



Enabling myGaru ID in Prebid allows publishers to enhance the value of their ad inventory with minimal effort. By transmitting a myGaru ID via Prebid, Publisher enables more accurate audience targeting and increase the likelihood of participating in high-value auctions. This results in higher CPMs and improved monetization, without the need for any code changes. Activating this feature is a simple yet effective step toward maximizing advertising revenue.

Preconditions:

- [Prebid.js](#) (version 9.x or higher) is already running on your website; auctions served to GAM.
- CMP (TCF v2.2) is present on your website.
- You can edit the page code (add two code blocks).

1. Add myGaru User ID module to your Prebid build:

1. Go to prebid.org → Download page ([link](#))
2. Scroll to the **User ID Modules** section and tick **MyGaru ID** (+ your existing bidder adapters).
3. Click **"Get Prebid JS"** to download configured `prebid.js` with myGaru ID module and host it on your CDN (or use your internal build with `myGaruid` included).

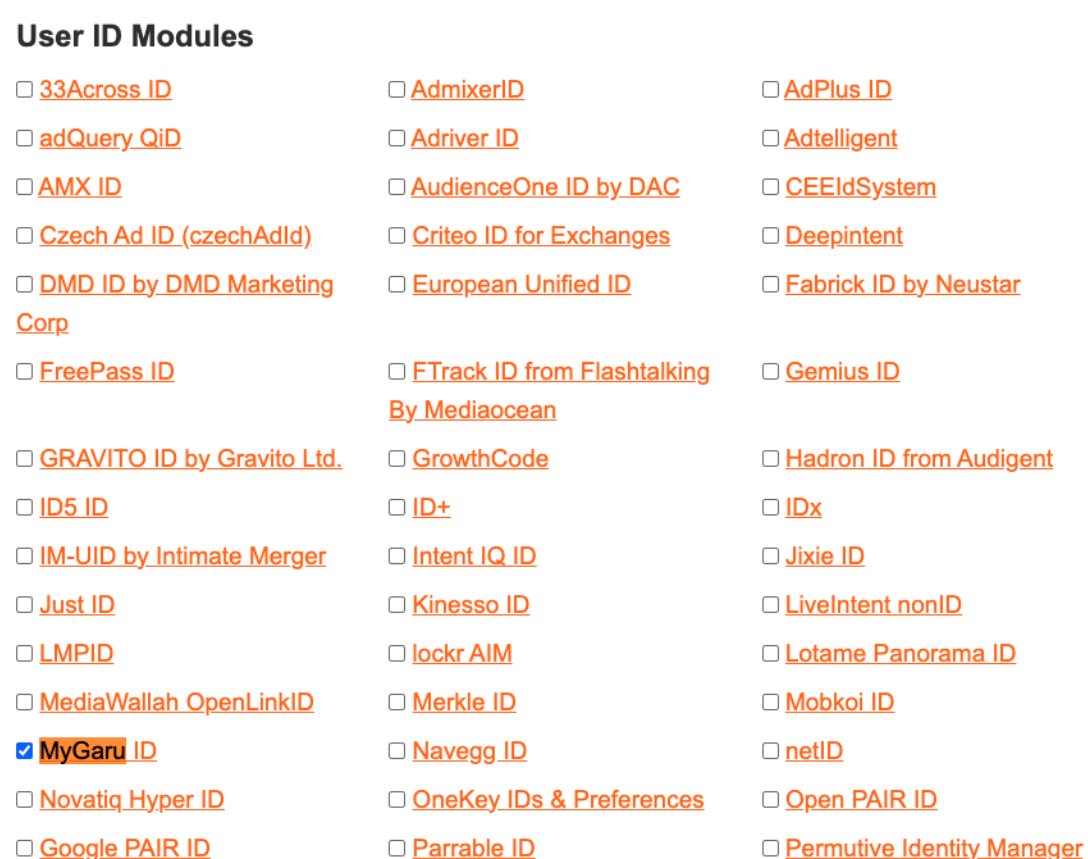


Figure 1. myGaru ID within Prebid User ID Modules

2. Deploy updated Prebid.js code on the page:

Deploy updated [Prebid.js](#) code onto your website

Required order of scripts:

- CMP (TCF v2.2)
- **Prebid.js** (build with myGaruid User ID Module)
- Prebid config (`pbjs.setConfig`) incl. `encryptedSignalSources`
- `pbjs.registerSignalSources()`
- GPT (Google Publisher Tags) / GAM calls
- `pbjs.requestBids(...)` -> `setTargeting` -> `googletag.display(...)`

Code snippet

```
// window.pbjs.setConfig(/* your prebid config */)
pbjs.setConfig({

  userSync: {
    encryptedSignalSources: {
      "sources": [{
        source: ['mygaruID'],
        encrypt: false
      }]
    },
    userIds: [{
      name: 'mygaruId',
      storage: {
        type: 'html5',
        name: 'mygId',
        expires: 1, // storage lasts for 1 days
        refreshInSeconds: 11*3600 // refresh ID every 11 hours to ensure i
t's fresh
      }
    }],
    auctionDelay: 400 // 400ms maximum auction delay, applies to a
ll userId modules
  }
});
```

3. Verify integration in Browser DevTools:

- Console → `pbjs.getUserIds()` - must include `myGaruid`.
- Console → `pbjs.getUserIdsAsEids()` - must include our `source`.
- Network → bidder requests payload contains `user.eids`.
- No GDPR/HTTPS/CORS errors in Console.

4. Troubleshooting:

- myGaruid is empty in `getUserIds()` → check if Prebid build contains myGaruid Module enabled
- No EIDs in requests → check Prebid version (must be 9.x or later)
- `registerSignalSources` not working → call again after `window.load` or SDK callback, don't limit `registerDelay` too much