

# accounts.addEventHandlers JS

## Description

The Gigya service generates several global application events for various situations that are driven by user interactions. Global application events are fired whenever the event to which they refer occurs, regardless of which application component triggered the event. This method allows setting event handlers for each of the supported global events. To learn more about how to handle events generated by Gigya, see [Events](#).

**Note:** If you plan on integrating Gigya's Accounts API, we highly recommend reading the [Registration-as-a-Service Guide](#). [Registration-as-a-Service](#) (RaaS) is a premium platform that requires separate activation. If **RaaS** is not part of your site package, please contact Gigya by filling in a [support form](#) through the Console. You can access the support page by clicking **Support** on the upper menu after logging into the Gigya Console.

The following are two of the most used available global application events:

- **onLogin** - Fired whenever a user successfully logs in to your site. Please note, that several social & sharing plugins may generate this event. You may identify the source of the event using the 'source' field in the returned event data (refer to the [#onLogin](#) eventdata below).
- **onLogout** - Fired whenever a user logs out from your site.

For a complete list of available events, visit the [events guide](#).

An Event Handler is a JavaScript function with the following signature:

```
functionName(eventObj)
```

The single argument **eventObj**, contains information about the event and has different data fields for different events. In "[Event Handlers - Event Data](#)" below, you will find the specification of the available fields in the **eventObj** for each event.

## Syntax

## Parameters

The following table lists the available parameters:

Required	Name	Type	Description
	onLogin	function	A reference to a function that is called when the user is successfully authenticated by Gigya.
	onLogout	function	A reference to a function that is called when the user has logged out.
	callback	function	A reference to a callback function. Gigya calls the specified function along with the results of the API method when the API method completes. The callback function should be defined with the following signature: <b>functionName(Response)</b> . The "Response Object Data Members" table below provides specification of the data that is passed to the callback function.
	cid	string	A string of maximum 100 characters length. The CID sets categories for transactions that can be used later for filtering <a href="#">reports generated by Gigya</a> in the "Context ID" combo box. The CID allows you to associate the report information with your own internal data. For example, to identify a specific widget or page on your site/application. You should not define more than 100 different context IDs. <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"><b>Note:</b> This parameter overrides the value of the identical parameter in <a href="#">Global Conf</a> (the global configuration object). If the parameter is not set for the method, the value from Global Conf is used.</div>
	context	object	A developer-created object that is passed back unchanged to the application as one of the fields in the <a href="#">response object</a> .

## Response Object Data Members

Field	Type	Description
errorCode	integer	The result code of the operation. Code '0' indicates success, any other number indicates failure. For a complete list of error codes, see the <a href="#">Response Codes and Errors</a> table.
errorMessage	string	A short textual description of an error associated with the errorCode for logging purposes.
callId	string	Unique identifier of the transaction, for debugging purposes.
context	object	The context object passed by the application as a parameter to the API method, or <i>null</i> if no context object has been passed.

## Event Handlers - Event Data

The following tables specify the list of available data fields in the *eventObj* for each event.

### onLogin Event Data

Field	Type	Description
eventName	string	The name of the event: 'login'.
source	string	The source plugin that generated this event. The value of this field is the name of the plugin's API method, e.g. 'showLoginUI'. This field will not be available if the source of this event is not a plugin (e.g. if the source is a <a href="#">socialize.login</a> API call).
context	object	The context object passed as a parameter to the plugin/method that triggered this event, or null if no object was passed.
UID	string	The User ID that should be used for login verification*. <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"><b>Note:</b> The UID string must be encoded using the <code>encodeURIComponent()</code> function before sending it from your client to your server.</div>
UIDSignature	string	The signature that should be used for login verification, as described under the <a href="#">Validate the UID Signature in the Social Sign-On Process</a> section.
signatureTimestamp	string	The GMT time of the response in UNIX time format (i.e. The number of seconds since Jan. 1st 1970). The time stamp should be used for login verification, as described under the <a href="#">Validate the UID Signature in the Social Sign-On Process</a> section.
loginMode	string	The type of login: <ul style="list-style-type: none"><li>• <i>standard</i> - the user is logging into an existing account.</li><li>• <i>reAuth</i> - the user is proving ownership of an account by logging into it.</li></ul>
newUser	boolean	Whether the current login is for a newly created or existing user (in the database). <div style="border: 1px solid #ffeb3b; padding: 5px; margin-top: 10px;">When using TFA, a new user is created in the database prior to the user completing TFA and then logging in. This means that <code>newUser</code> is <i>always false</i> when TFA is enabled, even for users whose initial creation triggered the TFA.</div>
provider	string	The name of the provider that the user used in order to login (e.g. "Facebook"). Note: If this event is fired as a result of a <code>socialize.notifyLogin</code> call, i.e. the user was authenticated by your site, the <i>provider</i> field will be set to "site".
profile	<a href="#">Profile</a> object	The user's profile information as described in the <a href="#">Profile</a> object.
data	JSON object	Custom data. Any data that you want to store regarding the user, which isn't part of the <a href="#">Profile</a> object.

remember	Boolean	Whether the user asked to remember his credentials (using a "Remember me" checkbox).
----------	---------	--------------------------------------------------------------------------------------

\* To learn more about login verification, please refer to the [Validate the UID Signature in the Social Sign-On Process](#) section in the [Security](#) page of the Developer's Guide.

## onLogout Event Data

Field	Type	Description
eventName	string	The name of the event: 'logout'.
context	object	The context object passed as a parameter to the plugin/method that triggered this event, or null if no object was passed.

### Notes:

- This sample is not meant to be fully functional code. For brevity's sake, only the code required for demonstrating the API call itself is presented.
- To run the code on your own domain, add your Gigya API key to the `gigya.js` URL. A Gigya API key can be obtained on the [Site Dashboard](#) page on Gigya's website. Please make sure that the domain from which you are loading the page is the same domain name that you used for generating the API key.