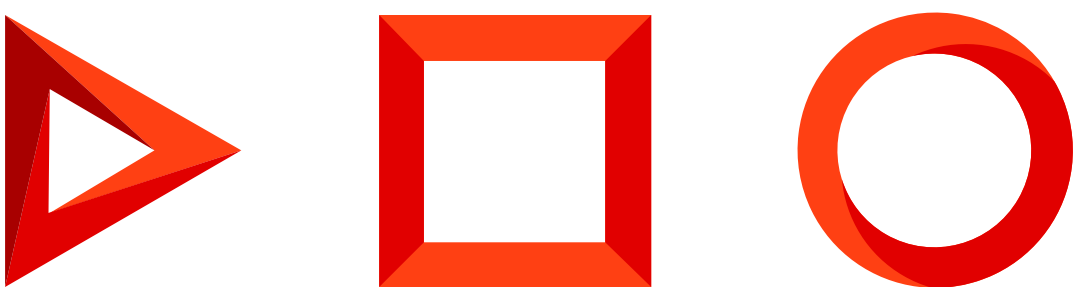
Creatio Academy

Page customization basics

Freedom UI page customization basics

Version 8.0



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

© 2023 Creatio. All rights reserved.

Table of Contents

Freedom UI page customization basics **4**
Page customization procedure 4
Close the WebSocket when destroying the View of the model 5
Manage the system setting values on a page **5**
1. Set up page UI 5
2. Manage system setting values 8
Outcome of the example 11
Change where the query handler is invoked on the page **12**
1. Set up page UI 12
2. Change the origin of the query handler call 14
Outcome of the example 16

© 2023 Creatio. All rights reserved.

Freedom UI page customization basics

Beginner

Configure the business logic of Freedom UI pages in the `validators`, `converters`, and `handlers` sections of the client schemas. We recommend using no-code tools to set up business logic. Learn more in the user documentation: Freedom UI Designer. If no-code tools are not sufficient to implement your customization, we recommend setting up business logic in the source code of the Freedom UI page schema. Learn more about creating a Freedom UI page in a separate article: Client module.

For example, you can use a Freedom UI page to implement the following **business logic**:

- element visibility
- element locking
- element requirement
- element filtering
- field population
- Creatio data querying
- HTTP request sending
- page navigation

Learn more about pages in a separate article: Freedom UI page.

Page customization procedure

1. Set up an **app page**.
   a. Create a custom app. To do this, follow the procedure in the user documentation: Create a custom app.
   b. Add one or more elements whose business logic to set up. To do this, follow the procedure in the user documentation: Set up the app UI.
2. Set up the **business logic of a page item**.
   Configure the business logic in the client schema of the Freedom UI page. To **open the client module schema**, click the  button on the action panel of the Freedom UI Designer on the corresponding page.
   The source code of the Freedom UI page opens after you save the page settings. Perform the setup in the corresponding client schema sections of the Freedom UI page Learn more in a separate article: Client schema.

Query handlers are the preferred way to customize the page. **Query handlers** are items of the `HandlerChain` mechanism that lets you describe business logic as an action request and chain of handlers. You can manage data sources using the `handlers` schema section. Query examples: page readiness, data load and save, business process launch.

Creatio 8 Atlas lets you organize query handlers in event chains. For example, trigger the base Creatio save handler first and execute the custom page logic later upon the page saver query.

© 2023 Creatio. All rights reserved.

To **limit the scope of a query handler**, fill out the `scopes` property of the client module with the names of the client schemas for which the handler must work.

View detailed examples of handler invocation in a separate block: Examples.

Close the WebSocket when destroying the `View` of the model

To **close the WebSocket when destroying** the `View` of the **model**, add a custom implementation of the `crt.HandleViewModelDestroyRequest` system query handler to the `handlers` schema section. The handler is executed when the `View` of the model is destroyed (for example, when you open another page). Designed to destroy resources. We do not recommend writing asynchronous code in the handler (server calls, timeouts, etc.) except for reading the value of attributes.

View an example of the `crt.HandleViewModelDestroyRequest` query handler that closes the custom `SomeWebSocket` WebSocket below.

