

Integration & setup

This book guides you through the process of setting up your Adhese integration.

- [Intro](#)
- [Connecting to Adhese](#)
 - [Typescript Web SDK](#)
 - [Prebid](#)
 - [Request API](#)
- [Video & Audio Advertising](#)
 - [VAST](#)
 - [Instream & Outstream](#)
- [Digital Out Of Home](#)
 - [DOOH endpoints](#)
- [General JSON response structure](#)
- [Request target parameters](#)
- [Attribute Sync Service](#)
- [Supported Currencies](#)
- [IP override for server-side connections](#)

Intro

Adhese Gateway and Direct Ad Server

Gateway and Ad Server work entirely server-side, keeping all client implementations as simple as possible to avoid exposing business rules and configuration. This allows for easy integration across many different platforms and devices without the need for complicated development.

Once implemented, no further client-side changes are required. Publishers can enable new demand sources, add or modify the data they share with their partners, and set business rules, multipliers and exceptions through a centralised web application. These changes are executed instantly and require no intervention from web administrators or application developers.

The implementation for using Gateway or Direct Ad Server technologies is the same. Below are the links to the various repositories with examples and code.

Libraries & SDKs

Adhese has a TypeScript based SDK for web integrations. The SDK and it's documentation can be found via the following links:

[TypeScript Adhese SDK](#) and it's [documentation](#)

Adhese also provides an open-source JavaScript SDK for implementation in web-based environments. It can be used standalone or integrated with various existing JS frameworks and wrappers. Developers pick and choose the parts they need to keep the code as lightweight as possible. The JavaScript SDK should be considered as a legacy implementation.

[Adhese JavaScript Library](#) on [GitHub](#)

Prebid Compatible

Gateway is available as a bidder for both Prebid.js and Prebid Server and has been used in various custom wrappers created by publishers and SSPs.

[Prebid.js Bidder on Prebid](#) [Prebid Server Bidder on Prebid.org](#)

Native Mobile SDK

An open-source native Android and iOS SDK is freely available. Mobile apps can implement the Adhese instance as a pure API, giving them full control over the user experience.

[Adhese for Android on GitHub](#)

[Adhese for iOS on GitHub](#)

API implementation

Any publisher can implement their Adhese instance as a pure API and call the ad server endpoints directly from their mobile app, CMS system, connected TV app, ... In one request, they receive campaigns for multiple placements that can be visualised according to device and context.

Detailed documentation is available at [Server Side Ad Request Endpoint](#).

Adhese Ad Tags

There are several ways to integrate the Adhese ad tag into your platforms:

- The most common implementation method is JSON or JSONP. The JSON or JSONP method bundles all ads into one request and visualises the creatives in the reserved spots on your site. With this method, it is easy to consider when an ad has an actual change to be seen ('viewability'). JSON or JSONP is best suited for use in a responsive environment.
- Classic or legacy method: tag.js
The legacy method puts JavaScript functions into containers. The request to the ad server results in a `document.write` instruction on the implementing page.

Migrating your old ad server to Adhese

Depending on the customer's requirements, migration from an existing ad server to Adhese can be achieved in several ways.

One approach is to focus on getting **the Adhese tags** up and running. Based on the inventory setup, all tags are made available. Existing tags from the legacy ad server can be implemented in Adhese as a standard campaign, serving the legacy tags as third-party ads. One hundred percent of the inventory is sent to these campaigns to ensure continuity. New campaigns are booked in Adhese and can take priority over the legacy tags if required. Eventually, the legacy campaigns will stop running as no more traffic is required, and any new or updated campaigns will be booked directly into Adhese.

Another approach is to postpone tagging and **start booking new campaigns in Adhese**. Adhese tags or third-party tags can then be uploaded to the legacy ad server. You can do this on a per-campaign basis, or send 100% of the traffic to these Adhese campaigns in the legacy system. As the tags are distributed across the client's network, they will show the same Adhese creatives but will either be called directly through the Adhese tags or passed through the tags of the legacy ad server.

Connecting to Adhese

Connecting to Adhese

Typescript Web SDK

The Adhese Typescript SDK for web implementations can be found on
https://adhese.github.io/sdk_typescript/

Connecting to Adhese

Prebid

Adhese is available as a bidder for both Prebid.js and Prebid Server and has been used in various custom wrappers created by publishers and SSPs.

Once connected you can use the Adhese adserver to run your own campaigns, connect to other Adhese instances or connect to [Adhese Gateway](#).

[Prebid.js Bidder on Prebid](#) + [Prebid Server Bidder on Prebid](#)

Connecting to Adhese

Request API

For integrations where you don't want to use an SDK or other library, we provide the request endpoint so you can build the request URLs yourself. Both GET and POST requests are supported, but there is a slight difference between them.

POST endpoint

This endpoint provides access to the ad server and gateway functionality for server applications. This functionality allows implementers to send an HTTP POST request to their Adhese endpoint, returning an array of ads for the requested ad slots.

The POST body can contain a number of parameters, the structure of which is documented below.

The API is public and requires no authentication. Any client-side framework that can interpret a JSON response can request the same endpoint.

Endpoint URL

To obtain the endpoint URL for your account, contact [Adhese Support](#).

Default look of the endpoint URL:

```
https://ads-{customer}.adhese.com/json/
```

Request Type

HTTP POST request over an HTTPS connection. In a production environment, no cookies are needed; all data is passed in the request body.

For debugging purposes, add a 'set-cookie' header with the value 'debugKey=[any_value]'. This will make all activity from your requests available in [Gateway Debug Logging](#).

Do not use this debug cookie for production requests. If used on a large scale, it will add significant latency to your responses.

