integer	Length of content in seconds; appropriate for video or audio.
language	string	Content language using ISO-639-1-alpha-2. Only one of language or langb should be present.
embeddable	integer	Indicator of whether the content is embeddable (e.g., an embeddable video player), where 0 = no, 1 = yes
data	object array	Additional content data. Each data object represents a different data source.
network	object	Details about the network the content is on. Refer to the IAB content network object below.
channel	object	Details about the channel the content is on. Refer to the IAB content channel object below.

IAB content producer object

Property	Type	Description
producer.id	string	Content producer or originator ID. Useful if content is syndicated and may be posted on a site using embed tags.
producer.name	string	Content producer or originator name.
producer.cattax	integer: default 1	The taxonomy in use. Refer to the list of IAB category taxonomies for values below.
producer.cat	string array	Array of IAB content categories that describe the content producer. The taxonomy to be used is defined by the cattax field. If no cattax field is supplied IAB Content Category Taxonomy 1.0 is assumed.
producer.domain	string	Highest level domain of the content producer.

IAB content data object

Property	Type	Description
data.id	string	Exchange-specific ID for the data provider.
data.name	string	Exchange-specific name for the data provider.
data.segment	object array	Array of Segment objects that contain the actual data values.

IAB content data segment object

Property	Type	Description
data.segment.id	string	ID of the data segment specific to the data provider.
data.segment.name	string	Name of the data segment specific to the data provider.
data.segment.value	string	String representation of the data segment value.
data.segment.ext	object	Placeholder for exchange-specific extensions to OpenRTB.

IAB content network object

Property	Type	Description
network.id	string	A unique identifier assigned by the publisher. This may not be a unique identifier across all supply sources.
network.name	string	Network the content is on.
network.domain	string	The primary domain of the network.

IAB content channel object

Property	Type	Description

channel.id	string	A unique identifier assigned by the publisher. This may not be a unique identifier across all supply sources.
channel.name	string	Channel the content is on.
channel.domain	string	The primary domain of the channel.

IAB category taxonomies

Value	Definition
1	IAB Tech Lab Content Category Taxonomy 1.0. - Deprecated, and recommend NOT be used since it does not have SCD flags.
2	IAB Tech Lab Content Category Taxonomy 2.0: Deprecated, and recommend NOT be used since it does not have SCD flags.
3	IAB Tech Lab Ad Product Taxonomy 1.0.
4	IAB Tech Lab Audience Taxonomy 1.1
5	IAB Tech Lab Content Taxonomy 2.1
6	IAB Tech Lab Content Taxonomy 2.2
500+	Vendor-specific codes.

Mobile specific parameters

Parameter	Name	Required	Description
lat	Latitude		e.g. 51.56639159936839
lon	Longitude		e.g. 9.926878832422114
yl_rtb_ifa	Mobile identifier		*Unique Identifier for mobile Apps, e.g.: Apple IFA, Android Advertising ID*
yl_rtb_devicetype	Device type		3 = Connected TV, 4 = Phone, 5 = Tablet
yl_rtb_connection type	Connection type		0 = Unknown, 2 = Wif,i 3 = Cellular Network - Unknown Generation, 4 = Cellular Network - 2G, 5 = Cellular Network - 3G, 6 = Cellular Network - 4G

Video specific parameters

Parameter	Name	Required	Description
min_d	Minimum duration		Allows to set a minimum duration in seconds for the requested adslot in the format: min_d= <minimum_duration> Example: min_d=15 sets the minimum duration for the respective adslot to 15 second
max_d	Maximum duration		Allows to set a maximum duration in seconds for the requested adslot in the format: max_d= <maximum_duration> Example: max_d=120 sets the maximum duration for the respective adslot to 120 second
skip	Video skippable		Allows to override if the video (except DOOH) is skippable, 1 skippable, 0 non-skippable Example: skip=1 sets the adslot to skippable. Invalid values for skip will be ignored! Example: skip=123 does not override anything; the configured value for the adslot from YRD will be used

Iframe

It is also possible to integrate the Yieldlab adtag as an IFrame. The following steps describe how to convert to use an iframe adtag.

Example Code

```
<iframe src="https://ad.yieldlab.net/d/0000/1111?type=h&pubref=[PublisherReferer]" width="729" height="90"
marginwidth="0" marginheight="0" hspace="0" vspace="0" frameborder="0" scrolling="no"></iframe>
```

Example of an iframe adtag where the encoded referrer URL is provided as an additional parameter:

```
<iframe src="https://ad.yieldlab.net/d/0000/1111?type=h&pubref=http%3A%2F%2Fwww.site.com%2Fweather" width="729"
height="90" marginwidth="0" marginheight="0" hspace="0" vspace="0" frameborder="0" scrolling="no"></iframe>
```

Important Notes

- Referrer: If the Yieldlab adtag is implemented as an IFrame, the Yieldlab platform is unable to read the referrer of the publishers page. This information is vital for Demand Partners and is required for optimal campaign performance. In order to pass the referrer information to the Yieldlab platform, you will have to pass an additional parameter (pubref=[PublisherReferer]) into the iFrame. This parameter must be filled from your adserver and encoded appropriately.

