Developer's Guide

Visitor Identification

12/17/2025

Visitor Identification

Contents

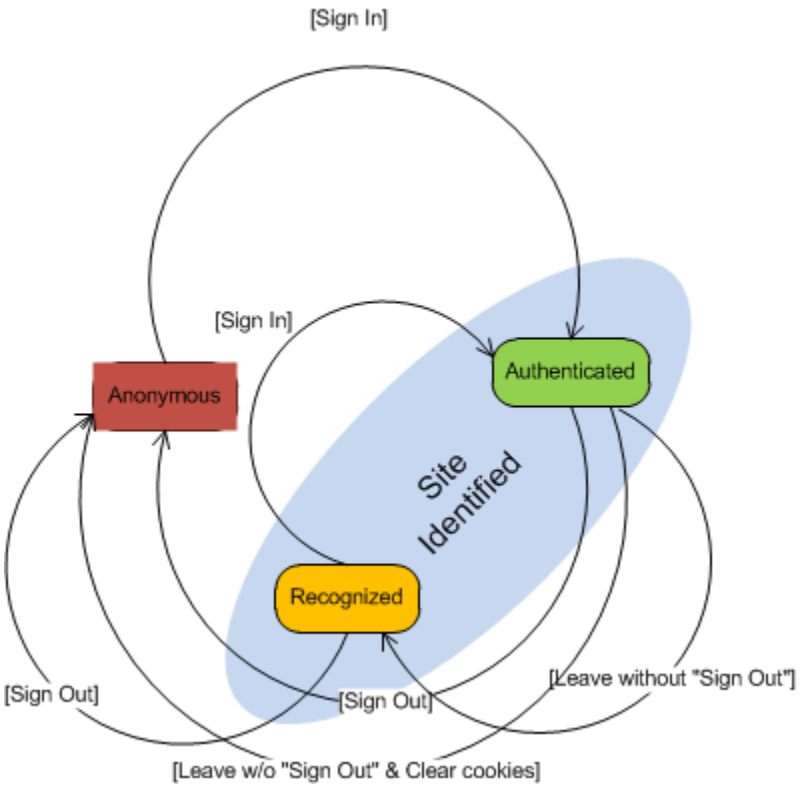
Overview

Genesys Web Engagement allows you to capture visitor activities on your website and to build a complete history of the visitor's interactions with your contact center.

When a visitor browses your website, the tracking code submits System events to the Web Engagement servers that constitute a visit, such as VisitStarted, PageEntered, SignIn, UserInfo, and so on. The association or relationship between the visit and the visitor is based on the flow derived from System events, in addition to the information retrieved from the Contact Server. In the end, you can access visit history through the Event Resource in the History REST API.

To associate the visitor with the visit, Genesys Web Engagement must "identify" the visitor as one of three possible states:

- **Authenticated** — The visitor logged in to the website with a username and password. The username can be an e-mail address, an account name or other similar identifier, depending on your website. When a user is authenticated, Genesys Web Engagement can maintain an association between the visitor and the visit.
- **Recognized** — The visitor closed the browser window and did not log out, but cookies are saved. The next time the visitor comes to the website, the website can submit cookie-based user information, which contains the **userId**.
- **Anonymous** — The visitor is anonymous.



Visitor states

Genesys Web Engagement relies on your website to trigger the transitions between visitor states. You can do this by updating the tracking code with the following events in the Monitoring JS API:

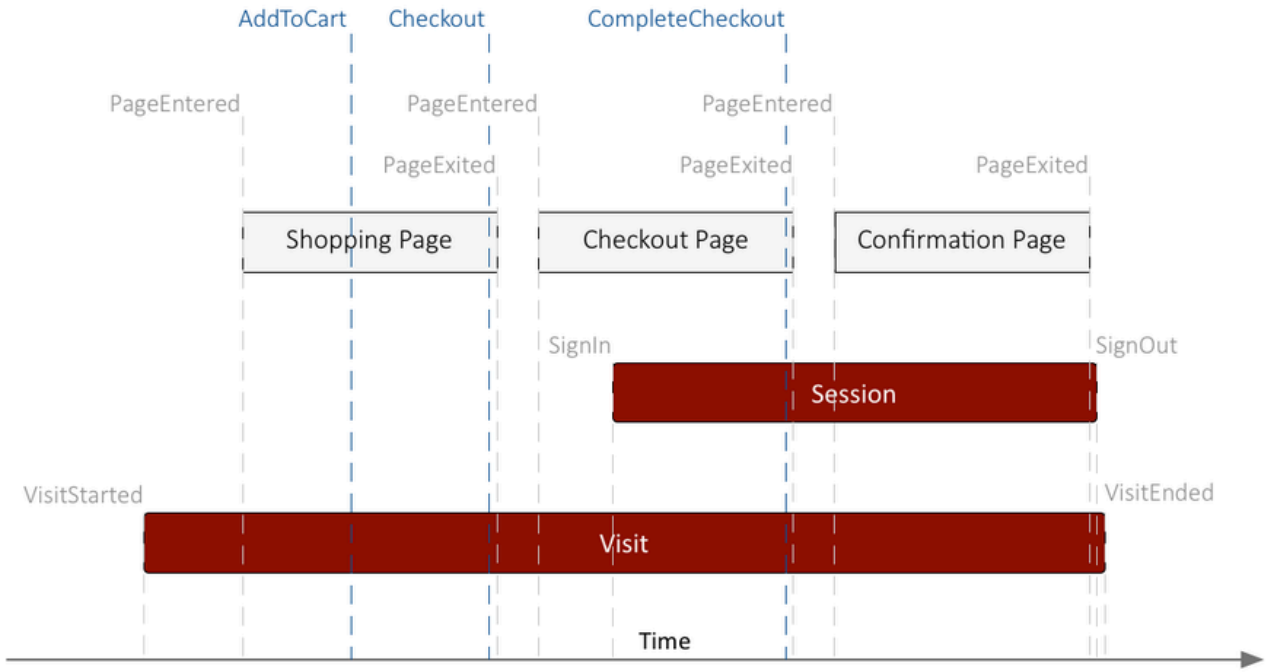
- _gt.push(['event', 'SignIn', { data: options }]) or _gt.push(['event', 'sendSignIn', options]) — Send this event when the user is authenticated by the website. This allows the system to identify the user and creates a new "session" with a **sessionId** that is unique to a visit and will last the duration of the visit. Only Authenticated visitors have an associated **sessionId**.

- _gt.push(['event', 'SignOut', { data: options }]) or _gt.push(['event', 'sendSignOut', options]) — Send this event when the user logs out of the website.
  Note: The **sessionId** lasts for the duration of the authenticated user's visit to your website. It is stored in a cookie and sent with every event that occurs between SignIn and SignOut, and is changed automatically after every SignIn event.

- _gt.push(['event', 'UserInfo', { data: options }]) or _gt.push(['event', 'sendUserInfo', options]) — Send this event when the user visits your website after closing the browser window on an authenticated session. For details, see Recognized Visitors.

Visitor Event Timeline

The figure below shows the timeline for events that take place when a visitor browses your website.



All visitors to your website are identified with a **visitId**, which can be used to associate the visitor to events, such as PageEntered or PageExited, during the span of the visit.

Accessing Visitor Information

The History REST API is a RESTful interface for accessing visit and identity information — in the form of a collection of JSON objects — via POST and GET HTTP requests:

- The visit resource represents the sequence of pages that a given visitor went through.
- The identity resource contains information about authenticated and recognized visitors.
- The page resource contains information about browsed pages. If a visitor revisits a page, a new page resource is created.
- The event resource contains information about System and Business events. For details about how these events are structured, see Events Structure.

Authenticated Visitors

When the visitor is Authenticated on the website, you should use the_gt.push(['event', 'SignIn', { data: options }]) or _gt.push(['event', 'sendSignIn', options]) event so that Genesys Web Engagement can start a new session. When the Web Engagement Server receives the related command, it creates a new session for the current visit. This process is completely transparent to the customer. The identifying information used to log in (for instance, the email address) is available in the SignIn event and is used to:

- Create the **identityId** or search the visitor's identity resource.
- Associate the visitor with a contact in the Genesys solution.

Recognized Visitors

When an Authenticated visitor closes the browser window without signing out and then later revisits your site, you can use the _gt.push(['event', 'UserInfo', { data: options }]) or _gt.push(['event', 'sendUserInfo', options]) command to tell Genesys Web Engagement that the visitor is now Recognized.

You will need to send the **userId** in the **_gt.push(['event', 'UserInfo', { data: options }])** or **_gt.push(['event', 'sendUserInfo', options])** event. How you track the **userId** depends on your website. For example, you could create a persistent cookie to store the **userId** when a visitor logs in to you website. Then when a visitor first browses your site, you could check the cookie and call the **_gt.push(['event', 'UserInfo', { data: options }])** or **_gt.push(['event', 'sendUserInfo', options])** event if the cookie contains the **userId**. There are many possible scenarios - the best implementation is entirely dependent on your website and its workflow.

> ### Important
> The visitor's identity cannot be guaranteed in the Recognized state. For instance, another member of the visitor's family could be browsing the website with the same computer.

Anonymous Visitors

If the visitor is not Authenticated or Recognized, he or she is treated as Anonymous. The visitor's activity on the website—including events and pages visited—is still associated with the visit.