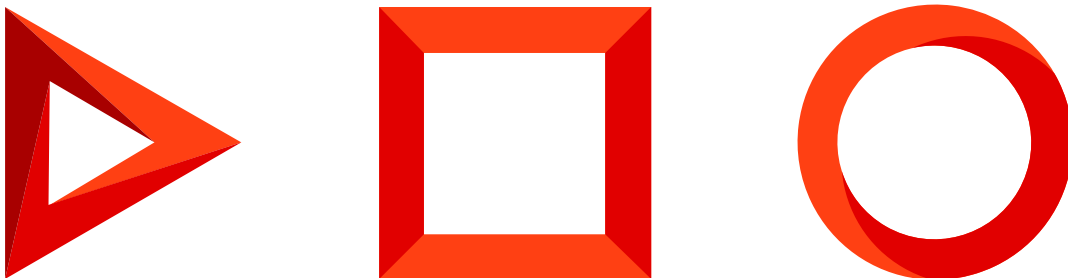


# Freedom UI page customization

Open a page from a custom handler

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.

# Table of Contents

<b>Open a page from a custom handler</b>	<b>4</b>
Open a record page from a custom handler	4
Open a Freedom UI page from a custom handler	5
<b>Open a record page from a custom handler</b>	<b>5</b>
1. Set up the UI of the pages	5
2. Set up the way record pages open	7
Outcome of the example	9
<b>Open a Freedom UI page from a custom handler</b>	<b>10</b>
1. Set up the page UI	10
2. Set up the way the Freedom UI page opens	11
Outcome of the example	12

# Open a page from a custom handler

## Beginner

Creatio 8 Atlas uses the `sdk.HandlerChainService` service to open pages. Creatio 7.X and Creatio 8 Atlas use the same method to open record pages. You can pass the needed default column values when Creatio adds a record.

Creatio 8 Atlas provides the following **actions** to open pages from a custom handler:

- Open a record page from a custom handler.
- Open a Freedom UI page from a custom handler.

## Open a record page from a custom handler

1. Add a page button that opens the record page on click at step 1 of the [Freedom UI page customization procedure](#) if needed.
2. Set up **how to open the record page from a custom handler** at step 2 of the [Freedom UI page customization procedure](#).
  - a. Enable the `sdk.HandlerChainService` service that opens pages. Enable the service similarly to the [display procedure for the value of system variables](#).
  - b. Bind the `clicked` property to the corresponding query in the `viewConfigDiff` schema section. Describe the business logic that opens the page in the `handlers` schema section. The `clicked` property is responsible for the action performed on button click.

View an example that binds the `clicked` property to the `usr.SomeRequest` custom query below.

`viewConfigDiff` [schema section](#)

- c. Add the implementation of a custom query to the `handlers` schema section.

To **open a page**:

- a. Get an instance of the `sdk.HandlerChainService` singleton service that opens pages.
- b. Send a `crt.UpdateRecordRequest` system query that opens the page by the specified ID.

View an example of the `usr.SomeRequest` query handler that sends the `crt.UpdateRecordRequest` system query below. The `crt.UpdateRecordRequest` query opens the page of the `SomeSchemaName` record with the `SomeRecordId` ID.

`handlers` [schema section](#)

To **open the page and populate the fields with the specified values**:

- a. Get an instance of the `sdk.HandlerChainService` singleton service that opens pages.

- b. Send the `crt.CreateRecordRequest` system query that creates a page with fields populated with the specified values.

View an example of the `usr.SomeRequest` query handler that sends the `crt.CreateRecordRequest` system query below. The `crt.CreateRecordRequest` query opens the `SomeSchemaName` record page and populates the `[ SomeField ]` field with the "SomeRecordId" value.

`handlers` [schema section](#)

View a detailed example that opens a record page in a separate article: [Open a record page from a custom handler](#).

## Open a Freedom UI page from a custom handler

1. Take steps 1-2 from the [procedure to open the record page from a custom handler](#).
2. Set up **how to open a Freedom UI page from a custom handler** at step 2 of the [procedure for opening the record page from a custom handler](#). To do this, add the implementation of a custom query to the `handlers` schema section.
  - a. Get an instance of the `sdk.HandlerChainService` singleton service that opens pages.
  - b. Send a `crt.OpenPageRequest` system query that opens the Freedom UI page with the specified name.

View an example `usr.SomeRequest` query handler that sends the `crt.OpenPageRequest` system query below. The `crt.OpenPageRequest` query opens the `SomePageName` page.

`handlers` [schema section](#)

View a detailed example that opens a Freedom UI page in a separate article: [Open a Freedom UI page from a custom handler](#).

