



Bidding Player

Prebid + GAM video header bidding

Publisher Integration Guide

A one-tag drop-in. Runs a Prebid auction against limelightDigital, stitches the winning bid into your GAM ad call, plays the result through video.js + Google IMA.

VERSION

v2.5.0

LICENSE

MIT

REPOSITORY

github.com/aryanvani-projects/bidding-player

ENGINE

cdn.jsdelivr.net/gh/aryanvani-projects/bidding-player@v2.5.0/engine/player.js

DOCUMENTATION

aryanvani-projects.github.io/bidding-player/docs.html

TAG GENERATOR

aryanvani-projects.github.io/bidding-player

Contents

01 Overview & how it works

02 Prerequisites

03 Quick start

04 Embed code reference

05 GAM setup

06 Limelight onboarding

07 Consent management

08 Testing & verification

09 Troubleshooting

10 FAQ

01 — Overview & how it works

One `<script>` tag handles everything: lazy-loads video.js + IMA + Prebid, runs the auction, stitches the full `hb_*` targeting set into your GAM tag's `cust_params`, plays the resulting VAST through IMA.

The engine is hosted at versioned, immutable jsDelivr URLs. Pin to a tag — the bytes cannot silently change.

Auction sequence

- 1 Engine** parallel-loads video.js, IMA, Prebid; reads `data-*`; registers a video ad unit.
- 2 Prebid → Limelight:** `POST {host}/ortbh`
- 3 Limelight** responds with a bid (or no-bid). Winning VAST cached at Prebid Cache; UUID returned.
- 4 Engine** stitches full `hb_*` set into GAM `cust_params`; applies floor bias; refreshes correlator.
- 5 IMA → GAM:** fetches VAST; line item targeting `hb_pb` wins.
- 6 GAM** returns wrapper with `%%PATTERN:hb_uuid%%` resolved.
- 7 IMA → Prebid Cache:** fetches and plays the bidder creative; content resumes after.

FALLBACK

Prebid load failure, bidder timeout, or no-bid → engine still calls GAM with `hb_pb=0.00&hb_bidder=none`. Your house line item fills.

02 — Prerequisites

- **Google Ad Manager** account with video inventory; a **Master Ad Tag URL** ending with `correlator=`.
- **Limelight Digital** publisher ID + ad unit ID, plus the bidder host.
- **Domain allowlisted** by Limelight (1-3 business days). Without this, every auction returns no-bid.
- **HTTPS** publisher pages (mixed content blocks IMA + bidder calls).
- **Content video** URL (.mp4 / .m3u8) on your CDN.

#1 CAUSE OF ZERO FILL

The publisher domain not being on Limelight's allowlist. Verify *before* launch.

03 — Quick start

1 Generate the tag

Open the Tag Generator. Fill in version, GAM tag, Limelight publisher / ad-unit IDs, content URL. Copy the resulting `<script>`.

2 Set up GAM

Create the key-values, the VAST wrapper creative, and one line item per \$0.10 bucket. One-time setup — see section 05.

3 Embed

Paste the snippet where the player should render. The engine auto-creates the mount div.

4 Verify

Open with `?debug=true`. See section 08.

5 Monitor

GAM Reports → filter by `hb_bidder=limelightDigital` for impressions, eCPM, fill.

04 — Embed code reference

The **v2** tag uses `data-bidders` — a JSON array of every SSP that should bid in this auction. The Prebid bundle at `data-prebid-url` must include adapters for every bidder listed.

```
<script src="https://cdn.jsdelivr.net/gh/aryanvani-projects/bidding-player@v2.5.0/engine/p
layer.js"
  id="adtech-player-core"
  data-bidders=' [
    {"bidder": "limelightDigital", "params": {"host": "ads-jbi003...", "publisherId": "649
658371", "adUnitId": "972556929", "adUnitType": "video"}},
    {"bidder": "appnexus", "params": {"placementId": "12345678"}}
  ]'
  data-tag="https://pubads.g.doubleclick.net/gampad/ads?iu=/...&output=vast&correlat
or="
  data-video="https://your-cdn.example.com/content.mp4"
  data-prebid-url="https://cdn.jsdelivr.net/gh/aryanvani-projects/bidding-player@v2.
5.0/prebid/prebid-limelight-appnexus.js"
  data-timeout="1200" data-bias="0.10"
  data-autoplay="true" data-muted="true" data-fluid="true"
  async></script>
```