  Note:
  If you switch from JavaScript to iFrame you have to make sure that your adserver or another script is able to fill the parameter [PublisherReferer] with the relevant information.
- Special Characters: The referrer URL may only contain ASCII (no special characters like accents are supported).
- Encoded: The referrer URL which is passed to our adtag must be encoded correctly (e.g. using JavaScript's encodeURI function)
- Please contact support@yieldlab.de for further questions.

Native

Native ad tags differ from standard Javascript/Iframe ad tags. A Native ad tag only describes the structure of the ad in JSON format. The styling and display of the ad is being handled by the publisher. The Native format is pre-defined by Yieldlab and contains a picture with a definable size, a title and an optional teaser text.

JSON Output

Example of a Native ad tag response:

```
{
   "native":{
      "link":{
         "url":"https://www.yieldlab.de"
      },
      "assets":[
         {
            "id":1,
            "title":{
               "text":"This is a great headline"
            }
         },
         {
            "id":2,
            "img":{
               "url":"https://www.yieldlab.net/static/yl-logo400x200.jpg",
               "w":400,
               "h":200
            }
         },
         {
            "id":3,
            "data":{
               "value":"You should really buy our totally amazing product right now because it's super awesome"
            }
         }
      ],
      "imptrackers":[
         "https://counts.yieldlab.net/ve?
d=ODE9ZSY3MTUyMDIzMTI0MjUxPXN0JjE4ZWFiYzg2YjM3MC1jMTJiLWE1ZDQtNDNhZS0wOTMyMGU3Nz1kaXQmMj12",
         "https://counts.yieldlab.net/imp?s=21691&d=5094511&a=5101251&ts=1524213202517&tid=77e02390-ea34-4d5a-
b21c-073b68cbae81&l=0"
      ],
      "ver":"1.1"
   }
}
```

Assets

The Native format consists of three assets:

ID	Name	Description
1	Title	Title of the Native ad
2	Image	Image of the Native ad. The image's size is determined by the respective adslot's setting (Format).
3	Teaser	Teaser text of the Native ad. Optional adslot setting.

Links & Trackers

Key	Name	Description
link.url	Landing page	The landing page to which the native ad links.

imptrackers	Impression pixel	Impression pixels to be fired upon successful display of the ad

Integration

Yieldlab recommends to call the Native ad tag via XHR request from the client. Please ensure not only the proper displaying of the rendered ad but also the firing of the impression pixels as well as the external linking to the landing page when a user clicks the ad.

Example

The following example demonstrates how the response of a Native ad tag could be processed:

```
// call adtag
var nativeXHR = new XMLHttpRequest();
nativeXHR.addEventListener("load", reqListener);
nativeXHR.withCredentials = true;
nativeXHR.open("GET", "https://ad.yieldlab.net/d/123456/7890/400x200?ts=152224202279"); // sample native adslot
2:1
nativeXHR.send();

function reqListener () {
  response = JSON.parse(this.responseText)

  // render ad
  document.querySelector("#ad .title").innerHTML = response.native.assets[0].title.text
  document.querySelector("#ad .img").src = response.native.assets[1].img.url
  document.querySelector("#ad .text").innerHTML = response.native.assets[2].data.value

  // add click to landing page
  document.querySelector("#ad").addEventListener('click', function() {
          window.open(response.native.link.url);
  });

  // call imp trackers
  for (i=0;response.native.imptrackers.length>i;i++) {
          var imp_pixel = new Image();
        imp_pixel.src = response.native.imptrackers[i];
  }
}
```

You can view a live example on our demo page at:

```
http://test.yieldlab.de/native.html
```

Yieldprobe

Using Yieldprobe, publishers can obtain important insight into any available campaigns running on the Yieldlab platform. This information is made accessible to publisher websites via a JavaScript object. Yieldlab campaign information can thus be forwarded to a publisher's adserver system e.g. via a request key/value mechanism. By signaling the availability and conditions of a Yieldlab campaign, publishers can greatly improve their fill rates. At the same time, they can continue to use their existing adserver systems to honor any preexisting campaign prioritization rules.

Furthermore, additional JavaScript key/values can be set to meet publisher's custom requirements. For example, it is possible to define custom priorities based on price ranges or customers. Please contact Yieldlab for more details if you have advanced needs.

