Article & Comment Component

Version: 0.1

**Published**: 08/12/2016



Template Version: 10

**TEMPLATE REVISION HISTORY**

| **Version #** | **Date** | **Revised By** | **Description of Changes** |
| --- | --- | --- | --- |
| 7.0 | 3/25/2013 | Dan Knapp | Initial release to company |
| 8.0 | 7/12/2013 | Dan Knapp | App Dev links added |
| 9.0 | 8/4/2014 | Dan Knapp | Summer/ Fall 2014 Update |
| 10 | \_\_/\_\_/2014 | Kendra Chatwood | Incorporate certain Dev Methodology templates |

**PROJECT LLD REVISION HISTORY**

| **Version #** | **Date** | **Revised By** | **Description of Changes** |
| --- | --- | --- | --- |
| 1.0 | 8/12/2016 | Sandeep Chopra |  |
|  |  |  |  |

**SUPPORTING DOCUMENTS**

|  |  |
| --- | --- |
| **Document Number** | **URL** |
| 1 |  |
|  |  |

Cox Communications, Inc. © 2012, 2013, 2014, 2015 All rights reserved.

This documentation is the confidential and proprietary intellectual property of Cox Communications, Inc. Any unauthorized use, reproduction, preparation of derivative works, performance, or display of this document, or software represented by this document is strictly prohibited.

Cox Communications and the Cox Communications logo design are trademarks and/or service marks of Cox Enterprises, Inc. All other trademarks, service marks, and trade names are owned by their respective companies.

TABLE OF CONTENTS

[1 Low Level Design – App Dev 4](#_Toc457808422)

[1.1 Low Level Design Overview 4](#_Toc457808423)

[1.2 Design Considerations 4](#_Toc457808424)

[1.2.1 Pseudo-Code/ Code Plan 4](#_Toc457808425)

[1.2.2 System Environment 4](#_Toc457808426)

[1.2.3 Hardware Requirements and Interfaces 4](#_Toc457808427)

[1.2.4 Design Methodology 5](#_Toc457808428)

[1.2.4.1 Landing Page Mockup 6](#_Toc457808429)

[1.2.4.2 High Level Functions 7](#_Toc457808430)

[1.3 Application Architecture Component Design 7](#_Toc457808431)

[1.4 Detailed System Design 8](#_Toc457808432)

[1.4.1 Design Description – narrative 8](#_Toc457808433)

# Low Level Design – App Dev

## Low Level Design Overview

The News Article page will display the stylized content for a particular news item in the News list. The entry area for a news item’s content will be a SharePoint announcement list that is extended with a custom content type that adds columns to support the news entry requirements.

## Design Considerations

### Pseudo-Code/ Code Plan

A SharePoint Hosted App project would be created in Visual Studio. Below are the components to be created under this project:

1. Site Columns
2. Content Types
3. List Definitions
4. Application Pages
5. Features
6. JavaScript

### System Environment

|  |  |
| --- | --- |
| Environments | FARM Server Details |
| O365 | Site Collection hosted in O365 |

### Hardware Requirements and Interfaces

|  |  |
| --- | --- |
| **Component** | **Minimum requirement** |
| Predefined | Use Resources specified by O365 |

### Design Methodology

The News Article page is comprised of several components that are supported by a SharePoint two column page layout. On the right upper corner of the page layout will display the Details for the main Article that was click on from the sending component on a different page, for example, a user clicks on an article in the News Carousel. The list ID (GUID) and the Article’s SharePoint list item ID are passed in the Query String and used as the filters for the Content Search Web Part to display the details for that particular list item on the News Article page.

|  |
| --- |
|  |
| Figure 1 – Main News Article on the News Article page |

The display of the News Article is controlled via customization of the content search web part’s display template. The image for the article is placed in a div that occupies the top right corner of the News Article page. The Body text is entered in a multiline textbox control that uses WYWIG.

The controls for Likes and Share are placed in the upper right hand corner and view comments are placed in the lower left hand corner.

An additional content Search web part is placed on the article page for displaying related news articles based on the requirements.

In the right section of the page layout are the two additional web parts for the Recommended and Trending article web parts that when an article is clicked in one of these web parts the respective article is shown in the main article section.

