

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.

Revision #13

Created 10 June 2024 13:16:44 by Casper Steuperaert

Updated 4 April 2025 07:30:30 by Casper Steuperaert