

■ Obsidian Grid

A few details about our flagship grid





12

1

TIMEX

2

3

4

QUARTZ

5

6



What is the Grid?

UI Control

One of the most powerful controls in the Rock Obsidian UI Toolkit

Person List							6 People	🔍	+	⋮
<input type="checkbox"/>	Name	Birth Date	Age	Email Active	Campus	Connection Status				
<input type="checkbox"/>	Alisha Admin				Main Campus	Visitor				
<input type="checkbox"/>	Giver Anonymous			✓	Main Campus	Participant				
<input type="checkbox"/>	Presence Presence			✓	Main Campus	Visitor				
<input type="checkbox"/>	Ted Decker	2/10/1980	44	✓	Main Campus	Member				
<input type="checkbox"/>	Cindy Decker	3/10/1981	43	✓	Main Campus	Member				
<input type="checkbox"/>	Noah Decker	3/10/2008	16	✓	Main Campus	Attendee				

50 ▾



The screenshot displays the Obsidian Control Gallery interface. On the left is a sidebar with a list of controls, including 'Entity Picker' which is highlighted. The main area shows a preview of the 'Entity Picker' control. It features a 'Test Control' section with two dropdown menus: 'Entity Type' (set to 'Campus') and 'Campus' (set to 'Main Campus'). Below this is a 'Demo Settings' section with three checkboxes: 'Disabled' (unchecked), 'Hide Entity Type Picker' (checked), and 'Entity Control Help Text Template' (checked). The help text template is shown in a text box as 'This is the help text. Single Entity: {0}; Multip'. To the right of the preview is a 'Current Values' section displaying JSON data for 'value' and 'entityType'.

Obsidian Control Gallery

- Component Picker
- Connection Request Picker
- Content Channel Item Picker
- Content Drop Down Picker
- Copy Button
- Countdown Timer
- Currency Box
- Data View Picker
- Date Parts Picker
- Date Picker
- Date Range Picker
- Date Time Picker
- Day Of Week Picker
- Defined Value Picker
- Dialog
- Drop Down Content
- Drop Down List
- Drop Down Menu
- Electronic Signature
- Email Box
- Entity Picker**
- Entity Type Picker
- Ethnicity Picker
- Event Calendar Picker
- Event Item Picker
- Field Filter Editor
- Field Type Editor
- Field Type Picker
- File Uploader

Entity Picker

Test Control

Entity Type •
Campus

Campus ⓘ •
Main Campus

Demo Settings

Disabled

Hide Entity Type Picker ⓘ

Entity Control Help Text Template ⓘ
This is the help text. Single Entity: {0}; Multip

Current Values

```
value
"76882ae3-1ce8-42a6-a2b6-8c0b29cf8cf"

entityType
{
  "value": "00096bed-9587-415e-8ad",
  "text": "Campus",
  "category": "All Entities",
  "disabled": null
}
```

UI Toolkit

Rock Control Gallery

This Obsidian block shows most all of the UI controls in the Obsidian arsenal



Main Zone



Page

Layout (Full Width)

Site

Add Page Block

Name *

Obsidian Gallery List

Type

Obsidian Gallery List 🇬🇧



Common Block Types

Content Channel Item View

Content Channel View

Content Component

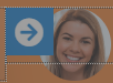
HTML Content

Page Menu

Save

Cancel

Done





Features

- Paging
- Filtering
- Sorting
- Grid Actions
- Exporting
- and much more



Grid: Four Parts

**Grid
State**

Internally Managed

This is an internal piece that you need not worry about.

