Extending Oracle CX Commerce Agent Console

F32071-01
August 2020

Extending Oracle CX Commerce Agent Console,

F32071-01

Contents

1

About This Guide

Creating widgets allows you to customize your agent environment. Information within this guide should help you begin working on these widgets.

This guide is intended for site developers working on Oracle CX Commerce Agent Console implementations. It explains, in general terms, the widget development framework that you can use to customize agent functionality. It also provides some examples of common customizations.

For information on developing widgets for specific features, see Integrate with an External Pricing System or Integrate with a Web Checkout System which builds on the information contained in this guide.

This guide also provides direction on various tasks that you can perform using the REST Agent API. This functionality may or may not be available using the Agent administration interface.

For information on working with the Agent Console in general, see Understand the Agent Console.

Assumptions

Before you create widgets for your agent environment, you should be familiar with specific technologies.

This guide assumes that you are comfortable with site administration, have experience with the following technologies and have reviewed the following documentation:

- JavaScript
- Data binding using Knockout
- Bootstrap
- Standard CSS and CSS Less
- MVVM Architecture
- Familiarity with the Developing Widgets guide, which provides a detailed overview of how widgets work in Oracle CX Commerce
- Familiarity with the Using Oracle CX Commerce Agent Console guide, which contains information on working with the agent console

About extensions

Oracle CX Commerce Agent Console provides tools that you can use to extend the capabilities of your system.

The tools consist of the following:

- Widgets that allow you to extend the functionality of your agent's environment by accessing features that are not exposed by default. This document focuses

on describing agent console-specific widgets. For general information on other widgets, see Developing Widgets.

- An extensive set of REST APIs allows external applications to make calls into the Oracle CX Commerce server. This document focuses on agent-specific API.

Work with widgets

Widgets, which are a type of server extension, provide functionality that is deployed on pages of the agent's console.

Widgets allow you to display content to your agents or to execute specific functions. Widgets are comprised of templates, JavaScript, CSS, locale resources and images. For detailed information on widgets and extensions, refer to the Developing Widgets guide.

When you customize widgets, you create functionality that appears on your website pages. These widgets allow you to add pre-made and pre-configured snippets of text or information to your web pages without having to recreate them for each page. Widgets also allow you to extend the functionality of your storefront by communicating with the Oracle CX Commerce Agent server to access a variety of features.

You can use agent-specific widgets to create a customized agent environment. Widgets are comprised of a set of resource and source files. Using these files, as well as auxiliary files that contain information such as meta-data, you can customize an agent's environment.

The agent environment uses agent-specific widgets, as well as Oracle Cloud Commerce storefront widgets. There are separate instances for agent and storefront widgets. Note that these instances differ in the template, and sometimes the configuration properties, they use. If you delete an agent widget instance, which uses a different template than the Display template, you will be unable to recreate the widget.

Important: Please note that Oracle CX Commerce Agent Console, unlike Oracle CX Commerce Storefront, does not support application JS, global widgets, or global elements. Any element used in default widgets should be downloaded and customized.

Tasks done by widgets

The following are some general tasks that can be performed by widgets:

- Agent Navigation - Enables an agent to navigate throughout the console to review a variety of data. Additionally, you can rename fields and buttons, and hide or display certain fields or navigational items. For information on working with navigation in the Agent Console, refer to Navigate the Agent Console. Configure customer and order searchFor information on working with agent navigation widgets, refer to the Work with Navigation Widgets section.
- Search - Allows an agent to search for customers, orders or returns. For information on using Search in the agent console, refer to For information on working with agent search widgets, refer to the Configure Search Widgets section.
- Create Orders - You can display navigation, customer cart dialog, shipping details, address books, promotions, loyalty details, scheduled orders and product details, in addition to other displays. For information on working with agent order widgets, refer to the Create and Edit Orders Widgets section.

- Order Details - You can create widgets to display order details for account orders, return and exchange orders, quotes and order approvals. You can also customize carts to show subtotals and logic for payment and pricing. For information on working with order details widgets, see Order Details. For information on working with agent order detail widgets, refer to Review order details during checkout.
- Customer Details - You can create customized widgets to display account addresses and details, customer profile information, order histories, purchase lists, account contacts and other profile information. For information on working with agent customer details widgets, refer to the Work with Customer Information Widgets section.
- Returns and Exchanges - You can create widgets that display or create return requests and refunds, as well as process returns. You can also create widgets that create exchanges. For information on working with returns and exchange-specific widgets, Return Items. For information on working with agent returns and exchanges widgets, refer to the Configure Returns and Exchanges Widgets section.

This documentation provides the following information for each widget:

- Description- A description of the widget.
- Widget Name - The code name of the widget. The name of the widget is identified in the `widget.json` file. For example:

  ```
  "javascript": "agent-order-search",
  "widgetFamily": "agentOrderSearch",
  "widgetType": "agentOrderSearch"
  ```

- Display Name - The widget's name as displayed in the Design page. The display name can also be found in the `widget.json` file:

  ```
  "name": "Order Search - Agent"
  ```

- Supported Page Type - The types of pages that the widget can be applied to. This is also identified in the `widget.json` file:

  ```
  "pageTypes": ["agentOrderSearch"],
  ```

  For information on page types, refer to Define stack meta-data in stack.json.
- Layouts - The page layouts that this widget is associated with. A widget's associated layouts are identified on the Layout tab of the Design page. For detailed information on layouts, see Create Page Layouts that Support Different Types of Shoppers.
- Elements - The elements used by this widget. Each element represents a single piece of the structure of the widget. Elements can be configured as drag-and-drop sub-components, which allow you to control their location on a page's layout. Elements are added to the widget's `display.template` file and can be reviewed using the Design tab code view.
  For detailed information on elements, see Fragment a Widget into Elements.

For information on Storefront widgets refer to Appendix: Layout Widgets and Elements.

Access widgets

You can review Oracle CX Commerce default widgets by opening the Design page on your administration server. This page displays both the layouts that are used in the agent console, but also the components, such as widgets, that are available. Click the Component tab to see the list of widgets.

Each widget displays the associated page layouts. By selecting the page layouts, you can customize the pages by adding or removing widgets. For information on working with widgets and layouts, see Appendix: Layout Widgets and Elements.

Note that widgets that are specific to the agent console contain the word “Agent”. The agent console also uses a number of widgets that are also used by the Storefront.

Extend functionality with widgets

You can customize widgets by downloading the widget source code and making modifications to the JavaScript and HTML files. For example, you could add custom JavaScript to an existing widget. Once you have made all of your changes, you then upload the widget by adding it to an extension as described in Understand Extension Features.

Important: When creating or renaming a widget, ensure that the name is less than 50 characters. Errors will occur if the widget name is greater than 50 characters.

You can also extend a widget’s JavaScript by using the JavaScript Code Layering feature that allows you to layer custom JavaScript on top of the widget. For information on the JavaScript Code Layering feature, see Use the JavaScript Code Layering User Interface feature.

Note: For a widget’s JavaScript code to be editable, ensure that the `jsEditable` flag in the `widget.json` file is set to true. By default, the flag is set to `false`, indicating that the JavaScript associated with the widget cannot be edited.

When you upgrade or modify a widget, you must remove any existing instances of it from the page layouts and replace it with the new widget.

Upgrade widgets

Widgets that have been deployed are not automatically updated when a newer version of the widget is released. This allows you to customize widgets without the fear of them being overwritten. To update a widget, you must remove the old widget from all of your page layouts and replace them with the new widget. Additionally, you must recreate templates or style sheets that you have customized for the new widget.

For detailed information on how to upgrade widgets, see Customize your store layouts.

Migrate widgets with multiple templates

Some widgets use more than one template. If you migrate your widgets and load the latest version of the widget, the agent-specific template is not available for newly created widget instances. To update the widget with the new version, you must ensure that the widget's new template file is recognized by the layout.

To do this, perform the following steps:

1. Open the widget in the administrative interface using the **Settings** icon.
2. Open the About tab. Click the **Go to widget code** button.

3. Click the **Download Source** button.
4. Unzip the file that you downloaded.
5. Open the `/widget/widgetName/layouts/layoutThatContainsTheLayout` file and copy the contents of the newer version of the `widget-layoutAndVersion.template` file.
   For example, you might copy the `widget-orderDetailsDefaultLayout_v3.template` file.
6. Return to the administrative interface from where you downloaded the source. In the **Template** tab, replace the existing code with the code you have copied from the new version of the `widget.template` file.
7. Click **Save**.
8. Ensure that you update the associated layouts.

Migrate widget with multiple LESS files

If you have recently migrated, your widgets will be listed in the page layout components. However, if you use a widget in a page layout region, and then create a new widget instance, the widget instance will be created with the default template, which won't be applicable to the page layout. To update these widgets, perform the following steps:

1. Open the widget in the administrative interface using the **Settings** icon.
2. Open the About tab. Click the **Go to widget code** button.
3. Click the **Download Source** button.
4. Unzip the file that you downloaded.
5. Open the `/widget/widgetName/less/` file and copy the contents of the newer version of the `widget.less` file.
   For example, you might copy the `widget-agentAccountOrderDetailsInst_v4.less` file.
6. Return to the administrative interface from where you downloaded the source for the widget. In the **Style Sheet** tab, replace the code with the code you have copied from the new version of the LESS file.
7. Click **Save**.
8. Ensure that you update the associated layouts.

Recreate shared widgets after deletion

Some widgets also have corresponding storefront widgets. These widgets, although designed to perform similar tasks, use different templates and, sometimes, configuration properties. If an agent-specific widget is deleted, and the agent widget uses a different template than the storefront widget, you will not be able to recreate the widget from the administrative interface.

To rectify this, download the agent-specific widget and recreate the template for the instance.

Use the REST Agent API

Oracle CX Commerce uses REST APIs that consist of several sets of endpoints, which allow you to perform storefront, administrative and agent tasks.

These include Store, Admin, Social Wish List and Agent APIs.

The Agent API endpoints provide access to agent-specific functionality on the administration server. These endpoints can be used in conjunction with agent-specific widgets, for example to pass a response filter key in a REST call that was made from a widget.

Note that there are many similar endpoints that exist in each API. For example, each set of APIs may have endpoints for working with customers, although each endpoint differs in the functionality that they perform.

You can access a `ccdebug` REST client on your administration server in your test environment. This client is available at the following URL:

```
http://<admin-server-hostname>/ccebug
```

Each API is available only in certain environments. The Agent API is available only on administration servers. You can find information on endpoints in the REST API documentation, which is available through the Oracle Help Center at the following URL:

```
http://docs.oracle.com/cloud/latest/commercecs_gs/CX0CC/
```

Note that the documentation on the Oracle Help Center reflects the most recent version of Oracle CX Commerce. If you are using an earlier version of Commerce, the API documentation on the Oracle Help Center may include endpoints that are not available on your version.

For additional information on working with REST APIs, see Use the REST APIs.

Find additional resources

Oracle CX Commerce provides widgets for customizing your agent environment. Information is available throughout this document.