POST Body

The body of your POST request can contain JSON following a fixed structure. There are three main properties.

```
{
  "slots": [
    {
      "slotname": "some_slot"
    },
    ...
  ],
  "parameters": {
    "kw": [
      "cheese",
      "wine"
    ],
    ...
  },
  "user": {
    "ext": {
      "eids": []
    }
  }
}
```

slots

An array of slot objects containing all the ad units required for a given page or state of the end-user application. At least one slot is required.

Each slot contains at least a 'slotname', which is a unique identifier for a specific advertisement placement. This string is stored in the Adhese database and used for targeting campaigns.

```
"slots": [
  {
    "slotname": "some_slot"
  },
  {
    "slotname": "another_slot"
  }
]
```

parameters

An object that contains a set of properties and their corresponding array of values. The purpose of this object is to define one or more attributes that tell you something about the page, user, context, etc.

The parameter keys are fixed two-character codes defined in your Adhese account. The value is always an array of strings. All parameter properties are optional. You can leave them out if they are empty or irrelevant, or send an empty array as value.

```
"parameters": {  
  "kw": [  
    "cheese",  
    "wine"  
  ],  
  "mi": [  
    "ABCDEF123456"  
  ]  
}
```

In the example above, “kw” is used for “search keyword” and “mi” for “member id.”

slot level parameters

A slot can also contain a 'parameters' attribute that follows the same structure. Parameters can be defined at both the request and slot levels. The values of these parameters will be merged for each corresponding slot.

```
"slots": [  
  {  
    "slotname": "some_slot",  
    "parameters": {  
      "ps": "top"  
    }  
  },  
  {  
    "slotname": "another_slot",  
    "parameters": {  
      "ps": "bottom"  
    }  
  }  
]
```

user

The user property is reserved for situations where external ID providers are available, and their identifiers need to be transferred to buyers further down the line. This is mainly useful in an open market context. For more info, please contact our Support Team.

Example of a complete request body

```
{
  "slots": [
    {
      "slotname": "homepage_banner"
    },
    {
      "slotname": "homepage_rectangle_top"
    },
    {
      "slotname": "homepage_rectangle_bottom"
    },
    {
      "slotname": "homepage_halfpage",
      "parameters": {
        ...
      }
    }
  ],
  "parameters": {
    // Example user ID
    "id": [
      "1234567890ABCD"
    ],
    // Example product categories
    "ct": [
      "computers",
      "laptops"
    ],
    // search keywords
    "kw": [
      "cheese",
      "wine"
    ]
  }
}
```

Response Codes

200

All is OK. If no advertisements are available, an empty array will be in the response body.

454

No slots were added to the request body, so no advertisements can be returned. An error message containing the message 'slots cannot be empty' is added to the response as header 'x-adhese-bad-request'.

442

A duplicate slotname has been used for one or more slots. An error message is added to the response as header 'x-adhese-bad-request' containing message 'request contains duplicate slots: [list of duplicate slots]'

500

Internal server error. Further information could be available in the debug log if it was a debug request or via Adhese Support.

Response Object

To be moved to its own page

The response object contains a number of attributes. A full description can be found here: <https://github.com/adhese/sdk#response-object-structure>.

The relevant attributes for a server-side integration are the following:

slotName

A string containing the name of the slot for which this response is intended. When requesting multiple slots simultaneously, this can be used as a key to link the responses to the slots you have added to the request body.

tag

The ad markup used to render the advertisement on the client is returned as a string and can be of any type interpretable by the client or requester.

For typical web and display advertising, the markup is usually a fragment of HTML that can be inserted into a container on the page.

For video or audio advertising, the markup is usually a VAST-compliant XML document.

For native advertising, the markup typically comprises a JSON object containing all the elements required by a client application or browser script to render the ad correctly.

The markup can contain JSON as a string, which the application consuming the responses must parse to get the actual JSON.

width

The ad container width in pixels is needed for the ad to display as intended.

height

The ad container height in pixels is needed for the ad to display as intended.

trackedImpressionCounter

The client application can call a unique URL when the ad is added to a page, even if it is not yet visible and the user has to scroll to see it.

The call to the URL can be asynchronous, and the response can be ignored.

The call to this URL registers a 'paid impression' for a specific ad, as defined by the IAB.

viewableImpressionCounter

The client application can call a unique URL when the ad is visible in the viewport for at least 50% and for at least one second. The call to the URL can be asynchronous, and the response can be ignored.

A call to this URL will register a 'viewable impression', a metric defined by the IAB and used in digital advertising.

clickTag

The clickTag attribute contains a unique URL that can be used to create a link that is counted as a 'click' on the ad and redirects the user to the ad's landing page.

When implemented in an application that does not use links or URLs, the clickTag URL can register a click while ignoring the response, just like the impression counters.

Event tracking

To be moved to its own page

Creating an Event Tracking URL

You can create unique URLs based on the JSON output to register any event related to the ad once it is rendered on screen.

A tracking event URL structure looks like this:

```
https://[ad_request_host]/track-  
[your_event_string]/[response.id]/sl[response.slotID]/II[response.impressionID]
```

Adding an Event Tracking URL to a template

In an Advar or HTML5 template, event-tracking URLs can be created using server-side request parameters. A tracking event would then look like this:

```
https://[adheseDomain:ad_host]/track-  
[your_event_string]/[adheseLogID]/sl[adheseReplace:sl]/II[adheseReplace:II]
```

