

# OpenWrap Hooks for JS

OpenWrap hooks for JS allow you to customize OpenWrap behavior. This page outlines these hooks:



See [Publisher API Reference](#) for more a list of prebi.js object.

Uncaught exceptions from code running on hooks should be caught not propagated. When running in debug mode, exceptions must be logged to the console

## Prebid set config

The `PWT.HookForPrebidSetConfig` hook can be used to change configs such as Prebid user syncs. It can also be used to pass external UIDs.



For more information, see [Prebid user syncing](#).

## Description

- The Prebid Config object: `prebidConfig`.
- This hook fetches a universal ID by invoking a third-party script and then setting that ID as a Prebid config. The script may need to make a network call, so the callout to the publisher code must be able to support asynchronous return.
- This hook is executed before calling `owpbjs.setConfig` method, the same `prebidConfig` object is passed to `owpbjs.setConfig` method.
- The `prebidConfig` object can be modified in the hook code.
- This hook fires after the Prebid object (`pbjs`) has initiated.

### Prebid set config sampe

```
<html>
<head>
  <!--GPT Tag with Callback begins here -->
  <script type="text/javascript">
    var PWT = {}; //Initialize Namespace
    var googletag = googletag || {};
    googletag.cmd = googletag.cmd || [];
    PWT.jsLoaded = function() { // Wrapper tag on load callback is used to load GPT
      (function() {
        var gads = document.createElement('script');
        var useSSL = 'https:' == document.location.protocol;
        gads.src = (useSSL ? 'https:' : 'http:') + '//www.googletagservices.com/tag/js/gpt.js';
        var node = document.getElementsByTagName('script')[0];
        node.parentNode.insertBefore(gads, node);
      })();
    };

    PWT.HookForPrebidSetConfig = function(config){
      console.log('HookForPrebidSetConfig: Here is the Prebid config:', config);
      console.log('HookForPrebidSetConfig: ORIGINAL: pbConf.currency.adServerCurrency: ' , pbConf.
currency.adServerCurrency);
      config.currency.adServerCurrency = "USD";
      setTimeout(function(){
        var pbConf = owpbjs.getConfig();
        console.log('HookForPrebidSetConfig: UPDATED: pbConf.currency.adServerCurrency: ' , pbConf.
currency.adServerCurrency);
      }, 2000);
    };

  </script>
  <!--GPT Tag ends here -->
  <!--Wrapper Tag starts here -->
  <script type="text/javascript">
    (function() {
      var wtads = document.createElement('script');
      wtads.async = true;
      wtads.type = 'text/javascript';
      wtads.src = 'owt.min.js';
      var node = document.getElementsByTagName('script')[0];
```

```

        node.parentNode.insertBefore(wtads, node);
    })();
</script>
<!--Wrapper Tag ends here -->

<script>
    googletag.cmd.push(function () {
        googletag.defineSlot('/19968336/header-bid-tag-0', [[300, 250], [300, 600]], 'div-gpt-ad-1460505748561-0').addService(googletag.pubads());
        googletag.defineSlot('/19968336/header-bid-tag1', [[728, 90], [970, 90]], 'div-gpt-ad-1460505661639-0').addService(googletag.pubads());
        googletag.pubads().enableSingleRequest();
        googletag.enableServices();
    });
</script>
</head>

<body>
<h2>Prebid.js Test</h2>
<h5>Div-1</h5>
<div id='div-gpt-ad-1460505748561-0'>
    <script type='text/javascript'>
        googletag.cmd.push(function() { googletag.display('div-gpt-ad-1460505748561-0'); });
    </script>
</div>

<h5>Div-2</h5>
<div id='div-gpt-ad-1460505661639-0'>
    <script type='text/javascript'>
        googletag.cmd.push(function() { googletag.display('div-gpt-ad-1460505661639-0'); });
    </script>
</div>

</body>
</html>

```

## Request bids

The `PWT.HookForPrebidRequestBids` hook enables you to make changes to the `Prebid adUnitArray` before an auction is initiated. Use cases include setting bidder-specific floors and adding page-specific bidder parameters such as site section. This hook enables you to have one `OpenWrap` unit with different floors and PMP inclusion based on a site section.