`handlers` schema section

Manage the system setting values on a page

Medium

Example. Display the following on the record page of the [*Requests*] custom section:

- User's city.
- The number of the last created record in the [*Requests*] section, increased by 1. Increase the number when creating a new request or copying an existing one.

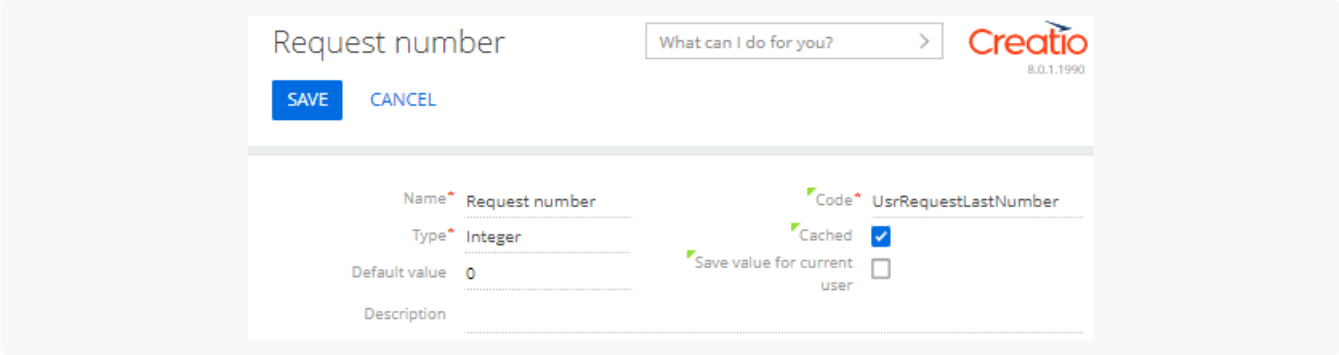
Use the values of custom system settings.

1. Set up page UI

1. Create a custom `Requests` app based on the [*Records & business processes*] template. To do this, follow the procedure in the user documentation: Create a custom app.
2. Add a **system setting that stores the request number**.
   a. Click [*Run app*] on the `Requests` app page.
   b. Click to open the System Designer. Click [*System settings*] in the [*System setup*] block.
   c. Click [*Add setting*] on the section toolbar.
   d. Fill out the **system setting properties**:
      - Set [*Name*] to "`Request number`."
      - Set [*Code*] to "`UsrRequestLastNumber`."

© 2023 Creatio. All rights reserved.

- Select "`Integer`" in the [*Type*] property.
- Set [*Default value*] to "`0`."

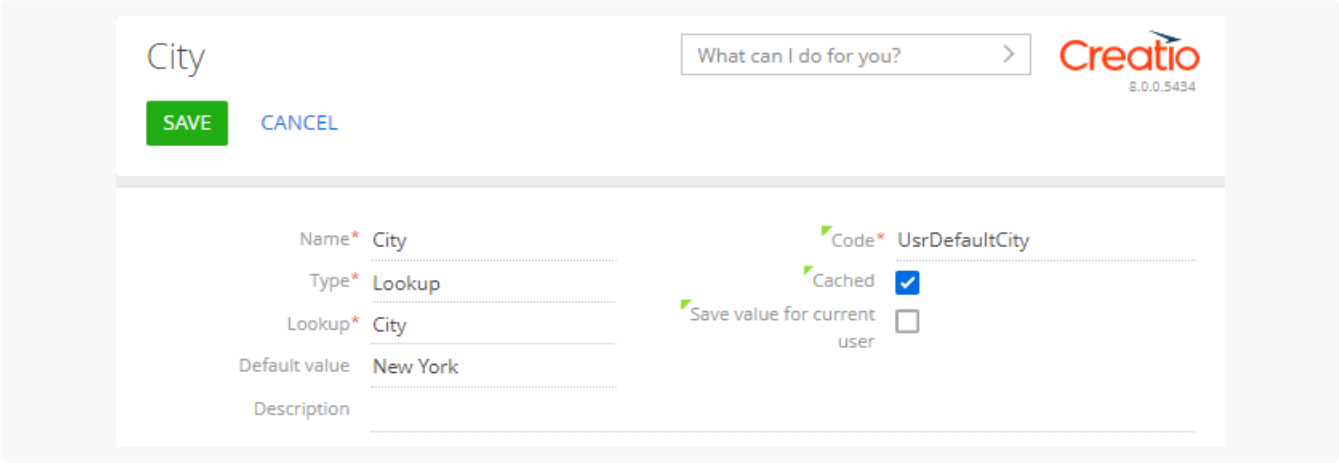


3. Add a **system setting that stores the city name**.

   a. Click [*Add setting*] on the section toolbar.

