

神策分析

Web 端数据接入方案

目录

Web端 数据接入	3
1. 数据接入	3
1.1 获取和引入 Sensors Analytics SDK	3
1.2 页面浏览事件追踪	6
1.3 自定义事件追踪	6
1.4 关闭 元素点击和视觉停留事件	7
2. 设置用户属性	7
2.1 sa.setProfile(properties[, callback])	8
2.2 sa.setOnceProfile(properties[, callback])	8
3 事件公共属性	9
3.1 sa.registerPage(object)	9
3.2 sa.register(object)	9
3.3 sa.registerOnce(object)	9
4 用户ID关联	9
4.1 用户标识说明	9
4.2 用户关联	10

Web端 数据接入

1. 数据接入

相关文档链接：https://sensorsdata.cn/manual/js_sdk.html

1.1 获取和引入 Sensors Analytics SDK

首先从 Sensors Analytics 的主页中，进入数据导入向导页面：

然后在右上角点击 数据接入 显示如下界面，然后点击 生成导入代码，进入代码生成页面，按需选择合适的选项，然后点击生成代码，如下图



JavaScript SDK 自动生成工具

查看文档 >>

网站是否开启了 HTTPS

☒ 否 ☐ 是

是否开启自动采集

☐ 禁用

☒ 仅采集页面浏览事件
每次打开页面会发送 \$pageview 事件，并针对新用户增加一些用户属性，详情 [点击这里 >>](#)

☐ 采集页面浏览和交互元素的点击
除了采集页面浏览事件以外，还会自动采集页面内 html 标签为 a button input 的点击事件 [详细配置信息 >>](#)

SDK 文件位置

☒ 使用神策提供的 CDN
稳定性一般，国外访问延时较高。只建议测试使用，不推荐在正式环境使用

☐ 从 [GitHub](#) 下载并放到站点服务器上
一般情况下下载最新版本即可

输入你们网站目录下 sensorsdata.min.js 的文件地址

输入 SDK 的全局变量名字

sa

如定义成 sa 的话，后面接可以使用 sa.track() 用来调用方法

展开高级配置

生成

请点击生成按钮

点击生成，将代码片段放入网页中的合适位置即可。

异步载入代码片段 如下：（注：将代码放入 html 的 head 里面,一般最好放在稍微靠前点的位置。您在 sa.track() 时，只要保证写在上面引用的代码的下面就可以，不需要等 window.onload 后再执行。）

```

<script>
(function(para) {
    var p = para.sdk_url, n = para.name, w = window, d =
document, s = 'script', x = null, y = null;
    w['sensorsDataAnalytic201505'] = n;
    w[n] = w[n] || function(a) {return function() {(w[n]._q =
w[n]._q || []).push([a, arguments]);}};
    var ifs =
['track', 'quick', 'register', 'registerPage', 'registerOnce', 're
gisterSession', 'registerSessionOnce', 'trackSignup',
'trackAbtest', 'setProfile', 'setOnceProfile', 'appendProfile',
'incrementProfile', 'deleteProfile', 'unsetProfile',
'identify'];
    for (var i = 0; i < ifs.length; i++) {
        w[n][ifs[i]] = w[n].call(null, ifs[i]);
    }
    if (!w[n]._t) {
        x = d.createElement(s), y = d.getElementsByTagName(s)
[0];
        x.async = 1;
        x.src = p;
        y.parentNode.insertBefore(x, y);
        w[n]._t = 1 * new Date();
        w[n].para = para;
    }
})();({
    sdk_url: '在 github 下载新版本的 sensorsdata.min.js ',
    name: 'sa',
    server_url: '数据接收地址'
});
sa.quick('autoTrack'); //神策系统必须是1.4最新版及以上
</script>

```

如果使用 从 npm 获取， npm install sa-sdk-javascript，这里获取后建议使用 webpack 等工具来打包。这里获取到的是最新版本。

```

import sa from 'sa-sdk-javascript';
sa.init({
    server_url: '...',
    .....
});
sa.login(user_id);
sa.quick('autoTrack');

export default sa;

```

1.2 页面浏览事件追踪

相关文档链接：https://www.sensorsdata.cn/manual/js_sdk.html#33-追踪页面跳转

为了方便使用，Sensors Analytics 的 SDK 也提供了一些默认事件的收集，例如如果需要采集页面浏览事件（即 PageView），可以通过增加如下代码：

```
sa.quick('autoTrack')
```

设置之后，SDK 就会自动收集页面浏览事件，以及设置初始来源。

\$pageview 事件的预置属性

字段名称	类型	说明	版本
\$referrer	字符串	前向地址	
\$referrer_host	字符串	前向域名	
\$url	字符串	页面地址	
\$url_path	字符串	页面路径	
\$title	字符串	页面标题	
\$is_first_time	布尔值	是否首次访问	1.7修改
\$utm_source	字符串	广告系列来源	
\$utm_medium	字符串	广告系列媒介	
\$utm_term	字符串	广告系列字词	
\$utm_content	字符串	广告系列内容	
\$utm_campaign	字符串	广告系列名称	

