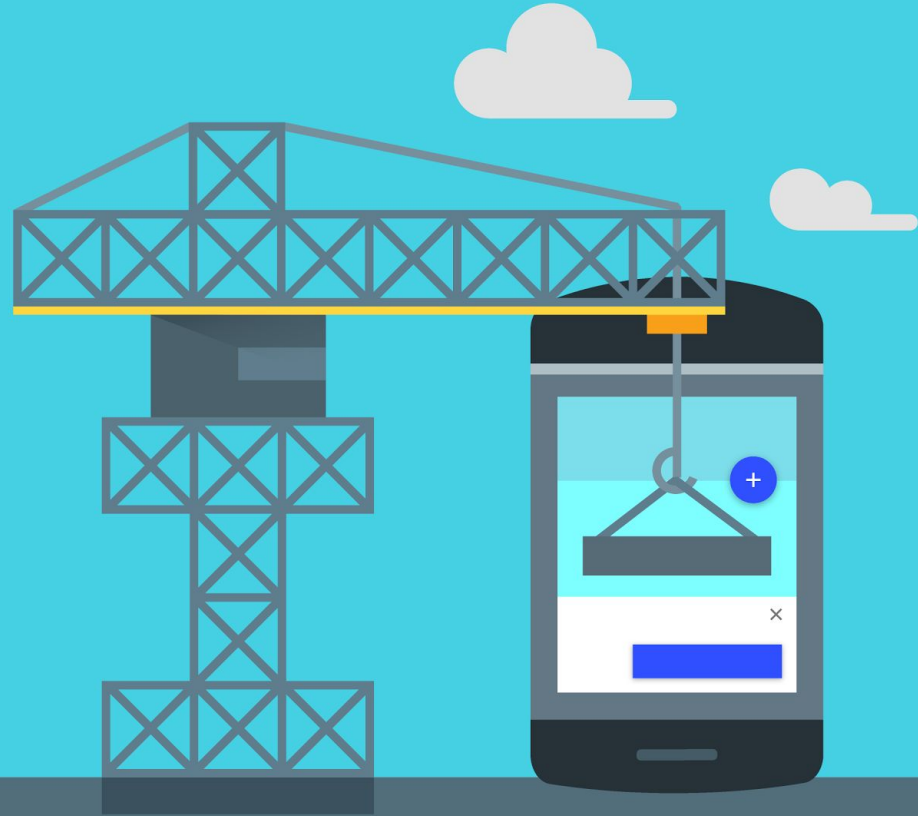


Integrating Google Analytics



What is Google Analytics?