**Grid
(Shell)**

**Definition &
Properties**

Handles displaying the rows, pagination, filters, etc. and handling grid actions

**Column
Definitions**

(exactly that)

**Row
Data**

(same)























Grid Definition

Grid :definition=...

```
1 <Grid :definition="config.gridDefinition ?? undefined"  
2   :data="gridDataSource"  
3   keyField="guid"  
4   personKeyField="guid"  
5   itemTerm="Person"  
6   tooltipField="email"  
7   stickyHeader  
8   liveUpdates  
9   :entityTypeGuid="EntityType.Person"  
10  :expectedRowCount="config.expectedRowCount ?? undefined"  
11  :customActions="customActions"  
12  :onAddItem="config.isAddEnabled ? onAddItem : undefined"  
13  @selectItem="onSelectItem">
```



Person List							57 of 59 People	🔍	+	⋮
<input type="checkbox"/>	Name	Birth Date	Age	Email Active	Campus	Connection Status				
<input type="checkbox"/>	 Giver Anonymous			✓	Main Campus	Participant				
<input type="checkbox"/>	 Presence Presence			✓	Main Campus	Visitor				
<input type="checkbox"/>	 Ted Decker	2/10/1986	38	✓	Main Campus	Member				
<input type="checkbox"/>	 Cindy Decker	3/10/1987	37	✓	Main Campus	Member				
<input type="checkbox"/>	 Noah Decker	3/10/2014	10	✓	Main Campus	Attendee				



Grid :definition=...

Grid Definition

```
1 <Grid :definition="config.gridDefinition ?? undefined"  
2   :data="gridDataSource"  
3   keyField="guid"  
4   personKeyField="guid"  
5   itemTerm="Person"  
6   tooltipField="email"  
7   stickyHeader  
8   liveUpdates  
9   :entityTypeGuid="EntityType.Person"  
10  :expectedRowCount="config.expectedRowCount ?? undefined"  
11  :customActions="customActions"  
12  :onAddItem="config.isAddEnabled ? onAddItem : undefined"  
13  @selectItem="onSelectItem">
```

Person List 57 of 59 People 🔍 + ⋮

<input type="checkbox"/>	Name	Birth Date	Age	Email Active	Campus	Connection Status			
<input type="checkbox"/>	Giver Anonymous			✓	Main Campus	Participant			
<input type="checkbox"/>	Presence Presence			✓	Main Campus	Visitor			
<input type="checkbox"/>	Ted Decker	2/10/1986	38	✓ ted@rocksolidchurchdemo.com	Main Campus	Member			
<input type="checkbox"/>	Cindy Decker	3/10/1987	37	✓	Main Campus	Member			
<input type="checkbox"/>	Noah Decker	3/10/2014	10	✓	Main Campus	Attendee			



Grid Definition

Grid :definition=...

```
1 <Grid :definition="config.gridDefinition ?? undefined"  
2   :data="gridDataSource"  
3   keyField="guid"  
4   personKeyField="guid"  
5   itemTerm="Person"  
6   tooltipField="email"  
7   stickyHeader  
8   liveUpdates  
9   :entityTypeGuid="EntityType.Person"  
10  :expectedRowCount="config.expectedRowCount ?? undefined"  
11  :customActions="customActions"  
12  :onAddItem="config.isAddEnabled ? onAddItem : undefined"  
13  @selectItem="onSelectItem">
```

Person List 57 of 59 People 🔍 + ⋮

<input type="checkbox"/>	Name	Birth Date	Age	Email Active	Campus	Connection Status			
<input type="checkbox"/>	Giver Anonymous			✓	Main Campus	Participant			
<input type="checkbox"/>	Presence Presence			✓	Main Campus	Visitor			
<input type="checkbox"/>	Ted Decker	2/10/1986	38	✓	Main Campus	Member			
<input type="checkbox"/>	Cindy Decker	3/10/1987	37	✓	Main Campus	Member			
<input type="checkbox"/>	Noah Decker	3/10/2014	10	✓	Main Campus	Attendee			



Custom Actions

Grid :definition=...

```
1  /**
2   * The custom actions supported by this grid.
3   */
4  const customActions = computed(): GridAction[] => {
5    return [
6      {
7        title: "Open Selected Batches",
8        iconCssClass: "fa fa-folder-open",
9        disabled: !hasSelectedRows.value,
10       handler: onOpenSelectedBatches
11      },
12      {
13        title: "Close Selected Batches",
14        iconCssClass: "fa fa-folder",
15        disabled: !hasSelectedRows.value,
16        handler: onCloseSelectedBatches
17      }
18    ];
19  });
```

