



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Knowledge Center Developer's Guide

Improving Contact Us Form

12/14/2025

Improving Contact Us Form

Overview

You can utilize Knowledge Center capabilities to assist customers as they fill out forms on your corporate web site (for example, when providing feedback). This integration provides suggested answers to a customer's query by utilizing the **Category** selections, and keywords used in the **Subject** line. This simple guide will show you how Knowledge Center can be easily integrated into a Webform.

Webform integration

Before integration

Integrate the knowledge with a simple feedback form:

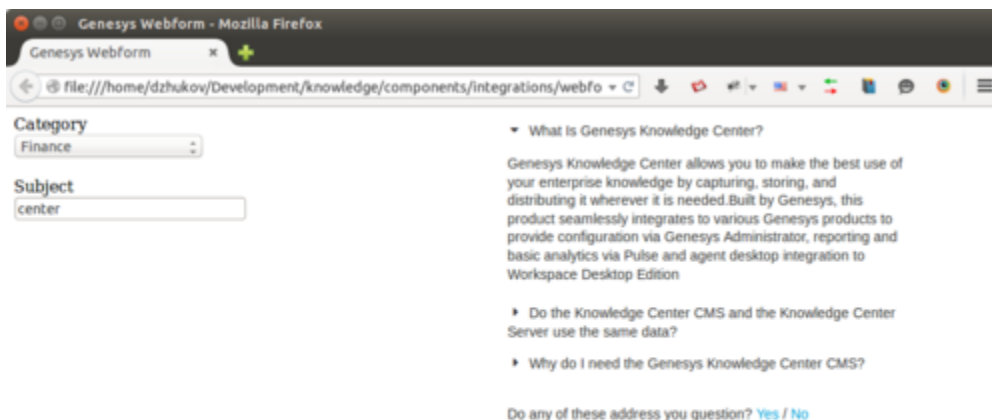
HTML

```
<div class="container">
  <div class="main">
    <div>
      <label>
        Category <br>
        <select class="categories"></select>
      </label>
    </div>
    <br>
    <div>
      <label>
        Subject <br>
        <input autocomplete="off" class="search" placeholder="What is knowledge
Center?" type="text">
      </label>
    </div>
  </div>
  <!-- MAIN DIV -->
  <div class="gkc-webform"></div>
</div>
```

Java Script

```
$(document).ready(function () {
    webform.init({
        host: 'http://%your_server_host%/gks-server/v1',
        categories: {
            'Finance': 'knowledgefaq',
            'Account': 'knowledgefaq',
            'Signing in': 'knowledgefaq',
            'Buying': 'knowledgefaq',
            'Shipping & tracking': 'knowledgefaq',
            'Booking trips ': 'knowledgefaq',
            'Gifts ': 'knowledgefaq',
            'Mobile ': 'knowledgefaq',
            'Email subscriptions ': 'knowledgefaq',
            'Restaurant reservations ': 'knowledgefaq'
        }
    });

    webform.markKbsDropdownWithMap('.gkc-kbs');
    webform.markSearchInput('.gk-search');
});
```



Example of simple integration.

Integration steps

1. Add all files (1 .css and 1 .js) from folder **<knowledge_center_server_root>\server\tools\webform** to site context. Core .js file applies only to rendering results ("Suggestions" window in the above figure).
2. Configure added script through **window.webform** variable:
 - Use **webform.init()** method to pass general options
 - Use **webform.markKbsDropdownWithMap()** to mark a specific <select> tag as a Categories

selector.

- Use **webform.markSearchInput()** to mark a specific <input> tag as to which Knowledge Center is performing the search.