Integration

The Yieldprobe tag can be used to retrieve matching campaign information for one or more ad placements or *adslots* running on the Yieldlab system. In the case where multiple adslots are present on the page, these should be combined in a **single Yieldprobe request** for performance reasons. Upon completion of the Yieldprobe request, a Javascript object will be returned with campaign-related information for each adslot included in the original request. The Yieldprobe tag must be positioned at the **top of every page**, prior to any ad tags. Since all Requests to Yieldlab are routed through Akamai, we can guarantee 100% availability, even in the unlikely case of Yieldlab's Datacenter being unresponsive.

A generic Yieldprobe tag is provided below, where *ts* is a cache-busting parameter.

```
<script type="text/javascript" src="https://ad.yieldlab.net/yp/${adslots}?ts=${timestamp}
&consent=${gdpr_consent_string}&schain=${supply_chain_object}&pubref=${url_encoded_referrer}"></script>
```

Request Parameters

Parameter	Name	Required	Description
adslots	Placement IDs	X	Comma-separated list of adslot IDs (e.g. 3418,3419). You can call a maximum of **10** adslots.
ts	Cachebuster	X	The cachebuster must be replaced with a unique, random number (e.g.: timestamp or random number). The timestamp is necessary to avoid impression discrepancies due to browser caching
content	Reponse type		json - JSON Output. See example in "Yieldlab Response" below amp - JSON Output. Like "json" but compatible with AMP RTC (https://github.com/ampproject/amphtml/blob/master/extensions/amp-a4a/rtc-documentation.md)
redirect	Adserver redirect		You can append a url to this parameter if you want to pass the YP recommendation directly as key-values in a 302 HTTP redirect. e.g. https://ad.yieldlab.net/yp/24062?ts=903991786996&redirect=http://adserver.com?keywords=[ypkws] (More information in section "Redirect")
pvid	PageView ID		true or false (alternative: 1 or 0) Append this GET parameter to force a page view behavior regardless of user-agent
pubref pubappname	Publisher referrer URL/app name		This parameter allows the publisher to override the referrer URL for all the adslots listed in the Yieldprobe request. Use "pubref" for web adslots, and "pubappname" for in-app adslots. When providing either "pubref" or "pubappname", make sure that they are URL-encoded! Please note: • If you are using referrer groups, the parameter "pubref" is required! • "pubref" should be a valid URI (including protocol)
pubbundlename	Publisher bundle name		Overrides the bundle name configured on adslot level. E.g. com.yieldlab.ssp
pubstoreurl	Publisher store URL		Overrides the store url configured on adslot level. String needs to be URLEncoded.
consent	Consent string		Base64Url Encoded Vendor Consent String as per the GDPR Transparency and Consent Framework
floor			This parameter allows the publisher to define a dynamic floor for specific adslots. If there is also a static floor defined, the highest of both floor prices will be considered. Sample: 1234:200,234:100 Means: for adslot 1234 the requested floor is 200 cents while for adslot 234 the requested floor is 100 cent.
sizes	Requested sizes		Sample: 1234:100x100|200x200,234:400x300|400x400 Means: for adslot 1234 the requested sizes are 100x100 and 200x200 while for adslot 234 the requested sizes are 400x300 and 400x400 Cases: 1. if no sizes are specified all configured sizes on the adslot are used 2. if some sizes are specified only those sizes take part in the auction 3. if some sizes are specified that don't match any on the adslot configured sizes they are ignored 4. if all sizes specified don't match any on the adslot no auction takes place for this adslot
t	Targeting		Sample: interest=auto,sport&gender=male (encodiert: interest%3Dauto%2Csport%26gender%3Dmale%20) • Key-Values are separated by ampersand "&" • Multiple Values are separated by comma "," String needs to be URLEncoded.

schain	Supply chain		SupplyChainObject with allowed and ordered properties, which must be separated by a comma (“,”) *and each node needs to be headed by a bang ("!"): "ver,complete!asi,sid,hp,rid,name,domain,ext".* Optional parameters (*rid, name,domain,ext)* can be left empty, but commas in between stay necessary. Example: schain=1.0,1!exchange1.com,1234,1,bid-request-1,publisher,publisher.com Read more about the SupplyChainObject here: https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/supplychainobject.md
ids	Extended IDs		An array of Extended IDs that will be sent selectively to demand technologies in the bid request. The single IDs within the *ids* parameter are separated by commas, and each id name and its value are separated by a colon. Currently supported Extended IDs: • unifiedId - 'adserver.org' • identityLink - 'liveramp.com' • criteo - 'criteo.com' • NetId - 'netid.de' • id5Id - 'id5-sync.com' • pubCommonId - 'pubcid.org' • Adition Id - 'adition.com' Example: https://ad.yieldlab.net/yp/1234567,987654?ts=1589966746344&ids=netid.de:fH5A3n2O8_CZZyPoJVD-eabc6ECb7jhxCicsds7qSg,sample.com:1111-111-1111-111
atypes	Extended IDs' Atypes		An array of Extended Atypes that will be sent selectively to demand technologies in the bid request. *atypes* parameters are separated by commas, each id name and its atype value are separated by a colon. **Please note:** • single atype value is in format of integer, invalid atype value will be ignored • atype sent without a corresponding extended id will be ignored, in the following example, "ppid.som=3" will be ignored ◦ e.g. `ids=netid.de:sample_netid,id5-sync.com:sample_id5` and `atypes=netid.de:1,id5-sync.com:2,ppid.som=3` Example: https://ad.yieldlab.net/yp/1234567,987654?ts=1589966746344&ids=netid.de:fH5A3n2O8_CZZyPoJVD-eabc6ECb7jhxCicsds7qSg,sample.com:1111-111-1111-111&atypes=netid.de:111,sample.com:222 More info about openRTB *Extended Identifiers* Object can be found here: https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/extensions/2.x_official_extensions/eids.md
gp	Google providers		true or false (alternative: 1 or 0) Append this GET parameter to force sending your Google provider whitelist in bid requests. Your Google provider whitelist can be found in YRD (Settings/Google DV360 Provider List, see https://my.yieldlab.net/settings/provider-whitelist).

