Implement Knowledge Advanced on Customer Portal

Contents

Implement Knowledge Advanced on Customer Portal 0

Overview of Knowledge Advanced Implementation 0

Before You Start 0

Implement Knowledge Advanced in a New Customer Portal Instance 0

Download the Reference Implementation 0

Replace the Customer Portal Standard Pages 0

Add the Knowledge Advanced Look and Feel 0

Promote Changes to Go Live 0

Implement Knowledge Advanced in an Existing Customer Portal Instance 0

Add the Knowledge Advanced Look and Feel 0

Replace the Standard Template 0

Edit the Standard Template 0

Add Knowledge Advanced Search to the Home Page 0

Replace the Home Page 0

Edit the Home Page 0

Replace or Edit the Account Overview Page 0

Replace the Account Overview Page 0

Edit the Account Overview Page 0

Add Knowledge Advanced Browse 0

Add the Knowledge Advanced Answer Detail Page 0

Add the Answer Detail Page 0

Edit the Answer Detail Page 0

Add the Knowledge Advanced Results Page 0

Enable Users to Manage Answer Notifications 0

Add Knowledge Advanced SmartAssistant 0

Enable Knowledge Advanced on the Chat Page 0

Replace the Chat Landing Page 0

Edit the Chat Landing Page 0

Implement the Knowledge Advanced Widgets 0

Update Widgets to the Latest Version 0

Set the Article Display on the Home Page 0

Configure Searches to Filter by Product or Category 0

Implement Suggested Searches 0

Configure Search Results Facets 0

Enable Users to Set the Number of Results Per Page 0

Disable Search by Document ID 0

Change the Style of Intent Answers on the Results Page 0

Configure a Product or Category Landing Page 0

Configure the Article Attributes 0

Configure Article Attributes on the Answer Page 0

Configure Article Attributes on the Ask a Question Page 0

See a Translated Version of an Article 0

Enable Users to Display Answers for All Content Types 0

Implement Content Type Subscriptions 0

Suppress SmartAssistant Filters 0

Enable Users to Open an Article Attachment 0

Enable Agents to Open an Article from Agent Desktop 0

Add Knowledge to Website Pages 0

Configure Knowledge Widget 0

Add the Widget to a Page 0

About Community Discussions 0

Add Community Discussions to the Home Page 0

Add Community Discussions to the Results Page 0

Add Recently Viewed Articles to the Social Page 0

Add Related Answers to the Social Page 0

Add Community Discussions to SmartAssistant 0

Implementing Knowledge Advanced on Mobile Customer Portal 0

Add Knowledge Advanced Search to the Mobile Home Page 0

Add Mobile Knowledge Advanced Browse 0

Add the Mobile Knowledge Advanced Answer Detail Page 0

Add the Mobile Knowledge Advanced Results Page 0

Add Knowledge Advanced SmartAssistant 0

# Implement Knowledge Advanced on Customer Portal

## Overview of Knowledge Advanced Implementation

You can implement Knowledge Advanced in Customer Portal to enable self-service users to quickly get answers to their questions without contacting your support team. They can search for additional information, rate the answers that they get, and if needed, escalate a question by submitting it to the support organization. You can implement Knowledge Advanced in a new instance or in an existing instance using the Knowledge Advanced reference implementation files.

The reference implementation is a set of PHP files that you can use to implement knowledge features. The reference implementation is located at cp/src/core/framework/views/pages/okcs/.

You must implement the Knowledge Advanced search (OKCSSearch) and browse (OKCSBrowse) functionality on different pages. Search and browse functions use different data sources. The widgets that implement search, browse, and other related functions, such as results pagination, use unique source IDs to reference these data sources. If you implement both search and browse functions on a single page, a source ID conflict may occur.

## Before You Start

Here are some things you should make sure you have done before you start implementing. You can use [Documentation for Oracle Service Cloud Products (Answer ID 5168)](https://cx.rightnow.com/app/answers/detail/a_id/5168) to locate the documentation for specific release, then use the Documentation Library or Online Help User Guide link to locate the relevant topics for more information on these steps.

* Enable the MOD\_CP\_DEVELOPMENT\_ENABLED configuration setting to make changes to your Customer Portal development site. You can find more information about enabling this setting in Service Cloud User Guide.
* Set up a WebDAV connection to manage your development site files. You can find more information about connecting to a Customer Portal site using WebDAV in Service Cloud User Guide.
* Create a backup copy of your files using WebDAV. Navigate to cp/customer/development/views and download all the folders. You can find more information about downloading files in Service Cloud User Guide.
* Update widgets to the latest version. See Update Widgets for more information.
* Make sure you created and published articles in at least one content type so that search results are available.

## Implement Knowledge Advanced in a New Customer Portal Instance

You can enable Knowledge Advanced in a new Customer Portal instance by replacing the standard Customer Portal files with the Knowledge Advanced reference implementation files.

You can enable Knowledge Advanced in a new instance by replacing the standard pages in your development folder with the Knowledge Advanced reference implementation pages. You can replace the standard pages by downloading the reference implementation, copying them into your site, configuring the pages, and promoting the new implementation to production.

### Download the Reference Implementation

Download the folder that contains the Knowledge Advanced reference implementation pages.

1. Connect your Customer Portal site through WebDav.
2. Download the okcs folder from here:

/cp/core/framework/views/pages/

Related information

[Video: Connecting WebDav to Configure Customer Portal for Knowledge Advanced](https://www.youtube.com/embed/t7bVDJTOlIU?rel=0&autoplay=1)

### Replace the Customer Portal Standard Pages

Replace the standard pages in your current instance with the reference implementation pages.

1. Copy the reference implementation pages from the okcs folder.
2. Place them here:

/cp/customer/development/views/pages/

### Add the Knowledge Advanced Look and Feel

Add the reference implementation templates to your current instance to update the look and feel for page style, navigation menu, and search results.

1. Copy the okcs\_standard.php and okcs\_mobile.php template files from here:

/cp/core/framework/views/templates/okcs

1. Place the template files here:

/cp/customer/development/views/templates/

### Promote Changes to Go Live

Stage your current instance pages to see how your changes will appear on the production site. Then, promote the changes to production so your users can begin using them.

1. Go to the Customer Portal administration page (https://<your\_site>/ci/admin/), click Development Mode from the dashboard. The Customer Portal site opens in the development mode.
2. Preview your development pages to verify your changes in the development mode.
3. Promote your changes to stage and production from the Customer Portal administration page.

You can find more information about staging and promoting changes into production in Service Cloud User Guide. See [Documentation for Oracle Service Cloud Products (Answer ID 5168)](https://cx.rightnow.com/app/answers/detail/a_id/5168) to locate the documentation for your specific release, then use the Documentation Library or the Online Help User Guide link to locate the relevant topics.

## Implement Knowledge Advanced in an Existing Customer Portal Instance

You can implement Knowledge Advanced in an existing instance by replacing or modifying the standard files in your development folder with the Knowledge Advanced reference implementation files.

You can preserve your customized pages by modifying only the specific lines of code required to implement Knowledge Advanced. You can use the reference implementation pages as examples on how to modify the pages instead of replacing them. You must promote your changes to production so that users can begin using them.

You can find more information about staging and promoting changes into production in Service Cloud User Guide. See [Documentation for Oracle Service Cloud Products (Answer ID 5168)](https://cx.rightnow.com/app/answers/detail/a_id/5168) to locate the documentation for your specific release, then use the Documentation Library or the Online Help User Guide link to locate the relevant topics.

### Add the Knowledge Advanced Look and Feel

You must replace or edit the standard template in your existing instance to add the Knowledge Advanced look and feel for page style, navigation menu, search results, and the password page. You can implement the look and feel using either of the following methods:

* Replace the standard template in your existing instance with the reference implementation standard template.
* Edit the standard template in your existing instance to include code from the reference implementation standard template.

#### Replace the Standard Template

Replace the standard template with the reference implementation standard template.

1. Copy the implementation standard template from here:

/cp/core/framework/views/templates/okcs/okcs\_standard.php

1. Place it here:

/cp/customer/development/views/templates/

#### Edit the Standard Template

Edit the standard template to include code from the reference implementation standard template and rename the modified template file.

1. Edit the file:

cp/customer/development/views/templates/standard.php

1. Add the style sheet by changing this line:

<rn:theme path="/euf/assets/themes/standard" css="site.css"/>

to:

<rn:theme path="/euf/assets/themes/standard" css="site.css, okcs.css"/>

1. Add search by changing this line:

<rn:widget path="search/SimpleSearch" report\_page\_url="/app/results"/>

to:

<rn:widget path="search/OkcsSimpleSearch" icon\_path="" report\_page\_url="/app/search"/>

1. Modify the navigation menu by changing this code:

<ul class="rn\_NavigationMenu">  
   <li>  
      <rn:widget path="navigation/NavigationTab" label\_tab="#rn:msg:SUPPORT\_HOME\_TAB\_HDG#" link="/app/#rn:config:CP\_HOME\_URL#" pages="home"/>  
   </li>  
   <li>  
      <rn:widget path="navigation/NavigationTab" label\_tab="#rn:msg:ASK\_QUESTION\_HDG#" link="/app/ask" pages="ask, ask\_confirm"/>  
   </li>  
   <li>  
      <rn:widget path="navigation/NavigationTab" label\_tab="#rn:msg:COMMUNITY\_HOME\_LBL#" link="/app/social/home" pages="social/home"/>  
   </li>  
</ul>

to:

<ul class="rn\_NavigationMenu">  
   <li>  
      <rn:widget path="navigation/NavigationTab" label\_tab="#rn:msg:SUPPORT\_HOME\_TAB\_HDG#" link="/app/#rn:config:CP\_HOME\_URL#" pages="home"/>  
   </li>  
   <rn:condition config\_check="OKCS\_ENABLED == true">  
      <li>  
         <rn:widget path="navigation/NavigationTab" label\_tab="#rn:msg:ANSWERS\_HDG#" link="/app/browse" pages="browse, answers/list, answers/detail, answers/intent, answers/answer\_view, search"/>  
      </li>  
   <rn:condition\_else>  
      <li>  
         <rn:widget path="navigation/NavigationTab" label\_tab="#rn:msg:ANSWERS\_HDG#" link="/app/search" pages="answers/list, answers/detail, answers/intent, answers/answer\_view, search"/>  
      </li>  
   </rn:condition>  
   <li>  
      <rn:widget path="navigation/NavigationTab" label\_tab="#rn:msg:ASK\_QUESTION\_HDG#" link="/app/ask" pages="ask, ask\_confirm"/>  
   </li>  
</ul>

1. Modify the password login by adding this code:

sub:login:create\_account\_fields="Contact.Emails.PRIMARY.Address;Contact.Login;Contact.NewPassword;Contact.FullName"

after this line:

sub:input\_Contact.NewPassword:label\_input="#rn:msg:PASSWORD\_LBL#"

1. Rename the template file to:

okcs\_template.php

### Add Knowledge Advanced Search to the Home Page

You can add the Knowledge Advanced search (OKCSSearch) functionality so that users can search for knowledge base articles. You can implement the Knowledge Advanced search functionality in the support home (home.php) page using either of the following methods:

* Replace the home page in your existing instance with the reference implementation home page.
* Edit it the home page in your existing instance to include code from the reference implementation home page.