   b. Fill out the **system setting properties**:

   - Set [*Name*] to "`City`."
   - Set [*Code*] to "`UsrDefaultCity`."
   - Select "`Lookup`" in the [*Type*] property.
   - Select "`City`" in the [*Lookup*] property.
   - Select "`New York`" in the [*Default value*] property.



4. Go to the `Requests` app page and open the [*Requests form page*] page in the workspace.

   The [*Requests form page*] page includes the [*Name*] field by default.

5. Add a **request number label**.

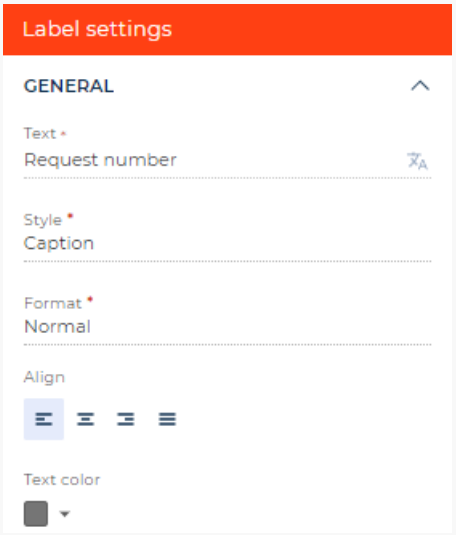
   a. Add a [*Label*] component to the Freedom UI Designer's workspace.

   b. Click the button on the Freedom UI Designer's action panel and fill out the **title's properties** in the setup area:

   - Set [*Text*] to "`Request number`."

© 2023 Creatio. All rights reserved.

- Select "`Caption`" in the [*Style*] property.
- Select the grey color in the [*Text color*] property.



6. Add the following **labels** the same way:

   - The value of the system setting that stores the request number.
   - Cities.
   - The value of the city system setting.

   View the label properties to add in the table below.

© 2023 Creatio. All rights reserved.

Label property values

Element	Property	Property value
A label that contains the value of the system setting that stores the request number.	[*Text*]	"Request number (value)"
	[*Style*]	Select "Body text"
City label	[*Text*]	"City"
	[*Style*]	Select "Caption"
	[*Text color*]	Select the grey color
A label that contains the value of the city system setting.	[*Text*]	"City (value)"
	[*Style*]	Select "Body text"

7. Click the </> button on the Freedom UI Designer's action bar. The source code of the Freedom UI page opens after you save the page settings.

2. Manage system setting values

Configure the business logic in the Client Module Designer. In this example, configure how to manage system settings values.

1. Enable the `sdk.SysSettingsService` **system setting service**. To do this, add the `@creatio-devkit/common` dependency to the AMD module.

AMD module dependencies

```
/* AMD module declaration. */
define("UsrAppRequests_FormPage", /**SCHEMA_DEPS*/["@creatio-devkit/common"] /**SCHEMA_DEPS*/
    return {
        ...
    };
});
```

© 2023 Creatio. All rights reserved.

2. Add the following **attributes** to the `viewModelConfig` schema section:

   - `UsrDefaultCity`. Stores data about the value of the `UsrDefaultCity` system setting.
   - `UsrRequestLastNumber`. Stores data about the value of the `UsrRequestLastNumber` system setting.