#### Landing Page Mockup

The News Article page showing the four web parts:

1. The Main Article Section
2. The Related Article Section
3. The Recommended Article Section
4. The Trending Article Section

|  |
| --- |
|  |
| Figure 2 – The four section of the News Article Page |

#### High Level Functions

The news article page is the landing place once a user clicks on an article in a web part that presents the user a snippet of the news article. The page is composed of four key web parts the main article, the web part that displays related articles, the web part that displays the recommended articles based on the user’s set preferences and the trending articles which are the articles with the most views.

## Application Architecture Component Design

**Requirements – Article and Comments**

|  |
| --- |
| 1. The news article page displays the selected article, additional x news articles based on the selected article's related component, Trending Articles component, and Recommended component  Ex:  a. If the article was clicked on from a news channel, additional stories from that news channel display below the article by published date in descending order  b. If the article was clicked on from the Trending Articles component, additional stories from that Trending Articles component display below the article by published date in descending order  c. If the article was clicked on from the front page carousel, additional stories from the Company channel display below the article by published date in descending order 2. If the article was clicked on from the additional stories below the article, additional stories from the original origination point of the original article should display below the article by published date in descending order  Ex: If the user first clicks an article from the Field Services channel, the Field Services article displays followed by other Field Services articles. IF the user clicks on one of the articles below, it displays followed by other Field Services articles. 3. The Trending Articles component displays 4 news tiles (see REQ\_NEWS 108) driven by likes/comments and sorted in descending order by trend 4. The Trending Articles component can only show news articles published within the last 90 days 5. A trending article can be suppressed from the Trending Articles with a checkbox in the article publishing form 6. The Recommended component displays 4 news tiles based on news preferences sorted in descending order by published date |
| 1. Additional news articles in the same channel as the first-opened article display below the first-opened article and load as the user scrolls down 2. **For news tiles in the Trending Articles and Recommended, see REQ\_NEWS 108-109** |
| 1. The news article displays the news article channel name, news title, byline, number of likes, likes icon, share icon, body of text, tags, "View Comments (##)" link 2. The news article channel name is populated based on the origin of the clicked article. For example,  a. If the user clicked on a story from the carousel, the article's channel name (and the channel name for all subsequent articles in the feed) displays as "Company"  b. If the user clicked on a story from the news grid, the article's channel name (and the channel name for all subsequent articles in the feed) displays as the channel from which the article was clicked (If the article was clicked from Field Services, "Field Services" would display as the news article channel name)  c. If the user clicked on a story from the Trending component, the article's channel name (and the channel name for all subsequent articles in the feed) displays as "Trending"  d. If the user clicked on a story from the Recommended component, the article's channel name (and the channel name for all subsequent articles in the feed) displays as "Recommended" 3. When the comments are expanded, "View Comments (##)" link changes to "Hide Comments" 4. When the comments are expanded, the following displays: the text "Comments", "Start a conversation" text box, up to 5 comments are showed initially followed by a "SHOW MORE COMMENTS link (if there are more than 5 comments), and "Show More" link |
| 1. When the user clicks on the like icon, the number of likes increments up  2. If the user has already liked the story and clicks on the like icon again, the number of likes increments down 3. When the user clicks on the share icon, a share sheet displays, giving the user the option to share via email 4. When the user opts to share via email and clicks the email button, the user's email client opens a drafted email with the Title, Short Description, and URL 4. When the user clicks on the "View Comments (##)" link, the comments expand 5. When the user clicks in the "Start a conversation" text box, he can type a comment and click "Submit" 6. When the user clicks on the "SHOW MORE COMMENTS" link, up to 5 more comments are loaded into the comments section for the given article 7. When the user clicks "Like" on a given comment, the likes indicator on that comment increments up 8. If the user has already liked the comment and clicks on the like icon again, the number of likes increments down 9. In order to reply to a particular comment, he clicks "Reply", which opens a reply comment text box  10. When the user clicks the "Hide Comments" link, the comments collapse 11. When the user clicks any picture in the article, a larger rendition of the picture displays in a modal window "in front of" the article.   (Note: The image that is displayed in the modal window should be the largest resolution rendition that the modal window can take) 12. With the picture modal window open, when the user clicks the close visual indicator, "X", the modal window closes, and the article displays |
| 1. The news carousel cycles through all featured stories every 7 seconds. This duration is configurable. 2. When the user begins to interact with the carousel, the automated cycling stops (Ex: it shouldn't rotate while the user is swiping through or in the process of tapping a story) 3. When the user clicks the picture, the corresponding page from the configurable "Learn More" link displays 4. When the user clicks the configurable "Learn More" link, the corresponding page from the configurable "Learn More" link displays 5. When the user clicks on the like icon, the number of likes increments up  6. If the user has already liked the story and clicks on the like icon, the number of likes increments down |
| 1. When the user clicks the "Load More" button, more articles display according to the logic in REQ\_NEWS 121.1 and 121.2 2. For news tiles in the Trending Articles and Recommended see REQ\_NEWS 108-109 |
| 1. When the user clicks on the like icon, the number of likes increments up  2. If the user has already liked the story and clicks on the like icon again, the number of likes increments down 3. When the user clicks on the share icon, a share sheet displays, giving the user the option to share via email 4. When the user clicks on the "View Comments (##)" link, the comments expand 5. When the user clicks in the "Start a conversation" text box, he can type a comment and click "Submit" 6. When the user clicks on the "SHOW MORE COMMENTS" link, up to 5 more comments are loaded into the comments section for the given article 7. When the user clicks "Like" on a given comment, the likes indicator on that comment increments up 8. If the user has already liked the comment and clicks on the like icon again, the number of likes increments down 9. In order to reply to a particular comment, he clicks "Reply", which opens a reply comment text box  10. When the user clicks the "Hide Comments" link, the comments collapse |