You can find additional information on working with widgets in the following documents:

- Developing Widgets - Describes how to work with and implement widgets in your environment. This document contains many examples of how to implement and customize widgets.
- Extending Oracle CX Commerce - Describes various task-specific widgets, as well as how to use REST APIs for customization.
- Layout Widgets and Elements - Provides information on Storefront-specific widgets.
- Using Oracle CX Commerce Agent Console - Provides information specific to the Oracle CX Commerce Agent Console user interface that can be customized using widgets.

2

Work with Navigation Widgets

Oracle CX Commerce provides default widgets that provide navigation.

This allows you to customize the way that agents navigate around the application.

Note: While navigation can be customized for the agent, the Log In page cannot be customized.

Customize agent navigation

This widget allows you to customize the navigation bar at the top of the page.

This widget is specifically for navigation between agent pages.

Widget Name: `agentNavigation`

Display Name: `Navigation - Agent`

Supported Page Types:

- `agentAccountContacts`
- `agentAccountDetails`
- `agentAddressBook`
- `agentCheckout`
- `agentCreateExchange`
- `agentCreateReturn`
- `agentCustomerDetails`
- `agentCustomerSearch`
- `agentHome`
- `agentMultiShipCheckout`
- `agentOrderDetails`
- `agentOrderHistory`
- `agentOrderSearch`
- `agentOrdersPendingApproval`
- `agentProcessReturns`
- `agentPurchaseList`
- `agentRefunds`
- `agentRegtistrationRequestDetail`
- `agentReturnSearch`
- `agentScheduledOrder`

- `agentSelfRegistrationPage`
- `agentViewReturnRequest`

Layouts:`All`

Elements: `None`

Note: The Customer Profile page, and the Registration Request page, do not contain a back navigation button. To provide a button, you must create a custom property on the navigation widget.

Customize guided navigation

This widget presents an overlayed guided navigation to the agent.

See Understand the Agent Consolefor information on working with agent navigation.

Widget Name: `overlayedGuidedNavigation`

Display Name: `Overlayed Guided Navigation`

Supported Page Types: `category, searchResults`

Layouts:

- Collection Layout – Agent
- No Search Results Layout – Agent
- Search Results Layout - Agent

Elements: None

Customize the agent dashboard

This widget displays the agent dashboard.

This widget allows you to update the KPT/Graphics, announcements, quick links, recent orders and return requests, as well as pending actions. This widget displays updated information, information on recent customers and a quick search for customers and orders. For information on working with the agent dashboard, see Navigate the Agent Console.

Widget Name: `agentDashboard`

Display Name: Dashboard - Agent

Supported Page Types: `agentHomePageType`

Layouts: Home Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `agent-pending-actions`
- `announcements`
- `ordersChart`
- `price-group-drop-down`

- `quick-links`
- `recent-orders`
- `returnRequestsChart`

Present errors

This widget presents errors if it encounters an invalid `siteId`, `organizationId` or `profileId`.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template file, which may not be available to the page layout.

To perform this upgrade, refer to the instructions in Work with widgets.

Widget Name: `agentError`

Display Name: `Page Not Found Widget - Agent`

Supported Page Types: All

Layouts:

- Error Layout - Agent

Elements: None

3
Configure Search Widgets

The widgets described in this section allow you to customize agent searches for customers, returns and orders. These widgets also provide you with the following functionality:

- The ability to determine which search attributes are displayed.
- The ability to rename the labels of search fields and results tables.
- You can rearrange sections or fields to customize the search page.
- You can choose to display only relevant columns for search results.
- You may add other widgets or elements within the search page.

Search for a customer

The Customer Search - Agent widget allows you to customize how an agent searches for a customer.

It holds criteria needed to search through customer, account and custom profile level attributes. This widget also displays the search button that the agent uses to initiate the search. For detailed information on searching for a customer in the agent console, see Search for Customers and Orders.

Widget Name: `agentCustomerSearch`

Display Name: Customer Search - Agent

Supported Page Types: `agentCustomerSearchPageType`

Layouts: Customer Search Layout - Agent

Elements: This widget uses the `account-search` element. To understand how this element is used, refer to the widget's code view.

For information on working with agent-based customer widgets, refer to the Work with Customer Information Widgets section.

Search for returns

The Return Search - Agent widget allows you to configure what an agent sees when searching for returns.

It holds all of the applicable criteria needed to search through return request, account and custom profile level attributes. This widget also provides a search button that allows the agent to initiate the search, while displaying both navigation and cancel elements. For detailed information on searching for a return request in the agent console, see Process returns.

Widget Name: `agentReturnSearch`

Display Name: Return Search - Agent

Supported Page Types: `agentReturnSearch` page

Layouts: Returns Search Layout - Agent

Elements: This widget uses the `account-search` element. To understand how this element is used, refer to the widget's code view.

For additional information on agent-based return widgets, refer to the Configure Returns and Exchanges Widgets section.

Search for orders

The Order Search - Agent widget allows you to customize order searches.

It holds all of the applicable criteria needed to search through return request, account and custom profile level attributes, as well as advanced search attributes, such as shipping or billing addresses. This widget also provides a search button that allows the agent to initiate the search, while displaying both navigation and cancel elements. For detailed information on searching for an order in the agent console, see Search for Customers and Orders.

Widget Name: `agentOrderSearch`

Display Name: Order Search - Agent

Supported Page Types: `agentOrderSearch` page

Layouts: Order Search Layout - Agent

Elements: This widget uses the following elements. To understand how these elements are used, refer to the widget's code view:

- `account-search`
- `product-search`

For additional information on agent-based order widgets, refer to the Create and Edit Orders Widgets section.

Find search results

The No Search Results - Agent widget is displayed to the agent when no search results are found.

Widget Name: `noSearchResults`

Display Name: No Search Results- Agent

Supported Page Types: `noSearchResults`

Layouts: No Search Results Layout - Agent

Elements: None

Customize search pages

Agents spend a great deal of time searching for customer information. By customizing your search environment, you can make these searches more efficient.

One of the most common ways that an agent obtains customer information is through searches. Therefore, you may want to customize the search pages so that you can determine what an agent needs when obtaining pertinent information.

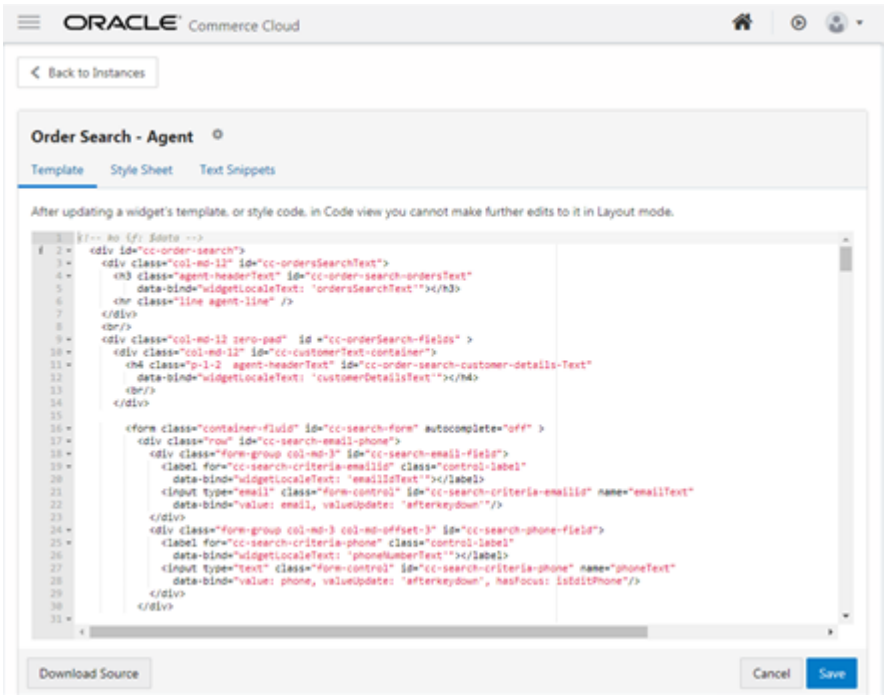
Before making any customizations, you should be familiar with how widgets are downloaded, created and added to the server. For detailed information on widgets, see Developing Widgets.

Note that the example code in this section is for illustrative purposes only; it is not intended to be production-ready, and may not adequately handle all possible use cases or implement the exact behavior you want.

Remove fields from a search page

The agent console can be modified so that it displays only certain search fields. The following information explains how to remove fields from a search page.

Remove a field using the code view screen on the **Templates** tab.



You can any add any number of fields using the code view, but only the fields that are supported by the Search API will display results. For example, to prevent the order ID field from displaying on the screen, you could remove the with the ID "cc-search-criteria-id-field", then save the code and publish it.

Add fields to a search page

Use the Oracle CX Commerce REST web services APIs to add customer properties to your search pages. See Extending Oracle Commerce Cloud for information you need to know before using the services.

To add a property, for example, `myNewProperty`, to the screen you must make changes to the widget's JavaScript as well as the HTML template. For example, you can use the `searchOrders` endpoint to view a new property, `paymentType`, which you may want to add to the page to allow an agent to search for an order based on the type of payment used. The endpoints that you use for this example are:

```
/ccagent/v1/orders/serachOrders
/ccagent/v1/profiles/searchProfiles
/ccagent/v1/returnRequests/searchReturns
```

To make modifications to a widget, you must first download the widget. Then you make changes to the JavaScript and HTML template file. Once you have completed your changes, upload the widget. This upload process is described in detail in the Developing Widgets guide.

To modify the widget's JavaScript:

1. Identify the fields in the search API for the new field. For example, `paymentType`.
2. In the widget's JavaScript add a new observable in the `initOrderSearchCriteriaViewModel` method.

```
initOrderSearchCriteriaViewModel: function () {
   var self = this;
   self.accountNameSelected = ko.observableArray([]);
   self. paymentType = ko.observable('');
   self.orderId = ko.observable('');
   self.email = ko.observable('');
   self.firstName = ko.observable('');
   self.lastName = ko.observable('');
   self.account = ko.observable('');
   self.approver = ko.observable('');
   self.selectedSite = ko.observable('');
   self.selectedOrderState = ko.observable('');
   self.skuId = ko.observable('');
   self.productId = ko.observable('');
   self.timeValueForLastOrders = ko.observable('');
   self.timeUnitForLastOrders = ko.observable('');
   self.phone = ko.observable('');
   self.isEditPhone = ko.observable(false);
   self.isAdvancedSearch = ko.observable(false);
   self.timeUnits = [];
```

3. Clear the observable that was just added in the `resetBasicSearch` function.

```
/** * Resets the basic search fields.
*/
resetBasicSearch: function() {
this.orderId('');
this.email('');
this. paymentType ('');
this.firstName('');
this.lastName('');
this.accountNameSelected([]);
this.approver('');
```

```
this.selectedSite('');
this.selectedOrderState('');
this.timeValueForLastOrders('');
this.timeUnitForLastOrders(this.resources().days);
this.skuId('');
this.productId('');
this.phone('');
```

4. Modify the `getTextSearchCriteria` function to include the newly added property for text search queries.

```
/**
* Gets the Text search criteria.
*/
getTextSearchCriteria: function() {
  var self = this;
  var searchCriteria = self.getBasicSearchCriteria();
  searchCriteria[CCConstants.LIMIT] = this.orderSearchViewModel.
              itemsPerPage;
  searchCriteria[CCConstants.REQUIRE_COUNT] = false;
  return searchCriteria;
},
```

5. Modify the `getSCIMSearchCriteria` function to include the newly added property for SCIM search.

```
getSCIMSearchCriteria: function() {
  var self = this;
  var data = {};
  ...
  data.fields = "id,priceGroupId,siteId, paymentType,
    submittedDate,state,profile, priceInfo,
    payShippingInSecondaryCurrency,payTaxInSecondaryCurrency
    secondaryCurrencyCode,organization";
  return data;
},
```

6. Modify the widget's HTML template. To do this, add the required `<div>` in the template, this can be done in any form whose ID is `cc-search-form`.

