

Overview

TopAds - Ad Management Script

If you're implementing on a new website, [check the New Website Implementation Guide](#).

A JavaScript library for managing Google Publisher Tag (GPT) advertisements on mobile websites. TopAds simplifies ad implementation with automatic placement detection, lazy loading, refresh management, and support for various ad formats including static banners, anchor ads, interstitials, and offerwalls.



Table of Contents

- [Overview](#)
- [Quick Start](#)
- [Configuration](#)
- [Configuration Reference](#)
- [Static Ads](#)
- [Refresh Management](#)
- [Single Page Applications \(SPA\)](#)
- [Advanced Features](#)

Overview

Purpose

TopAds is designed to streamline ad management on mobile websites by:

- Automatic Ad Detection: Automatically finds and processes static ad placements in your HTML
- Multiple Ad Formats: Supports static banners, anchor ads (top/bottom), interstitials, and offerwalls
- Smart Refresh: Configurable automatic ad refresh based on visibility and time intervals
- Lazy Loading: Optimizes performance with configurable lazy loading strategies
- SPA Support: Built-in support for Single Page Applications with route change handling
- Mobile-Only: Automatically detects and runs only on mobile devices
- Page Control: Include/exclude specific pages from showing ads

How It Works

1. The script automatically initializes when the page loads (unless `autoStart: false`)
2. It detects mobile devices and validates page permissions
3. Static ad placements are found by scanning for elements with `data-topads` attributes
4. Dynamic ad formats (anchor, interstitial, offerwall) are loaded based on configuration
5. Google Publisher Tag (GPT) is loaded and configured
6. Ads are displayed with lazy loading and refresh capabilities as configured

Quick Start

2. Customize your settings

Check below the reference for customizing the settings for topAds.

```
window.topAds = window.topAds || {};
```

```
topAds.config = {
  domain: 'WEBSITE_NAME',
  networkCode: 'YOUR_NETWORK_CODE',
  lazyLoad: 'hard',
  refresh: {
    time: 60,
    status: 'active',
    anchor: 'active',
  },
  formats: {
    anchor: {
      status: 'active',
      position: 'bottom',
    },
    interstitial: {
      status: 'active',
    },
    offerwall: {
      status: 'active',
      logoUrl: 'https://example.com/logo.png',
      websiteName: 'My Website',
      cooldown: '12',
    },
  },
},
```

```
};
```

3. Add Static Ad Placements (optional)

Add ad containers in your HTML:

```
<div id="square01" data-topads data-topads-size="square"></div>
```

That's it! TopAds will automatically detect and display ads.

Configuration

Configuration File Example

Here's a complete configuration example with all available options:

```
window.topAds = window.topAds || {};
```

```
topAds.config = {  
  // Required: This is the identifier for all adunits on your website. The  
  // final name of adunits will be the domain + DIV ID or format name.  
  domain: 'example.com',  
  
  // Required: Your Google Ad Manager network code  
  networkCode: '12345678901',  
  
  // Optional: Auto-start script on page load (default: true)  
  autoStart: true,  
  
  // Optional: Lazy loading strategy  
  lazyLoad: 'hard', // 'soft' | 'hard' | 'auto'  
  
  // Refresh configuration (optional)  
  refresh: {  
    time: 60, // Refresh interval in seconds (minimum: 30)  
    status: 'active', // 'active' | 'inactive'  
    anchor: 'active', // 'active' | 'inactive' (for anchor ads only)  
  },  
  
  // Page-level settings  
  pageSetting: {  
    // Include only these pages (optional)  
    include: ['/home', '/articles/*'],  
    // Exclude these pages (optional)  
    exclude: ['/terms', '/privacy', '/contact'],  
  },  
  
  // Ad format configurations
```

```

formats: {
  // Anchor ads (sticky top/bottom banners)
  anchor: {
    status: 'active',          // 'active' | 'inactive'
    position: 'bottom',        // 'top' | 'bottom'
    include: [],               // Optional: specific pages to include
    exclude: [],               // Optional: specific pages to exclude
  },

  // Interstitial ads (full-screen)
  interstitial: {
    status: 'active',          // 'active' | 'inactive'
    include: [],               // Optional: specific pages to include
    exclude: [],               // Optional: specific pages to exclude
  },

  // Offerwall (rewarded ads)
  offerwall: {
    status: 'active',          // 'active' | 'inactive'
    logoUrl: 'https://example.com/logo.png', // Your website logo
    websiteName: 'My Website', // Your website name
    cooldown: '12',            // Cooldown period in hours (default: 12)
    include: [],               // Optional: specific pages to include
    exclude: [],               // Optional: specific pages to exclude
  },
},
};

