Microsoft Dynamics D365 Retail Experts - https://www.meetup.com/Dynamics-User-Group-India/
Sheikhmydheen
https://www.linkedin.com/in/sheikhmydheen/
Sheikhmydheen@gmail.com , d365retail@gmail.com

Add a custom control to POS views:

You can enhance the information displayed on the Dynamics 365 for Retail POS views by adding custom controls. Custom control allows you to add your own custom information to the existing POS views. Custom controls can be implemented by using the POS extension framework. This topic is applicable for Dynamics 365 for Finance and Operations or Dynamics 365 for Retail platform update 8 with retail App update 4 hotfix. In this document we will focus on how to add custom control to non-screen layout designer based screens.

In the following example, we will show you how you can add custom control to one of the POS existing view using extension:

Scenario: We want to show the product availability information in the product details view by adding custom data list which has four columns (Location, Inventory, Reserved and Ordered). You can follow the same pattern to show any custom information you want to show in the POS views.

1. Open visual studio 2015 in administrator mode.
2. Open ModernPOS solution from …\RetailSDK\POS
3. Under the POS.Extensions project create a new folder called ProdDetailsCustomColumnExtensions.
4. Under ProdDetailsCustomColumnExtensions, create new folder called ViewExtensions.
5. Under ViewExtensions, create new folder called SimpleProductDetails.
6. Add new HTML inside the SimpleProductDetails folder and name it as ProductAvailabilityPanel.html

   In the HTML you can add whatever information you want to show in the custom control and in the .ts file add the corresponding logic. Custom control is nothing but simple HTML page with your custom information.

7. Open the ProductAvailabilityPanel.html and copy the below code:

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
       <meta charset="utf-8" />
       <title></title>
</head>
<body>
       <!-- Note: The element id is different than the id generated by the POS
       extensibility framework. This 'template' id is not used by the POS
       extensibility framework. -->
       <script id="Microsot_Pos_Extensibility_Samples_ProductAvailabilityPanel"
       type="text/html">
     <h2 class="marginTop8 marginBottom8" data-bind="text: title"></h2>
       <div class="width400 grow col">
       <div
       id="Microsot_Pos_Extensibility_Samples_ProductAvailabilityPanel_DataList"
       data-bind="msPosDataList: dataList"></div>
     </div>
```

```
    </script>
</body>
</html>
```

In the file, we just added POS data list control to show the product availability information and specified the width of the control.

8. In the SimpleProductDetails folder, add a new ts (typescript) file and name it has ProductAvailabilityPanel.ts
9. Add the below import statement to import the relevant entities and context.

```
import {
    SimpleProductDetailsCustomControlBase,
    ISimpleProductDetailsCustomControlState,
    ISimpleProductDetailsCustomControlContext
} from "PosApi/Extend/Views/SimpleProductDetailsView";
import { InventoryLookupOperationRequest, InventoryLookupOperationResponse } from
"PosApi/Consume/OrgUnits";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { ArrayExtensions } from "PosApi/TypeExtensions";
import { DataList, SelectionMode } from "PosUISdk/Controls/DataList";
```

10. Create a new class called `ProductAvailabilityPanel` and extend it from `SimpleProductDetailsCustomControlBase`.

```
export default class ProductAvailabilityPanel extends
SimpleProductDetailsCustomControlBase { }
```

11. Inside the class declare the below variables to state and data list information:

```
private static readonly TEMPLATE_ID: string =
"Microsot_Pos_Extensibility_Samples_ProductAvailabilityPanel";

    public readonly orgUnitAvailabilities:
ObservableArray<ProxyEntities.OrgUnitAvailability>;
    public readonly dataList: DataList<ProxyEntities.OrgUnitAvailability>;
    public readonly title: Observable<string>;
    private _state: ISimpleProductDetailsCustomControlState;
```

12. Create a class constructor method to initialize the data list columns:

```
constructor(id: string, context: ISimpleProductDetailsCustomControlContext) {
    super(id, context);

    this.orgUnitAvailabilities = ko.observableArray([]);
    this.title = ko.observable("Product Availability");
    this.dataList = new DataList<ProxyEntities.OrgUnitAvailability>({
        columns: [
            {
                title: "Location",
                ratio: 31,
                collapseOrder: 4,
                minWidth: 100,
                    computeValue: (value: ProxyEntities.OrgUnitAvailability):
string => {
```

Sheikhmydheen
https://www.linkedin.com/in/sheikhmydheen/
Sheikhmydheen@gmail.com , d365retail@gmail.com