iab_content	Content metadata		An object containing information about the content that will surround the placement. This information will be passed to DSPs in the OpenRTB *site.content* and *app.content* objects.
			
			For more information about the supported properties refer to the IAB content object below.
			
			Please note:
			
			• *All* string values must be URL encoded. This applies to individual string values as well as string values inside a string array.
			  ◦ e.g. `iab_content=[...],title:Game%20of%20Thrones,[...]`
			• String array values must be concatenated with a pipe symbol.
			  ◦ **please note:** this only applies to the *cat* field
			  ◦ e.g. `iab_content=[...],producer.cat:IAB1-7|IAB26-1,[...]`
			• Object array values must be denoted by their index
			  ◦ **please note:** array values inside that structure have to be denoted by their indices as well
			  ◦ e.g. `iab_content=[...],data.segment.0.name:one,data.segment.0.value:two,data.segment.0.ext.arbit.prop:erty-value,data.segment.0.ext.strarr.0:value1,data.segment.0.ext.strarr.1:value2,[...]`
			• Property names and their values must be separated by a colon
			  ◦ e.g. `iab_content=id:idvalue`
			• Properties themselves must be concatenated with a comma
			  ◦ e.g. `iab_content=id:idvalue,episode:1`

dsarequired	Digital Services Act compliance required		Flag to indicate if DSA information should be made available. This will signal if the bid request belongs to an Online Platform/VLOP, such that a buyer should respond with DSA Transparency information based on the dsapubrender value. • 0 = Not required • 1 = Supported, bid responses with or without DSA object will be accepted. For the time being this value will be treated as 0 = not required. • 2 = Required, bid responses without DSA object will not be accepted • 3 = Required, bid responses without DSA object will not be accepted, Publisher is an Online Platform

dsapubrender	Publisher will render Digital Services Act information		Indicates whether the publisher intends to render DSA information in the ad • 0 = The publisher cannot render transparency info • 1 = Publisher could render depending on adrender • 2 = Publisher will render
dsadatatopub	Publisher requires transparency information		Indicates whether the publisher needs transparency information from the advertiser regardless of dsapubrender • 0 = The publisher does not require transparency information • 1 = Transparency information from the advertiser is optional • 2 = The advertiser must send transparency information.
dsadomain	Domain name of the entity that applied user parameters		The domain name of the entity that applied user parameters.
dsaparams	Digital Services Act transparency parameters		A comma separated list of numbers indicating the type of profiling used.
dsatransparency	Digital Services Act transparency parameters		String formatted Digital Services Act transparency parameters. Composed of the two items from the transparency object; the domain string and the params array. These two items are separated by a tilde “~”. Values in the params array are separated by an underscore “_”. Multiple transparency objects are separated by two tildes “~~”. Example: &dsatransparency=platform1domain.com~1~~SSP2domain.com~1_2 If this value is present it takes precedence over dsadomain and dsaparams.
segtax	Google Topics taxonomy		A string identifying the Google Topics taxonomy
segclass	Google Topics machine learning classifier		A string identifying the Google topics machine learning classifier
segments	Google Topics		An array of integers defining the Google Topics

IAB content object

Property	Type	Description
id	string	ID uniquely identifying the content.
episode	integer	Episode number.
title	string	Content title.
series	string	Content series.
season	string	Con tent season.
artist	string	Artist credited with the content.
genre	string	Genre that best describes the content.
album	string	Album to which the content belongs; typically for audio.
isrc	string	International Standard Recording Code conforming to ISO - 3901.
producer	object	Details about the content Producer. Refer to the IAB content producer object below.
url	string	URL of the content, for buy - side contextualization or review.
cattax	integer; default 1	The taxonomy in use. Refer to the listing IAB category taxonomies for value below.
cat	string array	Array of IAB content categories that describe the content. The taxonomy to be used is defined by the cattax field. If no cattax field is supplied IAB Content Category Taxonomy 1.0 is assumed.