## Detailed System Design

### 1.4.1 Design Description – narrative

|  |  |
| --- | --- |
| **Sr. No.** | 1 |
| **List Display Name** | News Articles |
| **List Internal Name** | NewsArticles |
| **List Type** | <Custom list> |
| **Content Type** | CCI-NewsArticle |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Display Name** | **Type (Control)** | **Required** | **Description** | **Business Rules** |
| Title | Title |  |  |  |  |
| CCI-Body | CCI-Body |  |  |  |  |
| CCI-Byline | CCI-Byline |  |  |  |  |
| CCI-DoNotArchive | CCI-Do Not Archive |  |  |  |  |
| CCI-FeaturedArticle | CCI-Featured Article |  |  |  |  |
| CCI-NewsArticleImage | CCI-News Article Image |  |  |  |  |
| CCI-NewsHeadline | CCI-News Headline |  |  |  |  |
| CCI-Description | CCI-Description |  |  |  |  |
| CCI-TrendingSuppression | CCI-Trending Suppression |  |  |  |  |
| CCI-PersonOrGroup | CCI-Person or Group |  |  |  |  |

Item Template will have following components.

<html xmlns:mso="urn:schemas-microsoft-com:office:office" xmlns:msdt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882">

<head>

<style>

ul.cbs-List {

display: block !important;

}

</style>

<title>CoxOne Articles</title>

<!--[if gte mso 9]><xml>

<mso:CustomDocumentProperties>

<mso:TemplateHidden msdt:dt="string">0</mso:TemplateHidden>

<mso:ManagedPropertyMapping msdt:dt="string">'Picture URL'{Picture URL}:'PubImage;PublishingImage;PictureURL;PictureThumbnailURL;CCI-NewsArticleImageOWSIMGE','Link URL'{Link URL}:'Path','Work Email'{Work Email}:'WorkEmail','MainAuthor'{MainAuthor}:'AuthorOWSUSER','Line 1'{Line 1}:'Title','Line 2'{Line 2}:'RefinableString08','Line 3'{Line 3}:'ModifiedDateTime','Line 4'{Line 4}:'ReplyCount','Line 5'{Line 5}:'LikesCount','Line 6'{Line 6}:'LastReplyByOWSUSER','Line 7'{Line 7}:'DisplayAuthor','Line 8'{Line 8}:'ListItemId','Line 9'{Line 9}:'NewsCommentsOWSMTXT','Line 10'{Line 10}:'ListId'</mso:ManagedPropertyMapping>

<mso:MasterPageDescription msdt:dt="string">This Item Display Template will show a 100x100 picture of the item on the left. The title and the default item description will display to the right of the picture with an additional line that is available for a custom managed property.</mso:MasterPageDescription>

<mso:ContentTypeId msdt:dt="string">0x0101002039C03B61C64EC4A04F5361F385106603</mso:ContentTypeId>

<mso:TargetControlType msdt:dt="string">;#Content Web Parts;#</mso:TargetControlType>