Example of a full response object

```
[  
  {  
    "dm": "dmDMGo2g0c0;ADV0",  
    "adType": "714x224",  
    "adFormat": "714x224",  
    "timeStamp": "1642511791000",  
    "share": "0",  
    "priority": "0",  
    "orderId": "2",  
    "adspaceId": "2",  
    "adspaceKey": "CI-ADH-12345678",  
    "body": "",
```

```

        "trackingUrl": "",
        "tracker": "https://ads-
demo.adheses.com/track/121//sl6/brChrome/brChrome97/brOSX/brdesktop/dtdesktop/ogcontrol/II19656e
e55-a3cd-4a39-8ed5-28f575a2e9b1/tlnone/ctaardappel-groente-fruit/ctkruiden-knoflook-
pepers/umknown/kwkaas/om0/cuc/cxveel-
gebruikt:snel/cxkeuken:italiaans/cxmomenten:wat_eten_we_vandaag/id1234567890ABCD/miAH009988776
6/A2127.68.78.84/?t=1642610778623",
        "extraField1": "",
        "extraField2": "",
        "altText": "",
        "height": "224",
        "width": "714",
        "tag": "{
          '_links': {
            'self': {
              'href': 'https://ads-
demo.adheses.com/raylene//sl6/brChrome/brChrome97/brOSX/brdesktop/dtdesktop/ogcontrol/II19656e5
5-a3cd-4a39-8ed5-28f575a2e9b1/tlnone/ctaardappel-groente-fruit/ctkruiden-knoflook-
pepers/umknown/kwkaas/om0/cuc/cxveel-
gebruikt:snel/cxkeuken:italiaans/cxmomenten:wat_eten_we_vandaag/id1234567890ABCD/miAH009988776
6/A2127.68.78.84/ad121-4/UR/acties/'
            },
            'resourceType': 'Editorial',
            'type':
'HalfWidthSmallResponsiveEditorial',
            'position': {
              'x': 'left',
              'y':
'top'
            },
            'contentSize': 'half',
            'color': 'drop',
            'text': {
              'title': 'Maak
kans op een exclusieve Bud Dreams Jacket',
              'body': 'Bekijk de actie'
            },
            'images':
[
              {
                'title': '<IMAGE_TITLE>',
                'width': 714,
                'height':
224,
                'link': {
                  'href': 'https://pool-
demo.adheses.com/pool/lib/4_2nd_1.jpg'
                },
                {
                  'title':
'<IMAGE_TITLE>',
                  'width': 714,
                  'height': 224,
                  'link':
{
                    'href': 'https://pool-demo.adheses.com/pool/lib/4_2nd_1.jpg'
                  }
                }
            ],
            'navItem': {
              'title': 'Bekijk de actie',
              'link':
{
                'href': 'https://ads-
demo.adheses.com/raylene//sl6/brChrome/brChrome97/brOSX/brdesktop/dtdesktop/ogcontrol/II19656e5
5-a3cd-4a39-8ed5-28f575a2e9b1/tlnone/ctaardappel-groente-fruit/ctkruiden-knoflook-
pepers/umknown/kwkaas/om0/cuc/cxveel-
gebruikt:snel/cxkeuken:italiaans/cxmomenten:wat_eten_we_vandaag/id1234567890ABCD/miAH009988776
6/A2127.68.78.84/ad121-4/URhttps://ah.nl/acties/winactie-bud'
              }
            }
          }
        }",
        "tagUrl": "https://pool-demo.adheses.com/pool/lib/4_0.advar",
        "heightLarge": "224",
        "widthLarge": "714",
        "libId": "4",
        "id": "121",
        "advertiserId": "0",
        "orderProperty": "brands-budweiser",
        "ext": "advar",
        "swfSrc": "https://pool-demo.adheses.com/pool/lib/4_0.advar",
        "url": "https://ads-
demo.adheses.com/raylene//sl6/brChrome/brChrome97/brOSX/brdesktop/dtdesktop/ogcontrol/II19656e5
5-a3cd-4a39-8ed5-28f575a2e9b1/tlnone/ctaardappel-groente-fruit/ctkruiden-knoflook-
pepers/umknown/kwkaas/om0/cuc/cxveel-
gebruikt:snel/cxkeuken:italiaans/cxmomenten:wat_eten_we_vandaag/id1234567890ABCD/miAH009988776
6/A2127.68.78.84/ad121-4/UR",
        "clickTag": "https://ads-
demo.adheses.com/raylene//sl6/brChrome/brChrome97/brOSX/brdesktop/dtdesktop/ogcontrol/II19656e5
5-a3cd-4a39-8ed5-28f575a2e9b1/tlnone/ctaardappel-groente-fruit/ctkruiden-knoflook-
pepers/umknown/kwkaas/om0/cuc/cxveel-
gebruikt:snel/cxkeuken:italiaans/cxmomenten:wat_eten_we_vandaag/id1234567890ABCD/miAH009988776
6/A2127.68.78.84/ad121-4/UR",
        "swfSrc2nd": "https://pool-demo.adheses.com/pool/lib/4_2nd_1.jpg",