#### Replace the Home Page

Replace the home page with the reference implementation home page.

1. Copy the reference implementation home page from here:

/cp/core/framework/views/pages/okcs/home.php

1. Place it here:

/cp/customer/development/views/pages/

#### Edit the Home Page

Edit the home page to include code from the reference implementation home page.

1. Edit the file:

cp/customer/development/views/pages/home.php

1. Replace Knowledge Foundation search with Knowledge Advanced search by changing this line:

<rn:container source\_id="KFSearch">

to:

<rn:container source\_id="OKCSSearch">

1. Add both the recent searches and the suggested searches widgets by replacing this code:

<rn:widget path="searchsource/SourceSearchButton" search\_results\_url="/app/results"/>

with:

<rn:widget path="okcs/RecentSearches"/>  
<rn:widget path="okcs/OkcsSuggestions"/>  
<rn:widget path="searchsource/SourceSearchButton" initial\_focus="true" search\_results\_url="/app/results"/>

1. Add the knowledge product and category selector by changing this line:

<rn:widget path="navigation/VisualProductCategorySelector" numbered\_pagination="true"/>

to:

<rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered\_pagination="true"/>

1. Add both the popular answers and the recent answers widgets by adding this code:

<div class="rn\_PopularKB">  
   <div class="rn\_Container">  
     <rn:widget path="okcs/AnswerList" type="popular" target="\_self" view\_type="list"/>  
   </div>  
</div>  
<div class="rn\_PopularKB rn\_RecentKB">  
   <div class="rn\_Container">  
     <rn:widget path="okcs/AnswerList" type="recent" target="\_self" view\_type="list"/>  
   </div>  
</div>

after this code:

<div class="rn\_PageContent rn\_Home">  
   <div class="rn\_Container">  
     <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered\_pagination="true"/>  
   </div>

Example

Here’s an example of the reference implementation home page:

<rn:meta title="#rn:msg:SHP\_TITLE\_HDG#" template="okcs\_standard.php" clickstream="home"/>  
<div class="rn\_Hero">  
    <div class="rn\_HeroInner">  
        <div class="rn\_HeroCopy">  
            <h1>#rn:msg:WERE\_HERE\_TO\_HELP\_LBL#</h1>  
        </div>  
        <div class="rn\_SearchControls">  
            <h1 class="rn\_ScreenReaderOnly">#rn:msg:SEARCH\_CMD#</h1>  
            <form method="get" action="/app/results">  
                <rn:container source\_id="OKCSSearch">  
                    <div class="rn\_SearchInput">  
                        <rn:widget path="searchsource/SourceSearchField" initial\_focus="true" label\_placeholder="#rn:msg:ASK\_A\_QUESTION\_ELLIPSIS\_MSG#"/>  
                    </div>  
                    <rn:widget path="okcs/RecentSearches"/>  
 <rn:widget path="okcs/OkcsSuggestions"/>  
                    <rn:widget path="searchsource/SourceSearchButton" initial\_focus="true" search\_results\_url="/app/results"/>  
                </rn:container>  
            </form>  
        </div>  
    </div>  
</div>  
<div class="rn\_PageContent rn\_Home">  
    <div class="rn\_Container">  
        <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered\_pagination="true"/>  
    </div>  
    <div class="rn\_PopularKB">  
        <div class="rn\_Container">  
            <rn:widget path="okcs/AnswerList" type="popular" target="\_self" view\_type="list"/>   
        </div>  
    </div>  
    <div class="rn\_PopularKB rn\_RecentKB">  
        <div class="rn\_Container">  
            <rn:widget path="okcs/AnswerList" type="recent" target="\_self" view\_type="list"/>   
        </div>  
    </div>  
</div>

### Replace or Edit the Account Overview Page

You can replace or edit the account overview (overview.php) page so that users can view their support questions and content recommendations. Your users can also update their account settings, change their password, manage answer notifications, and manage content type notifications using the account overview page.

You can implement Knowledge Advanced notifications on the account overview page using either of the following methods:

* Replace the overview page in your current instance with the reference implementation overview page.
* Edit the overview page in your current instance to include code from the reference implementation overview page.

#### Replace the Account Overview Page

Replace the account overview page with the reference implementation overview page.

1. Copy the reference implementation overview page from here:

/cp/core/framework/views/pages/okcs/account/overview.php

1. Place it here:

/cp/customer/development/views/pages/account/

#### Edit the Account Overview Page

Edit the account overview page to include code from the reference implementation overview page.

1. Edit the file:

/cp/customer/development/views/pages/account/overview.php

1. Add manage content recommendations by adding this code:

<div id="rn\_LoadingIndicator" class="rn\_Browse">  
   <rn:widget path="okcs/LoadingIndicator"/>  
</div>  
   <rn:container source\_id="OKCSBrowse">  
      <div class="rn\_HeaderContainer">  
         <h2>a class="rn\_Profile" href="/app/account/recommendations/list#rn:session#">#rn:msg:MY\_RECOMMENDATIONS\_LBL#</a></h2>  
      </div>  
      <div id="rn\_OkcsManageRecommendations">  
         <rn:widget path="okcs/OkcsManageRecommendations"/>  
         <a href="/app/account/recommendations/list#rn:session#">#rn:msg:SEE\_ALL\_RECOMMENDATIONS\_LBL#</a>  
      </div>  
   </rn:container>

after this code:

<div class=”rn\_ContentDetail”>

1. Remove social discussion notifications by removing this code:

<div class="rn\_Discussions">  
   <rn:container report\_id="15156" per\_page="4">  
      <div class="rn\_HeaderContainer">  
         <h2><a class="rn\_Discussions" href="/app/social/questions/list/author/#rn:profile:socialUserID#/kw/\*#rn:session#">#rn:msg:MY\_DISCUSSION\_QUESTIONS\_LBL#</a></h2>  
      </div>  
      <rn:widget path="reports/Grid" static\_filter="created\_by=#rn:profile:socialUserID#" label\_caption="<span class='rn\_ScreenReaderOnly'>#rn:msg:YOUR\_RECENTLY\_SUBMITTED\_DISCUSSIONS\_LBL#</span>"/>  
      <a href="/app/social/questions/list/author/#rn:profile:socialUserID#/kw/\*#rn:session#">#rn:msg:SEE\_ALL\_MY\_DISCUSSION\_QUESTIONS\_LBL#</a>  
   </rn:container>  
</div>

1. Add the content type notifications link in the side rail (<div class=”rn\_SideRail”>) section by adding this line:

<li>a href="/app/account/notif/content\_type\_list#rn:session#">#rn:msg:MANAGE\_YOUR\_CONTENT\_TYPE\_NOTIFICATIONS\_LBL#</a></li>

Here’s an example of the reference implementation account overview page:

<rn:meta title="#rn:msg:ACCOUNT\_OVERVIEW\_LBL#" template="okcs\_standard.php" login\_required="true" force\_https="true"/>  
<div class="rn\_Hero">  
    <div class="rn\_Container">  
        <h1>#rn:msg:ACCOUNT\_OVERVIEW\_LBL#</h1>  
    </div>  
</div>  
<div class="rn\_PageContent rn\_AccountOverview rn\_Container">  
    <div class="rn\_ContentDetail">  
        <div class="rn\_Questions">  
            <rn:container report\_id="196" per\_page="4">  
                <div class="rn\_HeaderContainer">  
                    <h2><a class="rn\_Questions" href="/app/account/questions/list#rn:session#">#rn:msg:MY\_SUPPORT\_QUESTIONS\_LBL#</a></h2>  
                </div>  
                <rn:widget path="reports/Grid" label\_caption="<span class='rn\_ScreenReaderOnly'>#rn:msg:YOUR\_RECENTLY\_SUBMITTED\_QUESTIONS\_LBL#</span>"/>  
                <a href="/app/account/questions/list#rn:session#">#rn:msg:SEE\_ALL\_MY\_SUPPORT\_QUESTIONS\_LBL#</a>  
            </rn:container>  
        </div>   
    <div id="rn\_LoadingIndicator" class="rn\_Browse">  
        <rn:widget path="okcs/LoadingIndicator"/>  
    </div>  
    <rn:container source\_id="OKCSBrowse">  
        <div class="rn\_HeaderContainer">  
            <h2><a class="rn\_Profile" href="/app/account/recommendations/list#rn:session#">#rn:msg:MY\_RECOMMENDATIONS\_LBL#</a></h2>    
        </div>  
        <div id="rn\_OkcsManageRecommendations">  
            <rn:widget path="okcs/OkcsManageRecommendations"/>  
            <a href="/app/account/recommendations/list#rn:session#">#rn:msg:SEE\_ALL\_RECOMMENDATIONS\_LBL#</a>  
        </div>  
    </rn:container>  
</div>  
<div class="rn\_SideRail">  
    <div class="rn\_Well">  
        <h3>#rn:msg:LINKS\_LBL#</h3>  
        <ul>  
            <li>a href="/app/account/profile#rn:session#">#rn:msg:UPDATE\_YOUR\_ACCOUNT\_SETTINGS\_CMD#/a>/li>  
            <rn:condition external\_login\_used="false">  
                <rn:condition config\_check="EU\_CUST\_PASSWD\_ENABLED == true">  
                    <li><a href="/app/account/change\_password#rn:session#">#rn:msg:CHANGE\_YOUR\_PASSWORD\_CMD#</a>/li>  
                </rn:condition>  
            </rn:condition>  
            <li><a href="/app/account/notif/list#rn:session#">#rn:msg:MANAGE\_YOUR\_NOTIFICATIONS\_LBL#</a></li>  
            <li><a href="/app/account/notif/content\_type\_list#rn:session#">#rn:msg:MANAGE\_YOUR\_CONTENT\_TYPE\_NOTIFICATIONS\_LBL#</a></li>  
            <rn:condition is\_active\_social\_user="true">  
                    <li><a href="/app/public\_profile/user/#rn:profile:socialUserID#">#rn:msg:VIEW\_YOUR\_PUBLIC\_PROFILE\_LBL#</a></li>  
            </rn:condition>  
        </ul>  
    </div>  
</div>

### Add Knowledge Advanced Browse

You must implement the Knowledge Advanced browse page so that users can browse articles and filter the articles by content type, product, and category. You implement the browse page by adding the browse page from the reference implementation to your existing instance.

