Creatio Academy

Data processing

Process collection type data

Version 8.0

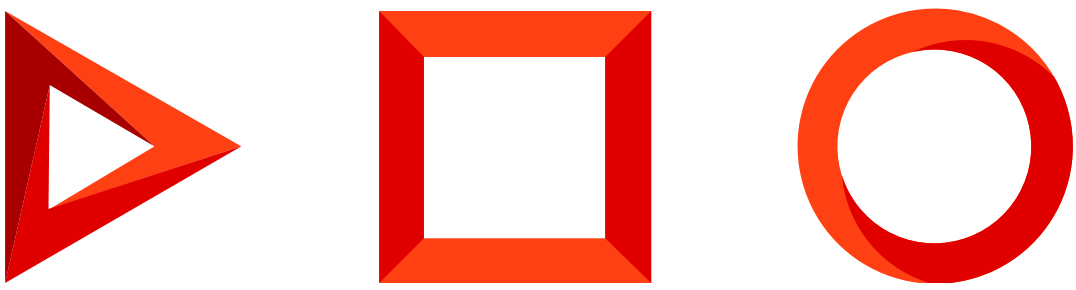

Table of Contents

Process collection type data

Additional data processing can be required when displaying collection type attributes.

Creatio lets you manage:

- pagination
- sorting
- filtering

Set up data processing in the `viewModelConfig` schema section.

Set up pagination

Use the internal `pagingConfig` property of the collection type attribute's modelConfig property to **set up pagination**.

The `pagingConfig` property includes the following **internal properties**:

- `rowCount`. The number of records to upload to the page. The `rowCount` property value can be both a constant and the name of the attribute that contains this number.
- `rowsOffset`. An initial position (offset) to load the first portion of data. Can only be the name of the attribute that contains the offset number, not a constant. If you omit the property, Creatio sets the offset to 0.

Example of pagination

```
viewModelConfig: /**SCHEMA_VIEW_MODEL_CONFIG*/{
    "attributes": {
        "LookupAttribute": {
            "isCollection": true,
            "modelConfig": {
                ...
                "pagingConfig": {
                    "rowCount": SOME_QUANTITY_OF_DATA,
                    "rowsOffset": "Some_Offset_Of_Data",
                },
                ...
             },
             "viewModelConfig": {
                ...
             },
             "embeddedModel": {
                 ...
            }
        }
    }
```

```
}/**SCHEMA_VIEW_MODEL_CONFIG*/,
```

Set up sorting

Use the internal `sortingConfig` property of the collection type attribute's `modelConfig` property to **set up sorting**.

The `sortingConfig` property includes the following **internal properties**:

- `attributeName`. Sorting attribute that manages sorting in Creatio UI, for example, in the section list. Required to load new data.

  Attention. You do not need to set the `columnName` and `direction` of the sorting attribute in the source code of the Freedom UI page schema. Creatio manages the values of this attribute automatically.

- `default`. Specifies the initial data sorting settings. An array of objects that have the following **properties**:
  - `columnName`. Name of the column by which to sort data.
  - `direction`. Sorting order. Available values: `asc` (ascending), `desc` (descending).

Example of sorting

```
viewModelConfig: /**SCHEMA_VIEW_MODEL_CONFIG*/{
    "attributes": {
        "LookupAttribute": {
            "isCollection": true,
            "modelConfig": {
                ...
                "sortingConfig": {
                    "attributeName": "Some_Attribute_Name",
                    "default": [
                        {
                            "columnName": SomeColumnName,
                            "direction": 'asc',
                         },
                     ],
                 },
                 ...
            },
            "viewModelConfig": {
                ...
            },
            "embeddedModel": {
        ...
            }
        },
        "Some_Attribute_Name": {
```

```
            "isCollection": true,
            "viewModelConfig": {
                "attributes": {
                    "columnName": {},
                    "direction": {},
                },
            },
        }
    }
}/**SCHEMA_VIEW_MODEL_CONFIG*/,
```

Set up filtering

Use the internal `filterAttributes` property of the collection type attribute's `modelConfig` property to **set up filtering**.

The `filterAttributes` property is an array of objects that have the following **properties**:

- `name`. Name of the attribute to filter data. For example, "`FolderTree_items_DS_filter`." Declare the attribute in the `viewModelConfig` schema section. Set the value property of this attribute to the object that configures the filter based on `EntitySchemaQuery`.
- `loadOnChange`. Specifies whether to reload the collection on filter change.

Example of filtering

```
viewModelConfig: /**SCHEMA_VIEW_MODEL_CONFIG*/{
    "attributes": {
        /* Collection type attribute to filter. */
        "FolderTree_items": {
            "isCollection": true,
            "viewModelConfig": {
                ...
            },
            "modelConfig": {
                "path": "FolderTree_items_DS",
                /* Set up filtering. */
                "filterAttributes": [
                    {
                        /* Name of the attribute to filter data. */
                        "name": "FolderTree_items_DS_filter",
                        "loadOnChange": true
                    }
                ]
            },
            "embeddedModel": {
                ...
            }
```

```
        },
        /* The attribute to filter data.*/
        "FolderTree_items_DS_filter": {
            "value": {
                "isEnabled": true,
                "trimDateTimeParameterToDate": false,
                "filterType": 6,
                "logicalOperation": 0,
                /* Load all folders whose EntitySchemaName = Account.*/
                "items": {
                    "6f20f5b6-42e6-d6b8-caee-5e54636b76d1": {
                        "isEnabled": true,
                        "trimDateTimeParameterToDate": false,
                        "filterType": 1,
                        "comparisonType": 3,
                        "leftExpression": {
                            "expressionType": 0,
                            "columnPath": "EntitySchemaName"
                        },
                        "rightExpression": {
                            "expressionType": 2,
                            "parameter": {
                                "dataValueType": 1,
                                "value": "Account"
                            }
                        }
                    }
                }
            }
        }
    }
}/**SCHEMA_VIEW_MODEL_CONFIG*/,
```