**How to implement AddThis SSI ?**

***Current System***

Assuming there is a pretty good mechanism for authenticate users into the system you have. Normal login system uses a user table having the username and password of the user. The system check for the combination of username with password and then authenticate the user as valid.

users table

|  |  |  |
| --- | --- | --- |
| id | username | password |

***SSI***

The lightweight plugin allows your users to login in one click using any of the three popular services(facebook, google). You can also use it to fill the profile data of users with the help of this plugin.

***Implementation***

In order to bring AddThis SSI in your system you only need a few modifications.

First of all you have to create applications on the desired services(facebook, google), if you have already an app, you can use it.

Next step is to generate the code

Navigate to http://www.addthis.com/labs/social-sign-in#code

Now you have a set of code with your desired services.

Sample code here:

*<div class="addthis\_toolbox">*

*<a class="addthis\_login\_facebook"></a>*

*<a class="addthis\_login\_google"></a>*

*</div>*

*<script>*

*var addthis\_config = {*

*login:{*

*services:{*

*facebook:{*

*appId:'myfacebookid'*

*},*

*google:{*

*clientId:'mygoogleid'*

*}*

*},*

*callback:function(user){*

*/\* Output user properties -- in your real app, you'd get rid of these\*/*

*console.log(user.addthis\_signature);*

*console.log(user.thumbnailURL);*

*console.log(user.service);*

*}*

*}*

*};*

*</script>*

*<script type="text/javascript" src="http://s7.addthis.com/js/300/addthis\_widget.js#pubid=ra-4eb14e177a332b5f"></script>*

Paste the generated code in appropriate place on your existing login page. Just refresh the your login page to ensure whether the buttons are rendering properly. Based on the services you selected, buttons will be shown on the place where you put the code.

A description...

***How to connect the SSI login to existing login mechanism***

Note the callback function portion in the generated code which is the function where all the user details including addthis signature are avail after the social sign in. Submit the form from that function and sent the user properties and signature to the server.

***AddThis Signature***

It is an encrypted code sent along with the user property to identify the user, it varies from user to user and domain to domain. On initial login you have to store this addthis signature against the id of the user in database.

social sign in table

|  |  |  |
| --- | --- | --- |
| socialid | id(user table) | addthis\_signature |

You can check for the addthis\_signature on the future logins and authenticate if the signature matches.

***Facebook Responses***

|  |  |
| --- | --- |
| 1 | id |
| 2 | name |
| 3 | first\_name |
| 4 | last\_name |
| 5 | link |
| 6 | username |
| 7 | hometown |
| 8 | location |
| 9 | work |
| 10 | education |
| 11 | gender |
| 12 | timezone |
| 13 | locale |
| 14 | verified |
| 15 | updated\_time |
| 16 | service |
| 17 | addthis\_signature |

It is also possible to request additional permissions in case of facebook.

ex:

*<script>*

*var addthis\_config = {login : {services : {facebook:{appId:’fb123456’, scope:’user\_photos, user\_relationships’},google:{clientId:’google12345’}}, callback:function(user) {*

*//console.log(user);*

*}*

*}*

*}*

*</script>*

***Google Responses***

|  |  |
| --- | --- |
| 1 | id |
| 2 | email |
| 3 | verified\_email |
| 4 | name |
| 5 | given\_name |
| 6 | family\_name |
| 7 | link |
| 8 | picture |
| 9 | gender |
| 10 | birthday |
| 11 | locale |
| 12 | service |
| 13 | addthis\_signature |