What is Google Analytics

Data is
generated from
user behavior



Data is processed
by Google
Analytics back end



Reports are
generated

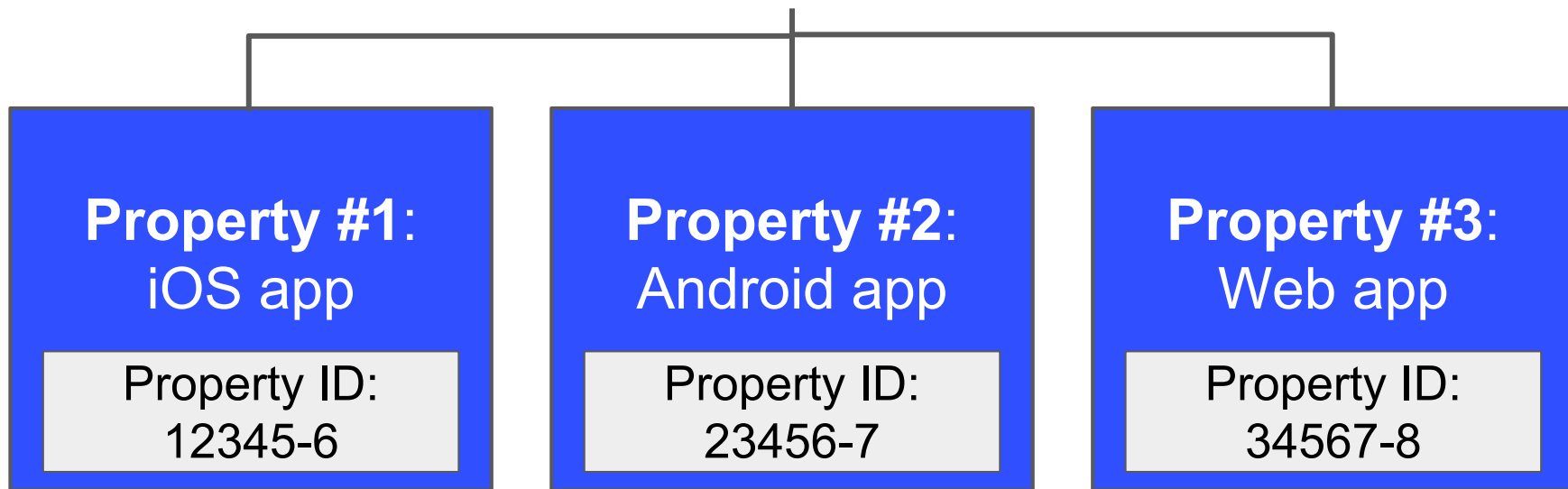
Integration process overview

1. Create account and “property”
2. Get tracking snippet
3. Paste tracking snippet
4. Add custom code

Accounts and properties

Accounts and properties

Account:
My Company



Creating an account

New Account

What would you like to track? _____

Website

Mobile app

Setting up your account _____

Account Name

Accounts are the top-most level of organization and contain one or more tracking IDs.

My New Account Name

Creating a property

Setting up your property

Website Name

My New Website

Website URL

http:// ▾

Example: http://www.mywebsite.com

Industry Category

Select One ▾

Reporting Time Zone

United States ▾

(GMT-08:00) Pacific Time ▾

Adding analytics to your app

Adding analytics to your app

```
<script>
```

```
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||  
function(){(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new  
Date();a=s.createElement(o),m=s.getElementsByTagName(o)[0];a.async  
=1;a.src=g;m.parentNode.insertBefore(a,m)})(window,document,'scr  
ipt','https://www.google-analytics.com/analytics.js','ga');
```

```
ga('create', 'UA-XXXXXXXX-Y', 'auto');
```

```
ga('send', 'pageview');
```

```
</script>
```

How the analytics code works

How analytics.js works

```
ga(command, [parameters]);
```

Examples:

1. `ga('create', 'UA-XXXXXXXX-Y', 'auto');`
2. `ga('create', {
 trackingId: 'UA-XXXXX-Y',
 cookieDomain: 'auto'
});`

Creating tracker objects

Default tracker

```
ga('create', 'UA-XXXXX-Y', 'auto');
```

Naming trackers

```
ga('create', 'UA-XXXXX-Y', 'auto', 'myTrackerName');
```

Setting tracker fields

```
ga('create', 'UA-XXXXX-Y', 'auto', 'myTrackerName', {  
  userId: '12345'  
});
```

Getting and setting tracker data

Ready callback

```
ga('create', 'UA-XXXXX-Y', 'auto');  
ga(function(tracker) {  
    console.log(tracker);  
});
```

Getting tracker fields

```
ga('create', 'UA-XXXXX-Y', 'auto');  
ga(function(tracker) {  
    console.log(tracker.get('clientId'));  
});
```

Setting tracker fields

```
ga('set', 'page', '/about');
```

```
ga('set', {  
  page: '/about'  
});
```


Sending data to Google Analytics

Sending tracker data

```
ga('create', 'UA-XXXXX-Y', 'auto');  
ga('send', 'pageview');
```

