

Simple Analytics

This pdf is generated on April 15, 2021. Go to [the documentation page](#) for the most up-to-date version.

Automated events

[Normal events](#) need to be added into your code. To make it easier we created a separate automated events script. This script has to be installed once, and it will track outbound links, email addresses clicks, and amount of downloads for common files (pdf, csv, docx, xlsx). Events will appear on [your events page](#).

Outbound

Outbound links are basically just links that go to a different website. For example you have a website with a few products listed on them. People pay you for listing those products on your website. Now you want to find a way to track the amount of visitors that go to that website. For this you'll need outbound events!

It's as easy as including the script below and making sure the `outbound` in config is set to `true`.

Emails

Do you have links to email addresses on your website? Then it could make sense to see how many clicks you get on your `mailto` links.

It's as easy as including the script below and making sure the `emails` in config is set to `true`.

Downloads

Do you have links to pdf's or other files? Collect the amount of downloads. You can specify the extensions of the files your want to collect downloads for.

It's as easy as including the script below and making sure the `downloads` in config is set to `true`. Feel free to add extensions to the list.

Script

This script is additional to the normal embed script. Make sure to also include [that script](#).

If you are a developer or you know your way around HTML it's easy to embed this script on your website. If you need help with that, we are happy to help. Just [contact us](#).

```

<script> (function(window) { var options = { // What to collect outbound: true,
emails: true, downloads: true, // Downloads: enter file extensions you want to
collect downloadsExtensions: ["pdf", "csv", "docx", "xlsx"], // All: use title
attribute if set for event name (for all events) // THIS TAKES PRECEDENCE OVER
OTHER SETTINGS BELOW title: true, // Outbound: use full URL of the links? false
for just the hostname outboundFullUrl: false, // Downloads: if taking event
name from URL, use full URL or just filename (default) downloadsFullUrl: false,
}; !function(u,t){var d,c;void 0!==u&&(d=function(t,e){return("warn"===e?
console.warn:console.log)("Simple Analytics automated events: "+t)},void 0===
(c=t)&&d("options object not found, please
specify", "warn"),u.saAutomatedLink=function(t,e){try{if(!t)return d("no element
found");function n(){o||t.hasAttribute("target")||
(document.location=t.getAttribute("href")),o=!0}var o=!1;if(u.sa_event){var
a=t.hostname,s=t.pathname,i=!1;if(c.title&&t.hasAttribute("title")&&("!"=
(l=t.getAttribute("title").trim())&&(i=!0)),i)r=l;else switch(e)
{case"outbound":r=a+(c.outboundFullUrl?
s:"");break;case"download":r=c.downloadsFullUrl?
a+s:s.split("/").pop();break;case"email":var r=
(t.getAttribute("href").split(":")[1]||"").split("?")[0]}var
l=e+"_" +r.replace(/[^\a-z0-9]/gi,"_").replace(/(^_|_+$/g,"");return
sa_event(l,n),d("collected "+l),u.setTimeout(n,5e3)}return d("sa_event is not
defined", "warn"),n()}catch(t)
{d(t.message, "warn")}},u.addEventListener("DOMContentLoaded", function()
{try{for(var t=document.getElementsByTagName("a"),e=0;e<t.length;e++){var
n=t[e],o=!1;n.getAttribute("onclick")||
(c.downloads&&/^https?:\/\//i.test(n.href)&&new RegExp("."+
(c.downloadsExtensions||[]).join("|")+"), "i").test(n.pathname)?
o="download":c.outbound&&/^https?:\/\//i.test(n.href)&&n.hostname!==u.location.
hostname?o="outbound":c.emails&&/^mailto:/i.test(n.href)&&(o="email"),o&&
(o="saAutomatedLink(this,
'+o+'");",n.hasAttribute("target")&&"_self"!==n.hasAttribute("target")||
return false;"),n.setAttribute("onclick",o))}}catch(t)
{d(t.message, "warn")}}))})(window,options); })(window); </script>

```

The script is compressed with the public tool jcompress.com.