ATTRIBUTE	REQ	DEFAULT	NOTES
<code>src</code>	Yes	—	Versioned jsDelivr URL. Pin to a tag, never <code>@main</code> .
<code>id</code>	Yes	<code>adtech- player-core</code>	Must be exactly this value.
<code>async</code>	Yes	—	Required HTML attribute.
<code>data-bidders</code>	Yes*	—	v2 multi-bidder. JSON array of Prebid bid objects (<code>{bidder, params}</code>); each entry's <code>params</code> shape is per-SSP. *Optional if using the legacy 3-attr shortcut below.
<code>data-pub-id</code> / <code>- adunit-id</code> / <code>- host</code>	Legacy	—	v1 legacy shortcut. Engine builds a 1-element <code>limelightDigital</code> array from these. Ignored when <code>data-bidders</code> is set.
<code>data-tag</code>	Yes	—	GAM master ad tag URL; must end with <code>correlator=</code> .
<code>data-prebid-url</code>	Yes	—	Must include adapters for every bidder in <code>data-bidders</code> .
<code>data-video</code>	No	—	Content video (.mp4 / .m3u8). Required for <code>instream</code> ; omitted for <code>outstream</code> .
<code>data-placement</code>	No	<code>instream</code>	<code>instream</code> plays the ad against your content video (pre-roll). <code>outstream</code> needs no content video — the slot stays collapsed until it scrolls ≥50% into view, autoplays muted, then collapses. See <i>Placement modes</i> below.
<code>data-sticky</code>	No	<code>false</code>	Instream only. <code>true</code> floats the player to the bottom-right corner when it scrolls out of view (keeps playing, × to dismiss) and reflows the live ad to fit. Lifts viewability & completed views.
<code>data-timeout</code>	No	<code>1200</code>	Bidder timeout (ms). Industry default 1500–2000.
<code>data-bias</code>	No	<code>0.10</code>	USD added to winning CPM before bucketing. An explicit <code>0.00</code> disables the bias (passes the raw bucketed CPM through).
<code>data-floor-min</code>	No	—	Reject bids below this CPM (they fall through to your house line item). Omitted = no floor.

ATTRIBUTE	REQ	DEFAULT	NOTES
<code>data-floor-max</code>	No	—	Cap the winning CPM at this value before bias + bucketing. Omitted = no cap.
<code>data-autoplay</code> / <code>-muted</code>	No	<code>true</code> / <code>true</code>	Autoplay requires muted (browsers block unmuted autoplay). Outstream + sticky both rely on muted autoplay.
<code>data-fluid</code>	No	<code>true</code>	Responsive 16:9 vs fixed 640×480.
<code>data-loop</code> / <code>-preload</code> / <code>-vpaid</code>	No	see notes	<code>false</code> / <code>metadata</code> / <code>insecure</code> .
<code>data-div-id</code>	No	<code>comparos-video-placement</code>	Mount div ID. Auto-created if missing.
<code>data-cache</code>	No	(AppNexus public cache)	Prebid Cache server.

Placement modes — instream vs outstream

The `data-placement` attribute picks how the player renders:

MODE	WHEN TO USE	BEHAVIOUR
<code>instream</code> default	You already have a content video (news clips, shows, sports).	The ad plays as pre-roll against your <code>data-video</code> , then content resumes. Best CPMs. Requires a content URL.
<code>outstream</code>	You have no video content — a text article, blog, or listing.	The unit drops into the article body, stays collapsed (zero height) until it scrolls ≥50% into view, autoplays muted, then collapses again when the ad finishes or errors. No <code>data-video</code> needed.

