Cloudinary PWA Integration Guide

****

Contents

[1 Headless (OCAPI) Architecture 3](#_Toc131796979)

[1.1 Configure OCAPI 3](#_Toc131796980)

[1.2 Add Shop API Resources 3](#_Toc131796981)

[1.3 Business Manager Cartridge Path 4](#_Toc131796982)

[1.4 OCAPI Endpoint Customizations 5](#_Toc131796983)

[1.4.1 Product Detail Page (PDP) 5](#_Toc131796984)

[1.4.2 Product Listing Page (PLP/CLP) 7](#_Toc131796985)

[1.4.3 Search Suggestion 7](#_Toc131796986)

[1.4.4 Cart and Checkout Page 8](#_Toc131796987)

[1.4.5 Order Confirmation & Order History 9](#_Toc131796988)

[1.5 Content Assets and Content Slots 10](#_Toc131796989)

[2 Cloudinary Necessary Code Implementation 11](#_Toc131796990)

[2.1 Cloudinary Scripts 11](#_Toc131796991)

[2.2 Cloudinary Image Gallery 11](#_Toc131796992)

[2.3 Cloudinary Videos 11](#_Toc131796993)

[3 Salesforce Commerce Cloud PWA KIT 12](#_Toc131796994)

Headless (OCAPI) Architecture

Configure OCAPI

This section describes how to configure Shop API resources under OCAPI settings. Cloudinary recommends that you configure Shop API version 21.1 or later for optimum utilization of the cartridge.

**Note**

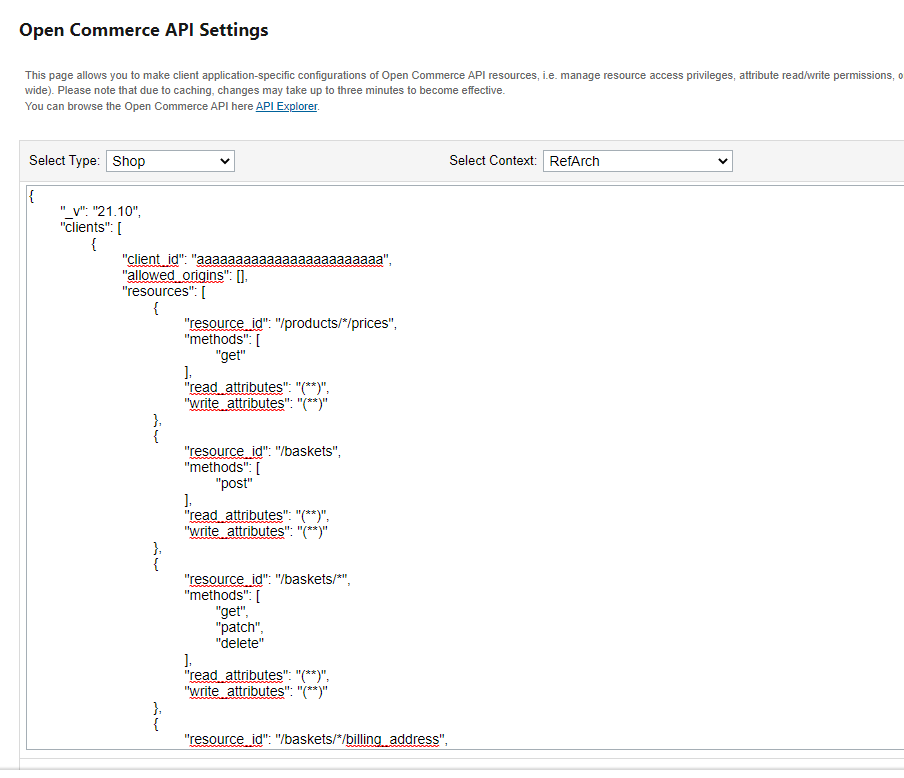
You can generate a new client ID and password by logging in to the Commerce Cloud Account Manager.

Add Shop API Resources

For the cartridge to work, you must add Shop API resources to the cartridge.

To add resources:

1. On the target site, go to **Administration** > **Site Development** > **Open Commerce API Settings**.
2. From the **Select type**drop-down, select **Shop**.