<input type="checkbox"/>	Name	Date	Accounts	Variances	Status	
<input checked="" type="checkbox"/>	My Well Pending Transactions #598 Transaction Count: 1	7/31/2024	General Fund: \$10.00		Pending	
<input checked="" type="checkbox"/>	SampleData1721545200000 #583 Transaction Count: 1	7/21/2024	General Fund: \$110.00 Mission Fund: \$50.00		Closed	
<input type="checkbox"/>	SampleData1720940400000 #415 Transaction Count: 2	7/14/2024	General Fund: \$155.00 Mission Fund: \$50.00		Closed	\$205.00 x

598 Batches 2

- Merge Template
- Export to Excel
- Export to CSV
- Launch Workflow
- Open Selected Batches
- Close Selected Batches



Grid Definition

Grid :definition=...

```
1 <Grid :definition="config.gridDefinition ?? undefined"  
2     :data="gridDataSource"  
3     keyField="guid"  
4     personKeyField="guid"  
5     itemTerm="Person"  
6     tooltipField="email"  
7     stickyHeader  
8     liveUpdates  
9     :entityTypeGuid="EntityType.Person"  
10    :expectedRowCount="config.expectedRowCount ?? undefined"  
11    :customActions="customActions"  
12    :onAddItem="config.isAddEnabled ? onAddItem : undefined"  
13    @selectItem="onSelectItem">
```



Columns

Lots of properties to control most columns

DateColumn Example

```
1     <DateColumn name="birthDate"  
2         title="Birth Date"  
3         field="birthDate"  
4         :filter="dateValueFilter"  
5         visiblePriority="md" />
```




DateColumn Example

Columns

```
1 <DateColumn name="birthDate"  
2 title="Birth Date"  
3 field="birthDate"  
4 :filter="dateValueFilter"  
5 visiblePriority="md" />
```


Great filtering capabilities


Birth Date 


2/10/1980

3/10/1981


3/10/2008


Birth Date 

2 Equals 

3 

Apply Clear

Birth Date 

2 Equals 

3 Before

3 After

3/1 Between

This Week

Last Week

Next Week

This Month

Last Month



Tons of column types to choose from!

BooleanColumn

Email Active ▾

All ▾

- All
- No
- Yes

LabelColumn

Connection Status ▾

Any

Search

Attendee

Member

Participant

Prospect

Visitor

Apply Clear

NumberColumn

Age ▾

Top N

- Greater Than
- Greater Than Or Equal
- Less Than
- Less Than Or Equal
- Between
- Top N
- Above Average
- Below Average



- BooleanColumn
- CurrencyColumn
- DateColumn
- DateTimeColumn
- LabelColumn
- NumberColumn
- PersonColumn

Column Types



- TextColumn
- DeleteColumn
- ButtonColumn
- SelectColumn
- ReorderColumn
- AttributeColumns
- Column

Column Types (cont.)



Rows

The row data is an array of objects. They are completely unknown in that the Grid itself does not care about the structure. The column definitions handle mapping the row data into something that can be displayed. Row data can be supplied to the grid one of two ways. Either as a direct array of objects, or as a function that returns a promise containing the array of objects.



