

OpenWrap JavaScript API

OpenWrap JavaScript API is a straightforward API you can use to integrate [OpenWrap](#) with any ad server. The JavaScript API is the recommended integration method when using multiple wrapper solutions like A9. OpenWrap JavaScript does not take control of Google Ad Manager/[Google Publisher Tag](#) flow, it lets publishers call OpenWrap APIs to initiate bidder calls.

Get Started

To use the OpenWrap JavaScript API, set **Platform** to *Display* and **AdServer** to *Custom* in your [OpenWrap profile](#) .



This guide assumes a working knowledge of OpenWrap

Your PubMatic account must meet [OpenWrap's Publisher Requirements](#). Learn more about setting up and using OpenWrap in the, [OpenWrap User Guide](#).

API Methods

The OpenWrap JavaScript API contains one main method that you can use on any ad server, and three convenience methods you can use on Google Ad Manager and [GPT](#). The following detail sections include parameter and response examples. See the [Example Pages](#) section for a variety of on-page implementation scenarios.

PWT.requestBids

Use on any ad server to call bidders with your OpenWrap configurations for the provided slots. Performs an auction for each slot in response to bidders. Generates key-value pairs for each slot/auction. Adds winning bid results and key-value-pairs as a `bidData` object to the slot configuration objects. Returns the modified config objects containing `bidData` to the callback function specified in the second parameter.

Parameters

1. Array of OpenWrap configuration objects...

```
[{
  code: "some-pub-friendly-unique-name-1", // mandatory
  divId: "pubmatic_test2", // mandatory
  adUnitId: "ad_unit-id_1", // mandatory: can be any string, GAM ad unit id is often used
  adUnitIndex: "0", // mandatory
  mediaTypes: { // mandatory
    banner: { // mandatory
      sizes: [
        [300, 250]
      ] // mandatory: array of sizes
    }
  },
  {
    code: "some-pub-friendly-unique-name-2", // mandatory
    divId: "pubmatic_test-3", // mandatory
    adUnitId: "ad_unit-id_2", // mandatory: can be any string, GAM ad unit id is often used
    adUnitIndex: "0", // mandatory
    mediaTypes: { // mandatory
      banner: { // mandatory
        sizes: [
          [300, 250]
        ] // mandatory: array of sizes
      }
    }
  }
}]
```

2. Callback function...

```
function(resp){
    console.log("response:", resp);
    for(var i = 0, l = resp.length; i<l; i++){
        var r = resp[i];
        console.log("");
        console.log("code: ", r.code);
        if(r.bidData){
            console.log("key value pairs", (r.bidData.kvp || ""));
            console.log("Bid details", (r.bidData.wb || {}));
            // Publisher to add keys on ad Slot
        }
    }
}
```

Response

Modified OpenWrap configuration objects containing the `bidData` object.

bidData...

```
bidData : {
  // winning bid data information
  wb: {
    "adHtml": "<CREATIVE CODE>"
    "width": 300,
    "height": 250,
    "netEcpm": "1.0",
    "grossEcpm": "1.0",
    "adapterID": "pubmatic"
  },
  // key value pairs generated by OpenWrap that publisher should add to respective adserver slot
  kvp: {
    "key1": "value1",
    "pwtSid": "12fjg89",
    pwtSz: "300x250"
  }
}
```

Convenience Methods

Use the three convenience methods below to streamline integration with Google Ad Manager and Google Publisher Tag.

For other ad servers...

Create your own methods that provide the following functionality:

- Generate an array of OpenWrap configurations.
- Add OpenWrap key-value pairs to GPT ad slots.
- Remove key-value pairs from ad slots.

Convenience Method	Description
OpenWrap JavaScript API#PWT.generateConfForGPT	Generates the configuration array used by <code>PWT.requestBids</code> .
OpenWrap JavaScript API#PWT.addKeyValuePairsToGPTSlots	Adds OpenWrap key-value pairs to GPT ad slots.
OpenWrap JavaScript API#PWT.removeKeyValuePairsFromGPTSlots	Removes key-value pairs added to ad slots by OpenWrap.

PWT.generateConfForGPT

Generates the OpenWrap configuration array required for `PWT.requestBids`.



Works only for Google Ad Manager/GPT.

Parameters

Array of GPT ad slot objects.

Response

Array of OpenWrap configuration objects required for the first parameter of `PWT.requestBids`.

PWT.addKeyValuePairsToGPTSlots

Adds OpenWrap key-value pairs on respective GPT ad slots.



Works only for Google Ad Manager/GPT.

Parameters

Array of configuration objects returned to the callback function specified in the second parameter of `PWT.requestBids`.

Response

None

PWT.removeKeyValuePairsFromGPTSlots

Removes the key-value pairs added by OpenWrap on the given GPT ad slots.



Works only for Google Ad Manager/GPT.

Parameters

Array of GPT ad slot objects.

Response

None

Example Pages

The following examples demonstrate using the OpenWrap JavaScript API in various on-page scenarios such as refreshing ad slots, infinitely scrolling pages, A9, and so on.

Simple Setup

A simple example of integrating OpenWrap on-page with a single call to `PWT.requestBids`.

Simple Example

```
<!DOCTYPE html>
<html>
<head>
  <title>OW Custom</title>
</head>
<body>

  <script type="text/javascript" src="owt.min.js"></script>
  <script type="text/javascript">
    PWT.requestBids(
      [{
        code: "some-pub-friendly-unique-name-1", // mandatory
        divId: "pubmatic_test2", // mandatory
        adUnitId: "ad_unit-id1", // mandatory
        adUnitIndex: "", // necessary in case of PubMatic, can be derived by our code by simply incrementing
used adUnitIds (NOT in scope now)
        mediaTypes: { // mandatory
          banner: { // mandatory in first phase? or atleast one type of mediaTypes should be present
            sizes: [
              [300, 250]
            ] // array of sizes
          }
        }
      }, {
        code: "some-pub-friendly-unique-name-2", // mandatory
        divId: "pubmatic_test-3", // mandatory
        adUnitId: "ad_unit-id2", // mandatory
        adUnitIndex: "", // necessary in case of PubMatic, can be derived by our code by simply incrementing
used adUnitIds (NOT in scope now)
        mediaTypes: { // mandatory
          banner: { // mandatory in first phase? or atleast one type of mediaTypes should be present
            sizes: [
              [300, 250]
            ] // array of sizes
          }
        }
      }
    ],
    function (resp) {
      console.log("response:", resp);
      for (var i = 0, l = resp.length; i < l; i++) {
        var r = resp[i];
        console.log("");
        console.log("code: ", r.code);
        if (r.bidData) {
          console.log("key value pairs", (r.bidData.kvp || {}));
          console.log("Bid details", (r.bidData.wb || {}));
        }
      }
    }
  );

  </script>

</body>
</html>
```

Typical Design Pattern

The following example illustrates a typical structure for integrating OpenWrap JavaScript APIs in your pages. All examples following this one are built upon the same pattern.