1. Copy the browse page from here:

/cp/core/framework/views/pages/okcs/browse.php

1. Place it here:

/cp/customer/development/views/pages/

Here’s an example of the reference implementation browse page:

<rn:meta title="#rn:msg:FIND\_ANS\_HDG#" template="okcs\_standard.php" clickstream="answer\_list"/>  
<rn:condition config\_check="OKCS\_ENABLED == true">  
    <div class="rn\_Container">  
        <div id="rn\_PageTitle" class="rn\_ScreenReaderOnly">  
            <h1>#rn:msg:BROWSE1\_AB\_HDG#</h1>  
        </div>  
        <rn:container source\_id="OKCSBrowse">  
            <div id="rn\_PageContentChannel" class="rn\_Browse rn\_PageContentChannel">  
                <div class="rn\_PageContentChannelInner">  
                    <rn:widget path="okcs/ContentType"/>  
                </div>  
            </div>  
  
            <div id="rn\_LoadingIndicator" class="rn\_Browse">  
               <rn:widget path="okcs/LoadingIndicator"/>  
           < /div>  
  
            <div id="rn\_PageContentArticles" class="rn\_AnswerList">  
                <rn:widget path="okcs/OkcsRecommendContent"/>  
                <div class="rn\_ResultPadding">  
                    <div>  
                        <div id="rn\_OkcsLeftContainer" class="rn\_OkcsLeftContainer">  
                            <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter\_type="products" view\_type="explorer"/>  
                            <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter\_type="categories" view\_type="explorer"/>  
                        </div>  
                        <div id="rn\_OkcsRightContainer" class="rn\_OkcsRightContainer">  
                            <div id="rn\_Browse\_Loading">/div>  
                            <div id="rn\_OkcsAnswerList">  
                                <rn:widget path="okcs/AnswerList" target="\_self"/>  
                                <div class="rn\_FloatRight">  
                                    <rn:widget path="okcs/OkcsPagination"/>  
                                </div>  
                            </div>  
                        </div>  
                    </div>  
                </div>  
            </div>  
        </rn:container>  
    </div>  
</rn:condition>

### Add the Knowledge Advanced Answer Detail Page

You must implement the answer detail (answer\_view.php) page to open articles with the article details page. Users can see the article content and its details, including document ID, Version, Status, and Published Date using the article details page.

You can implement the answer detail page using either of the following methods:

* Add the reference implementation answer detail page to your current instance.
* Edit the answer detail page in your current instance to include code from the reference implementation answer detail page.

#### Add the Answer Detail Page

Add the answer details page from reference implementation to your current instance.

1. Copy the reference implementation answer detail from here:

/cp/core/framework/views/pages/okcs/answers/answer\_view.php

1. Place it here:

/cp/customer/development/views/pages/answers/

#### Edit the Answer Detail Page

Edit the answer detail page in your current instance to include code from the reference implementation answer detail (answer\_view.php) page and rename the modified file.

Ensure that the answer detail page does not contain the answer details (answer\_details) attribute in the meta code <rn:meta...>. The presence of this attribute will cause the articles to display incorrectly.

1. Edit the file:

/cp/customer/development/views/pages/answers/detail.php

1. Replace the entire code with this code:

<div class="rn\_Container">  
    <rn:condition config\_check="OKCS\_ENABLED == true">  
        <div class="rn\_ContentDetail">  
            <rn:meta title="#rn:php:\RightNow\Libraries\SEO::getDynamicTitle('answer', \RightNow\Utils\Url::getParameter('a\_id'))#" template="okcs\_standard.php"  clickstream="answer\_view"/>  
            <div class="rn\_PageTitle rn\_RecordDetail">  
               <rn:widget path="okcs/OkcsProductCategoryBreadcrumb" display\_first\_item="true"/>  
                <div class="rn\_OkcsAnswerAction">  
                   <rn:widget path="okcs/SubscriptionButton"/>  
                   <rn:widget path="okcs/OkcsRecommendContent"/>  
                /div>  
                <rn:widget path="okcs/AnswerTitle">  
            </div>  
            <div class="rn\_AnswerView">  
                <rn:widget path="okcs/AnswerStatus">  
                <div class="rn\_SectionTitle"></div>  
                <rn:widget path="okcs/AnswerContent">  
            </div>  
            <div class="rn\_DetailTools rn\_HideInPrint">  
                <div class="rn\_Links">  
                    <rn:widget path="okcs/OkcsEmailAnswerLink" />  
                </div>  
            </div>  
            <rn:widget path="okcs/DocumentRating"/>  
            <rn:widget path="okcs/OkcsRelatedAnswers"/>  
        </div>  
        <aside class="rn\_SideRail" role="complementary">  
            <rn:widget path="okcs/OkcsRecentlyViewedContent"/>  
        </aside>  
    </rn:condition>  
</div>

1. Rename the file:

answer\_view.php

### Add the Knowledge Advanced Results Page

You must implement the results page so that users can see Knowledge Advanced search results. You implement the search functionality by replacing the results page in your current instance with the reference implementation results page. Users can see a list of articles in search results that match the entered search text.

Users can also enable the interactive spell checker on the results page to either correct misspelled words automatically, or to suggest spelling corrections to the user-entered text.

1. Copy the reference implementation results page from here:

/cp/core/framework/views/pages/okcs/results.php

1. Place it here:

/cp/customer/development/views/pages/

Here’s an example of the reference implementation results page:

<rn:meta title="#rn:msg:FIND\_ANS\_HDG#" template="okcs\_standard.php" clickstream="answer\_list"/>  
<rn:container source\_id="OKCSSearch" document\_id\_reg\_ex="^[\p{L}\p{Nd}\_.]\*\d{1,10}$" doc\_id\_navigation="true">  
    <div id="rn\_LoadingIndicator" class="rn\_Browse rn\_Container">  
       <rn:widget path="okcs/LoadingIndicator"/>  
    </div>  
    <div id="rn\_PageTitle" class="rn\_Search">  
        <div class="rn\_Hero">  
            <div class="rn\_HeroInner">  
                <div class="rn\_SearchControls">  
                    <h1 class="rn\_ScreenReaderOnly">#rn:msg:SEARCH\_CMD#</h1>  
                    <form>  
                        <div class="rn\_SearchInput">  
                            <rn:widget path="searchsource/SourceSearchField" initial\_focus="true" label\_placeholder="#rn:msg:ASK\_A\_QUESTION\_ELLIPSIS\_MSG#" filter\_label="Keyword" filter\_type="query"/>  
                        </div>  
                        <rn:widget path="okcs/RecentSearches"/>  
                        <rn:widget path="okcs/OkcsSuggestions"/>  
                        <rn:widget path="searchsource/SourceSearchButton" initial\_focus="true" history\_source\_id="OKCSSearch"/>  
                        <rn:widget path="okcs/OkcsInteractiveSpellChecker"/>  
                        <div class="rn\_StartOver">  
                            <a href="/app/home" >#rn:msg:START\_OVER\_LBL#/a>  
                        </div>  
                    </form>  
                </div>    
            </div>  
        </div>  
    </div>  
    <div id="rn\_PageContent" class="rn\_Container rn\_OkcsResultsContainer">  
        <div class="rn\_ResultList rn\_OkcsTable">  
            <div class="rn\_OkcsTableRow">  
                <div id="rn\_OkcsLeftContainer" class="rn\_OkcsLeftContainer rn\_OkcsTableCell">  
                    <rn:widget path="okcs/Facet"/>  
                </div>  
                <div id="rn\_OkcsRightContainer" class="rn\_OkcsRightContainer rn\_OkcsTableCell">  
                    <rn:widget path="okcs/SearchResult" hide\_when\_no\_results="false"/>  
                    <div class="rn\_FloatRight">  
                        <rn:widget path="okcs/OkcsPagination"/>  
                    </div>  
                </div>  
            </div>  
            <div class="rn\_OkcsTableRow">  
                <div class="rn\_OkcsLeftContainer rn\_OkcsTableCell">/div>  
                <div class="rn\_OkcsRightContainer rn\_OkcsTableCell">  
                    <rn:widget path="okcs/OkcsRecentlyViewedContent"/>  
                </div>  
            </div>  
        </div>  
        <aside class="rn\_SideRail" role="complementary">  
            <rn:widget path="utils/ContactUs" channels="question,chat,feedback"/>  
        </aside>  
    </div>  
</rn:container>

### Enable Users to Manage Answer Notifications

You must edit the notifications (list.php) page so that users can manage their notifications. Users can see notifications they have subscribed to and delete a subscription from the notifications page.

You can edit the notifications page in your current instance to include code from the reference implementation notifications page.