<table>
<tr><td>prodq</td><td>integer</td><td>Production quality.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Unknown</td></tr><tr><td>1</td><td>Professionally Produced</td></tr><tr><td>2</td><td>Prosumer</td></tr><tr><td>3</td><td>User Generated (UGC)</td></tr></table></td></tr>
<tr><td>context</td><td>integer</td><td>Type of content (game, video, text, etc.)<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Video (i.e., video file or stream such as Internet TV broadcasts)</td></tr><tr><td>2</td><td>Game (i.e., an interactive software game)</td></tr><tr><td>3</td><td>Music (i.e., audio file or stream such as Internet radio broadcasts)</td></tr><tr><td>4</td><td>Application (i.e., an interactive software application)</td></tr><tr><td>5</td><td>Text (i.e., primarily textual document such as a web page, eBook, or news article)</td></tr><tr><td>6</td><td>Other (i.e., none of the other categories applies)</td></tr><tr><td>7</td><td>Unknown</td></tr></table></td></tr>
<tr><td>contentrating</td><td>string</td><td>Content rating (e.g., MPAA).</td></tr>
<tr><td>userrating</td><td>string</td><td>User rating of the content (e.g., number of stars, likes, etc.).</td></tr>
<tr><td>qagmediarating</td><td>integer</td><td>Media rating per IQG guidelines.
1: All Audience;
2: Everyone Over Age 12;
3: Mature Audience</td></tr>
<tr><td>keywords</td><td>string</td><td>List of keywords describing the content</td></tr>
<tr><td>live</td><td>integer</td><td>0 = not live, 1 = content is live</td></tr>
<tr><td>sourcelrelationship</td><td>integer</td><td>0 = indirect,
1 = direct</td></tr>
<tr><td>len</td><td>integer</td><td>Length of content in seconds; appropriate for video or audio.</td></tr>
<tr><td>language</td><td>string</td><td>Content language using ISO-639-1-alpha-2. Only one of language or langb should be present.</td></tr>
<tr><td>embeddable</td><td>integer</td><td>Indicator of whether the content is embeddable (e.g., an embeddable video player), where 0 = no, 1 = yes</td></tr>
<tr><td>data</td><td>object array</td><td>Additional content data. Each data object represents a different data source.</td></tr>
<tr><td>network</td><td>object</td><td>Details about the network the content is on. Refer to the IAB content network object below.</td></tr>
<tr><td>channel</td><td>object</td><td>Details about the channel the content is on. Refer to the IAB content channel object below.</td></tr>
</table>

IAB content producer object

Property	Type	Description
producer.id	string	Content producer or originator ID. Useful if content is syndicated and may be posted on a site using embed tags.
producer.name	string	Content producer or originator name.
producer.cattax	integer: default 1	The taxonomy in use. Refer to the list of IAB category taxonomies for values below.
producer.cat	string array	Array of IAB content categories that describe the content producer. The taxonomy to be used is defined by the cattax field. If no cattax field is supplied IAB Content Category Taxonomy 1.0 is assumed.

producer. domain	string	Highest level domain of the content producer.

IAB content data object

Property	Type	Description
data.id	string	Exchange-specific ID for the data provider.
data.name	string	Exchange-specific name for the data provider.
data.segment	object array	Array of Segment objects that contain the actual data values.

IAB content data segment object

Property	Type	Description
data.segment.id	string	ID of the data segment specific to the data provider.
data.segment.name	string	Name of the data segment specific to the data provider.
data.segment.value	string	String representation of the data segment value.
data.segment.ext	object	Placeholder for exchange-specific extensions to OpenRTB.

IAB content network object

Property	Type	Description
network.id	string	A unique identifier assigned by the publisher. This may not be a unique identifier across all supply sources.
network.name	string	Network the content is on.
network.domain	string	The primary domain of the network.

IAB content channel object

Property	Type	Description
channel.id	string	A unique identifier assigned by the publisher. This may not be a unique identifier across all supply sources.
channel.name	string	Channel the content is on.
channel.domain	string	The primary domain of the channel.

IAB category taxonomies

Value	Definition
1	IAB Tech Lab Content Category Taxonomy 1.0. - Deprecated, and recommend NOT be used since it does not have SCD flags.
2	IAB Tech Lab Content Category Taxonomy 2.0: Deprecated, and recommend NOT be used since it does not have SCD flags.
3	IAB Tech Lab Ad Product Taxonomy 1.0.
4	IAB Tech Lab Audience Taxonomy 1.1
5	IAB Tech Lab Content Taxonomy 2.1
6	IAB Tech Lab Content Taxonomy 2.2
7	IAB Tech Lab Content Taxonomy 3.0
500+	Vendor-specific codes.

