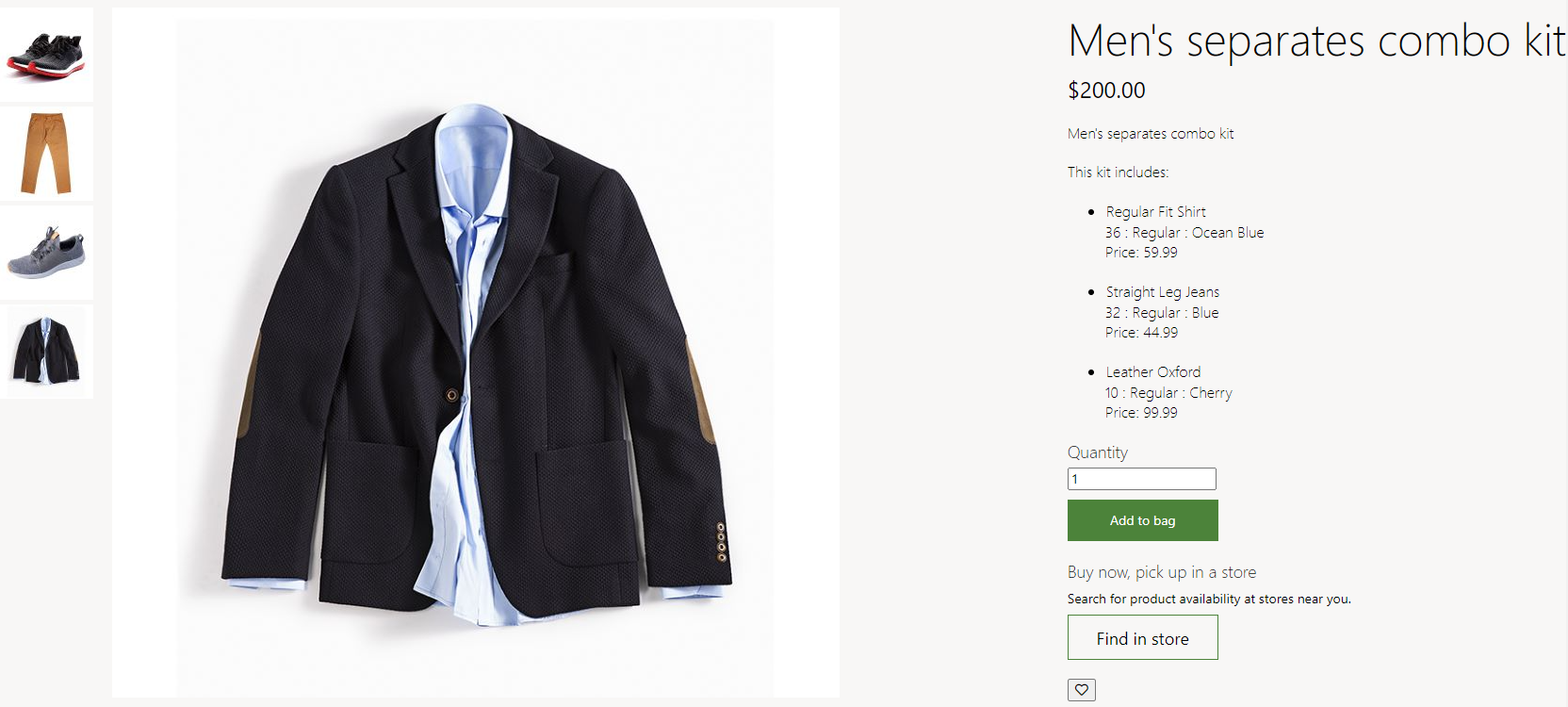
**Sample-Kit Product Customization**

# Objective

A sample kit product is a kind on bundle that comes up with multiple products in the product description page. It will have some discount on the purchase in comparison to buying those products individually. End user can see the discounted price and strikethrough price on UI based on the HQ configurations for the kit product. Below is the screenshot of product description page for the sample kit product.



# 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>

* Need to configure the sample kit product in HQ for this feature implementation.