Batches
Home > Batches

Batch List 598 Batches Search

<input type="checkbox"/>	Name	Date	Variances	Status	Total Amount	
<input type="checkbox"/>	My Well Pending Transactions #598 Transaction Count: 1	7/31/2024		Pending	\$10.00	x
<input type="checkbox"/>	SampleData17215452000 #583 Transaction Count: 1	7/21/2024		Closed	\$160.00	x
<input type="checkbox"/>	SampleData17209404000 #415 Transaction Count: 2	7/14/2024		Closed	\$205.00	x
<input type="checkbox"/>	SampleData17203356000 #582 Transaction Count: 1	7/7/2024		Closed	\$160.00	x
<input type="checkbox"/>	SampleData17197308000 #414 Transaction Count: 2	6/30/2024		Closed	\$205.00	x
<input type="checkbox"/>	SampleData17191260000 #413 Transaction Count: 2	6/23/2024		Closed	\$205.00	x
<input type="checkbox"/>	SampleData17185212000 #412 Transaction Count: 2	6/16/2024		Closed	\$205.00	x

All the Rows

Designed to do client-side filtering and sorting

All data sent down at once

So, if you have a grid of 10,000 rows with paging set to 500, you aren't just dealing with 500 rows.





























All 10,000 rows must be sent down to the browser before anything will render.



Let there be “light”

A lighter weight grid

To be used with small grids.

Person List							64 People	🔍	+	⋮
<input type="checkbox"/>	Name	Birth Date	Age	Email Active	Campus	Connection Status				
<input type="checkbox"/>	 Admin Admin			✓	Main Campus	Visitor				
<input type="checkbox"/>	 Ted Decker	2/10/1986	38	✓	Main Campus	Member				
<input type="checkbox"/>	 Cindy Decker	3/10/1987	37	✓	Main Campus	Member				
<input type="checkbox"/>	 Noah Decker	3/10/2014	10	✓	Main Campus	Attendee				
<input type="checkbox"/>	 Alex Decker	2/10/2017	7	✓	Main Campus	Attendee				
<input type="checkbox"/>	 Ben Jones	7/10/1988	36	✓	Main Campus	Member				
<input type="checkbox"/>	 Brian Jones	6/9/2018	6	✓	Main Campus	Member				

50 ▾

< 1 2 >



```
1 <Grid :definition="config.gridDefinition ?? undefined"  
2   :data="gridDataSource"  
3   light  
4   keyField="guid"  
5   ...
```



FYI



Side Note About Blocks

Client / Server

All Obsidian blocks are comprised of essentially two parts.

Server Part

C# / .cs

Compiles into a .dll

This part is always required. This contains any logic required to view, edit, and otherwise interact with the person viewing the page. This does not handle any UI, just the logic required to provide the information to the UI.

Client Part

TypeScript / .obs

Transpiled into .obs.js

It can either be Obsidian, Rock Mobile, or some future UI framework. This is what handles the UI and the direct interaction with the person viewing the page.



