

Modern Campus Catalog

Widget API Training Webinar

modern
campusTM

Confidentiality Note: This document contains confidential information, which is proprietary to Modern Campus. In consideration of receipt of this document, the recipient agrees to maintain such information in confidence and not disclose this information to any party outside of the recipient's organization. This information may not be reproduced or displayed to any other party without the written consent of Modern Campus.

Modern Campus shall not be held liable for any technical or editorial errors and omission contained herein, nor any damages directly or indirectly arising from this document.

© 2024 Modern Campus. All rights reserved. Confidential.

Version Control

Version	Description of Revisions	Date	Author
2	Rebranding	June 2024	E. Ralston

TABLE OF CONTENTS

Edit the HTML file	6
Link	6
List	7
Content	7
Catalogs	7
Courses	9
Programs	10
Loading Icons	10
Entities (Hierarchy)	11
Filters	12
Pages	12
Planner	12
Why can't I select the "Content" Widget?	13
My code all appears to be correct – but the widget will not load!	13
Lancaster Bible College	14
College of Charleston	14
SUNY Cortland	14

What is the Widget API?

API stands for Application Programming Interface – when a company like Modern Campus provides an API to clients, it means they have created a set of dedicated URLs to return pure data responses. The Widget API has been designed to simplify this and allows someone with HTML knowledge to take academic data currently housed within Modern Campus Catalog and display it on another – without needing to enter the data twice or be proficient in writing API calls.

If you have ever created a link to your current catalog – you know that each year, those links must be updated when you publish a new version. Using the Widget API is a way to generate a dynamic link instead that can point to the most recent version without having to create those changes each year. You can create widgets that display individual links to catalogs, courses, programs, specific custom pages, even filters or degree planners – that stay up to date year after year. This tool can help you create a no maintenance catalog landing page with links to all published catalogs or create an individual link to a program or course that will automatically point to the latest version when next year's catalog is published.

We understand that the information presented on your website is important and needs to be up to date and load quickly. The Widget API will create widgets that will load asynchronously on your sites, which means they will not affect the load time of your pages.

Note: This presentation will require a functional understanding of HTML and CSS, and while JavaScript is not required – it will clarify how the elements work together. For information on learning how to work with these languages, visit: <https://www.w3schools.com/>

Enable the Widget API

If you have not yet enabled the Widget API, you can have it enabled by contacting our client care team via the "Support" widget in the lower-left corner of your Publisher – or by accessing the Remote Services Module, clicking on the Widget API tab, and then clicking "Request API." Our team will send you a Release Candidate agreement to be signed and returned at which point the Widget API will be enabled.

Get Started with HTML

For this presentation, you may choose to follow along in the provided files – or you may choose to simply observe for now and try out the features later. The zipped folder will contain 8 files ending with .html, where we will be conducting the changes.

Before we open any of them; or if you are choosing to set up or modify your own existing HTML files – you'll need to ensure that your pages are set up with the latest design and development standards. This means:

- Make sure that you are using an HTML5 DOCTYPE (If you follow along with our template, this will be declared in the document for you. If you are unsure what this is referring to, you can review information on the doctype at: https://www.w3schools.com/html/html5_intro.asp).

- Utilizing the viewport meta tag (also included in the template).

Edit the HTML file

For this presentation, you'll see me using Notepad++ to manage the editing of my HTML files. This is a free program that you can download – you aren't required to use this as you can use a simple text editor like Notepad (which is included on PCs by default) or you could choose to use a much more robust web designing tool like Dreamweaver. Notepad++ works as a great, free resource that highlights the HTML in a much more readable format than just using Notepad.

The first file we'll open is going to be titled index.html. You'll need to open this in both your browser – and in Notepad++. We recommend using Firefox or Chrome, as these are the primary browsers suggested when working within Acalog. For this presentation, I'll be using Firefox.

When reviewing this file in the browser, we see a very simple template. You'll notice a section I have set up for "Link Widget" as well as for "List Widget" – both of which are currently displaying an error. This is the page we'll be modifying and adding different options to in order to see how the widget changes the contents brought in – but the other examples across the top – such as catalog, course, or program – and you'll see that it's actually bringing in some content for us.

I'm going to return us back to the "Start" tab, which if you'll notice my URL bar – is also the index.html file again.

Widget Type Options

There are three types of Widgets that you can create: a link, a list, or a content widget. The availability of each one will vary based on the type of content you are attempting to reference.

Link

A Link Widget will point to a single item on your Gateway and is returned as a clickable link.