Important

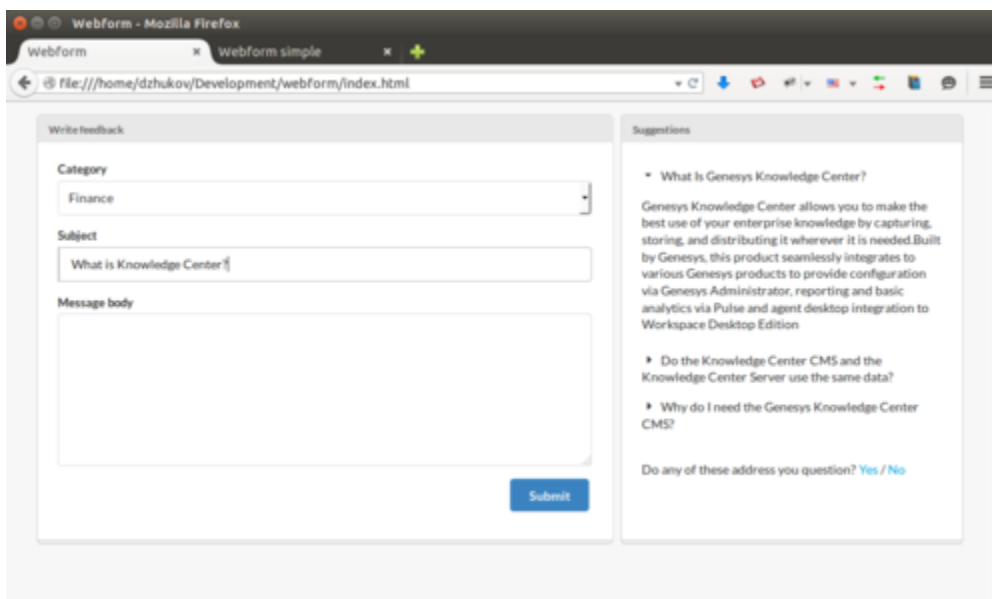
Examples of integrations can be found in <knowledge_center_server_root>/server/tools/webform/example folder.

After integration

As the result of this integration you now have a feedback form that pro-actively looks up the knowledge related to a customer inquiry and displays possible suggestions to the customer.

Important

WebForm can also contain a more complex demo based on **Semantic-UI** CSS-framework. See a complex integration at <knowledge_center_server_root>/server/tools/example/complex.html



Knowledge suggestions displayed

WebForm Widget API

Use the following information to integrate the WebForm Widget API on your html page.

webform.initialize(options)	Example
• Description: Configure the WebForm widget. • options Type: PlainObject A set of key/value pairs that configure the Agent. • host Type: string A network host where Knowledge API is stored. • categories Type: PlainObject A map of the predefined labels of categories. Keys are a labels and values are knowledgebase IDs.	<pre>webform.init({ host: 'http://192.168.66.176:9095/gks-server/v1', categories: { 'Finance': 'financefaq', 'Account': 'accounting', 'Signing in': 'webfaq', 'Gifts ': 'knowledgefaq', 'Mobile ': 'mobilefaq', 'Email subscriptions ': 'webfaq' } });</pre>
webform.markKbsDropdownWithMap(selector, callback)	Example
• Description: Create widget-dependent dropdown based on passed selector of <select> tag. This method uses categories passed to the .initialize method. • selector Type: jQuery Selector A string containing a selector expression to match elements against. • callback Type: Function() A function to be called after main operations.	<pre>webform.markKbsDropdownWithMap('.gkc-kbs', function () { \$('.gkc-kbs').dropdown(); });</pre>
webform.markKbsDropdown(selector, callback)	Example
• Description: Create widget-dependent dropdown based on passed selector of <select> tag. This method does not use categories passed to the .initialize method. It will load knowledge bases with labels directly from the Knowledge API. • selector Type: jQuery Selector A string containing a selector expression to match elements against. • callback Type: Function(kbs) A function to be called after main operations.	<pre>webform.markKbsDropdown('.gkc-kbs', function () { \$('.gkc-kbs').dropdown(); });</pre>

webform.markSearchInput(selector)	Example
• Description: Create widget-dependent search input based on passed selector of <input> tag. • selector Type: jQuery Selector A string containing a selector expression to match elements against.	<pre>webform.markSearchInput('.gk-search');</pre>
webform.getKbs(callback)	Example
• Description: Retrieves knowledge bases from the Knowledge API. • callback Type: Function(kbs) A function to be called after knowledge bases have been loaded.	<pre>webform.getKbs(function (kb) { console.log(kb) })</pre>
webform.makeSearch(query, callback)	Example
• Description: Searches documents from the Knowledge API based on query. • query Type: string User typed query string • callback Type: Function(documents) A function to be called after documents have been loaded.	<pre>webform.makeSearch('What is Knowledge Center?', function (documents) { console.log(documents) })</pre>