Typical Design Example...

```
<!DOCTYPE HTML>
```

```

<html lang="en-us">

<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <title>NON SRA</title>
  <script type="text/javascript">
    var PWT = {};
    var googletag = googletag || {};
    googletag.cmd = googletag.cmd || [];
    var gptRan = false;
    PWT.jsLoaded = function() {
      loadGpt();
    };
    (function() {
      var purl = window.location.href;
      var url = '//ads.pubmatic.com/AdServer/js/pwt/5890/651';
      var profileVersionId = '';
      if (purl.indexOf('pwtv=') > 0) {
        var regexp = /pwtv=(.*?)(&|$)/g;
        var matches = regexp.exec(purl);
        if (matches.length >= 2 && matches[1].length > 0) {
          profileVersionId = '/' + matches[1];
        }
      }
      var wtads = document.createElement('script');
      wtads.async = true;
      wtads.type = 'text/javascript';
      wtads.src = url + profileVersionId + '/pwt.js';
      var node = document.getElementsByTagName('script')[0];
      node.parentNode.insertBefore(wtads, node);
    })();
    var loadGpt = function() {
      // Check the gptRan flag
      if (!gptRan) {
        gptRan = true;
        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);
      }
    };
    // Failsafe to call gpt
    setTimeout(loadGpt, 500);
  </script>

  <script type="text/javascript">

    googletag.cmd.push(function() {

      function initAdserver() {
        if (PWT.initAdserverSet) return;
        PWT.initAdserverSet = true;
        googletag.display('pubmatic_test2');
        googletag.display('Div2');
      }

      var s1 = googletag.defineSlot('/43743431/DMDemo', [
        [728, 90],
        [300, 250],
        [160, 600]
      ], 'pubmatic_test2').addService(googletag.pubads());
      s1.setTargeting('a', [1]);
      s1.setTargeting('b', [2]);
      s1.setTargeting('c', 3);
      s1.setTargeting('d', ['a', 'b', 'c']);

      var s2 = googletag.defineSlot('/43743431/DMDemo1', [
        [300, 250],
        [728, 90],
        [160, 600]
      ], 'pubmatic_test2').addService(googletag.pubads());
    });
  </script>

```

```

    ], 'Div2').addService(googletag.pubads());
    s2.setTargeting('aa', [11]);
    s2.setTargeting('bb', [22]);
    s2.setTargeting('dd', ['aa', 'bb', 'cc']);

    // common targetings
    googletag.pubads().setTargeting('v', 'classifieds');
    googletag.pubads().setTargeting('cpage', 'browseads');
    googletag.pubads().setTargeting('article-id', '65207');
    googletag.pubads().setTargeting('automation', '1');

    googletag.enableServices();

    if(typeof PWT.requestBids === 'function'){
        PWT.requestBids(
            PWT.generateConfForGPT(googletag.pubads().getSlots()),
            function(adUnitsArray) {
                PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
                initAdserver();
            }
        );
    }else{
        initAdserver();
    }

    var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap timeout
    setTimeout(function() {
        initAdserver();
    }, FAILSAFE_TIMEOUT);
    });
</script>
</head>

<body style='background-color:whitesmoke;'>
    <h2 style='text-align:center;'>
        NON SRA
    </h2>
    <br><br>
    <!-- DMDemo -->
    Adslot : /43743431/DMDemo
    <br>
    <div id='pubmatic_test2'>
    </div>
    <br><br>
    <!-- DMDemo1 -->
    Adslot : /43743431/DMDemo1
    <br>
    <div id='Div2'>
    </div>
</body>

</html>

```

Refreshing Slots

Similar to the previous example, but a snippet at the end refreshes all as slots after 15 seconds.

Refreshing Slots Example...

```

<!DOCTYPE HTML>
<html lang="en-us">

<head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <title>NON SRA</title>

    <script type="text/javascript">
        var PWT = {};

```

```

var googletag = googletag || {};
googletag.cmd = googletag.cmd || [];
var gptRan = false;
PWT.jsLoaded = function() {
    loadGpt();
};
(function() {
    var purl = window.location.href;
    var url = '//ads.pubmatic.com/AdServer/js/pwt/5890/651';
    var profileVersionId = '';
    if (purl.indexOf('pwtv=') > 0) {
        var regexp = /pwtv=(.*?)(&|$)/g;
        var matches = regexp.exec(purl);
        if (matches.length >= 2 && matches[1].length > 0) {
            profileVersionId = '/' + matches[1];
        }
    }
    var wtads = document.createElement('script');
    wtads.async = true;
    wtads.type = 'text/javascript';
    wtads.src = url + profileVersionId + '/pwt.js';
    var node = document.getElementsByTagName('script')[0];
    node.parentNode.insertBefore(wtads, node);
})();
var loadGpt = function() {
    // Check the gptRan flag
    if (!gptRan) {
        gptRan = true;
        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);
    }
}
// Failsafe to call gpt
setTimeout(loadGpt, 500);
</script>

<script type="text/javascript">
    googletag.cmd.push(function() {

        function initAdserver() {
            if (PWT.initAdserverSet) return;
            PWT.initAdserverSet = true;
            googletag.display('pubmatic_test2');
            googletag.display('Div2');
        }

        var s1 = googletag.defineSlot('/43743431/DMDemo', [
            [728, 90],
            [300, 250],
            [160, 600]
        ], 'pubmatic_test2').addService(googletag.pubads());
        s1.setTargeting('a', [1]);
        s1.setTargeting('b', [2]);
        s1.setTargeting('c', 3);
        s1.setTargeting('d', ['a', 'b', 'c']);

        var s2 = googletag.defineSlot('/43743431/DMDemo1', [
            [300, 250],
            [728, 90],
            [160, 600]
        ], 'Div2').addService(googletag.pubads());
        s2.setTargeting('aa', [11]);
        s2.setTargeting('bb', [22]);
        s2.setTargeting('dd', ['aa', 'bb', 'cc']);

        // common targetings
        googletag.pubads().setTargeting('v', 'classifieds');
        googletag.pubads().setTargeting('cpage', 'browseads');
    });