```
                        return value.OrgUnitLocation.OrgUnitName;
                    }
                },
                {
                    title: "Inventory",
                    ratio: 23,
                    collapseOrder: 3,
                    minWidth: 60,
                        computeValue: (value: ProxyEntities.OrgUnitAvailability):
    string => {
                        return ArrayExtensions.hasElements(value.ItemAvailabilities) ?
    value.ItemAvailabilities[0].AvailableQuantity.toString() : "0";
                    }
                },
                {
                    title: "Reserved",
                    ratio: 23,
                    collapseOrder: 1,
                    minWidth: 60,
                        computeValue: (value: ProxyEntities.OrgUnitAvailability):
    string => {
                        return ArrayExtensions.hasElements(value.ItemAvailabilities) ?
    value.ItemAvailabilities[0].PhysicalReserved.toString() : "0";
                    }
                },
                {
                    title: "Ordered",
                    ratio: 23,
                    collapseOrder: 2,
                    minWidth: 60,
                        computeValue: (value: ProxyEntities.OrgUnitAvailability):
    string => {
                        return ArrayExtensions.hasElements(value.ItemAvailabilities) ?
    value.ItemAvailabilities[0].OrderedSum.toString() : "0";
                    }
                }
            ],
            itemDataSource: this.orgUnitAvailabilities,
            selectionMode: SelectionMode.None
        });
    }
```

13. Add the OnReady method to bind the Html control

```
    public onReady(element: HTMLElement): void {
     ko.applyBindingsToNode(element, {
         template: {
             name: ProductAvailabilityPanel.TEMPLATE_ID,
             data: this
         }
     });
        }
```

14. Add the init method to get the product availability details so that page loads the data is fetched and updated in the data list

Sheikhmydheen
https://www.linkedin.com/in/sheikhmydheen/
Sheikhmydheen@gmail.com , d365retail@gmail.com

```
public init(state: ISimpleProductDetailsCustomControlState): void {
this._state = state;
 let correlationId: string = this.context.logger.getNewCorrelationId();

 if (!this._state.isSelectionMode) {
     this.isVisible = true;
            let request:
InventoryLookupOperationRequest<InventoryLookupOperationResponse> =
         new InventoryLookupOperationRequest<InventoryLookupOperationResponse>
             (this._state.product.RecordId, correlationId);
     this.context.runtime.executeAsync(request)
                 .then((result:
ClientEntities.ICancelableDataResult<InventoryLookupOperationResponse>) => {
             if (!result.canceled) {
                 this.orgUnitAvailabilities(result.data.orgUnitAvailability);
             }
         }).catch((reason: any) => {
             this.context.logger.logError(JSON.stringify(reason), correlationId);
         });
 }
    }
```

The overall code should look like this:

```
/**
 * SAMPLE CODE NOTICE
 *
 * THIS SAMPLE CODE IS MADE AVAILABLE AS IS.  MICROSOFT MAKES NO WARRANTIES,
WHETHER EXPRESS OR IMPLIED,
 * OF FITNESS FOR A PARTICULAR PURPOSE, OF ACCURACY OR COMPLETENESS OF RESPONSES,
OF RESULTS, OR CONDITIONS OF MERCHANTABILITY.
 * THE ENTIRE RISK OF THE USE OR THE RESULTS FROM THE USE OF THIS SAMPLE CODE
REMAINS WITH THE USER.
 * NO TECHNICAL SUPPORT IS PROVIDED.  YOU MAY NOT DISTRIBUTE THIS CODE UNLESS YOU
HAVE A LICENSE AGREEMENT WITH MICROSOFT THAT ALLOWS YOU TO DO SO.
 */

import {
    SimpleProductDetailsCustomControlBase,
    ISimpleProductDetailsCustomControlState,
    ISimpleProductDetailsCustomControlContext
} from "PosApi/Extend/Views/SimpleProductDetailsView";
import { InventoryLookupOperationRequest, InventoryLookupOperationResponse } from
"PosApi/Consume/OrgUnits";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { ArrayExtensions } from "PosApi/TypeExtensions";
import { DataList, SelectionMode } from "PosUISdk/Controls/DataList";


export default class ProductAvailabilityPanel extends
SimpleProductDetailsCustomControlBase {
    private static readonly TEMPLATE_ID: string =
"Microsot_Pos_Extensibility_Samples_ProductAvailabilityPanel";
```

Sheikhmydheen
https://www.linkedin.com/in/sheikhmydheen/
Sheikhmydheen@gmail.com , d365retail@gmail.com

```
    public readonly orgUnitAvailabilities:
ObservableArray<ProxyEntities.OrgUnitAvailability>;
    public readonly dataList: DataList<ProxyEntities.OrgUnitAvailability>;
    public readonly title: Observable<string>;

    private _state: ISimpleProductDetailsCustomControlState;



    constructor(id: string, context: ISimpleProductDetailsCustomControlContext) {
        super(id, context);

        this.orgUnitAvailabilities = ko.observableArray([]);
        this.title = ko.observable("Product Availability");
        this.dataList = new DataList<ProxyEntities.OrgUnitAvailability>({
            columns: [
                {
                    title: "Location",
                    ratio: 31,
                    collapseOrder: 4,
                    minWidth: 100,
                    computeValue: (value: ProxyEntities.OrgUnitAvailability):
string => {
                        return value.OrgUnitLocation.OrgUnitName;
                    }
                },
                {
                    title: "Inventory",
                    ratio: 23,
                    collapseOrder: 3,
                    minWidth: 60,
                    computeValue: (value: ProxyEntities.OrgUnitAvailability):
string => {
                        return
ArrayExtensions.hasElements(value.ItemAvailabilities) ?
value.ItemAvailabilities[0].AvailableQuantity.toString() : "0";
                    }
                },
                {
                    title: "Reserved",
                    ratio: 23,
                    collapseOrder: 1,
                    minWidth: 60,
                    computeValue: (value: ProxyEntities.OrgUnitAvailability):
string => {
                        return
ArrayExtensions.hasElements(value.ItemAvailabilities) ?
value.ItemAvailabilities[0].PhysicalReserved.toString() : "0";
                    }
                },
                {
                    title: "Ordered",
                    ratio: 23,
                    collapseOrder: 2,
                    minWidth: 60,
```

Sheikhmydheen
https://www.linkedin.com/in/sheikhmydheen/
Sheikhmydheen@gmail.com , d365retail@gmail.com

