

CardioLog SaaS Client-Side API for SharePoint

The CardioLog SaaS Tracking Agent offers a client-side API to send custom events using JavaScript, such as button clicks, banner clicks, navigation links or interactions with other UI components that display content dynamically (without redirecting to the actual content page URL).

- [The sendEvent JavaScript Function](#)
- [Monitoring Internal Traffic Sources](#)

The sendEvent JavaScript Function

You can call the `sendEvent` JavaScript function on **any monitored page in SharePoint** (which includes the CardioLog SaaS Tracking Agent) to send events to CardioLog SaaS via the client browser.

```
/**
 * @method sendEvent
 * @param {JSON object} data
 * @param {Function} [optional] function to be executed after the event is sent
 */
CardioLogAgent.API.sendEvent(data, eventCallback);
```

The `sendEvent` function accepts the following parameters:

- **data** - Required. The event data in JSON format. For example:

```
{
  "e": "Custom event",
  "u": "http://www.intlock.com/somepage",
  "metaData": {
    "price": 5,
    "name": "book"
  }
}
```

e - Mandatory. Event name.

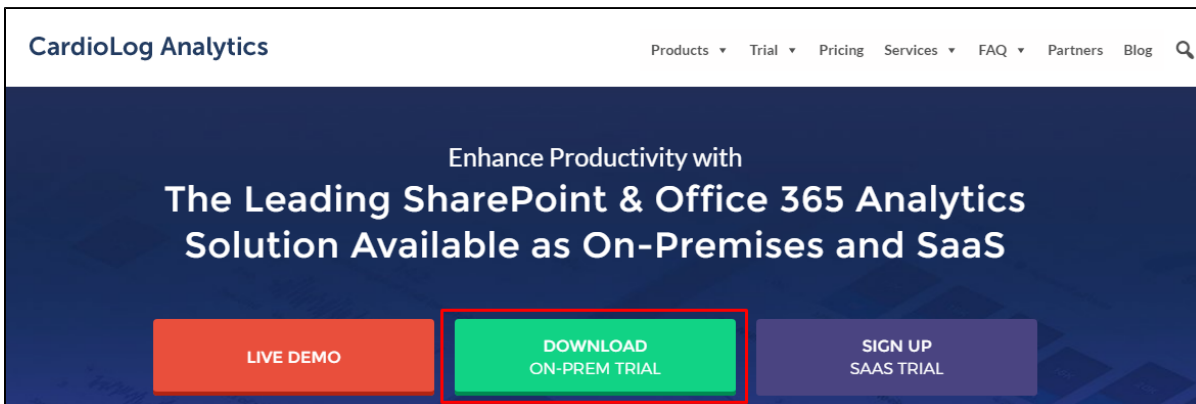
u - Optional. The URL path where the event occurred on or derived from.

metaData - Optional. A list of parameters and their values.

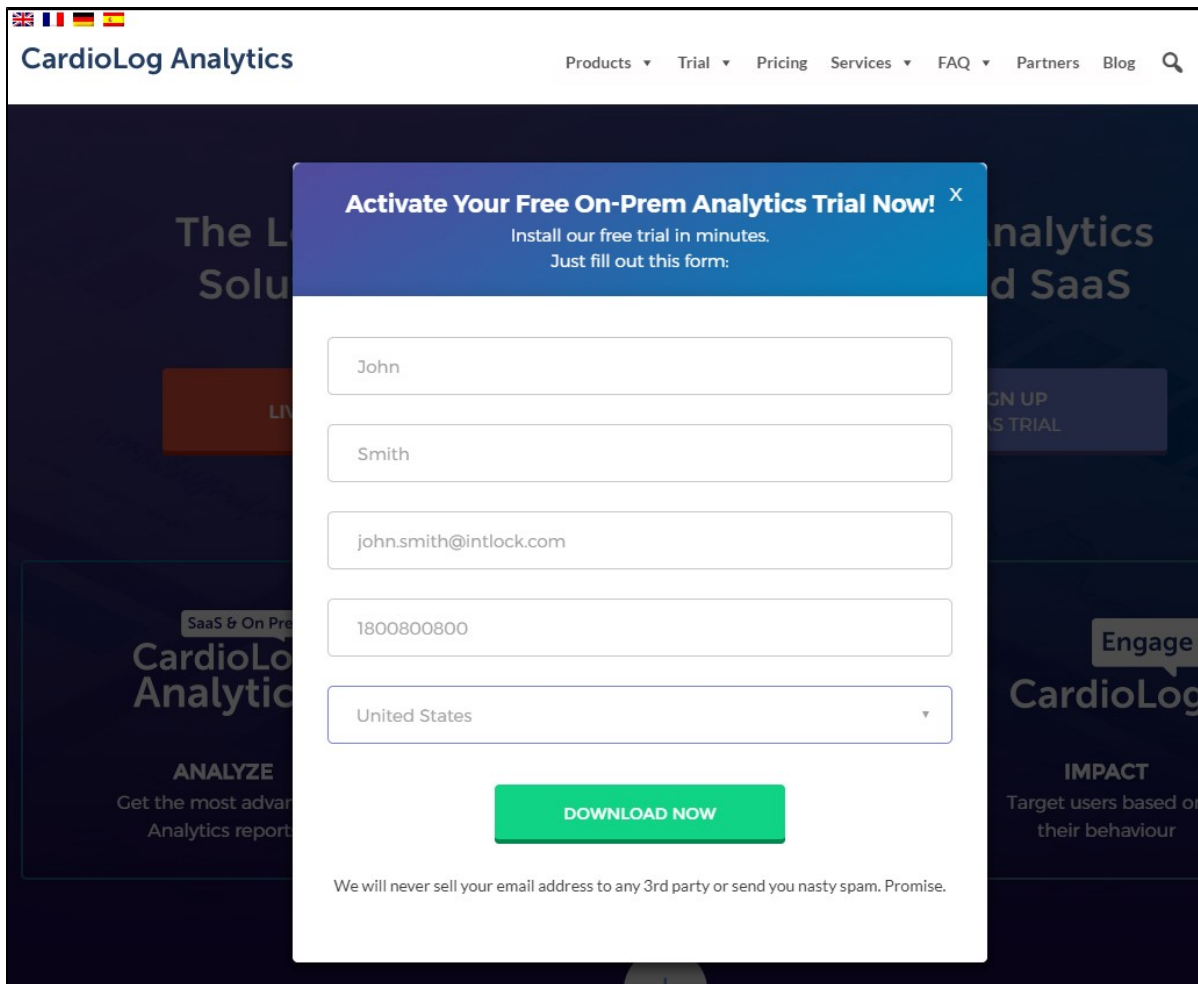
- **eventCallback** - Optional. Function to be executed after the event is sent.