Page Load

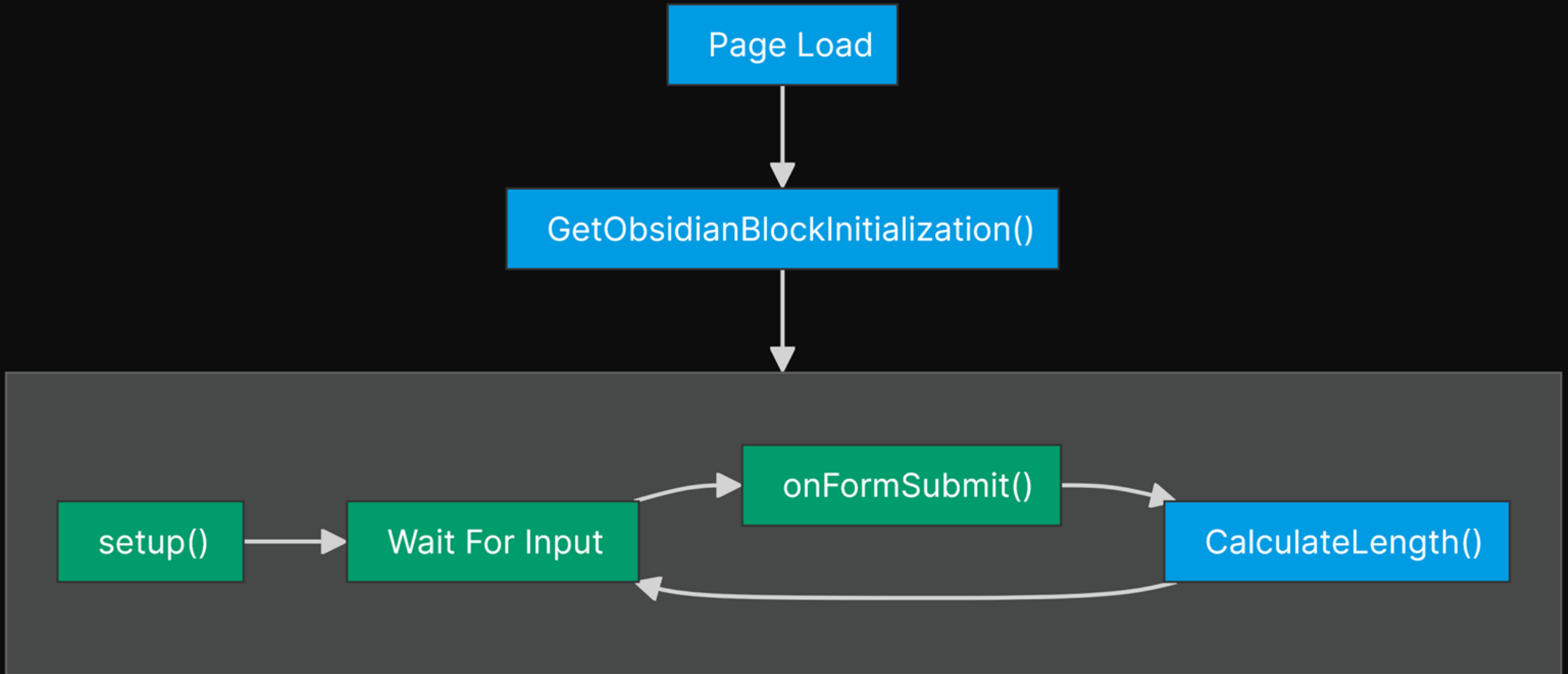
GetObsidianBlockInitialization()

setup()

Wait For Input

onFormSubmit()

CalculateLength()





Block Actions

A call to block action CalculateLength becomes a URL request.

```
POST https://www.rockrms.com/api/v2/<pageguid>/<blockguid>/CalculateLength
```

```
Block.cs

1 [BlockAction]
2 public BlockActionResult CalculateLength( string str )
3 {
4     if ( str = null )
5     {
6         return ActionBadRequest( "The string must be specified." );
7     }
8
9     return ActionOk( str.Length );
10 }
```



**We Now Resume
Our Regularly
Scheduled Program**

How to: C#

ObsidianListGallery.cs

```
37 [DisplayName( "Obsidian Gallery List" )]
38 [Category( "Example" )]
39 [Description( "Demonstrates the various parts of the Obsidian List blocks." )]
40 [IconCssClass( "fa fa-list" )]
41 [SupportedSiteTypes( Model.SiteType.Web )]
42
43 [Rock.SystemGuid.EntityTypeGuid( "4315fd92-f9f1-4038-abdc-a2d661b9dedf" )]
44 [Rock.SystemGuid.BlockTypeGuid( "121eec5e-f8aa-4cd8-a61d-9c99c269280e" )]
45 [CustomizedGrid]
46 public class ObsidianGalleryList : RockEntityListBlockType<Person>
47 {
48     #region Methods
49
50     /// <inheritdoc/>
51     public override object GetObsidianBlockInitialization()
52     {
53         var builder = GetGridBuilder();
54
55         return new ListBlockBox<Dictionary<string, object>>
56         {
57             IsAddEnabled = true,
58             IsDeleteEnabled = true,
59             GridDefinition = builder.BuildDefinition()
60         };
61     }
}
```

GetObsidianBlockInitialization()

