Creatio Academy

Develop your first application

Step 3. Add page validation

Version 7.17

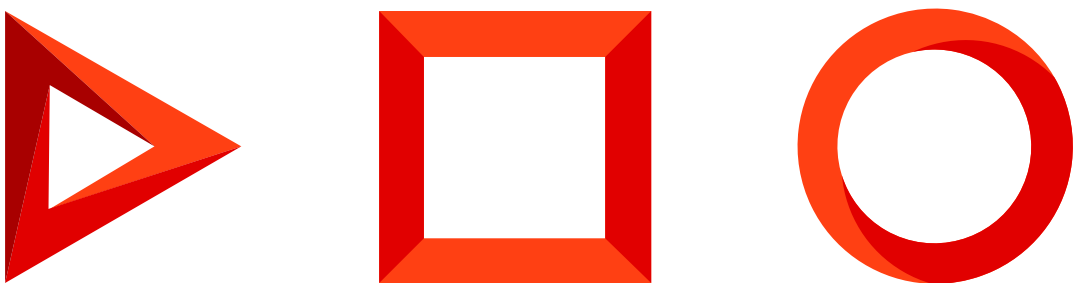


Table of Contents

Step 3. Add page validation

Beginner

On the previous step, we added the example's required data to the interface and attached it to the development package.

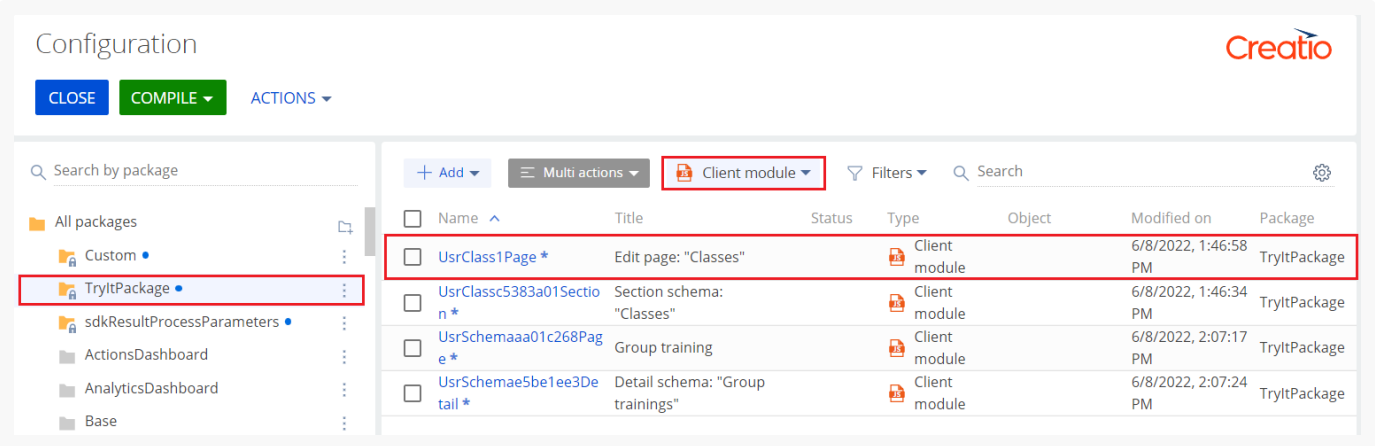
Now, implement the business logic of the example.

Provision.

- A daily class is scheduled only if there is an unoccupied gym.
- The number of the fitness center's gyms is set via a system setting and equals 4.
- When adding or editing a daily group class, Creatio must check if the changes result in the total number of active daily classes exceeding the value of the system setting. If yes, saving is disallowed and the message "No available gyms, no more than N classes allowed" is displayed, where N is the system setting's value.

To do this, modify the source code of the client module responsible for the section page operation.

1. Go to the [*Configuration*] section.
2. Select the "TryItPackage" package from the package list.
3. The Wizards add schemas of various types to the package. Filter schemas by the [*Client module*] type.



