# Hooking into event callbacks

Last Modified on 05.07.2023

## Why use it

The Userlane Javascript API provides some events that you can hook into. This allows you to get notified when a user does something within Userlane.

Hooking into these events empowers you to

- extend and enhance the functionality of Userlane,
- integrate it with your existing ecosystem,
- trigger other reactions to the activity of your users in Userlane,
- forward those events to your analytics solution to correlate them with all the other data you have there
- create a more tailored and interactive user experience.

There are three events that you can listen to:

- **onStart:** fired after a user started a Guide
- **onComplete:** fired after a user successfully completed a Guide at its last step
- **onExit:** fired after a user canceled a Guide or the Guide was canceled because of an error

## Hooking into events

All events work with the same structure:

```
// hook into onStart and provide a callback function
Userlane('onStart', function(userlaneId, user) {

// you can put any code in here

console.log('the user started a guide');
});
```

Use the event name as the first argument and provide a callback function in the second argument. This callback function will be called with two arguments:

1. *userlaneId* is a unique identifier for the Guide that triggered the event.
2. user is the object with all the data about the user we have at this point. You can assume that it always has an ID attribute.

## onStart - after a user started a Guide

This event is fired whenever a Guide is started, even if it is started from the API's start command.

```
// hook into onStart and provide a callback function
Userlane('onStart', function(userlaneId, user) {

// you can put any code in here

console.log('the user started a guide');
});
```

## onComplete - after a user successfully completed a Guide at its last step

This event is fired whenever a user reached the end of a Guide. This indicated that it was completed successfully. From now on, the Guide will be marked with a green checkmark in the Userlane Assistant for this user.

```
// hook into onComplete and provide a callback function
Userlane('onComplete', function(userlaneId, user) {
  // you can put any code in here
  console.log('the user completed a guide');
});
```

## onExit - a user canceled a Guide

This event is fired whenever a user cancels a Guide using the Close button in the guide step. It is also fired when a Guide cancels itself after an error or when an important step is not found.

```
// hook into onExit and provide a callback function
Userlane('onExit', function(userlaneId, user, stepIndex) {
  // you can put any code in here
  console.log('the user canceled a guide');
});
```

## Overwriting callbacks

We only keep one callback for each event. If you hook up to an event again, it will overwrite your previous hook.

## Removing callbacks

If you want to unbind a callback from these events, call the event name again with false, null, or undefined instead of a callback function.

```
// remove all callbacks from the onStart event
Userlane('onStart', false);
```

## The user object

The second argument passed to your callback function is the user object. It contains all the data we have about the user at the time of the event. It is guaranteed that the user at least has an ID.

In addition to the ID, we provide the flag generated that indicates whether the ID was randomly generated by Userlane or whether it was provided to our API or snippet.

An example for the user object is

```
{
 id: "9717166719265375",
 generated: true,
 time_signup: "2017-11-22T11:50:48.671Z"
}
```