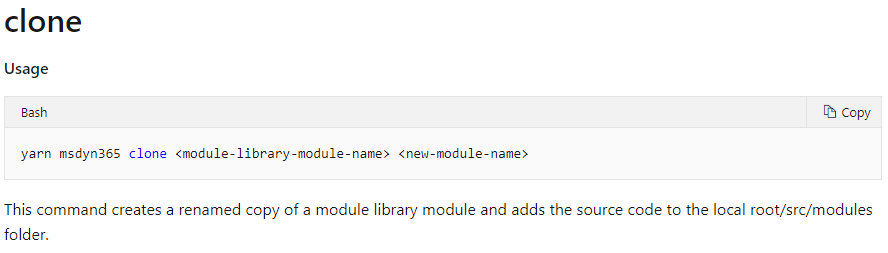
# Components Modified for this feature.

Cart Line-Item shared component need to extent to introduce business logic segregate rendering logic based on sample kit products and normal products. The same component is used on cart page, checkout page and order details page to render the sample kit product there as well/ Refer below code snippet.



# Modules Ejected for this feature.

Buybox module ejection is required 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 buybox 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=kit-demo>

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

**Option-2:** If the buybox 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 steps for the same.

1. **Component extension:** (*cartlineitem.component.tsx*): Check for this file under src/themes/”customer-theme”/views/components folder to make the necessary changes to this file.

**Note:** if the file does not exist on this path, execute below command to override the component. Change the highlighted text with the associated theme folder.

**CLI**: yarn msdyn365 add-component-override customer-theme cartlineitem.component

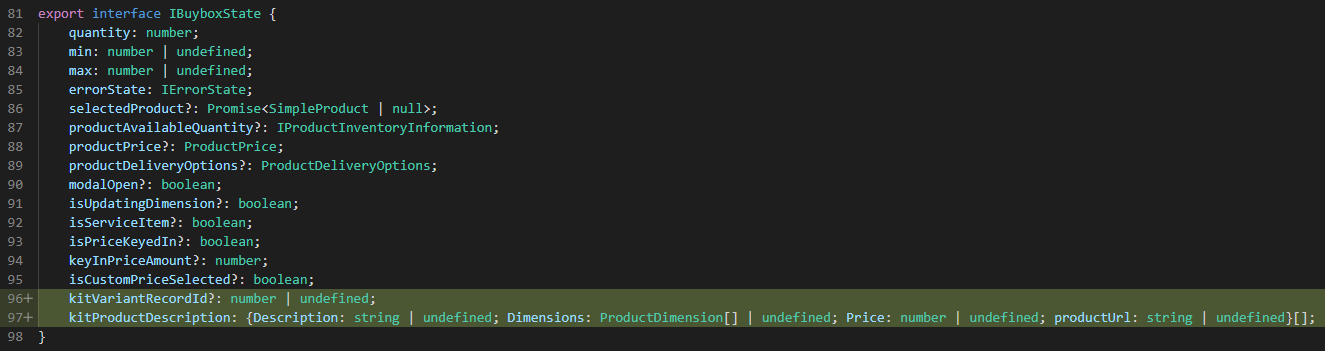
Once the component is added to the target location, add below code snippet form the sample code files.

Copy from line-117 to line 138 and add the same to *CartLineItemFunctions* of this file.



1. **Buybox module file changes:** Need to modify below files in buybox module for this feature implementation.
   * + **Buybox.tsx (business logic file)**