Alternatively

```
ga('send', {  
  [field]: [data],  
  ...  
});
```

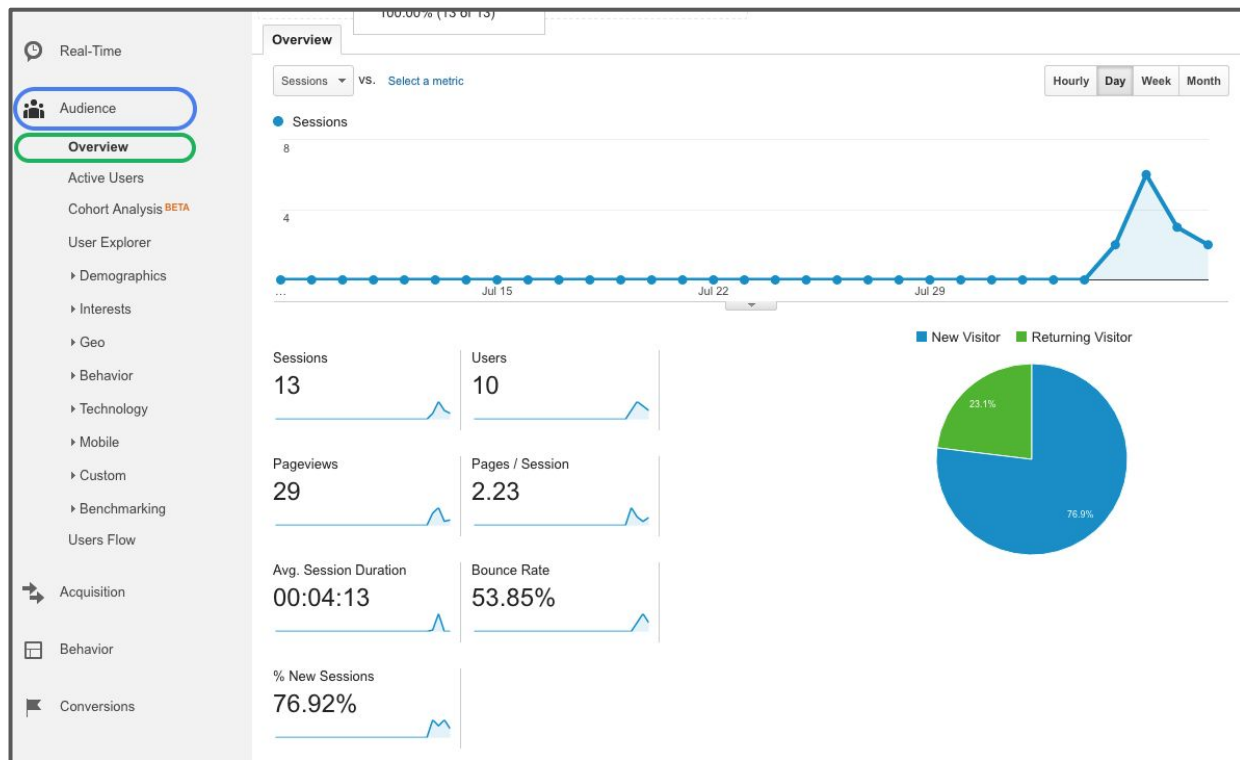
Knowing when data is sent

Hit callback

```
var form = document.getElementById('signup-form');
form.addEventListener('submit', function(event) {
  event.preventDefault();
  ga('send', {
    ...
    hitCallback: function() { form.submit(); }
  });
});
```

The Google Analytics Dashboard

The Google Analytics dashboard



The Google Analytics dashboard (cont.)

Demographics

Language

Country

City

System

Browser

Operating System

Service Provider

Mobile

Operating System

Service Provider

Screen Resolution

City

Sessions % Sessions

1. Mountain View

6



46.15%

2. (not set)

