

Custom List Views

Creating a custom list view to make it easier for content editors
to view product images and prices

Out of the box

This screenshot shows the 'Products' list view in a CMS. The left sidebar contains a navigation menu with 'Home' and 'Products' (selected). The main content area displays a table of products with columns for Name, Status, Last edited, and Created by. A 'Create Product' button is visible at the top left of the table.

<input checked="" type="checkbox"/>	Name	Status	Last edited	Created by
	Tattoo	Published	2021-12-03 21:35	David Sheiles
	Biker Jacket	Published	2021-12-03 20:49	David Sheiles
	Knitted West	Published	2021-12-03 20:49	David Sheiles
	Banjo	Published	2021-12-03 20:49	David Sheiles
	Jumpsuit	Published	2021-12-03 20:49	David Sheiles
	Bowling Ball	Published	2021-12-03 20:49	David Sheiles
	Ping Pong Ball	Published	2021-12-03 20:49	David Sheiles
	Unicorn	Published	2021-12-03 20:49	David Sheiles

At the bottom of the interface, there are buttons for 'Save and preview', 'Save', and 'Save and publish'.

This screenshot shows the 'Products' detail view in a CMS. The left sidebar contains a navigation menu with 'Home' and 'Products' (selected). The main content area displays a grid of product details for 'Tattoo', 'Biker Jacket', 'Knitted West', 'Banjo', 'Jumpsuit', and 'Bowling Ball'. Each product card shows its status, last edited time, and created by information. A 'Create Product' button is visible at the top left of the grid.

At the bottom of the interface, there are buttons for 'Save and preview', 'Save', and 'Save and publish'.

Our Custom View

Content

Media

Settings

Packages

Users

Members

Forms

Translation

Content

Home

Products

People

About Us

Blog

Contact

Recycle Bin

Products


Child items

Content

Info


Actions

Create Product




Tattoo.

\$499.00



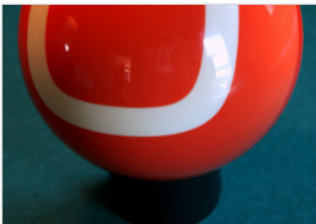
Unicorn

\$249.00



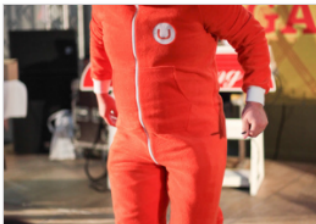
Ping Pong Ball

\$2.00




Bowling Ball

\$899.00



Jumpsuit

\$89.00



Banjo

\$399.00

Creating The Components

Package Manifest

```
/App_Plugins/DM.ListViews/package.manifest
```

```
{  
  "css": [ "~ / App_Plugins / DM.ListViews / Products / Products.css" ],  
  "javascript": [ "~ / App_Plugins / DM.ListViews / Products / Products.controller.js" ]  
}
```

HTML View

```
/App_Plugins/DM.ListViews/Products/Products.html
```

```
<div ng-controller="DM.Listview.ProductsController as vm">
  <div ng-repeat="item in items" ng-class="{ '-selected': item.selected}">
    <div class="thumbnail"
      style="background-image: url('{{item.photos[0].src}}?width=300"
      ng-click="vm.toggleItem(item)"></div>
    <h5 ng-click="vm.goToItem(item)">
      <umb-icon icon="icon-tag"></umb-icon>
      {{item.productName || item.name}}
    </h5>
    <p>{{item.price | currency}}</p>
  </div>
</div>
```

JS Controller View

```
/App_Plugins/DM.ListViews/Products/Products.controller.js
```

1. Create the Controller

```
function ListViewProductsController($scope, listViewHelper, mediaResource) {  
    var vm = this;  
}  
  
angular.module("umbraco").controller("DM.Listview.ProductsController",  
    ListViewProductsController);
```

JS Controller View

2. Media URLs

```
//Loop through each media item and get the image URL
angular.forEach($scope.items, function (value, key) {
    if (value.photos.length > 0) {
        updateMediaUrl(value.photos[0]);
    }
});

//Gets the media Url by the mediaKey and adds it to our object
function updateMediaUrl(photoObj) {
    mediaResource.getById(photoObj.mediaKey)
        .then(function (media) {
            photoObj.src = media.mediaLink;
        });
}
```

JS Controller View

3. Click the thumbnail to select / deselect the item

```
//Select / Unselect Item  
vm.toggleItem = function (item) {  
    if (item.selected) {  
        listViewHelper.deselectItem(item, $scope.selection);  
    } else {  
        listViewHelper.selectItem(item, $scope.selection);  
    }  
}
```


JS Controller View

4. Click the name to open / edit the item

```
//Open Item to edit  
vm.goToItem = function(item, $event, $index) {  
    listViewHelper.editItem(item, $scope);  
}
```

Bringing it all together


Columns Displayed
The properties that will be displayed for each column


Add

	Alias	Header	Template	
+	sortOrder <small>(system field)</small>	Sort		Remove
+	updateDate <small>(system field)</small>	Last edited		Remove
+	owner <small>(system field)</small>	Created by		Remove
+	productName	<input type="text" value="Product Name"/>	<input type="text"/>	Remove
+	price	<input type="text" value="Price"/>	<input type="text"/>	Remove
+	photos	<input type="text" value="Photos"/>	<input type="text"/>	Remove


Layouts

+

 Thumbnails




+

 List (system layout)

☒

+

 Grid (system layout)

☒

Add layout

Thanks!

- Component Examples <https://bit.ly/umb-components>
- Umbraco TV - <https://bit.ly/tv-listview>
- Demo Project - <https://bit.ly/DemoProj>
- Download this presentation -