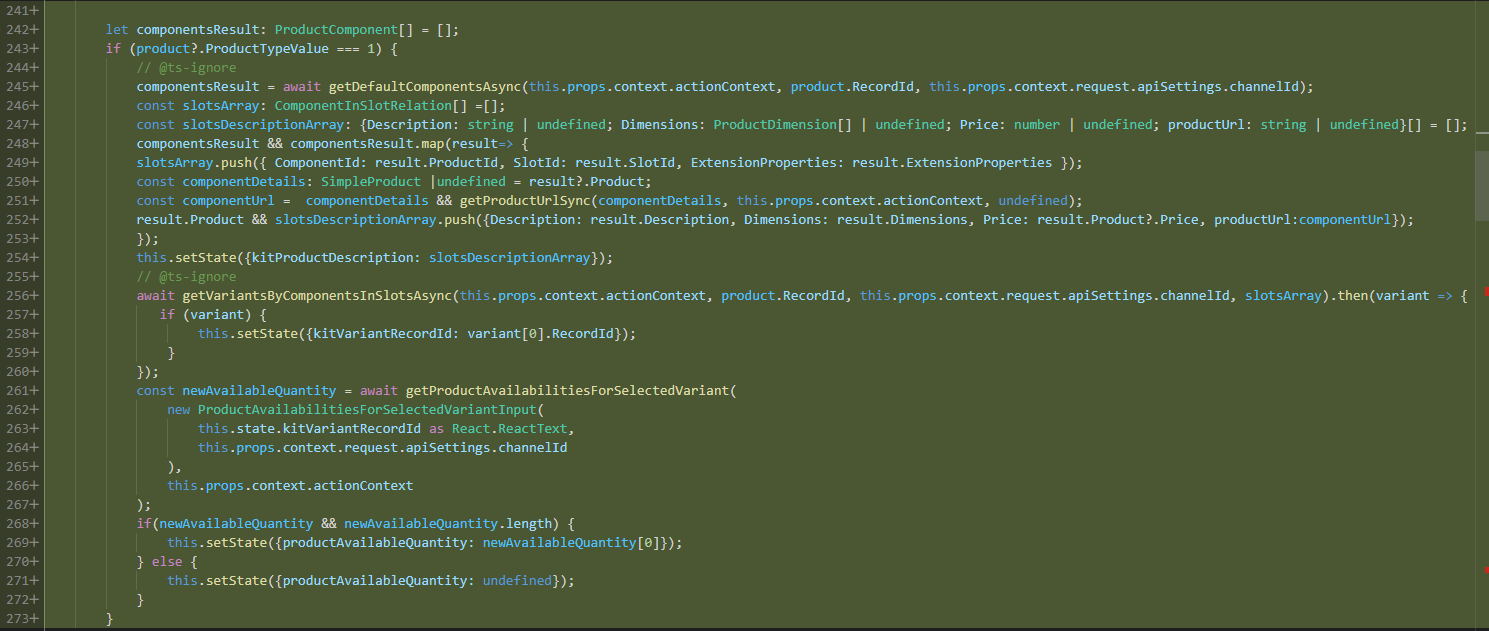
Add below highlighted props to the module state: Refer from sample code file.



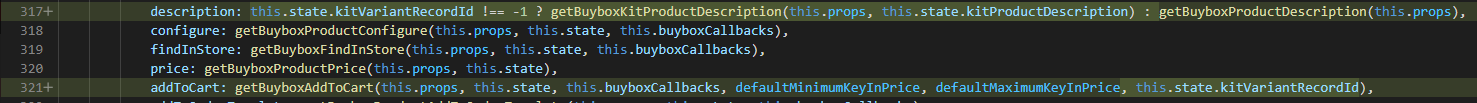
Initialize the same in constructor: Refer from sample code file.



In componentDidMount() function, add below block of code: Refer from sample code file.



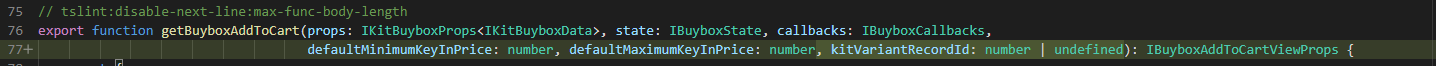
In render function modify below lines: Refer from sample code file.

