

Webpart to API Converter

Kentico has many great tools that allow users to display their web content in many different ways. However, there lacks a way to leverage this to create Web API calls. This web part converter bridges that gap, by allowing you to select a Web Belmont on a page to act as that page's only output. Given given the WebpartID的理想 (coupled with the绵内容-type), it clears all the response content except what that web part renders.

- Apply to a Smart Search results to create an api for ajax “suggested search” results.
- Apply to a Repeater or Hierarchy Viewer to create an API to retrieve objects in JSON
- Apply to a Custom Table Repeater to turn your database into a queryable API or XML feed.
- Apply to an Events Repeater to create an events subscription page or event file (ics)

It should be 0.00otte that some of the functionality of this Web Part are similar to the XML Repeater and Atom Repeater, but this Web Part does not rely on a Data Source, but instead can be applied to a wider variety of Web Parts and can output in a wider variety of formats and Content Types (ex: JSON)

Application

The Web part to API has a 0.00 couple options.

- **- **0.0000000000000000000000000000000000000干了**
- **0.0完**
- **0.0000000000c****

The Webpart to API第一次 0.000000قات does not alter the content (except when the content type is application/json, in which case the code will remove any HTML elements, requiring content to be HTML encoded prior. See transformation samples), so it is important that you set any transformations to output exactly as you want it to display in the API.

Included with this webpart is a Pageimini 0.0昕 (WebpartToAPI 0.00nese) that contains sample transformations for each of the below Sample/Cases and should get you started.

There is one Code file included (WebpartToAPIHelper.cs) included which contains some functions for the ICS rendering.

Any Questions or bugs, please contact tfayas@avastonetech.com

Installation

<0.0000 Sheets" data-label="List活" data-label="List不比" data-label="这些数据" data-label="Listshaw" data-label="List" data-label="List-Group">

- 1: Sites -> Import site or objects. Upload the WebpartToAPIConverter.zip and hit next.
- 2: Please be sure when importing this asset to check “Import Files (recommended)” and all 3 checkboxes below.
- 3: Ensure under All Objects -> Global Objects -> Development, that all items are selected.

Samples / Case Instructions

To help users understand how to set their transformations up, there is a Custom Doc Type included with the Web Part that contains sample transformations for various commonly used items, such as Repeaters, Hierarchy Viewers, and Smart Search Results.

For Smart Search Results

1. Add a Smart Search Results Webpart to a page, setting it up as you would normally.
2. Add the following configurations:
 - a. Set the transformation to `WebpartToAPI.SampleTransformations.SmartSearchResultsJSON` - OR - `WebpartToAPI.SampleTransformations.SmartSearchResultsXML`
 - b. Set the "No results text" to: `{"Title": "No Results Found"}` or `</SearchResult>`
3. Add a WebpartToAPI Converter Web Part
4. Configure as follows:
 - a. Set the WebpartID to the ID of the Smart Search Results
 - b. For JSON output:
 - i. Set the Content Type to `application/json`
 - ii. For the Content Before Results, set to: `{"SearchResults":[`
 - iii. For the Content After Results, set to: `]}`
 - c. For XML Output:
 - i. Set the Content Type to `text/xml`
 - ii. For the Content Before Results, set to: `<SearchResults>`
 - iii. For the Content After Results, set to: `</SearchResults>`
5. Lastly, reference that page as you would a normal search results, with the `searchtext` and `searchmode` in the URL.
 - a. Ex: `/MySearchAPI.aspx?searchtext=Some%20Search%20Text&searchmode=anyword`

NOTE: For Smart Search Results, the Smart Search Results Web Part will force its outer `<div>` into the results. The WebpartToAPI strips all HTML for the *JSON results*, so this is removed, however for `text/xml` and other content types, it is not stripped. It should not affect your API however as it is still valid XML, and will render similar to the below:

```
<SearchResults>
  <div id="p_lt_ctl01_SmartSearchResults_srchResults_pnlSearchResults">
    <SearchResult>...</SearchResult>
    <SearchResult>...</SearchResult>
  </div>
</SearchResults>
```

The image shows two side-by-side screenshots of the 'Web part properties' dialog boxes in a SharePoint environment.