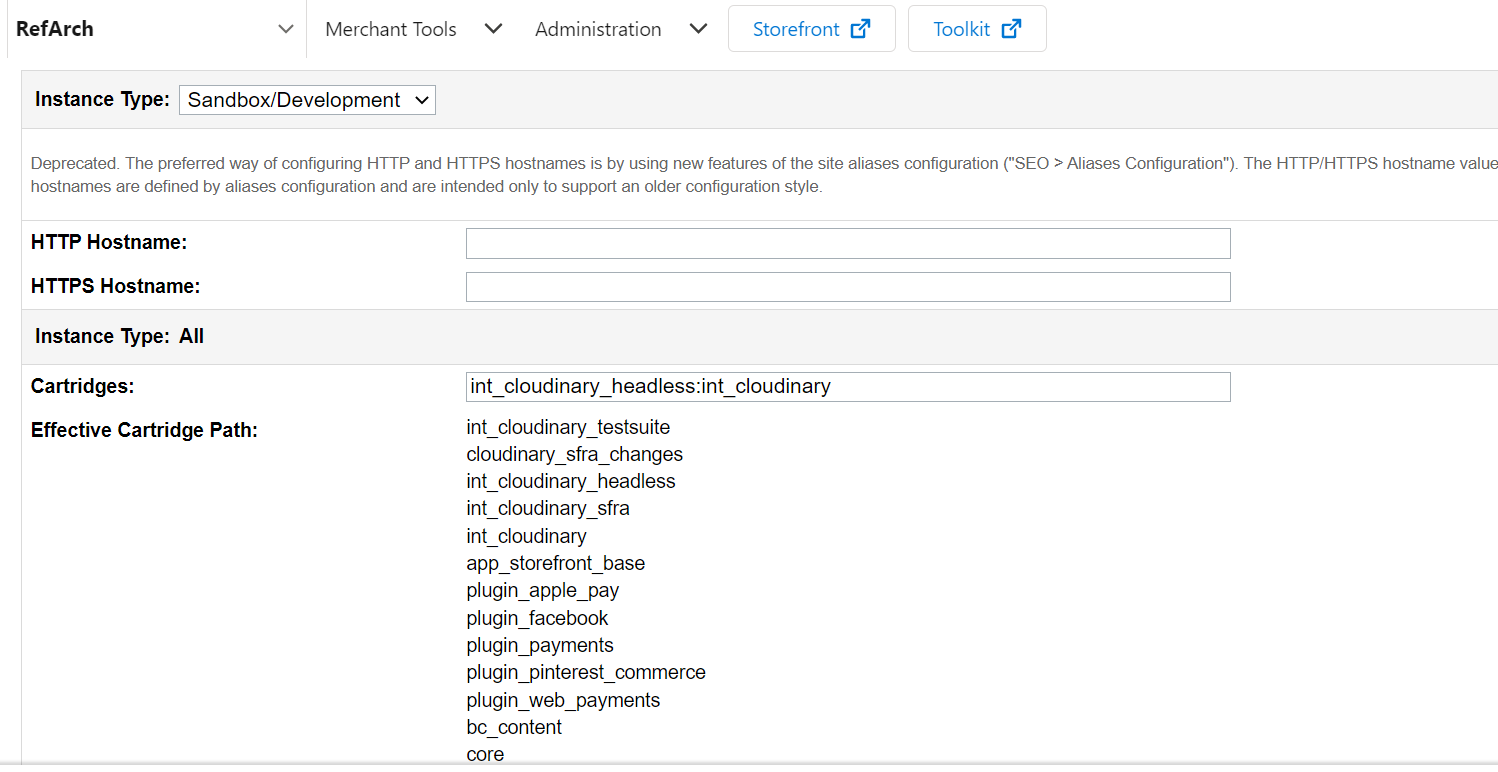
1. From the **Select Context** drop-down, select the appropriate context, for example, Global (organization-wide) or site-specific.
2. Enter the OCAPI data API resources provided in the **ocapi-configurations.json** file in the **documentation**folder.
3. Update the **client\_id** with the one you generated from the Commerce Cloud Account Manager.
4. Click **Save**.

Note: **int\_cloudinary\_headless** is an overlay cartridge that includes all OCAPI customizations. You can duplicate the customizations in your cartridge by including the function calls in your OCAPI hook call files.

Business Manager Cartridge Path

For PWA KIT to work, add the following cartridges in the cartridge path for PWA KIT.

int\_cloudinary\_headless:int\_cloudinary



OCAPI Endpoint Customizations

Product Detail Page (PDP)

* + - 1. OCAPI Endpoint

GET http://hostname:port/dw/shop/v21\_10/products/{id}

* + - 1. OCAPI Response

Product Document (Get product images, image gallery , and videos from Cloudinary and assigned them to CLD custom attributes.)

|  |  |
| --- | --- |
| **CLD Attribute** | **Attribute Details** |
| c\_cloudinary | Object (Contains data of CLD images and videos of all page types) |
| c\_cloudinary.cloudinaryImage | This attribute contains the image gallery obtained from Cloudinary |
| c\_cloudinary.video | This attribute contains Cloudinary product video URLs |
| c\_cloudinary.cldSwatchs | This attribute contains Cloudinary product swatch images data |
| c\_cloudinary.cartImage | Contains product line-item image for Cart |
| c\_cloudinary.checkoutImage | Contains product line-item image for Checkout |
| c\_cloudinary.miniCartImage | Contains product line-item image for Mini Cart |
| c\_cloudinary.orderConfirmationImage | Contains product line-item image for Order Confirmation |
| c\_cloudinary.pdpSwatch | Contains Cloudinary site preference value of enabling/disabling product swatches on PDP |
| c\_cloudinary.images | Contains URLs of product images received from CLD |
| c\_cloudinary.isEnabled | Contains Cloudinary site preference value of enabling/disabling Cloudinary Cartridge |
| c\_cloudinary.galleryEnabled | Contains Cloudinary site preference value of enabling/disabling product gallery on PDP |
| c\_cloudinary.cloudName | Contains Cloudinary site preference value of cloud name for rendering product video |
| c\_cloudinary.videoEnabled | Contains Cloudinary site preference value of enabling/disabling videos on the site |
| c\_cloudinary.videoPlayerEnabled | Contains Cloudinary site preference value of enabling/disabling videos on the site |
| c\_cloudinary.pdp | Contains Cloudinary site preference value of enabling/disabling CLD Images product detail page |
| c\_cloudinary.cartEnabled | Contains Cloudinary site preference value of enabling/disabling CLD cart images |
| c\_cloudinary.isCheckoutEnabled | Contains Cloudinary site preference value of enabling/disabling CLD checkout images |
| c\_cloudinary.miniCartEnable | Contains Cloudinary site preference value of enabling/disabling CLD images on add-to-cart-model |
| c\_cloudinary.orderConfirmation | Contains Cloudinary site preference value of enabling/disabling CLD order confirmation images |
| c\_cloudinary.orderHistory | Contains Cloudinary site preference value of enabling/disabling CLD images on order history and order detail page |
| c\_cloudinary.randomNumber | Contains random number used for rendering CLD video on PDP |

* + - 1. Customizations

|  |  |  |
| --- | --- | --- |
| **Extension Point** | **CLD Extension Script** | **Notes** |
| dw.ocapi.shop.product.modifyGETResponse | /int\_cloudinary\_headless/cartridge/scripts/hooks/productDetails.js | This extension script is mainly used to get product images and videos from CLD for PDP, Cart, Checkout, Order Confirmation, Order History, and Order Detail. |

* + 1. Product Listing Page (PLP/CLP)

GET http://hostname:port/dw/shop/v21\_10/product\_search

You can add search-related query parameters to the URL as well.