```

Configuration Reference

The following table describes all available configuration options:

Configuration	Type	Required	Default	Description
<code>domain</code>	string	Yes	-	Your website domain (e.g., 'example.com'). Used to generate ad unit names.
<code>networkCode</code>	string	Yes	-	Your Google Ad Manager network code.

<code>autoStart</code>	boolean	No	<code>true</code>	Automatically start TopAds when the page loads. Set to <code>false</code> to manually start with <code>topAds.start()</code> .
<code>lazyLoad</code>	string	No	-	Lazy loading strategy: <code>'soft'</code> (300% fetch margin, 150% render margin), <code>'hard'</code> (150% fetch margin, 75% render margin), or <code>'auto'</code> (default GPT settings).
<code>refresh.time</code>	number	No	30	Time interval in seconds between ad refreshes (minimum: 30 seconds).
<code>refresh.status</code>	string	No	<code>'inactive'</code>	Global refresh status: <code>'active'</code> enables refresh for static ads, <code>'inactive'</code> disables it.
<code>refresh.anchor</code>	string	No	<code>'inactive'</code>	Refresh status specifically for anchor ads: <code>'active'</code> or <code>'inactive'</code> .
<code>pageSetting.include</code>	array	No	-	Array of page paths to include. Only these pages will show ads. Supports wildcards (e.g., <code>['/articles/*']</code>).

<code>pageSetting.exclude</code>	array	No	-	Array of page paths to exclude. These pages will not show ads. Supports wildcards (e.g., <code>['/admin/*']</code>).
<code>formats.anchor.status</code>	string	No	<code>'inactive'</code>	Enable/disable anchor ads: <code>'active'</code> or <code>'inactive'</code> .
<code>formats.anchor.position</code>	string	No	<code>'bottom'</code>	Anchor ad position: <code>'top'</code> or <code>'bottom'</code> .
<code>formats.anchor.include</code>	array	No	-	Specific pages to show anchor ads (optional).
<code>formats.anchor.exclude</code>	array	No	-	Specific pages to exclude anchor ads (optional).
<code>formats.interstitial.status</code>	string	No	<code>'inactive'</code>	Enable/disable interstitial ads: <code>'active'</code> or <code>'inactive'</code> .
<code>formats.interstitial.include</code>	array	No	-	Specific pages to show interstitial ads (optional).
<code>formats.interstitial.exclude</code>	array	No	-	Specific pages to exclude interstitial ads (optional).
<code>formats.offerwall.status</code>	string	No	<code>'inactive'</code>	Enable/disable offerwall (rewarded ads): <code>'active'</code> or <code>'inactive'</code> .

<code>formats.offerwall.logoUrl</code>	string	No	-	URL of your website logo to display in the offerwall modal.
<code>formats.offerwall.websiteName</code>	string	No	-	Your website name to display in the offerwall modal.
<code>formats.offerwall.cooldown</code>	string	No	'12'	Cooldown period in hours before the offerwall can be shown again after a user completes it.
<code>formats.offerwall.include</code>	array	No	-	Specific pages to show offerwall (optional).
<code>formats.offerwall.exclude</code>	array	No	-	Specific pages to exclude offerwall (optional).