Sticky / floating player **instream only**

Set `data-sticky="true"` to keep an instream player — and the ad on it — visible as the reader scrolls. When the player scrolls out of view it shrinks and pins to the bottom-right corner, still playing; it snaps back inline when scrolled back into view, and a **×** button lets the reader dismiss it. The original layout space is preserved (no content jump), and the live ad creative is resized to fit the floating frame.

WHY ENABLE IT

Floating keeps the impression on-screen instead of scrolling away — it raises **viewability** and **completed-view** rates, the metrics that drive video CPMs. Off by default; opt in per placement.

Onboarding a new SSP / Prebid bundle

The `prebid/` directory holds pre-built Prebid.js bundles. Each `data-prebid-url` picks one. To add a bundle with different adapters, trigger the **"Build Prebid bundle"** workflow in the Actions tab: supply the comma-separated module list and a filename, the workflow runs `gulp build` and commits the result on `main`. Full instructions in `prebid/README.md`.

THREE THINGS MUST STAY IN SYNC

1) `src` + `data-prebid-url` on the same version tag. 2) Every bidder in `data-bidders` must have its adapter in the Prebid bundle. 3) Bundle must be from the *same* `@vX.Y.Z` as the engine. The Tag Generator enforces all three.

05 — GAM setup

One-time setup: **key-values**, a **VAST wrapper creative**, and **price-bucket line items**.

1. Custom key-values

GAM → **Inventory** → **Key-values** → **New key-value**:

KEY	TYPE	VALUES
<code>hb_pb</code>	Predefined	<code>0.00</code> , <code>0.10</code> , ..., <code>20.00</code> (one per \$0.10 step)
<code>hb_bidder</code>	Predefined	<code>limelightDigital</code> , <code>none</code>
<code>hb_uuid</code>	Free-form	(dynamic UUIDs)
<code>hb_size</code>	Predefined	<code>640x360</code> , <code>1280x720</code> , <code>1920x1080</code>
<code>hb_format</code>	Predefined	<code>video</code>
<code>hb_adid</code>	Free-form	(dynamic)
<code>hb_source</code>	Predefined	<code>client</code> , <code>s2s</code>

2. VAST wrapper creative

GAM → **Delivery** → **Creatives** → **New** → **Video** → **VAST 3.0 ad tag**. Paste:

```
https://prebid.adnxs.com/pbc/v1/cache?uuid=%%PATTERN:hb_uuid%
```

Duration: **0:00** (wrapper resolves the real duration). Substitute your own cache URL if not using the AppNexus public cache.

3. Price-bucket line items

One line item per \$0.10 bucket (200 total for \$0-\$20). Use Prebid's **Line Item Generator** (docs.prebid.org/tools/line-item-generator.html) to bulk-create via CSV.

FIELD	VALUE
Name	Prebid HB \$2.50
Type	Price Priority
CPM	Match the bucket exactly
Targeting → key-values	hb_pb is one of 2.50
Creative	The VAST wrapper from step 2

EXACT STRING MATCH

Prebid emits "2.50" (two decimals). A line item targeting "2.5" will never serve.

4. House fallback + order

Add a low-priority line item targeting hb_pb=0.00 with your house creative or AdSense passback. Wrap all line items in one Order ("Prebid Header Bidding") for one-click pause/activate.

06 — Limelight onboarding

1. Apply for a Limelight account via your CSM.
2. Receive credentials: `publisherId` → `data-pub-id` ; `adUnitId` → `data-adunit-id` ; `host` → `data-host` .
3. Submit publisher domain(s) for allowlist approval (1-3 business days).
4. Verify bidding using section 08.

PRE-ALLOWLIST

You can test integration mechanics before allowlisting completes — the bidder returns no-bid and GAM's fallback fills.

07 — Consent management

Defaults are tuned for non-EU traffic. EU and US-state-privacy publishers override.

FRAMEWORK	DEFAULT
GDPR (TCF v2)	Static <code>gdprApplies: false</code> — non-EU-safe.
USP (CCPA)	Auto-detects <code>__uspapi</code> (100 ms timeout).
GPP	Off — module cancels auction without a GPP CMP.