3



23.08%

3. San Mateo

1



7.69%

4. Gig Harbor

1



7.69%

5. Lakewood

1



7.69%

6. Vancouver

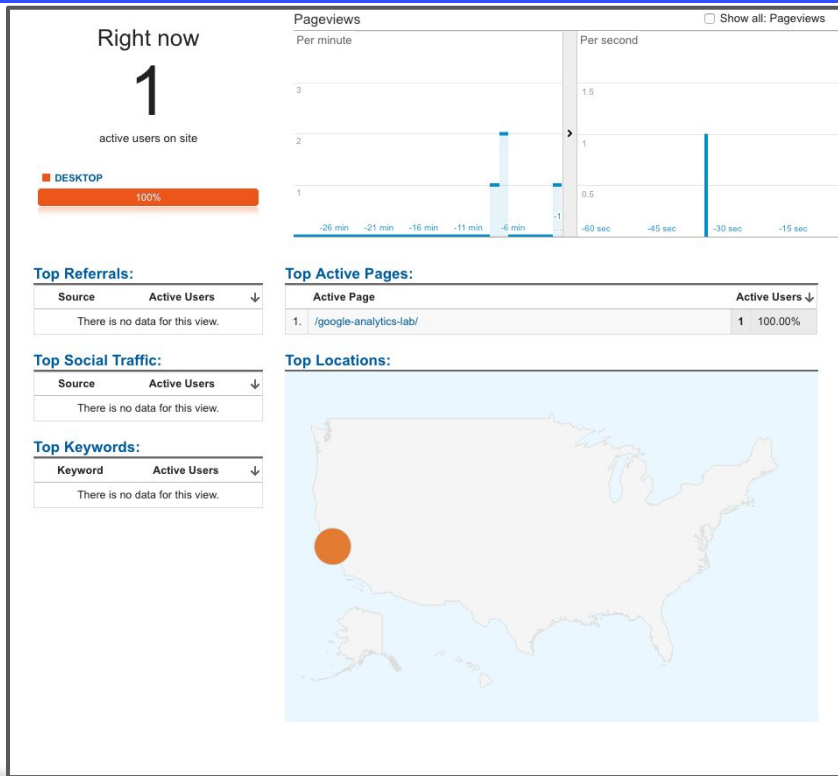
1



7.69%

[view full report](#)

Real-time analytics



Testing and debugging

Testing and debugging

```
<script>
```

```
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)})(window,document,'script','https://www.google-analytics.com/analytics\_debug.js','ga');
```

```
ga('create', 'UA-XXXXXXXX-Y', 'auto');
```

```
ga('send', 'pageview');
```

```
</script>
```


Adding custom events

Custom events

```
ga('send', {  
  hitType: 'event',  
  eventCategory: 'products',  
  eventAction: 'purchase',  
  eventLabel: 'Summer products'  
});
```

```
ga('send', 'event', 'products', 'purchase', 'Summer products');
```

Example: Push notifications

Use case: Checking for support

```
if (!('Notification' in window)) {  
  console.log('This browser does not support notifications!');  
  ga('send', 'event', 'notification', 'unsupported');  
  return;  
}
```

Example: Push notifications (cont.)

Use case: Subscriptions

```
registration.pushManager.subscribe({userVisibleOnly: true})  
.then(function(pushSubscription) {  
  ga('send', 'event', 'push', 'subscribe');  
})  
...
```

Example: Push notifications (cont.)

Use case: Subscriptions (cont.)

```
.catch(function(error) {  
  if (Notification.permission === 'denied') {  
    ga('send', 'event', 'push', 'subscribe-blocked');  
  } else {  
    ga('send', 'event', 'push', 'subscribe-error');  
  }  
});
```

Analytics and service workers

Analytics and service workers



Main thread

```
function openCommunication() {  
  var msgChannel = new MessageChannel();  
  msgChannel.port1.onmessage = function(msgEvent) {  
    sendAnalytics(msgEvent.data);  
  };  
  navigator.serviceWorker.controller.postMessage(  
    'Establishing contact with SW', [msgChannel.port2]  
  );  
}  
openCommunication();
```


Service worker

```
var mainClientPort;  
self.addEventListener('message', function(event) {  
  console.log('SW received message: ', event.data);  
  mainClientPort = event.ports[0];  
  console.log('Communication ports established');  
});
```

Service worker (cont.)

```
mainClientPort.postMessage({  
  eventCategory: 'push',  
  eventAction: 'received'  
});
```

Back in the main thread

```
function sendAnalytics(message) {  
  var eventCategory = message.eventCategory;  
  var eventAction = message.eventAction;  
  ga('send', 'event', eventCategory, eventAction);  
}
```

Back to push notifications

Use case: Push events

```
self.addEventListener('push', function(event) {  
  mainClientPort.postMessage({  
    eventCategory: 'push',  
    eventAction: 'received'  
  });  
  ...  
});
```

Back to push notifications (cont.)