* + - 1. OCAPI Response

Product Search Result Hits (Product Images are get from Cloudinary and assigned to CLD custom attributes.)

|  |  |
| --- | --- |
| **CLD Attribute** | **Attribute Details** |
| c\_cloudinary | Contains Cloudinary product images for Product Tile Page |
| c\_cloudinary.c\_autoResponsiveDimensions | Contains Site Preference Image Page Type Settings for PLP |
| c\_cloudinary.plpEnabled | Contains Site Preference value to Enable/Disable CLD features on PLP |

* + - 1. Customizations

|  |  |  |
| --- | --- | --- |
| **Extension Point** | **CLD Extension Script** | **Notes** |
| dw.ocapi.shop.product\_search.modifyGETResponse | int\_cloudinary\_headless/cartridge/scripts/hooks/productSearch.js | This extension script is mainly used to get product images from CLD and render it on PLP. |

* + 1. Search Suggestion

POST http://hostname:port/dw/shop/v21\_10/search\_suggestion

* + - 1. OCAPI Response

|  |  |
| --- | --- |
| **CLD Attribute** | **Attribute Details** |
| c\_cloudinary | Contains Cloudinary product images for search suggestion |
| c\_cloudinary.suggestionEnabled | Site Preference value to Enable/Disable CLD product images on search suggestions |

* + - 1. Customizations

|  |  |  |
| --- | --- | --- |
| **Extension Point** | **CLD Extension Script** | **Notes** |
| dw.ocapi.shop.search\_suggestion.modifyGETResponse | /int\_cloudinary\_headless/cartridge/scripts/hooks/searchSuggestions.js | This extension script is mainly used to get CLD images for Search Suggestions. |

Cart and Checkout Page

**Note:** Headless storefront does not have an ocapi call of the basket after add-to-cart, but if anybody using frontend technology other than headless then he can use this hook to render Cloudinary images on basket product line items.

POST http://hostname:port/dw/shop/21\_10/baskets/{basket\_id}/items

* + - 1. OCAPI Response

|  |  |
| --- | --- |
| **CLD Attribute** | **Attribute Details** |
| c\_cloudinary | Contains Cloudinary product images objects |
| c\_ cloudinary.miniCartImage | Contains product line-item image for Mini Cart |
| c\_ cloudinary.cartImage | Contains product line-item image for Cart |
| c\_cloudinary.checkoutImage | Contains product line-item image for Checkout |
| c\_cloudinary.miniCartEnabled | Contains Cloudinary site preference value of enabling/disabling CLD images on add-to-cart-model |
| c\_cloudinary.cartEnabled | Contains Cloudinary site preference value of enabling/disabling CLD cart images |
| c\_cloudinary.checkoutEnabled | Contains Cloudinary site preference value of enabling/disabling CLD checkout images |

#### **Customization**

|  |  |  |
| --- | --- | --- |
| **Extension Point** | **CLD Extension Script** | **Notes** |
| dw.ocapi.shop.basket.items.modifyPOSTResponse | /int\_cloudinary\_headless/cartridge/scripts/hooks/basket.js | This extension script is mainly used to add CLD images on basket product line items. |

Order Confirmation & Order History

**Note:** If anybody using frontend technology other than headless then he can use this hook to render Cloudinary images on basket product line items.

GET http://hostname:port/dw/shop/v21\_10/orders/{order\_no}

|  |  |
| --- | --- |
| **CLD Attribute** | **Attribute Details** |
| c\_cloudinary | Contains Cloudinary product images objects |
| c\_cloudinary.orderConfirmationImage | Contains CLD images for Order product line item |
| c\_cloudinary.orderConfirmationEnabled | Contains Site Preference value to enable/disable Cloudinary on the order confirmation |
| c\_cloudinary.orderHistoryEnabled | Contains Site Preference value to enable/disable Cloudinary on Order History |

#### **Customization**

|  |  |  |
| --- | --- | --- |
| **Extension Point** | **CLD Extension Script** | **Notes** |
| dw.ocapi.shop.order.modifyGETResponse | /int\_cloudinary\_headless/cartridge/scripts/hooks/order.js | This extension script is mainly used to add CLD images on order product line items. |

Content Assets and Content Slots

Cloudinary provides endpoints that must be called from the frontend, whether it's a PWA or any other frontend technology, to render content assets and content slots. To obtain the content asset, the endpoint below needs to be triggered from the frontend:

https://${cloudinary.paramenters.host}/on/demandware.store/Sites-${ cloudinary.paramenters.siteId}-Site/default/CloudinaryStaticContent-RenderAsset?contentId=${cloudinary.paramenters.contentAssetId}

For Content Slots:

At this moment we include out of the box SFCC SFRA content slots but feel free to create your own controllers or templates to render the slots.  
  
https://${ cloudinary.paramenters.host}/on/demandware.store/Sites-${ cloudinary.paramenters.siteId}-Site/default/CloudinaryStaticContent-RenderSlots

Cloudinary Necessary Code Implementation

Cloudinary Scripts

To integrate Cloudinary's product image gallery and video player into your frontend application, it is necessary to include the following Cloudinary scripts, irrespective of the frontend technology you are using.