```

```

googletag.pubads().setTargeting('article-id', '65207');
googletag.pubads().setTargeting('automation', '1');

googletag.enableServices();

if(typeof PWT.requestBids === 'function'){
    PWT.requestBids(
        PWT.generateConfForGPT(googletag.pubads().getSlots()),
        function(adUnitsArray) {
            PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
            initAdserver();
        }
    );
}else{
    initAdserver();
}

var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap timeout
setTimeout(function() {
    initAdserver();
}, FAILSAFE_TIMEOUT);
});
</script>
</head>

<body style='background-color:whitesmoke;'>
    <h2 style='text-align:center;'>
        NON SRA
    </h2>
    <br><br>
    <!-- DMDemo -->
    Adslot : /43743431/DMDemo
    <br>
    <div id='pubmatic_test2'>
    </div>
    <br><br>
    <!-- DMDemo1 -->
    Adslot : /43743431/DMDemo1
    <br>
    <div id='Div2'>
    </div>

<script>
    // refresh all slots after 15 seconds

    setTimeout(function(){

        function refreshSlots(){
            if (PWT.initAdserverSet) return;
            PWT.initAdserverSet = true;
            googletag.pubads().refresh(); // calling refresh slots function of GPT
        }

        if(typeof PWT.removeKeyValuePairsFromGPTSlots === 'function'){
            PWT.initAdserverSet = false;
            // first remove already added OpenWrap targeting key value pairs from slots
            PWT.removeKeyValuePairsFromGPTSlots( googletag.pubads().getSlots() );
            // now call OpenWrap to get bids and add respective key value pairs
            PWT.requestBids(
                PWT.generateConfForGPT( googletag.pubads().getSlots() ),
                function(adUnitsArray){
                    PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
                    refreshSlots();
                }
            );
        }
        refreshSlots();
    }

    var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap timeout
    setTimeout(function() {

```



```

        refreshSlots();
    }, FAILSAFE_TIMEOUT);

    }, 15000);
</script>
</body>

</html>

```

Infinite Scroll

The following example demonstrates using OpenWrap JavaScript APIs on infinitely scrolling pages.

Infinite Scroll Example...

```

<html>
<head>
<title>Infinite Scroll Example</title>

<script type="text/javascript">
    var PWT = {};
    var googletag = googletag || {};
    googletag.cmd = googletag.cmd || [];
    var gptRan = false;
    PWT.jsLoaded = function() {
        loadGpt();
    };
    (function() {
        var purl = window.location.href;
        var url = '//ads.pubmatic.com/AdServer/js/pwt/5890/651';
        var profileVersionId = '';
        if (purl.indexOf('pwtv=') > 0) {
            var regexp = /pwtv=(.*?)(&|$)/g;
            var matches = regexp.exec(purl);
            if (matches.length >= 2 && matches[1].length > 0) {
                profileVersionId = '/' + matches[1];
            }
        }
        var wtads = document.createElement('script');
        wtads.async = true;
        wtads.type = 'text/javascript';
        wtads.src = url + profileVersionId + '/pwt.js';
        var node = document.getElementsByTagName('script')[0];
        node.parentNode.insertBefore(wtads, node);
    })();
    var loadGpt = function() {
        // Check the gptRan flag
        if (!gptRan) {
            gptRan = true;
            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);
        }
    };
    // Failsafe to call gpt
    setTimeout(loadGpt, 500);
</script>

<script>

var adslot0;

googletag.cmd.push(function() {

    // Declare any slots initially present on the page

```

```

adslot0 = googletag.defineSlot('/6355419/Travel', [728, 90], 'leaderboard').
    setTargeting("test","infinitemscroll").
    addService(googletag.pubads());

// Infinite scroll requires SRA
googletag.pubads().enableSingleRequest();

// Disable initial load, we will use refresh() to fetch ads.
// Calling this function means that display() calls just
// register the slot as ready, but do not fetch ads for it.
googletag.pubads().disableInitialLoad();

// Enable services
googletag.enableServices();
});

// Function to generate unique names for slots
var nextSlotId = 1;
function generateNextSlotName() {
    var id = nextSlotId++;
    return 'adslot' + id;
}

// Function to add content to page, mimics real infinite scroll
// but keeps it much simpler from a code perspective.
function moreContent() {

    // Generate next slot name
    var slotName = generateNextSlotName();

    // Create a div for the slot
    var slotDiv = document.createElement('div');
    slotDiv.id = slotName; // Id must be the same as slotName
    document.body.appendChild(slotDiv);

    // Create and add fake content 1
    var h1=document.createElement("H2")
    var text1=document.createTextNode("Dynamic Fake Content 1");
    h1.appendChild(text1);
    document.body.appendChild(h1);

    // Create and add fake content 2
    var h2=document.createElement("H2")
    var text2=document.createTextNode("Dynamic Fake Content 2");
    h2.appendChild(text2);
    document.body.appendChild(h2);

    // Define the slot itself, call display() to
    // register the div and refresh() to fetch ad.
    googletag.cmd.push(function() {
        var slot = googletag.defineSlot('/6355419/Travel', [728, 90], slotName).
            setTargeting("test","infinitemscroll").
            addService(googletag.pubads());
        // Display has to be called before

        // refresh and after the slot div is in the page.
        // googletag.display(slotName); // commented original code
        // googletag.pubads().refresh([slot]); // commented original code

        function initAdserver() {
            if (PWT.initAdserverSet) return;
            PWT.initAdserverSet = true;
            googletag.display(slotName); // moved original code here
            googletag.pubads().refresh([slot]); // moved original code here
        }

        // this is a new code for OpenWrap
        if(typeof PWT.requestBids === 'function'){
            PWT.initAdserverSet = false;
            PWT.requestBids(
                PWT.generateConfForGPT( [slot] ),