   `viewModelConfig` schema section

```
viewModelConfig: /**SCHEMA_VIEW_MODEL_CONFIG*/{
    "attributes": {
        ...,
        /* The attribute that stores the value of the UsrDefaultCity system setting. */
        "UsrDefaultCity": {},
        /* The attribute that stores the value of the UsrRequestLastNumber system setting. */
        "UsrRequestLastNumber": {}
    }
}/**SCHEMA_VIEW_MODEL_CONFIG*/,
```

3. Change the following **property values** in the `viewConfigDiff` schema section:

   - `caption` property for the `RequestNumberValue` element. Bind the property to the value of the `$UsrRequestLastNumber` attribute. The `caption` property is responsible for the text that the element contains.
   - `caption` property for the `CityValue` element. Bind the property to the value of the `$UsrDefaultCity` attribute.

   `viewConfigDiff` schema section

```
viewConfigDiff: /**SCHEMA_VIEW_CONFIG_DIFF*/[
    ...,
    {
        "operation": "insert",
        "name": "RequestNumberValue",
        "values": {
            ...,
            /* The property responsible for the text that the element contains. Bound to the
            "caption": "$UsrRequestLastNumber",
            ...
        },
        ...
    },
    ...,
    {
        "operation": "insert",
        "name": "CityValue",
        "values": {
            ...,
```

© 2023 Creatio. All rights reserved.

```
            /* The property responsible for the text that the element contains. Bound to the
            "caption": "$UsrDefaultCity",
            ...
        },
        ...
    }
]/**SCHEMA_VIEW_CONFIG_DIFF*/,
```

4. Add a custom implementation of the `crt.HandlerViewModelInitRequest` system query handler to the `handlers` schema section. Execute the handler when the `View` model is initialized.

   a. Instantiate the system value service from `@creatio-devkit/common`.

   b. Retrieve the value of the `UsrDefaultCity` system setting and write it to the `UsrDefaultCity` attribute.

   c. Retrieve the page state.

   d. Retrieve the value of the `UsrRequestLastNumber` system setting.

   e. When creating a new record or copying an existing record, send a query to update the value of the `UsrRequestLastNumber` system setting (increase it by 1).

   f. Update the value of the `UsrRequestLastNumber` attribute.

`handlers` schema section

```
handlers: /**SCHEMA_HANDLERS*/[
    {
        request: "crt.HandleViewModelInitRequest",
        /* Custom implementation of a system query handler. */
        handler: async (request, next) => {
            /* Wait for the rest of the initialization handlers to finish. */
            await next?.handle(request);
            /* Instantiate a system value service from @creatio-devkit/common. */
            const sysSettingsService = new sdk.SysSettingsService();
            /* Retrieve the value of the UsrDefaultCity system setting and write it to the Us
            const defaultCity = await sysSettingsService.getByCode('UsrDefaultCity');
            request.$context.UsrDefaultCity = defaultCity.displayValue;
            /* Retrieve the page state. */
            const cardState = await request.$context.CardState;
            /* When creating a new record or copying an existing record, increase the value o
            if (cardState === 'add' || cardState === 'copy') {
                /* Retrieve the value of the UsrRequestLastNumber system setting and write it
                const requestLastNumber = await sysSettingsService.getByCode('UsrRequestLastN
                /* Send a query to update the value of the UsrRequestLastNumber system settin
                await sysSettingsService.update({
                    code: 'UsrRequestLastNumber',
                        /* The new value is higher than the previous one by 1. */
                        value: ++requestLastNumber.value
                });
                /* Update the value of the UsrRequestLastNumber attribute. */
```

© 2023 Creatio. All rights reserved.

```
                request.$context.UsrRequestLastNumber = requestLastNumber.value;
            }
        },
    }
]/**SCHEMA_HANDLERS*/,
```