Code Sample

In this example, when a user clicks on the DOWNLOAD button on the home page, a popup window is opened with the download form:



Download Button



Download Form

In order to send a click event on the *DOWNLOAD* button with the form fields values, add a call to the *sendEvent* function to the button onclick event as seen below:

```
<input type="button" value="DOWNLOAD" onclick="openDownloadForm();  
CardioLogAgent.API.sendEvent({e: 'Button Click', u: 'http://www.intlock.com/download/form.aspx', metaData: {name: 'DOWNLOAD NOW',  
firstName: 'John', lastName: 'Smith', email: 'john.smith@intlock.com', phone:1800800800, country: 'USA'}}, function () {console.log('event  
sent successfully');});" />
```

When clicking on the button, a custom event of type "Button Click" will be sent, indicating that the download button has been clicked on by the user. An "event sent" message will appear in the browser console.

In order to confirm that the event was sent successfully, open the browser Network tab and confirm that the request to the CardioLog Analytics SaaS events service (/cardiolog/api/client/events) returned the 200 status code.

Note: It is highly recommended to use the following code (or similar) to ensure that all the calls to the CardioLog API sendEvent function are executed and no events are lost (in case the CardioLog API object is not yet loaded on the page).



```

(function (CardioLogAgent) {
  if (CardioLogAgent.API) {
    return;
  }

  var WAIT_FOR_SEC = 10;
  var WAIT_CHECK_INTERVAL_MSEC = 500;

  function createSendEventProxy() {
    var proxy = {
      calls: [],

      fn: function (data, callback) {
        proxy.calls.push({ data: data, callback: callback });
      }
    };

    return proxy;
  }

  var sendEventProxy = createSendEventProxy();
  var totalAttemptsCount = Math.ceil(WAIT_FOR_SEC * 1000 / WAIT_CHECK_INTERVAL_MSEC);

  function waitForSendEventInit(attemptNumber) {
    if (attemptNumber >= totalAttemptsCount) {
      return;
    }

    window.setTimeout(function () {
      if (CardioLogAgent.API.sendEvent === sendEventProxy.fn) {
        waitForSendEventInit(attemptNumber + 1);
      } else {
        var call;
        while ((call = sendEventProxy.calls.shift())) {
          CardioLogAgent.API.sendEvent(call.data, call.callback);
        }
      }
    }, WAIT_CHECK_INTERVAL_MSEC);
  }

  CardioLogAgent.API = { sendEvent: sendEventProxy.fn };
  waitForSendEventInit(0);
})(window.CardioLogAgent || (window.CardioLogAgent = {}));

```

Monitoring Internal Traffic Sources

To monitor SharePoint content that is accessed via external locations, you can use the **referrer** parameter whenever you publish a link to SharePoint and you wish to track the source of the referring application or page.

This is most commonly used in email or social networks such as Teams or Yammer.

For example, if you have included an internal link in a promotional email add the **referrer** parameter to track the users who visited your site via this link. Define a unique referrer value that will be displayed in reports, such as **referrer=email_promotion_Dec_2020** in order to display the link accordingly:

```

https://intlock.sharepoint.com/sites/blogs/ceo_blog.aspx?referrer=email_promotion_Dec_2020

```

To see how many visitors arrived at your page through your email campaign, filter the Usage Overview report by the **QueryString** field which stores all URL parameters.