## Description

- The `Prebid adUnitsArray`: `adUnitsArray`.
- This hook is executed before calling `owpbjs.requestBids`.
- The `adUnitsArray` object can be modified in the hook code.
- This hook can be used to pass additional params to bidders.

```

<html>
<head>
<!--GPT Tag with Callback begins here -->
<script type="text/javascript">
    var PWT = {}; //Initialize Namespace
    var googletag = googletag || {};
    googletag.cmd = googletag.cmd || [];
    PWT.jsLoaded = function() { // Wrapper tag on load callback is used to load GPT
        (function() {
            var gads = document.createElement('script');
            var useSSL = 'https:' == document.location.protocol;
            gads.src = (useSSL ? 'https:' : 'http:') + '//www.googletagservices.com/tag/js/gpt.js';
            var node = document.getElementsByTagName('script')[0];
            node.parentNode.insertBefore(gads, node);
        })();
    };

    PWT.HookForPrebidRequestBids = function(adUnits){
        /*
        post this change you can see imp[].bidfloor is set to 1.6 in PubMatic translator request

```

```

        */
        console.log('HookForPrebidRequestBids: Original adUnits array :', adUnits);
        console.log('HookForPrebidRequestBids: adding kadfloor to pubmatic');
        adUnits.forEach(function(au){
            au.bids.forEach(function(bid){
                if(bid.bidder == 'pubmatic'){
                    bid.params.kadfloor = '1.6';
                }
            })
        })
    });
</script>
<!--GPT Tag ends here -->
<!--Wrapper Tag starts here -->
<script type="text/javascript">
    (function() {
        var wtads = document.createElement('script');
        wtads.async = true;
        wtads.type = 'text/javascript';
        wtads.src = 'owt.min.js';
        var node = document.getElementsByTagName('script')[0];
        node.parentNode.insertBefore(wtads, node);
    })();
</script>
<!--Wrapper Tag ends here -->

<script>
    googletag.cmd.push(function () {
        googletag.defineSlot('/19968336/header-bid-tag-0', [[300, 250], [300, 600]], 'div-gpt-ad-1460505748561-0').addService(googletag.pubads());
        googletag.defineSlot('/19968336/header-bid-tag1', [[728, 90], [970, 90]], 'div-gpt-ad-1460505661639-0').addService(googletag.pubads());
        googletag.pubads().enableSingleRequest();
        googletag.enableServices();
    });
</script>
</head>

<body>
<h2>Prebid.js Test</h2>
<h5>Div-1</h5>
<div id='div-gpt-ad-1460505748561-0'>
    <script type='text/javascript'>
        googletag.cmd.push(function() { googletag.display('div-gpt-ad-1460505748561-0'); });
    </script>
</div>

<h5>Div-2</h5>
<div id='div-gpt-ad-1460505661639-0'>
    <script type='text/javascript'>
        googletag.cmd.push(function() { googletag.display('div-gpt-ad-1460505661639-0'); });
    </script>
</div>

</body>
</html>

```

## Bids received

The PWT.HookForBidReceived enables modification of bids before they are processed. Use cases include changing a bid from gross revenue to net revenue.



if you change the eCPM in the bid object, the modified eCPM will be displayed in our reporting.

## Description

- This hook is called every time a bid is received from a bidder for an ad slot.

- The bid object can be modified to change eCPM or creative.
- This hook be used to boost bids on some conditions.

```

<html>
<head>
  <!--GPT Tag with Callback begins here -->
  <script type="text/javascript">
    var PWT = {}; //Initialize Namespace
    var googletag = googletag || {};
    googletag.cmd = googletag.cmd || [];
    PWT.jsLoaded = function() { // Wrapper tag on load callback is used to load GPT
      (function() {
        var gads = document.createElement('script');
        var useSSL = 'https:' == document.location.protocol;
        gads.src = (useSSL ? 'https:' : 'http:') + '//www.googletagservices.com/tag/js/gpt.js';
        var node = document.getElementsByTagName('script')[0];
        node.parentNode.insertBefore(gads, node);
      })();
    };

    PWT.HookForBidReceived = function(divId, adapterId, bid, latency){
      /*
        Post this change you can see the bid gross ecpm is modified in logger record
        I think here we can avoid calls for default bids
      */
      console.log('HookForBidReceived: ORIGINAL: bid: ' , arguments);
      bid.defaultBid = 0;
      bid.setGrossEcpm(15);
    };

  </script>
  <!--GPT Tag ends here -->
  <!--Wrapper Tag starts here -->
  <script type="text/javascript">
    (function() {
      var wtads = document.createElement('script');
      wtads.async = true;
      wtads.type = 'text/javascript';
      wtads.src = 'owt.min.js';
      var node = document.getElementsByTagName('script')[0];
      node.parentNode.insertBefore(wtads, node);
    })();
  </script>
  <!--Wrapper Tag ends here -->

  <script>
    googletag.cmd.push(function () {
      googletag.defineSlot('/19968336/header-bid-tag-0', [[300, 250], [300, 600]], 'div-gpt-ad-1460505748561-0').addService(googletag.pubads());
      googletag.defineSlot('/19968336/header-bid-tag1', [[728, 90], [970, 90]], 'div-gpt-ad-1460505661639-0').addService(googletag.pubads());
      googletag.pubads().enableSingleRequest();
      googletag.enableServices();
    });
  </script>
</head>

<body>
<h2>Prebid.js Test</h2>
<h5>Div-1</h5>
<div id='div-gpt-ad-1460505748561-0'>
  <script type='text/javascript'>
    googletag.cmd.push(function() { googletag.display('div-gpt-ad-1460505748561-0'); });
  </script>
</div>

<h5>Div-2</h5>
<div id='div-gpt-ad-1460505661639-0'>
  <script type='text/javascript'>
    googletag.cmd.push(function() { googletag.display('div-gpt-ad-1460505661639-0'); });
  </script>

