**PLP- Custom Pagination**

# Problem Statement:

C1 can set the page size through authoring tool for the product list page. But there is no option for an end user to set the page size on the product list page.

# Objective

To provide an option for C2 to set the custom page size directly on the product list page as needed.

Graphical user interface, website

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

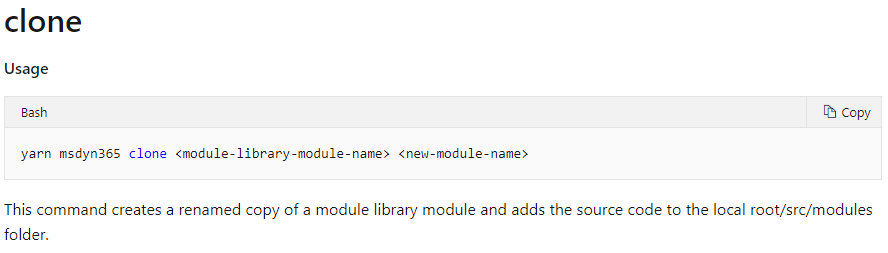
# Prerequisite

* 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>

# Modules Ejected for this feature.

Search Result Container module can be ejected/cloned to achieve the desired feature. Below is the CLI for cloning the core module. Please refer below CLI for cloning the module from module library.



# Steps to consume and test the sample code.

**Option-1:** In case if the search-result-container module is not ejected in the repository you want to implement this feature, you can follow below steps to consume the attached sample code to implement and test this feature.

1. Unzip the attached code and save it to some location.
2. Copy files and folders under the same folder structure in your repository as it’s there in sample.zip.
3. Execute yarn on the root path of your repository.
4. Execute yarn build.
5. Execute yarn start.
6. Browse the configured product using pagemock using below sample url.

https://localhost:4000/partnersite/page?mock=testmock

**Note**: check for the right *partnersite* in .env file of your repository and make a change in url accordingly.

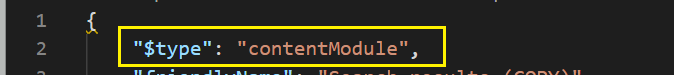
**Option-2:** If the search-result-container module is alreadyejected in the repository you want to implement this feature, you need to manually plugin the code from the attached sample files to the respective files. Please follow below file changes.

**Search Result Container module file changes:** Need to modify below files in search result container module which was already ejected.

**Note:** Sample code contains search-result-container module cloned with new name as custom-search-result-container. So please map and refer the associated files for changing the code manually.

* + - **search-result-container.definition.json (Module definition file)**

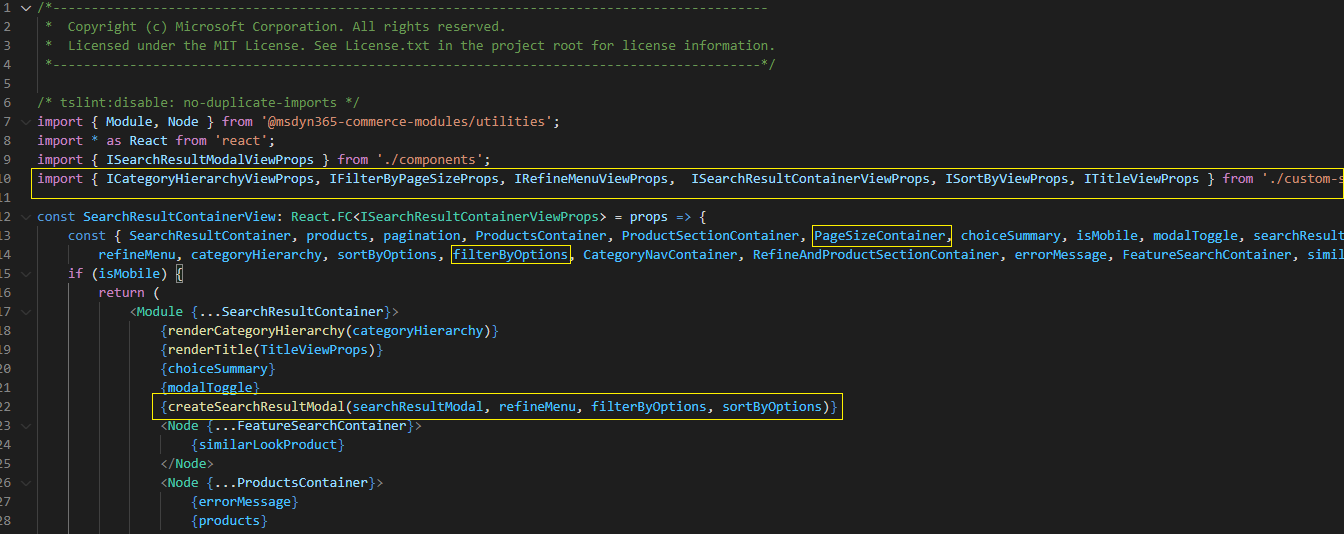
Update type to contentModule



* + - **search-result-container.view.tsx (Module view file)**

Add the code logic to include interface IFilterByPageSizeProps and filterByOptions from props.

Pass the filterByOptions to createSearchResultModal for mobile mode.



