

Add analytics.js to your website

Add analytics.js to your website

Analytics.js is a library developed in JavaScript by AFS Analytics in order to measure the traffic and analyze the behavior of visitors to a website.

AFS Analytics offers a library that is compatible with Google Analytics, which has the same structure and functions. This compatibility makes it easy to implement AFS Analytics for those accustomed to Google Analytics. At the end of this guide, a section is devoted to the difference between the JavaScript tracking snippet of Google and that of AFS Analytics, and how to replace it quickly.

The JavaScript tracking snippet to add to your website

The JavaScript tracking code, also called JavaScript tracking snippet consists of a few lines. It can be pasted between the `<head>` and `</head>` (highly recommended) or `<body>` and `</body>` HTML tags of your website pages.

The `XXXXXXXX` string must be replaced with your site identifier `ID`. This unique 8-digit number was created when you registered your website on AFS Analytics. You can also find it by choosing the `Manage Websites` option in the `Account` menu on the dashboard.

AFS Analytics JavaScript tracking code

```
<script type="text/javascript">
(function(i,s,o,g,r,a,m){i['AfsAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','//code.afsanalytics.com/js/analytics.js','aa');
aa('create','XXXXXXXX','auto');
aa('set','autotrack','dataset');
aa('send','pageview',);
</script>
```

The above code does the following:

1. Loads the analytics.js library asynchronously.
2. Initializes the library and defines the name of the global function, as `aa()`.
3. Creates the `aa()` global function, called the `aa()` command queue, which saves the commands in a waiting list to be sent to analytics.js.
4. Adds the `create` command to the `aa()` command queue to create a new tracker object. Note: The string `XXXXXXXX` must be replaced with your site's ID.
5. Adds a second command to the `aa()` command queue to select the events tracking mode `autotrack`.
6. Adds one last command to the `aa()` command queue to send data about the current page view to AFS Analytics.

The example above shows the basic implementation of AFS Analytics on a website. This JavaScript code may require changes or modifications dedicated to your website.

The `aa()` function offers numerous additional commands and parameters to help you to track more interactions.

Note:The HTML/JavaScript tracking code preceding the `aa()` function doesn't need to be changed.

The above javascript code tracks by this data.

The time spent by the visitor on the site.
The time spent and the detail of each page visited (page view).
The date of their last visit.
The location of the visitor.
The configuration used.
The referring site.
The events defined by the `dataset` in the tags.

About the AFS logo displayed on your website:

The new version no longer displays the AFS Logo on your website pages. However, if you want to display it, you have to specify its location by adding the following HTML code where you want the logo to appear:

```
<div id = 'afsanalytics'></ div>
```

Note: For the free version, this tag is useful if your traffic exceeds the limit.

The next steps:

For basic activity reports, implementing the previous JavaScript tracking code is sufficient. For an advanced and customized installation, meaning tracking the interactions of the visitors with the content of your site, it is advisable that you read the following guides:

1. [How does analytics.js work?](#)
2. [Creating a tracker.](#)
3. [Getting and Setting tracker data.](#)
4. [Sending data to AFS Analytics.](#)

The difference between Google Analytics and AFS Analytics codes.

There are three differences between the two codes.

1. The name of the global variable of the object (2nd line of the code).
For Google Analytics, the variable is: `GoogleAnalyticsObject`.
For AFS analytics, the variable is: `AfsAnalyticsObject`
2. The URL of the analytic.js script: (5th line)
For Google: `https://www.google-analytics.com/analytics.js`
For AFS: `https://code.afsanalytics.com/js/analytics.js`

Note: The definition of the `https:` protocol is optional.

3. The name of the main calling function. (5th line)
For Google : `ga`.
For AFS: `aa`.

To Replace Google Analytics with AFS Analytics in 30 seconds:

Simply change a small portion of the Google code on your site page and keep the name of the call function:

1. Replace `GoogleAnalyticsObject` with `AfsAnalyticsObject`
2. Replace `https://www.google-analytics.com/analytics.js` with `https://code.afsanalytics.com/js/analytics.js`
3. Keep `ga` as the name of the calling function in order to get compatibility of calls to the global function. **The original Google**

Analytics JavaScript tracking snippet.

```
<script>
(function(i,s,o,g,r,a,m){i['AnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','https://www.google-analytics.com/analytics.js','ga');
ga('create', 'UA-XXXXX-Y', 'auto');
ga('send', 'pageview');
</script>
```

To be replaced with:

```
<script>
(function(i,s,o,g,r,a,m){i['AfsAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','//code.afsanalytics.com/js/analytics.js','ga');
ga('create', 'XXXXXXXXX', 'auto');
ga('send', 'pageview');
</script>
```

Note: Don't forget to replace `XXXXXXXXX` with the identifier `ID` of your website.

Calls to analytics.js in case the function name is "ga"

All calls to the analytics.js library are done using the `ga()` command. You can add the `autotrack` command to set the events tracking mode:

```
ga('create', 'XXXXXXXXX', 'auto');
ga('set', 'autotrack', 'dataset');
ga('send', 'pageview');
```