



# Journey JavaScript SDK

autotrackURLChange

---

## Contents

- 1 Description
- 2 isUrlChange
  - 2.1 Example
- 3 onUrlChange
- 4 callback
  - 4.1 Example
- 5 delay
  - 5.1 Example

---

Learn how to customize tracking for Single Page Application (SPA) websites. Alternatively, use your preferred tag manager to customize and deploy the SPA tracking snippet. For more information, see [About event tracking with tag managers](#). If you are just getting started, read [Tracking snippet](#).

**Related pages:**

- 
- 
- 
- 
- 
- 

## Description

The `autotrackURLChange` module tracks activity on an SPA webpage when a user clicks through relative links or when software-driven activity changes the URL or browser history.

### Important

The `autotrackURLChange` module is automatically loaded when you load the SPA snippet.

To customize how `autotrackURLChange` tracks user activity, use the following options:

- `isUrlChange`
- `onUrlChange`
- `callback`
- `delay`

## isUrlChange

**Description:** Checks whether the URL changed since the last check.

**Type:** Function

**Status:** Default implementation available; can be overwritten

---

**Returns:** Boolean

**Arguments:** See the following table.

Name	Description	Type	Status
oldUrl	Previous URL before the change.	string	required
newUrl	Possibly changed URL.	string	required

## Example

This example shows how to exclude tracking when a visitor clicks relative links on a webpage.

```
ac('load', 'autotrackUrlChange', {
  isUrlChange: function(oldUrl, newUrl) {
    if (oldUrl !== newUrl) {
      if (oldUrl !== '' && newUrl.includes(oldUrl)) {
        return false;
      }
      return true;
    }
    return false;
  }
}, function() { });
```

## onUrlChange

**Description:** Runs when the URL changes through a relative link or SPA routing functionality.

**Type:** Function

**Status:** Default implementation available; tracks pageviews through `ac('pageview')`; can be overwritten

**Returns:** Void

**Arguments:** See the following table.

Name	Description	Type	Status
newURL	URL that the visitor changed to.	string	required

### Important

By default, the tracking snippet tracks page views via `ac('pageview')`. If you override the default behavior using `onUriChange`, remember to include this call in the new function manually.

---

## callback

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

### Example

Your SPA page may use routers or relative links to change the page URL without changing the page title. In this case, the default `ac('pageview')` call tracks all page views as occurring on the same page. Visit journey information that appears in Live Now and the agent UI does not reflect actual visitor behavior. To change this behavior, customize the `onUrlChange` option as shown in the following example.

```
ac('load', 'autotrackUrlChange', {
  onUrlChange: function(newUrl) {
    let customTitle = document.title;
    if (newUrl.includes('marketing')) {
      customTitle = customTitle + 'marketing';
    }
    if (newUrl.includes('contact-us')) {
      customTitle = customTitle + 'contact-us';
    }
    // supply pageOverrides with location and custom title field
    ac('pageview', {
      location: window.location.href,
      title: customTitle
    });
  }
}, function() { });
```

## delay

Use this option to set a delay between the time when the SDK realizes there is a URL change and when it sends the `onUrlChange` function.

### Example

This example sets a 1-second delay to give the application time to change the page title to match the current URL.

```
ac('load', 'autotrackUrlChange', { delay: 1000 });
```