```

```

"swfSrc3rd": "",
"swfSrc4th": "",
"poolPath": "https://pool-demo.adheses.com/pool/lib/",
"comment": "",
"adDuration": "0",
"adDuration2nd": "0",
"adDuration3rd": "0",
"adDuration4th": "0",
"orderId": "Example Bud Dream Jacket",
"creativeName": "Bud Dream Jacket",
"deliveryMultiples": "free",
"deliveryGroupId": "o2g0c0",
"adspaceStart": "1630447200000",
"adspaceEnd": "1643669999000",
"swfSrc5th": "",
"swfSrc6th": "",
"adDuration5th": "0",
"adDuration6th": "0",
"width3rd": "0",
"width4th": "0",
"width5th": "0",
"width6th": "0",
"height3rd": "0",
"height4th": "0",
"height5th": "0",
"height6th": "0",
"slotName": "ah.nl-test_bonus_bonusregulier_left-714x224",
"slotID": "6",
"impressionCounter": "https://ads-
demo.adheses.com/track/121//sl6/brChrome/brChrome97/brOSX/brdesktop/dtdesktop/ogcontrol/II19656
e55-a3cd-4a39-8ed5-28f575a2e9b1/tlnone/ctaardappel-groente-fruit/ctkruiden-knoflook-
pepers/umknown/kwkaas/om0/cuc/cxveel-
gebruikt:snel/cxkeuken:italiaans/cxmomenten:wat_eten_we_vandaag/id1234567890ABCD/miAH009988776
6/A2127.68.78.84/?t=1642610778623",
"trackedImpressionCounter": "https://ads-
demo.adheses.com/track/121//sl6/brChrome/brChrome97/brOSX/brdesktop/dtdesktop/ogcontrol/II19656
e55-a3cd-4a39-8ed5-28f575a2e9b1/tlnone/ctaardappel-groente-fruit/ctkruiden-knoflook-
pepers/umknown/kwkaas/om0/cuc/cxveel-
gebruikt:snel/cxkeuken:italiaans/cxmomenten:wat_eten_we_vandaag/id1234567890ABCD/miAH009988776
6/A2127.68.78.84/?t=1642610778623",
"viewableImpressionCounter": "https://ads-demo.adheses.com/track/121-
Adheses_IABview/sl6/brChrome/brChrome97/brOSX/brdesktop/dtdesktop/ogcontrol/II19656e55-a3cd-
4a39-8ed5-28f575a2e9b1/tlnone/ctaardappel-groente-fruit/ctkruiden-knoflook-
pepers/umknown/kwkaas/om0/cuc/cxveel-
gebruikt:snel/cxkeuken:italiaans/cxmomenten:wat_eten_we_vandaag/id1234567890ABCD/miAH009988776
6/A2127.68.78.84/?t=1642610778623",
"additionalCreatives": [],
"origin": "JERLICIA",
"originData": {},
"auctionable": "true",
"additionalViewableTracker": "",
"additionalCreativeTracker": "https://ads-demo.adheses.com/track/121-additional-
4/sl6/brChrome/brChrome97/brOSX/brdesktop/dtdesktop/ogcontrol/II19656e55-a3cd-4a39-8ed5-
28f575a2e9b1/tlnone/ctaardappel-groente-fruit/ctkruiden-knoflook-
pepers/umknown/kwkaas/om0/cuc/cxveel-

```



```
gebruikt:snel/cxkeuken:italiaans/cxmomenten:wat_eten_we_vandaag/id1234567890ABCD/miAH009988776
6/A2127.68.78.84/?t=1642610778623",
  "extension": {
    "mediaType": "banner",
    "prebid": {
      "cpm": {
        "amount": "13.047",
        "currency": "EUR"
      }
    }
  }
}
```

Video & Audio Advertising

VAST

Adhese fully supports the VAST protocol, which was released by the Interactive Advertising Bureau (IAB) to empower video ad serving through a video platform. VAST facilitates the scalable distribution of video ads across a set of video players from different technology vendors. By supporting VAST, ad servers can implement a single ad response format suitable for a range of different video players.

If a publisher wishes to serve ads to video players, it must develop distinct video ad tags based on each video player's technology. The IAB has created a standard for video advertising, VAST, which allows for scalable delivery of ads to video players from different vendors.

VAST enables ad servers to utilise a single ad response format, independent of the video player's technology. The video player must be capable of requesting and parsing XML documents. Therefore, by utilising XML, VAST serves as a universal solution for video ad serving, comparable to HTML for browser-based ad serving.

Since the initial release of VAST, online video technology has advanced, enabling more sophisticated possibilities for online video advertising. To support this evolution, VAST has been enhanced with additional features and functionality. VAST 4.0 represents the current industry standard.

Adhese fully supports VAST: we are compliant with VAST 2.0, 3.0, and 4.0 and can help you adopt its guidelines.

More info about the VAST standard can be found on the [IAB's website](#)

VAST offers a set of features intended to make online video advertising easy.

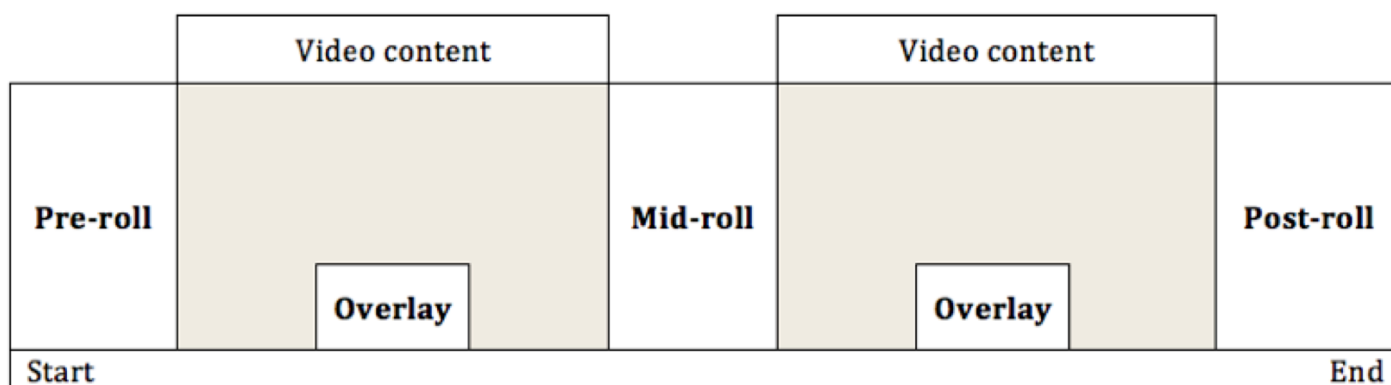
- **Platform—and device-agnostic:** Nothing about the protocol of VAST is specific to the functioning of a certain device or platform. This allows Adhese to serve ads across several video players in different situations, such as websites, mobile websites and applications, Internet-connected TVs, or set-top box environments.
- **Support for different ad types:** VAST supports different video ad formats. Next to the well-known linear and non-linear video ad formats, VAST enables the delivery of companion ads, skippable linear ads, and ad pods.