```

```
</div>

</body>
</html>
```

## Post-auction key values

`PWT.HookForPostAuctionKeyValues` enables modification of ad server targeting keys for an ad slot. It can be used to add, remove or modify targeting keys. Uses cases include adding targeting keys for PMP targeting.



### Custom integrations

`HookForPostAuctionKeyValues` is not available for custom integrations. For custom integrations, all the winning bids and respective key-values are sent to the publisher code in a call-back function, which can be modified by editing `adUnit.bidData.kvp`.



This hook is only called when we have at least one non-zero eCPM bid on a slot.

## Description

- This hook is executed for an ad slot post-auction of bids from bidders.
- This hook is only available for GAM-GPT controller, since the custom controller already returns control to the publisher code post-auction.
- If you add or change the key-value pairs, the modified key-value pairs will be added to the respective GAM slots, but the eCPM that was changed in the key-value pairs will not be updated in our reporting.
- This hook can be used to add extra key-value pairs.

### Key values sample

```
<html>
<head>
  <!--GPT Tag with Callback begins here -->
  <script type="text/javascript">
    var PWT = {}; //Initialize Namespace
    var googletag = googletag || {};
    googletag.cmd = googletag.cmd || [];
    PWT.jsLoaded = function() { // Wrapper tag on load callback is used to load GPT
      (function() {
        var gads = document.createElement('script');
        var useSSL = 'https:' == document.location.protocol;
        gads.src = (useSSL ? 'https:' : 'http:') + '//www.googletagservices.com/tag/js/gpt.js';
        var node = document.getElementsByTagName('script')[0];
        node.parentNode.insertBefore(gads, node);
      })();
    };

    PWT.HookForBidReceived = function(divId, adapterId, bid, latency){
      /*
       Post this change you can see the bid gross ecpm is modified in logger record
       I think here we can avoid calls for default bids
      */
      console.log('HookForBidReceived: ORIGINAL: bid: ', arguments);
      bid.defaultBid = 0;
      bid.setGrossEcpm(1);
    };

    PWT.HookForPostAuctionKeyValues = function(keyValuePairs, googleAdSlot){
      /*
       Note: on this page we have intentionally added a hook for HookForBidReceived so that we will
       always get a non-zero ecpm bid
       Post this change we can see that extra key value pair user-name: Harshad is added on Google ad-
       slot, same can be verified in call to securepubads.g.doubleclick.net/
      */
      console.log('HookForPostAuctionKeyValues: ORIGINAL:' , arguments);
```

```

        keyValuePairs['user-name'] = 'Harshad';
    };

</script>
<!--GPT Tag ends here -->
<!--Wrapper Tag starts here -->
<script type="text/javascript">
    (function() {
        var wtads = document.createElement('script');
        wtads.async = true;
        wtads.type = 'text/javascript';
        wtads.src = 'owt.min.js';
        var node = document.getElementsByTagName('script')[0];
        node.parentNode.insertBefore(wtads, node);
    })();
</script>
<!--Wrapper Tag ends here -->

<script>
    googletag.cmd.push(function () {
        googletag.defineSlot('/19968336/header-bid-tag-0', [[300, 250], [300, 600]], 'div-gpt-ad-1460505748561-0').addService(googletag.pubads());
        googletag.defineSlot('/19968336/header-bid-tag1', [[728, 90], [970, 90]], 'div-gpt-ad-1460505661639-0').addService(googletag.pubads());
        googletag.pubads().enableSingleRequest();
        googletag.enableServices();
    });
</script>
</head>

<body>
<h2>Prebid.js Test</h2>
<h5>Div-1</h5>
<div id='div-gpt-ad-1460505748561-0'>
    <script type='text/javascript'>
        googletag.cmd.push(function() { googletag.display('div-gpt-ad-1460505748561-0'); });
    </script>
</div>

<h5>Div-2</h5>
<div id='div-gpt-ad-1460505661639-0'>
    <script type='text/javascript'>
        googletag.cmd.push(function() { googletag.display('div-gpt-ad-1460505661639-0'); });
    </script>
</div>

</body>
</html>

```