```

```

        function(adUnitsArray){
            PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
            initAdserver();
        }
    );
}else{
    initAdserver();
}

var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap timeout
setTimeout(function() {
    initAdserver();
}, FAILSAFE_TIMEOUT);
});
}
</script>

<style>
body > div {
    margin-bottom: 1em;
    border: solid 1px black;
    width: 728px
}
body > img {
    margin-bottom: 1em;
    display: block
}
body > button {
    position: fixed;
    bottom: 10px;
}
</style>
</head>

<body>
<h1>GPT Test Page - Infinite Scroll</h1>

<!-- First fake content -->
<text>
<h2> Initial Fake Content </h2>
</text>

<!-- First ad -->
<div id="leaderboard">
    <script>
        // Call display() to register the slot as ready
        // and refresh() to fetch an ad.
        googletag.cmd.push(function() {

            //googletag.display('leaderboard'); // commented this original code
            //googletag.pubads().refresh([adslot0]); // commented this original code

            function initAdserver() {
                if (PWT.initAdserverSet) return;
                PWT.initAdserverSet = true;
                googletag.display('leaderboard'); // moved original code here
                googletag.pubads().refresh([adslot0]); // moved original code here
            }

            // this is a new code for OpenWrap
            if(typeof PWT.requestBids === 'function'){
                PWT.requestBids(
                    PWT.generateConfForGPT( [adslot0] ),
                    function(adUnitsArray){
                        PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
                        initAdserver();
                    }
                );
            }else{
                initAdserver();
            }
        }
    </script>

```

```

        var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap timeout
        setTimeout(function() {
            initAdserver();
        }, FAILSAFE_TIMEOUT);
    });
</script>
</div>

<!-- More initial fake content -->
<text>
  <h2> Initial Fake Content 1 </h2>
  <h2> Initial Fake Content 2 </h2>
</text>

<!-- Button to load more content dynamically. -->
<button onclick="moreContent()">More Content</button>

</body>
</html>

```

Working With A9

In addition to demonstrating integration with A9, this example also uses GPT's `disableInitialLoad` API.

A9 Integration Example...

```

<!DOCTYPE HTML>
<html lang="en-us">
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <title>Wrapper Tag GPT Async Demo with A9</title>

  <h4>GPT implementation is ASYNC SRA <a href="https://support.google.com/dfp_premium/answer/1638622?hl=en">
(Refer this for types of GPT Tags)</a></h4>

<!-- OpwenWrap Script begins here -->
<script type="text/javascript">
  var PWT = {};
  var googletag = googletag || {};
  googletag.cmd = googletag.cmd || [];
  var gptRan = false;
  PWT.jsLoaded = function() {
    loadGpt();
  };
  (function() {
    var purl = window.location.href;
    var url = '//ads.pubmatic.com/AdServer/js/pwt/5890/4162';
    var profileVersionId = '';
    if (purl.indexOf('pwtv=') > 0) {
      var regexp = /pwtv=(.*?)(&|$)/g;
      var matches = regexp.exec(purl);
      if (matches.length >= 2 && matches[1].length > 0) {
        profileVersionId = '/' + matches[1];
      }
    }
    var wtads = document.createElement('script');
    wtads.async = true;
    wtads.type = 'text/javascript';
    wtads.src = url + profileVersionId + '/pwt.js';
    var node = document.getElementsByTagName('script')[0];
    node.parentNode.insertBefore(wtads, node);
  })();
  var loadGpt = function() {
    // Check the gptRan flag
    if (!gptRan) {
      gptRan = true;

```

```

        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);
    }
}
// Failsafe to call gpt
setTimeout(loadGpt, 500);
</script>
<!--Wrapper Script ends here -->

<script type="text/javascript">
    // this function will act as a lock and will call the GPT API
    function initAdserver(forced) {
        if((forced === true && window.initAdserverFlag !== true) || (PWT.a9_BidsReceived && PWT.
ow_BidsReceived)){
            window.initAdserverFlag = true;
            PWT.a9_BidsReceived = PWT.ow_BidsReceived = false;
            googletag.pubads().refresh();
        }
    }
</script>

<!-- A9 Script Start -->
<script>
    !function(a9,a,p,s,t,A,g){if(a[a9])return;function q(c,r){a[a9]._Q.push([c,r])}a[a9]={init:function(){q
("i",arguments)},fetchBids:function(){q("f",arguments)},setDisplayBids:function(){},_Q:[]};A=p.createElement(s);
A.async=!0;A.src=t;g=p.getElementsByTagName(s)[0];g.parentNode.insertBefore(A,g)}("apstag",window,document,"
script","//c.amazon-adsystem.com/aax2/apstag.js");
    apstag.init({
        pubID: 'xxxx',
        adServer: 'googletag'
    });
    apstag.fetchBids({
        slots: [{
            slotID: 'Div2',
            sizes: [[300, 250], [300, 600]]
        },
        {
            slotID: 'Div1',
            sizes: [[728 ,90]]
        }],
        timeout: 2e3 // Make Sure this timeout is less than or equal to OpenWrap TimeOut
    }, function(bids) {
        googletag.cmd.push(function(){
            apstag.setDisplayBids();
            PWT.a9_BidsReceived = true;
            initAdserver(false);
        });
    });
</script>
<!-- A9 Script End -->

<script type="text/javascript">
    googletag.cmd.push(function() {
        var s1 = googletag.defineSlot('/43743431/DMDemo', [
            [728, 90]
        ], 'Div1').addService(googletag.pubads());
        s1.setTargeting('abc', 123);
        var s2 = googletag.defineSlot('/43743431/DMDemo1', [
            [300, 250],
            [300, 600]
        ], 'Div2').addService(googletag.pubads());
        s2.setTargeting('2abc', 123);
        // common targetings
        googletag.pubads().setTargeting('v', 'classifieds');
        googletag.pubads().setTargeting('article-id', '65207');
        googletag.pubads().enableSingleRequest();
        googletag.pubads().disableInitialLoad();
        googletag.enableServices();
    });

```

```

// OpenWrap code START here
if(typeof PWT.requestBids === 'function'){
  PWT.requestBids(
    PWT.generateConfForGPT(googletag.pubads().getSlots()),
    function(adUnitsArray) {
      PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
      PWT.ow_BidsReceived = true;
      initAdserver(false);
    }
  );
}
// No need to handle "else" part as we have A9 wrapper on page
// OpenWrap code END here

var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap and A9 timeout
setTimeout(function() {
  initAdserver(true); // calling this function with forced mode set to true so that GPT API is
always executed
}, FAILSAFE_TIMEOUT);

});
</script>
</head>
<body style='background-color:whitesmoke;'>
  <br><br>
  <!-- DMDemo -->
  Adslot : /15671365/DMDemo
  <br>
  <div id='Div1'>
  <script>
    googletag.cmd.push(function() {
      googletag.display("Div1");
    });
  </script>
  </div>
  <br><br>
  <!-- DMDemo1 -->
  Adslot : /15671365/DMDemo1
  <br>
  <div id='Div2'>
  <script>
    googletag.cmd.push(function() {
      googletag.display("Div2");
    });
  </script>
  </div>
</body>
</html>

```

OpenWrap JavaScript API is a straightforward [API](#) you can use to integrate [OpenWrap](#) with any ad server. The JavaScript API is the recommended integration method when using multiple wrapper solutions like A9. OpenWrap JavaScript does not take control of Google Ad Manager/[Google Publisher Tag](#) flow, it lets publishers call OpenWrap APIs to initiate bidder calls.