1. Edit the file:

/cp/customer/development/views/pages/account/notif/list.php

1. Add the loading indicator by adding this code:

<div id="rn\_LoadingIndicator" class="rn\_Browse">  
   <rn:widget path="okcs/LoadingIndicator"/>  
</div>

before this code:

<div class=”rn\_PageContent”>

1. Add the answer notification manager widget by adding this code:

<rn:widget path="okcs/OkcsAnswerNotificationManager" view\_type="list"/>

within this section:

<div class=”rn\_PageContent”>

Here’s an example of the reference implementation notifications (list.php) page:

<rn:meta title="#rn:msg:NOTIFICATIONS\_HDG#" template="okcs\_standard.php" login\_required="true" force\_https="true"/>  
<div class="rn\_Hero">  
    <div class="rn\_Container">  
        <h1>#rn:msg:NOTIFICATIONS\_HDG#</h1>  
    </div>  
</div>  
<div id="rn\_LoadingIndicator" class="rn\_Browse">  
    <rn:widget path="okcs/LoadingIndicator"/>  
</div>  
<div class="rn\_PageContent rn\_AccountNotifications rn\_Container">  
    <h2>#rn:msg:ANS\_NOTIFICATIONS\_LBL#</h2>  
    <rn:widget path="okcs/OkcsAnswerNotificationManager" view\_type="list"/>   
</div>

### Add Knowledge Advanced SmartAssistant

You must edit the ask a question (ask.php) page to add the Knowledge Advanced version of SmartAssistant. Users can submit a questions to a support agent using the ask a question page. Users can also see knowledge articles as well as community discussions that match the user-entered text before submitting the question.

You can edit the ask a question page in your existing instance by copying code from the reference implementation ask a question page.

1. Edit the file:

cp/customer/development/views/pages/ask.php

1. Replace standard SmartAssistant with Knowledge Advanced SmartAssistant by changing this code:

<rn:condition content\_viewed="2" searches\_done="1">  
   <rn:condition\_else/>  
   <rn:widget path="input/SmartAssistantDialog" label\_prompt="#rn:msg:OFFICIAL\_SSS\_MIGHT\_L\_IMMEDIATELY\_MSG#"/>  
</rn:condition>

to:

<rn:widget path="okcs/OkcsSmartAssistant" view\_type="explorer"/>

1. Remove the following code:

<div class="translucent">  
   <strong>#rn:msg:TIPS\_LBL#:</strong>  
      <ul>  
         <li> #rn:msg:INCLUDE\_AS\_MANY\_DETAILS\_AS\_POSSIBLE\_LBL#</li>  
      </ul>  
</div>  
<br>  
<p>#rn:msg:NEED\_A\_QUICKER\_RESPONSE\_LBL# <a href="/app/social/ask#rn:session#">#rn:msg:ASK\_OUR\_COMMUNITY\_LBL#</a></p>

Here’s an example of the reference implementation ask a question (ask.php) page:

<rn:meta title="#rn:msg:ASK\_QUESTION\_HDG#" template="okcs\_standard.php" clickstream="incident\_create"/>  
<div class="rn\_Container">  
    <div id="rn\_PageTitle" class="rn\_AskQuestion">  
        <h1>#rn:msg:SUBMIT\_QUESTION\_OUR\_SUPPORT\_TEAM\_CMD#</h1>  
    </div>  
</div>  
<div id="rn\_PageContent" class="rn\_AskQuestion rn\_Container">  
    <div class="rn\_Padding">  
        <form id="rn\_QuestionSubmit" method="post" action="/ci/ajaxRequest/sendForm">  
            <div id="rn\_ErrorLocation">/div>  
            <rn:condition logged\_in="false">  
                <rn:widget path="input/FormInput" name="Contact.Emails.PRIMARY.Address" required="true" initial\_focus="true" label\_input="#rn:msg:EMAIL\_ADDR\_LBL#"/>  
                <rn:widget path="input/FormInput" name="Incident.Subject" required="true" label\_input="#rn:msg:SUBJECT\_LBL#"/>  
            </rn:condition>  
            <rn:condition logged\_in="true">  
                <rn:widget path="input/FormInput" name="Incident.Subject" required="true" initial\_focus="true" label\_input="#rn:msg:SUBJECT\_LBL#"/>  
            </rn:condition>  
                <rn:widget path="input/FormInput" name="Incident.Threads" required="true" label\_input="#rn:msg:QUESTION\_LBL#"/>  
                <rn:widget path="input/FileAttachmentUpload"/>  
                <rn:widget path="input/ProductCategoryInput" name="Incident.Product"/>  
                <rn:widget path="input/ProductCategoryInput" name="Incident.Category" data\_type="Category"/>  
                <rn:widget path="input/FormSubmit" label\_button="#rn:msg:CONTINUE\_ELLIPSIS\_CMD#" on\_success\_url="/app/ask\_confirm" error\_location="rn\_ErrorLocation"/>  
                <rn:widget path="okcs/OkcsSmartAssistant" view\_type="explorer"/>  
        </form>  
    </div>  
</div>

### Enable Knowledge Advanced on the Chat Page

You must replace or edit the chat landing (chat\_landing.php) page to add the Knowledge Advanced chat enhancements to both the chat report page and the chat popup window. Users can see their position in the support queue, estimated and average wait times, and a search field on the chat landing page. The chat window opens when an agent is available.

You must implement the Knowledge Advanced chat enhancements using either of the following methods:

* Replace the chat landing page in your current instance with the reference implementation chat landing page.
* Edit the chat landing page in your current instance to include code from the reference implementation chat landing page.

#### Replace the Chat Landing Page

Replace the chat landing page with the reference implementation chat landing page.

1. Copy the reference implementation chat landing page from here:

/cp/core/framework/views/pages/okcs/chat/chat\_landing.php

1. Place it here:

/cp/customer/development/views/pages/chat/

#### Edit the Chat Landing Page

Edit the chat landing page by copying code from the reference implementation chat landing page.

1. Edit the file:

/cp/customer/development/views/pages/chat/chat\_landing.php

1. Add the chat queue by replacing this line:

<rn:widget path="chat/ChatQueueSearch" popup\_window="true"/>

with:

<rn:condition config\_check="OKCS\_ENABLED == true">  
<rn:widget path="chat/ChatQueueSearch" report\_page\_url="/app/search" popup\_window="true" />  
<rn:condition\_else>  
<rn:widget path="chat/ChatQueueSearch" popup\_window="true" />  
</rn:condition>

## Implement the Knowledge Advanced Widgets

Your instance of Service Cloud includes Knowledge widgets. They enable users to access knowledge features, including content type subscription, suggested searches, filter search by product and category, and search result facets.

You can implement widgets and configure them to tailor knowledge for your organization and users. You configure widgets by adding or editing the widget attributes on the pages in your existing instance. You can see the list of knowledge widgets by reading the documentation contained in the Customer Portal administration page: https://<your\_site>/ci/admin/docs/widgets/standard/okcs

### Update Widgets to the Latest Version

Before you implement widgets, update them to the latest version.

1. Go to the Customer Portal administration page:

https://<your\_site>/ci/admin

1. Click on Widgets, then Widget Versions. The widgets page opens.
2. Click the Update All option. The widgets will be updated to the latest version.

### Set the Article Display on the Home Page

Users see most popular answers and most recent answers in a table view by default. You change the home page so that users can see articles in a list view. You implement list view using the view type parameter.

1. Edit the file:

/cp/customer/development/views/pages/home.php

1. Edit the most popular answers code by changing this line:

<rn:widget path="okcs/AnswerList" type="popular" target="\_self"/>

to:

<rn:widget path="okcs/AnswerList" type="popular" target="\_self" view\_type="list"/>

1. Edit the most recent answers code by changing this line:

<rn:widget path="okcs/AnswerList" type="recent" target="\_self"/>

to:

<rn:widget path="okcs/AnswerList" type="recent" target="\_self" view\_type="list"/>

### Configure Searches to Filter by Product or Category

You can enable users to filters search results using product and category on the home page. Users see only the articles associated with the selected product, category, or both product and category. You implement these filters by including the product category search filter widget. The filters appear only in a tree view.

1. Edit the file:

cp/customer/development/views/pages/home.php

1. Add the product filter by adding this line:

<rn:widget path= "OkcsProductCategorySearchFilter" filter\_type="products"/>

1. Add the category filter by adding this line:

<rn:widget path= "OkcsProductCategorySearchFilter" filter\_type="categories"/>

Here’s an example of how to include the product and category filter in the home page:

<rn:meta title="#rn:msg:SHP\_TITLE\_HDG#" template="okcs\_standard.php" clickstream="home"/><div class="rn\_SearchControls">  
<div class="rn\_Hero">  
    <div class="rn\_HeroInner">  
        <div class="rn\_HeroCopy">  
            <h1>#rn:msg:WERE\_HERE\_TO\_HELP\_LBL#</h1>  
        </div>  
        <div class="rn\_SearchControls">  
            <h1 class="rn\_ScreenReaderOnly">#rn:msg:SEARCH\_CMD#</h1>  
            <form method="get" action="/app/results">  
                <rn:container source\_id="OKCSSearch">  
                    <div class="rn\_SearchInput">  
                        <rn:widget path="searchsource/SourceSearchField" initial\_focus="true" label\_placeholder="#rn:msg:ASK\_A\_QUESTION\_ELLIPSIS\_MSG#"/>  
                    </div>  
                    <rn:widget path="okcs/RecentSearches"/>  
                    <rn:widget path="okcs/OkcsSuggestions"/>  
                    <rn:widget path="searchsource/SourceSearchButton" initial\_focus="true" search\_results\_url="/app/results"/>  
 <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter\_type="products" view\_type="explorer"/>  
 <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter\_type="categories" view\_type="explorer"/>  
                </rn:container>  
            </form>  
        </div>  
    </div>  
</div>  
<div class="rn\_PageContent rn\_Home">  
    <div class="rn\_Container">  
        <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered\_pagination="true"/>  
    </div>  
    <div class="rn\_PopularKB">  
        <div class="rn\_Container">  
            <rn:widget path="okcs/AnswerList" type="popular" target="\_self" view\_type="list"/>   
        </div>  
    </div>  
    <div class="rn\_PopularKB rn\_RecentKB">  
        <div class="rn\_Container">  
            <rn:widget path="okcs/AnswerList" type="recent" target="\_self" view\_type="list"/>   
        </div>  
    </div>  
</div>

### Implement Suggested Searches

You can edit the home page so that users can use suggested searches. Users can see a list of articles that match the search term or phrase as they type in the search field. Suggestions with exact keywords match in article titles appear at the top of the results, then suggestions that have partial match. You implement suggested searches to provide a faster way for your users to locate and select relevant articles.

