



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Developer's Guide

Monitoring

5/1/2025

# Monitoring

## Contents

- [1 Monitoring](#)
  - [1.1 Basic Configuration](#)
  - [1.2 Advanced Configuration](#)
  - [1.3 Monitoring JS API](#)
  - [1.4 Related Links](#)

The Monitoring Agent service records web browsing activity on your site. It generates basic system events such as `VisitStarted`, `PageEntered`, and additional custom business events, such as 'add-to-shopping cart'. Then it sends these events to the Web Engagement Server for further processing (you can read more about the structure of these events [here](#)).

To implement monitoring, you simply include the Monitoring Service JS script in your web pages. This short piece of regular JavaScript activates monitoring and notification functions by inserting one of the following scripts into the page: **GT.min.js**, **GTC.min.js**, **GPE.min.js**. The script depends on your requirements — see [Configuring the Instrumentation Script](#) for details. The JavaScript asynchronously loads the application into your pages, which means that Monitoring Service JS does not block other elements on your pages from loading.

## Basic Configuration

The simplest way to get the Monitoring Service JS for your site is by creating and adding a script to your web pages. See [Configuring the Instrumentation Script](#) for details.

### Important

If you plan to use Web Engagement chat, make sure to include the Chat JS Application script into your web pages, as well. See [Engagement](#) for details.

## Advanced Configuration

Once you have a basic configuration script, you can use it "as is" or implement the advanced configuration options to customize the script to suit your requirements. See [Configuring the Instrumentation Script](#) for details.

## Monitoring JS API

You can also take a highly customized approach and use the [Monitoring JS API](#) to submit events to the Web Engagement Server. You can submit `UserInfo`, `SignIn`, `SignOut`, and even your own custom business events using this API. For example, you can use the API to identify visitors on your website. See [Visitor Identification](#) for details.

## Related Links

- [Visitor Identification](#)
- [Events Structure](#)