Get Started

To use the OpenWrap JavaScript API, set **Platform** to *Display* and **AdServer** to *Custom* in your [OpenWrap profile](#) .



This guide assumes a working knowledge of OpenWrap...

Your PubMatic account must meet [OpenWrap's Publisher Requirements](#). Learn more about setting up and using OpenWrap in the, [OpenWrap User Guide](#).

API Methods

The OpenWrap JavaScript API contains one main method that you can use on any ad server, and three convenience methods you can use on Google Ad Manager and [GPT](#). The following detail sections include parameter and response examples. See the [Example Pages](#) section for a variety of on-page implementation scenarios.

PWT.requestBids

Use on any ad server to call bidders with your OpenWrap configurations for the provided slots. Performs an auction for each slot in response to bidders. Generates key-value pairs for each slot/auction. Adds winning bid results and key-value-pairs as a `bidData` object to the slot configuration objects. Returns the modified config objects containing `bidData` to the callback function specified in the second parameter.

Parameters

1. Array of OpenWrap configuration objects...

```
[
  {
    code: "some-pub-friendly-unique-name-1", // mandatory
    divId: "pubmatic_test2", // mandatory
    adUnitId: "ad_unit-id-from-DFP_1", // mandatory
    adUnitIndex: "0", // mandatory
    mediaTypes: { // mandatory
      banner: { // mandatory
        sizes: [ [300, 250] ] // mandatory: array of sizes
      }
    }
  },
  {
    code: "some-pub-friendly-unique-name-2", // mandatory
    divId: "pubmatic_test-3", // mandatory
    adUnitId: "ad_unit-id-from-DFP_2", // mandatory
    adUnitIndex: "0", // mandatory
    mediaTypes: { // mandatory
      banner: { // mandatory
        sizes: [ [300, 250] ] // mandatory: array of sizes
      }
    }
  }
]
```

2. Callback function...

```
function(resp){
  console.log("response:", resp);
  for(var i = 0, l = resp.length; i<l; i++){
    var r = resp[i];
    console.log("");
    console.log("code: ", r.code);
    if(r.bidData){
      console.log("key value pairs", (r.bidData.kvp || ""));
      console.log("Bid details", (r.bidData.wb || {}));
      // Publisher to add keys on ad Slot
    }
  }
}
```

Response

Modified OpenWrap configuration objects containing the `bidData` object.

bidData...

```
bidData : {
  // winning bid data information
  wb: {
    "adHtml": "<CREATIVE CODE>"
    "width": 300,
    "height": 250,
    "netEcpm": "1.0",
    "grossEcpm": "1.0",
    "adapterID": "pubmatic"
  },

  // key value pairs generated by OpenWrap that publisher should add to resepective adserver slot

  kvp: {
    "key1": "value1",
    "pwtSid": "12fjg89",
    "pwtSz": "300x250"
  }
}
```

Convenience Methods

Use the three convenience methods below to streamline integration with Google Ad Manager and Google Publisher Tag.

For other ad servers...

Create your own methods that provide the following functionality:

- Generate an array of OpenWrap configurations.
- Add OpenWrap key-value pairs to GPT ad slots.
- Remove key-value pairs from ad slots.

Convenience Method	Description
OpenWrap JavaScript API#PWT.generateConfForGPT	Generates the configuration array used by <code>PWT.requestBids</code> .
OpenWrap JavaScript API#PWT.addKeyValuePairsToGPTSlots	Adds OpenWrap key-value pairs to GPT ad slots.
OpenWrap JavaScript API#PWT.removeKeyValuePairsFromGPTSlots	Removes key-value pairs added to ad slots by OpenWrap.

PWT.generateConfForGPT

Generates the OpenWrap configuration array required for `PWT.requestBids`.

 Works only for Google Ad Manager/GPT.

Parameters

Array of GPT ad slot objects.

Response

Array of OpenWrap configuration objects required for the first parameter of `PWT.requestBids`.

PWT.addKeyValuePairsToGPTSlots

Adds OpenWrap key-value pairs on respective GPT ad slots.



Works only for Google Ad Manager/GPT.

Parameters

Array of configuration objects returned to the callback function specified in the second parameter of `PWT.requestBids` .

Response

None

PWT.removeKeyValuePairsFromGPTSlots

Removes the key-value pairs added by OpenWrap on the given GPT ad slots.



Works only for Google Ad Manager/GPT.

Parameters

Array of GPT ad slot objects.

Response

None

Example Pages

The following examples demonstrate using the OpenWrap JavaScript API in various on-page scenarios such as refreshing ad slots, infinitely scrolling pages, A9, and so on.

Simple Setup

A simple example of integrating OpenWrap on-page with a single call to `PWT.requestBids`.

Simple Example

```
<!DOCTYPE html>
<html>
<head>
  <title>OW Custom</title>
</head>
<body>

  <script type="text/javascript" src="owt.min.js"></script>
  <script type="text/javascript">
    PWT.requestBids(
      [
        {
          code: "some-pub-friendly-unique-name-1", // mandatory
          divId: "pubmatic_test2", // mandatory
          adUnitId: "ad_unit-id-from-DFP", // mandatory
          adUnitIndex: "", // necessary in case of PubMatic, can be derived by our code by simply
incrementing used adUnitIds (NOT in scope now)
          mediaTypes: { // mandatory
            banner: { // mandatory in first phase? or atleast one type of mediaTypes should be
present
              sizes: [ [300, 250] ] // array of sizes
            }
          }
        }, {
          code: "some-pub-friendly-unique-name-2", // mandatory
          divId: "pubmatic_test-3", // mandatory
          adUnitId: "ad_unit-id-from-DFP", // mandatory
          adUnitIndex: "", // necessary in case of PubMatic, can be derived by our code by simply
incrementing used adUnitIds (NOT in scope now)
          mediaTypes: { // mandatory
            banner: { // mandatory in first phase? or atleast one type of mediaTypes should be
present
              sizes: [ [300, 250] ] // array of sizes
            }
          }
        }
      ],
      function(resp){
        console.log("response:", resp);
        for(var i = 0, l = resp.length; i<l; i++){
          var r = resp[i];
          console.log("");
          console.log("code: ", r.code);
          if(r.bidData){
            console.log("key value pairs", (r.bidData.kvp || ""));
            console.log("Bid details", (r.bidData.wb || {}));
          }
        }
      }
    );
  </script>

</body>
</html>
```

Typical Design Pattern

The following example illustrates a typical structure for integrating OpenWrap JavaScript APIs in your pages. All examples following this one are built upon the same pattern.