You could use this to establish a link that pulls the most current catalog type and points to a course name, so that each year when you publish a new catalog of that type, as long as the name of the course is the same, this link would then point to the course in the new catalog.

As an example, let's look at the Courses tab of the template.

Clicking the link will direct you to the current published catalog, "Catalog Manager Training" and to the course, "AG 335 – Wildlife Management I".

Now next year, when I need to archive this catalog – I can go into my Publisher, mark this as archived, and then publish my "2018-2019 Undergraduate Webinar Catalog." When I refresh my website and click the link again – notice how it now directs me to the same course in the newly published catalog!

List



A List Widget will return a bulleted list of items that will link to the Gateway.

Like the Link widget, it can also be set to reference elements like a course type, or a catalog type. If you look again at our Courses tab, you'll find the List Widget is currently pulling a list of courses with the "Accounting" type from the catalog with the type of "Webinar." When a

new catalog is published, this list would update as well.

Note: If you set the Widget to include specific elements such as the Legacy ID, or a Name, your list Widget may only return ONE result as you have narrowed the search results down to a single item.

Content

A Content Widget will take the entire contents of an item and display the full text. Similar to how an inline PermaLink can display content directly in the catalog the content widget can be used to display the requested content directly onto the page with no link required.

For example, on our Courses tab, you'll see that it is pulling in the information for "ANS 118 - Introduction to Animal Science". As this widget is also set to reference the catalog type of "Webinar", even if we publish a new catalog we'll still see this course being pulled in as long as the name doesn't change.

Note: Content Widgets are only available for courses, programs, entities, filters, and pages.

Widget API – Sandbox

The Widget API provides a sandbox environment to help you in writing the call and testing to ensure the information returned is what you expected. We'll be referring to this sandbox often during this presentation.

To access the sandbox, your Widget API must be enabled. Once enabled, you can access the Remote Services Module, click on the Widget API tab, and then click on Sandbox.

Widget Data Options

Note: All widgets contain an option for "ID" and "Legacy ID". When referring to specific items, be sure to always use "Legacy ID" – the "ID" option will not return any results.

Next, we'll begin reviewing the different types of data that you can pull through the Widget API. We'll construct the code in the sandbox, and then enter the information into our **index.html** file to see it in action.

Catalogs

The Catalog Widgets can be provided through a Link or a List.

Options available:

- Link Widget

- Catalog Legacy Id ("456", which is the Catalog OID for a specific catalog)
- Catalog Type (Undergraduate)
- Catalog Name (Undergraduate Catalog 2018-2019)
- List Widget
 - Catalog Type

To generate our widget, we'll need to select either Link or List to start. Let's start with Link.

Once you have selected an option, you'll need to indicate the type of data you want to retrieve – in this case, Catalogs.

Next, we need to indicate a Catalog Filter to help narrow it down – else it will simply return the first published catalog. As this is a Link Widget, we can choose Legacy ID – which would be a catalog's OID, a catalog type – like graduate or undergraduate and would retrieve the current published catalog of that type, or catalog name – where you would need to enter the catalog's exact name.

Using Legacy ID or Name requires a specific option but using type would allow it to select a different catalog whenever a new one is published. For this example, let's select Catalog Type as the filter, and then enter "Widget."

Note that as we made changes, the code updated. We can click Test Code to preview the link and verify that we've got the right content. If we're satisfied – we can copy this code – and then open our **index.html** file in Notepad++, and locate the section titled 'Link Widget. Replace the code with the one we've just copied, which should now read:

```
<a href="#" class="acalog" data-acalog-data="catalogs" data-acalog-catalog-type="Widget">Loading...</a>
```

The final step will be to update the JavaScript string at the bottom of the page to allow it to bring in the correct content. At the bottom of the page is a short script that begins with `$('.acalog').acalogWidgetize` and will need to be updated with the information you're pulling.

You can again refer to your Widget API documentation in the Publisher to locate the information needed – head to the catalogs tab. Here you'll find all of the information about the types of calls that can be made, and if we scroll down to 'Catalog Type' we'll find the JavaScript information. The example in the documentation shows 'Undergraduate', but as we've used 'Widget' we'll need to change that. Our new script should look as follows:

```
$(function() {  
  $('.acalog').acalogWidgetize({  
    gateway: 'http://training2.catalog.training.acalog.com',  
    data: 'catalogs',  
    catalogType: 'Widget'  
  });  
});
```

The last step is to save the index.html file, then refresh it in the browser – you should see your new link appear!

Courses

Courses can pull in as a link, a list, or as a content widget. As we looked at a link widget for a catalog, we'll look at a list widget for the courses example.