The current implementation contains suggested searches by default on both the home page and the results page. If you are using a previous implementation, you can implement suggested searches using the suggestions widget.

Before you implement suggested searches, you must enable attribute level searching for the master identifier (title field) of each content type in the knowledge base. You can find more information about enabling attribute level searching in the About Schema Attribute Options section in the [Knowledge Advanced Administration Guide](http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=FAAKA) .

1. Edit a Customer Portal page, for example:

/cp/customer/development/views/pages/home.php

1. Add the suggested search widget code as follows:

<rn:widget path="okcs/OkcsSuggestions"/>

1. Add the suggestion count attribute to specify the number of suggested searches you want to display. The default is 7. The maximum value is 100.

<rn:widget path="okcs/OkcsSuggestions" suggestion\_count="15"/>

Here’s an example of how to include suggested search widget in the home page:

<rn:meta title="#rn:msg:SHP\_TITLE\_H<>DG#" template="okcs\_standard.php" clickstream="home"/>  
<div class="rn\_Hero">  
 <div class="rn\_HeroInner">  
 <div class="rn\_HeroCopy">  
 <h1>#rn:msg:WERE\_HERE\_TO\_HELP\_LBL#</h1>  
 </div>  
 <div class="rn\_SearchControls">  
 <h1 class="rn\_ScreenReaderOnly">#rn:msg:SEARCH\_CMD#</h1>  
 <form method="get" action="/app/results">  
 <rn:container source\_id="OKCSSearch">  
 <div class="rn\_SearchInput">  
 <rn:widget path="searchsource/SourceSearchField" initial\_focus="true" label\_placeholder="#rn:msg:ASK\_A\_QUESTION\_ELLIPSIS\_MSG#"/>  
 </div>  
 <rn:widget path="okcs/RecentSearches">  
 <rn:widget path="okcs/OkcsSuggestions">  
 <rn:widget path="searchsource/SourceSearchButton" initial\_focus="true" search\_results\_url="/app/results"/>  
 </form>  
 </div>  
 </div>  
</div>  
<div class="rn\_PageContent rn\_Home">  
 <div class="rn\_Container">  
 <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered\_pagination="true"/>  
 </div>  
 <div class="rn\_PopularKB">  
 <div class="rn\_Container">  
 <rn:widget path="okcs/AnswerList" type="popular" target="\_self" view\_type="list"/>  
 </div>  
 </div>  
 <div class="rn\_PopularKB rn\_RecentKB">  
 <div class="rn\_Container">  
 <rn:widget path="okcs/AnswerList" type="recent" target="\_self" view\_type="list"/>  
 </div>  
 </div>  
</div>

For more information on the OkcsSuggestions widget, see the Standard Widgets section in Oracle Service Cloud Online Help User Guide. See [Documentation for Oracle Service Cloud Products (Answer ID 5168)](https://cx.rightnow.com/app/answers/detail/a_id/5168) to locate documentation for your specific release, then use the Documentation Library or the Online Help User Guide link to locate the relevant topics.

### Configure Search Results Facets

You can configure the display of facets on the search results page. You can use the top facet list attribute in the facet widget to:

* reorder facets
* hide facets
* rename facets

The following list shows the default order in which the facets display on the results page.

* Document Types
* Collections
* Categories
* Products

The following table describes four values with an example each on how to reorder, hide, or rename the facets using the top facet list attribute. You must separate the facets by the pipe (|) character and use the comma (,) character for the facet label description.

Table : Attribute Values for Configuring Facets

| Configuration | To display on the results page | Attribute Value |
| --- | --- | --- |
| Default | Document Types, Collections, Categories, Products | <rn:widget path=”okcs/Facet” top\_facet\_list=”DOC\_TYPES, Document Types|COLLECTIONS, Collections|CMS-CATEGORY-REF, Categories/CMS-PRODUCT, Products”/> |
| Reorder | Categories, Products, Collections, Document Types | <rn:widget path=”okcs/Facet” top\_facet\_list=”CMS-CATEGORY-REF|CMS-PRODUCT|COLLECTIONS|DOC\_TYPES”/> |
| Hide | Document Types, Collections, Categories | <rn:widget path=”okcs/Facet” top\_facet\_list=”DOC\_TYPES|COLLECTIONS|CMS-CATEGORY\_REF”/> |
| Rename | MyCategory, MyDocuments, MyProduct, MyCollections | <rn:widget path=”okcs/Facet” top\_facet\_list=”CMS-CATEGORY-REF, MyCategory|DOC\_TYPES, MyDocuments|CMS-PRODUCT, MyProducts|COLLECTIONS, MyCollections”/> |

### Enable Users to Set the Number of Results Per Page

You can configure the results per page drop-down so that users can change the number of display results that appear on each page by adding the results per page attribute to the results page. Specify the options for the number of results as a comma-separated list. The default options are 10, 20, 30, and 50. The results page uses the first number in the list as the default. You can remove the options from the results page by leaving the results per page attribute empty. Search will show ten results on each page by default.

You can follow these steps to configure the results per page drop-down on the results page.

1. Edit the file:

cp/customer/development/views/pages/results.php

1. Add the results\_per\_page attribute code as follows:

<rn:container source\_id="OKCSSearch" document\_id\_reg\_ex="^[\p{L}\p{Nd}\_.]\*\d{1,10}$" doc\_id\_navigation="true" results\_per\_page="2,6,12,18,20,25,30">

### Disable Search by Document ID

You can enable users to bypass the search results list and open an article directly in the article detail page. Users can open an article by entering its document ID in the search field from the results page.

The search by document ID feature is enabled by default. Modify the results page in your existing instance to disable search by document ID.

1. Edit the file:

cp/customer/development/views/pages/results.php

1. Remove the document\_id\_reg\_ex parameter by changing this line:

<rn:container source\_id="OKCSSearch" document\_id\_reg\_ex="^\w{1,10}\d{1,10}$"/>

to:

<rn:container source\_id="OKCSSearch"/>

### Change the Style of Intent Answers on the Results Page

You can apply a visual style to answers that are associated to intents. Using a distinct visual style for intent answers helps users identify them as featured answers. Intents are dictionary objects that represent a class of similar questions and answers. Intents provide a way for knowledge managers to feature a single best answer to a set of questions that may vary in wording or detail.

The current implementation contains a default stylesheet to distinguish intent answers from other search results. If you upgrade from a previous implementation to the current implementation, the upgrade process automatically installs the intent style sheet in the default location, /cp/customer/assets/themes/standard/intent.css. To use the default intent style, you must update the search result widget to the latest version, which includes the intent style, as shown in this example:

<rn:widget path="okcs/SearchResult" apply\_style\_on\_intents="true"/>

You can change the default style by editing the attributes in the intent style sheet.

1. Edit the file:

/cp/customer/assets/themes/standard/intent.css

1. Edit the default style attributes to match your desired style.

Example

Here’s an example of an intent style sheet that uses the default style:

.rn\_IntentResult {  
    padding: 0.75em;  
    color: #000;  
    border-radius: 0.3em;  
    border: 0.5em solid #40526b;  
}  
.rn\_IntentResult .rn\_ResultIcon:before {  
    display: none;  
}  
.rn\_SearchResult .rn\_SearchResultContent .rn\_IntentResult a {  
    font-weight: bold;  
}  
.rn\_IntentResult.rn\_ResultElement .rn\_Element1 {  
    padding-left: unset;  
}

### Configure a Product or Category Landing Page

You can configure the support home page as a product landing page or as a category landing page. Product and category landing pages restrict search and search results to a selected product or category, enabling users to easily find specific information. All searches from the page will automatically use the selected product or category, and answers displayed will be specific to the selected product or category. Users can also navigate to product and categories of interest using breadcrumbs and images.

You configure the home page using the knowledge visual product category selector widget to display either a product landing page or category landing page. The home page is configured as a product landing page by default.

You can add images to display for each product and category. To add product and category images, you must upload an image for each product or category to the following directory:

cp/customer/assets/images/prodcat-images

Images must have the same name as their corresponding product or category. If an image corresponding to a product or category is not found, default image (default.png) from the images directory is displayed on the home page.

Modify the home page in your existing instance to configure the product or category landing page.

1. Edit the file:

cp/customer/development/views/pages/home.php

1. Locate the OkcsVisualProductCategorySelector widget code, for example:

<rn:widget path="okcs/OkcsVisualProductCategorySelector" type="product"/>

1. Edit the type attribute to the desired configuration, either product or categories, for example:

<rn:widget path="okcs/OkcsVisualProductCategorySelector" type="categories"/>

1. Add the maximum\_items attribute to specify the number of products or categories you want to display. The default value is 8. The maximum value is 40.

<rn:widget path="okcs/OkcsVisualProductCategorySelector" maximum\_items="12"/>

### Configure the Article Attributes

You can configure the article attributes so that users can see the article details along with its content. You configure the article attributes that you want to display including the custom metadata parameter in the answer view page and the ask a question page. By default, your users can see article details, including Document ID, Version, Status, and Display Date.

Table : List of Article Attributes

| Attribute | Description |
| --- | --- |
| document\_id | The article’s document ID. |
| version | The article’s version. |
| status | The article’s status. Displays whether the article is published or in draft state. |
| display\_date | The publish date of a published article or the last modified date of a draft article. |
| creator | The user who created the article. |
| aggregate\_rating | The aggregate rating of article. |
| owner | The article’s owner. |
| answer\_id | The article’s answer ID. |
| last\_modified | The date that the article was last modified. |
| last\_modifier | The user who last updated the article. |

#### Configure Article Attributes on the Answer Page

You can configure article attributes on the answer view page using the custom metadata attribute.

1. Edit the file:

cp/customer/development/views/pages/answer/answer\_view.php

1. Add the custom\_metadata parameter code as follows:

<rn:widget path="okcs/AnswerStatus" custom\_metadata="attribute | attribute | attribute"/>

1. Specify the attributes that you want to display as a list of items separated by the pipe character (|), for example:

<rn:widget path="okcs/AnswerStatus" custom\_metadata="document\_id | version | status | display\_date"/>

#### Configure Article Attributes on the Ask a Question Page

You can configure article attributes on the Ask a Question page using the custom metadata attribute.