Complete source code of the page schema

5. Click [*Save*] on the Client Module Designer's toolbar.

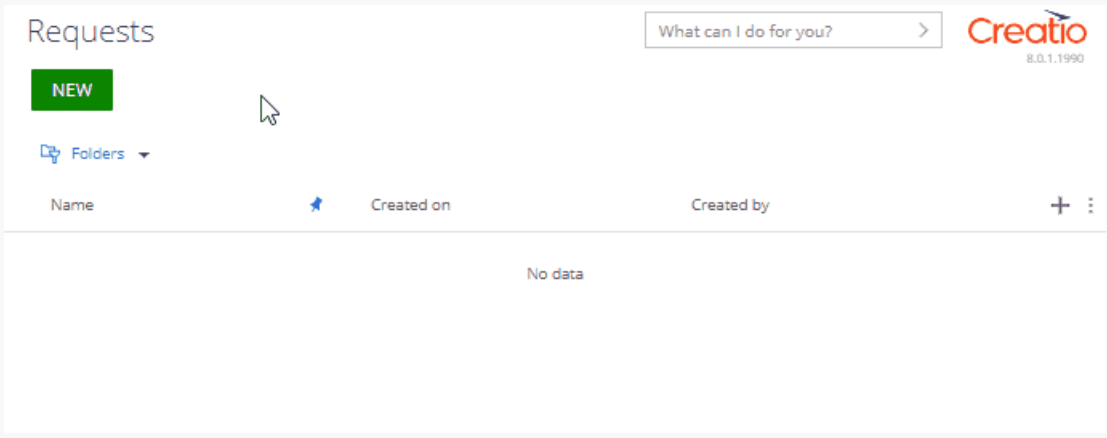
Outcome of the example

To **view the outcome of the example when creating a new request**:

1. Go to the `Requests` app page and click [*Run app*].
2. Click [*New*] on the `Requests` app toolbar.

As a result, the record page of the [*Requests*] custom section will display:

- User's city. The default value of the [*City*] (`UsrDefaultCity` code) system setting is used.
- The number of the last created record in the [*Requests*] section, increased by 1. The value of the [*Request number*] (`UsrRequestLastNumber` code) system setting is used. It is increased by 1 when a new request is created.



To **view the outcome of the example when copying an existing request**:

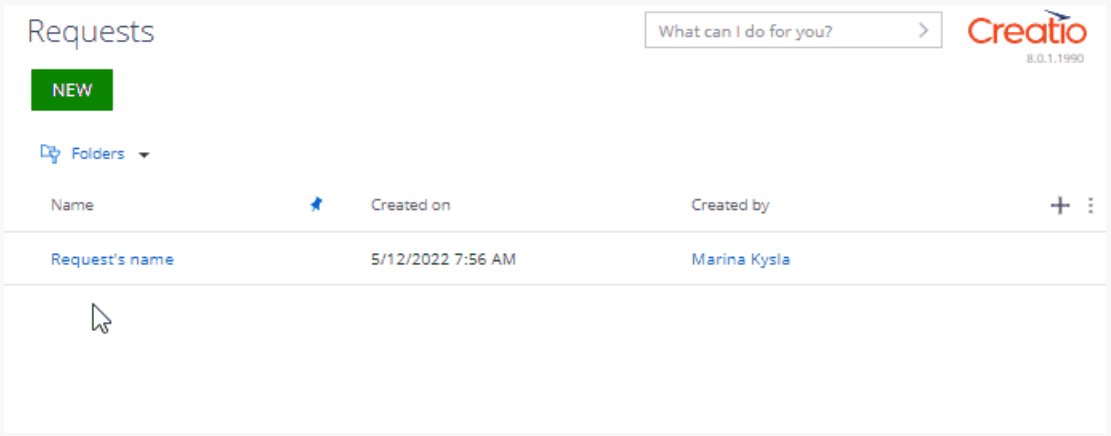
1. Add a [*Request's name*] request first.
2. Click → [*Copy*] in the [*Request's name*] request's row in the record list.

As a result, the page of the copied record of the [*Requests*] custom section will display:

- User's city. The default value of the [*City*] (`UsrDefaultCity` code) system setting is used.
- The number of the last created record in the [*Requests*] section, increased by 1. The value of the [*Request number*] (`UsrRequestLastNumber` code) system setting is used. It is increased by 1 when an existing request is

© 2023 Creatio. All rights reserved.

copied.