Left Window: Web part properties (Smart search results)

- Search options: Basic
- Typo tolerant search:
- Search condition: [Empty]
- Search sort: [Empty]
- No results text: `<SearchResult />` (labeled **Xml**)
- No results text: `{"Title": "No Results Found"}` (labeled **Json**)

Right Window: Web part properties (WebpartToApiConverter)

- Webpart ID: SmartSearchResults
- Content Type: text/xml (labeled **Xml**)
- Render as Attachment with Filename: [Empty]
- Content Before Results: `<SearchResults>` (labeled **Xml**)
- Content After Results: `</SearchResults>` (labeled **Xml**)
- Content Type: application/json (labeled **Json**)
- Render as Attachment with Filename: [Empty]
- Content Before Results: `{"SearchResults":[` (labeled **Json**)
- Content After Results: `]}` (labeled **Json**)

For Hierarchy Viewers

For Hierarchy Viewers, you will most likely need to adjust the Sample Transformations to match the Item Structure you are referring to. Please feel free to clone the `WebpartToAPI.SampleTransformations.HierarchyViewerJson` or `WebpartToAPI.SampleTransformations.HierarchyViewerXML` and adjust to your needs. Be sure to adjust the sub items as well.

1. Add a Hierarchy Viewer Web Part to the Page, setting up as you would normally.
2. Add the following Configurations:
 - a. For JSON
 - i. Set the Transformation to `WebpartToAPI.SampleTransformations.HierarchyViewerJSON`
 - ii. Set the No Data Behavior to either Hide if no record found, or a No Record found text to some valid JSON like : `{"Error": "No Documents Found"}`
 - b. For XML
 - i. Set the Transformation to `WebpartToAPI.SampleTransformations.RepeaterArticleXML`
 - ii. Set the No Data Behavior to either Hide if no record found, or a No Record found text to some valid XML like: `<Article></Article>`
 - c. Disabled Paging
3. Add a WebpartToAPI Convert Web Part
 - a. Set the WebpartID to the ID of the Repeater
 - b. For the Content type, set to either `application/json`, or `text/xml`.
 - c. For the Content Before Results, this can be blank as the "Before" content is defined in the Level 0 Header Transformation in the Hierarchy Transformation
 - d. For the Content After Results, this can be blank as the "After" content is defined in the Level 0 Footer Transformation in the Hierarchy Transformation



For Events to ICS (Calendar)

In order to export a list of events (or a specific event) to Outlook or other Calendars, we format a repeater of events and export it either as a feed or as a file. Lastly, ICS integration is not an exact science, Outlook handles multiple events fine, where as other programs may not be able to import multiple events.

1. Add a Repeater Web Part to the Page, setting up as you would normally, pointing to either events or booking events (booking events will allow for start and end times).
2. Add the following Configurations:
 - a. Set the Transformation to:
 - i. WebpartToAPI.SampleTransformations.RepeaterBookingEventToICS - OR -
 - ii. WebpartToAPI.SampleTransformations.RepeaterEventToICS
 - b. Set the No Data Behavior to either Hide if no record found
 - c. Disabled Paging
3. Add a WebpartToAPI Convert Web Part
 - a. Set the WebpartID to the ID of the Repeater
 - b. For the Content type, set to either text/calendar.
 - c. If you wish to have this export as a file (vs. a feed)
 - i. For Render as Attachment with Filename, put a filename such as: events.ics
 - d. For the Content Before Results, put the below 2 lines:
 - i. BEGIN:VCALENDAR
 - ii. METHOD:PUBLISH
 - e. For the Content After Results, put the below line:
 - i. END:VCALENDAR
4. If you are exporting as a file, point the user to the page you just created to have them prompted to save the events file.
5. If you wish to have this as a web subscription instead, simply point to the page, using [webcal://](#) instead of [http://](#) as this will notify calendar clients that this is a events feed (so when the page changes as you add events, their clients should receive the updated events)

It should be noted that you can use the repeater's WHERE Filter and URL Macros to make the ICS export either a specific event (by EventID) or a list.

- Example /SpecialPages/MyEventExport.aspx?eventID=123
- WHERE Statement of either:
 - BookingEventID = '{? eventID ?}' OR " = '{? eventID ?}' - OR -
 - EventID = '{? eventID ?}' OR " = '{? eventID ?}'

The image shows two side-by-side screenshots of web part property panes. The left pane is titled 'Web part properties (Repeater)' and shows the 'Content filter' section with a 'WHERE condition' set to 'BookingEventID = '{? eventID ?}' OR " = '{? eventID ?}''. The right pane is titled 'Web part properties (WebpartToApiConverter)' and shows the 'General' section with 'Webpart ID' set to 'Repeater', 'Content Type' set to 'text/calendar', and 'Render as Attachment with Filename' set to 'events.ics'. Below this, there are two sections for 'Content Before Results' and 'Content After Results', both containing the ICS format strings: 'BEGIN:VCALENDAR', 'METHOD:PUBLISH', and 'END:VCALENDAR'. The top section is labeled 'As File' and the bottom section is labeled 'As Feed'.