Client: Opus Agency
Year: 2015
Description: I was sole author on a procedural guide for internal Quality Assurance Analysts. It contains guidelines for setting up and testing customer integrations with a particular vendor, referred to here as "CMS", for Content Management System.

CMS Integration

Testing Guide

General Information

Overview

CMS is an Opus partner that offers tools for session and speaker management. We typically engage CMS on large conferences with many session offerings so we can use their session management interface to import, edit, and categorize event sessions, and enter information about the speakers who will present at those sessions. The overall purpose is to prepare sessions for display on Module X.

In these engagements, CMS, rather than Internal System, becomes the system of record for Module X sessions. Information from the CMS API gets pulled down into our database on a regular schedule, generally every five to ten minutes, and will overwrite any changes that may have been made in Internal System.

[...]

Execute Your Tests

CMS Data Verification

If Opus carried out a data import into CMS, then you should request access to the source data from the client in order to compare it side-by-side with the data in CMS. The requirements document(s) for the project should specify how the client-supplied session data is to be mapped to CMS data fields.

CMS session data is divided into three parts: Standard Fields, Custom Fields, and Session Management.

Standard Fields

Most of these are self-explanatory, such as title and description. Another standard field is “Track”, which allows Module X users to filter the session list by subject Internal System. Potential Track values are user-defined, so you should verify that they were entered correctly by the person who did the import.

A session can have zero to many speakers assigned to it. Speakers are created as “users” in CMS and must be assigned a type of “Speaker” in order to become available to add to sessions. If the client has provided speaker information, make sure they have been imported correctly and assigned to the right sessions.

Custom Fields

Author: Kristen Martin
Email: kristenmartin503@yahoo.com

Custom fields can be created to accommodate any other type of categorization the client wants. For example, they may group sessions by “Subject” or “Experience Level”. The person who did the import should have created these custom fields as needed, set up the correct options within them, and assigned them to the appropriate sessions.

Session Management

This section includes Room and Time Slot. Verify that these Internal System the places and times provided by the client. If the sessions are supposed to have a maximum capacity, then the capacity needs to be tied to the Room definition in CMS.

CMS Data Integration

Next you will have to execute the SQL job to pull down the data from the CMS API. Jobs are housed on Internal Server Z. Follow these steps to execute:

1. Connect to Internal Server Z in SQL Server Management Studio Object Explorer
2. Open SQL Server Agent > Jobs
3. Right-click the correct job name (should include the client’s name and “CMS”)
4. Select “Start Job at step…”
5. Wait for confirmation that job is complete

If an error occurs, report it to the DBA.

CMS API Information

When data is entered or updated in the CMS interface, it takes three to five minutes for it to become available in their API. You can use a Chrome extension called POSTMan (or your preferred REST API client) to call the API directly and verify that your test data is available before running the SQL job.

How to set up a CMS API request in POSTMan:

1. Open POSTMan from your Google Chrome app launcher
2. Enter the request URL **[redacted]**
    a. The “Sessions” endpoint returns basic session data only. For other endpoints, refer to the CMS API documentation at [internal URL]
3. Make sure the request type is “GET”
4. In the “Headers” section, add the following three headers:
    a. “Accept” with a value of “application/xml”
    b. “content-type” with a value of “application/xml”
    c. “Authorization” with an authorization key value to be obtained from CMS; this should be listed somewhere in the documentation for the project
5. Click the “Send” button to initiate the request
6. If multiple pages of results are returned (i.e. more than 125 records), you can append “?$skip=xx” to the request URL where xx is a number divisible by 125
    a. For example, use “?$skip=250” to skip the first two pages of results and view the third page

[...]

Author: Kristen Martin
Email: kristenmartin503@yahoo.com