<mso:HtmlDesignAssociated msdt:dt="string">1</mso:HtmlDesignAssociated>

<mso:HtmlDesignStatusAndPreview msdt:dt="string">https://mod223137.sharepoint.com/sites/coxone/\_catalogs/masterpage/Display Templates/Content Web Parts/CoxOne\_Item\_Articles.html, Conversion successful.</mso:HtmlDesignStatusAndPreview>

<mso:HtmlDesignConversionSucceeded msdt:dt="string">True</mso:HtmlDesignConversionSucceeded>

<mso:CrawlerXSLFile msdt:dt="string"></mso:CrawlerXSLFile>

<mso:HtmlDesignPreviewUrl msdt:dt="string"></mso:HtmlDesignPreviewUrl>

</mso:CustomDocumentProperties>

</xml><![endif]-->

</head>

<body>

<!--

Warning: Do not try to add HTML to this section. Only the contents of the first <div>

inside the <body> tag will be used while executing Display Template code. Any HTML that

you add to this section will NOT become part of your Display Template.

-->

<script>

$includeLanguageScript(this.url, "~sitecollection/\_catalogs/masterpage/Display Templates/Language Files/{Locale}/CustomStrings.js");

</script>

<!--

Use the div below to author your Display Template. Here are some things to keep in mind:

\* Surround any JavaScript logic as shown below using a "pound underscore" (#\_ ... \_#) token

inside a comment.

\* Use the values assigned to your variables using an "underscore pound equals"

(\_#= ... =#\_) token.

-->

<div id="Item\_Picture3Lines">

<!--#\_

var encodedId = $htmlEncode(ctx.ClientControl.get\_nextUniqueId() + "\_picture3Lines\_");

var linkURL = $getItemValue(ctx, "Link URL");

linkURL.overrideValueRenderer($urlHtmlEncode);

var line1 = $getItemValue(ctx, "Line 1");

var line2 = $getItemValue(ctx, "Line 2");

var line3 = $getItemValue(ctx, "Line 3");

var line4 = $getItemValue(ctx, "Line 4");

var line5 = $getItemValue(ctx, "Line 5");

var line6 = $getItemValue(ctx, "Line 6");

var line7 = $getItemValue(ctx, "Line 7");

var line8 = $getItemValue(ctx, "Line 8");

var line9 = $getItemValue(ctx, "Line 9");

var line10 = $getItemValue(ctx, "Line 10");

var liked = '';

var pictureURL = $getItemValue(ctx, "Picture URL");

var pictureId = encodedId + "picture";

var pictureMarkup = Srch.ContentBySearch.getPictureMarkup(pictureURL, 100, 100, ctx.CurrentItem, "cbs-picture3LinesImg", line1, pictureId);

line1.overrideValueRenderer($contentLineText);

line2.overrideValueRenderer($contentLineText);

line3.overrideValueRenderer($contentLineText);

line4.overrideValueRenderer($contentLineText);

line5.overrideValueRenderer($contentLineText);

line6.overrideValueRenderer($contentLineText);

line7.overrideValueRenderer($contentLineText);

line8.overrideValueRenderer($contentLineText);

line9.overrideValueRenderer($contentLineText);

line10.overrideValueRenderer($contentLineText);

var containerId = encodedId + "container";

var pictureLinkId = encodedId + "pictureLink";

var pictureContainerId = encodedId + "pictureContainer";

var dataContainerId = encodedId + "dataContainer";

var line1LinkId = encodedId + "line1Link";

var line1Id = encodedId + "line1";

var line2Id = encodedId + "line2";

var line3Id = encodedId + "line3";

var line4Id = encodedId + "line4";

var line5Id = encodedId + "line5";

var line6Id = encodedId + "line6";

var line7Id = encodedId + "line7";

var line8Id = encodedId + "line8";

var line9Id = encodedId + "line9";

var line10Id = encodedId + "line10";

if(!line5.value)

{

line5.value = 0;

}

if(!line4.value)

{

line4.value = 0;

}

if(line6.value)

{

line6.value = 'Last Reply by: ' + line6.value;

}

else

{

line6.value = '';

}

//alert(line9.value);

var authors2=line7.value;

var author =authors2.split(";", 1);

var dataDisplayTemplateTitle = "ItemPicture3Lines";

var parseDate = new Date(line3.value);

var formattedDate = parseDate.format('MMM dd, yyyy') + ' at ' + parseDate.format('hh:mm tt');