Page Path Matching

The `include` and `exclude` arrays support wildcard patterns:

- Exact match: `'/home'` matches only `/home`
- Wildcard: `'/articles/*'` matches `/articles/123, /articles/abc, etc.`
- Multiple patterns: `[' /home', ' /articles/*', ' /blog/*']`

Priority:

1. If `exclude` is set and the current page matches, ads are disabled
2. If `include` is set and the current page doesn't match, ads are disabled
3. Otherwise, ads are enabled

Static Ads

Static ads are HTML elements that you place in your page markup. TopAds automatically detects and processes them.

Basic Usage

Add a `div` element with the `data-topads` attribute and specify the size:

```
<div id="my-ad" data-topads data-topads-size="square"></div>
```

Available Sizes

Use the `data-topads-size` attribute with one of these predefined size groups:

Size Value	Ad Sizes Included
<code>square</code>	336x280, 300x250, 250x250
<code>vertical</code>	300x600, 300x250, 160x600, 120x600
<code>horizontal-sm</code>	320x100, 320x50, 300x100, 300x50
<code>horizontal-lg</code>	970x90, 728x90
<code>horizontal-xl</code>	970x250, 970x90, 728x90

Element Requirements

- ID Required: Each ad container must have a unique `id` attribute
- Unique IDs: Duplicate IDs will be detected and marked as invalid
- Automatic Styling: TopAds automatically sets width, height, and display properties if not already set

Example: Multiple Static Ads

```
<!-- Square ad in sidebar -->
<aside>
  <div id="sidebar-ad-1" data-topads data-topads-size="square"></div>
</aside>
```

```

<!-- Horizontal banner in content -->
<article>
  <div id="content-ad-1" data-topads
data-topads-size="horizontal-lg"></div>
  <p>Article content here...</p>
  <div id="content-ad-2" data-topads data-topads-size="square"></div>
</article>

<!-- Vertical ad at the end -->
<footer>
  <div id="footer-ad-1" data-topads data-topads-size="vertical"></div>
</footer>

```

Disabling Refresh for Specific Ads

To prevent a specific static ad from refreshing, add the `data-topads-norefresh` attribute:

```

<div id="static-ad" data-topads data-topads-size="square"
data-topads-norefresh></div>

```

This ad will load once but will not refresh, even if global refresh is enabled.

Refresh Management

TopAds can automatically refresh ads after they've been viewed for a specified time period. This helps maximize ad revenue by showing fresh ads to users.

How Refresh Works

1. When an ad becomes viewable (visible in the viewport), TopAds starts a timer
2. After the configured `refresh.time` interval, the ad is refreshed
3. The refresh only occurs if the ad is still visible (for static ads)
4. Refresh can be enabled/disabled globally or per ad format

Enabling Refresh

To enable refresh, set the refresh configuration:

```

topAds.config = {
  // ... other config ...
  refresh: {
    time: 60,           // Refresh every 60 seconds
    status: 'active',   // Enable refresh globally
    anchor: 'active',   // Enable refresh for anchor ads
  }
}

```

```
    },  
};
```

Disabling Refresh

Global Disable

Set `refresh.status` to `'inactive'`:

```
refresh: {  
  status: 'inactive', // Disables refresh for all static ads  
  anchor: 'inactive', // Disables refresh for anchor ads  
},
```

Per-Ad Disable (Static Ads Only)

Add the `data-topads-norefresh` attribute to specific ad containers:

```
<div id="ad-no-refresh" data-topads data-topads-size="square"  
data-topads-norefresh></div>
```

Disable Anchor Refresh Only

Keep static ad refresh enabled but disable anchor refresh:

```
refresh: {  
  status: 'active', // Static ads will refresh  
  anchor: 'inactive', // Anchor ads will NOT refresh  
},
```