<script src='https://unpkg.com/cloudinary-core@2.6.3/cloudinary-core-shrinkwrap.min.js'></script>

<script src='https://unpkg.com/cloudinary-video-player@1.3.3/dist/cld-video-player.min.js'></script>

<script src='https://product-gallery.cloudinary.com/1.0.50/all.js'></script>

<link rel="stylesheet" href="https://unpkg.com/cloudinary-video-player@1.3.3/dist/cld-video-player.min.css">

These scripts are essential to ensure that Cloudinary's product image gallery and video player are fully functional on your website.

Cloudinary Image Gallery

To render Cloudinary image gallery, please make the following necessary JS Cloudinary implementations.

var galleryOptions = c\_cloudinary.cloudinaryImage.galleryWidget.options;

window.cldGallery = cloudinary.galleryWidget(galleryOptions);

cldGallery.render();

Make sure to include a div with the id **"cld-gallery"** in your HTML code. This is where the Cloudinary product image gallery will be rendered.

<div id="cld-gallery"></div>

Here c\_cloudinary is the object that contains data of Cloudinary image gallery retrieved from cloudinary.

Cloudinary Videos

To render Cloudinary videos, please make the following JS implementations.

var cld = cloudinary.Cloudinary.new({ cloud\_name: c\_cloudinary.cloudName })

var player = cld.videoPlayer('cld-video-player' + (videoPlayerID ? '-' + videoPlayerID : ''),cldObj.video.widgetOptions);

player.source(cldObj.video.videoURL, {}).play();

player.transformation(cldObj.video.widgetOptions.transformations);

<video id={`cld-video-player-${videoPlayerID}`} class="cld-video-player cloudinary-data-container"></video>

Salesforce Commerce Cloud PWA KIT

If you are using React as your frontend on PWA KIT Headless, then you don’t need to follow the above steps. You just need to implement the following code changes on your PWA KIT that are mentioned below:

**File1:** packages\pwa-cloudinary\overrides\app\components\footer\index.jsx

**Line No:** 34 to 36

****

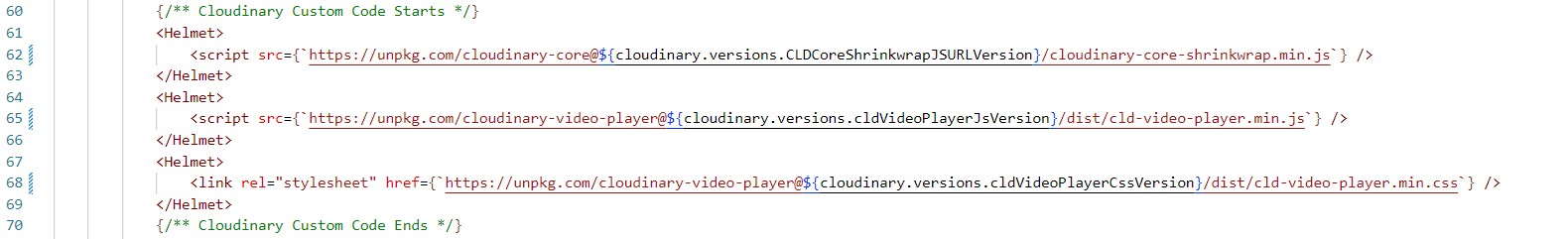
{/\*\* Cloudinary Custom Code Starts \*/}

import Helmet from 'react-helmet'

import { cloudinary } from '../../../../config/default'

{/\*\* Cloudinary Custom Code Ends \*/}

**Line No:** 60 to 70

****

{/\*\* Cloudinary Custom Code Starts \*/}

<Helmet>

    <script src={`https://unpkg.com/cloudinary-core@${cloudinary.versions.CLDCoreShrinkwrapJSURLVersion}/cloudinary-core-shrinkwrap.min.js`} />

</Helmet>

<Helmet>

    <script src={`https://unpkg.com/cloudinary-video-player@${cloudinary.versions.cldVideoPlayerJsVersion}/dist/cld-video-player.min.js`} />

</Helmet>

<Helmet>

    <link rel="stylesheet" href={`https://unpkg.com/cloudinary-video-player@${cloudinary.versions.cldVideoPlayerCssVersion}/dist/cld-video-player.min.css`} />

</Helmet>

{/\*\* Cloudinary Custom Code Ends \*/}

**File2:** packages\pwa-cloudinary\overrides\app\components\order-summary\index.jsx

**Line No:** 26 to 27



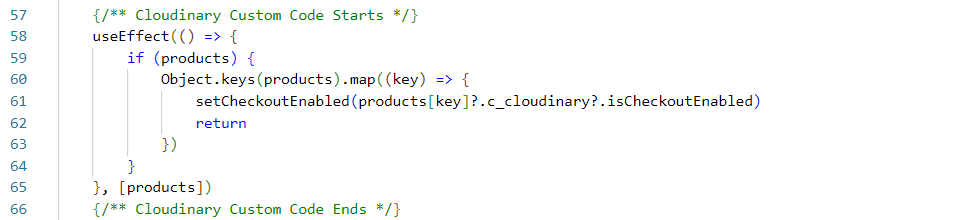
// Cloudinary Component to Render CLD images on Checkout

import CldCartItemVariantImage from '../item-variant/cloudinary-item-image'