Use case: Notification interactions

```
self.addEventListener('notificationclick', function(event)
{
  if (event.action === 'decline') {
    mainClientPort.postMessage({
      eventCategory: 'notification',
      eventAction: 'decline'
    });
  }
  ...
});
```

Using analytics offline

Offline analytics

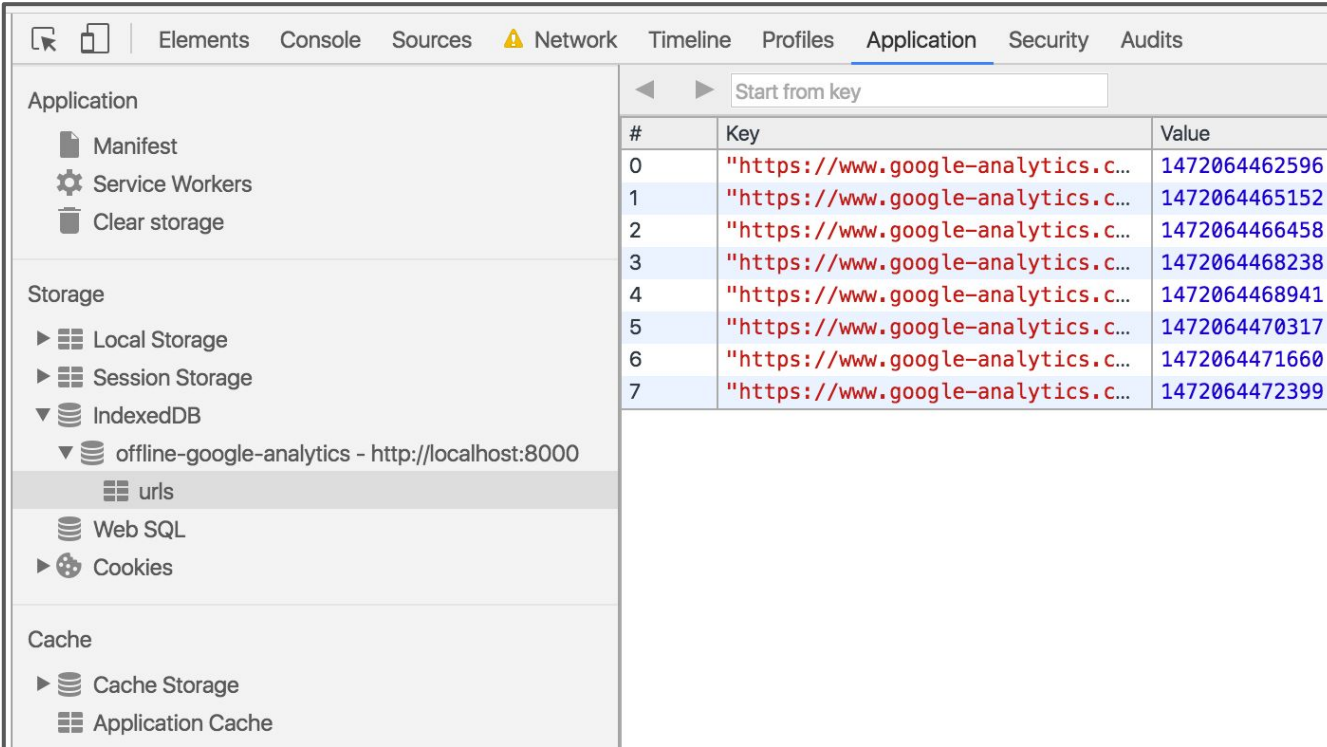
Install

```
$ npm install --save-dev sw-offline-google-analytics
```

Service worker

```
importScripts('path/to/offline-google-analytics-import.js');  
goog.offlineGoogleAnalytics.initialize();
```

Offline analytics (cont.)



The screenshot shows the Chrome DevTools Application tab. The left sidebar has three main sections: Application, Storage, and Cache. Under Application, there are links for Manifest, Service Workers, and Clear storage. Under Storage, there are links for Local Storage, Session Storage, IndexedDB, Web SQL, and Cookies. Under Cache, there are links for Cache Storage and Application Cache. The IndexedDB section is expanded, showing a database named 'offline-google-analytics - http://localhost:8000'. The 'urls' key is selected, and the right pane displays a table of data.