Refresh Behavior by Ad Type

Ad Type	Refresh Support	Notes
Static Ads	Yes	Controlled by <code>refresh.status</code> and <code>data-topads-norefresh</code> attribute
Anchor Ads	Yes	Controlled by <code>refresh.anchor</code> (requires <code>refresh.status: 'active'</code>)

Interstitial	No	Interstitials cannot be refreshed
Offerwall	No	Offerwalls cannot be refreshed

Refresh Timing

- Minimum Interval: 30 seconds (enforced by TopAds)
- Recommended: 60 seconds or more to comply with ad network policies
- Best Practice: Use longer intervals (60-120 seconds) to avoid user annoyance

Refresh Visibility Check

For static ads, refresh only occurs if:

1. The ad is currently visible in the viewport
2. The refresh timer has elapsed
3. The ad hasn't been manually disabled with `data-topads-norefresh`

Single Page Applications (SPA)

TopAds includes built-in support for Single Page Applications (SPAs) like React, Vue, Angular, etc. When routes change in an SPA, you need to reinitialize TopAds to clean up old ads and load new ones.

How SPA Support Works

The `topAds.spa()` method:

1. Destroys all existing ad slots
2. Clears refresh timers
3. Removes processed ad elements
4. Reinitializes TopAds for the new page/route

Implementation

Call `topAds.spa()` after each route change in your SPA:

React Router Example

```

import { useEffect } from 'react';
import { useLocation } from 'react-router-dom';

function App() {
  const location = useLocation();

  useEffect(() => {
    // Reinitialize TopAds on route change
    if (window.topAds && window.topAds.spa) {
      window.topAds.spa();
    }
  }, [location]);

  return (
    // Your app content
  );
}

```

Vue Router Example

```

import { watch } from 'vue';
import { useRoute } from 'vue-router';

export default {
  setup() {
    const route = useRoute();

    watch(
      () => route.path,
      () => {
        // Reinitialize TopAds on route change
        if (window.topAds && window.topAds.spa) {
          window.topAds.spa();
        }
      }
    );
  }
}

```

Vanilla JavaScript Example

```

// Example with a simple router
function handleRouteChange() {
  // Your route change logic

  // Reinitialize TopAds
  if (window.topAds && window.topAds.spa) {
    window.topAds.spa();
  }
}

```

Important Notes

- Always check for existence: Verify `window.topAds` and `window.topAds.spa` exist before calling
- Call after DOM updates: Ensure new ad containers are in the DOM before calling `spa()`
- Route-specific configs: Page-level `include/exclude` settings will be re-evaluated on each route change
- Offerwall cooldown: The offerwall cooldown persists across route changes (stored in `localStorage`)

Manual Initialization

If you set `autoStart: false`, you can manually start TopAds:

```
topAds.config = {  
  // ... config ...  
  autoStart: false,  
};  
  
// Later, start manually  
topAds.start();  
  
// Or in SPA route changes  
topAds.spa(); // This will also start if not already started
```

Advanced Features

Debug Mode

Enable debug mode to see detailed logs in the browser console:

```
// In browser console  
topAds.debug();
```

This toggles debug mode. Refresh the page to see detailed logs about:

- Ad placement detection
- GPT loading
- Refresh timers
- Slot rendering
- Error messages

To disable debug mode, run `topAds.debug()` again.

UTM Parameter Tracking

TopAds automatically captures and stores UTM parameters from the URL:

- UTM parameters are extracted from the current page URL
- They are stored in `sessionStorage` for the session duration
- They are passed to Google Ad Manager as targeting parameters
- Useful for tracking campaign performance

Preloader

TopAds shows a preloader while ads are loading. The preloader:

- Displays automatically when TopAds initializes
- Shows a loading spinner and message
- Automatically hides when the first ad renders
- Can be customized via CSS (class: `topads-preloader-active`)

Offerwall Flow

The offerwall (rewarded ads) works as follows:

1. Initialization: Offerwall is prepared when TopAds loads (if enabled)
2. Cooldown Check: If user completed offerwall recently, it's skipped
3. Modal Display: When the rewarded ad is ready, a modal is shown
4. User Action: User clicks button to watch the ad
5. Reward: After completing the ad, user gets site-wide access
6. Cooldown: Offerwall is disabled for the configured cooldown period (default: 12 hours)

Support & Troubleshooting

Common Issues

Ads not showing:

- Check browser console for errors
- Verify `networkCode` and `domain` are correct
- Ensure ad containers have unique IDs
- Check if page is excluded in `pageSetting`

Refresh not working:

- Verify `refresh.status` is set to `'active'`
- Check that `refresh.time` is at least 30 seconds
- For anchor ads, ensure `refresh.anchor` is `'active'`
- Check if ad has `data-topads-norefresh` attribute

SPA not working:

- Ensure `topAds.spa()` is called after route changes
- Verify new ad containers are in DOM before calling `spa()`
- Check browser console for errors

Browser Compatibility

TopAds works on:

- Modern mobile browsers (iOS Safari, Chrome Mobile, etc.)
- Requires JavaScript enabled
- Uses modern browser APIs (`localStorage`, `sessionStorage`)
-



New Website

TopAds Implementation Guide for New Websites

This guide will help you implement TopAds on a new website step by step.

Step 1: Gather Required Information

Before creating the configuration file, you need to gather the following information:

1.1 Pages to include/exclude

- Identify which pages of the site should NOT display ads (e.g., /privacy-policy, /terms, /contact-us)
- Identify if there are specific pages where the interstitial (full-screen ad) should NOT appear
- Identify if there are specific pages where the offerwall (offer wall) should NOT appear

1.2 Static placements

- Determine which static ad units will be rendered on the website. For performance publishing, usually the required ad units are:

Page Type	Block Name	Location
Landing Page	square01	Before the first paragraph
Benefits Page	square02	Before the first paragraph
Requirements Page	square03	Before the first paragraph
Requirements Page	square04	Before the third paragraph

1.3 Ad Unit Naming Convention

- You should also define how these ad units will be named. All ad units of the website should start with the same prefix, to make it easier for report pulling. For TopFinanzas US, for example, we used the `TOPFIN_US` prefix.
- With the ad unit prefix in hand, you can see how the ad units will be named. For example:

Prefix	Device	Ad Unit Name	Final Ad Unit Code
<code>TOPFIN_US</code>	<code>_mob</code>	<code>_square01</code>	<code>TOPFIN_US_mob_square01</code>
<code>TOPFIN_US</code>	<code>_mob</code>	<code>_square02</code>	<code>TOPFIN_US_mob_square02</code>
<code>TOPFIN_US</code>	<code>_mob</code>	<code>_square03</code>	<code>TOPFIN_US_mob_square03</code>
<code>TOPFIN_US</code>	<code>_mob</code>	<code>_square04</code>	<code>TOPFIN_US_mob_square04</code>
<code>TOPFIN_US</code>	<code>_mob</code>	<code>_interstitial</code>	<code>TOPFIN_US_mob_interstitial</code>
<code>TOPFIN_US</code>	<code>_mob</code>	<code>_offerwall</code>	<code>TOPFIN_US_mob_offerwall</code>

1.4 Logo for the Offerwall

- Get the public URL of the logo that will appear in the offerwall
- The URL must be publicly accessible (e.g., `https://example.com/logo.png`)