1. Edit the file:

cp/customer/development/views/pages/ask.php

1. Add the custom\_metadata parameter code as follows:

<rn:widget path="okcs/OkcsSmartAssistant" custom\_metadata="attribute | attribute | attribute"/>

1. Specify the attributes that you want to display as a list of items separated by the pipe character (|), for example:

<rn:widget path="Okcs/SmartAssistant" custom\_metadata="document\_id | version | status | display\_date"/>

### See a Translated Version of an Article

You can implement the language drop-down so that users can see translated versions of an article from the article detail page. You implement the language drop-down by adding the translated answer selector widget to the article detail page.

Users select a locale from the language drop-down to see the article and its content in the selected locale. The language drop-down displays locales for only the published articles. In addition, a user can see translated versions of an article based on the usergroup access at the article level.

1. Edit the file:

cp/customer/development/views/pages/answer/answer\_view.php

1. Add the language drop-down by adding this line:

<rn:widget path="okcs/OkcsTranslatedAnswerSelector"/>

within this section:

<div class="rn\_OkcsAnswerAction">  
 <rn:widget path="okcs/SubscriptionButton"/>  
 <rn:widget path="okcs/OkcsRecommendContent"/>  
</div>

1. Specify the label attribute to change label for the drop-down as follows:

<rn:widget path="okcs/OkcsTranslatedAnswerSelector" label\_drop\_down="<label\_name>"/>

Here’s an example on how to include the language drop-down in the answer detail (answer\_view.php) page:

<div class="rn\_Container">  
 <rn:condition config\_check="OKCS\_ENABLED == true">  
 <div class="rn\_ContentDetail">  
 <rn:meta title="#rn:php:\RightNow\Libraries\SEO::getDynamicTitle('answer',   
 \RightNow\Utils\ Url::getParameter('a\_id'))#" template="okcs\_standard.php"    
 clickstream="answer\_view"/>  
 <div class="rn\_PageTitle rn\_RecordDetail">  
 <rn:widget path="okcs/OkcsProductCategoryBreadcrumb" display\_first\_item="true"/>  
 <div class="rn\_OkcsAnswerAction">  
 <rn:widget path="okcs/SubscriptionButton"/>  
 <rn:widget path="okcs/OkcsRecommendContent"/>  
 <rn:widget path="okcs/OkcsTranslatedAnswerSelector"/>  
 </div>  
 <rn:widget path="okcs/AnswerTitle">  
 </div>  
 <div class="rn\_AnswerView">  
 <rn:widget path="okcs/AnswerStatus">  
 <div class="rn\_SectionTitle"></div>  
 <rn:widget path="okcs/AnswerContent">  
 </div>  
 <div class="rn\_DetailTools rn\_HideInPrint">  
 <div class="rn\_Links">  
 <rn:widget path="okcs/OkcsEmailAnswerLink"/>  
 </div>  
 </div>  
 <rn:widget path="okcs/DocumentRating"/>  
 <rn:widget path="okcs/OkcsRelatedAnswers"/>  
 </div>  
 <aside class="rn\_SideRail" role="complementary">  
 <rn:widget path="okcs/OkcsRecentlyViewedContent"/>  
 </aside>  
 </rn:condition>  
</div>

### Enable Users to Display Answers for All Content Types

You can edit the browse page to add the All option so that users can see articles for all content types. Users can see articles for all the content types or filter articles by a content type. You add the All option by including the display all label attribute in the content type widget.

1. Edit the file:

/cp/customer/development/views/pages/browse.php

1. Add the All option by changing this line:

<rn:widget path="okcs/ContentType"/>

to:

<rn:widget path="okcs/ContentType" display\_all\_content="true" display\_all\_label="<new\_label\_name>" />

### Implement Content Type Subscriptions

Article subscription by content type is implemented by default on the account overview page. If you are using a previous implementation, you can implement content type subscription to enable users to subscribe to articles by content type. When users subscribe to a content type, they get an email every time there is a new or updated article of that type.

Users can limit their subscription to articles about specific products and categories in the content types that they subscribe to.

You implement content type subscription by adding a reference to the content type subscription page.

1. Copy the content\_type\_list.php file from /cp/core/framework/views/pages/okcs/account/notif to /cp/customer/development/views/pages/account/notif.
2. Edit a Customer Portal page, for example, the account overview page:

/cp/customer/development/views/pages/account/overview.php

1. Add the code snippet to refer to the content type subscription page, as follows:

<a href="/app/account/notif/content\_type\_list#rn:session#">#rn:astr:Manage your content type answer notifications#</a>

Here’s an example on how to include a reference to the content type subscription page in the account overview page:

<rn:meta title="#rn:msg:ACCOUNT\_OVERVIEW\_LBL#" template="okcs\_standard.php" login\_required="true" force\_https="true"/>  
  
<div class="rn\_Hero">  
    <div class="rn\_Container">  
        <h1>#rn:msg:ACCOUNT\_OVERVIEW\_LBL#</h1>  
    </div>  
</div>  
  
<div class="rn\_PageContent rn\_AccountOverview rn\_Container">  
    <div class="rn\_ContentDetail">  
        <div class="rn\_Questions">  
            <rn:container report\_id="196" per\_page="4">  
                <div class="rn\_HeaderContainer">  
                    <h2><a class="rn\_Questions" href="/app/account/questions/list#rn:session#">#rn:msg:MY\_SUPPORT\_QUESTIONS\_LBL#/<a></h2>  
                </div>  
                <rn:widget path="reports/Grid" label\_caption="<span class='rn\_ScreenReaderOnly'>#rn:msg:YOUR\_RECENTLY\_SUBMITTED\_QUESTIONS\_LBL#</span>"/>  
                <a href="/app/account/questions/list#rn:session#">#rn:msg:SEE\_ALL\_MY\_SUPPORT\_QUESTIONS\_LBL#</a>  
            </rn:container>  
        </div>  
      
    <div id="rn\_LoadingIndicator" class="rn\_Browse">  
        <rn:widget path="okcs/LoadingIndicator"/>  
    </div>  
    <rn:container source\_id="OKCSBrowse">  
        <div class="rn\_HeaderContainer">  
            <h2><a class="rn\_Profile" href="/app/account/recommendations/list#rn:session#">#rn:msg:MY\_RECOMMENDATIONS\_LBL#</a></h2>  
        </div>  
        <div id="rn\_OkcsManageRecommendations">  
           < rn:widget path="okcs/OkcsManageRecommendations"/>  
            <a href="/app/account/recommendations/list#rn:session#">#rn:msg:SEE\_ALL\_RECOMMENDATIONS\_LBL#</<a>  
        </div>          
    </rn:container>      
</div>  
  
<div class="rn\_SideRail">  
    <div class="rn\_Well">  
        <h3>#rn:msg:LINKS\_LBL#</h3>  
        <ul>  
            <li><a href="/app/account/profile#rn:session#">#rn:msg:UPDATE\_YOUR\_ACCOUNT\_SETTINGS\_CMD#</a></li>  
            <rn:condition external\_login\_used="false">  
                <rn:condition config\_check="EU\_CUST\_PASSWD\_ENABLED == true">  
                    <li><a href="/app/account/change\_password#rn:session#">#rn:msg:CHANGE\_YOUR\_PASSWORD\_CMD#</a></li>  
                </rn:condition>  
            </rn:condition>  
            <li><a href="/app/account/notif/content\_type\_list#rn:session#">#rn:astr:Manage your content type answer notifications#</a></li>  
            <rn:condition is\_active\_social\_user="true">  
                    <li><a href="/app/public\_profile/user/#rn:profile:socialUserID#">#rn:msg:VIEW\_YOUR\_PUBLIC\_PROFILE\_LBL#</a></li>  
            </rn:condition>  
        </ul>  
    </div>  
</div>

### Suppress SmartAssistant Filters

SmartAssistant uses the user-provided product and category as search filters to restrict results by default. You can override the default filtering to use only product, category, or neither to filter search results.

SmartAssistant uses the user-provided product and category as search filters to restrict results. By default, it uses both product and category. When using SmartAssistant to search, you can override the default filtering to use only product, category, or neither.

A hook in the file hooks.php, which contains hook definitions, calls a method defined in Sample.php. To apply a specific choice of filter, or to ignore both filters, modify the files hooks.php and Sample.php.

1. Edit the file:

cp/customer/development/config/hooks.php

1. Add the following code at the end of the file:

$rnHooks['pre\_retrieve\_smart\_assistant\_answers'][] = array(  'class' => 'Sample', // Name of the custom model  'function' => 'retrieveSmartAssistantRequest',  'filepath' => ''  );

1. Edit the file:

cp/customer/development/models/custom/Sample.php

1. Add the following code at the end of the file:

<?php  
namespace Custom\Models;    
require\_once CPCORE . 'Models/Okcs.php';    
  
class Sample extends \RightNow\Models\Okcs   
{  
function \_\_construct()    
{  
parent::\_\_construct();   
}  
public function retrieveSmartAssistantRequest(&$hookData) {  
// Second argument determines which SA results to show. Possible values are 'Product', 'Category', 'ProductAndCategory', 'None'.  
parent::retrieveSmartAssistantRequest(&$hookData, 'ProductAndCategory');  
}  
}

### Enable Users to Open an Article Attachment

Users can open file attachments of an article in a separate browse window with the answer page. You implement the answer page by adding answer.php from the reference implementation to your current instance.

1. Copy the answer.php file from here:

/cp/core/framework/views/admin/okcs/

1. Place it here:

/cp/customer/development/views/admin/

### Enable Agents to Open an Article from Agent Desktop

You can enable the support agents to open articles that users have viewed before submitting their question using the Ask a Question page. You enable agents to open articles in Agent Desktop by adding the answer preview page (okcs\_answer\_full\_preview.php) from the reference implementation to your current instance.

1. Copy the okcs\_answer\_full\_preview.php file from here:

/cp/core.framework/views/admin/okcs/

1. Place it here:

/cp/customer/development/views/admin/

## Add Knowledge to Website Pages

You can use the knowledge syndication widget to add knowledge to pages on your organization’s website in addition to the customer portal. This enables users to search for and use knowledge from the most relevant places on your web page. You can set up the knowledge widget to display articles by content type, product, and category, popular and recent articles, and external articles.

You add the knowledge widget by configuring the widget, then pasting the JavaScript code that the widget generates into your organization’s web page.