#	Key	Value
0	"https://www.google-analytics.c...	1472064462596
1	"https://www.google-analytics.c...	1472064465152
2	"https://www.google-analytics.c...	1472064466458
3	"https://www.google-analytics.c...	1472064468238
4	"https://www.google-analytics.c...	1472064468941
5	"https://www.google-analytics.c...	1472064470317
6	"https://www.google-analytics.c...	1472064471660
7	"https://www.google-analytics.c...	1472064472399

Additional Resources

- [Google Analytics Academy](#) (non-technical)
- [Google Analytics account signup](#)
- [Analytics for mobile applications](#)
- [Chrome debugger extension](#)

More examples

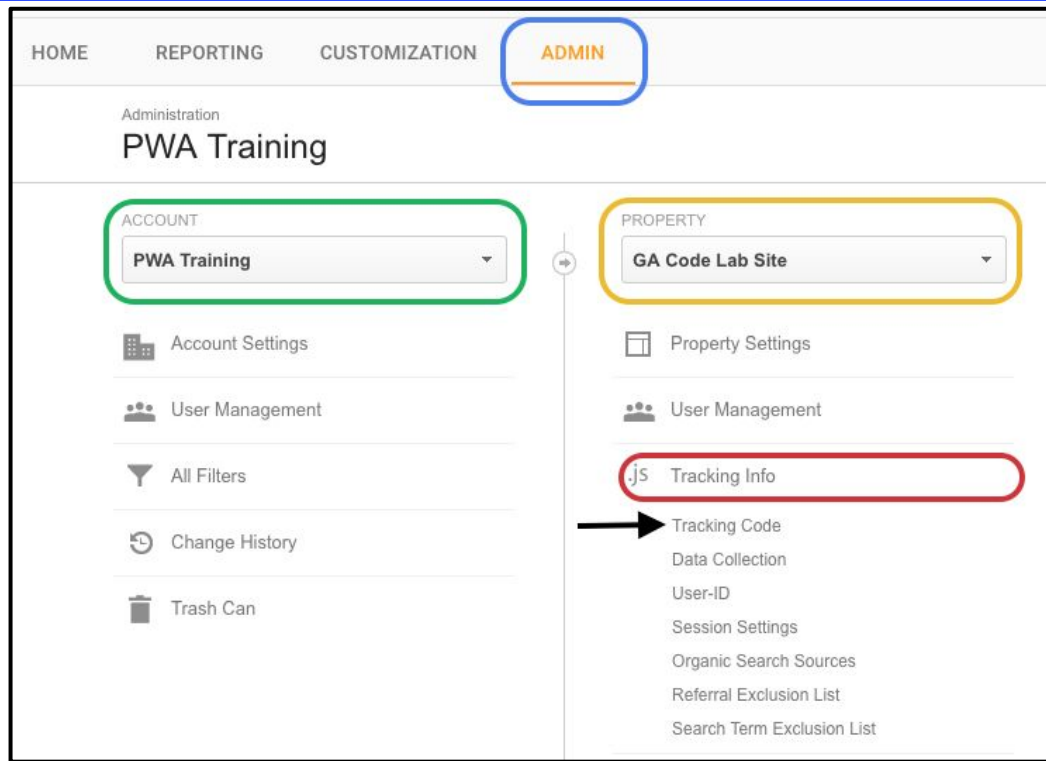
- [Using the User Timing API with Google Analytics code lab](#)
- [Communicating with service worker example](#)
- [Google I/O offline example](#)

Documentation

- [analytics.js documentation](#)
- [Reporting API](#)
- [MessageChannel](#)
- [ImportScripts](#)
- [Offline Google Analytics npm package](#)
- [IndexedDB](#)

