Travelogix

Google Login Integration

Document Identification

|  |  |  |  |
| --- | --- | --- | --- |
| Author | | Document Location (repository/path/name) | |
| Deekshitha A.R | | JAM Project Collaboration Site | |
| Version | Status | Date (YYYY-MM-DD) | Document Classification |
| 1.0 | Initial | 2016-07-26 | Internal |

Revision history

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 1.0 | July 26 2016 | 1st Release of the document |
|  |  |  |

approvals/Sign-off

|  |  |  |  |
| --- | --- | --- | --- |
| Date | SME / Function | Sections in this document | Topics/Section |
|  |  |  |  |
|  |  |  |  |

Reference documents

|  |  |  |
| --- | --- | --- |
| Topic | Owner | Storage Location / Link |
|  |  |  |
|  |  |  |
|  |  |  |

Document Review

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Owner | Reviewer | Comments |
| 26/07/2016 | Deekshitha A.R | Sanu |  |
|  |  |  |  |

Contents

[Travelogix 1](#_Toc454350368)

[Document Identification 1](#_Toc454350369)

[approvals/Sign-off 2](#_Toc454350370)

[1. About this Document 4](#_Toc454350371)

[2. Implementation 4](#_Toc454350372)

[3. Acronyms 7](#_Toc454350373)

## 

* **Logging into Hybris system**:

If the token is valid using the “GoogleAutoLoginStrategy” using the token and resultant Profile can looking into the hybis system.

## About this Document

This document outlines the integration of [y]-commerce with Gmail for logging into the website using the Gmail credentials.

## Implementation

Sending the request to the Gmail and getting the token.

* Authenticating the token
* Logging in into the hybris system using the token and the resultant profile information from Gmail.

Sending the request using the google api’s and js libraries and getting the token in response, the application ids is given in jsp.

**Steps to integrate the google login.**

1. Copy the js and alter the styles for ui of google login button

Created a button and integrated the button to call the google service to return the token

Here is the lines of code. Need to integrate this with the frontend button wherever required.

<html>

<head>

<link href="https://fonts.googleapis.com/css?family=Roboto" rel="stylesheet" type="text/css">

<script src="https://apis.google.com/js/api:client.js"></script>

<script>

var googleUser = {};

var startApp = function() {

gapi.load('auth2', function(){

// Retrieve the singleton for the GoogleAuth library and set up the client.

auth2 = gapi.auth2.init({

client\_id: '953549100422-dh282r3i334d9gnstivdlr7bpu9l3a13.apps.googleusercontent.com',

cookiepolicy: 'single\_host\_origin',

// Request scopes in addition to 'profile' and 'email'

//scope: 'additional\_scope'

});

attachSignin(document.getElementById('customBtn'));

});

};

function attachSignin(element) {

console.log(element.id);

auth2.attachClickHandler(element, {},

function(googleUser) {

document.getElementById('name').innerText = "Signed in: " +

googleUser.getBasicProfile().getName()+ "token" + googleUser.getAuthResponse().id\_token;

$.ajax({

url: 'googleoauth2callback?token=googleUser.getAuthResponse().id\_token',

data: 'data',

success: function (data) {

alert(data);

}

});

}, function(error) {

alert(JSON.stringify(error, undefined, 2));

});

}

</script>

<style type="text/css">

#customBtn {

display: inline-block;

background: white;

color: #444;

width: 190px;

border-radius: 5px;

border: thin solid #888;

box-shadow: 1px 1px 1px grey;

white-space: nowrap;

}

#customBtn:hover {

cursor: pointer;

}

span.label {

font-family: serif;

font-weight: normal;

}

span.icon {

background: url('/identity/sign-in/g-normal.png') transparent 5px 50% no-repeat;

display: inline-block;

vertical-align: middle;

width: 42px;

height: 42px;

}

span.buttonText {

display: inline-block;

vertical-align: middle;

padding-left: 42px;

padding-right: 42px;

font-size: 14px;

font-weight: bold;

/\* Use the Roboto font that is loaded in the <head> \*/

font-family: 'Roboto', sans-serif;

}

</style>

</head>

<body>

<!-- In the callback, you would hide the gSignInWrapper element on a

successful sign in -->

<div id="gSignInWrapper">

<span class="label">Sign in with:</span>

<div id="customBtn" class="customGPlusSignIn">

<span class="icon"></span>

<span class="buttonText">Google</span>

</div>

</div>

<div id="name"></div>

<script>startApp();</script>

</body>

</html>

When added this code, the button looks like



Need to add the js for which button you need for, and this is the application id

client\_id: '953549100422-dh282r3i334d9gnstivdlr7bpu9l3a13.apps.googleusercontent.com'

for the change in url need to create the application in developers google application and register the url and generate the client id. Generated client id should be added her in the js.

