



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.

Journey JavaScript SDK

Form Tracking API

10/19/2024

Contents

- [1 About the Form Tracking API](#)
- [2 Enable form tracking](#)
- [3 Example: Track a webpage with 3 forms](#)
 - [3.1 Sample webpage with multiple forms](#)
 - [3.2 Track all forms in the HTML page](#)
 - [3.3 Track by element IDs](#)
 - [3.4 Track by element classes](#)
 - [3.5 Track by a combination of element IDs and element classes](#)
- [4 Manage form field data](#)
- [5 Transform data before sending it](#)
- [6 Sensitive form fields that are never tracked](#)

Learn how to use an API to track form submission and abandonment events, including what information visitors provide in your forms.

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.

About the Form Tracking API

To track customer activities on your website, you must first deploy the tracking snippet. Then, you can track form-level activity with the Form tracking API.

To track a form, it needs an ID, name, or action field that identifies it uniquely.

By default, the values of the input, select, and text area elements provided in the form (fields with personal information such as name, email, phone number, and company name) are sent with the form submission and abandoned events. The customer profile updates accordingly upon form submission.

Important

You can use Genesys Predictive Engagement to track visitor activity in a GDPR compliant-manner. However, you need to modify the tracking snippet to be compliant. For more information on how to be compliant with the GDPR requirements, see [General Data Protection Regulation \(GDPR\)](#).

Enable form tracking

To enable form tracking, call `forms:track` and specify the CSS selector of the form(s) that you want to track. For example:

```
ac('forms:track', '.hs-form');
```

To enable tracking for all forms, call the function without specifying the selector. For example:

```
ac('forms:track');
```

To enable tracking for multiple forms based on the CSS selector, specify a group of selectors as a comma-separated list. For example:

```
ac('forms:track', '#firstFormId, #secondFormId');
```

Important

Where the same selector is applied to multiple forms, each form with that selector is tracked separately.

```
ac('forms:track', '#firstFormId, #secondFormId');
```

Example: Track a webpage with 3 forms

Sample webpage with multiple forms

Tracking Forms Example

Get Name Form

First name:

Last name:

Submit

Get Account Information Form

Account No.:

Account branch:

Submit

Register for Newsletter Form

First name:

Last name:

Email address:

Submit

Track all forms in the HTML page

The following code tracks all of the form tags in the HTML page:

```
ac('forms:track');
```

or

```
ac('forms:track', 'form');
```

Track by element IDs

The following code tracks only the Get Account Information form:

```
ac('forms:track', '#get-account-information');
```

The following code tracks both the Get Account Information Form and Get Name Form:

```
ac('forms:track', '#get-account-information, #get-name');
```

Track by element classes

The following code tracks all forms with the `target-class` class. Both the Get Account Information form and Register for Newsletter form are tracked:

```
ac('forms:track', '.target-class');
```

Track by a combination of element IDs and element classes

The following code tracks a form with the `get-name` ID and all forms with the `target-class` class. As a result, all forms shown in the sample are tracked.

```
ac('forms:track', '#get-name, .target-class');
```

Manage form field data

To manage how form field data is sent with form submission and abandonment events, use the `captureFormDataOnAbandon` and `captureFormDataOnSubmit` options with the `forms:track` method. Set these options to `true` or `false`, depending on the result that you want to obtain.

For example, to record a form abandonment event, but exclude any form field data, set `captureFormDataOnAbandon` to `false`.

```
ac('forms:track', '.hs-form', {
  captureFormDataOnAbandon: false
});
```

Important

If you do not set the `captureFormDataOnAbandon` option, or you set it to anything other than `false`, the option assumes that the value is `true`, and the recorded form abandonment event will contain serialized form data. This data is subject to custom transformation; sensitive fields are excluded. For more information, see [Transform data before sending it](#) and [Sensitive form fields that are never tracked](#).

To capture serialized form data for form submission events, set the `captureFormDataOnSubmit` option to `true`.

You can set both options at the same time, as shown in the following example.

```
ac('forms:track', '.hs-form', {
  captureFormDataOnAbandon: false,
  captureFormDataOnSubmit: true
});
```

Transform data before sending it

To configure the format of the data that is sent with the form submission and abandonment events, specify a custom `transform` function in the `forms:track` options. This function receives an object with the form data; it should return an object with the data to send with the form events. You can exclude certain fields, rename fields that are not meaningful, or make any other transformations to the form data before it is sent. The data object is a JSON object of "name-value" pairs, where:

-
- "name" is the value of the element's ID or name attribute
 - "value" is the visitor's input

```
ac('forms:track', 'form', {  
  transform: function (data) {  
    return {  
      deliveryOption: data['radioButton_3'],  
      acceptedLicense: data['checkbox_license']  
    }  
  }  
});
```

Sensitive form fields that are never tracked

Important

Passwords, hidden fields, and sensitive fields are never tracked.

To denote sensitive fields, use the following regular expression, after removing non-alphanumeric characters from the field name.

```
/pass|billing|creditcard|cardnum|^cc|ccnum|exp|seccode|securitycode|securitynum|cvc|cvv|ssn|socialsec|socsec|csc/i.
```

If we track a field that you consider to be sensitive, please contact customercare@genesys.com.