Just returns a ListBlockBox

How to: C#

ObsidianListGallery.cs

```
71     /// <summary>
72     /// Gets the grid builder that will be used to construct the definition
73     /// and the final row data.
74     /// </summary>
75     /// <returns>A <see cref="GridBuilder{T}"> instance that will handle building
76     protected override GridBuilder<Person> GetGridBuilder()
77     {
78         return new GridBuilder<Person>()
79             .WithBlock( this )
80             .AddField( "guid", p => p.Guid.ToString() )
81             .AddTextField( "nickName", p => p.NickName )
82             .AddTextField( "lastName", p => p.LastName )
83             .AddTextField( "photoUrl", p => p.PhotoUrl )
84             .AddTextField( "email", p => p.Email )
85             .AddField( "isEmailActive", p => p.IsEmailActive )
86             .AddDateTimeField( "birthDate", p => p.BirthDate )
87             .AddField( "campus", p => p.PrimaryCampus?.Name )
88             .AddField( "connectionStatus", p => GetConnectionStatus( p ) )
89             .AddField( "age", p => p.Age )
90             .AddAttributeFields( GetGridAttributes() );
91     }
```

GetGridBuilder()

Just returns a GridBuilder that you configure.

How to: C#

ObsidianListGallery.cs

```
1      /// <summary>
2      /// Gets the connection status as a <see cref="ListItemBag" />. This
3      /// is a special format used by the Label column to handle the color.
4      /// </summary>
5      /// <param name="person">The person whose connection status we want.</param>
6      /// <returns>A <see cref="ListItemBag" /> that represents the connection status.
7      private ListItemBag GetConnectionStatus( Person person )
8      {
9          if ( !person.ConnectionStatusValueId.HasValue )
10         {
11             return null;
12         }
13
14         var connectionStatus = DefinedValueCache.Get( person.ConnectionStatusValueId );
15         if ( connectionStatus == null )
16         {
17             return null;
18         }
19
20         var color = connectionStatus.GetAttributeValue( "Color" );
21         return new ListItemBag
22         {
23             Value = color.IsNullOrEmpty() ? "#c3c3c3" : color,
24             Text = connectionStatus.Value
25         };
26     }
```

Special Function Methods ()

Used when you need to build a custom value (ListItemBag) that holds your **Value** and **Text**.



How to: Obsidian/TypeScript

obsidianListGallery.obs

<template>

Obsidian. This is where your UI is defined.

```
1 <template>
2   <Grid :definition="config.gridDefinition ?? undefined"
3     :data="gridDataSource"
4     keyField="guid"
5     personKeyField="guid"
6     itemTerm="Person"
7     tooltipField="email"
8     stickyHeader
9     liveUpdates
10    :entityTypeGuid="EntityType.Person"
11    :expectedRowCount="config.expectedRowCount ?? undefined"
12    :customActions="customActions"
13    :onAddItem="config.isAddEnabled ? onAddItem : undefined"
14    @selectItem="onSelectItem">
15
```



How to: Obsidian/TypeScript

obsidianListGallery.obs

Columns

Then you define your columns.

```
16      <SelectColumn />
```




How to: Obsidian/TypeScript

obsidianListGallery.obs

Columns
More...

```
18     <PersonColumn name="leader"  
19                 title="Campus Leader"  
20                 field="leader"  
21                 :filter="pickExistingValueFilter"  
22                 visiblePriority="sm" />
```

 This requires that the field be added to the grid with the `.AddPersonField()` method so that all the required values are available.