## Add the googleaddon and run addon install command, already included in the git.

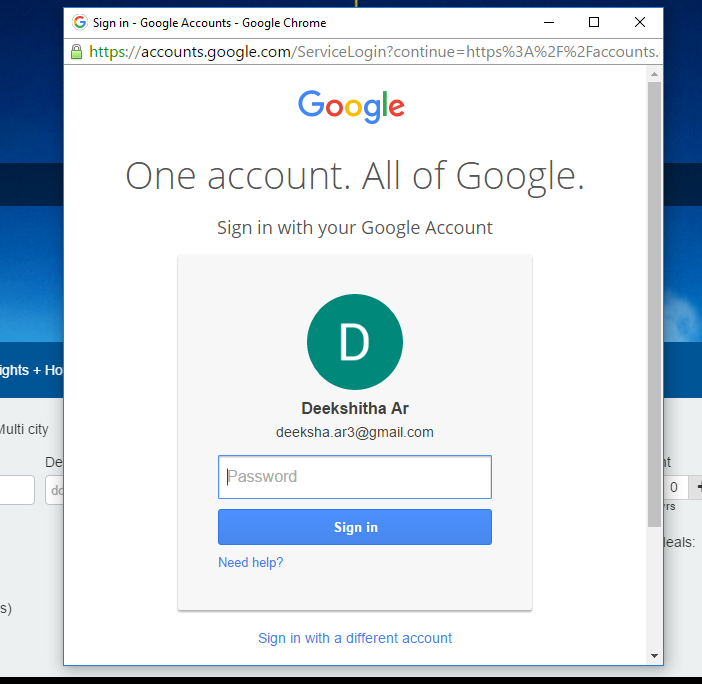
**Google Login Implementation**

The return response from the google authenticates again while login and uses GoogleAutoLoginStrategy , GoogleAuthenticationToken, GoogleAuthenticationProvider.

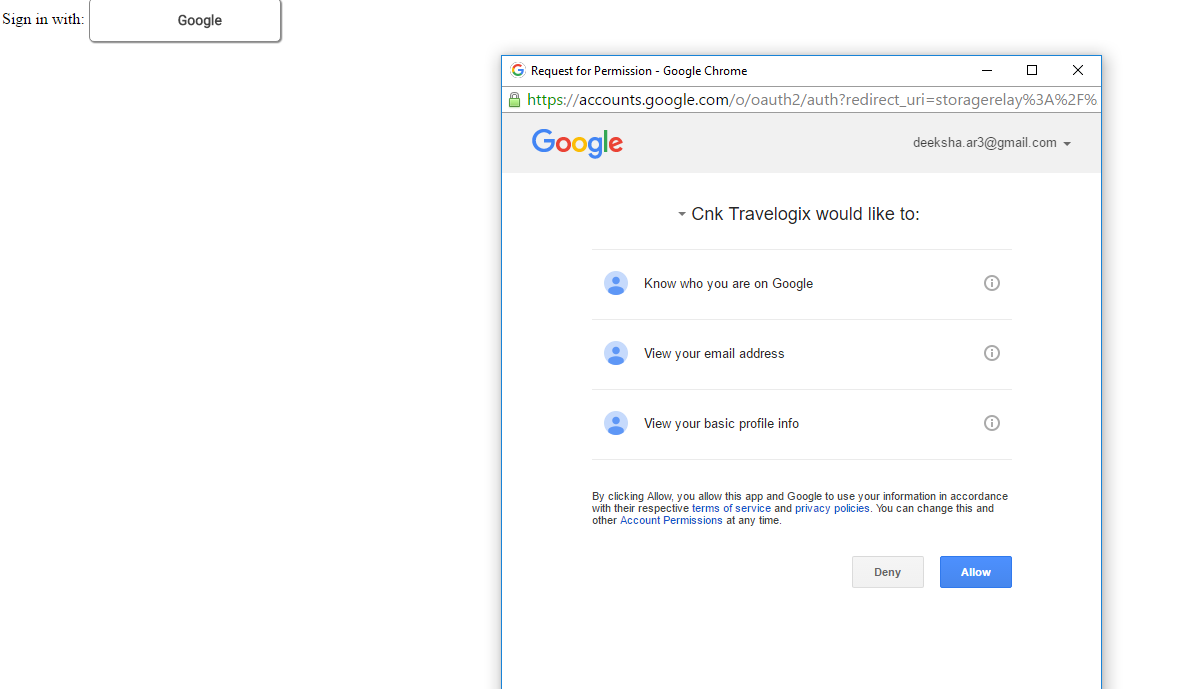
Once the google button is integrated.

On click of goggle login button, will be able to see the pop up to login and ask for the permeation to access the credential details. If allowed, will be able to login to the site.

On click of gmail login button, if the gmail is not logged in, in the current session then gives the pop up to login into the gmail.

****

For the first time to get the permeation to access the information, it gives the pop up shown in below.

****

On click of allow it logs in to the website.