The non-minified version of the above script

```

<script> (function saAutomatedEvents(window) { // Skip server side rendered
pages if (typeof window === "undefined") return; var options = { // What to
collect outbound: true, emails: true, downloads: true, // Downloads: enter file
extensions you want to collect downloadsExtensions: ["pdf", "csv", "docx",
"xlsx"], // All: use title attribute if set for event name (for all events) //
THIS TAKES PRECEDENCE OVER OTHER SETTINGS BELOW title: true, // Outbound: use
full URL of the links? false for just the hostname outboundFullUrl: false, //
Downloads: if taking event name from URL, use full URL or just filename
(default) downloadsFullUrl: false, }; var log = function (message, type) { var
logger = type === "warn" ? console.warn : console.log; return logger("Simple
Analytics automated events: " + message); }; // For minifying the script var
optionsLink = options; if (typeof optionsLink === "undefined") log("options
object not found, please specify", "warn"); window.saAutomatedLink = function
saAutomatedLink(element, type) { try { if (!element) return log("no element
found"); var sent = false; var callback = function () { if (!sent &&
!element.hasAttribute("target")) document.location =
element.getAttribute("href"); sent = true; }; if (window.sa_event) { var
hostname = element.hostname; var pathname = element.pathname; var useTitle =
false; if (optionsLink.title && element.hasAttribute("title")) { var theTitle =
element.getAttribute("title").trim(); if (theTitle !== "") useTitle = true; }
var event; if (useTitle) { event = theTitle; } else { switch (type) { case
"outbound": { event = hostname + (optionsLink.outboundFullUrl ? pathname : "");
break; } case "download": { event = optionsLink.downloadsFullUrl ? hostname +
pathname : pathname.split("/").pop(); break; } case "email": { var href =
element.getAttribute("href"); event = (href.split(":")[1] || "").split("?")[0];

```

```

break; } } } var clean = type + "_" + event.replace(/[^\a-z0-9]+/gi,
"_").replace(/(^_|_+$)/g, ""); sa_event(clean, callback); log("collected " +
clean); return window.setTimeout(callback, 5000); } else { log("sa_event is not
defined", "warn"); return callback(); } } catch (error) { log(error.message,
"warn"); } }; function onDOMContentLoaded() { try { var a =
document.getElementsByTagName("a"); // Loop over all links on the page for (var
i = 0; i < a.length; i++) { var link = a[i]; var collect = false; // We don't
want to overwrite website behaviour so we check for the onclick attribute if
(!link.getAttribute("onclick")) { // Collect download clicks if (
optionsLink.downloads && /^https?:\/\//i.test(link.href) && new RegExp( "\.(" +
(optionsLink.downloadsExtensions || []).join("|") + ")", "i"
).test(link.pathname) ) { collect = "download"; // Collect outbound links
clicks } else if ( optionsLink.outbound && /^https?:\/\//i.test(link.href) &&
link.hostname !== window.location.hostname ) { collect = "outbound"; // Collect
email clicks } else if (optionsLink.emails && /^mailto:/i.test(link.href)) {
collect = "email"; } if (collect) { var onClickAttribute =
"saAutomatedLink(this, " + collect + "');" if ( !link.hasAttribute("target")
|| link.hasAttribute("target") === "_self" ) onClickAttribute += " return
false;"; link.setAttribute("onclick", onClickAttribute); } } } } catch (error)
{ log(error.message, "warn"); } } window.addEventListener("DOMContentLoaded",
onDOMContentLoaded); })(window); </script>

```

You can compress above code (after changing the options) via jscompress.com.

When you test with this script, note that it uses the `DOMContentLoaded` event. This event only fires when the script has been added before the page has loaded. If you want to test the script in your console make sure to replace

```

window.addEventListener("DOMContentLoaded", onDOMContentLoaded); with
onDOMContentLoaded().

```

It takes care of sending it as an event to Simple Analytics. As events can only be letters, numbers, and underscores we already convert this name.

We love to improve our automated events. Please let us know if you need any help! We don't mind getting our hands dirty.

This pdf is generated on April 15, 2021. Go to [the documentation page](#) for the most up-to-date version.