**Line No:** 30 to 32



**Line No:** 57 to 66

****

    {/\*\* Cloudinary Custom Code Starts \*/}

    useEffect(() => {

        if (products) {

            Object.keys(products).map((key) => {

                setCheckoutEnabled(products[key]?.c\_cloudinary?.isCheckoutEnabled)

                return

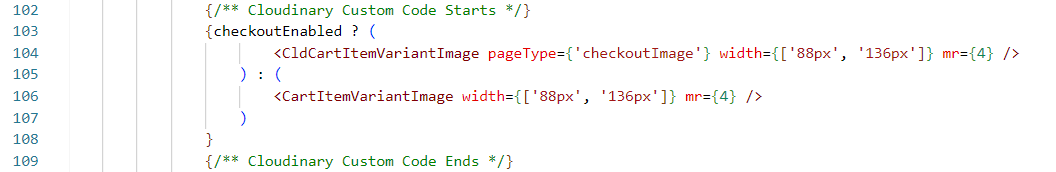
            })

        }

    }, [products])

    {/\*\* Cloudinary Custom Code Ends \*/}

**Line No:** 102 to 109

****

{/\*\* Cloudinary Custom Code Starts \*/}

{checkoutEnabled ? (

    <CldCartItemVariantImage pageType={'checkoutImage'} width={['88px', '136px']} mr={4} />

 ) : (

     <CartItemVariantImage width={['88px', '136px']} mr={4} />

     )

}

{/\*\* Cloudinary Custom Code Ends \*/}

**File3:** packages\pwa-cloudinary\overrides\app\components\product-item\index.jsx

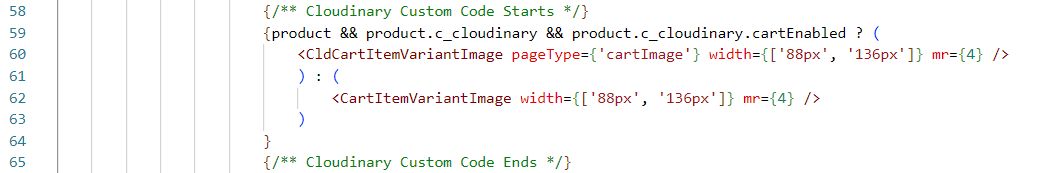
**Line No:** 30 to 31



// Cloudinary Component to Render CLD images on Cart

import CldCartItemVariantImage from '../../../../app/components/cloudinary-item-image'

**Line No:** 58 to 65

****

{/\*\* Cloudinary Custom Code Starts \*/}

{product && product.c\_cloudinary && product.c\_cloudinary.cartEnabled ? (

    <CldCartItemVariantImage pageType={'cartImage'} width={['88px', '136px']} mr={4} />

  ) : (

     <CartItemVariantImage width={['88px', '136px']} mr={4} />

     )

}

{/\*\* Cloudinary Custom Code Ends \*/}

**File4:** packages\pwa-cloudinary\overrides\app\components\product-tile\index.jsx

**Line No:** 33 to 34

****

// Cloudinary Component to Render CLD Images on PLP

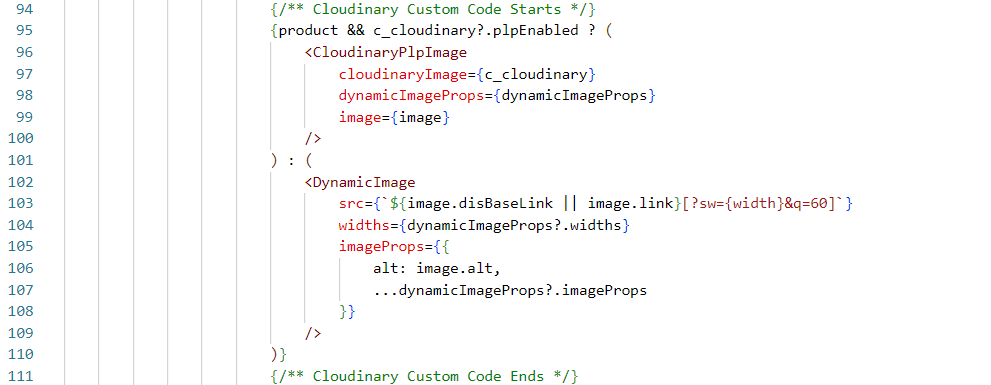
import CloudinaryPlpImage from '../../components/cloudinary-plp-images'

**Line No:** 72

****

const {currency, image, price, productId, hitType, c\_cloudinary} = product

**Line No:** 94 to 111



{/\*\* Cloudinary Custom Code Starts \*/}

{product && c\_cloudinary && c\_cloudinary.plpEnabled ? (

     <CloudinaryPlpImage

         cloudinaryImage={c\_cloudinary}

         dynamicImageProps={dynamicImageProps}

         image={image}

      />

) : (

    <DynamicImage

         src={`${image.disBaseLink || image.link}[?sw={width}&q=60]`}

         widths={dynamicImageProps?.widths}

         imageProps={{

          alt: image.alt,

          ...dynamicImageProps?.imageProps

         }}

      />

)}

{/\*\* Cloudinary Custom Code Ends \*/}

**File5:** packages\pwa-cloudinary\overrides\app\hooks\use-add-to-cart-model.js

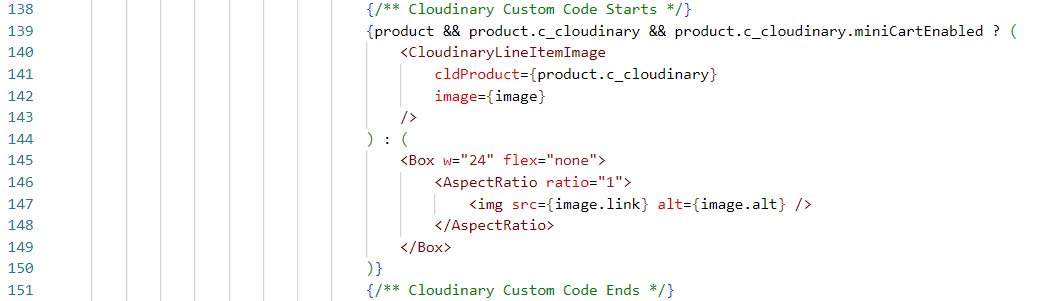
**Line No:** 35 to 36

****

// Cloudinary Component to Render CLD images on add-to-cart-model

import CloudinaryLineItemImage from '../components/cloudinary-lineItem-image'