Mobile specific parameters

Parameter	Name	Required	Description
lat	Latitude		Z.B. 51.56639159936839
lon	Longitude		Z.B. 9.926878832422114
yl_rtb_ifa	Mobile identifier		Unique identifier for mobile apps, e.g. Apple IFA, Android Advertising ID
yl_rtb_devicetype	Device type		3 = Connected TV, 4 = Phone, 5 = Tablet
yl_rtb_connection type	Connection type		0 = Unknown, 2 = Wifi, 3 = Cellular network - Unknown generation, 4 = Cellular network - 2G, 5 = Cellular network - 3G, 6 = Cellular network - 4G

Video specific parameters

Parameter	Name	Required	Description
min_d	Minimum duration		Allows to set a minimum duration in seconds for the request for specific adslots: min_d=<website_id>:<minimum_duration> Example: min_d=12345:15,67890:20 sets the minimum duration for adslot 12345 to 15 seconds and for adslot 67890 to 20 seconds.
max_d	Maximum duration		Allows to set a maximum duration in seconds for the request for specific adslots: max_d=<website_id>:<maximum_duration> Example: max_d=12345:45,67890:120 sets the maximum duration for adslot 12345 to 45 seconds and for adslot 67890 to 120 seconds.
startdelay	Start delay		This parameter allows the publisher to override the start delay for specific adslots.
mimes	Mime types		This parameter allows the publisher to override mimes for specific adslots.
protocols	Protocols		This parameter allows the publisher to override protocols for specific adslots.
api	API		This parameter allows the publisher to override api for specific adslots.
skip	Video skippable		This parameter allows the publisher to override if the video is skippable for specific video adslots (except DOOH), 1 skippable, 0 non-skippable: skip=<website_id>:<skippable> Example: skip=12345:0,67890:1 sets adslot 12345 to non-skippable and adslot 67890 to skippable Invalid values for skip will be ignored! Example: skip=12345:123 does not override anything; the configured value for adslot 12345 from YRD will be used

The following example demonstrates the use of the Yieldprobe tag for adslots 3418 and 3419, along with the cache-busting parameter ts (for example, the current timestamp).

```
<script type="text/javascript" src="https://ad.yieldlab.net/yp/3418,3419?ts=270712590&consent=..."></script>
```

Yieldprobe Response

Standard

```
var yl=yl||{};yl.YpResult=yl.YpResult||function(){var a={};return{add:function(b){a[b.id]=b},get:function(b)
{return a[b]},getAll:function(){return a}}}(); yl.YpResult.add({'id':3418,'advertiser':'werbung.
de','curl':'https://www.werbung.de','adsize':'160x600','partnershipType':'Private Auction'});
yl.YpResult.add({'id':3419,'advertiser':'werbung.de','curl':'https://www.werbung.
de','adsize':'300x600','adtype':'BANNER','partnershipType':'Deal'});
```

JSON


```
[{"id":3418,"advertiser":"werbung.de","curl":"https://www.werbung.de/","adsize":"160x600","adtype":"BANNER","
partnershipType":"Deal"}, {"id":3419,"advertiser":"werbung.de","curl":"https://www.werbung.de","adsize:"
300x600","adtype":"BANNER","partnershipType":"Deal"}]
```


Invoking the standard Yieldprobe tag returns a Javascript object encapsulating matching campaign information:

Javascript Object	Value
yl.YpResult	Campaign information for the specified adslots

The Javascript global object yl.YpResult can be queried using the following functions:

- get(adslot_id): returns campaign object for the specified adslot, or null if no campaign was found.
- getAll(): returns all campaign objects for which a matching campaign was found.

The campaign object provides the following attributes:

Attribute	Required	Description
id	X	Numeric value for the adslot ID (e.g. 3418)
price		Numeric value for the CPM price in cents (e.g. 125)
advertiser		String containing the advertiser name (e.g. "yieldlab")
curl		String containing the campaign URL (e.g. "www.yieldlab.de")
adsize		String containing the ad size (e.g. "160x600")
adtype		String containing the ad type ("BANNER", "VIDEO", "AUDIO" or "NATIVE")
partnershipType		String containing the partnership type ("Private Auction", "Direct Deal", "Programmatic Guaranteed" or "Preferred Deal")
format		*Deprecated*
native	only for adtype "NATIVE"	Object containing link, assets and imptrackers
dsa		Object containing Digital Services Act response parameters. See DSA

DSA

Attribute	Required	Description
behalf		Advertiser Transparency: Free UNICODE text string with a name of whose behalf the ad is displayed. Maximum 100 characters.
paid		Advertiser Transparency: Free UNICODE text string of who paid for the ad. Must always be included even if it's the same as what is listed in the behalf attribute. Maximum 100 characters
adrender		Flag to indicate that buyer/advertiser will render their own DSA transparency information inside the creative. 0 = buyer/advertiser will not render 1 = buyer/advertiser will render
transparency		Array of objects of the entities that applied user parameters and the parameters they applied. See Transparency

Transparency

Attribute	Required	Description
domain		Domain of the entity that applied user parameters
dsaparams		Array of buy-side applied user parameter targeting (using the list provided by DSA Transparency Taskforce). Include support for multiple vendors who may add their own user-targeting parameters.