Change where the query handler is invoked on the page

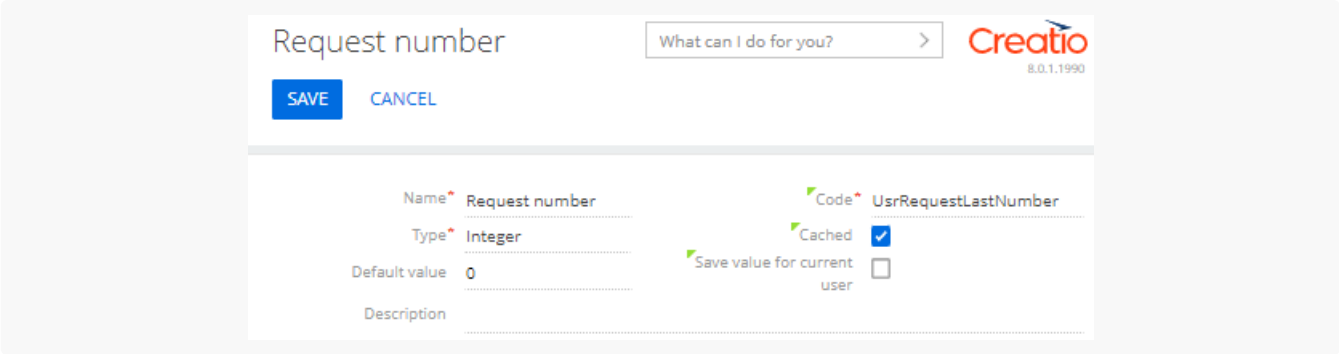
Medium

Example. Display the number of the newest section record on the record page of the [*Requests*] custom section. Set the number to the custom system setting. Increment the number by 1 after saving the request page.

1. Set up page UI

1. Create a custom `Requests` app based on the [*Records & business processes*] template. To do this, follow the procedure in the user documentation: Create a custom app.
2. Add **a system setting that stores the request number**.
   a. Click to open the System Designer. Click [*System settings*] in the [*System setup*] block.
   b. Click [*Add setting*] on the section toolbar.
   c. Fill out **the system setting properties**:
      - Set [*Name*] to "`Request number`."
      - Set [*Code*] to "`UsrRequestLastNumber`."
      - Select "`Integer`" in the [*Type*] property.
      - Set [*Default value*] to "`1`."

© 2023 Creatio. All rights reserved.



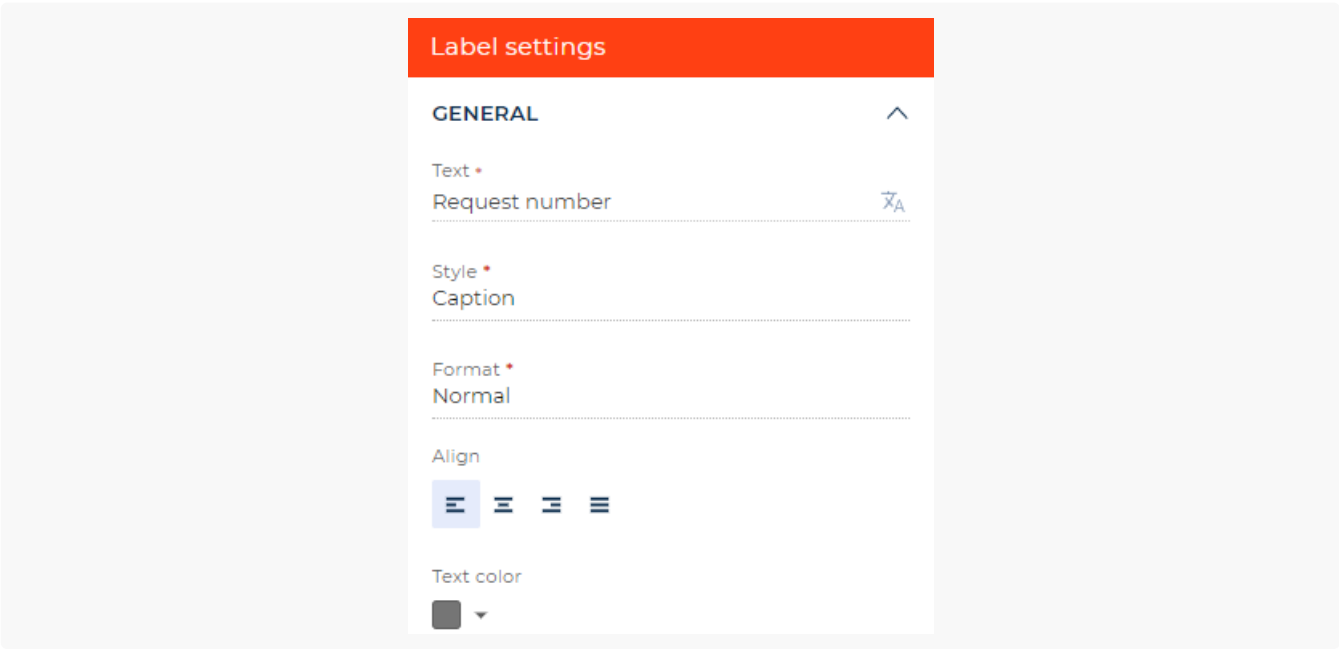
3. Open the [*Requests form page*] page in the workspace of the `Requests` app page.