**Line No:** 138 to 151

****

{/\*\* Cloudinary Custom Code Starts \*/}

{product && product.c\_cloudinary && product.c\_cloudinary.miniCartEnabled ? (

    <CloudinaryLineItemImage

         cldProduct={product.c\_cloudinary}

         image={image}

     />

) : (

    <Box w="24" flex="none">

       <AspectRatio ratio="1">

           <img src={image.link} alt={image.alt} />

       </AspectRatio>

     </Box>

)}

{/\*\* Cloudinary Custom Code Ends \*/}

**File6:** packages\pwa-cloudinary\overrides\app\pages\account\order-detail.jsx

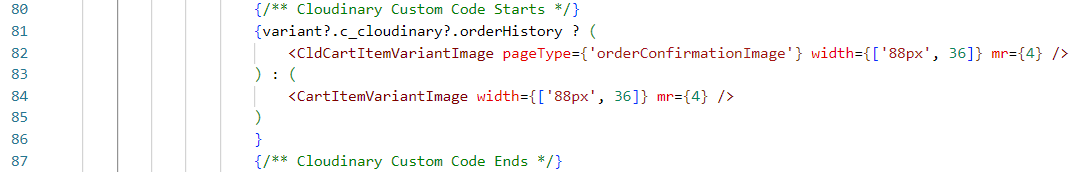
**Line No:** 37 to 38



// Cloudinary Component to Render CLD images on order-history

import CldCartItemVariantImage from '../../components/item-variant/cloudinary-item-image'

**Line No:** 80 to 87



{/\*\* Cloudinary Custom Code Starts \*/}

{variant?.c\_cloudinary?.orderHistory ? (

<CldCartItemVariantImage pageType={'orderConfirmationImage'} width={['88px', 36]} mr={4} />

) : (

<CartItemVariantImage width={['88px', 36]} mr={4} />

)

}

{/\*\* Cloudinary Custom Code Ends \*/}

**File7:** packages\pwa-cloudinary\overrides\app\pages\account\order-history.jsx

**Line No:** 36 to 37



// Cloudinary Component to Render CLD images on order-history

import CloudinaryLineItemImage from '../../components/cloudinary-lineItem-image'

**Line No:** 53 to 87

****

            {products && products[0]?.c\_cloudinary?.orderHistory ? (

                products?.map((product) => (

                    <CloudinaryLineItemImage

                        pageType={'orderConfirmationImage'}

                        cldProduct={product?.c\_cloudinary.orderConfirmationImage}

                        image

                    />

                )

                )

            ) : (

                <Box>

                    {!isLoading && products ?

                        images.map((image, index) => {

                            return (

                                <AspectRatio

                                    key={index}

                                    ratio={1}

                                    width="88px"

                                    w="88px"

                                    borderRadius="base"

                                    overflow="hidden"

                                >

                                    <Img

                                        alt={image?.alt}

                                        src={image?.disBaseLink || image?.link}

                                        fallback={<Box background="gray.100" boxSize="full" />}

                                    />

                                </AspectRatio>

                            )

                        })

                        : productItems.map((item, index) => {

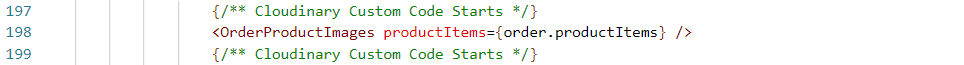
                            return <Skeleton key={index} h="88px" w="88px" />

                        })}

                </Box>

            )}

**Line No:** 197 to 199

****

{/\*\* Cloudinary Custom Code Starts \*/}

<OrderProductImages productItems={order.productItems} />

{/\*\* Cloudinary Custom Code Starts \*/}

**File8:** packages\pwa-cloudinary\overrides\app\pages\account\confirmation.jsx

**Line No:** 40 to 42



// Cloudinary Component to Render CLD images on Order-Confirmation

import CldCartItemVariantImage from '../../components/item-variant/cloudinary-item-image'

**Line No:** 50 to 52

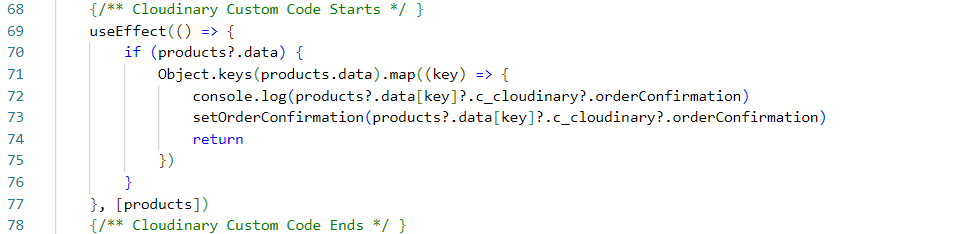


{/\*\* Cloudinary Custom Code Starts \*/ }

const [orderConfirmation, setOrderConfirmation] = useState(false)

{/\*\* Cloudinary Custom Code Ends \*/ }

**Line No:** 68 to 78



{/\*\* Cloudinary Custom Code Starts \*/ }

    useEffect(() => {

        if (products?.data) {

            Object.keys(products.data).map((key) => {

                console.log(products?.data[key]?.c\_cloudinary?.orderConfirmation)

                setOrderConfirmation(products?.data[key]?.c\_cloudinary?.orderConfirmation)

                return

            })