Consuming the Yieldprobe Response

The snippet below shows how to invoke the Yieldprobe tag to extract campaign information to present to the user's browser:

```
<script type="text/javascript" src="https://ad.yieldlab.net/yp/3418,3419?ts=2707125907"></script>

<!-- publisher-specific code to integrate with publisher adserver -->
<script type="text/javascript">
        // iterate through all matching campaigns
        for (var id in yl.YpResult.getAll()) {
                c = yl.YpResult.get(id);
                alert("Found campaign for adslot " + c.id);
                alert("Price: " + c.price); alert("Advertiser: " + c.advertiser);
                alert("URL: " + c.curl);
        }
</script>
```

Note: If you are using the JSON Output you would just use the information directly, e.g. after an XHR Request.

Passing the Yieldprobe information to the adserver

If you want to deliver certain campaigns, that were recommended by the Yieldprobe, you need to deliver the corresponding Yieldlab adtag. You should pass the Yieldprobe information to your adserver, so you know when it makes sense to deliver a Yieldlab campaign over a direct campaign. Most of the time this is done by passing keywords/key-value-pairs, e.g.:

https://youradserver.com/1234/9876/?kw=yl3418=1,yl3418price=300,yl3418size=160x600

With this ad call your adserver would know that we have a campaign for adslot *3418 and* the price of *3€*. A Yieldlab campaign could be targeted on *yl3418=*1 and *yl3418price > 200* in this case, and would deliver:

https://ad.yieldlab.net/d/3418/8877?ts=[timestamp]

Redirect

If you are using the parameter "redirect", the Yieldprobe will automatically redirect to the appended URL. The macro [ypkws] in the URL will be replaced by key-values based on the recommendation.

Requirements:

- Maximum of one appended adslot.
- Has to be a valid URL
- URL need to include "[ypkws]"

The following example shows the sequence of the call:

1. Call to the YP with an URL (including the macro [ypkws] - you can put it wherever you want) appended to the parameter "redirect"

```
https://ad.yieldlab.net/yp/24062?ts=903991786996&redirect=http://adserver.com?keywords=[ypkws]
```

2. 302 redirect to the URL with the [ypkws] macro replaced by key-values

```
https://adserver.com?keywords=id%3D12345%26price%3D300%26advertiser%3Dadvertiser.com%26curl%3Dcampaign.
com%26format%3D0%26adsize%3D160x600
```

The key-values are based on your Yieldprobe settings, separated by ampersand and encoded.

Default Keys

Parameter	Description
id	Adslot Id
price	Price in cents
advertiser	Advertiser name
curl	Campaign url
format	Special Format Id (see section "Special Formats")
adsize	Size of the ad (e.g. 160x600)

Redirect via a VAST wrapper

If you use the parameter "type=vast", the Yieldprobe will not result in a HTTP redirect but will deliver a VAST wrapper using the URL as defined above.

The following example shows the sequence of the call:

1. Call to the YP with an URL (including the macro [ypkws] - you can put it wherever you want) appended to the parameter "redirect"

```
https://ad.yieldlab.net/yp/24062?ts=903991786996&redirect=http://adserver.com?keywords=[ypkws]&type=vast
```

2. The response will be a VAST wrapper pointing to the URL with the [ypkws] macro replaced by key-values

```
<VAST version="2.0"><Ad id="Yieldprobe"><Wrapper><AdSystem>Yieldlab</AdSystem><VASTAdTagURI><![CDATA
[https://adserver.com?keywords=id%3D12345%26price%3D300%26advertiser%3Dadvertiser.com%26curl%3Dcampaign.
com%26format%3D0]]></VASTAdTagURI><Impression></Impression></Wrapper></Ad></VAST>
```

Cookieless

We are introducing a new attribute in the Yieldprobe Response, so we can directly match the recommendation to the delivered adtag without relying on cookie ids. This is especially important for mobile web/mobile app/server side scenarios.