//alert(line7.value);

if(!$isNull(ctx.CurrentItem) && !$isNull(ctx.ClientControl))

{

var NoOfLikes = 0;

var NoOfReplies = 0;

if(ctx.CurrentItem.LikesCount != null)

{

NoOfLikes=ctx.CurrentItem.LikesCount;

}

if(ctx.CurrentItem.ReplyCount != null)

{

NoOfReplies=ctx.CurrentItem.ReplyCount;

}

}

NoOfReplies=0;

if(line9.value != null)

{

NoOfReplies = String(line9.value).split('<div>').length-1;

}

var listGuid;

var listItemId;

listGuid = line10.value;

listItemId = line8.value;

AddPostRenderCallback(ctx, function(){

//getLikes(line9.value, line8.value);

//alert(line8.value);

$('.comments').hide();

$("#lnkViewComments").click(function(){

$('.comments').toggle();

});

if(String(listGuid).indexOf("{") > -1)

listGuid = listGuid.substring(1,listGuid.length-1);

$.ajax({

url: \_spPageContextInfo.webAbsoluteUrl + "/\_api/contextinfo",

type: "POST",

headers: { "Accept": "application/json; odata=verbose"},

success: function (data) {

$('#\_\_REQUESTDIGEST').val(data.d.GetContextWebInformation.FormDigestValue);

var restUrl = \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/GetById('" + listGuid + "')/GetItems(query=@v1)?@v1={\"ViewXml\":\"<View><Query><Where><Eq><FieldRef Name='ID'/><Value Type='Counter'>" + listItemId + "</Value></Eq></Where></Query><RowLimit>1</RowLimit></View>\"}";

//alert(restUrl);

//var restUrl = \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/GetById('" + listGuid + "')/items(" + listItemId + ")?$select=Title,LikesCount,LikedBy/ID,LikedBy/Title&$expand=LikedBy";

$.ajax(

{

url: restUrl,

type: "POST",

headers: {

"Accept": "application/json; odata=verbose",

"X-RequestDigest": $("#\_\_REQUESTDIGEST").val()

},

success: function (data) {

if (data.d.results.length > 0)

{

var likeDisplay = true;

var result = '';

var likesCountInfo = 0;

$.each(data.d.results, function (index, item) {

if(item.LikedByStringId != null)

{

likesCountInfo = item.LikesCount;

var members = item.LikedByStringId.results;

if (members != null) {

for (i = 0; i < members.length; i++) {

var member = members[i];

if(parseInt(member) === \_spPageContextInfo.userId)

likeDisplay = false;

}

}

}

});

var likebuttonid = "newsarticlelikebutton" + listItemId;

var likecountid = "newsarticlelikecount" + listItemId;

if (likeDisplay) {

$("#" + likebuttonid).text('Like');

}

else {

$("#" + likebuttonid).text('Unlike');

}

var htmlstring = "(" + String(likesCountInfo) + ") " + "<i class='fa fa-thumbs-up'></i>";

var htmlstring1 = "(" + String(likesCountInfo) + ") " + "<i class='fa fa-thumbs-up'></i>";

if (likesCountInfo > 0)

{

//$(".likecount").html(htmlstring)

$("#" + likecountid).html(htmlstring)

}

else

{

//$(".likecount").html(htmlstring1);

$("#" + likecountid).html(htmlstring1);

}

}

},

error: function (error) {

alert(JSON.stringify(error));

}

});

},

error: function (data, errorCode, errorMessage) {

alert(errorMessage)

}

});

/\*

//Post Comment Section

$("#btnSave").click(function(){

//$("#btnSave").prop('disabled', true);

//$("#txtMessage").prop('disabled', true);

//alert("Button was clicked.");

if($('textarea#txtMessage').val() != '')

{

var clientContext = new SP.ClientContext.get\_current();

var user = clientContext.get\_web().get\_currentUser();

var list = clientContext.get\_web().get\_lists().getByTitle("News Article");

var item = list.getItemById(listItemId);

var camlQuery = new SP.CamlQuery();

camlQuery.set\_viewXml('<View><Query><Where><Eq><FieldRef Name=\'ID\'/>' +

'<Value Type=\'Counter\'>' + listItemId + '</Value></Eq></Where></Query><RowLimit>1</RowLimit></View>');

var collListItem = list.getItems(camlQuery);