How to: Obsidian/TypeScript

obsidianListGallery.obs

After all the columns...
We start the code part.

```
85     </Grid>
86 </template>
87
88 <script setup lang="ts">
89     import { useConfigurationValues, useInvokeBlockAction } from "@Obsidian/Utility/bl
90     import { EntityType } from "@Obsidian/SystemGuids/entityType";
91     import { ListBlockBox } from "@Obsidian/ViewModels/Blocks/listBlockBox";
92     import { GridDataBag } from "@Obsidian/ViewModels/Core/Grid/gridDataBag";
93     import Grid, { AttributeColumns, BooleanColumn, Column, DateColumn, dateValueFilter
94     import { alert } from "@Obsidian/Utility/dialogs";
95     import { GridAction } from "@Obsidian/Types/Controls/grid";
96     import { reactive, ref } from "vue";
97
98     const config = useConfigurationValues<ListBlockBox<Record<string, string>>>();
99     const invokeBlockAction = useInvokeBlockAction();
100
101     // #region Values
102
103     const gridDataSource = ref<Promise<GridDataBag>>();
104     let gridData: GridDataBag | undefined;
105
106     const customActions: GridAction[] = [
107     {
108         title: "Custom Action",
109         tooltip: "This is an example of a custom block action.",
110         iconCssClass: "fa fa-key"
111     }
```



How to: Obsidian/TypeScript

obsidianListGallery.obs

Functions

```
117     /**
118     * Called when the grid is requesting the row data be loaded.
119     */
120     async function loadGridData(): Promise<GridDataBag> {
121         const result = await invokeBlockAction<GridDataBag>("GetGridData");
122
123         if (result.isSuccess && result.data) {
124             gridData = reactive(result.data);
125             return gridData;
126         }
127         else {
128             throw new Error(result.errorMessage ?? "Unknown error while trying to load
129         }
130     }
```



How to: Obsidian/TypeScript

obsidianListGallery.obs

Event Handlers

```
137     /**
138      * Called when a row has been selected by the person.
139      *
140      * @param key The key of the row that was selected.
141      */
142     function onSelectItem(key: string): void {
143         window.location.href = `/Person/${key}`;
144     }
145
146     /**
147      * Called when a row's edit button has been clicked by the person.
148      *
149      * @param key The key of the row that should be edited.
150      */
151     function onEditClick(key: string): void {
152         window.location.href = `/Person/${key}`;
153     }
```



How to: Obsidian/TypeScript

obsidianListGallery.obs

Event Handlers

...continued.

```
155     /**
156     * Called when a row's delete button has been clicked by the person. This
157     * happens only if they have confirmed the delete action.
158     *
159     * @param key The key of the row that should be deleted.
160     */
161     async function onDeleteClick(key: string): Promise<void> {
162         await alert("Deleting is not supported for this sample grid, it will only be s
163
164         if (gridData) {
165             const index = (gridData.rows ?? []).findIndex(r => r["guid"] === key);
166
167             if (index !== -1) {
168                 gridData.rows?.splice(index, 1);
169             }
170         }
171     }
172
173     /**
174     * Called when the Add button has been clicked by the person.
175     */
176     function onAddItem(): void {
177         window.location.href = "/NewFamily";
178     }
```



How to: Obsidian/TypeScript

obsidianListGallery.obs

Bind and Finished

```
180     // #endregion
181
182     gridDataSource.value = loadGridData();
183 </script>
```



Goals & Guidelines

Clean Consistency

(when practical)

<input type="checkbox"/>	Person	Pledge Total	Total Giving Amount	Percent Complete	Giving Count
No Items Found					
Total Results					
				Total Amount Pledged:	\$0.00
				Total Amount Given:	\$0.00

Batch List					2 Batches	🔍	☰	+	⋮
<input type="checkbox"/>	Name	Date	Variances	Status	Total Amount				
<input type="checkbox"/>	Online Giving VMD #856 Transaction Count: 1	8/2/2024		Open	\$23.00	✕			
<input type="checkbox"/>	Event Registration VMD #855 Transaction Count: 1	8/2/2024		Open	\$10.00	✕			
					50	▼			
					Totals		Show all		
					General Fund	\$23.00			
					Event Registration	\$10.00			
					Total	\$33.00			



Take Away

The Obsidian Development Guide

<https://sparkdevnetwork.gitbook.io/obsidian/>