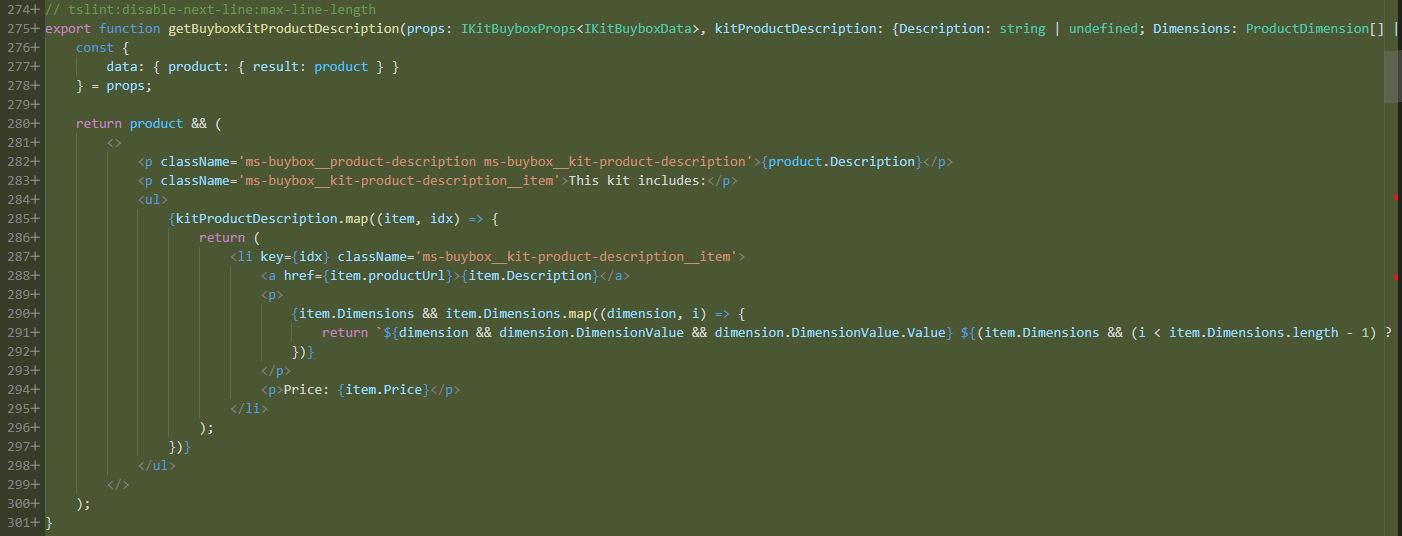


**Note**: You may need to update the imports for the missing references

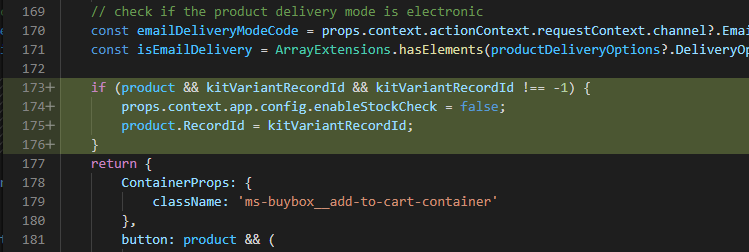
* + - **Buybox-small-components.tsx (internal component file)**

Modify the signature of below function to add the kit parameter.



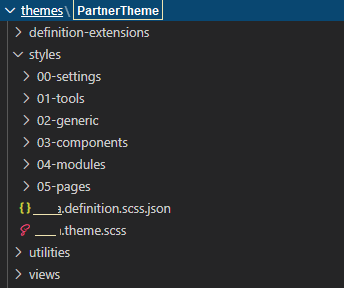
Add below function to this file: Refer from sample code file.

Add below condition to the same function: Refer from sample code file.



**Note**: You may need to update the imports for the missing references

1. **Implement styles:** copy the .scss file for the attached sample code and add it to the styles folder. Also make an entry of this file in *index.scss.* You can also integrate the same styles in *buybox.scss* and other classed where ever required. Below is the sample folder structure of theme folder.



**Note**: Here the name of theme folder depends on theme folder name in partner’s repository.

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=kit-demo>

**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:

* Store starter kit: 10.0.16 (“@msdyn365-commerce-modules/starter-pack": "~9.26.0)
* Retail-Proxy: 10.0.16 ("@msdyn365-commerce/retail-proxy": "~9.26.0")