//clientContext.load(user);

clientContext.load(collListItem);

clientContext.executeQueryAsync(

function(){

var listItemEnumerator = collListItem.getEnumerator();

var prevMessage;

var username = user.get\_title();

while (listItemEnumerator.moveNext()) {

var oListItem = listItemEnumerator.get\_current();

prevMessage = oListItem.get\_item('NewsComments');

}

//alert(username);

//alert(prevMessage);

var todaysDate = new Date();

var date = (todaysDate.getMonth() + 1) + "/" + todaysDate.getDate() + "/" + todaysDate.getFullYear();

var message = "<div>" + username + "(" + date + " " + formatAMPM(new Date()) + "): " + $('textarea#txtMessage').val() + "</div>" + prevMessage;

//alert(message);

item.set\_item('NewsComments', message);

item.update();

clientContext.executeQueryAsync(

function () {

$('textarea#txtMessage').val('');

alert("Message posted successfully but it will take some time to reflect.");

},

function (sender, args) {

alert('Exception occurred:' + args.get\_message());

});

},

function(sender, args){

alert('Exception occurred:' + args.get\_message());

});

}

else

{

alert("Please enter some message to post!");

}

});

\*/

});

\_#-->

<div class="row news\_box top-new-box">

<div class="box\_accordian padd\_comon">

<div class="share\_like\_row">

<div class="left\_side">

<h3>\_#= line1.value =#\_<span> by \_#= author =#\_</span></h3>

</div>

<div class="share\_social">

<ul>

<!--<li> <a href="#" class="like"><span> (\_#= NoOfLikes =#\_)</span> <i class="fa fa-thumbs-up"></i></a></li>-->

<li>

<div class="LikeSection">

<a href="#" class="like">

<span class="likecount" id="newsarticlelikecount\_#= line8.value =#\_" onclick="javascript: LikePage('\_#= line10.value =#\_', '\_#= line8.value =#\_', 'newsarticle')"></span>

<a href="#" class="LikeButton" id="newsarticlelikebutton\_#= line8.value =#\_" style="display:none;"></a>

</a>

</div>

</li>

<li><a href="#" class="share\_icon"><i class="fa fa-share-alt"></i></a></li>

</ul>

</div>

</div>

<div class="content\_row">

<div class="img\_data left\_side\_img"> \_#= pictureMarkup =#\_</div>

<p> \_#= line2.value =#\_</p>

<div class="clearfix"></div>

<a href="#">products loremipsum justovitae</a>

<div class="clearfix"></div>

<a href="#" id="lnkViewComments" class="comments-view">View Comments (\_#= NoOfReplies =#\_)</a>

<div class="comments">\_#= line9.value =#\_</div>

</div>

</div>

</div>

<!--#\_

var accountName = line7.value;

EnsureScriptFunc('jquery.js', 'jQuery', function () {

// Replace the next line with a function call

//console.log('jQuery ready');

$.ajax({

url: \_spPageContextInfo.webAbsoluteUrl + "/\_api/contextinfo",

type: "POST",

headers: { "Accept": "application/json; odata=verbose"},

success: function (data) {

$('#\_\_REQUESTDIGEST').val(data.d.GetContextWebInformation.FormDigestValue);

//alert($("#\_\_REQUESTDIGEST").val());

ensureUser(accountName)

.done(function(data)

{

console.log('Done.');

//alert(data.d.LoginName);

console.log(data.d.LoginName);

var requestHeaders = {

"Accept": "application/json;odata=verbose",

"X-RequestDigest": $("#\_\_REQUESTDIGEST").val()

};

shortAccountName = data.d.LoginName.split("|")[1];

//alert(shortAccountName);

console.log(shortAccountName);

jQuery.ajax({

url:\_spPageContextInfo.webAbsoluteUrl + "/\_api/SP.UserProfiles.PeopleManager/GetPropertiesFor(accountName=@v)?@v='" + shortAccountName + "'",

type:"POST",

contentType : "application/json;odata=verbose",

headers: requestHeaders,

success:function(data){

//console.log(data);

//alert(data.d.PictureUrl);

//console.log(data.d.PictureUrl);

if(!data.d.PictureUrl)

pictureURL = \_spPageContextInfo.webAbsoluteUrl + '/\_layouts/15/images/PersonPlaceholder.96x96x32.png';

else

pictureURL = data.d.PictureUrl;

pictureId = '#picture' + line8.value;