Typical Design Example...

```
<!DOCTYPE HTML>
```

```

<html lang="en-us">

<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <title>NON SRA</title>
  <script type="text/javascript">
    var PWT = {};
    var googletag = googletag || {};
    googletag.cmd = googletag.cmd || [];
    var gptRan = false;
    PWT.jsLoaded = function() {
      loadGpt();
    };
    (function() {
      var purl = window.location.href;
      var url = '//ads.pubmatic.com/AdServer/js/pwt/5890/651';
      var profileVersionId = '';
      if (purl.indexOf('pwtv=') > 0) {
        var regexp = /pwtv=(.*?)(&|$)/g;
        var matches = regexp.exec(purl);
        if (matches.length >= 2 && matches[1].length > 0) {
          profileVersionId = '/' + matches[1];
        }
      }
      var wtads = document.createElement('script');
      wtads.async = true;
      wtads.type = 'text/javascript';
      wtads.src = url + profileVersionId + '/pwt.js';
      var node = document.getElementsByTagName('script')[0];
      node.parentNode.insertBefore(wtads, node);
    })();
    var loadGpt = function() {
      // Check the gptRan flag
      if (!gptRan) {
        gptRan = true;
        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);
      }
    };
    // Failsafe to call gpt
    setTimeout(loadGpt, 500);
  </script>

  <script type="text/javascript">

    googletag.cmd.push(function() {

      function initAdserver() {
        if (PWT.initAdserverSet) return;
        PWT.initAdserverSet = true;
        googletag.display('pubmatic_test2');
        googletag.display('Div2');
      }

      var s1 = googletag.defineSlot('/43743431/DMDemo', [
        [728, 90],
        [300, 250],
        [160, 600]
      ], 'pubmatic_test2').addService(googletag.pubads());
      s1.setTargeting('a', [1]);
      s1.setTargeting('b', [2]);
      s1.setTargeting('c', 3);
      s1.setTargeting('d', ['a', 'b', 'c']);

      var s2 = googletag.defineSlot('/43743431/DMDemo1', [
        [300, 250],
        [728, 90],
        [160, 600]
      ], 'pubmatic_test2').addService(googletag.pubads());
    });
  </script>

```

```

    ], 'Div2').addService(googletag.pubads());
    s2.setTargeting('aa', [11]);
    s2.setTargeting('bb', [22]);
    s2.setTargeting('dd', ['aa', 'bb', 'cc']);

    // common targetings
    googletag.pubads().setTargeting('v', 'classifieds');
    googletag.pubads().setTargeting('cpage', 'browseads');
    googletag.pubads().setTargeting('article-id', '65207');
    googletag.pubads().setTargeting('automation', '1');

    googletag.enableServices();

    if(typeof PWT.requestBids === 'function'){
        PWT.requestBids(
            PWT.generateConfForGPT(googletag.pubads().getSlots()),
            function(adUnitsArray) {
                PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
                initAdserver();
            }
        );
    }else{
        initAdserver();
    }

    var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap timeout
    setTimeout(function() {
        initAdserver();
    }, FAILSAFE_TIMEOUT);
    });
</script>
</head>

<body style='background-color:whitesmoke;'>
  <h2 style='text-align:center;'>
    NON SRA
  </h2>
  <br><br>
  <!-- DMDemo -->
  Adslot : /43743431/DMDemo
  <br>
  <div id='pubmatic_test2'>
  </div>
  <br><br>
  <!-- DMDemo1 -->
  Adslot : /43743431/DMDemo1
  <br>
  <div id='Div2'>
  </div>
</body>

</html>

```

Refreshing Slots

Similar to the previous example, but a snippet at the end refreshes all as slots after 15 seconds.

Refreshing Slots Example...

```

<!DOCTYPE HTML>
<html lang="en-us">

<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <title>NON SRA</title>

  <script type="text/javascript">
    var PWT = {};

```

```

var googletag = googletag || {};
googletag.cmd = googletag.cmd || [];
var gptRan = false;
PWT.jsLoaded = function() {
    loadGpt();
};
(function() {
    var purl = window.location.href;
    var url = '//ads.pubmatic.com/AdServer/js/pwt/5890/651';
    var profileVersionId = '';
    if (purl.indexOf('pwtv=') > 0) {
        var regexp = /pwtv=(.*?)(&|$)/g;
        var matches = regexp.exec(purl);
        if (matches.length >= 2 && matches[1].length > 0) {
            profileVersionId = '/' + matches[1];
        }
    }
    var wtads = document.createElement('script');
    wtads.async = true;
    wtads.type = 'text/javascript';
    wtads.src = url + profileVersionId + '/pwt.js';
    var node = document.getElementsByTagName('script')[0];
    node.parentNode.insertBefore(wtads, node);
})();
var loadGpt = function() {
    // Check the gptRan flag
    if (!gptRan) {
        gptRan = true;
        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);
    }
}
// Failsafe to call gpt
setTimeout(loadGpt, 500);
</script>

<script type="text/javascript">
    googletag.cmd.push(function() {

        function initAdserver() {
            if (PWT.initAdserverSet) return;
            PWT.initAdserverSet = true;
            googletag.display('pubmatic_test2');
            googletag.display('Div2');
        }

        var s1 = googletag.defineSlot('/43743431/DMDemo', [
            [728, 90],
            [300, 250],
            [160, 600]
        ], 'pubmatic_test2').addService(googletag.pubads());
        s1.setTargeting('a', [1]);
        s1.setTargeting('b', [2]);
        s1.setTargeting('c', 3);
        s1.setTargeting('d', ['a', 'b', 'c']);

        var s2 = googletag.defineSlot('/43743431/DMDemo1', [
            [300, 250],
            [728, 90],
            [160, 600]
        ], 'Div2').addService(googletag.pubads());
        s2.setTargeting('aa', [11]);
        s2.setTargeting('bb', [22]);
        s2.setTargeting('dd', ['aa', 'bb', 'cc']);

        // common targetings
        googletag.pubads().setTargeting('v', 'classifieds');
        googletag.pubads().setTargeting('cpage', 'browseads');
    });

