Journey JavaScript SDK

autotrackIdle

5/25/2026

Contents

Learn how to configure when Genesys Predictive Engagement detects inactivity on a webpage. This configuration provides more accurate page tracking information for use in segments and outcomes.

> ### Important
> This article only applies to customers using web chat. If you are a Genesys Cloud CX customer, we encourage you to use the new web messaging feature to replace web chat.

Description

The `autotrackIdle` module tracks when and where a visitor becomes inactive on a webpage. To use `autotrackIdle`, configure the idle events to track using the following options:

- Config
- Callback

Signature

```
ac('load', 'autotrackIdle', [config], [callback]);
```

Example

```
ac('load', 'autotrackIdle'); // This is for all one idle event config with defaults

ac('load', 'autotrackIdle', {
  idleEvents: [
    {}, // This is for all defaults
    { eventName: 'stuck_on_page' },
    { idleAfter: 60 },
    { eventName: 'idle_for_2_min', idleAfter: 120 },
  ]
}, function () {
    console.log('"autotrackIdle" has been loaded');
});
```

Config (optional)

Description: Identifies an array of idle events to track.

Type: Object

Properties: See the following table.

Name	Description	Type	Status
idleAfter	Number of seconds of inactivity after which an event fires. The default is 60 seconds and the minimum is 30 seconds. **Note:** If you specify less than 30 seconds, 30 seconds is used.	Number	optional
eventName	String used as the event name when an element matching the selector is clicked.	String	optional

Example

```
{
  idleEvents: [
    {}, // This is for all defaults
    { eventName: 'stuck_on_page' },
    { idleAfter: 60 },
    { eventName: 'idle_for_2_min', idleAfter: 120 },
  ]
}
```

Callback (optional)

When a module loads, callback is executed. No arguments pass to the callback.