Note that the search API also supports search by dynamic properties. To add a dynamic property to the search screen, such as `isVipMember`, ensure that you follow the naming convention for the search criteria given in the order search API documentation.

4

Create and Edit Orders Widgets

The widgets described in this section allow you to customize how an agent creates or edits an order. You may customize the following:

- Customer Details section
- Cart operations such as Add by SKU, Add by Product Name and Add from Catalog)
- Address book for both shipping and billing
- Shipping Method
- Payment Details. Note that if you have already customized payment properties, you should use the storefront payment widget to work with your customized properties. Refer to Appendix: Layout Widgets and Elements for information on storefront widgets.
- Notes
- Display and arrange custom order properties both internal and external
- Ability to make agent-specific operations such as price override.

Help a customer with their shopping cart

This widget displays the customer's shopping cart in the agent's context, allowing an agent to log on and assist a shopper.

You can use this widget to display and manage agent-specific functions, such as override prices, change quantities, customize items or edit and delete add-on products. For detailed information on working with a customer's shopping cart in the agent console, see Create new orders.

Widget Name: `agentShoppingCart`

Display Name: Shopping Cart - Agent

Supported Page Types: `agentCheckoutPageType`

Layouts:

- B2B Edit Layout - Agent
- B2B Checkout Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Layout - Agent

Elements: None

Search for and add items to a shopper's cart

This widget presents an interface that allows an agent to search for a product or SKU and add it to the shopper's cart.

The agent can perform the search by the product name or ID, or the SKU ID. For information on searching for and adding items to a cart in the agent console, see Create new orders.

Widget Name: `searchAndAddItemsToCart`

Display Name: Search And Add Items To Cart

Supported Page Types: `agentCheckoutPageType`

Layouts:

- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Layout - Agent

Elements: This widget contains the `sku-search` and `product-search` elements. It also contains the `add-from-catalog` widget-specific element. To understand how these elements are used, refer to the widget's code view.

Place an order for a shopper during checkout

This widget displays the place order or scheduled order button to the agent, during the checkout process, allowing the agent to place an order on behalf of the shopper.

For detailed information on working with a shopping cart in the agent console, see Create new orders.

Widget Name: `agentCheckoutPlaceOrderSummary`

Display Name: Place Order - Agent Checkout

Supported Page Types:

- `agentCheckoutPageType`
- `agentMultiShipCheckoutPageType`

Layouts:

- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Edit Layout For Multi Ship
- Checkout Layout - Agent
- Checkout Layout For Multi Ship
- Checkout Layout For Pending Payment - Agent

Elements: This widget uses the `dynamic-property` element. To understand how this element is used, refer to the widget's code view.

Work with an organization address book

This widget, which is used only in account-based environments, displays an interface that allows an agent to manage an account's address book.

For information on account address books, see Understand account addresses.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To migrate, you must modify the following `widget.template` and `.less` files:

- `/layouts/organizationAddressSelectorAgentLayout/widget.template`
- `/less/widget-organizationAddressSelectorAgentInst_`*`latest_version_number`*`.less`

To perform these upgrades, refer to the instructions in Work with widgets.

Widget Name: `organizationAddressSelector`

Display Name: Address book for B2B - Agent

Supported Page Types:

- `agentCheckoutPageType`
- `checkout`

Layouts:

- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent

Elements: This widget uses the `dynamic-property` element. To understand how this element is used, refer to the widget's code view.

View product details

This widget displays an interface that allows an agent to view the Product Detail page when an access point to the product detail page has been selected.

The agent-specific widget contains stock table and add-ons elements. For detailed information on viewing product details in the agent console, see View Orders.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `agentProductDetails`

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template file, which may not be available to the page layout. The `widget.template` instances that must be migrated are the following:

- `/layouts/addToPurchaseListLayout/widget.template`
- `/layouts/agentProductDetailsDefaultLayout/widget.template`
- `/layouts/productDetailsCPQChildeItemsLayout/widget.template`

To perform these upgrades, refer to the instructions in Work with widgets.

Display Name: Product Detail - Agent

Supported Page Types:

- `agentCheckoutPageType`
- `agentCreateExchangePageType`
- `agentOrderDetailsPageType`
- `agentPurchaseLsitPageType`
- `categoryPageType`
- `productPageType`
- `searchResultsPageType`

Layouts:

- Agent Collection Layout
- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Layout - Agent
- Create Exchange Request Layout - Agent
- Create Return Layout - Agent
- Process Returns Layout - Agent
- Product Layout - Agent
- Refunds Layout - Agent
- Search Results Layout - Agent
- View Return Request Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `add-to-purchase-list`
- `agent-product-details-add-item-to-purchase-list`
- `agent-product-details-product-addons`

- `agent-product-details-product-add-to-cart`
- `agent-product-details-product-backorder-message`
- `agent-product-details-product-configure`
- `agent-product-details-product-description`
- `agent-product-details-product-external-price`
- `agent-product-details-product-image`
- `agent-product-details-product-in-stock-message`
- `agent-product-details-product-list-price`
- `agent-product-details-product-long-description`
- `agent-product-details-product-out-of-stock-message`
- `agent-product-details-product-preorder-message`
- `agent-product-details-product-price-range`
- `agent-product-details-product-quantity`
- `agent-product-details-product-sale-price`
- `agent-product-details-product-shipping-surcharge`
- `agent-product-details-product-title`
- `agent-product-details-product-variants`
- `dynamic-property`

Review an order summary during checkout

This widget displays a summary of order details, shipping method and also allows an agent to place orders.

For information on reviewing an order summary, see View Orders.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To perform the migration, you must update the following `widget.template` and `.less` files:

- `/layouts/checkoutOrderSummaryAgentLayout/widget.template`
- `/less/widget-agentCheckoutOrderSummaryInst_`*`latest_version_number`*`.less`

To perform these upgrades, refer to the instructions in Work with widgets.

Widget Name: `checkoutOrderSummary`

Display Name: Checkout Order Summary - Agent

Supported Page Types: `agentCheckoutPageType`

This widget is also used in the Storefront user interface, and can be applied to storefront page types.

Layouts:

- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent
- Checkout Layout For Multi Ship - Agent

Elements: None.

Review order details during checkout

This widget displays specific information about the order, such as name, email, account, to the agent.

For information on reviewing order details in the agent console, see View Orders.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To migrate, you must update the following `widget.template` and `.less` files:

- `/layouts/orderDetailsDefaultLayout/widget.template`
- `/less/widget-agentAccountOrderDetailsInst_`*`latest_version_number`*`.less`

To make these changes, refer to the instructions in Work with widgets

Widget Name: `checkoutOrderDetails`

Display Name: Checkout Order Details

Supported Page Types: `agentCheckoutPageType`

Layouts:

- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent

- Checkout Layout For Multi Ship - Agent

Elements: None

Note: Quick links are not available when creating, editing or viewing order details. To enable quick links, you must create a custom property on this widget.

Review order details with pending payments

This widget displays specific information about an order that is pending payment, such as name, email, account, etc.

For information on reviewing order details in the agent console, see View Orders.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To migrate, you must update the following `widget.template` and `widget.less` files:

- `/layouts/pendingPaymentOrderDetailsLayout/widget.template`
- `/less/widget.less`

To make these changes, refer to the instructions in Work with widgets

Widget Name: `agentAccountOrderDetails`

Display Name: Order Details for Pending Payment - Agent

Supported Page Types: `agentCheckoutPageType`

Layouts:

- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent
- Checkout Layout For Multi Ship - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `agent-email-order-details`
- `agent-order-payment-details`
- `agent-order-price-details`
- `agent-order-refresh`
- `agent-order-summary`

- `agent-promotion-summary`
- `scheduled – order-actions`
- `scheduled-order-executionList`
- `scheduled-order-instruction`
- `shopping-cart-details`

Provide notes

This widget allows access to the agent-notes global element that enables an agent to view and add notes to the order or profile.

For detailed information on writing notes in the agent console, see View notes.

Widget Name: `agentNotes`

Display Name: Notes Widget - Agent

Supported Page Types:

- `agentCheckoutPageType`
- `agentMultiShipCheckoutPageType`
- `agentOrderDetailsPageType`

Layouts:

- Agent Checkout Layout - Pending Payment
- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent
- Checkout Layout For Multi Ship - Agent
- Order Details Layout - Agent

Elements: This widget uses the `agent-notes` element. To understand how this element is used, refer to the widget's code view.

Apply split payments

This widget provides a way for an agent to apply multiple payment types within an order.

It reflects applied payments, the amount due and any pending payments. The widget supports both single and split payment mode. For detailed information on searching for an order in the agent console, see Create new orders.

Note: Shoppers can pay for an order using either loyalty points or in a monetary currency or a mix of currencies. Loyalty points can be used if `allowAlternateCurrency` is enabled.

Widget Name: `agentSplitPayments`

Display Name: Split Payments - Agent

Supported Page Types:

- `agentCheckoutPageType`
- `agentMultiShipCheckoutPageType`

Layouts:

- Agent Checkout Layout - Pending Payment
- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent
- Checkout Layout For Multi Ship - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `agentsplitpayment-addpayment`
- `agentsplitpayment-cash`
- `agentsplitpayment-creditcard`
- `agentsplitpayment-giftcard`
- `agentsplitpayment-header`
- `agentsplitpayment-invoice`
- `agentsplitpayment-placeorder`
- `agentsplitpayment-storecredit`

Display shipping options

This widget displays various shipping options to the agent.

An agent can use the functionality of this widget to assist a customer in picking up an item in a store. The widget also displays to the agent the number of stores available. For detailed information on working with shipping in the agent console, see Create new orders. For working with Buy Online Pick Up In Store, see Configure Buy Online Pick Up In Store.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template file, which may not be available to the page layout.

To update, you must update the following `widget.template` file:

- `/layouts/multishipAddressBookAgentLayout/widget.template`

To perform this upgrade, refer to the instructions in Work with widgets.

Widget Name: `agentShippingOptions`

Display Name: Shipping Options - Agent

Supported Page Types: `agentAdvancedCheckoutPageType`

Layouts: Checkout Layout For Multi Ship - Agent

Elements: None

Work with promotions

This widget allows an agent to see which promotions have been applied to the order, as well as to apply a specific promotion.