To set up the link, we'll start by returning to our sandbox. We'll select List, then Courses.

For Catalog Filter, I'm going to use Type again, and set it to "Widget." I'm using type a lot so that each time I publish a new catalog of the same type, the link can continue to work rather than needing to be changed.

Next, I'll add Course Filters. If I choose not to add any, it'll return a list of all courses in the catalog. If I'd like to narrow it down by Course Type, I can select that option and enter any item type – like Astronomy for example - to see those results.

You can choose more than one option – but if you apply multiple items that would return 0 results (for example: Choosing "Art" type and then entering a prefix of a course that is not that type, like CHEM) you will get no results.

The Course Legacy Id will point to a specific course as will the Course Name. You may not find it beneficial to choose both options due to the redundancy.

As before, once you have generated the code, we'll click copy:

```
<ul class="acalog" data-acalog-data="courses" data-acalog-catalog-type="Widget" data-acalog-course-type="Astronomy"><li>Loading...</li></ul>
```

Like before, you'll need to paste this into your index.html file over the section titled 'List Widget'. Then, we'll need to update the JavaScript at the bottom. In the API Documentation we can refer to the Courses tab to see how to update the script.

Note that the script doesn't vary much between Widgets – but when we save the file with the new script below, it will still pull the information from our Catalog widget, in addition to our course widget. In the script below, the new information has been added in bold, and then this has been pasted into our file.

```
$(function() {  
  $('acalog').acalogWidgetize({  
    gateway: 'http://training2.catalog.training.acalog.com',  
    data: 'catalogs',  
    data: 'courses',  
    catalogType: 'Widget',  
    courseType: 'Astronomy'  
  });  
});
```

Note: You are not able to pull Inactive-Hidden courses or pull archived catalogs from the "Catalog Type" Catalog Filter.

Filter names must be exact matches. "Accounting" will return results, but "Acc" will not.

Programs

Now that you have seen a Link and a List Widget, we'll use a program to demonstrate the Content Widget. The Content Widget will display the full item we're calling – rather than just providing a clickable link. You will still need to define the catalog that you are pulling from, just like we did with courses, but again you can use type to allow it to roll over when a new catalog is published.

If we return to the sandbox again, notice you can't choose content right away. The default for data is 'Catalogs', and you can't use Content with that option. You'll have to first change the data to 'Programs', and then change the Widget to Content.

For Catalog Filter, I'm going to use Type again, and set it to "Widget." For the program, I'm going to go ahead and reference the program by a specific name – since I want a very particular one. You could also use Program Legacy ID – but keep in mind that when the catalog is published for next year, the OID will change and the link won't work. Using name instead allows it to continue functioning.

We can test the link to preview it, then copy it and insert the new code into our page where we see the 'Content Widget' information:

```
<div class="acalog" data-acalog-data="programs" data-acalog-catalog-type="Widget" data-acalog-program-name="Widget Building Program">Loading...</div>
```

Like before, you'll need to paste this into your **index.html** file over the section titled 'Content Widget'. Then, we'll need to update the JavaScript at the bottom. In the API Documentation we can refer to the Programs tab to see how to update the script.

Under the Program Name section, you'll see we need to add the information to include the programs data, and the program filter, so our new script will be as follows:

```
$(function() {  
  $('acalog').acalogWidgetize({  
    gateway: 'http://training2.catalog.training.acalog.com',  
    data: 'catalogs',  
    data: 'courses',  
    data: 'programs',  
    catalogType: 'Widget',  
    courseType: 'Astronomy',  
    programName: 'Widget Building Program'  
  });  
});
```

Once you save and refresh, you'll see the new widget pulling in our program information at the bottom of the page.

Loading Icons

You can also choose to place a loading icon or image to display while your widgets load on the page. It's common to see text such as "loading..." Or a small animated icon. It's not recommended to using anything other than text or basic HTML as this may impact the overall load time of your widget.

For example, in the zipped folder provided, you'll find a file titled 'loading.gif'. This is an animated gif that can serve as a loading icon, and we can insert this into our widget.

If we navigate to the Advanced tab of the Widget API documentation, you'll see there is a section titled 'Placeholder', which provides three lines of code we can use, depending on the widget we'd like to place it within.