1.3 自定义事件追踪

相关文档链接：https://www.sensorsdata.cn/manual/js_sdk.html#3-自定义事件追踪

Sensors Analytics SDK 初始化成功后，即可以通过 `sa.track(event_name[, properties][, callback])` 记录事件：

- `event_name`: string, 必选。表示要追踪的事件名。
- `properties`: object, 可选。表示这个事件的属性。
- `callback`: function, 可选。表示已经发送完数据之后的回调。

以 搜索 为例 需要在 搜索的点击函数中 调用这个自定义方法

```
sa.track('search',{
  keyWord:"具体值",           //string 搜索关键词
  hasResult:"具体值",         //bool 是否有结果
  useHistoryKeyword:"具体值", //bool 是否使用历史词
  useRecommendKeyword:"具体值", //bool 是否使用推荐词
});
```

1.4 关闭 元素点击和视觉停留事件

```
heatmap: {
  // 关闭 $WebStay (Web 视区停留) 事件
  scroll_notice_map: 'not_collect',
  // 关闭 $WebClick (Web 元素点击) 事件
  clickmap: 'not_collect'
}
```

2. 设置用户属性

相关文档链接：https://sensorsdata.cn/manual/js_sdk.html#5-设置用户属性

为了更准确地提供针对人群的分析服务，可以使用 Sensors Analytics SDK 的 `profileSet()` 等方法设置用户属性，如年龄、性别等。用户可以在留存分析、分布分析等功能中，使用用户属性作为过滤条件，精确分析特定人群的指标。

你可以首先使用上面的 `identify(userid)` 来标识用户，然后可以对这个特定的用户设置一些属性。如果你不用这两个方法去标识用户，就会以 Cookie 中的匿名 `distinct_id` 来标识用户。

2.1 sa.setProfile(properties[, callback])

直接设置用户的属性，如果存在则覆盖。

- `properties`: `object`，必选。
- `callback`: `function`，可选。

```
sa.setProfile({
  sex: "男",           //string
  birthday: "2015-06-19", //number
  phone: "13111111111", //string
  email: "xxx@wwwwww.com" //string
});
```

建议前端采集比较方便的可以前端设置用户属性，后端也可以设置。

2.2 sa.setOnceProfile(properties[, callback])

如果不存在则设置，存在就不设置。

- `properties`: `object`，必选。
- `callback`: `function`，可选。

```
sa.setOnceProfile({email: 'xxx@xx'});
```


3 事件公共属性

https://www.sensorsdata.cn/manual/js_sdk.html#32-事件公共属性

特别地，如果某个事件的属性，在所有事件中都会出现，可以通过 register 系列方法将该属性设置为事件公共属性。

3.1 sa.registerPage(object)

在当前页面的，后续事件中都注入这些属性，只针对当前页面有效。

3.2 sa.register(object)

在 Cookie 中永久保存属性，永久有效，如果存在这个属性了则覆盖。

```
sa.register({  
    platform:"h5"           //string 平台类型  
})
```

```
sa.track('button_A_click'); // 这时候，这个 button_A_click 事件，就会带有  
平台类型这个属性。永久有效，如果存在了则覆盖。
```

3.3 sa.registerOnce(object)

在 Cookie 中永久保存属性，如果存在这个属性了则不覆盖。

4 用户ID关联

相关文档链接：https://www.sensorsdata.cn/manual/js_sdk.html#2-如何标识用户

4.1 用户标识说明

在进行任何埋点之前，都应当先确定如何标识用户。distinct_id 是用来标识用户的，是一段唯一的字符串，默认情况下 JavaScript SDK 会自动生成一个 distinct_id 并永久保存在浏览器中的 Cookie 中（我们暂时称这个为 cookie_id）。

4.2 用户关联

在登录和注册成功后，如果你已知了真实的用户 id，你可以通过 `sa.login(user_id)` 来修改当前的 `distinct_id`。

我们的 JavaScript SDK 从 1.6.9 版本开始支持 `sa.login()` 方法。

神策 id 体系，针对你们的情况，应该以企业账号为 `userid` 即，前端 用户注册登录的时候 `login`（“传入用户的企业 id”）。

后端 `track` 或者 `profile` 的时候，里面的 `id` 应为 企业 id，且 `is_login_id` 这个参数为 `true`。

```
sa.login('user1312312123');
```

获取 Cookie 中的 `distinct_id`：

前端方式：

可以通过 `sa.store.getDistinctId()` 方法获取到 `distinct_id`。

后端方式：

可以在 Cookie 里面找到 key 为 `sensorsdata2015jssdkcross` 的 value 值然后进行 `decodeURIComponent` 解码，最后通过 `JSON.parse` 方法得到一个对象，对象里面的 `distinct_id` 即为用户所需要的。