For detailed information on working with promotions in the agent console, see Create new orders.

Note that a version of this widget exists within the Storefront framework, as well.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To do this, you must migrate the following `widget.template` and `.less` files:

- `/layouts/promotionAgentLayout/widget.template`
- `/less/widget-promotionAgentInst_`*`latest_version_number`*`.less`

To perform these upgrades, refer to the instructions in Work with widgets.

Widget Name: `promotions`

Display Name: Promotion Widget - Agent

Supported Page Types:

- `agentCheckoutPageType`
- `agentMultiShipCheckoutPageType`

Layouts:

- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent
- Checkout Layout For Multi Ship - Agent

Elements: This widget uses the `promotions-summary` global element. To understand how this element is used, refer to the widget's code view.

Review cart shipping details

This widget allows an agent to view shipping details from the cart.

For information on viewing shipping details in the agent console, see View shipping details.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration, as described in Work with widgets.

To migrate, you must update the following `widget.template` and `.less` files:

- `/layouts/agentShippingMethodsDropdownLayout/widget.template`
- `/less/widget-cartShippingDetailsAgentInst_`*`latest_version_number`*`.less`

Widget Name: `cartShippingDetails`

Display Name: Cart Shipping - Agent

Supported Page Types:

- `agentCheckoutPageType`
- `cart`

Layouts:

- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Layout - Agent

Elements: The Storefront version uses `cart-shipping-title`, `cart-shipping-address`, and `cart-shipping-options`. The Agent version uses only the `cart-shipping-options` element. To understand how these elements are used, refer to the widget's code view.

Manage a checkout address book

This widget displays a page that enables the agent to add a new address or to select an address that has been stored on a registered shopper's profile during the order process.

For information on managing the address book in the agent console, see Understand punchout orders.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be

available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To perform the migration, you must edit the following `template.widget` and `.less` files:

- `/layouts/checkoutAddressBookAgentLayout/widget.template`
- `/less/widget-agentCheckoutOrderSummaryInst_`*`latest_version_number`*`.less`

To perform these upgrades, refer to the instructions in Work with widgets.

Widget: `checkoutAddressBook`

Display Name: Checkout Address Book

Supported Page Types:

- `agentCheckoutPageType`
- `checkout`

Layouts:

- Checkout Edit Layout - Agent
- Checkout Layout - Agent

Elements: None

Review product purchase list information

This widget displays purchase list information for the product.

For information on working with purchase lists, refer to Enable Purchase Lists.

Widget: `agentProductDetails`

Display Name: Purchase List Product Details - Agent

Supported Page Types:

- `product`
- `category`
- `searchResults`
- `agentCheckout`
- `agentPurchaseList`
- `agentOrderDetails`
- `agentCreateExchange`

Layouts: Purchase List Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `agent-product-details-add-item-to-purchase-list`
- `agent-product-details-product-addons`

- agent-product-details-product-add-to-cart
- agent-product-details-product-add-to-space
- agent-product=details-product-back-link
- agent-product-details-product-backorder-message
- agent-product-details-product-configure
- agent-product-details-product-description
- agent-product-details-product-external-price
- agent-product-details-product-image
- agent-product-details-product-image-caroursel
- agent-product-details-product-in-stock-message
- agent-product-details-product-list-price
- agent-product-details-product-long-description
- agent-product-details-product-out-of-stock-message
- agent-product-details-product-preorder-message
- agent-product-details-product-price-range
- agent-product-details-product-quantity
- agent-product-details-product-sale-price
- agent-product-details-product-shipping-surcharge
- agent-product-details-product-stock-status-table
- agent-product-details-product-title
- agent-product-details-product-variants
- agent-product-details-text

Review shopping cart product information

This widget displays information to the agent on the current shopping cart.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget: `agentProductDetails`

Display Name: Purchase List Product Details - Agent

Supported Page Types:

- product
- category
- searchResults
- agentCheckout
- agentPurchaseList
- agentOrderDetails

- `agentCreateExchange`

Layouts:

- B2B Edit Layout – Agent
- B2B Layout – Agent
- Checkout Edit Layout - Agent
- Checkout Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `agent-product-details-add-item-to-purchase-list`
- `agent-product-details-line-break`
- `agent-product-details-product-addons`
- `agent-product-details-product-add-to-cart`
- `agent-product-details-product-add-to-space`
- `agent-product=details-product-back-link`
- `agent-product-details-product-backorder-message`
- `agent-product-details-product-configure`
- `agent-product-details-product-description`
- `agent-product-details-product-external-price`
- `agent-product-details-product-image`
- `agent-product-details-product-image-caroursel`
- `agent-product-details-product-in-stock-message`
- `agent-product-details-product-list-price`
- `agent-product-details-product-long-description`
- `agent-product-details-product-out-of-stock-message`
- `agent-product-details-product-preorder-message`
- `agent-product-details-product-price-range`
- `agent-product-details-product-quantity`
- `agent-product-details-product-sale-price`
- `agent-product-details-product-shipping-surcharge`
- `agent-product-details-product-stock-status-table`
- `agent-product-details-product-title`
- `agent-product-details-product-variants`
- `agent-product-details-text`

Define a scheduled order during checkout

This widget displays an interface that allows an agent to define a scheduled order based on the current order.

The agent can select a start or end date, the frequency of the order, and the ability to suspend the scheduled order. Note that this widget exists within the Storefront framework, as well. See Create Scheduled Orders for more information. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `checkoutScheduledOrder`

Display Name: Scheduled Order - Checkout

Supported Page Types: `agentCheckoutPageType`

Layouts:

- B2B Checkout Layout - Agent
- Checkout Layout - Agent
- Checkout Layout For Multi Ship - Agent

Elements: None

Display order information

This widget displays site, account and pricelist group selections to an agent.

This widget also provides backwards navigation. It contains the title, account and site dropdowns, as well as the links to navigate to all sections, etc. For information on creating orders in the agent console, see Create new orders.

If you have an account-based environment, available selectors include `site` and `account` selection. For standard storefront environments, selectors include `site` and `pricelist` group.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template file, which may not be available to the page layout.

To migrate, you must update the following `widget.template` instances:

- `/layouts/createOrderDefaultLayout/widget.template`
- `/layouts/editOrderLayout/widget.template`

To perform these upgrades, refer to the instructions in Work with widgets.

Widget Name: `createOrderHeader`

Display Name: Header - Agent Create Order

Supported Page Types:

- `agentCheckoutPageType`
- `agentMultiShipCheckoutPageType`

Layouts:

- B2B Checkout Layout - Agent

- B2B Edit Layout - Agent
- Checkout Edit Layout – Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent
- Checkout Layout For Multi Ship - Agent
- Checkout Layout For Pending Payment - Agent

Elements: The `header-text` element and `additional-custom-factors` are widget-specific elements. The `additional-custom-factors` elements allow you to create and display customizations. To understand how these elements are used, refer to the widget's code view:

- `account-selector`
- `additiona-custom-factors`
- `back-button`
- `header-text`
- `plg-selector`
- `site-selector`

Note: To provide header-level quick links, you must create a custom property on this widget.

Display an account's order information

This widget displays information on an account's order.

For information on creating orders in the agent console, see Create new orders.

In an account-based environment, available selectors include `site` and `account` selection.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `createOrderHeaderB2BLayout`

Display Name: Create Order Header B2B Layout

Supported Page Types:

- `agentCheckoutPageType`
- `agentMultiShipCheckoutPageType`

Layouts:

- B2B Edit Layout - Agent
- Checkout Edit Layout – Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout For Multi Ship - Agent
- Checkout Layout For Pending Payment - Agent

Elements: The `header-text` element and `back-button` are widget-specific elements. To understand how these elements are used, refer to the widget's code view.

Work with the order states and numbers

This widget displays an order's state and number.

It also supports **Cancel**, **Edit** and **Make Payment** buttons. For detailed information on working with order details in the agent console, see Create new orders.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template, which may not be available to the page layout. In this case, you will need to manually add it by copying and pasting the new widget's template code into the existing widget's template code. Additionally, this widget may not be available in the layout. You must update the widget's LESS file.

To perform these tasks, refer to the information in Work with widgets

Widget Name: `agentOrderDetailsHeader`

Display Name: Header - Agent Order Details

Supported Page Types: `agentOrderDetails` page

Layouts: Order Details Layout - Agent

Elements: None

View the catalog page

This widget displays the header for the catalog page.

When an agent clicks on the Complete Order button, they are taken to the agent checkout page, which allows them to complete an order on behalf of a shopper. For detailed information on viewing a catalog in the agent console, see Create new orders.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `agentCatalogHeader`

Display Name: Header - Agent Catalog

Supported Page Types:

- `category`
- `nosearchresults`
- `product`
- `searchresults`

Layouts:

- Collection Layout - Agent
- No Search Results Layout - Agent
- Product Layout - Agent

- Search Results Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `dropdown-minicart`

View collections

This widget displays navigation that allows an agent to view collections.

For information on collections, see Organize products in collections.

Widget Name: `megMenu`

Display Name: Collection Navigation Widget

Supported Page Types: All

Layouts:

- Product Layout – Agent
- No Search Results Layout – Agent
- Collection Layout – Agent
- Search Results Layout - Agent

Elements: None

Review loyalty payments

This widget displays information to an agent regarding the shopper's loyalty payments.

For detailed information on working with loyalty programs, see Work with Loyalty Programs.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `loyaltyPayment`

Display Name: Loyalty Payment

Supported Page Types:

- `agentCheckoutPageType`
- `agentMultiShipCheckoutPageType`
- `checkout`

Layouts:

- Agent Checkout Layout - Pending Payment
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent

- Checkout Layout For Multi Ship - Agent

Elements: This widget uses the `select-redeem-points` element. To understand how this element is used, refer to the widget's code view.

Obtain loyalty details

This widget allows an agent to see a shopper's loyalty information, such as information on points accumulated, if the points are available for spending or have already been used.

For detailed information on working with loyalty programs, see Work with Loyalty Programs.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `loyaltyDetails`

Display Name: Loyalty Details

Supported Page Types: All

Layouts:

- Agent Checkout Layout - Pending Payment
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent
- Checkout Layout For Multi Ship - Agent

Elements: This widget uses the `loyalty-details-basic` element. To understand how this element is used, refer to the widget's code view.

Configure CyberSource payment authorization

This widget provides functionality that allows an agent to enter data that is used during payment authentication.

For information on working with CyberSource authorization, see Configure Payment Processing.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `cyberSourcePaymentAuthorization`

Display Name: CyberSource Payment Authorization

Supported Page Types: `paymentPageType`

Layouts: Payer Authentication Layout

Elements: None

Customize a create order page

The following widgets, which are associated with the Checkout Layout - Agent, provide an interface that allows an agent to create an order.

