

# Sitelok Javascript Access Plugin



## **Sitelok Javascript Access Plugin**

Copyright 2021 Vibralogix. All rights reserved.

This document is provided by Vibralogix for informational purposes only to licensed users of the Sitelok product and is provided on an 'as is' basis without any warranties expressed or implied.

Information in this document is subject to change without notice and does not represent a commitment on the part of Vibralogix. The software described in this document is provided under a license agreement. The software may be used only in accordance with the terms of that license agreement. It is against the law to copy or use the software except as specifically allowed in the license.

It is the users responsibility to ensure the suitability of the product before using it. In no circumstances will Vibralogix be responsible for any loss or damage of data or programs as a result of using the product. Your use of the product implies acceptance of these terms.

# Contents

<b>Chapter 1 Introduction .....</b>	<b>4</b>
What is the Javascript Access plugin? .....	4
<b>Chapter 2 Installation .....</b>	<b>5</b>
Installing for the first time or upgrading .....	5
Setting up the plugin.....	5
Disabling the Plugin.....	6
Uninstalling the plugin .....	6
<b>Chapter 3 Using the javascript API .....</b>	<b>7</b>
Adding the javascript API to your page .....	7
The sljsdata object.....	7
Read a users data .....	8
Write user data .....	9
See if user is a member of a usergroup .....	10
Displaying user data.....	11
Logout.....	12
<b>Chapter 4 Support .....</b>	<b>13</b>

# Chapter 1 Introduction

## What is the Javascript Access plugin?

Normally Sitelok is used to secure .php pages control by server side PHP scripts. For most websites this is ideal and provides a very secure solution.

However sometimes there may be a need to access user variables (name, email, custom fields etc) from a javascript or jQuery application. The Javascript Access plugin can provide access to a logged in users data and also allows writing to custom fields. This can be done on pages that are not .php as well, increasing the flexibility.

As javascript is run client side there are security issues to be considered. The javascript will only have access to the current user session (the logged in user) and you can set in the plugin the permissions for each field.

No Access

Read

Write

Read & write

The plugin assumes that you have sufficient knowledge of Javascript ,although the function are quite easy to understand.

# Chapter 2 Installation

## Installing for the first time or upgrading

- 1) Extract the contents of the zip file to your PC.
- 2) Upload the plugin\_sljs folder to your existing Sitelok slpw folder using FTP. There are no special permissions required on most servers.
- 3) Login to the Sitelok control panel.
- 4) Open the following URL in the browser

[https://www.yoursite.com/slpw/plugin\\_sljs/install.php](https://www.yoursite.com/slpw/plugin_sljs/install.php)

which will start the installation process. If all is well you will be taken to the plugin preferences page where you will see the plugin listed.

If you have any problems with installation please let us know so that we can help you.

## Setting up the plugin

You can access the plugin configuration by selecting **Javascript Access** from the **Plugins** menu.

### Permissions

The javascript API will only be able to access the user data fields for the current user session (logged in user). You can set the type of access for each field.

No access	Javascript will not have any access to this field
Read	Javascript will be able to read the value stored in this field
Write	Javascript will be able to write a value to this field
Read & Write	Javascript can read and write to this field

**Important** You should only allow read access to fields where you have no problem for the user to be able to see the value stored. Write access should only be provided to fields that you would otherwise provide user access to change using a profile form for example. All other fields should have their access to to None

## Disabling the Plugin

To disable the Javascript Access plugin select **Plugin Preferences** in the **Plugin** menu option of Sitelok. Uncheck the enable box for the plugin and click the **Save** button.

You can enable it again in the same way.

## Uninstalling the plugin

To permanently remove the plugin and its settings follow these steps.

- 1) Disable the plugin as above.
- 2) Click the delete icon next the plugin in the disabled plugins section.
- 3) Confirm the action in the alert box.

If the plugin is uninstalled successfully you will be returned to the plugin preferences page.

# Chapter 3 Using the javascript API

## Adding the javascript API to your page

The javascript API can be used on any page and doesn't need to be .php or have any Sitelok security added to it. The user will need to be logged in for the javascript to be able to access the users data though.

Before you can use the API you need to add sljslib.js to your page. Normally this would be done in the `<HEAD>` section but can also be loaded just before the `</BODY>` tag if preferred.

```
<script src="/slpw/plugin_sljs/sljslib.js"></script>
```

## The sljsdata object

After the library is loaded a global object called `sljsdata` is setup which will be used to store the users data and can be access from anywhere in your code. The values are empty until the user data is requested.

<code>sljsdata.slloggedin</code>	Set to true if the user is logged in or false if not
<code>sljsdata.sluserid</code>	Contains the users unique numeric id
<code>sljsdata.slusername</code>	The users username
<code>sljsdata.slname</code>	Full name
<code>sljsdata.slfirstname</code>	First name
<code>sljsdata.sllastname</code>	Last name
<code>sljsdata.slemail</code>	Email address
<code>sljsdata.slcreated</code>	The account created time. Unix timestamp
<code>sljsdata.slcustom1</code>	Custom 1 value
to	
<code>sljsdata.slcustom50</code>	Custom 50 value

## XSS safe values

The above properties contain the raw data stored in the database. As this data may have been entered by the user you need to be careful when displaying this on the page, or inserting it in the DOM. The object also contains XSS safe values for each property you can use in this case. Just add `_html` to the property name to access it. For example

[sljsdata\\_slcustom1\\_html](#)

## Usergroup membership

The [sljsdata](#) object contains information about the usergroups the user belongs to.