Attributes	Required	Description
pvid		String value to be appended onto the adtag. Matching between Yieldprobe and adtag. The same for every recommendation of one page view. E.g. 721c642d-6941-467e-9afd-b30ea215fd76

Additionally to other Key-Values, you also have to pass the pvid to the adserver. E.g.

https://youradserver.com/1234/9876/?kw=yl3418=1,yl3418price=300,yl3418size=160x600,pvid=721c642d-6941-467e-9afd-b30ea215fd76

And append it to the adtag you are delivering. Most adservers have macros for passed key-values ({adser_macros}).

https://ad.yieldlab.net/d/3418/8877?ts=[timestamp]&pvid={value_of_the_pvid}

AMP RTC

Yieldprobe can be used in Google AMP via AMP RTC (Real Time Config). As Yieldlab is a registered vendor this is simple and straightforward.

For every amp-ad element, you have to define the rtc config with yieldlab as a vendor and a corresponding adslot id. See an example for adslot id "1234" below.

```
<amp-ad width="320" height="50" type="network-foo" data-slot="/1234/5678" rtc-config='{
        "vendors": {
                "yieldlab": {"ADSLOT_ID": "1234"}
        },
        "timeoutMillis": 750}'>
</amp-ad>
```

The default key-values that are passed to your adserver automatically are the following.

Key	Value
id	Numeric value for the adslot ID (e.g. 3418)
price	Numeric value for the CPM price in cents (e.g. 125)
advertiser	String containing the advertiser name (e.g. "yieldlab")
curl	String containing the campaign URL (e.g. "www.yieldlab.de")
adsize	String containing the size of the ad (e.g. "160x600")
adtype	String containing the ad type (e.g. "banner")

If you want additional key-values (dealid, for example) or want them renamed (e.g. "yl_price") speak to your account manager.

For more info, see the official AMP publisher implementation guide:

https://github.com/ampproject/amphtml/blob/master/extensions/amp-a4a/rtc-publisher-implementation-guide.md

Prebid

Publishers can use Prebid instead of integrating the Yieldprobe directly. You can find the most recent version of the Yieldlab Prebid adapters here:

Prebid.js: https://github.com/prebid/Prebid.js

Prebid Server Go: https://github.com/prebid/prebid-server

Prebid Server Java*: https://github.com/prebid/prebid-server-java

**Please note that Java adapter is currently not maintained by Yieldlab but by Prebid.org, so updates may be delayed.*

Documentation and integration guidelines

Please refer to this list of the bidder params and adserver keys for a Yieldlab integration:

https://docs.prebid.org/dev-docs/bidders/yieldlab.html

Example configuration:

https://github.com/prebid/Prebid.js/blob/master/modules/yieldlabBidAdapter.md

Multisize

Prebid.js: If you are using multisize adslots, please refrain from using the optional "adSize" in the Bidder params, because it overrides the dynamic ad sizes passed by the bidder.

Supported features, platforms and media types

Platform & Media types

Web	**Prebid.js**	**Prebid Server Go**	**Prebid Server Java**
Banner			
Video			
Native			
Mobile Web	**Prebid.js**	**Prebid Server Go**	**Prebid Server Java**
Banner			
Video			
Native			
App	**Prebid.js**	**Prebid Server Go**	**Prebid Server Java**
Banner			
Video			
Native			
AMP	**Prebid.js**	**Prebid Server Go**	**Prebid Server Java**
Banner			
Video			
Native			

Features

Feature	**Prebid.js**	**Prebid Server Go**	**Prebid Server Java**
Floor module			

Extended IDs			
Content signals			
Dynamically restrict ad sizes			
Supply chain			

Changelog

21.05.2024

- added Google Topics parameters

16.02.2024

- added DSA parameters
- updated Prebid bidder documentation URL

16.06.2023

- added targeting parameter "t"

03.03.2023

- added information about feature support for Prebid.js / Prebid Server Java / Prebid Server Go adapters

31.01.2023

- added "atypes" for extended ids

15.09.2022

- added more "iab_content" signals and further description of the syntax

29.03.2022

- added key "native" to the yieldprobe response

31.08.2021

- added parameter "iab_content" to request endpoints

17.03.2021

- added a clarification for handling multisize adslots with prebids.js

26.01.2021

- added "skip" parameter to video adslots

21.08.2020

- added "ids" and "gp" parameters to request endpoints

20.06.2020

- added "schain" parameter to request endpoints

17.03.2020

- added "partnershipType" parameter in Yieldprobe response
- added examples for "floor" parameter

09.03.2020

- added "format" in Yieldprobe response as deprecated

20.01.2020

- added Prebid documentation
- added "adtype" parameter in Yieldprobe response

10.12.2019

- added multisize adslot specifications
- adtag width/height parameters are now optional

05.11.2019

- added "pubstoreurl" as new parameter

14.08.2019

- added "pubbundlename" as new parameter

26.06.2019

- added "amp" as a content output type
- added "pubref" and "pubappname" as new parameters
- deleted deprecated section "Showroom"
- deleted deprecated section "Non-Standard Formats"