- Navigation – Agent
- Notification Widget – Agent
- Header – Agent Create Order
- Additional Shopper Context
- Checkout Order Details
- Checkout Order Summary – Agent
- Search And Add Items To Cart
- Product Details – Agent
- Shopping Cart – Agent
- Shopping Cart Product Details – Agent
- Promotion Widget – Agent
- Scheduled Order – Agent Checkout
- Address book for B2C Customer – Agent
- Cart Shipping – Agent
- Loyalty Payment
- Loyalty Details
- Split Payments – Agent
- Notes Widget – Agent
- Request Quote Widget – Agent
- Place Order – Agent Checkout

You can customize this layout by adding or removing widgets. An example of how you could modify the default checkout layout is described in Configure layouts and widgets for in-store pick up.

Note: The list price is not displayed in the shopping cart price column. To display the list price, you must update the Shopping Cart - Agent widget.

Note that the following layouts also exist for an agent. These layouts contain a subset of the widgets listed above, as well as widgets specific to the layout's purpose:

- Checkout Edit Layout – Agent
- Checkout Layout For Pending Payment – Agent
- Checkout Layout For Multi Ship – Agent
- Checkout Edit Layout For Multi Ship - Agent

Note: To add quick links when creating, editing or viewing order details, you must create custom properties on the Header - Agent Create Order widget.

5

Work with Customer Information Widgets

This section describes the widgets that allow an agent to work with customer information.

There are a number of ways to work with customer information, you can access customer information from a customer profile, orders or purchase lists.

Work with customer profile details

This widget allows an agent to view in-depth details of the shopper.

The agent can use the interface provided by this widget to view the customer information, loyalty details, storefront roles, store credit, and email references. The agent can also use this interface to resend the password, create an order or view the cart link, as well as launch the store on behalf of the shopper.

You could customize this widget to display addresses in a different format, display the customer's order history, display orders pending approval or display scheduled orders.

For information on working with customer profiles in the agent console, see View order results.

Widget Name: `customerProfileDetails`

Display Name: Customer Profile Details - Agent

Supported Page Types: `agentProfilePageType`

Layouts: Customer Details Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `agent-notes`
- `customer-details`
- `customer-loyalty-programs`
- `customer-pending-orders`
- `customer-profile-reset-password`
- `customer-profile-save-cancel`
- `customer-profile-status`
- `customer-store-credit-balance`
- `customer-store-roles`
- `dynamic-property`
- `launch-store-as-customer`

Note: The Customer Profile page does not have a back navigation button by default. To create one, you must update the widget with a custom property.

Display saved credit cards

You can modify the Customer Profile Details widget to display saved credit cards on the profile details page. This allows a customer to save their credit card information and reuse it whenever they want. The following is an overview of the steps you would take to display stored credit cards.

For detailed information on working with and creating saved credit cards widgets, see Support stored credit cards.

1. Use the REST API to initiate the `listCreditCards` endpoint for a shopper profile.
2. This returns the customer's credit card information. Each item in the results displays card information. Save the results.
3. Create a new instance of the `CreditCard` view model.
4. Use the `populateData` method to add the information returned in the response.
5. 5. The shopper's card information is stored in the `allCreditCards` observable array, in the Customer Profile Details widget.
6. Modify the Customer Profile Details widget to display the card details in the UI.

View a customer summary

This widget is used in the profile page to show a summary of customers' basic information along with site selector and account selector options.

For information on working with customer profiles in the agent console, see Search for customers.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `customerSummary`

Display Name: Customer Summary Widget

Supported Page Types: `agentProfilePageType`

Layouts:

- Account Contacts Layout - Agent
- Account Details Layout - Agent
- Address Book Layout - Agent
- Customer Details Layout - Agent
- Order History Layout - Agent
- Orders Pending Approval Layout - Agent
- Purchase List Layout - Agent
- Scheduled Order Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `account-selector`
- `customer-basic-information`
- `site-selector`

Note: The Customer Profile page does not have a back navigation button by default. To create one, you must update the widget with a custom property.

Review order history

This widget can display a shopper's order history.

For information on reviewing order histories in the agent console, see Search for customers.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To migrate, you must update the following `widget.template` and `.less` files:

- `/layouts/agentOrderHistoryLayout/widget.template`
- `/less/widget-orderHistoryWidgetInst_agent_`*`latest_version_number`*`.less`

To perform these upgrades, refer to the instructions in Work with widgets.

Widget Name: `orderHistoryWidget`

Display Name: Order History Widget - Agent

Supported Page Types:

- `agentOrderHistoryPageType`
- `agentProfilePageType`
- `orderHistoryPageType`
- `profilePageType`

Layouts: Order History Layout - Agent

Elements: None

Note: To filter order histories by Order ID, you must create a custom property for this widget.

View purchase lists

This widget displays a list of purchase lists.

For information on working with purchase lists and customer profiles in the agent console, see Search for customers.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To migrate, you must update the following `widget.template` and `.less` files:

- `/layouts/agentPurchaseListsLayout/widget.template`
- `/less/widget-purchaseListsDetailsWidgetAgent_`*`latest_version_number`*`.less`

To perform these upgrades, refer to the instructions in Work with widgets.

Note: An agent can no longer indicate that the purchase list is applicable for all sites when working in the administrative interface. To allow a purchase list to be applicable for all sites, you must create a customized widget, similar to the widget used in the storefront application.

Widget Name: `purchaseLists`

Display Name: Purchase List

Supported Page Types: All

Layouts: Purchase List Layout - Agent

Elements: None

Provide additional shopper context

This widget displays shopper context information to an agent.

For information on shopper context in the agent console, see Understand external catalog and price group assignment.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `additionalShopperContext`

Display Name: Additional Shopper Context

Supported Page Types: `agentCheckoutPageType`

Layouts:

- B2B Checkout Layout - Agent
- Checkout Layout - Agent

Elements: None

Display address for account-based contacts

This widget displays customer address information.

It also allows an agent to select a site or account if there are multiple options. For information on working with customer addresses, see Understand account addresses.

Widget Name: `checkoutAddressBook`

Display Name: Address Book for B2C Customer - Agent

Supported Page Types:

- `Checkout`
- `agentCheckout`

Layouts:

- Checkout Edit Layout – Agent
- Checkout Layout - Agent

Elements: This widget uses the `dynamic-property` element. To understand how this element is used, refer to the widget's code view.

Work with tab navigation

This widget provides tab navigation on the profile pages.

For information working with tab navigation, see Navigate the Agent Console.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `secondaryNavigation`

Display Name: Secondary Navigation

Supported Page Types: To understand how these elements are used, refer to the widget's code view:

- `agentAccountContactsPageType`
- `agentAccountDetailsPageType`
- `agentAddressBookPageType`
- `agentCustomerDetailsPageType`
- `agentCustomerSearchPageType`
- `agentOrderHistoryPageType`
- `agentOrdersPendingApprovalPageType`
- `agentPurchaseListPageType`
- `agentScheduledOrderPageType`
- `agentSelfRegistrationPagePageType`

Layouts:

- Account Contacts Layout - Agent
- Account Details Layout - Agent
- Address Book Layout - Agent

- Customer Details Layout - Agent
- Order History Layout - Agent
- Orders Pending Approval Layout - Agent
- Purchase List Layout - Agent
- Scheduled Order Layout - Agent

Elements: None

Migrating widgets

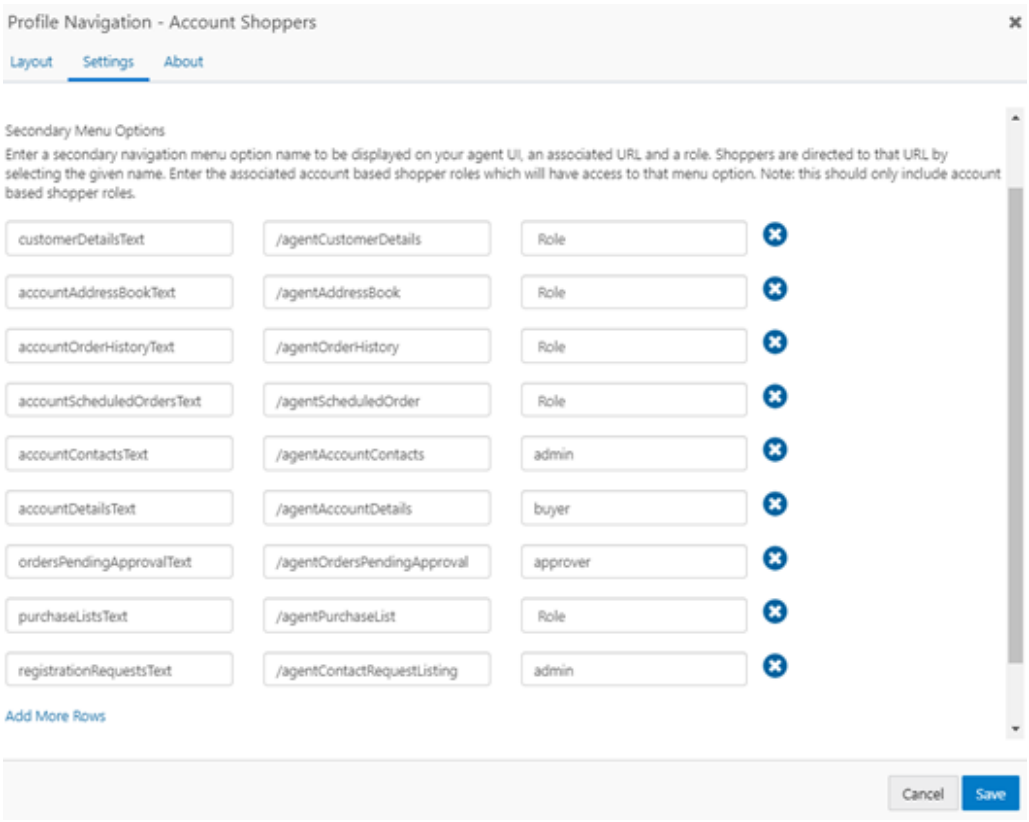
If you have recently migrated and this widget may not be visible in your widget list. In this case, you will need to manually add it by copying and pasting the new widget's template code into the existing widget's template code.

To do this, refer to the instructions in Work with widgets.

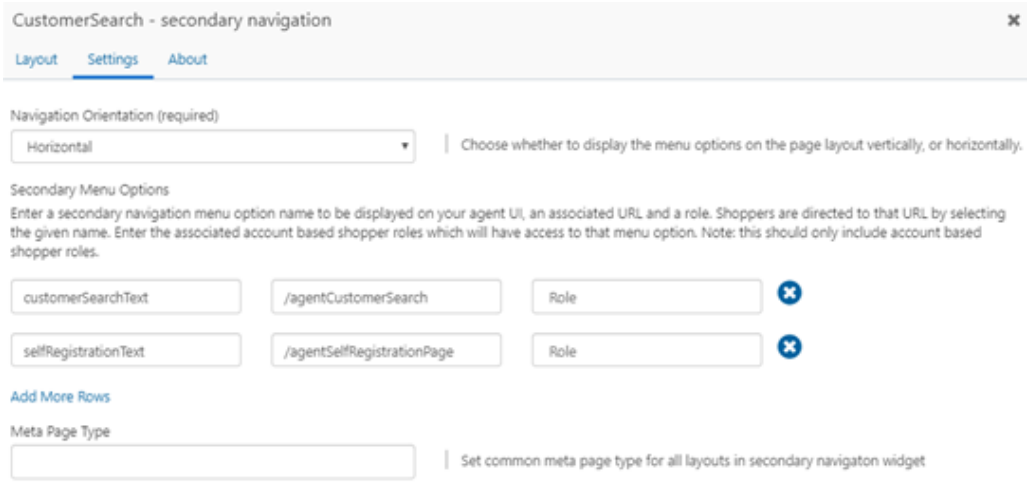
The `widget.template` must be migrated for each instance of the widget:

- `/layouts/agentSecondaryNavigationLayout/widget.template`

When migrating the template code for the widget, ensure that the following settings are enabled in the Profile Navigation-Account Shoppers instance:



When migrating the template code for the Customer Search - Secondary Navigation instance, ensure that the following settles are enabled:



6

Customize Self-Registration Widgets

This section describes widgets that can be customized for self-registration. These include widgets that allow you to customize registration details and configure notification or navigation.

