Media Gallery - Video Support

Contents

[Objective 2](#_Toc78317271)

[Prerequisite 2](#_Toc78317272)

[Modules Extension. 2](#_Toc78317273)

[Steps to consume and test the sample code. 3](#_Toc78317274)

# Objective

Media gallery module supports only product images to display on product description page (PDP) at present. The objective is to leverage the existing media gallery module to support both images and videos through view extensions.

Graphical user interface, website

Description automatically generated

# Prerequisite

* SSK Module code need to be deployed and running.

Reference link: <https://docs.microsoft.com/en-us/dynamics365/commerce/e-commerce-extensibility/setup-dev-environment>

* If you are using specific versions of SDK, make sure to update to **1.30.12**, else, it should automatically download the latest version **(1.30.12)**. Highlighted one is SDK.

A picture containing text, screenshot, marker

Description automatically generated

# Modules Extension.

**Media Gallery View Extension:** Media gallery view can be extended to achieve the desired feature.

# Steps to consume and test the sample code.

1. **Download latest dependencies**: If you are using specific versions of SDK, make sure to update to 1.30.12, else, it should automatically download the latest version (**1.30.12**). Highlighted one is SDK.
   1. Go to package.json file on the root directory and update the suggested package version.
   2. Remove node\_modules and yarn.lock file.
   3. Execute yarn.
   4. Go to node\_module and validate the version of bootloader. This should be **1.30.12** or above.

Graphical user interface, application

Description automatically generated

1. **Media-gallery view extension:** Go to the theme folder and extend media-gallery.view.tsx

**Option-1:** If media-gallery view is not extended, extend it first.

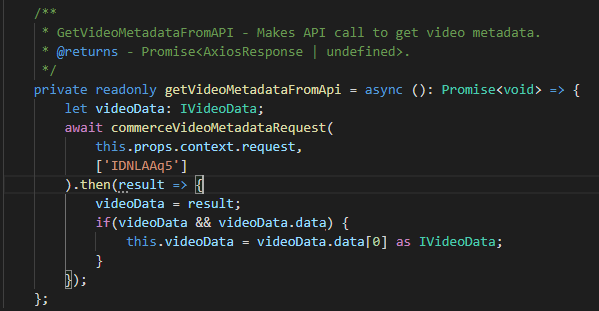
**Ref:** [https://docs.microsoft.com/en-us/dynamics365/commerce/e-commerce-extensibility/theme-module-extensions](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fdynamics365%2Fcommerce%2Fe-commerce-extensibility%2Ftheme-module-extensions&data=04%7C01%7Cv-srpas%40microsoft.com%7C422aafd5ed574b63a20808d92d16bbc6%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637590397940845613%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=%2F3iirUOaQCq7sSE2wikpst9mwzg2WIa9nnY2nGCrsiQ%3D&reserved=0)

Once extended, follow steps mentioned in option-2 to plug-in the required code to achieve the desired feature.

**Option-2**: If already extended, partner need to plug-in below code manually.

1. Create a function ***getVideoMetadataFromApi()*** used to make the API call for retrieving the metadata of Video. Currently, we have hard coded the **video ID** to get the video metadata.

**Note**: Video Identity Id hardcoded in below function can be replaced with product attribute if this video identity id is configured in HQ as product attribute.

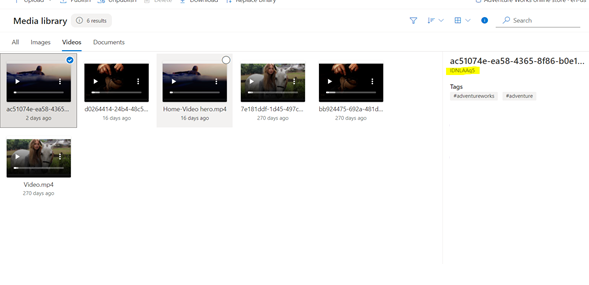


1. Add *getVideoMetadataFromApi()* function in *ComponentDidMount* of media-gellery.view.tsx file to retrieve the video metadata.