- ****Tracking:****To provide details about the delivered ads, VAST enables the simultaneous tracking of several user events related to the video ad. That is to say, VAST can tell you, for example, whether a video ad was completely viewed or if the user has muted the sound. VAST can also notify you if a user skipped a linear ad by explicitly closing it before completion, or if the user clicked a non-linear ad away.
- **Detailed error reporting:** When a video player cannot display an ad, VAST enables the player to provide specific feedback to the ad server about why the ad can not be served.

Instream & Outstream

The biggest difference between instream and outstream video ads is the placement. Instream video ads are placed within an existing video player that will be used to play content, while outstream video ads are placed in a standalone player which gets embedded in a page similar to a display banner.

Instream



Linear

Linear video ads are played before, between or after the playback of video content. Linear video advertising is known to interrupt the playback of a video clip, with the linear ad taking over the full video experience for a period of time. There are three distinct formats of linear video advertising:

- **Pre-roll** ads play before the start of the video playback;
- **Mid-roll** ads play during the playback of the video clip;
- **Post-roll** ads play after the end of the video playback.

Non-linear

Non-linear video ads do not disrupt the playback of a video; they run alongside the video content within the video player for a brief period of time or after the ad is clicked away. The original video content remains visible throughout the duration of the non-linear video ad, which is displayed in a portion of the video player. Non-linear video ads are typically displayed in the bottom area of a

video player.

An **overlay ad** is a banner ad delivered over the video content at the bottom of the video player. The ad uses text, graphics, or video overlays to convey the message of the advertiser.

Companion ads

To enhance the campaign's visibility, a linear or non-linear video ad can be paired with a companion ad that is in tune with the original video ad. A companion ad is served outside the video player's environment. Any display advertising format can be coupled.

An example of a companion ad is a **branded player**. A branded player consists of an outer layer or skin that is wrapped around the video player.

Outstream

Video creatives that are used for linear advertising can also be used for outstream advertising.

Digital Out Of Home

DOOH endpoints

Adhese has three endpoints in place for DOOH integrations:

1. Assets Download Endpoint (aka Headsup request)
`https://headsup-[customer].adhese.org/api/headsup/download-list/sl[positioncode]`
2. Playlist Endpoint with a limited number of ads
`https://ads-[customer].adhese.com/m/stack/sl[positioncode]?max_ads=[amount]`
3. Playlist Endpoint with unlimited amount of ads
`https://ads-[customer].adhese.com/e/stack/sl[positioncode]`

The Headsup and Unlimited Playlist endpoints must be enabled by Adhese support before they can be used.

Assets download endpoint

To ensure the assets become available in the response, the "Use in heads-up file" setting must be enabled on the format level. All admin users can perform this action in the Adhese UI.

```
https://headsup-[customer].adhese.org/api/headsup/download-list/sl[positioncode]
```

Parameters	Value	Required
customer	The name of your Adhese account	Yes
position code	The code of the position you wish to request: [location code]-[format code]. This value is always prefixed by 'sl'.	Yes

The headsup endpoint returns **all** the materials for the campaigns running on the requested DOOH position. The JSON response consists of an array "media" that may be empty or populated with

multiple ad objects, depending on the number of active campaigns.

An example response:

```
{
  "media": [
    {
      "ad": {
        "id": "https://pool-demo.adhese.com/pool/lib/562_2nd_1.mp4",
        "mime": "video/mp4",
        "curl": "https://pool-demo.adhese.com/pool/lib/562_2nd_1.mp4",
        "filesize": 15470592,
        "checksum": "071f717962db99cc137d138696d33209f2e4818d42a54f70dfa6606eeb1b640b"
      }
    },
    {
      "ad": {
        "id": "https://pool-demo.adhese.com/pool/lib/560_2nd_1.mp4",
        "mime": "video/mp4",
        "curl": "https://pool-demo.adhese.com/pool/lib/560_2nd_1.mp4",
        "filesize": 15470592,
        "checksum": "4e78745a33518ff22c1c9f39852a49c1c0fadd0adf722e3394ad1215d5e5ff5b"
      }
    },
    {
      "ad": {
        "id": "https://pool-demo.adhese.com/pool/lib/561_2nd_1.mp4",
        "mime": "video/mp4",
        "curl": "https://pool-demo.adhese.com/pool/lib/561_2nd_1.mp4",
        "filesize": 6582272,
        "checksum": "64d2eab6d51b208c0cec3fc4d64c53381e5d41ae2a1d4c5b7704b1818bdb6158"
      }
    }
  ]
}
```

Playlist endpoints

With an amount limit

```
https://ads-[customer].adhese.com/m/stack/sl[positioncode]/pi[playerid]?max_ads=[amount]
```

Parameters	Value	Required
customer	The name of your Adhese account	Yes

Parameters	Value	Required
position code	The code of the position you wish to request is [location code]-[format code]. This value is always prefixed by 'sl'.	Yes
max_ads	The maximum number of ads you wish to request	Yes
player ID	The ID of the player. The value needs to be prefixed by a 2-letter code. Adhese support needs to configure this code.	No

The playlist endpoint returns a subset of DOOH campaigns at the requested DOOH position. The JSON response consists of an array "ads" which can be empty or populated with multiple objects depending on the number of active campaigns.

The **max_ads** parameter defines the maximum number of returned ads. A separate configuration on the adserver can be activated to ensure this value never exceeds a specific limit.