### Configure Knowledge Widget

Use the Configure Widget section of the widget configuration page to configure the knowledge widget.

1. Go to the your site’s administration page:

https://<your\_site>/ci/admin

1. Click Widgets, then Syndicated Widgets.
2. Click OkcsKnowledgeSyndication.
3. Configure the OkcsKnowledgeSyndication widget attributes using the Configure Widget section.

You can see the list of the widget attributes and their descriptions on your site’s administration page: https://<your\_site>/ci/tags/syndicated\_widgets/standard/OkcsKnowledgeSyndication.

1. Click Apply.
2. Preview your changes in the Preview this Configuration section.

### Add the Widget to a Page

Use the Copy Script code to your page section of the widget configuration page to copy and paste the syndication widget code into your web page.

1. Open the source code for the web page that you want to add the widget to.
2. Click the Select Text button to select the code that defines your site URL, then press Ctrl+c to copy the code.

Here’s an example of the code that includes your site URL:

<script type="text/javascript" scr="https://<your\_site>/euf/rightnow/RightNow.Client.js"></script>

1. Paste the code into the source code of your web page just before the closing </body> tag.
2. Click the Select Text button to select the code that defines the widget and its attributes, then press Ctrl+c to copy the code.

Here’s an example of the JavaScript code that includes the widget configuration:

<script type="text/javascript">  
 RightNow.Client.Controller.addComponent(  
 {  
 div\_id: "myDiv",  
 enable\_recent: true,  
 presist\_contenttype: false,  
 persist\_prodcat: false,  
 q: "oracle",  
 instance\_id: "okcs\_0",  
 module: "OkcsKnowledgeSyndication",  
 type: 10  
 }  
 "https://<your\_site>/ci/ws/get"  
 );  
</script>

1. Paste the copied code just below the code you copied in the previous step.
2. Click the Select Text button to select the <div> tag code, then press Ctrl+c to copy the code. This code calls the JavaScript code of the widget generated in the previous step.

Here’s an example of the <div> tag code:

<div id="myDiv"></div>

1. Place the <div> tag within the </body> tag

where you want the widget to appear on the web page.

## About Community Discussions

Community Self Service in Service Cloud provides a platform for community and peer-to-peer knowledge sharing. It provides self-service users with an additional channel that community members can use to ask questions, provide answers, and discuss issues.

You can add community discussions to knowledge in your current instance to enable self-service users to get answers from discussions along side knowledge base answers. Community discussions are organized using the same product hierarchy as knowledge base articles.

You can include community discussions on the home page, the results page, and in the Knowledge Advanced version of SmartAssistant. When users search for answers on these pages, their answers will include both knowledge base articles and community discussions.

You can find more information about Community Self Service in Service Cloud User Guide. See [Documentation for Oracle Service Cloud Products (Answer ID 5168)](https://cx.rightnow.com/app/answers/detail/a_id/5168) to locate the documentation for your specific release, then use the Documentation Library or the Online Help User Guide link to locate the relevant topics.

### Add Community Discussions to the Home Page

Edit the home page so that users can to see community discussions in addition to most popular answers and most recent answers.

1. Edit the file: /cp/customer/development/views/pages/home.php
2. Add community discussions by adding this code:

<div class="rn\_PopularSocial">  
<div class="rn\_Container">  
<h2>#rn:msg:RECENT\_COMMUNITY\_DISCUSSIONS\_LBL#</h2>  
<rn:widget path="discussion/RecentlyAnsweredQuestions" show\_excerpt="true" maximum\_questions="5"/>  
<span class="rn\_DiscussionLink">  
<a href="/app/social/questions/list/kw/\*#rn:session#">#rn:msg:SHOW\_MORE\_COMMUNITY\_DISCUSSIONS\_LBL#</a>  
</span>  
</div>  
</div>

### Add Community Discussions to the Results Page

Edit the results page so that users can see community discussions in addition to the articles in search results.

1. Edit the file: /cp/customer/development/views/pages/results.php
2. Add community discussions by adding this code:

<div class="rn\_Container">  
<div class="rn\_QuestionResults">  
<rn:container source\_id="SocialSearch" per\_page="5">  
<rn:widget path="searchsource/SourceResultDetails"/>  
<rn:widget path="searchsource/SocialResultListing" label\_heading="#rn:msg:RESULTS\_FROM\_THE\_COMMUNITY\_LBL#" more\_link\_url="/app/social/questions/list"/>  
</rn:container>  
</div>  
</div>

### Add Recently Viewed Articles to the Social Page

Users can see the discussions about a selected community questions on the social detail page. You can add recently viewed knowledge base articles to the social detail page so that they will see recently viewed articles in addition to community discussions. You add them by including the recently viewed answers widget in the social detail page.

Users can see five recently viewed articles by default. You can set the limit to a value to five or fewer using the widget’s content count attribute.

1. Edit the file:

cp/customer/development/views/pages/social/questions/detail.php

1. Add the recently viewed articles widget code by adding this line:

<rn:widget path="okcs/OkcsRecentlyViewedContent"/>

within this section:

<div class="rn\_SideRail">

### Add Related Answers to the Social Page

Users can see the discussions about a selected community question on the social detail page. You can add related knowledge base answers so that they will also see relevant articles in addition to community discussions. You add them by including the related answers widget in the social detail page.

Users can see up to ten related articles by default. You can set the limit number of articles to ten or fewer using the widget's limit attribute.

1. Edit the file: cp/customer/development/views/pages/social/questions/detail.php
2. Add this line:

<rn:widget path="okcs/RelatedKnowledgeAdvanceAnswers" limit="5"/>

to this section:

<div class="rn\_SideRail">

### Add Community Discussions to SmartAssistant

The Knowledge Advanced version of SmartAssistant includes community discussions as well as knowledge base answers. When users submit a question, Knowledge Advanced search returns answers and social search returns discussions that match the search terms that were entered.

Knowledge Advanced uses its own version of SmartAssistant. You can add community discussions to the Knowledge Advanced SmartAssistant so that users can see both community content and knowledge base articles in search results. You update SmartAssistant by editing the KFAPI\_SSS\_ENABLED configuration setting. The default value of the configuration setting is No.

1. Go to Site Configuration, Configuration Settings.
2. Set the value of the configuration setting KFAPI\_SSS\_ENABLED to Yes.

## Implementing Knowledge Advanced on Mobile Customer Portal

You enable Knowledge Advanced in an existing Customer Portal mobile implementation by replacing or modifying the existing Customer Portal mobile pages. You can replace the entire set of Customer Portal files with the Knowledge Advanced mobile reference implementation. However, the new Knowledge Advanced mobile files will overwrite any customized files in your current implementation. You can preserve your customized files by using the Knowledge Advanced mobile files as examples on how to modify the files instead of replacing them.

### Add Knowledge Advanced Search to the Mobile Home Page

You can add the mobile search (OKCSSearch) functionality so that users can search for knowledge base articles from the home page. You implement the search functionality by replacing the mobile home page in your existing instance with the reference implementation mobile home page.

1. Copy the reference implementation mobile home page from here:

/cp/core/framework/views/pages/okcs/mobile/home.php

1. Place it here:

/cp/customer/development/views/pages/mobile

Here’s an example of the reference implementation mobile home page:

<rn:meta title="#rn:msg:SHP\_TITLE\_HDG#" template="okcs\_mobile.php" clickstream="home"/>  
<div class="rn\_Hero">  
    <div class="rn\_HeroInner">  
        <div class="rn\_HeroCopy">  
            <h1>#rn:msg:WERE\_HERE\_TO\_HELP\_LBL#</h1>  
        </div>  
        <div class="rn\_SearchControls">  
            <h1 class="rn\_ScreenReaderOnly">#rn:msg:SEARCH\_CMD#</h1>  
            <form method="get" action="/app/results">  
                <rn:container source\_id="OKCSSearch">  
                    <div class="rn\_SearchInput">  
                        <rn:widget path="searchsource/SourceSearchField" initial\_focus="true" label\_placeholder="#rn:msg:ASK\_A\_QUESTION\_ELLIPSIS\_MSG#"/>  
                    </div>  
                    <rn:widget path="okcs/RecentSearches"/>  
                    <rn:widget path="okcs/OkcsSuggestions"/>  
                    <rn:widget path="searchsource/SourceSearchButton" initial\_focus="true" search\_results\_url="/app/results"/>  
                </rn:container>  
            </form>  
        </div>  
    </div>  
</div>  
<div class="rn\_PageContent rn\_Home">  
    <div class="rn\_Container">  
        <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered\_pagination="true"/>  
    </div>  
    <div class="rn\_PopularKB">  
        <div class="rn\_Container">  
            <rn:widget path="okcs/AnswerList" type="popular" target="\_self" view\_type="list"/>   
        </div>  
    </div>  
    <div class="rn\_PopularKB rn\_RecentKB">  
        <div class="rn\_Container">  
            <rn:widget path="okcs/AnswerList" type="recent" target="\_self" view\_type="list"/>   
        </div>  
    </div>  
</div>

### Add Mobile Knowledge Advanced Browse

You must implement the mobile browse (OKCSBrowse) functionality so that users can browse articles and filter the articles by content type, product, and category. You implement the browse functionality by adding the browse page from your reference implementation to your existing instance.