Add node block to render pagesize dropdown

Text

Description automatically generated

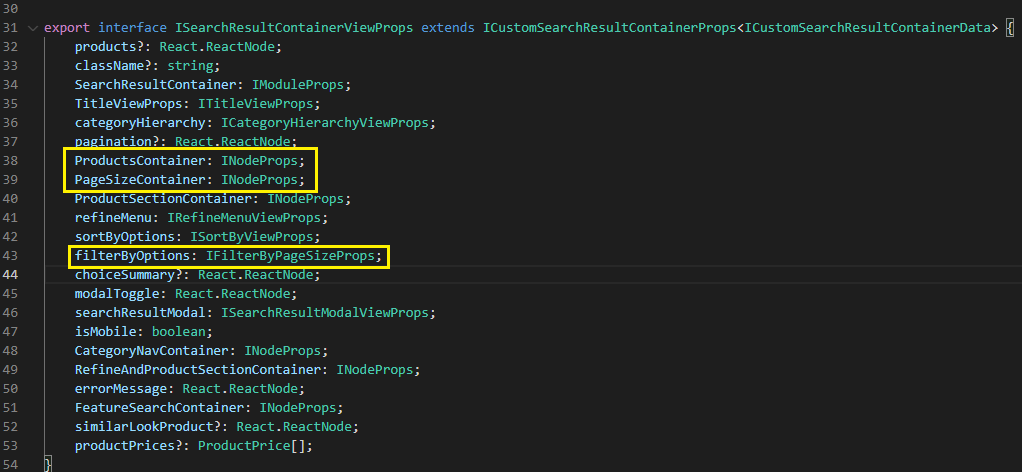
Modify code blocks to add filterByOptions and renderPageSize method as shown below

Timeline

Description automatically generated with medium confidence

* + - **search-result-container.tsx (business logic file)**

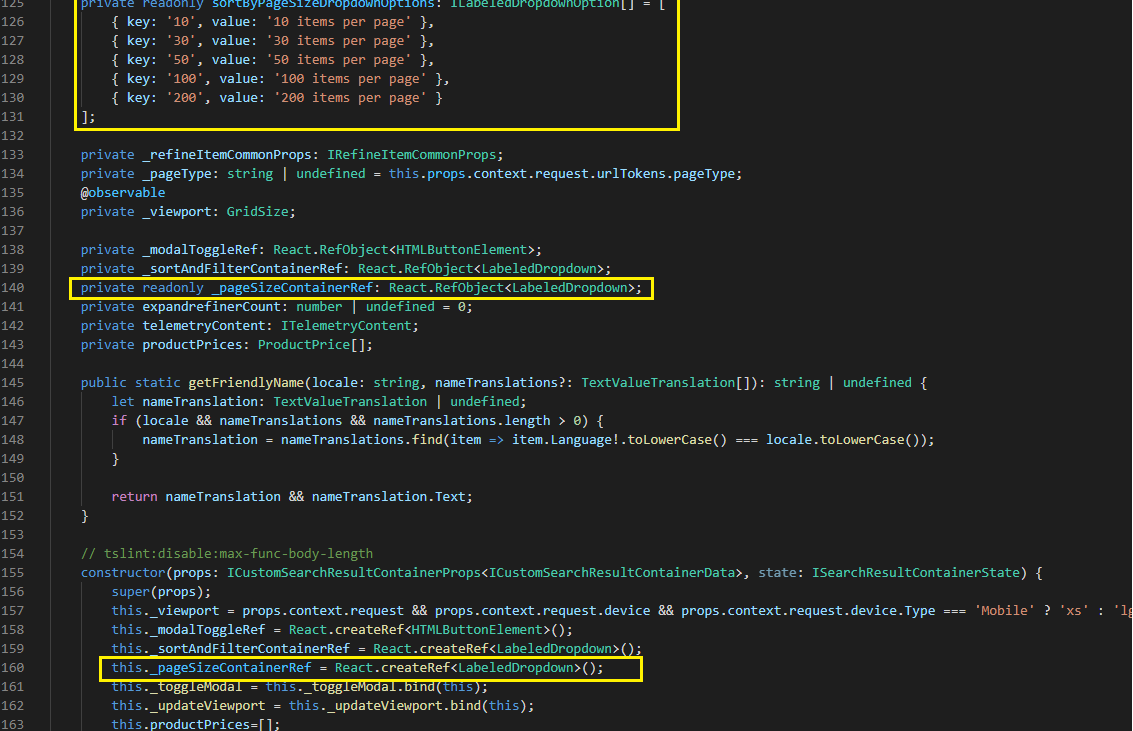
Add below highlighted code to tsx logic file. Refer sample code.



Add highlighted interface.

Text

Description automatically generated

****

Text

Description automatically generated

Add reaction for pagesize, refer code sample.

Text

Description automatically generated

Text

Description automatically generated

Add \_getPageSizeDropDown and modify related code method.

Text

Description automatically generated

Add below methods in code, refer sample code.

Graphical user interface

Description automatically generated

* + - **index.ts (Index file under module’s action folder)**

Add required interfaces and you may need to modify imports. Refer sample code file.

Text

Description automatically generated

* + - **get-full-products-by-collection.ts (Module action file)**

Modify below code block

A screenshot of a computer

Description automatically generated with medium confidence

1. **Build the repository**:

Execute yarn build from command line interface on your repository after making above code changes.

1. **Test the code changes:**

Execute yarn start.

Browse the configured product using pagemock using below sample url.

https://localhost:4000/partnersite/page?mock=testmock

**Note**: check for the right *partnersite* in .env file of your repository and make a change in url accordingly.

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

Graphical user interface, text

Description automatically generated