A player ID can be provided as an optional parameter in the request. This ID can be used for a number of purposes:

- Targeting campaigns on specific players
- Creating reports on player level
- Server-side mapping of additional data. This data is then available for targeting and reporting

The structure of the JSON object that is returned for each campaign must be configured in the adserver using an advar template.

An example response:

```
{
  "ads": [
    {
      "id": "https://pool-demo.adhese.com/pool/lib/562_2nd_1.mp4",
      "dur": 28.07, "prio": 15,
      "booking_prio": 0,
      "publisher": "1",
      "key": "",
      "proofOfPlay": "https://ads-demo.adhese.com/track/3474/sl357/tlnone/piplayer_id/A2?1746011926824",
      "error": "https://ads-demo.adhese.com/track/3474-PLAY_ERROR_[ERRORCODE]/sl357/tlnone/piplayer_id/A2/?1746011926824",
      "frequency": "1;1;1",
    }
  ]
}
```

```

        "starttime": "0:00",
        "stoptime": "not set"
    },
    {
        "id": "https://pool-demo.adhese.com/pool/lib/561_2nd_1.mp4",
        "dur": 11.45, "prio": 15,
        "booking_prio": 0,
        "publisher": "1",
        "key": "",
        "proofOfPlay": "https://ads-
demo.adhese.com/track/3444/sl357/tlnone/piplayer_id/A2?1746011926824",
        "error": "https://ads-demo.adhese.com/track/3444-
PLAY_ERROR_[ERRORCODE]/sl357/tlnone/piplayer_id/A2/?1746011926824",
        "frequency": "2;1;1",
        "starttime": "not set",
        "stoptime": "not set"
    },
    {
        "id": "https://pool-demo.adhese.com/pool/lib/560_2nd_1.mp4",
        "dur": 18.85, "prio": 15,
        "booking_prio": 0,
        "publisher": "1",
        "key": "",
        "proofOfPlay": "https://ads-
demo.adhese.com/track/3455/sl357/tlnone/piplayer_id/A2?1746011926824",
        "error": "https://ads-demo.adhese.com/track/3455-
PLAY_ERROR_[ERRORCODE]/sl357/tlnone/piplayer_id/A2/?1746011926824",
        "frequency": "2;1;1",
        "starttime": "not set",
        "stoptime": "not set"
    }
]
}

```

Without an amount limit

```
https://ads-[customer].adhese.com/e/stack/sl[positioncode]/pi[playerid]
```

Parameters	Value	Required
customer	The name of your Adhese account	Yes
position code	The code of the position you wish to request: [location code]-[format code]. This value is always prefixed by 'sl'.	Yes
player ID	The ID of the player. Value needs to be prefixed by a 2-letter code. Adhese support needs to configure this code.	No

The response for the playlist endpoint without a limit is identical to the one with a limit. However, you will receive all available ads rather than a subset.

General JSON response structure

The following code block is an extracted example of a JSON object. A list of all available JSON fields and their descriptions can be found below the code block.

```
{
  ...
  "adFormat": "wideskyscraper",
  "adspaceId": "61721",
  ...
  "adspaceStart": "1433714400000",
  "adspaceEnd": "1483225199000",
  "adType": "SKY",
  ...
  "creativeName": "Example Billboard News",
  ...
  "deliveryMultiples": "free",
  ...
  "ext": "swf",
  ...
  "height": "600",
  "id": "295057",
  "libId": "96393",
  "orderId": "16643",
  "orderName": "Example - BillBoard Campaign",
  ...
  "priority": "1",
  ...
  "slotName": "_test-site_homepage_-SKY",
  "swfSrc": "http://1.adhesecdn.be/pool/lib/96393.swf",
  "tag": "<object id='-1756524077' classid='clsid:D27CDB6E-AE6D-11cf-96B8-444553540000'
codebase='http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0'
WIDTH=160 HEIGHT=600><param NAME=movie
VALUE='http://1.adhesecdn.be/pool/lib/96393.swf?clickTAG=http://host4.adhese.be/295057/http%3A%
2F%2Ftrack.adform.net%2FC%2F%3Fbn%3D3515419' /><!--[if !IE]>--><object type='application/x-
shockwave-flash'
data='http://1.adhesecdn.be/pool/lib/96393.swf?clickTAG=http://host4.adhese.be/295057/http%3A%
2F%2Ftrack.adform.net%2FC%2F%3Fbn%3D3515419' width='160' height='600'><!--<![endif]--><param
NAME='quality' VALUE='high' /><param NAME='allowScriptAccess' VALUE='always' /><param
NAME='wmode' VALUE='transparent' /><a target='_blank'
href='http://host4.adhese.be/295057/http://track.adform.net/C/?bn=3515419'><img
src='http://1.adhesecdn.be/pool/lib/96394.jpg'></a><!--[if !IE]>--></object><!--<![endif]--
></object>",
  ...
  "timestamp": "1396357433000",
  "tracker":
```

```
"http://ads.adhese.be/track/295057//sl242////////inadttr12842;adttrbiz;adttrfood;adttrhealth;adttrimmo;adttrlifestyle;adttrmultimedia;adttrsport;adttrtrav;adttrvoetbal;adttrwielrennen/brTelenet N.V./coBE/rgBE11///isTelenet N.V.//////////A2141.135.96.213.1395820307192918/O_/A_/C_",
  "trackingUrl": "http://track.adform.net/adfserve/?bn=3515419;1x1inv=1;srctype=3;ord=",
  "url": "http://host4.adhese.be/295057/http://track.adform.net/C/?bn=3515419",
  "width": "160",
}
```