1. Copy the reference implementation mobile browse page from here:

/cp/core/framework/views/pages/okcs/mobile/browse.php

1. Place it here:

/cp/customer/development/views/pages/mobile/

Here’s an example of the reference implementation mobile browse (browse.php) page:

<rn:meta title="#rn:msg:BROWSE1\_AB\_HDG#" template="okcs\_mobile.php" clickstream="answer\_list"/>  
<rn:condition config\_check="OKCS\_ENABLED == true">  
<div id="rn\_PageTitle" class="rn\_Home">  
    <div class="rn\_Hero">  
        <div class="rn\_HeroInner">  
            <div class="rn\_OkcsMobileBrowseHeader">  
                <h1 class="rn\_ScreenReaderOnly">rn:page\_title/></h1>  
                <rn:container source\_id="OKCSBrowse">  
                    <h2 id="rn\_AccordTriggerContentType" class="rn\_Collapsed"><a href="javascript:void(0);" role="button">#rn:msg:CONTENT\_TYPE\_LBL#span class="rn\_ButtonOff">/span></a></h2>  
                    <h2 id="rn\_AccordTriggerProduct" class="rn\_Collapsed"><a href="javascript:void(0);" role="button">#rn:msg:PRODUCT\_LBL#span class="rn\_ButtonOff">/span></a></h2>  
                    <h2 id="rn\_AccordTriggerCategory" class="rn\_Collapsed"><a href="javascript:void(0);" role="button">#rn:msg:CATEGORY\_LBL#span class="rn\_ButtonOff">/span></a></h2>  
                       <div class="rn\_ClearBoth">  
                        <div id="rn\_ContainerContentType" class="rn\_Hidden">  
                            <rn:widget path="okcs/ContentType" list\_display="vertical" toggle\_selection="true" toggle="rn\_AccordTriggerContentType" item\_to\_toggle="rn\_ContainerContentType"/>  
                        </div>  
                        <div id="rn\_ContainerProduct" class="rn\_Hidden">  
                            <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter\_type="products" toggle\_selection="true" toggle="rn\_AccordTriggerProduct" item\_to\_toggle="rn\_ContainerProduct" view\_type="explorer"/>  
                        </div>  
                        <div id="rn\_ContainerCategory" class="rn\_Hidden">  
                            <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter\_type="categories" toggle\_selection="true" toggle="rn\_AccordTriggerCategory" item\_to\_toggle="rn\_ContainerCategory" view\_type="explorer"/>  
                        </div>  
                    </div>  
                </div>  
           </div>  
    </div>    
</div>  
<div class="rn\_Container">  
<section id="rn\_PageContent" class="rn\_Home">  
        <div id="rn\_LoadingIndicator" class="rn\_Browse">  
           <rn:widget path="okcs/LoadingIndicator"/>  
        </div>  
        <div id="rn\_PageContentArticles">  
            <div class="rn\_ResultPadding">  
                <div id="rn\_Browse\_Loading">/div>  
                <div id="rn\_OkcsAnswerList">  
                    <div class="rn\_Report">  
                        <rn:widget path="okcs/OkcsRecommendContent"/>  
                        <rn:widget path="okcs/AnswerList" view\_type="list" show\_headers="false" per\_page="5" target="\_self"/>  
                    </div>  
                    <div class="rn\_FloatRight">  
                        <rn:widget path="okcs/OkcsPagination"/>  
                    </div>  
                </div>  
            </div>  
        </div>  
    </rn:container>  
</section>  
</div>  
</rn:condition>

### Add the Mobile Knowledge Advanced Answer Detail Page

You must implement the mobile answer detail (answer\_view.php) page to enable users to open articles with the article details page. Users can see the article content and its details, including document ID, Version, Status, and Published Date using the article details page.

You implement the mobile answer detail page by adding answer\_view.php from reference implementation to your existing instance.

1. Copy the reference implementation mobile answer detail from here:

/cp/core/framework/views/pages/okcs/mobile/answers/answer\_view.php

1. Place it here:

/cp/customer/development/views/pages/mobile/answers/

The following is an example of the reference implementation mobile answer detail (answer\_view.php) page:

<div class="rn\_Container">  
    <rn:condition config\_check="OKCS\_ENABLED == true">  
        <rn:widget path="okcs/OkcsProductCategoryBreadcrumb" display\_first\_item="false"/>  
        <rn:meta title="#rn:php:\RightNow\Libraries\SEO::getDynamicTitle('answer', \RightNow\Utils\Url::getParameter('a\_id'))#" template="okcs\_mobile.php"  clickstream="answer\_view"/>  
        <rn:widget path="okcs/OkcsRecommendContent"/>  
        <rn:widget path="okcs/SubscriptionButton"/>  
        <section id="rn\_PageContent" class="rn\_AnswerDetail">  
            <div id="rn\_AnswerText">  
                <rn:widget path="okcs/AnswerTitle">  
                <rn:widget path="okcs/AnswerStatus">  
                <div class="rn\_SectionTitle"></div>  
                <rn:widget path="okcs/AnswerContent">  
            </div>  
            <div class="rn\_DetailTools rn\_HideInPrint">  
                <div class="rn\_Links">  
                    <rn:widget path="okcs/OkcsEmailAnswerLink" />  
                </div>  
            </div>  
            <rn:widget path="okcs/DocumentRating"/>  
            <rn:widget path="okcs/OkcsRelatedAnswers"/>  
        </section>  
    </rn:condition>  
</div>

### Add the Mobile Knowledge Advanced Results Page

You must implement the mobile Knowledge Advanced results page so that users can see knowledge articles in search results. You implement the search functionality by replacing the mobile results page in your current instance with the reference implementation mobile results page.

Users can also use the interactive spell checker on the results page to either correct misspelled words automatically, or to suggest spelling corrections to the user-entered text.

1. Copy the reference implementation mobile results page from here:

/cp/core/framework/views/pages/okcs/mobile/results.php

1. Place it here:

/cp/customer/development/views/pages/mobile/

Here’s an example of the reference implementation mobile results (results.php) page:

<rn:meta title="#rn:msg:FIND\_ANS\_HDG#" template="okcs\_mobile.php" clickstream="answer\_list"/>  
<rn:container source\_id="OKCSSearch" document\_id\_reg\_ex="^\w{1,10}\d{1,10}$">  
<div id="rn\_PageTitle" class="rn\_Search">  
    <div class="rn\_Hero">  
        <div class="rn\_HeroInner">  
            <div class="rn\_SearchControls">  
                <h1 class="rn\_ScreenReaderOnly">#rn:msg:SEARCH\_CMD#</h1>  
                form method="get" action="/app/results">  
                    <rn:container source\_id="OKCSSearch">  
                        <div class="rn\_SearchInput">  
                            <rn:widget path="searchsource/SourceSearchField" initial\_focus="true" label\_placeholder="#rn:msg:ASK\_A\_QUESTION\_ELLIPSIS\_MSG#"/>  
                        </div>  
                        <rn:widget path="okcs/RecentSearches" display\_tooltip="true"/>  
                        <rn:widget path="okcs/OkcsSuggestions"/>  
                        <rn:widget path="searchsource/SourceSearchButton" initial\_focus="true" history\_source\_id="OKCSSearch"/>  
                        <rn:widget path="okcs/OkcsInteractiveSpellChecker"/>  
                    </rn:container>  
                    <div class="rn\_StartOver">  
                        <a href="/app/home" >#rn:msg:START\_OVER\_LBL#</a>  
                    </div>  
                </form>  
            </div>  
        </div>  
    </div>    
</div>  
</rn:container>  
<div class="rn\_Container">  
    <section id="rn\_PageContent" class="rn\_Container">  
        <rn:container source\_id="OKCSSearch" truncate\_size="200">  
            <div class="rn\_Module">  
                <rn:widget path="okcs/Facet" toggle\_title="true"/>  
            </div>  
            <div id="rn\_OkcsRightContainer" class="rn\_Padding">  
                <rn:widget path="okcs/SearchResult" hide\_when\_no\_results="false"/>  
                <div class="rn\_FloatRight rn\_Padding">  
                    <rn:widget path="okcs/OkcsPagination"/>  
                </div>  
            </div>  
        </rn:container>  
    </section>  
</div>

### Add Knowledge Advanced SmartAssistant

You must add the mobile ask a question (ask.php) page to implement the Knowledge Advanced version of SmartAssistant. Users can submit a question to a support agent from the ask a question page. Users can also see knowledge articles as well as community discussions that match the user-entered text before submitting the question.

You implement mobile SmartAssistant by replacing the mobile ask a question page in your current instance with the reference implementation mobile ask a question page.

1. Copy the reference implementation mobile ask a question page from here:

/cp/core/framework/views/pages/okcs/mobile/ask.ph

1. Place it here:

/cp/customer/development/views/pages/mobile/

Here’s an example of the reference implementation mobile ask a question (ask.php) page:

<rn:meta title="#rn:msg:ASK\_QUESTION\_HDG#" template="okcs\_mobile.php" clickstream="incident\_create"/>  
<div class="rn\_Hero">  
   <div class="rn\_HeroInner">  
      <div class="rn\_HeroCopy">  
         <h1>#rn:msg:SUBMIT\_QUESTION\_OUR\_SUPPORT\_TEAM\_CMD#</h1>  
         <p>#rn:msg:OUR\_DEDICATED\_RESPOND\_WITHIN\_48\_HOURS\_MSG#</p>  
      </div>  
   </div>  
</div>  
  
<div class="rn\_PageContent rn\_AskQuestion rn\_Container">  
   <form id="rn\_QuestionSubmit" method="post" action="/ci/ajaxRequest/sendForm">  
      <div id="rn\_ErrorLocation">/div>  
      <rn:condition logged\_in="false">  
      <rn:widget path="input/FormInput" name="Contact.Emails.PRIMARY.Address" required="true" initial\_focus="true" label\_input="#rn:msg:EMAIL\_ADDR\_LBL#"/>  
      <rn:widget path="input/FormInput" name="Incident.Subject" required="true" label\_input="#rn:msg:SUBJECT\_LBL#"/>  
      </rn:condition>  
      <rn:condition logged\_in="true">  
      <rn:widget path="input/FormInput" name="Incident.Subject" required="true" initial\_focus="true" label\_input="#rn:msg:SUBJECT\_LBL#"/>  
      </rn:condition>  
      <rn:widget path="input/FormInput" name="Incident.Threads" required="true" label\_input="#rn:msg:QUESTION\_LBL#"/>  
      <rn:widget path="input/FileAttachmentUpload"/>  
      <rn:widget path="input/MobileProductCategoryInput" name="Incident.Product"/>  
      <rn:widget path="input/MobileProductCategoryInput" name="Incident.Category"/>  
      <rn:widget path="input/FormSubmit" label\_button="#rn:msg:SUBMIT\_YOUR\_QUESTION\_CMD#" on\_success\_url="/app/ask\_confirm" error\_location="rn\_ErrorLocation"/>  
      <rn:condition content\_viewed="2" searches\_done="1">  
      <rn:condition\_else/>  
      <rn:widget path="okcs/OkcsSmartAssistant" accesskeys\_enabled="false" view\_type="inline"/>  
      </rn:condition>  
   </form>  
</div>