4. Double-click the "`UsrClass1Page`" schema to open it and modify its source code.

`UsrClass1Page.js`

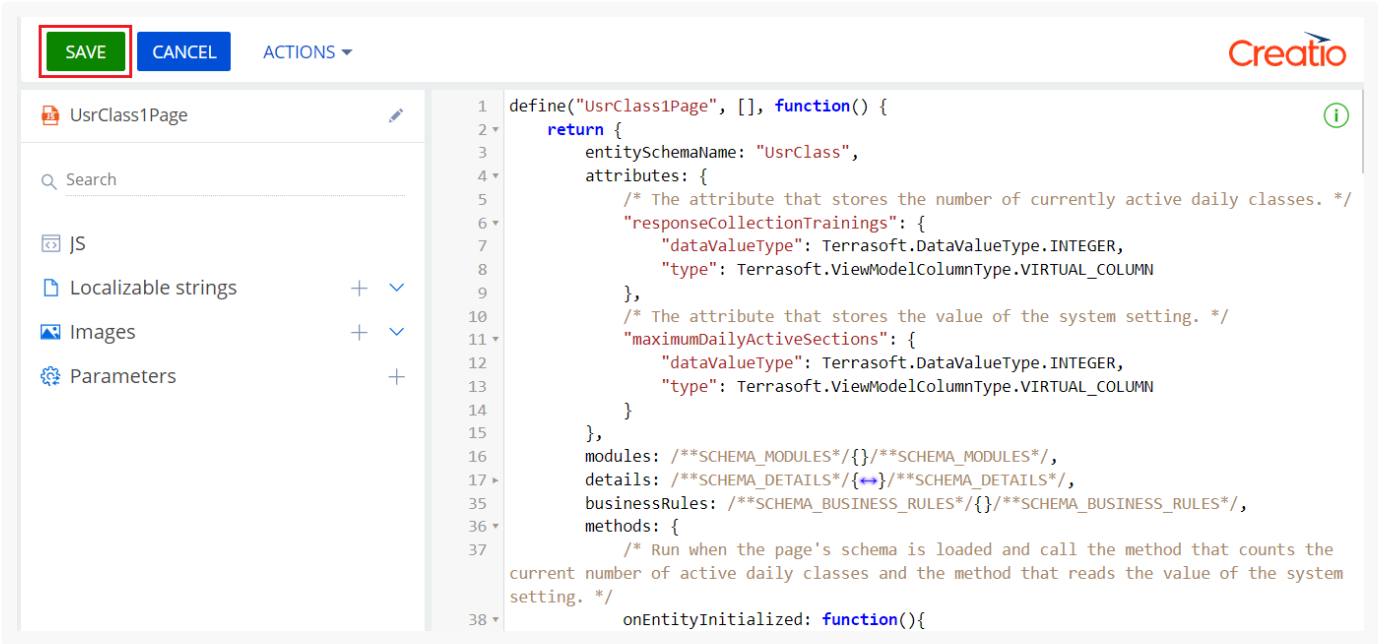
```
define("UsrClass1Page", [], function() {
    return {
        entitySchemaName: "UsrClass",
```

```
        messages: {},
        attributes: {
            /* The attribute that stores the number of currently active daily classes. */
            "responseCollectionTrainings": {
                "dataValueType": Terrasoft.DataValueType.INTEGER,
                "type": Terrasoft.ViewModelColumnType.VIRTUAL_COLUMN
            },
            /* The attribute that stores the value of the system setting. */
            "maximumDailyActiveSections": {
                "dataValueType": Terrasoft.DataValueType.INTEGER,
                "type": Terrasoft.ViewModelColumnType.VIRTUAL_COLUMN
            }
        },
        methods: {
            /* Run when the page's schema is loaded and call the method that counts the curre
            onEntityInitialized: function(){
                this.callParent(arguments);
                this.getPeriodicityActiveNumber();
                this.getMaximumDailyActiveSections();
            },
            /* Calculate the current number of active daily classes and write the calculated
            getPeriodicityActiveNumber: function() {
                var periodicity = "Daily";
                var esqPeriodicity = this.Ext.create("Terrasoft.EntitySchemaQuery", {
                    rootSchemaName: "UsrClass"
                });
                esqPeriodicity.addColumn("UsrName");
                var groupFilters = this.Ext.create("Terrasoft.FilterGroup");
                var filterPerodicity = this.Terrasoft.createColumnFilterWithParameter(this.Te
                var thisId = this.get("Id");
                var filterId = this.Terrasoft.createColumnFilterWithParameter(this.Terrasoft.
                var filterIsActive = this.Terrasoft.createColumnFilterWithParameter(this.Terr
                groupFilters.addItem(filterPerodicity);
                groupFilters.logicalOperation = this.Terrasoft.LogicalOperatorType.AND;
                groupFilters.addItem(filterIsActive);
                groupFilters.logicalOperation = this.Terrasoft.LogicalOperatorType.AND;
                groupFilters.addItem(filterId);
                esqPeriodicity.filters.add(groupFilters);
                esqPeriodicity.getEntityCollection(function(result) {
                    if (!result.success) {
                        this.showInformationDialog("Request error");
                        return;
                    }
                    else {
                        var lengthCollection = result.collection.collection.length;
                        this.set("responseCollectionTrainings", lengthCollection);
                    }
                }, this);
            },
```