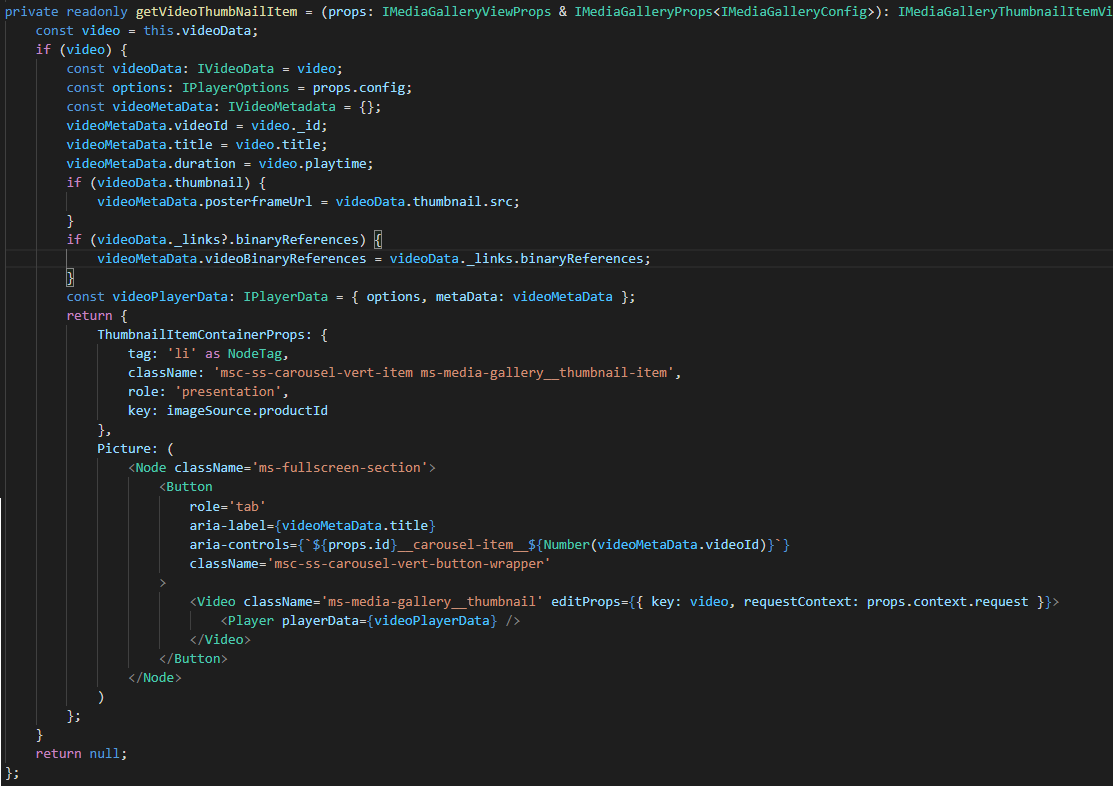
Graphical user interface, text, website

Description automatically generated

We can get this video ID from CMS under “**Media Library**”. Users can select any video to get the ID as shown in the below snapshot.\



1. Once, we get all the necessary video metadata from the API response, we are rendering the Video using the metadata in below function.



**Note**: Please refer sample code file for the same

1. **Build the repository**: Once you are done with all the code changes, you are good to go for testing. Execute *yarn build* from command line interface on your repository.
2. **Test the code changes:**

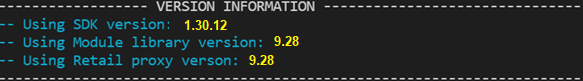
Execute *yarn start*.

Browse the local host 4000 port and navigate to product description page to view the video added in your media gallery along with the product images. Refer below screenshot.

Graphical user interface, website

Description automatically generated

**Note**: The sample code is tested on below package version:

****