   The [*Requests form page*] page includes the [*Name*] field by default.
4. Add a [*2 columns*] component to the Freedom UI Designer workspace.
5. Add a **request number label**.

   a. Add a component of the [*Label*] type to the first column of the [*2 columns*] component of the Freedom UI Designer.

   b. Click the button on the Freedom UI Designer's action panel and fill in the **label's properties** in the setup area:

   - Set [*Title*] to "`Create request`."
   - Select "`Caption`" in the [*Style*] property.
   - Select the gray color in the [*Text color*] property.



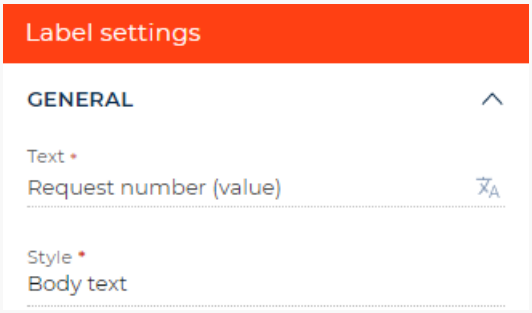
6. Add a **title that contains the value of the system setting with the request number**.

   a. Add a component of the [*Label*] type to the second column of the [*2 columns*] component of the Freedom UI Designer.

© 2023 Creatio. All rights reserved.

b. Click the button on the Freedom UI Designer's action panel and fill in the **title's properties** in the setup area:

- Set [*Title*] to "`Request number (value).`"
- Select "`Body text`" in the [*Style*] property.



7. Click the button on the Freedom UI Designer's action bar. The source code of the Freedom UI page opens after you save the page settings.

2. Change the origin of the query handler call

Configure the business logic in the Client Module Designer. In this example, change the origin of the query handler call.

1. Enable the `sdk. SysSettingsService` system variable service. To do this, add the `@creatio-devkit/common` dependency to the AMD module.

AMD module dependencies

```
/* AMD module declaration. */
define("UsrAppRequests_FormPage", /**SCHEMA_DEPS*/["@creatio-devkit/common"] /**SCHEMA_DEPS*/
    return {
        ...
    };
});
```

2. Add the `UsrRequestLastNumber` attribute to the `viewModelConfig` schema section. The attribute stores the value of the `UsrRequestLastNumber` system setting.

`viewModelConfig` schema section

```
viewModelConfig: /**SCHEMA_VIEW_MODEL_CONFIG*/{
    "attributes": {
        ...,
        /* The attribute that stores the value of the UsrRequestLastNumber system setting. */
        "UsrRequestLastNumber": {}
```

© 2023 Creatio. All rights reserved.

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

3. Bind the `caption` property of the `RequestNumberValue` element to the `$UsrRequestLastNumber` model attribute in the `viewConfigDiff` schema section. The `caption` property is responsible for the text that the element contains.

`viewConfigDiff` schema section

```
viewConfigDiff: /**SCHEMA_VIEW_CONFIG_DIFF*/[
    ...,
    {
        "operation": "insert",
        "name": "RequestNumberValue",
        "values": {
            ...,
            /* The property responsible for the text that the element contains. Bound to the
            "caption": "$UsrRequestLastNumber",
            ...
        },
        ...
    }
]/**SCHEMA_VIEW_CONFIG_DIFF*/,
```

4. Add a custom implementation of the `crt.SaveRecordRequest` system query handler to the `handlers` schema section. The handler is called by clicking the [*Save*] button on the request page toolbar.