These include widgets that allow you to customize registration details and configure notification or navigation.

Search registration requests

This widget presents a page that allows an agent to search for account and contact registration requests.

For information on searching for registration requests with the agent console, see Search for customers.

Widget Name: `registrationRequestSearch`

Display Name: Registration Request Search

Supported Page Types: `agentSelfRegistrationPagePageType`

Layouts: Self Registration Layout - Agent

Elements: None

View self registration details

This widget enables the agent to view read-only data for accounts or contacts self registration requests.

For information on viewing registration requests with the agent console, see Search for customers.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `selfRegistrationDetail`

Display Name: Self Registration Detail

Supported Page Types: `agentRegistrationRequestDetailPageType`

Layouts: Registration Request Detail Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `dynamic-property`
- `back-button`

Note: Backward navigation is not available on the view self registration details page. To create a back button, create a custom property on this widget.

Configure notifications

This widget provides an interface that allows an agent to view notifications when an error or a success message occurs.

For information on agent notifications, see Place orders.

Widget Name: `notifications`

Display Name: Notification Widget - Agent

Page Types: All

Layouts:

- Account Contacts Layout - Agent
- Account Details Layout - Agent
- Address Book Layout - Agent
- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout - Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent
- Checkout Layout For Multi Ship - Agent
- Collection Layout - Agent
- Create Exchange Request Layout - Agent
- Create Return Layout - Agent
- Customer Details Layout - Agent
- Customer Search Layout - Agent
- No Search Results Layout - Agent
- Order Details Layout - Agent
- Order History Layout - Agent
- Orders Pending Approval Layout - Agent
- Process Returns Layout - Agent
- Product Layout - Agent
- Purchase List Layout - Agent
- Refund Layout - Agent
- Registration Request Detail Layout - Agent
- Scheduled Order Layout -Agent
- Search Results Layout - Agent
- View Return Request Layout - Agent

Elements: This widget uses the `notification-message-box` element. To understand how this element is used, refer to the widget's code view.

Review and create customer registration

This widget provides an interface that allows an agent to create and register a new contact or account.

For information on account registration, see Understand Accounts.

Widget Name: `agentCustomerRegistration`

Display Name: Customer Registration - Agent

Page Types: `agentCustomerSearchPageType`

Layouts: Customer Search Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `dynamic-property`
- `site-selector`

View contact registration

This widget displays a list of pending contact registrations requests.

For information on account registration, see Understand Accounts.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `agentCustomerRegistration`

Display Name: Customer Registration - Agent

Supported Page Types:

- `agentContactRequestsPageType`
- `agentCustomerSearchPageType`

Layouts: Contact Registration Listing

Elements: None

Customize navigation for customer search and self registration

This widget displays the tab navigation in the Customer Search page.

For information on customizing search and self registration, see Configure customer and order search.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `secondaryNavigation`

Display Name: Secondary Navigation

Supported Page Types:

- `agentCustomerSearchPageType`
- `agentSelfRegistrationPagePageType`

Layouts:

- Customer Search Layout - Agent
- Self Registration Layout - Agent

Elements: None

Note: There is no paging available for addresses displayed in the Address tab. To enable paging, you must create a custom property.

7

Configure Returns and Exchanges Widgets

The following widgets allow you to customize how an agent works with return requests and exchanges.

These widgets allow you to configure how returns and exchanges are processed or created. These widgets also allow you to view history information and customize refunds.

Work with returns

The following widgets can be used by an agent to view, create and process returns.

Create return requests

This widget displays an interface used by an agent to initiate a return. The agent can make or edit comments for each return item.

For information on creating return requests in the agent console, see Process returns.

Widget Name: `agentCreateReturn`

Display Name: Create Return - Agent

Supported Page Types: `agentCreateReturnPageType`

Layouts: Create Return Layout - Agent

Elements: This widget uses the `back-button` and `custom-properties` global elements. To understand how these elements are used, refer to the widget's code view.

Process returns

This widget displays an interface used by an agent to acknowledge the receipt of items that have been shipped back to the warehouse. This interface also allows the agent to update the system with the quantity received against a return request or a reason for the return. The agent can make or edit comments for each return item.

For information on processing return requests in the agent console, see Process returns.

Widget Name: `agentProcessReturns`

Display Name: Process Returns - Agent

Supported Page Types: `agentProcessReturnsPageType`

Layouts: Process Returns Layout - Agent

Elements: This widget uses the `back-button` and custom-properties global elements. To understand how these elements are used, refer to the widget's code view.

View return history details

This widget displays return history for the order. It also displays return-specific details, as well as the ability for the agent to take actions such as receive the return or refund payment.

For information on viewing return histories in the agent console, see Process returns.

Widget Name: `agentReturnHistoryDetails`

Display Name: Return History - Agent

Supported Page Types: `agentOrderDetails` page

Layouts: Order Details Layout - Agent

Elements: None

View a return request

This widget displays a return request to an agent once the return has been processed or submitted, and the items are marked as complete.

For information on viewing return requests in the agent console, see Process returns.

Widget Name: `agentViewReturnRequest`

Display Name: View Return Request - Agent

Supported Page Types: `agentViewReturnRequestPageType`

Layouts: View Return Request Layout - Agent

Elements: This widget uses the `back-button` and `custom-properties` global elements. To understand how these elements are used, refer to the widget's code view.

Work with exchanges

The following widgets allow an agent to create and review exchanges.

Create an exchange

This widget provides an interface that can be used by an agent to initiate or create an exchange request. This interface also allows agents to select the items and quantities to be exchanged. Additionally, the agent may add a reason for the exchange. Once the exchange has been submit, a new exchange order is created.

For information on creating an exchange in the agent console, see Exchange products.

Widget Name: `agentCreateExchange`

Display Name: Create Exchange - Agent

Supported Page Types: `agentCreateExchangePageType`

Layouts: Create Exchange Layout - Agent

Elements: This widget uses the `back-button` and `custom-properties` global elements. To understand how these elements are used, refer to the widget's code view.

View exchange history details

This widget displays exchange history for the order. It also displays exchange-specific details, as well as a way for the agent to take actions, such as receive or process the exchange.

For information on creating an exchange in the agent console, see Exchange products.

Widget Name: `agentExchangeHistoryDetails`

Display Name: Exchange History - Agent

Supported Page Types: `agentOrderDetails` page

Layouts: Order Details Layout - Agent

Elements: None

Adjust refunds

This widget displays an interface that allows an agent to initiate the refund process. The agent may adjust the system-generated refund before processing the refund.

For information on working with refunds in the agent console, see Exchange products.

Widget Name: `agentRefunds`

Display Name: Refunds - Agent

Supported Page Types: `agentRefundsPageType`

Layouts: Refunds Layout - Agent

Elements: This widget uses the `back-button` and `custom-properties` global elements. To understand how these elements are used, refer to the widget's code view.

8

Use Account-Based Widgets

You can create widgets for account-based environments, that work specifically for business-to-business customers.

The following widgets are used with account-based contacts and organizations. For information on account-based environments, refer to the Manage account-based commerce accounts and roles section of the Using Oracle CX Commerce Agent Console guide.

View account customer carts

This widget displays a page that allows an agent to view customer carts.

This widget also displays the search results in a table, and allows the agent to access different pages, such as clicking on an order ID that directs the agent to the order details page. For detailed information on viewing customer carts in the agent console, see Search for customers.

Widget Name: `agentCustomerCartsDialog`

Display Name: Customer Carts Dialog - Agent

Supported Page Types:

- `agentCheckoutPageType`
- `agentCustomerSearchPageType`

Layouts: Customer Search Layout-Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `account-selector`
- `additional-shopper-context`
- `site-selector`

View account order details

This widget displays the contents of the order, including order items, shipping details payment details and other items.

You could customize this widget by adding widgets for loyalty information, price overrides and split shipping. For information on account contacts, see Manage account-based commerce accounts and roles.

Note that this widget can be used in both account-based and storefront environments.

Widget Name: `agentAcountOrderDetails`

Display Name: Order Details - Agent

Supported Page Types: `agentOrderDetails` page

Layouts: Order Details Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `add-to-purchase-list`
- `agent-email-order-details`
- `agent-order-payment-details`
- `agent-order-price-details`
- `agent-order-refresh`
- `agent-order-summary`
- `agent-promotion-summary`
- `copy-order`
- `dynamic-property`
- `return-order`
- `scheduled-order-actions`
- `scheduled-order-executionList`
- `scheduled-order-instruction`
- `shopping-cart-details`

Add an image to an account detail page example

The following is an example of a customization that you could make to the account detail page. This example shows you how to add an image of a company logo to the account's page.

Note that the example code in this section is for illustrative purposes only; it is not intended to be production-ready, and may not adequately handle all possible use cases or implement the exact behavior you want.

1. Download the `agentAccountOrderDetails` element.
2. Copy the element to change. For example, the `agent-order-summary` element.
3. Edit the template code. Access the `$parent.user().organizations` array can access the `organizationLogo`.
   Note: When editing this code, ensure that you have a check for an array index. The `organizationLogo` is returned in response to the `getOrganization` endpoint. Organization details are stored in the user view model by default, which can be accessed from the widget using `widget.user()`.
4. Upload the element as an extension.
5. Drag and drop the new element and remove the existing `agent-order-summary` element from the **Design** page.
6. Publish the changes.

Display shipping tracking information

The following is an example of a customization that you could make to the account detail page. This example shows you how to display shipping tracking information for each line item of the order, or for each shipping group.

1. Download the `agentAccountOrderDetails` element.
2. Copy the `shopping-carts-details` element.
3. Edit the template code where `trackingInfo` is read from `$data.trackingInfo`.
4. Upload the edited element as an extension.
5. Drag and drop the new element and remove the existing `shopping-carts-details` element from the design studio.
6. Publish the changes.

View account details

This widget allows the agent to view account details of an account-based contact.

This is an account-based-specific widget that allows an agent to see static and dynamic properties of an account. For information on account contacts, see Manage account-based commerce accounts and roles.

Widget Name: `agentAccountDetails`

Display Name: Account Details - Agent

Supported Page Types: `agentProfilePageType`

Layouts: Account Details Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `account-approval-setting`
- `account-general-info`
- `dynamic-property`
- `registration-request-details`

Manage account contacts

This widget allows the agent to view a list of account contacts.

