# Office AMS: TAXONOMY MENU

|  |  |
| --- | --- |
| Summary: | Applies to: |
| This sample shows how to create a menu that is populated from Term Store using JavaScript CSOM. The menu uses the language labels on terms and shows translated navigation nodes depending on users preferred language in user profile. This solution works cross site collections.  To set up this sample a provider-hosted app using .NET CSOM creates a term group, a term set with terms in term store. Also JavaScript files are uploaded to host web and script links are added. | * Office 365 Multi Tenant (MT) * Office 365 Dedicated (D) * SharePoint 2013 on-premises |
| Solution: | Contoso.TaxonomyMenu, version 1.0 |
| Author: | Johan Skårman (Microsoft) |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // THIS CODE IS PROVIDED \*AS IS\* WITHOUT WARRANTY OF  // ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING ANY  // IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR  // PURPOSE, MERCHANTABILITY, OR NON-INFRINGEMENT.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |



# STEP 1: Set Up Term Store

The first step is to set up some terms in term store that can be used by the navigation. This is all done using .NET CSOM in the TaxonomyHelper class.

## Setup TERM STORE LANGUAGES

To start with the code checks to see that all required languages (in this example English, French, German and Swedish) are enabled in Term Store. If not, the languages are added. This will enable language specific term labels to be created.

var languages = new int[] { 1031, 1033, 1036, 1053 };

Array.ForEach(languages, l => {

if (!termStore.Languages.Contains(l))

termStore.AddLanguage(l);

});

termStore.CommitAll();

clientContext.ExecuteQuery();

## Create Term Group

Before setting up the terms the code checks to see if a Term Group with a specific ID exists. If not, the group is created.

termGroup = termStore.CreateGroup("Taxonomy Navigation", groupId);

## Create Term Set

Next, the code checks to see if a Term Set with a specific ID exists. If not, the Term Set is created. As part of the creation a custom property (\_Sys\_Nav\_IsNavigationTermSet) is set to True. This is the same as checking “Use this Term Set for Site Navigation” on the Term Set in Term Store Management Tool. The property is set so that the Navigation tab will be shown on Terms to make it easy to administer Term URLs. The code also loads the Terms collection on the Term Set for later use.

termSet = termGroup.CreateTermSet("Taxonomy Navigation", termSetId, 1033);

termSet.SetCustomProperty("\_Sys\_Nav\_IsNavigationTermSet", "True");

clientContext.Load(termSet, ts => ts.Terms);

## Create TERMS

Next, the code creates the terms. For each term three language specific labels are also created. Also the custom property \_Sys\_Nav\_SimpleLinkUrl is set which is the same as setting “Simple Link or Header” on terms in Term Store Management Tool.

var term = termSet.CreateTerm(termName, 1033, Guid.NewGuid());

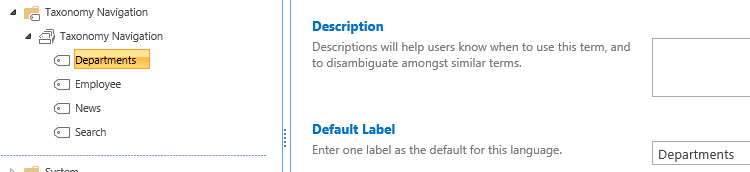
term.CreateLabel(termNameGerman, 1031, false);

term.CreateLabel(termNameFrench, 1036, false);

term.CreateLabel(termNameSwedish, 1053, false);

term.SetLocalCustomProperty("\_Sys\_Nav\_SimpleLinkUrl", clientContext.Web.ServerRelativeUrl);

When the first step is completed the Term store should look like this in Term Store Management Tool:



# Step 2: Add Scripts

To demonstrate the menu on the host web taxnav.js and JQuery are uploaded to the Site Assets library. Script links are also added using CustomActions so that the files will be referenced in master page. This is all done using .NET CSOM. If building a custom branding solution with master page the links could of course be added directly to the master page.

var customActionTaxonomy = existingActions.Add();

customActionTaxonomy.Description = "taxonomyNavigationScript";

customActionTaxonomy.Location = "ScriptLink";

customActionTaxonomy.ScriptSrc = "~site/SiteAssets/taxnav.js";

customActionTaxonomy.Sequence = 1010;

customActionTaxonomy.Update();

# menu CREATION

The menu is created using JavaScript CSOM and JQuery.

## GETTING USER PREFERRED LANGUAGE

First the code checks the current users profile to see preferred language. Because the property value consists of language codes (e.g. en-US, sv-SE) and Term Store uses LCID (1033, 1053) the language codes are translated using a key-value array. In production code the result could be cached to minimize client callbacks.

var peopleManager = new SP.UserProfiles.PeopleManager(context);

var userProperty = peopleManager.getUserProfilePropertyFor(targetUser, "SPS-MUILanguages");

## GETTING TERMS

After the code has checked that the users preferred language is also one of the languages set up in Term Store, the terms are loaded as well as the labels for that language.

while (termEnumerator.moveNext()) {

var currentTerm = termEnumerator.get\_current();

var label = currentTerm.getDefaultLabel(lcid);

termItems.push(currentTerm);

termLabels.push(label);

context.load(currentTerm);

}

Finally a HTML unordered list is created and added to the DIV element with ID DeltaTopNavigation. In production code the result could be cached to minimize client callbacks.

var linkName = termLabel.get\_value() != 0 ? termLabel.get\_value() : term.get\_name();

var linkUrl = term.get\_localCustomProperties()['\_Sys\_Nav\_SimpleLinkUrl'];

The end result should look like this:



Or like this when setting French as preferred language:

