



Advanced UI Patterns

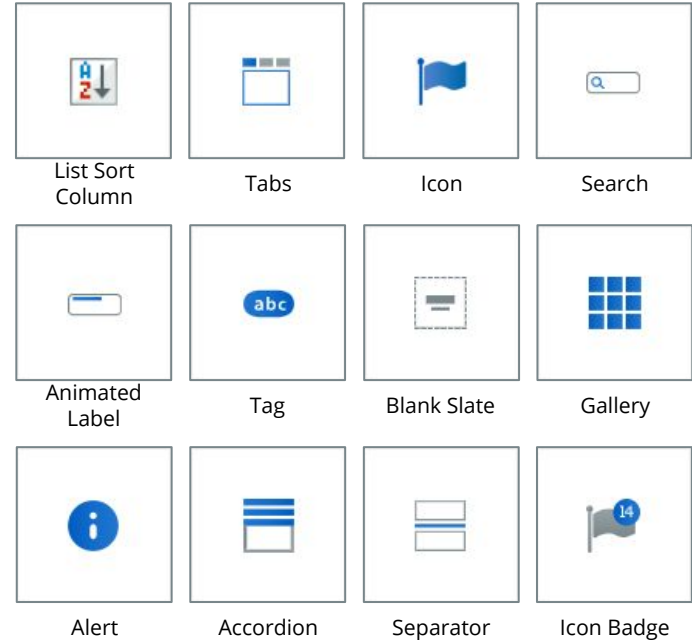


Topics

- OutSystems UI
 - UI Patterns
- RichWidgets
 - List_Navigation
 - List_SortColumn

Advanced UI Patterns

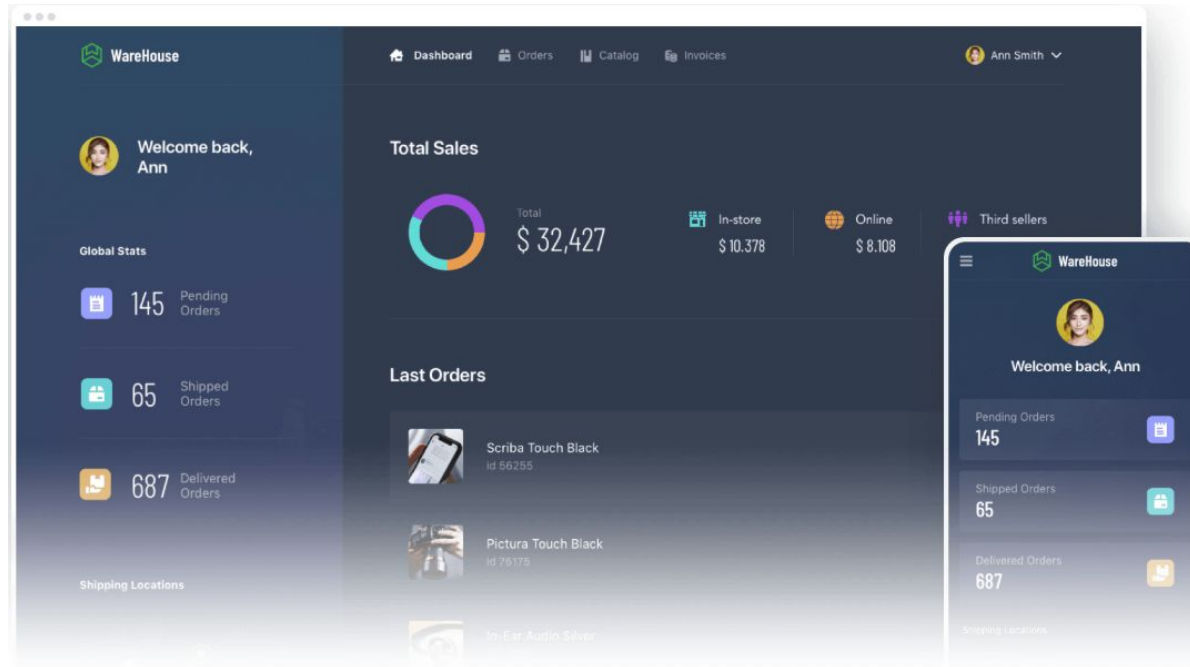
- OutSystems UI
 - UI Framework for web and mobile apps
 - 100+ UI Patterns and Screens
- Rich Widgets
 - Groups of Widgets, Themes and Actions
- Widgets are present in the Service Studio Toolbar



OutSystems UI



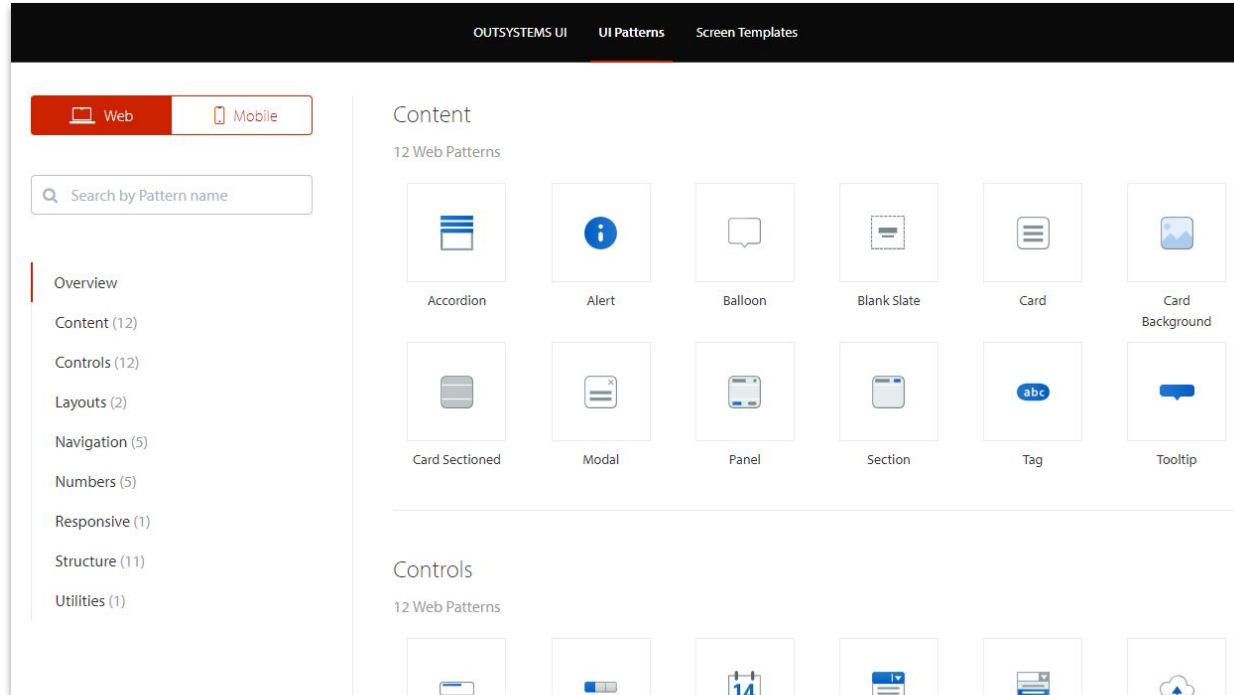
OutSystems UI



<https://www.outsystems.com/outsystems-ui/>

UI Patterns


Pre-built UI patterns that can be easily placed in Screens (drag & drop)





UI Patterns Documentation


There's full documentation to support the usage of the OutSystems UI Patterns


Start Building With the UI Patterns


Beautiful Design


Right-to-Left Language Support


Accessibility Friendly


Extensible with JS or CSS


Responsive Native Experience

[Go to Documentation](#)

outsystems

Search OutSystems 11 docs...

Developing an Application

New in OutSystems 11

- + Getting Started
- Developing an Application
 - Application Templates
 - + Use Data
 - Design UI
 - + Screen
 - + Inputs
 - + Forms
 - + Images
 - + Look and Feel
 - + Reuse UI
 - + **Patterns**
 - + Navigation

Patterns

[Save as PDF](#)

In OutSystems you can use the pre-built patterns, which implement common UI patterns and components. Patterns cover the most used UI patterns and interactions in mobile and web apps, to distance developers as much as possible from the need to program CSS.

Customization

You can customize the patterns by writing additional CSS that changes their look and feel. Additionally, you can change the structure to be able to display something else or remove unwanted elements. You can even change the code and the pattern behavior like adding swipe gestures or adding new events. Keep in mind that code customizations can make patterns behave unpredictably and are not officially supported.

By creating additional [CSS](#) on a new module, it is possible to override the look of most patterns, so this is the simplest way of customizing them.

Rich Widgets



Rich Widgets

Additional Widgets for common functionalities in Web applications



Examples of RichWidgets patterns

- DropDownMenu
- Icon
- List_Navigation
- List_SortColumn

The screenshot shows an 'Employees' web application interface. At the top, there is a search bar with 'Search' and 'Reset' buttons. Below is a table with columns 'Name' and 'Contact'. The 'Name' column header has a sort icon (two arrows) highlighted with a red box. The table contains three rows of data, all showing 'Vivian J. Martin' and '+1 520-234-9295'. At the bottom of the table, it says '51 to 100 of 721 records'. To the right of this text is a pagination control with buttons for '<', '...', '8', '9', '10', '11', '12', and '>'. The '12' button is highlighted with a red box. A red line connects the sort icon in the 'Name' column header to the 'List_SortColumn' widget in the RichWidgets palette. Another red line connects the '12' button in the pagination control to the 'List_Navigation' widget in the RichWidgets palette. The RichWidgets palette is located at the bottom right and contains a list of widgets: RichWidgets, DropDownMenu, Icon, List_Counter, List_Navigation, and List_SortColumn.

Employees

Search Reset

Name 	Contact 
Vivian J. Martin	+1 520-234-9295
Vivian J. Martin	+1 520-234-9295
Vivian J. Martin	+1 520-234-9295

51 to 100 of 721 records

< ... 8 9 10 11 12 >

RichWidgets

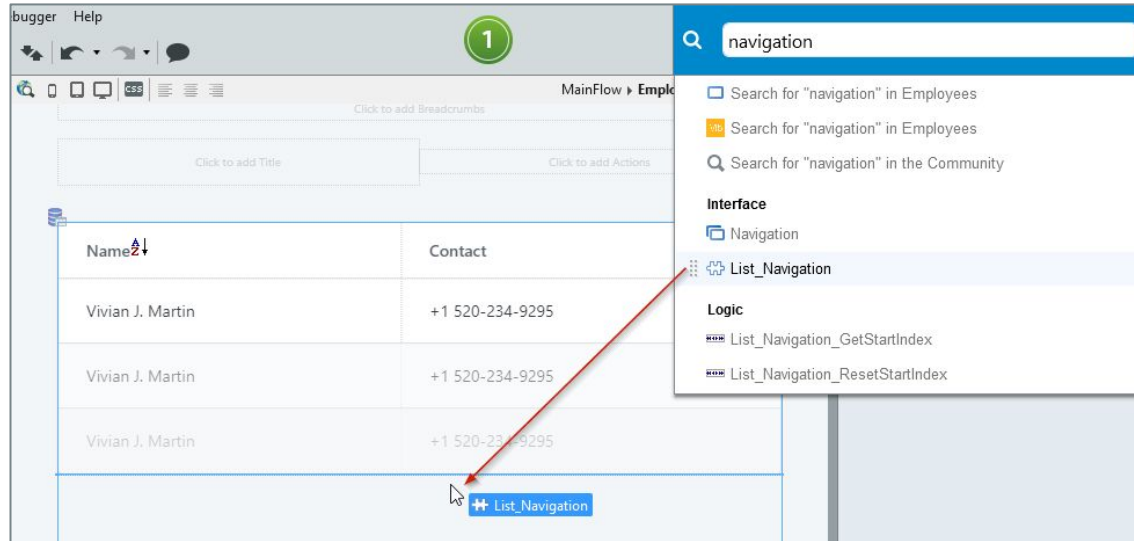
- RichWidgets
 - DropDownMenu
 - Icon
 - List_Counter
 - List_Navigation
 - List_SortColumn

List_Navigation

Navigator for a Table Records with multiple pages




- Shows a defined number of elements per page

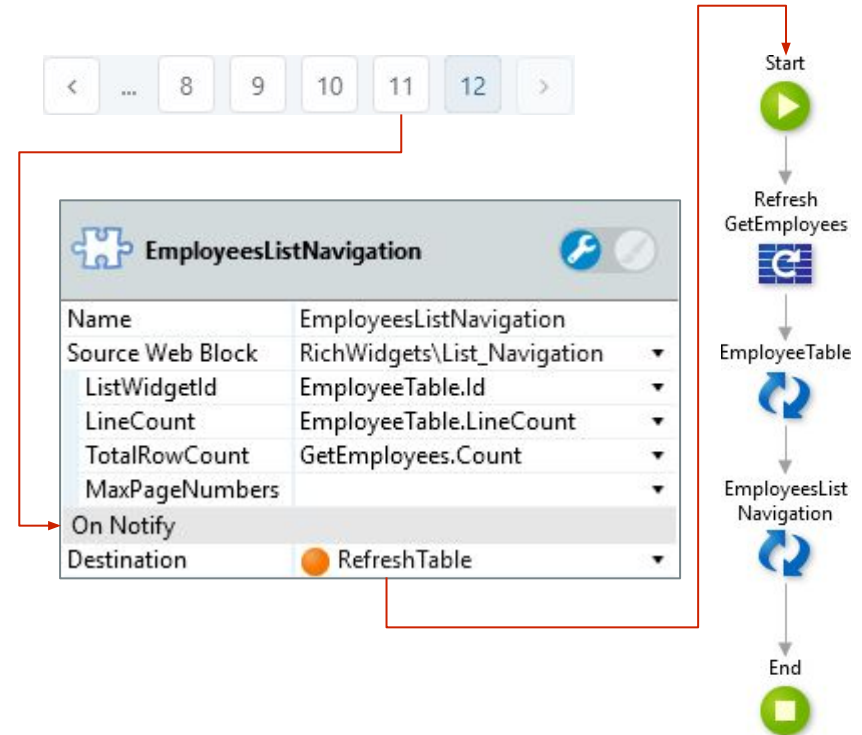
Drag and drop the **List_Navigation** Widget below the Table Records



List_Navigation properties

Set the List_Navigation properties

- **ListWidgetId:** Associate it with the Table Records
- **LineCount:** Number of records displayed per page
- **TotalRowCount:** Total number of records returned by the query
- **OnNotify:** Action to execute when a page is selected
 - Refresh the  Data Source and the  Table Records +  List Navigation



List_Navigation in the Table Records

Set the **Start Index** to depend on the List_Navigation

- The displayed elements are influenced by the page clicked on the navigator
- **List_Navigation_GetStartIndex**(<TableRecordsName>.Id)

Function retrieves from session the page clicked and calculates the starting index

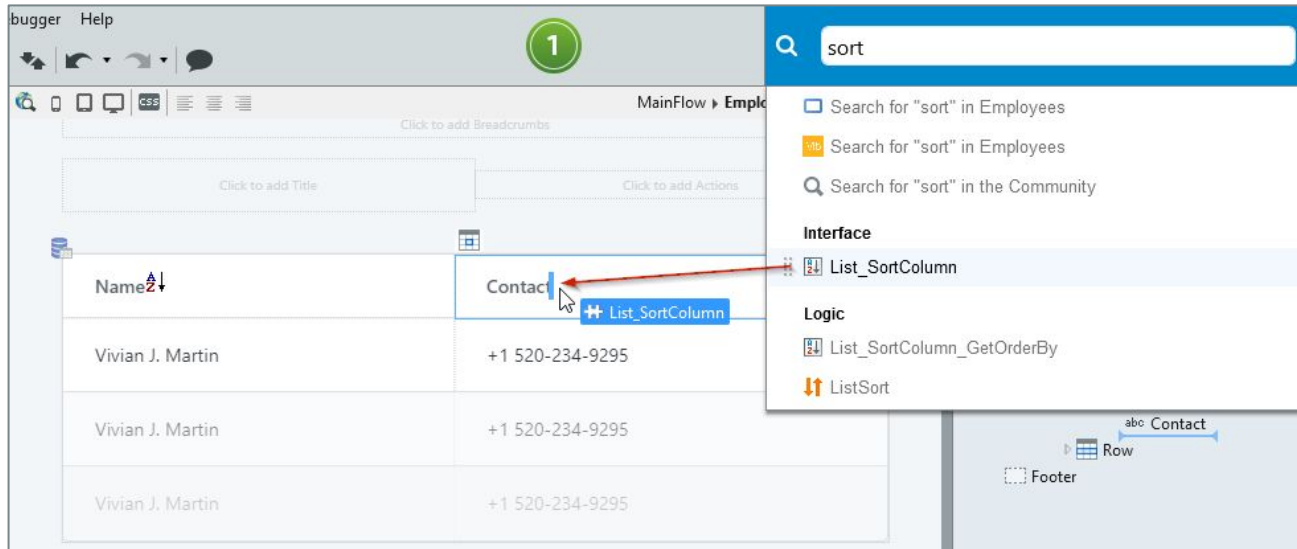
- Depends on the Line Count of the Table Records

EmployeeTable	
Name	EmployeeTable
Source Record...	GetEmployees.List
Empty Message	"No employees to show..."
Line Count	10
Start Index	List_Navigation_GetStartIndex(EmployeeTable.Id)
Cell Height	
Cell Spacing	
Show Header	Yes
Style Classes	TableRecords
Header Style	TableRecords_Header
Odd Line Style	TableRecords_OddLine
Even Line Style	TableRecords_EvenLine

List_SortColumn Widget

Defines dynamic sorting in Table Records columns

Drag and drop the **List_SortColumn** Widget next to the header to be sorted






List_SortColumn properties

Set the **Column** to

"{EntityName}.[AttributeName]"

Set the **OnNotify** destination to a Screen Action

- Action triggered when clicking on the Table Records header
- Refresh the Table Records' data source  and the Table Records widget 

RichWidgets\List_SortColumn	
Name	
Source Web Block	RichWidgets\List_SortColumn
Column	"{Employee}.[Contact]"
On Notify	
Destination	 RefreshTable

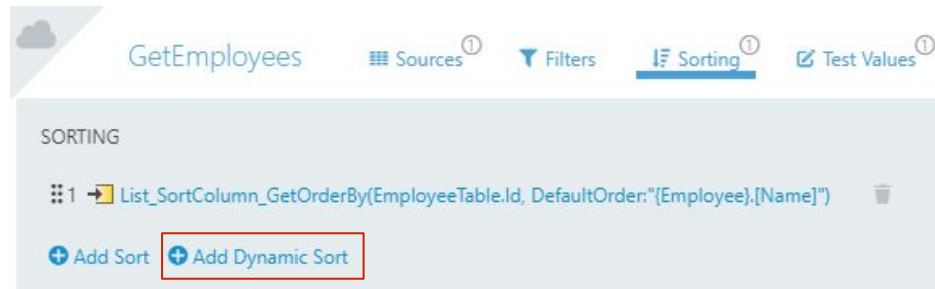
List_SortColumn input variable in the Aggregate

In the Aggregate used as source of the Table Records:

- Add a dynamic sort
- Set the Expression to:

List_SortColumn_GetOrderBy(<TableRecordsName>.Id, DefaultOrder:
"{EntityName}.[AttributeName]")

This function retrieves the column to sort by from the session



Summary

- OutSystems UI
 - UI Patterns
- RichWidgets
 - List_Navigation
 - List_SortColumn

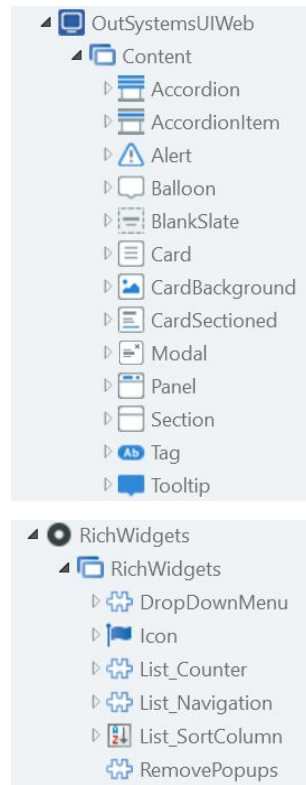
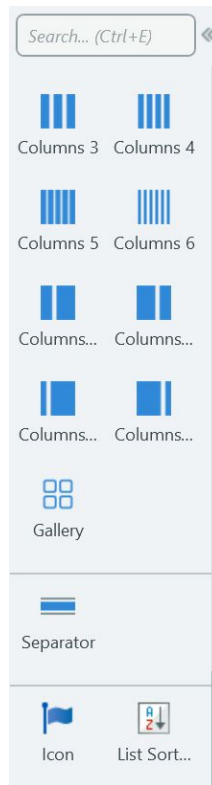


Advanced UI Patterns

Thank You!

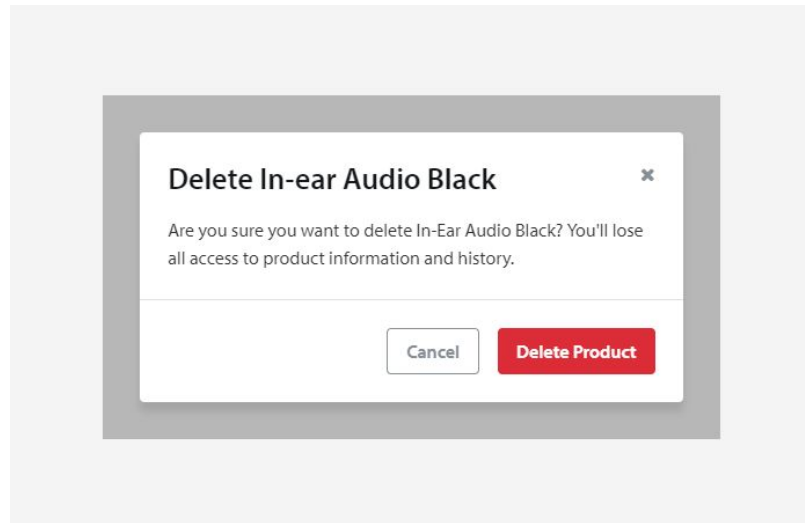
Advanced UI Patterns

- OutSystems UI
 - UI Framework for web and mobile apps
 - 100+ UI Patterns and Screens
- Rich Widgets
 - Groups of Widgets, Themes and Actions
- Widgets are present in the Service Studio Toolbar



Modal

- Similar to a Popup
- Interrupts the end-user to demand an Action
 - Important when a confirmation is required
 - Relevant when manipulating critical information
- Consists on
 - Modal Pattern
 - ToggleModal Server Action



Using the Modal

- Drag the Model pattern to your Screen and edit it to your needs
- Toggle Model Action opens / closes the Modal window
 - e.g. if Modal is open, the Toggle closes it
 - To be used in a Preparation or Screen Action
 - WidgetId expects the Id of the Modal
 - Modal must have a name