The following line has already been edited to include the image based on its location in the folder:

```

```

If we take this link, and place it within any of our Widgets, replacing the existing "Loading...." text, we'll see the following:

Link

```
<a href="#" class="acalog" data-acalog-data="catalogs" data-acalog-catalog-type="Widget"></a>
```

List

```
<ul class="acalog" data-acalog-data="courses" data-acalog-catalog-type="Widget" data-acalog-course-type="Astronomy"><li></li></ul>
```

Content

```
<div class="acalog" data-acalog-data="programs" data-acalog-catalog-type="Widget" data-acalog-program-name="Widget Building Program"></div>
```

Once these have been entered, save the page and refresh. You should now see the loading wheel while the page loads the widgets!

Entities (Hierarchy)

You will need to define the catalog that you are pulling from.

Note: You can choose more than one option. The Entity Legacy Id will point to a specific hierarchy entity as will the Entity Name. You may not find it beneficial to choose both options due to the redundancy.

Options available:

- Link Widget
 - Entity Legacy Id
 - Entity Name
- List Widget
 - Whether you choose Entity Legacy Id or Entity Name, it will just pull one hierarchy entity like the Link Widget.
- Content
 - Entity Legacy Id

- Entity Name

Filters

You will need to define the catalog that you are pulling from.

Note: You can choose more than one option. The Filter Legacy Id will point to a specific filter as will the Filter Name. You may not find it beneficial to choose both options due to the redundancy.

Options available:

- Link Widget
 - Filter Legacy Id
 - Filter Name
- List Widget
 - Whether you choose Filter Legacy Id or Filter Name, it will just pull one filter like the Link Widget.
- Content
 - This will not pull the filter inline, but instead gives you a link similar to the Link Widget.

Pages

This is your custom page data and you will need to define the catalog that you are pulling from.

Note: the content has to be active-visible/active-hidden.

You can choose more than one option. The Page Legacy Id will point to a specific custom page as will the Page Name. You may not find it beneficial to choose both options due to the redundancy.

Options available:

- Link Widget
 - Page Legacy Id
 - Page Name
- List Widget
 - Whether you choose Page Legacy Id or Page Name, it will just pull one custom page entity like the Link Widget.
- Content
 - Page Legacy Id
 - Page Name

Planner

You will need to define the catalog and program that you are pulling from.

Note: You can choose more than one option. The Program Legacy Id will point to a specific program as will the Program Name. You may not find it beneficial to choose both options due to the redundancy.

Options available:

- Link Widget – will show the program name, but links to the Print Degree Planner
- List Widget
 - Program Type
 - Program Degree Type
 - Program Code

Troubleshooting

How to find the OID? (Legacy ID)

`https://training2.training.acalogadmin.com/courses/view_edit_course.php?courseoid=521297`

One way to determine this is to look at your URL bar. Find the corresponding code below within the URL text.

- catoid- catalog
- courseoid – course
- progoid – program
- entoid- hierarchy entity
- navoid – filter or custom page

Another option, specific to courses, is to run a course export. There will be a “Course OID” column listing every course in the catalog’s OID number.

Why can’t I select the “Content” Widget?

The Content Widget is only available for courses, programs, entities, filters and pages. You will be unable to select data for catalogs or the degree planner options.

My code all appears to be correct – but the widget will not load!

The most common issue here is missing the second set of brackets within the JavaScript at the bottom of the page. Make sure BOTH brackets are present.

```
$(function() {  
    $('.acalog').acalogWidgetize({  
        gateway: 'http://training2.catalog.training.acalog.com'  
    });  
});
```

The second common issue is the formatting of the quote around the data.

```
$(function() {
  $('.acalog').acalogWidgetize({
    gateway: 'http://training2.catalog.training.acalog.com'
    data: 'courses',
    catalogType: 'Widget'

  });
});
```

Note the difference between the quotes around courses and widget. Widget is correct.

What Other Schools are Doing

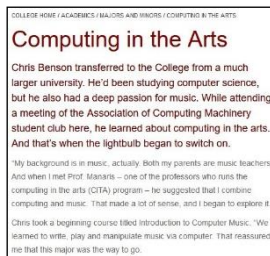
Lancaster Bible College

Use the Widget API to display the main courses for each major per year on their institutional site.

<https://www.lbc.edu/undergraduate/academics/majors/business-administration/index.html>



College of Charleston



Eliminate annually relinking to the catalog. They chose the Widget API because it requires minimal technical support and the scope of the work involved. It allows for their decentralized content management practice to continue.

<http://www.cofc.edu/academics/majorsandminors/computingarts.php>

SUNY Cortland

Use the Widget API to replace links with content that pulls directly from the current catalog into their CMS.

<http://www2.cortland.edu/departments/kinesiology/undergraduate-courses.dot>