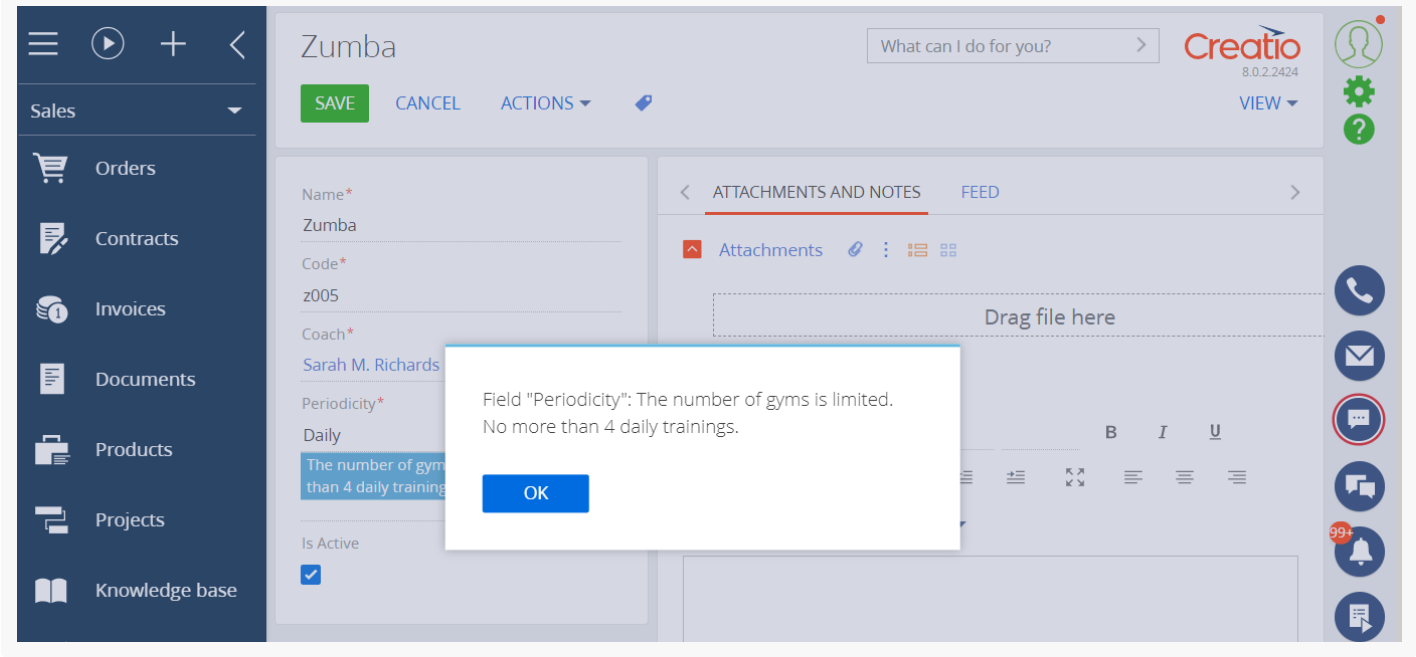
```
            /* Provide validation for the "Periodicity" field. The validator method will be c
            setValidationConfig: function() {
                this.callParent(arguments);
                this.addColumnValidator("UsrPeriodicity", this.periodicityValidator);
            },
            /* The validator method: if the class is daily, compare the number of active dail
            periodicityValidator: function() {
                var invalidMessage= "";
                var periodicity = this.get("UsrPeriodicity").displayValue;
                if (periodicity==="Daily") {
                    var isActive = this.get("UsrIsActive");
                    var myVariable = this.get("maximumDailyActiveSections");
                    var lengthCollection = this.get("responseCollectionTrainings");
                    if (lengthCollection >= myVariable && isActive) {
                        invalidMessage = "The number of gyms is limited. No more than " + myV
                    }
                }
                else {
                    invalidMessage = "";
                }
                return {
                    invalidMessage: invalidMessage
                };
            },
            /* Retrieve the value of the "GymNumber" system setting. */
            getMaximumDailyActiveSections: function() {
                var myVariable;
                var callback = function(value) {
                    myVariable = value;
                };
                this.Terrasoft.SysSettings.querySysSettingsItem("GymsNumber", callback, this)
                if (myVariable === undefined) {
                    return;
                }
                else {
                    this.set("maximumDailyActiveSections", myVariable);
                }
            }
        },
        modules: /**SCHEMA_MODULES*/{}/**SCHEMA_MODULES*/,
        /* No changes. */
        details: /**SCHEMA_DETAILS*/{

        // …

        }/**SCHEMA_DETAILS*/,
        businessRules: /**SCHEMA_BUSINESS_RULES*/{}/**SCHEMA_BUSINESS_RULES*/,
        dataModels: /**SCHEMA_DATA_MODELS*/{}/**SCHEMA_DATA_MODELS*/,
```

```
        /* No changes. */
        diff: /**SCHEMA_DIFF*/[

        // …

        ]/**SCHEMA_DIFF*/
    };
});
```

5. Click the [*Save*] button to save the schema.



The result of the changes:



As a result, we implemented the required business logic of the page. On the next step, implement the population of the group class timetable.