```

```

googletag.pubads().setTargeting('article-id', '65207');
googletag.pubads().setTargeting('automation', '1');

googletag.enableServices();

if(typeof PWT.requestBids === 'function'){
  PWT.requestBids(
    PWT.generateConfForGPT(googletag.pubads().getSlots()),
    function(adUnitsArray) {
      PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
      initAdserver();
    }
  );
}else{
  initAdserver();
}

var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap timeout
setTimeout(function() {
  initAdserver();
}, FAILSAFE_TIMEOUT);
});
</script>
</head>

<body style='background-color:whitesmoke;'>
  <h2 style='text-align:center;'>
    NON SRA
  </h2>
  <br><br>
  <!-- DMDemo -->
  Adslot : /43743431/DMDemo
  <br>
  <div id='pubmatic_test2'>
  </div>
  <br><br>
  <!-- DMDemo1 -->
  Adslot : /43743431/DMDemo1
  <br>
  <div id='Div2'>
  </div>

<script>
  // refresh all slots after 15 seconds

  setTimeout(function(){

    function refreshSlots(){
      if (PWT.initAdserverSet) return;
      PWT.initAdserverSet = true;
      googletag.pubads().refresh(); // calling refresh slots function of GPT
    }

    if(typeof PWT.removeKeyValuePairsFromGPTSlots === 'function'){
      PWT.initAdserverSet = false;
      // first remove already added OpenWrap targeting key value pairs from slots
      PWT.removeKeyValuePairsFromGPTSlots( googletag.pubads().getSlots() );
      // now call OpenWrap to get bids and add respective key value pairs
      PWT.requestBids(
        PWT.generateConfForGPT( googletag.pubads().getSlots() ),
        function(adUnitsArray){
          PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
          refreshSlots();
        }
      );
    }
  }else{
    refreshSlots();
  }

  var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap timeout
  setTimeout(function() {

```

```

        refreshSlots();
    }, FAILSAFE_TIMEOUT);

    }, 15000);
</script>
</body>

</html>

```

Infinite Scroll

The following example demonstrates using OpenWrap JavaScript APIs on infinitely scrolling pages.

Infinite Scroll Example...

```

<html>
<head>
<title>Infinite Scroll Example</title>

<script type="text/javascript">
    var PWT = {};
    var googletag = googletag || {};
    googletag.cmd = googletag.cmd || [];
    var gptRan = false;
    PWT.jsLoaded = function() {
        loadGpt();
    };
    (function() {
        var purl = window.location.href;
        var url = '//ads.pubmatic.com/AdServer/js/pwt/5890/651';
        var profileVersionId = '';
        if (purl.indexOf('pwtv=') > 0) {
            var regexp = /pwtv=(.*?)(&|$)/g;
            var matches = regexp.exec(purl);
            if (matches.length >= 2 && matches[1].length > 0) {
                profileVersionId = '/' + matches[1];
            }
        }
        var wtads = document.createElement('script');
        wtads.async = true;
        wtads.type = 'text/javascript';
        wtads.src = url + profileVersionId + '/pwt.js';
        var node = document.getElementsByTagName('script')[0];
        node.parentNode.insertBefore(wtads, node);
    })();
    var loadGpt = function() {
        // Check the gptRan flag
        if (!gptRan) {
            gptRan = true;
            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);
        }
    };
    // Failsafe to call gpt
    setTimeout(loadGpt, 500);
</script>

<script>

var adslot0;

googletag.cmd.push(function() {

    // Declare any slots initially present on the page

```

```

adslot0 = googletag.defineSlot('/6355419/Travel', [728, 90], 'leaderboard').
    setTargeting("test","infinitemscroll").
    addService(googletag.pubads());

// Infinite scroll requires SRA
googletag.pubads().enableSingleRequest();

// Disable initial load, we will use refresh() to fetch ads.
// Calling this function means that display() calls just
// register the slot as ready, but do not fetch ads for it.
googletag.pubads().disableInitialLoad();

// Enable services
googletag.enableServices();
});

// Function to generate unique names for slots
var nextSlotId = 1;
function generateNextSlotName() {
    var id = nextSlotId++;
    return 'adslot' + id;
}

// Function to add content to page, mimics real infinite scroll
// but keeps it much simpler from a code perspective.
function moreContent() {

    // Generate next slot name
    var slotName = generateNextSlotName();

    // Create a div for the slot
    var slotDiv = document.createElement('div');
    slotDiv.id = slotName; // Id must be the same as slotName
    document.body.appendChild(slotDiv);

    // Create and add fake content 1
    var h1=document.createElement("H2")
    var text1=document.createTextNode("Dynamic Fake Content 1");
    h1.appendChild(text1);
    document.body.appendChild(h1);

    // Create and add fake content 2
    var h2=document.createElement("H2")
    var text2=document.createTextNode("Dynamic Fake Content 2");
    h2.appendChild(text2);
    document.body.appendChild(h2);

    // Define the slot itself, call display() to
    // register the div and refresh() to fetch ad.
    googletag.cmd.push(function() {
        var slot = googletag.defineSlot('/6355419/Travel', [728, 90], slotName).
            setTargeting("test","infinitemscroll").
            addService(googletag.pubads());
        // Display has to be called before

        // refresh and after the slot div is in the page.
        // googletag.display(slotName); // commented original code
        // googletag.pubads().refresh([slot]); // commented original code

        function initAdserver() {
            if (PWT.initAdserverSet) return;
            PWT.initAdserverSet = true;
            googletag.display(slotName); // moved original code here
            googletag.pubads().refresh([slot]); // moved original code here
        }

        // this is a new code for OpenWrap
        if(typeof PWT.requestBids === 'function'){
            PWT.initAdserverSet = false;
            PWT.requestBids(
                PWT.generateConfForGPT( [slot] ),

```



```

        function(adUnitsArray){
            PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
            initAdserver();
        }
    );
}else{
    initAdserver();
}

var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap timeout
setTimeout(function() {
    initAdserver();
}, FAILSAFE_TIMEOUT);
});
}
</script>

<style>
body > div {
    margin-bottom: 1em;
    border: solid 1px black;
    width: 728px
}
body > img {
    margin-bottom: 1em;
    display: block
}
body > button {
    position: fixed;
    bottom: 10px;
}
</style>
</head>

<body>
<h1>GPT Test Page - Infinite Scroll</h1>

<!-- First fake content -->
<text>
<h2> Initial Fake Content </h2>
</text>

<!-- First ad -->
<div id="leaderboard">
    <script>
        // Call display() to register the slot as ready
        // and refresh() to fetch an ad.
        googletag.cmd.push(function() {

            //googletag.display('leaderboard'); // commented this original code
            //googletag.pubads().refresh([adslot0]); // commented this original code

            function initAdserver() {
                if (PWT.initAdserverSet) return;
                PWT.initAdserverSet = true;
                googletag.display('leaderboard'); // moved original code here
                googletag.pubads().refresh([adslot0]); // moved original code here
            }

            // this is a new code for OpenWrap
            if(typeof PWT.requestBids === 'function'){
                PWT.requestBids(
                    PWT.generateConfForGPT( [adslot0] ),
                    function(adUnitsArray){
                        PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
                        initAdserver();
                    }
                );
            }else{
                initAdserver();
            }
        }
    </script>