   - Implement waiting for the execution of the next handler. In this example, it is the handler that saves the record.
   - Instantiate the system value service from `@creatio-devkit/common` after saving the record.
   - Get the value of the `UsrRequestLastNumber` system setting.
   - Send a query to update the value of the `UsrRequestLastNumber` system setting (increases it by 1).
   - Update the value of the `UsrRequestLastNumber` attribute.
   - Return the result of the saving.

`handlers` schema section

```
handlers: /**SCHEMA_HANDLERS*/[
    {
        request: "crt.SaveRecordRequest",
        /* Implement the custom query handler. */
        handler: async (request, next) => {
            /* Wait for the next handler (in this case, the handler that saves the record) to
            const saveResult = await next.handle(request);
            /* Instantiate the system value service from @creatio-devkit/common after saving
```

© 2023 Creatio. All rights reserved.

```
            const sysSettingsService = new sdk.SysSettingsService();
            /* Retrieve the value of the UsrRequestLastNumber system setting. */
            const requestLastNumber = await sysSettingsService.getByCode('UsrRequestLastNumbe
            /* Send a query to update the value of the UsrRequestLastNumber system setting. *
            await sysSettingsService.update({
                code: 'UsrRequestLastNumber',
                /* The new value is larger than the previous one by 1. */
                value: ++requestLastNumber.value
            });
            /* Update the value of the UsrRequestLastNumber attribute. */
            request.$context.UsrRequestLastNumber = requestLastNumber.value;
            /* Return the result of the saving. */
            return saveResult;
        }
    }
] /**SCHEMA_HANDLERS*/,
```

Complete source code of the page schema

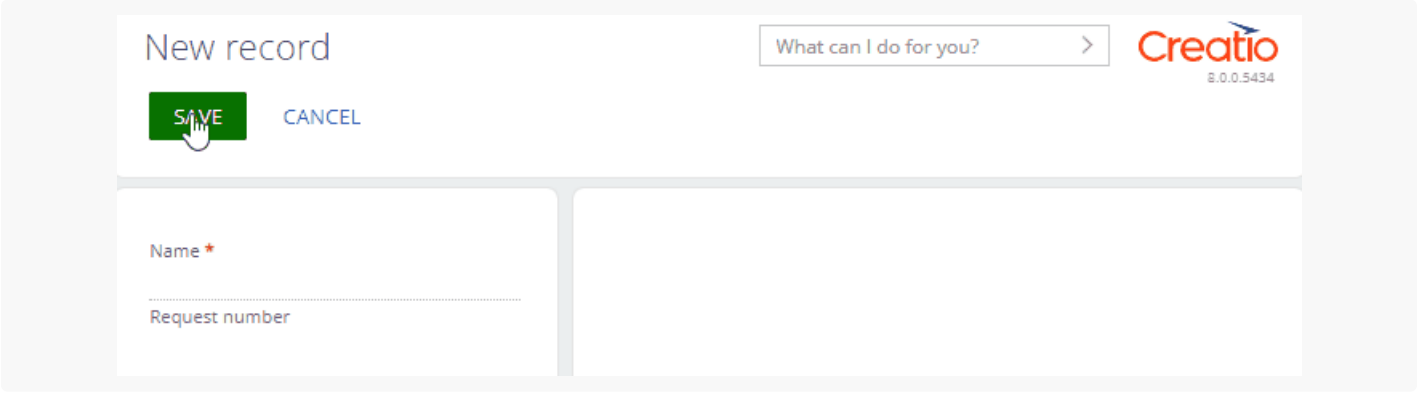
5. Click [*Save*] on the Client Module Designer's toolbar.

Outcome of the example

To **view the outcome of the example**:

1. Go to the `Requests` app page and click [*Run app*]
2. Click [*New*] on the `Requests` app toolbar.

As a result, when you click the [*Save*] button on the request page toolbar, Creatio will display the number of the last record created in the [*Requests*] section, increased by 1. The value is taken from the [*Request number*] (`UsrRequestLastNumber` code) system setting.



© 2023 Creatio. All rights reserved.