Field name	Description
additionalCreatives	Lists the additional creatives
adDuration	Duration in seconds of the primary creative
adDuration2nd	Duration in seconds of the second creative file
adDuration3rd	Duration in seconds of the third creative file
adDuration4th	The optional duration of the fourth creative file
adDuration5th	The optional duration of the fifth creative file
adDuration6th	The optional duration of the sixth creative file
adFormat	the assigned format name determined by your Adhese account, e.g. wideskyscraper (the Code export field from the Admin > Formats screen)
adspaceEnd	The end date of the booking in UNIX timestamp
adspaceId	The Adhese booking ID
adspaceKey	An optional creative foreign key
adspaceStart	The start date of the booking in UNIX timestamp
adType	The name of the format as requested, e.g. SKY (the Code tag field from the Admin > Formats screen)
advertiserId	The ID of the advertiser

altText	Optional text to be shown as the value of the `` attribute of the container
auctionable	The auctionable (compete with RTB) setting of a booking, can be used for header bidding
body	The third-party code to be inserted in a container (if applicable)
clickTag	The URL of the click tag used for the counting of clicks and which should be followed by the actual target URL
comment	Optional free text comment
creativeName	The name of the creative as determined in the Adhese interface
deliveryGroupld	The ID of the booking or creative group for all-together or one-at-a-time bookings
deliveryMultiples	The type of delivery
dm	The ID of the delivery limitation
ext	The extension of the file type
extraField1	Optional field used by the uploader
extraField2	Second optional field used by the uploader
height	The height of the primary creative in pixels
height3rd	The height of the third creative file in pixels
height4th	The height of the fourth creative file in pixels
height5th	The height of the fifth creative file in pixels
height6th	The height of the sixth creative file in pixels
heightLarge	The height of the second creative file in pixels

id	The traffic ID of the link between an uploaded creative and a booking
impressionCounter	The URL to count a tracked impression
libId	The ID of the uploaded creative
orderId	The ID of the campaign
orderName	The name of the campaign
orderProperty	An optional comma-separated list of properties containing codes as defined by your Adhese account
origin	String identifying the source of the ad: JERLICIA or RUBICON
originData	An object containing more info related to the origin of the ad
poolPath	An optional path to a CDN where files for this creative can be retrieved
priority	The priority of the campaign
share	An optional number that indicates the weight of the creative
slotName	The value of the prefix `sl` as requested
swfSrc	The URL of the primary creative file
swfSrc2nd	The URL of the second creative file
swfSrc3rd	The URL of the third creative file
swfSrc4th	The URL of the fourth creative file
swfSrc5th	The URL of the fifth creative file
swfSrc6th	The URL of the sixth creative file
tag	The complete HTML code for inserting in the container
tagUrl	An optional URL of the tag's content

timeStamp	The timestamp of the latest change to this creative (can be used for caching)
tracker	The tracker URL that needs to be requested for counting an impressions
trackingUrl	The third-party tracking URL that needs to be requested when visualising the ad's creative
url	The click-through URL
viewableImpressionCounter	The URL to count a viewable impression
width	The width of the primary creative in pixels
width3rd	The width of the third creative file in pixels
width4th	The width of the fourth creative file in pixels
width5th	The width of the fifth creative file in pixels
width6th	The width of the sixth creative file in pixels
widthLarge	The width of the second creative file in pixels

Request target parameters

All target codes which are part of the following list and all codes starting with **x**, **y** or **z** are reserved

Reserved for the Adhese backend			
Code	Reserved	Description	Example
co	yes	Country as alpha 2, based on IP	coBE
CO	yes	Country as alpha 3	
rg	yes	Sub devision of a country, based on IP	rgBE11
ci	yes	City postcode, based on IP	ci9000
da	yes	Reserverd for newsletter implementations	da20140110
il	yes	Adhese impression ID	
pr	yes	Reserved for rotation file 'priority' logic	
SL	yes	Position as a string	
A2	yes	Adhese cookie prefix	
dm	yes	Booking groups (all-together etc)	
dt	yes	device type	
sl	yes	Position (or the combination of a location and template, slot is used as backend name)	sl_nbo_22_156_-LAYER
tl	yes	Binary consent	tlnone tlall
xt	yes	IAB consent string	

Reserved for the Adhese backend			
yd	yes	Device type (based on useragent)	Desktop, console, TV_Device, ...
ys	yes	Device OS (based on useragent)	ChromeOS, Linux, macOS,...
yb	yes	Browser (based on useragent)	Chrome, Edge, Firefox, ...
yp	yes	Device Maker(based on useragent)	Samsung, Philips, Lenovo, ...

Reserved for general use			
Code	Reserved	Description	Example
br	yes	Pre-configured parameter for target group 'brands'. Often used to capture and target device data	brChrome;Chrome7;Mac
in	yes	Pre-configured parameter for target group 'interests'. Often used to capture page data	

OpenRTB			
Code	Reserved	Description	Example
xa	yes	(Gateway specific) Allows passing target information to Dale	xatl,1
xb	yes	(Gateway specific) Bundle ID. For Apple iOS devices pass iTunes ID. For Android devices pass package name (e.g. com.foo.mygame).	iOS: iTunes ID Android: package name
xc	yes	(Gateway specific) Coordinates. Latitude;longitude: two floats separated by a semicolon, e.g. [-90..90];[-180..180].	
xs	yes	(Gateway specific) SHA1-encoded device ID	

OpenRTB			
xn	yes	(Gateway specific) video min duration	
xx	yes	(Gateway specific) video max duration	
xk	yes	(Gateway specific)	xk123
xd	yes	(Gateway specific)	
xu	yes	(Gateway specific)	
xi	yes	(Gateway specific)	
xv	yes	(Gateway specific) Unique visit ID	
x5	yes		
xd	yes		
xe	yes		
xf	yes		
xg	yes		
xh	yes		
xl	yes		
xm	yes		
xo	yes		
xp	yes		
xr	yes		
xs	yes		
xt	yes		
xv	yes		
xy	yes		
xz	yes		
xj	yes		
xq	yes		

Attribute Sync Service

Introduction

The Data Sync Service provides implementers with the option of storing items on the Adhese server, which can be used for inventory, campaign insights, and targeting. The storage can be linked to any kind of identifier as per configuration.