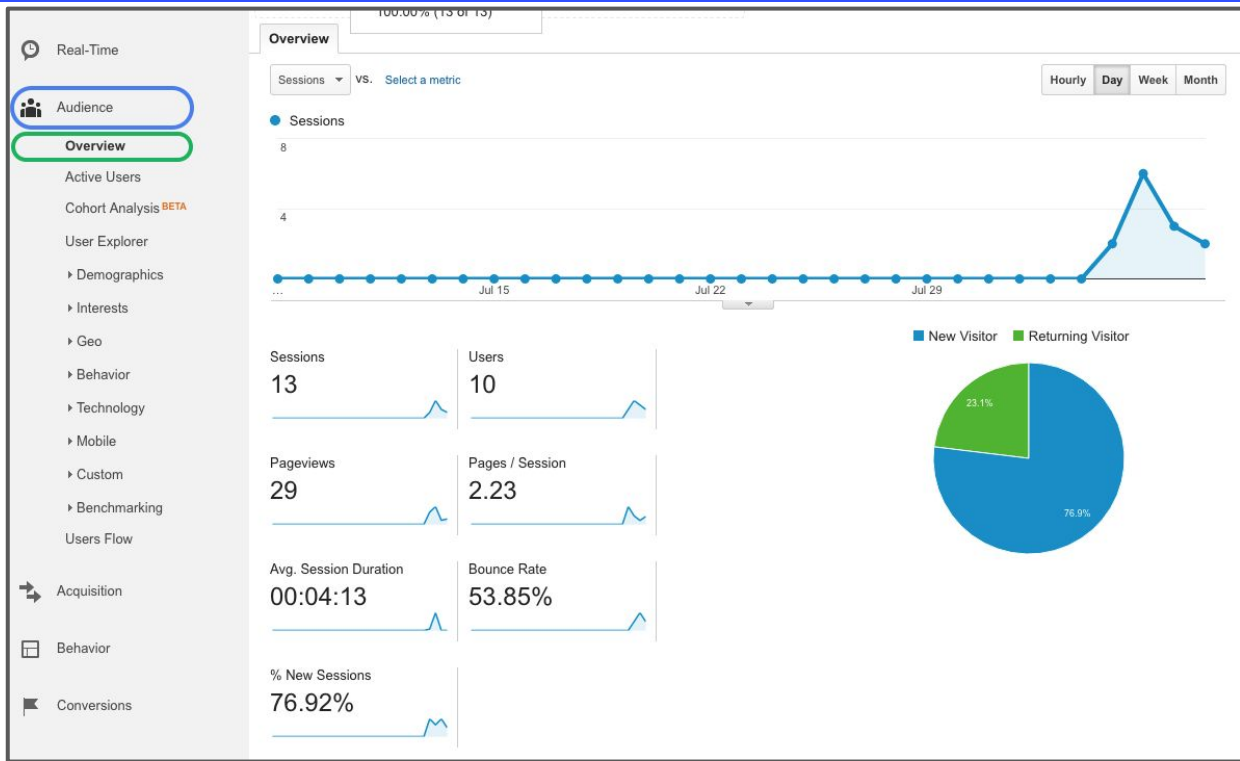
Finding your tracking snippet & ID

1. Select the Admin tab
2. Under “account”, select your account from the drop down list.
3. Then under “properties”, select your property from the down list.
4. Now choose “tracking info”, followed by “tracking code”.