        }

    }, [products])

    {/\*\* Cloudinary Custom Code Ends \*/ }

**Line No:** 330 to 343



{/\*\* Cloudinary Custom Code Starts \*/ }

{

    orderConfirmation ? (

        <CldCartItemVariantImage

            pageType={'orderConfirmationImage'}

            width="80px"

            mr={2}

        />

    ) : (

        <CartItemVariantImage

            width="80px"

            mr={2}

        />

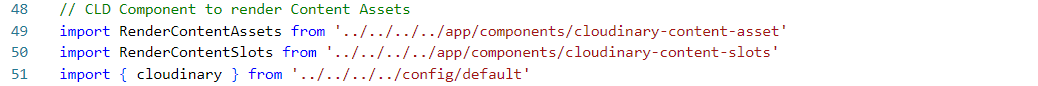
    )

}

{/\*\* Cloudinary Custom Code Ends \*/ }

**File9:** packages\pwa-cloudinary\overrides\app\pages\home\index.jsx

**Line No:** 48 to 51



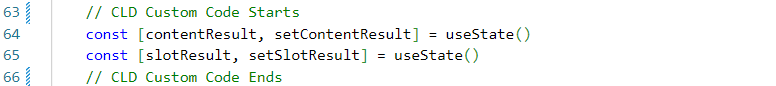
// CLD Component to render Content Assets

import RenderContentAssets from '../../../../app/components/cloudinary-content-asset'

import RenderContentSlots from '../../../../app/components/cloudinary-content-slots'

import { cloudinary } from '../../../../config/default'

**Line No:** 63 to 66

****

    // CLD Custom Code Starts

    const [contentResult, setContentResult] = useState()

    const [slotResult, setSlotResult] = useState()

    // CLD Custom Code Ends

**Line No:** 79 to 100



// CLD Custom Code Starts

const getAsset = async () => {

    //Make a call to get Content Assets from SFCC

    const res = await fetch(

        `https://${cloudinary.parameters.host}/on/demandware.store/Sites-${cloudinary.parameters.siteId}-Site/default/CloudinaryStaticContent-RenderAsset?contentId=${cloudinary.parameters.contentAssetId}`

    )

    if (res.ok) {

        setContentResult(await res.text())

    }

    //Make a call to get Content Slots from SFCC

    const slotRes = await fetch(

        `https://${cloudinary.parameters.host}/on/demandware.store/Sites-${cloudinary.parameters.siteId}-Site/default/CloudinaryStaticContent-RenderSlots`

    )

    if (slotRes.ok) {

        setSlotResult(await slotRes.text())

    }

    if (process.env.NODE\_ENV !== 'production') {

        console.log(contentResult)

        console.log(slotResult)

    }

}

// CLD Custom Code Ends

**Line No:** 105 to 107

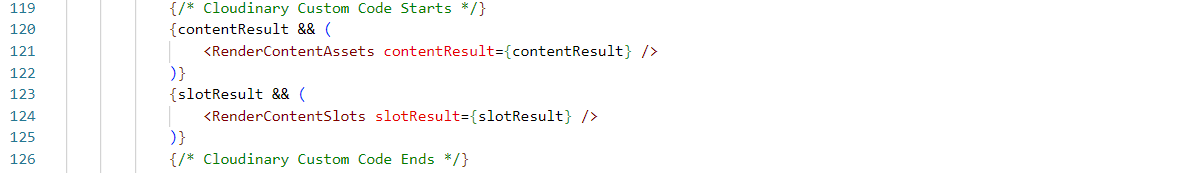


{/\* Cloudinary Custom Code Starts \*/ }

getAsset()

{/\* Cloudinary Custom Code Ends \*/ }

**Line No:** 119 to 126



{/\* Cloudinary Custom Code Starts \*/}

{contentResult && (

    <RenderContentAssets contentResult={contentResult} />

)}

{slotResult && (

    <RenderContentSlots slotResult={slotResult} />

)}

{/\* Cloudinary Custom Code Ends \*/}

contentAssetId, host and siteId used in screenshot of above script are declared inside default file, contains path below:

**packages\pwa-cloudinary\config\default.js**

****

        cloudinary: {

        parameters: {

            clientId: '6ba55594-b69a-42da-b17b-6cd85ede044f',

            organizationId: 'f\_ecom\_zzgt\_003',

            shortCode: 'kv7kzm78',

            siteId: 'RefArch',

            host: 'zzgt-003.dx.commercecloud.salesforce.com',

            contentAssetId: 'home-main'

        },

        versions: {

            CLDVideoPlayerJsVersion: '1.10.1',

            CLDVideoPlayerCssVersion: '1.10.1',

            CLDCoreShrinkwrapJSURLVersion: '2.13.0',

            CLDGalleryVersion: '1.0.50'

        }

    }

**File10:** packages\pwa-cloudinary\overrides\app\pages\product-detail\index.jsx

**Line No**: 31

Replace the code with

****

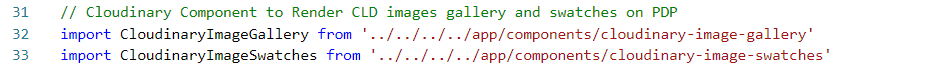
import ProductView from '../../components/product-view'

**File11:** packages\pwa-cloudinary\overrides\app\components\product-view\index.jsx

**Line No:** 25  


import { useAddToCartModalContext } from '../../hooks/use-add-to-cart-modal'

**Line No:** 42 to 44



// Cloudinary Component to Render CLD images gallery and swatches on PDP

import CloudinaryImageGallery from '../../components/cloudinary-image-gallery'

import CloudinaryImageSwatches from '../../components/cloudinary-image-swatches'