1.5 Network Code

- Get the Network Code from the Ad Manager that will be used for this site

Step 2: Create the Configuration File

Create a new JavaScript file with the following base format:

```
window.topAds = window.topAds || {};  
  
topAds.config = {  
  domain: 'AD_UNIT_PREFIX', // Add the ad unit prefix defined in step 1.3  
  networkCode: 'YOUR_NETWORK_CODE', // Add the network code from GAM  
  lazyLoad: 'soft',  
  pageSetting: {  
    exclude: [  
      '/page-1',  
      '/page-2',  
      // Add all pages that should NOT display ads  
    ]  
  },  
  formats: {  
    interstitial: {  
      status: 'active',  
      exclude: [  
        '/specific-page-without-interstitial',  
        // Add pages where the interstitial should NOT appear  
      ],  
    },  
    offerwall: {  
      status: 'active',  
      logoUrl: 'PUBLIC_LOGO_URL',  
      websiteName: 'Website Name',  
      cooldown: '12',  
      exclude: [  
        '/specific-page-without-offerwall',  
        // Add pages where the offerwall should NOT appear  
      ],  
    },  
  },  
}  
};  
  
(function () {  
  var w = window.top, d = w.document, h = d.head ||  
d.getElementsByTagName("head")[0];  
  var s = d.createElement("script");  
  s.src = "https://topads.topnetworks.co/topAds.min.js";  
  s.type = "text/javascript";  
  s.defer = true;  
  s.async = true;  
  s.setAttribute("data-cfasync", "false");  
  h.appendChild(s);  
})();
```

Complete configuration example:

```
window.topAds = window.topAds || {};  
  
topAds.config = {  
  domain: 'EXAMPLE_US',  
  networkCode: '23062212598',  
  lazyLoad: 'soft',  
  pageSetting: {  
    exclude: [  
      '/contact-us',  
      '/privacy-policy',  
      '/terms',  
      '/cookie-policy',  
      '/about-us',  
    ]  
  },  
  formats: {  
    interstitial: {  
      status: 'active',  
      exclude: [  
        '/credit-card-recommender-p3'  
      ],  
    },  
    offerwall: {  
      status: 'active',  
      logoUrl: 'https://example.com/logo.png',  
      websiteName: 'My Website',  
      cooldown: '12',  
      exclude: [  
        '/credit-card-recommender-p2',  
        '/credit-card-recommender-p3'  
      ],  
    },  
  },  
}  
};  
  
(function () {  
  var w = window.top, d = w.document, h = d.head ||  
d.getElementsByTagName("head")[0];  
  var s = d.createElement("script");  
  s.src = "https://topads.topnetworks.co/topAds.min.js";  
  s.type = "text/javascript";  
  s.defer = true;  
  s.async = true;  
  s.setAttribute("data-cfasync", "false");  
  h.appendChild(s);  
})();
```

Step 3: Create Ad Units in Ad Manager

1. Access Google Ad Manager, Inventory > Ad Units tab.
2. Create all ad units according to the nomenclature defined in Step 1.3. For example:
 - **Name:** TOPFIN_US_mob_square01
 - **Code:** TOPFIN_US_mob_square01

Step 4: Insert the Configuration Script on the Site

1. Copy the complete content of the configuration file created in Step 2
2. Insert the script in the `<head>` or `<body>` of all pages of the site where ads should appear
3. Alternatively, you can insert it in a global template if your site uses a template system

For SPA websites: Remember to call the `topAds.spa()` function every time the page is changed, so the script knows it's time to request new ads.






Step 5: Insert Static Ad Units on the Website.

1. On each page where you need to display static ads, insert the code for the corresponding block
2. The block code must follow the nomenclature defined in Step 1.2
3. Insertion example:

```
<div id="square01" data-topads data-topads-size="square"></div>
```

4. Place these divs in the desired positions within the HTML of each page

Summary of Steps

1.  Gather information (pages, static placements, naming convention, logo, network code)
2.  Create configuration file with the base structure
3.  Create ad blocks in the Ad Manager
4.  Insert configuration script on the site
5.  Insert static ad blocks on pages

Important Notes

- The offerwall `cooldown` is configured in hours (default: 12 hours)
- Routes in `exclude` must exactly match the page URLs
- The `logoUrl` must be a public and accessible URL
- The `networkCode` must be the correct code from the Ad Manager
- The `domain` must be unique for each site