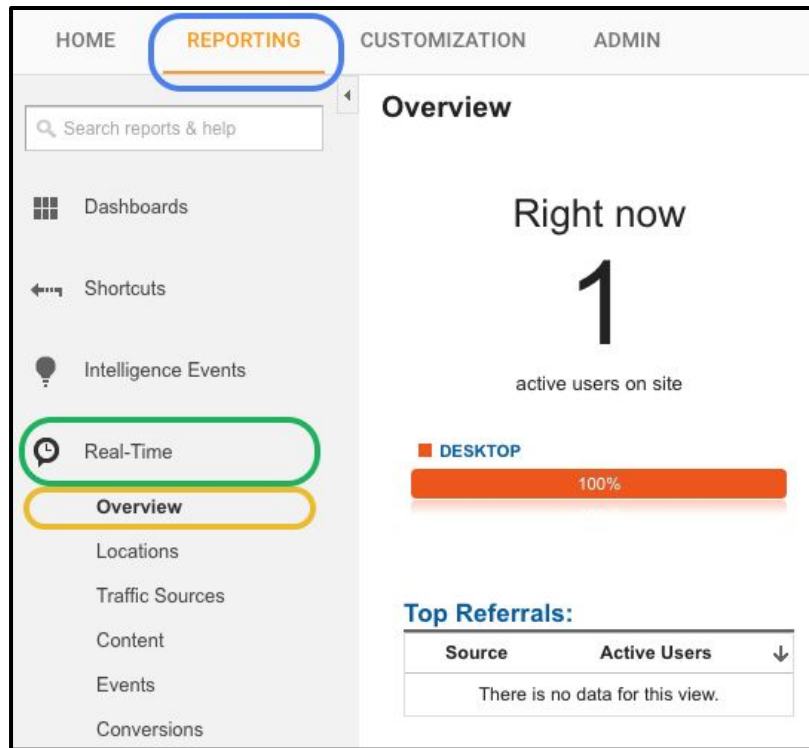
Finding the Audience Overview interface

1. Select the Reporting tab
2. Select Audience in the side panel
3. Select Overview



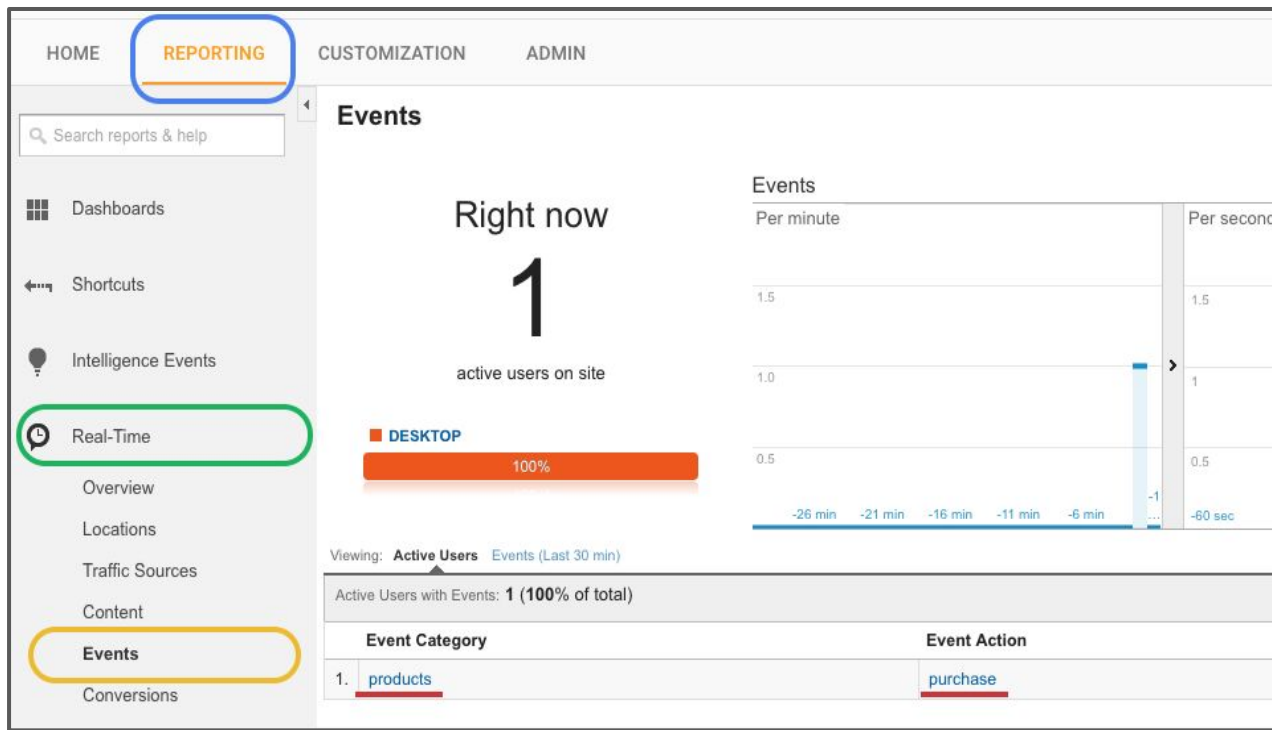
Finding the Real-time Overview interface

1. Select the Reporting tab
2. Select Real-time in the side panel
3. Select Overview



Finding the Real-time Events interface

1. Select the Reporting tab
2. Select Real-time in the side panel
3. Select Events



Finding the Events Overview interface

1. Select the Reporting tab
2. Select Behavior in the side panel
3. Select Events
4. Select Overview