It allows an agent to view, add, remove or modify account contacts. This is an account-based-specific widget. For information on account contacts, see Manage account-based commerce accounts and roles.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be

available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To migrate, you must update the following `widget.template` and `.less` files:

- `/layouts/delegatedAdminContactsAgentLayout/widget.template`
- `/less/widget-agentAccountContactsInst_`*`latest_version_number`*`.less`

To perform these upgrades, refer to the instructions in Work with widgets.

Widget Name: `delegated-admin-contacts`

Display Name: Account Contacts Widget - Agent

Supported Page Types:

- `agentAccountContactsPageType`
- `agentProfilePageType`
- `profilePageType`

Layouts: Account Contacts Layout - Agent

Elements: None.

Display contact information

This widget displays an interface that allows an agent to navigate through a contact's information.

For information on accounts, see Understand Accounts.

Widget Name: `secondaryNavigation`

Display Name: Profile Navigation – Account Shoppers

Supported Page Types: All

Layouts:

- Account Contacts Layout – Agent
- Account Details Layout – Agent
- Address Book Layout – Agent
- Customer Details Layout – Agent
- Order History Layout – Agent
- Orders Pending Approval Layout – Agent
- Purchase List Layout – Agent
- Scheduled Order Layout - Agent

Elements: None

Manage account addresses

This widget displays an interface where an agent can manage an account contact's address book.

This interface allows the agent to add, view, delete or update addresses. It can also be used to view, update, delete or manage the shopper's addresses. For information on account address books, see Understand account addresses.

Note that a version of this widget exists within the Storefront framework, as well.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration..

To migrate, update the following `widget.template` and `.less` files:

- `/layouts/agentAccountAddressesLayout/widget.template`
- `/less/widget-accountAddressesDetailsWidgetAgent_`*`latest_version_number`*`.less`

To perform these tasks, refer to the instructions in Work with widgets

Widget Name: `accountAddresses`

Display Name: Account Address Details - Agent

Supported Page Types:

- `agentProfilePageType`
- `profilePageType`

Layouts: Address Book Layout - Agent

Assign a Delegated Administrator

This widget provides a display that allows the agent to perform administration on an account similar to that of a Delegated Administrator.

For information on the Delegated Administrator feature, see Search for customers.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `delegatedAdminContacts`

Display Name: Account Contacts

Supported Page Types:

- `agentProfilePageType`
- `profilePageType`

Layouts: Account Contacts Layout - Agent

Elements: None.

View pending contact registration requests

This widget allows an agent to see a list of pending contact registrations requests.

For information on working with scheduled orders, see Search for customers.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `agentContactRegistrationRequests`

Display Name: Contact Registration Requests

Supported Page Types: `agentContactRequestsPageType`

Layouts: Registration Request Listing

Elements: None

Work with scheduled orders

The following widgets provide an interface that allows an agent to view and create scheduled orders for an account.

View a scheduled order list

This widget displays a list of scheduled orders. For information on working with scheduled orders, see Understand punchout orders.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To migrate, you must update the following `widget.template` and `.less` files:

- `/layouts/agentScheduledOrderListLayout/widget.template`
- `/less/widget-scheduledOrderListInst_agent_`*`latest_version_number`*`.less`

To perform these upgrades, refer to the instructions in Work with widgets.

Widget Name: `scheduledOrderList`

Display Name: Scheduled Order Listing Widget - Agent

Supported Page Types:

- `agentScheduledOrderPageType`
- `profilePageType`

Layouts:

- Scheduled Order Layout – Agent

Elements: None

Create a scheduled order

This widget displays an interface that allows an agent to create scheduled orders. For information on working with scheduled orders, see Understand punchout orders.

Widget Name: `checkoutScheduledOrder`

Display Name: Scheduled Order – Agent Checkout

Supported Page Types:

- `agentCheckout`
- `agentMultiShipCheckout`
- `checkout`

Layouts:

- B2B Checkout Layout – Agent
- Checkout Layout – Agent
- Checkout Layout For Multi Ship - Agent
- Scheduled Orders

Elements: None

Work with quotes

Agents can request quotes for account-based customers. Customizing quote widgets allows you to configure this process.

Create a request for a quote

This widget creates an interface that allows the agent to create either an order or a request for a quote. This interface also enables an agent to disable payment, and hides the place order button and displays the “request for quote” text. For information on working with quotes, see Understand the Oracle CPQ integration.

Migrate widgets

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

You must migrate the following `widget.template` file and `.less` files:

- `/layouts/requestQuoteWidgetAgentLayout/widget.template`
- `/less/widget-requestQuoteInst_`*`latest_version_number`*`.less`

To perform these upgrades, refer to the instructions in Work with widgets.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `requestQuote`

Display Name: Request Quote - Agent

Supported Page Types: All

Layouts:

- B2B Checkout Layout - Agent
- B2B Edit Layout - Agent
- Checkout Edit Layout For Multi Ship - Agent
- Checkout Layout - Agent
- Checkout Layout For Multi Ship - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `quote-notes-history`
- `requester-notes-text-area`
- `request-quote-button`

View quote details

This widget provides an interface that allows the agent to review a quoted order, accept or reject quotes and to re-request quotes. For information on working with quotes, see Understand the Oracle CPQ integration.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `agentQuoteOrderDetails`

Display Name: Quote Order Details - Agent

Supported Page Types: `agentOrderDetails` page

Layouts: Order Details Layout - Agent

Elements: To understand how these elements are used, refer to the widget's code view:

- `agent-accept-quote-button`
- `agent-add-note-button`
- `agent-notes-history-text-area`
- `agent-quote-note-text-area`
- `agent-reject-quote-button`
- `agent-request-requote-button`

Work with approvals

The following widgets can be used by an agent with the correct role to access layouts that allow them to approve or review pending approvals.

Approve an order

This widget provides an interface that can be used by an agent with the correct role to approve an account-based order in the Pending Approval state. This interface also shows, and allows the agent to add to, the approval comments associated with an order.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

Widget Name: `agentOrderApproval`

Display Name: Order Approval - Agent

Supported Page Types: `agentOrderDetails`

Layouts: Order Details Layout - Agent

Elements: None

View orders pending approval

Agents can view orders that are pending approval if they are members of the Approval role. This widget lists the pending orders and approval reasons, and allows the agent to approve or reject the request.

For additional information on approvals, refer to Understand account-based commerce approvals.

Note that a version of this widget exists within the Storefront framework, as well. See Appendix: Layout Widgets and Elements for information on Storefront widgets.

If you have recently migrated and this widget may be visible in your widget list. However, the widget is using the default template and .less file, which may not be available to the page layout. Additionally, this widget's style format must be updated so that it appears in the layout after a migration.

To migrate, you must update the following `widget.template` and `.less` files:

- `/layouts/agentOrderPendingApprovalsLayout/widget.template`
- `/less/widget-orderPendingApprovalDetailsWidgetAgent_`*`latest_version_number`*`.less`

To do this refer to the instructions in Work with widgets.

Note: The agent can no longer select the Check for approval button from the administrative interface when creating or editing an order. To implement this functionality, use the REST API.

Widget Name: `ordersPendingApproval`

Display Name: Orders Pending Approval

Supported Page Types: `agentOrdersPendingApproval`

Layouts: Orders Pending Approval Layout - Agent

Elements: None

9

Understand Agent-Based Layouts

This section describes agent-based layouts. Layouts are what comprise the pages that are displayed to the agent.

Each individual layout represents a various page, and act as a template for the agent's console. Additionally, layouts contain the widgets that allow you to define the page structure. Layouts can contain more than one widget, with each widget containing specific functionality. These widgets combined to for the page layout.

Using the Design page, you can view and customize layouts to suit your environment. When you click the design page, the default Layout page is displayed, which contains information on each layout used by the agent console. You can drag components from the component pane onto layouts, customizing each page.

Important: While you can work with and create customized layouts, the Preview function does not work with agent-based layouts.

Use the layout's configuration to customize the agent console, and extend the functionality of agent-specific widgets.

For detailed information on working with layouts, see Design Your Store Layout.

Agent-specific page layouts

The following layouts have been created for agent console widgets:

Layout	Associated Widgets
Account Contacts Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Customer Summary Widget, Profile Navigation – Account Shoppers, Account Contacts Widget – Agent
Account Details Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Customer Summary Widget, Profile Navigation – Account Shoppers, Account Details – Agent
Address Book Layout -Agent	Navigation – Agent, Notifications Widget – Agent, Customer Summary Widget, Profile Navigation – Account Shoppers, Account Address Details – Agent
Checkout Edit Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Create Order Header B2B Layout, Checkout Order Details, Checkout Order Summary – Agent, Search And Add Items To Cart, Shopping Cart – Agent, Promotions Widget – Agent, Address Book for B2C Customer – Agent, Cart Shipping – Agent, Loyalty Payment, Loyalty Details, Split Payments – Agent, Notes Widget – Agent, Place Order – Agent Checkout

Layout	Associated Widgets
Checkout Edit Layout For Multi Ship - Agent	Navigation – Agent, Notifications Widget – Agent, Create Order Header B2B Layout, Checkout Order Details, Checkout Order Summary – Agent, Shipping Options - Agent, Promotion Widget – Agent, Loyalty Payment Loyalty Details, Split Payments – Agent, Notes Widget – Agent, Request Quote Widget – Agent, Place Order – Agent Checkout
Checkout Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Header – Agent Create Order, Checkout Order Details, Checkout Order Summary – Agent, Search And Add Items To Cart, Shopping Cart – Agent, Promotion Widget – Agent, Scheduled Order – Agent Checkout, Address Book for B2C Customer – Agent, Cart Shipping – Agent, Loyalty Payment, Loyalty Details, Split Payments – Agent, Notes Widget – Agent, Request Quote Widget – Agent, Place Order – Agent Checkout
Checkout Layout For Multi Ship - Agent	Navigation – Agent, Notifications Widget – Agent, Create Order Header B2B Layout, Checkout Order Details, Checkout Order Summary – Agent, Shipping Options – Agent, Promotion Widget – Agent, Scheduled Order – Agent Checkout, Loyalty Payment, Loyalty Details, Split Payments – Agent, Notes Widget – Agent, Request Quote Widget – Agent, Place Order – Agent Checkout
Checkout Layout for Pending Payment - Agent	Navigation – Agent, Create Order Header B2B Layout, Order Details for Pending Payment – Agent, Loyalty Payment, Loyalty Details, Split Payments – Agent, Notes Widget – Agent, Place Order – Agent Checkout
Collection Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Overlayed Guided Navigation, Header – Agent Catalog, Collection Navigation Widget, Product Listing Widget - Agent
Create Exchange Request Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Create Exchange – Agent
Create Return Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Create Return – Agent
Customer Details Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Customer Summary Widget, Profile Navigation – Account Shoppers, Customer Profile Details – Agent
Customer Search Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Customer Search and Self Registration – secondary Navigation, Customer Registration – Agent, Customer Search - Agent
Home Layout - Agent	Navigation – Agent, Dashboard - Agent