**Line No:** 306 to 343



{/\*\* Cloudinary Custom Code Starts \*/ }

{

    product?.c\_cloudinary?.pdp ? (

        <Box flex={1} mr={[0, 0, 0, 6, 6]}>

            <CloudinaryImageGallery

                size={imageSize}

                cloudinaryImageGallery={product?.c\_cloudinary ? product?.c\_cloudinary : null}

            />

        </Box>

    ) : (

    <Box flex={1} mr={[0, 0, 0, 6, 6]}>

        {product ? (

            <>

                <ImageGallery

                    size={imageSize}

                    imageGroups={product.imageGroups}

                    selectedVariationAttributes={variationParams}

                    lazy={isProductPartOfSet}

                />

                <HideOnMobile>

                    {showFullLink && product && (

                        <Link to={`/product/${product.master.masterId}`}>

                            <Text color="blue.600">

                                {intl.formatMessage({

                                    defaultMessage: 'See full details',

                                    id: 'product\_view.link.full\_details'

                                })}

                            </Text>

                        </Link>

                    )}

                </HideOnMobile>

            </>

        ) : (

            <ImageGallerySkeleton />

        )}

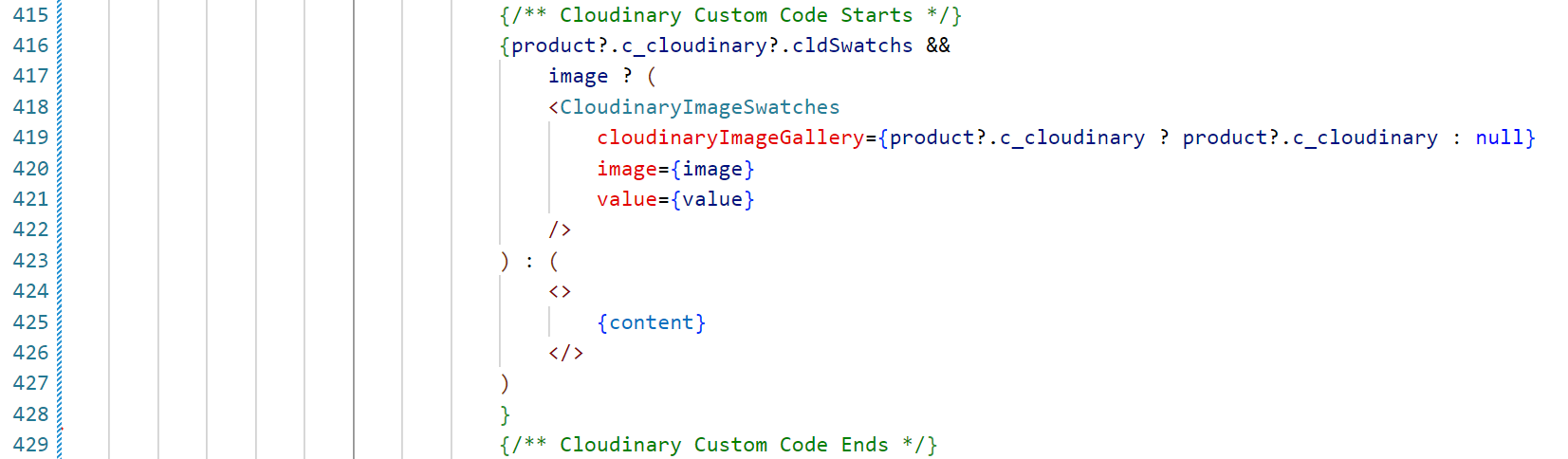
    </Box>

)

}

{/\*\* Cloudinary Custom Code Ends \*/ }

**Line No:** 415 to 429



{/\*\* Cloudinary Custom Code Starts \*/}

{product?.c\_cloudinary?.cldSwatchs &&

image ? (

    <CloudinaryImageSwatches

        cloudinaryImageGallery={product?.c\_cloudinary ? product?.c\_cloudinary : null}

        image={image}

        value={value}

    />

) : (

    <>

        {content}

    </>

)

}

{/\*\* Cloudinary Custom Code Ends \*/}

**3.3.1 Components to be used in PWA:**

* Cloudinary-content-asset **(located at packages/pwa-cloudinary/app/components/cloudinary-content-asset)**is used for rendering content assets.
* Cloudinary-content-slots **(located at packages/pwa-cloudinary/app/components/cloudinary-content-slots)** are used for rendering content slots.
* Cloudinary-image-gallery **(located at packages/pwa-cloudinary/app/components/cloudinary-image-gallery)** is used for rendering Cloudinary image galleries on a PDP.
* Cloudinary-image-swatches **(located at packages/pwa-cloudinary/app/components/cloudinary-image-swatches)** are used for rendering Cloudinary swatches on a PDP.
* Cloudinary-lineItem-image **(located at packages/pwa-cloudinary/app/components/cloudinary-lineItem-image)** is used for rendering Cloudinary images on the cart, checkout, order confirmation, order history, and order detail pages.
* Cloudinary-plp-images **(located at packages/pwa-cloudinary/app/components/cloudinary-plp-images)** is used for rendering Cloudinary images on a PLP.
* Cloudinary-product-video **(located at packages/pwa-cloudinary/app/components/cloudinary-product-video)** is used for rendering Cloudinary videos on a PDP.
* Cloudinary-widgets **(located at packages/pwa-cloudinary/app/components/cloudinary-widgets)** are used for rendering Cloudinary galleries on a PDP.
* item-variant/cld-item-image.jsx **(located at packages/pwa-cloudinary/app/components/cloudinary-item-image.jsx)** is used to call the Cloudinary component to render CLD images on the cart, checkout, order confirmation, order history, and order detail pages.