```
                        computeValue: (value: ProxyEntities.OrgUnitAvailability):
string => {
                            return
ArrayExtensions.hasElements(value.ItemAvailabilities) ?
value.ItemAvailabilities[0].OrderedSum.toString() : "0";
                        }
                    }
                ],
                itemDataSource: this.orgUnitAvailabilities,
                selectionMode: SelectionMode.None
            });
    }

    /**
     * Binds the control to the specified element.
     * @param {HTMLElement} element The element to which the control should be
bound.
     */
    public onReady(element: HTMLElement): void {
        ko.applyBindingsToNode(element, {
            template: {
                name: ProductAvailabilityPanel.TEMPLATE_ID,
                data: this
            }
        });
    }

    /**
     * Initializes the control.
     * @param {ISimpleProductDetailsCustomControlState} state The initial state of
the page used to initialize the control.
     */
    public init(state: ISimpleProductDetailsCustomControlState): void {
        this._state = state;
        let correlationId: string = this.context.logger.getNewCorrelationId();

        if (!this._state.isSelectionMode) {
            this.isVisible = true;
            let request:
InventoryLookupOperationRequest<InventoryLookupOperationResponse> =
                new
InventoryLookupOperationRequest<InventoryLookupOperationResponse>
                    (this._state.product.RecordId, correlationId);
            this.context.runtime.executeAsync(request)
                .then((result:
ClientEntities.ICancelableDataResult<InventoryLookupOperationResponse>) => {
                    if (!result.canceled) {

this.orgUnitAvailabilities(result.data.orgUnitAvailability);
                    }
                }).catch((reason: any) => {
                    this.context.logger.logError(JSON.stringify(reason),
correlationId);
                });
        }
```

Sheikhmydheen
https://www.linkedin.com/in/sheikhmydheen/
Sheikhmydheen@gmail.com , d365retail@gmail.com

```
        }
           }
```

15. Create a new json file and under the ProdDetailsCustomColumnExtensions folder and name it as manifest.json.
16. In the manifest.json file, copy and paste the below code:

```
{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_Samples",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
      "extend": {
      "views": {
        "SimpleProductDetailsView": {
          "controlsConfig": {
            "customControls": [
              {
                "controlName": "productAvailabilityPanel",
                "htmlPath":
"ViewExtensions/SimpleProductDetails/ProductAvailabilityPanel.html",
                "modulePath":
"ViewExtensions/SimpleProductDetails/ProductAvailabilityPanel"
              }
            ]
          }
        }
      }
    }
  }
}
```

17. Open the extensions.json file under POS.Extensions project and update it with ProdDetailsCustomColumnExtensions samples, so that POS during runtime will include this extension.

```
{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions2"
    },
    {
      "baseUrl": "POSAPIExtension"
    },
    {
      "baseUrl": "CustomColumnExtensions"
    },
    {
      "baseUrl": "EODSample"
    },
    {
```

Sheikhmydheen
https://www.linkedin.com/in/sheikhmydheen/
Sheikhmydheen@gmail.com , d365retail@gmail.com

```
            "baseUrl": "ProdDetailsCustomColumnExtensions"
        }
    ]
        }
```

18. Open the tsconfig.json to comment out the extension package folders from the exclude list. POS will use this file to include or exclude the extension. By default, the list contains all the excluded extensions list, if you want to include any extension part of the POS then you need add the extension folder name and comment the extension from the extension list like below.

```
"exclude": [
    "AuditEventExtensionSample",
    "B2BSample",
    "CustomerSearchWithAttributesSample",
    "FiscalRegisterSample",
    "PaymentSample",
    "PromotionsSample",
    "SalesTransactionSignatureSample",
    "SampleExtensions",
    //"SampleExtensions2",
    "StoreHoursSample",
    "SuspendTransactionReceiptSample"
    //"POSAPIExtension",
    //"CustomColumnExtensions",
    //"EODSample",
    //"ProdDetailsCustomColumnExtensions"
        ],
```

19. Compile and rebuild the project.

Validate the customization:

How to test your extension:

1. Press F5 and deploy the POS to test your customization.
2. Once the POS is launched, login to POS and then search for any product and navigate to product details screen, then you should see the custom control you added something like below:

Microsoft Dynamics D365 Retail Experts - https://www.meetup.com/Dynamics-User-Group-India/
Sheikhmydheen
https://www.linkedin.com/in/sheikhmydheen/
Sheikhmydheen@gmail.com , d365retail@gmail.com