Server Side Memory

The memory service allows an implementer to save certain data points linked to a unique key. The key can be either a user ID, which is only allowed with the user's explicit consent, or any other non-personal key (e.g., a DOOH device ID).

The event is a string that will be saved and reported exactly as sent in. A human-readable label can be added in the Adhese inventory configuration.

The third parameter of the memory service is optional and indicates the time-to-live of the saved data point. If no value is passed, the default of 14 days will be applied.

```
https://ads-[customer].adhese.com/usersync/handlers/[service_handler]/user_sync?id=[event1],[event2]&u=[userid]&ttl=[expiration_in_minutes]
```

Example

```
https://ads-[customer].adhese.com/usersync/handlers/ec_view/user_sync?id=pricewatch&u=123abc&ttl=20160
```

Example with multiple events

```
https://ads-[customer].adhese.com/usersync/handlers/ec_view/user_sync?id=pricewatch,laptop,macbook&u=123abc&ttl=20160
```

Removing Data for a Key or User

All data stored in Redis will expire after a given time. The 'right to be forgotten' is one of the most important reasons, and it is essential to provide a way to remove stored data at the request of the rights holder (the end-user).

To remove data, simply call the user sync service with the user ID or key you want. The service will remove all data stored for that key and then remove the key itself.

Example

```
https://ads-[customer].adhese.com/usersync/handlers/ec_view/user_sync?iu=123abc
```

User sync for Open Market OpenRTB

A very specific application of the Data Sync Service can be used to support the OpenRTB User Sync process.

The User Sync process is designed to match P users with SSP users. We recommend syncing users once every day at the earliest opportunity.

The User Sync service is triggered by Adhese directly on the client, enabling the SSP to establish a connection and transfer the resulting user ID to Adhese.

According to the setup, a simple or complex workflow is available. A complex workflow leads to a 1st party user sync cookie, while a simple workflow leads to a 3rd party user sync cookie.

User Consent: This application of the sync service depends on a cookie that contains a unique ID. Users need to be informed about this and give consent prior to calling the service. Adhese has the contractual right to audit the use of this service to ensure it is legally implemented.

Simple workflow

1. The user navigates to newz.zz
2. Browser calls SSP endpoint `https://ssp.com/sync?partner=adhese`
3. The SSP redirects to the user sync endpoint
4. Browser calls Adhese user sync endpoint `https://user-sync.adhese.com/handlers/ssp/user_sync?u=IH3D8DXL-20-KV8Y`
5. We issue a cookie `Set-Cookie: ssp_uid=IH3D8DXL-20-KV8Y; Domain=.adhese.com; Expires=Fri, 25-Dec-2024 09:57:40 GMT; Path=/`
6. Browser requests an ad on `https://ads-adhese.com/ad/sl_foo_-bar`
7. We read ssp_uid cookie to send it over to the SSP's API

Complex workflow for 1st domain implementations

If a publisher serves on its own domains, the process is more complicated. There are no calls made to *.adhese.com.

1. The user navigates to publisher-domain.com
2. Browser registers request domain with Adhese user sync discovery `https://user-sync.adhese.com/handlers/ssp/user_sync_discovery?domain=adhese.publisher-domain.com&protocol=https`

protocol query parameter is optional, it defaults to https

1. We read the domain parameter
2. We issue a cookie: `Set-Cookie: user_sync_domain=adhese.publisher-domain.com; Path=/; Expires=Fri, 25-Dec-2024 10:16:23 GMT`
3. The browser calls the SSP endpoint `https://ssp.com/sync?p=publisher`
4. SSP redirects to user sync discovery
5. The browser calls Adhese user sync discovery `https://user-sync.adhese.com/handlers/ssp/user_sync_discovery?u=IH3D8DXL-20-KV8Y`
6. We read the user_sync_domain cookie
7. We redirect the call to the user sync endpoint running on the correct domain
8. The browser calls Adhese user sync endpoint `https://adhese.publisher-domain.com/handlers/ssp/user_sync?u=IH3D8DXL-20-KV8Y`
9. We issue a cookie: `Set-Cookie: ssp_uid=IH3D8DXL-20-KV8Y; Expires=Fri, 25-Dec-2024 09:57:40 GMT; Path=/`
10. Browser requests an ad on `https://adhese.publisher-domain.com/ad/sl_foo_-bar`
11. We read the ssp_uid cookie to send it over SSP's API

Supported Currencies

Adhese supports multiple currencies for use in e.g. campaign budgets. The table below lists all currencies currently supported by Adhese.

Currency Code	Currency
EUR	Euro
USD	United States Dollar
GBP	British Pound Sterling
CHF	Swiss Franc
JPY	Japanese Yen
AUD	Australian Dollar
CAD	Canadian Dollar
PLN	Polish Złoty
DKK	Danish Krone
SGD	Singapore Dollar
SEK	Swedish Krona
NOK	Norwegian Krone
CZK	Czech Koruna

IP override for server-side connections

What

By default, all ad requests to the Adhese ad server use the IP of the device that sends the request for all IP-related logic, such as geo-location targeting.

This behaviour makes it impossible to set up user-based location targeting in server-to-server setups, as the ad server would use the server's IP instead of the end user's IP.

To solve this, Adhese provides functionality that overrides the server's IP with the user's IP.

How

The user's IP can be added to the request via the **X-Forwarded-For** header. The ad server will pick up this header and override the default IP address from which the request is sent.

Please [contact Support](#) if you'd like to make use of this feature.