```

```

        var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap timeout
        setTimeout(function() {
            initAdserver();
        }, FAILSAFE_TIMEOUT);
    });
</script>
</div>

<!-- More initial fake content -->
<text>
  <h2> Initial Fake Content 1 </h2>
  <h2> Initial Fake Content 2 </h2>
</text>

<!-- Button to load more content dynamically. -->
<button onclick="moreContent()">More Content</button>

</body>
</html>

```

Working With A9

In addition to demonstrating integration with A9, this example also uses GPT's `disableInitialLoad` API.

A9 Integration Example...

```

<!DOCTYPE HTML>
<html lang="en-us">
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <title>Wrapper Tag GPT Async Demo with A9</title>

  <h4>GPT implementation is ASYNC SRA <a href="https://support.google.com/dfp_premium/answer/1638622?hl=en">
(Refer this for types of GPT Tags)</a></h4>

<!-- OpwenWrap Script begins here -->
<script type="text/javascript">
  var PWT = {};
  var googletag = googletag || {};
  googletag.cmd = googletag.cmd || [];
  var gptRan = false;
  PWT.jsLoaded = function() {
    loadGpt();
  };
  (function() {
    var purl = window.location.href;
    var url = '//ads.pubmatic.com/AdServer/js/pwt/5890/4162';
    var profileVersionId = '';
    if (purl.indexOf('pwtv=') > 0) {
      var regexp = /pwtv=(.*?)(&|$)/g;
      var matches = regexp.exec(purl);
      if (matches.length >= 2 && matches[1].length > 0) {
        profileVersionId = '/' + matches[1];
      }
    }
    var wtads = document.createElement('script');
    wtads.async = true;
    wtads.type = 'text/javascript';
    wtads.src = url + profileVersionId + '/pwt.js';
    var node = document.getElementsByTagName('script')[0];
    node.parentNode.insertBefore(wtads, node);
  })();
  var loadGpt = function() {
    // Check the gptRan flag
    if (!gptRan) {
      gptRan = true;

```

```

        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);
    }
}
// Failsafe to call gpt
setTimeout(loadGpt, 500);
</script>
<!--Wrapper Script ends here -->

<script type="text/javascript">
    // this function will act as a lock and will call the GPT API
    function initAdserver(forced) {
        if((forced === true && window.initAdserverFlag !== true) || (PWT.a9_BidsReceived && PWT.
ow_BidsReceived)){
            window.initAdserverFlag = true;
            PWT.a9_BidsReceived = PWT.ow_BidsReceived = false;
            googletag.pubads().refresh();
        }
    }
</script>

<!-- A9 Script Start -->
<script>
    !function(a9,a,p,s,t,A,g){if(a[a9])return;function q(c,r){a[a9]&&_Q.push([c,r])}a[a9]={init:function(){q
("i",arguments)},fetchBids:function(){q("f",arguments)},setDisplayBids:function(){},_Q:[]};A=p.createElement(s);
A.async=!0;A.src=t;g=p.getElementsByTagName(s)[0];g.parentNode.insertBefore(A,g)}("apstag",window,document,"
script","//c.amazon-adsystem.com/aax2/apstag.js");
    apstag.init({
        pubID: 'xxxx',
        adServer: 'googletag'
    });
    apstag.fetchBids({
        slots: [{
            slotID: 'Div2',
            sizes: [[300, 250], [300, 600]]
        },
        {
            slotID: 'Div1',
            sizes: [[728 ,90]]
        }
    ],
        timeout: 2e3 // Make Sure this timeout is less than or equal to OpenWrap TimeOut
    }, function(bids) {
        googletag.cmd.push(function(){
            apstag.setDisplayBids();
            PWT.a9_BidsReceived = true;
            initAdserver(false);
        });
    });
</script>
<!-- A9 Script End -->

<script type="text/javascript">
    googletag.cmd.push(function() {
        var s1 = googletag.defineSlot('/43743431/DMDemo', [
            [728, 90]
        ], 'Div1').addService(googletag.pubads());
        s1.setTargeting('abc', 123);
        var s2 = googletag.defineSlot('/43743431/DMDemo1', [
            [300, 250],
            [300, 600]
        ], 'Div2').addService(googletag.pubads());
        s2.setTargeting('2abc', 123);
        // common targetings
        googletag.pubads().setTargeting('v', 'classifieds');
        googletag.pubads().setTargeting('article-id', '65207');
        googletag.pubads().enableSingleRequest();
        googletag.pubads().disableInitialLoad();
        googletag.enableServices();
    });
</script>

```

```

// OpenWrap code START here
if(typeof PWT.requestBids === 'function'){
    PWT.requestBids(
        PWT.generateConfForGPT(googletag.pubads().getSlots()),
        function(adUnitsArray) {
            PWT.addKeyValuePairsToGPTSlots(adUnitsArray);
            PWT.ow_BidsReceived = true;
            initAdserver(false);
        }
    );
}
// No need to handle "else" part as we have A9 wrapper on page
// OpenWrap code END here

var FAILSAFE_TIMEOUT = 1000; // this timeout should be more than OpenWrap and A9 timeout
setTimeout(function() {
    initAdserver(true); // calling this function with forced mode set to true so that GPT API is
always executed
}, FAILSAFE_TIMEOUT);

});
</script>
</head>
<body style='background-color:whitesmoke;'>
<br><br>
<!-- DMDemo -->
Adslot : /15671365/DMDemo
<br>
<div id='Div1'>
<script>
    googletag.cmd.push(function() {
        googletag.display("Div1");
    });
</script>
</div>
<br><br>
<!-- DMDemo1 -->
Adslot : /15671365/DMDemo1
<br>
<div id='Div2'>
<script>
    googletag.cmd.push(function() {
        googletag.display("Div2");
    });
</script>
</div>
</body>
</html>

```