EU — enable TCF v2

Place a registered CMP (OneTrust, Sourcepoint, Quantcast, Didomi...). Add this script *before* the engine tag:

```
<script>
window.pbjs = window.pbjs || { que: [] };
pbjs.que.push(function() {
  pbjs.setConfig({
    consentManagement: {
      gdpr: { cmpApi: 'iab', timeout: 8000, defaultGdprScope: true }
    }
  });
});
</script>
```

US state privacy — enable GPP

```
pbjs.setConfig({ consentManagement: { gpp: { cmpApi: 'iab', timeout: 3000 } } });
```

COMPLIANCE ≠ ENGINE CONFIG

The engine ships the technical hooks. Legal compliance depends on your CMP, privacy policy, and DPAs. Consult counsel.

08 — Testing & verification

Append `?debug=true` to your URL.

Console (filter for **STEP**)

- **STEP 1** Auction Initialized · Target: limelightDigital · Timeout: 1200ms
- **STEP 1.5** Requesting from Limelight · Player size: W×H
- **WINNER** <bidder> · Raw CPM, or **NO MARKET DEMAND**
- **STEP 6** Dispatching VAST Request

Network

REQUEST	STATUS
<code>cdn.jsdelivr.net/.../player.js</code> + <code>.../prebid.js</code>	200
<code>imasdk.googleapis.com/.../ima3.js</code>	200
POST <code>{host}/ortbh</code> — inspect Payload	200
<code>prebid.adnxs.com/pbc/v1/cache</code> (winning bid only)	200
GET <code>pubads.g.doubleclick.net/gampad/ads</code> — decode <code>cust_params</code> , must contain <code>hb_pb</code> , <code>hb_bidder</code> , <code>hb_uuid</code>	200

Visual

- Pre-roll plays before content; Video.js controls (play / pause / mute / volume / progress / fullscreen) respond.

09 — Troubleshooting

No `POST /ortbhb` request in Network

- `pbjs.installedModules` must include `LimelightDigitalBidAdapter`. If missing, fix `data-prebid-url`.
- Look for "Canceling auction as per consentManagement config" — see section 07.

`200` response but empty `seatbid`

- **Domain not allowlisted** by Limelight — the most common cause.
- Player rendered < 300×200 — bidders may reject small inventory.

Bid wins but no ad plays / GAM falls through

- `cust_params` must contain `hb_uuid`. Missing → engine < v1.2.0; upgrade.
- Line item targeting must match Prebid's string format exactly: `"2.50"` not `"2.5"`.
- VAST wrapper must use `%%PATTERN:hb_uuid%%`.

Request timeout after 1200ms

Increase `data-timeout` to `1500` – `2000` ms, or ask Limelight for a regional endpoint.

Autoplay doesn't start

Confirm `data-autoplay="true"` AND `data-muted="true"`. HTTPS required. iOS Safari has stricter rules with no engine-level workaround.

Mixed-content warnings

HTTPS page + `http://` resource (content / GAM / creative). Force everything to HTTPS.

10 — FAQ

Multiple players per page?

Not currently. Workaround: iframe per player.

Other Prebid bidders / multiple SSPs?

Yes — v2 supports multi-bidder via `data-bidders`. Ask your account manager which SSPs you need and they will generate a Prebid bundle containing those adapters.

If Prebid fails to load?

Engine catches it and calls GAM directly with `hb_pb=0.00&hb_bidder=none`. Your house line item fills.

Latency impact?

`async` script — zero impact on Time to Interactive. Adds the bidder timeout (default 1200 ms) before pre-roll start.

`@v2.5.0` vs `@main` ?

Tags are immutable (1-year CDN cache). `@main` = latest commit (~12 h cache). Pin tags for prod.

Auction analytics?

Impression-level reporting via GAM filtered by `hb_bidder`. Auction-level (bid latency / rate) requires a Prebid build with an analytics adapter — ask your account manager.

Bidding Player • v2.5.0 • Built on Prebid.js, video.js, Google IMA SDK

Engine: `cdn.jsdelivr.net/gh/aryanvani-projects/bidding-player@v2.5.0/engine/player.js`