[sljsdata.slusergroups](#)

Contains an array of usergroups the user belongs to whether or not the membership to the group is expired or not. The [sljsdata.slusergroup](#) property contains the expiry time and other data about each usergroup. So for example [sljsdata.slusergroups\[0\]](#) would contain the first usergroup.

More data is available about each usergroup as follows.

[sljsdata.slusergroup.group.expiry](#)

This contains the expiry time in unix timestamp format or 0 where there is no expiry. [group](#) should be the usergroup name you want to access.

[sljsdata.slusergroup.group.desc](#)

Contains the usergroup description if set in the usergroup section of the dashboard.

[sljsdata.slusergroup.group.loginaction](#)

Contains the login action if set in the usergroup section of the dashboard. This can be blank for no action, URL for a specific url or custom1 if the redirection is based on a users custom1 value etc.

[sljsdata.slusergroup.group.loginvalue](#)

If the log inaction is set to URL then this contains the URL.

## Read a users data

You can use the [sljs\\_readall\(\)](#) function to read a users data, assuming they are logged in. Only the fields that have read or read & write permission will be returned. Other fields will be left blank.

[sljs\\_readall\(dbupdate,donefunction\);](#)

[dbupdate](#)

This determines if the call will update the users data from the database or just read it from the user session which is more efficient. Set to true to update from the DB or false to use the session data. Normally you only need to set this to



true of the users data is updated externally. In Sitelok V6.3 even changing data in the Sitelok dashboard will update a logged in users session so you can usually set this parameter to false.

**donefunction** As a call to the server using AJAX is required you will need to provide a function that is called when the server has finished processing and returns the user data. This function is sent one **status** parameter with the following values.

**0** Unable to retrieve data. If the value of `sljsdata.slloggedin` is false then the reason is that the user is not logged in.

**1** Data retrieved and `sljsdata` updated

The function can be a function defined elsewhere in your code or can be defined inline as in the example below.

Example call

```
sljs_readall(function(status) {  
  if ((status==1) && (sljsdata.slloggedin))  
    alert("Logged in");  
  if ((status==1) && (!sljsdata.slloggedin))  
    alert("Not logged in");  
  if (status==0)  
    alert("read data failed");  
});
```

## Write user data

You can use the `sljs_writefield()` function to read a users data, assuming they are logged in. Only the fields that have write permission can be written to.

Currently only the custom fields support writing. Other fields should be updated using update profile forms.

`sljs_writefield(fieldname,value,donefunction);`

**filename** This is the fieldname to write and can be 'slcustom1' to 'slcustom50'.

**value**

This is the string value to write to the field.

**donefunction**

As a call to the server using AJAX is required you will need to provide a function that is called when the server has finished processing. This function is sent the following parameters.

**status** which can have one of the following values.

**0** Failed. If the value of `sljsdata.slloggedin` is false then the reason is that the user is not logged in.

**1** Data written and `sljsdata` updated

The function can be a function defined elsewhere in your code or can be defined inline as in the example below.

### Example call

```
sljs_writefield('slcustom25', 'Hello', function(status, writefield, value) {
    if (status==1)
        alert("Updated");
    if ((status==0) && (!sljsdata.slloggedin))
        alert("User not logged in");
    if ((status==0) && (sljsdata.slloggedin))
        alert("write data failed");
});
```

## See if user is a member of a usergroup

Functions are provided to check if a user is a member of a usergroup. Before you can use these function you must have initialised the `sljsdata` object by calling `sljs_readall()` at some point previously. You must also ensure you have enabled read access for the Usergroups field in the plugin settings. The functions don't need to make an ajax call so can return true or false immediately.

### [sljs\\_ismemberof\(group\)](#)

This function returns true if the user belongs to the usergroup, even if expired.

example

```
if (sljs_ismemberof('GOLD'))
    alert("Member of GOLD");
```

### [sljs\\_isactivememberof\(group\)](#)

This function returns true if the user belongs to the usergroup and not expired.

example

```
if (sljs_isactivememberof('GOLD'))
    alert("Member of GOLD");
```

## Displaying user data

You can of course display user data directly in the DOM using javascript or jQuery directly. However we also provide an easy way to automatically update the user data on the page.

First of all you should add `<span>` tags for each piece of data to be displayed. Set the class of the `<span>` to one of the following

- [sluserid](#)
- [slusername](#)
- [slfirstname](#)
- [sllastname](#)
- [slemail](#)
- [slcreateddmy](#)
- [slcreatedmdy](#)
- [slcustom1](#)
- [to](#)
- [slcustom50](#)

example

```
<span class="slcustom1"></span>
```

After calling `sljs_readall()` or other changes are made you can call this function to automatically update the displayed values. The CSS safe value is used.

### Logout

You can logout the user using the function `sljs_logout(page)`. If page is "" then the user will be logged out and redirect to the logout page defined in the Sitelok configuration. If a page URL is provided then the user will be redirect to that URL. This function could be attached to a logout button onclick event for example.

example

```
<button onclick="sljslogout('');">Logout</button>
```

## Chapter 4 Support

Hopefully if you have followed this manual carefully everything will be working fine. However sometimes things don't go quite so smoothly so if you have any questions or problems then please check the FAQ on the support page or email us.

Support area: <https://www.vibralogix.com/support/>

Email: [support@vibralogix.com](mailto:support@vibralogix.com)