$(pictureId).attr("src",pictureURL);

$("#username" + line8.value).text(accountName);

//alert(data.d.UserProfileProperties.results[11].Value);

//console.log(data.d.UserProfileProperties.results[11].Value);

//setUserProfilePropertyToControl(data.d.UserProfileProperties, "Department", '#department' + line8.value);

},

error:function(jqxr,errorCode,errorThrown){

console.log(jqxr.responseText);

}

});

})

.fail(function(error){

console.log(JSON.stringify(error));

});

},

error: function (data, errorCode, errorMessage) {

alert(errorMessage)

}

});

function setUserProfilePropertyToControl(userProfileCollection, userProfileKey, controlId)

{

if (userProfileCollection.results.length > 0) {

for (var i = 0; i < userProfileCollection.results.length; i++) {

if (userProfileCollection.results[i].Key === userProfileKey) {

$(controlId).text(userProfileCollection.results[i].Value);

}

}

}

}

function ensureUser(accountName)

{

var endpointUrl = \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/ensureUser('" + encodeURIComponent(accountName) + "')";

return $.ajax({

url: endpointUrl,

type: "POST",

contentType: "application/json;odata=verbose",

headers: {

"Accept": "application/json;odata=verbose",

"X-RequestDigest": $("#\_\_REQUESTDIGEST").val()

}

});

}

});

\_#-->

</div>

</body>

</html>

**Control template:**

<html xmlns:mso="urn:schemas-microsoft-com:office:office" xmlns:msdt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882">

<head>

<title>CoxOne Article List</title>

<!--[if gte mso 9]><xml>

<mso:CustomDocumentProperties>

<mso:TemplateHidden msdt:dt="string">0</mso:TemplateHidden>

<mso:MasterPageDescription msdt:dt="string">This is the default Control Display Template that will list the items. It does not allow the user to page through items.</mso:MasterPageDescription>

<mso:ContentTypeId msdt:dt="string">0x0101002039C03B61C64EC4A04F5361F385106601</mso:ContentTypeId>

<mso:TargetControlType msdt:dt="string">;#Content Web Parts;#</mso:TargetControlType>

<mso:HtmlDesignAssociated msdt:dt="string">1</mso:HtmlDesignAssociated>

</mso:CustomDocumentProperties>

</xml><![endif]-->

</head>

<body>

<!--

Warning: Do not try to add HTML to this section. Only the contents of the first <div>

inside the <body> tag will be used while executing Display Template code. Any HTML that

you add to this section will NOT become part of your Display Template.

-->

<script>

$includeLanguageScript(this.url, "~sitecollection/\_catalogs/masterpage/Display Templates/Language Files/{Locale}/CustomStrings.js");

</script>

<!--

Use the div below to author your Display Template. Here are some things to keep in mind:

\* Surround any JavaScript logic as shown below using a "pound underscore" (#\_ ... \_#) token

inside a comment.

\* Use the values assigned to your variables using an "underscore pound equals"

(\_#= ... =#\_) token.

-->

<div id="Control\_List">

<!--#\_

if (!$isNull(ctx.ClientControl) &&

!$isNull(ctx.ClientControl.shouldRenderControl) &&

!ctx.ClientControl.shouldRenderControl())

{

return "";

}

ctx.ListDataJSONGroupsKey = "ResultTables";

var $noResults = Srch.ContentBySearch.getControlTemplateEncodedNoResultsMessage(ctx.ClientControl);

var noResultsClassName = "ms-srch-result-noResults";

var ListRenderRenderWrapper = function(itemRenderResult, inCtx, tpl)

{

var iStr = [];

iStr.push('<li>');

iStr.push(itemRenderResult);

iStr.push('</li>');

return iStr.join('');

}

ctx['ItemRenderWrapper'] = ListRenderRenderWrapper;

\_#-->

<ul class="cbs-List">

\_#= ctx.RenderGroups(ctx) =#\_

</ul>

<!--#\_

if (ctx.ClientControl.get\_shouldShowNoResultMessage())

{

\_#-->

<div class="\_#= noResultsClassName =#\_">\_#= $noResults =#\_</div>

<!--#\_

}

\_#-->

</div>

</body>

</html>

<script type="text/javascript">

var g\_duration = 130;

var g\_iisLatency = 2;

var g\_requireJSDone = new Date().getTime();

</script>