Layout	Associated Widgets
No Search Results Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Overlayed Guided Navigation, Header – Agent Catalog, Collection Navigation Widget, No Search Results - Agent
Order Details Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Header – Agent Order Details, Order Details – Agent, Return History – Agent, Exchange History – Agent, Order Approval – Agent, Quote Order Details – Agent, Notes Widget - Agent
Order History Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Customer Summary Widget, Profile Navigation – Account Shoppers, Order History Widget - Agent
Order Search Layout - Agent	Navigation – Agent, Order Search - Agent
Orders Pending Approval Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Customer Summary Widget, Profile Navigation – Account Shoppers, Orders Pending Approval Details – Agent
Payer Authentication Layout	CyberSource Payment Authorization
Process Returns Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Process Returns - Agent
Product Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Header – Agent Catalog, Collection Navigation Widget, Product Details - Agent
Purchase List Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Customer Summary Widget, Profile Navigation – Account Shoppers, Purchase List
Refunds Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Refunds - Agent
Registration Request Detail Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Self Registration Detail
Return Search Layout - Agent	Navigation – Agent, Return Search - Agent
Scheduled Order Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Customer Summary Widget, Profile Navigation – Account Shoppers, Scheduled Orders Listing Widget –Agent
Search Results Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Overlayed Guided Navigation, Header – Agent Catalog, Collection Navigation Widget, Search Results Widget - Agent
Self Registration Layout - Agent	Navigation – Agent, Notifications Widget – Agent, Customer Search and Self Registration – secondary navigation, Registration Request Search
View Return Request Layout - Agent	Navigation – Agent, Notifications Widget – Agent, View Return Request – Agent

Create a new layout

You can create a new layout by cloning an existing layout and them making modifications to the copy. This allows you to configure things like site settings, notes, viewports and other information.

1. After accessing the **Design** page, click the **Layouts** tab.
2. Highlight the layout that you want to clone and click the **Clone Layout** option from the toolbar.
3. Enter the new layout's name and other information.
4. Click **Save** to confirm all of the new settings.

Once you have created a new layout, you can add widgets or other components to it as necessary. For detailed information on creating and customizing new layouts, see Design Your Store Layout.

Note: When creating a new layout, make sure that the roles are set correctly. To begin using the new copy, you must delete the roles from the old copy of the layout and add them to the new layout.

10

Use Agent Themes

You can use an agent-specific theme when creating your application.

A theme is made up of a number of LESS files that are then compiled into a set of CSS files. These themes contain pre-configured settings that assist you in creating the look and feel of your site. You can have a number of different themes for your sites. For each active theme, there is a set of CSS files.

Note: If the storefront CSS files have been updated or added to a theme, ensure that you have also update the agent files have also been updated.

Your agent application uses its own specific agent theme. Because there can only be one agent theme, a configuration property lets you specify the theme ID to use for the agent application.

The agent theme is listed on the Theme tab of the Theme Manager page. However, note the following differences from the storefront themes:

- There can be only one agent theme.
- Although they are displayed, the agent theme cannot be edited using the Theme Manager page in the administration interface.
- Agent themes cannot use the Go to Theme Code button in the administration interface.
- Agent themes are not listed in the dropdown for themes in Site Settings in the administration interface.

The agent theme is stored in the `AdminPageRepoistory` and uses the endpoints listed below. Set the `AgentTheme` flag in the repository to `true` to use an agent theme.

For information on working with storefront themes, see Customize Your Store's Design Theme.

Work with themes

The Agent REST API contains theme-based endpoints that allow you to perform various functions.

For detailed information on these endpoints, refer to the Agent REST API documentation.

Get all themes

Stores can use many themes; however, the agent console can only use a single theme. You can review all of the themes available and make modifications as necessary. To see all themes, use the Themes API:

Issue a `GET` command to get all themes, including the agent theme:

```
/ccadmin/v1/themes?includeagenttheme=true
```

Note: If the `includeagenttheme` query parameter is set to `false` or omitted, the endpoint will not return the agent theme.

Get the active agent theme

An active theme is a theme that is currently in use. You can issue a `GET` command to get the active agent theme:

```
/ccadmin/v1/themes/agentThemeDetails
```

The response may be something similar to this:


```
{
  "isAgentTheme": true,
  "thumbnail": "",
  "theme_additional_fonts": {},
  "notes": "This is the Agent Theme.",
  "is_active": false,
  "theme_styles_color": {
    "@ButtonPrimarySize": "medium",
    "@ButtonSecondaryTextDecoration": "none",
    "@ButtonPrimaryTextColor": "#ffffff",
    "@SubNavigationTextColor": "#6F7178",
    "@ButtonPrimaryUseGradient": "false",
    "@ButtonPrimaryFontStyle": "normal",
    "@ButtonSecondaryFontFamily": "@sansFontFamily",
    "@ButtonPrimaryFontWeight": "normal",
    "@NavbarLinkColor": "#3D3D3D",
    "@NavbarLinkHoverColor": "#195D8D",
    "@NavbarTextColor": "#333333",
    "@mobileNavBarButtonColor": "#333333",
    "@SubNavigationLinkHoverColor": "#195D8D",
    "@NavbarBackgroundHoverColor": "#FFFFFF",
    "@SitePageBorderColor": "#CCD7DF",
    "@ButtonSecondaryUseGradient": "false",
    "@LinkVisitedColor": "#195d8d",
    "@LinkColor": "#195d8d",
    "@ButtonPrimaryBackgroundColor": "#0572ce",
    "@NavbarBackgroundColor": "#FFFFFF",
    "@SubNavigationLinkColor": "#3D3D3D",
    "@ButtonPrimaryFontFamily": "@sansFontFamily",
    "@ButtonPrimaryTextDecoration": "none",
    "@ButtonSecondaryBorderRadius": "4px",
    "@ButtonSecondaryFontStyle": "normal",
    "@ButtonSecondaryTextColor": "#3d3d3d",
    "@SubNavigationBackgroundColor": "#FFFFFF",
    "@ButtonSecondaryFontWeight": "normal",
    "@ButtonSecondaryBackgroundColor": "#ffffff",
    "@ButtonPrimaryBorderRadius": "4px",
    "@LinkHoverColor": "#114062",
    "@ButtonSecondarySize": "medium",
    "@SubNavigationBackgroundHoverColor": "#FFFFFF",
    "@TextColor": "#000000"
  },
```

```
"usingCodeView": false,
"is_default": false,
"associatedSites": [],
"theme_styles_typography": {
  "@H6LineHeight": "150%",
  "@H5TextDecoration": "inherit",
  "@H3FontFamily": "inherit",
  "@H5FontStyle": "inherit",
  "@H2FontStyle": "inherit",
  "@H2TextColor": "inherit",
  "@SiteLineHeight": "150%",
  "@SiteFontFamily": "@sansFontFamily",
  "@H1TextAlign": "inherit",
  "@ParagraphFontWeight": "normal",
  "@H5FontSize": "1.00rem",
  "@H4TextAlign": "inherit",
  "@H1TextDecoration": "inherit",
  "@H3LineHeight": "150%",
  "@H2FontWeight": "bold",
  "@H3FontSize": "1.75rem",
  "@ParagraphLineHeight": "150%",
  "@H3TextColor": "inherit",
  "@H3TextAlign": "inherit",
  "@H4FontFamily": "inherit",
  "@SiteFontSize": "14px",
  "@SiteTextAlign": "left",
  "@H5FontWeight": "normal",
  "@H6FontFamily": "inherit",
  "@H1FontSize": "2.75rem",
  "@H6FontStyle": "inherit",
  "@H6TextDecoration": "inherit",
  "@H2TextDecoration": "inherit",
  "@ParagraphTextColor": "inherit",
  "@H1FontStyle": "inherit",
  "@ParagraphTextAlign": "inherit",
  "@H4LineHeight": "150%",
  "@H1FontWeight": "bold",
  "@SiteTextColor": "#000000",
  "@ParagraphFontSize": "1.00rem",
  "@H5FontFamily": "inherit",
  "@H4FontWeight": "bold",
  "@ParagraphFontFamily": "inherit",
  "@H1FontFamily": "inherit",
  "@H2TextAlign": "inherit",
  "@H1LineHeight": "150%",
  "@H6FontSize": "0.85rem",
  "@H3TextDecoration": "inherit",
  "@H4TextColor": "inherit",
  "@ParagraphFontStyle": "inherit",
  "@ParagraphTextDecoration": "inherit",
  "@H1TextColor": "inherit",
  "@SiteTextDecoration": "none",
  "@H4FontSize": "1.25rem",
  "@SiteFontStyle": "normal",
  "@H3FontWeight": "bold",
```

```
    "@H6TextAlign": "inherit",
    "@H4TextDecoration": "inherit",
    "@H2FontFamily": "inherit",
    "@H3FontStyle": "inherit",
    "@SiteFontWeight": "normal",
    "@H5LineHeight": "150%",
    "@H2FontSize": "2.25rem",
    "@H6TextColor": "inherit",
    "@H4FontStyle": "inherit",
    "@H2LineHeight": "150%",
    "@H5TextColor": "inherit",
    "@H5TextAlign": "inherit",
    "@H6FontWeight": "normal"
}
```

Clone a theme

To create a new theme, clone an existing theme.

1. Issue a `POST` command to:

   ```
   /ccadmin/v1/themes/<name of theme to clone>/clone
   {name: <name of new theme>}
   ```

   For example:

   ```
   POST /ccadmin/v1/themes/PrimarySiteTheme/clone
   {name:"SecondarySiteTheme"}
   ```

2. Make the modifications to the theme as necessary.
3. To set the theme as the active agent theme, issue a `POST` command to:

   ```
   /ccadmin/v1/themes/{id}/setAsAgentTheme
   ```

   For example:

   ```
   POST /ccadmin/v1/themes/SecondarySiteTheme/setAsAgentTheme
   ```

Important: Once you have cloned a theme, you must use the `setAsAgentTheme` endpoint to indicate which theme is the assigned theme. This ensures that you do not have multiple agent themes.

Compile a theme

Once you have created a new agent theme and identified it as such, you compile it.

Issue a `POST` command to:

```
/ccadmin/v1/themes/compileAgentTheme
```

For example:

```
POST /ccadmin/v1/themes/compileAgentTheme
```

Update theme details

To update the agent theme's detail, such as the theme name, notes or details about the theme, do the following:

1. Issue a `GET` command to:

   ```
   /ccadmin/v1/themes/agentThemeDetails
   ```

2. This returns the theme's details. Update the theme elements as necessary.
3. Once you have finished making edits, issue a `PUT` command, with the updated JSON values, to:

   ```
   /ccadmin/v1/themes/agentThemeDetails
   ```

   For example:

   ```
   POST /ccadmin/v1/themes/agentThemeDetails
   {"assetType", "pageLayout", "assetId", "404PageLayout"}
   ```

Update the theme source

1. Issue `GET` command to:

   ```
   /ccadmin/v1/themes/agentThemeSource
   ```

2. Make the modifications as necessary.
3. To reload the theme, issue PUT command with the updated JSON values to:

   ```
   /ccadmin/v1/themes/agentThemeSource
   ```