## Open a record page from a custom handler

 Medium

**Example.** Add the following buttons to the record page of the custom `[ Handler Chain Service ]` section:



- `[ Edit contact ]`. Must open the page of the contact that has the specified ID.
- `[ Create request ]`. Must open the page of a new request in the custom `[ Requests ]` section. The `[ Name ]` field must be populated with the "New request" value.

Records open similarly in Creatio versions 7.X and 8.X. When Creatio adds a new record, you can pass the needed default column values.


### 1. Set up the UI of the pages

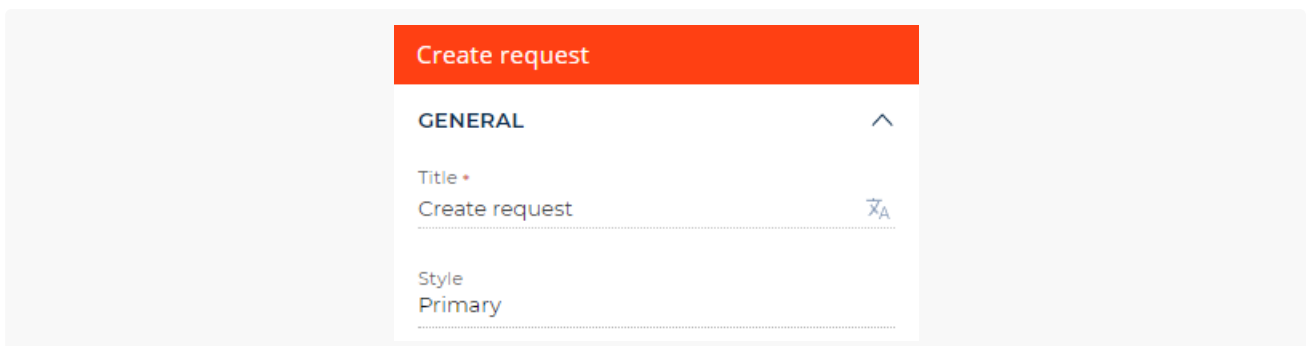
1. Set up the UI of the custom [ *Requests* ] **section page**. Create a custom `Requests` app based on the [ *Records & business processes* ] template. To do this, follow the guide in the user documentation: [Create a custom app](#).

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

2. Set up the UI of the custom [ *Handler Chain Service* ] **section page**.
  - a. Click  on the `Requests` app page.
  - b. Create a custom `Handler Chain Service` app based on the [ *Records & business processes* ] template. To do this, follow the guide in the user documentation: [Create a custom app](#).
  - c. Open the [ *Handler Chain Service form page* ] page in the working area of the `Handler Chain Service` app page.
  - d. Delete the [ *Name* ] field the [ *Handler Chain Service form page* ] page includes by default.
  - e. Add a **button that opens the page of the contact that has the specified ID**.
    - a. Add a [ *Button* ] type component to the toolbar of the Freedom UI Designer.
    - b. Click  in the action panel of the Freedom UI Designer and set the [ *Title* ] button property in the setup area to "Edit contact."



- f. Add a **button that opens the page of a new request** in the custom [ *Requests* ] section.
  - a. Add a [ *Button* ] type component to the toolbar of the Freedom UI Designer.
  - b. Click  in the action panel of the Freedom UI Designer and fill out the **button properties** in the setup area.
    - Set [ *Title* ] to "Create request."
    - Select "Primary" in the [ *Style* ] property.



- g. Click  in the action panel of the Freedom UI Designer. After you save the page settings, Creatio

opens the source code of the Freedom UI page.

## 2. Set up the way record pages open

Configure the business logic in the Client Module Designer. For this example, set up the way record pages open.

1. Enable the `sdk.HandlerChainService` service that opens pages. To do this, add `@creatio-devkit/common` to the AMD module as a dependency.

### AMD module dependencies

```
/* Declare the AMD module. */
define("UsrAppHandlerChainSe_FormPage", /**SCHEMA_DEPS*/["@creatio-devkit/common"] /**SCHEMA_
    ...
    });
});
```

2. Change the `clicked` **property value** in the `viewConfigDiff` schema section to the following:

- `usr.EditContactRequest` for the `EditContactButton` element
- `usr.CreateUsrRequestRequest` for the `CreateRequestButton` element

The `clicked` property handles the action executed on button click.

### viewConfigDiff schema section

```
viewConfigDiff: /**SCHEMA_VIEW_CONFIG_DIFF*/[
  {
    "operation": "insert",
    "name": "EditContactButton",
    "values": {
      ...,
      "clicked": {
        /* Bind the sending of the custom usr.EditContactRequest query to the clicked
        "request": "usr.EditContactRequest"
      }
    },
    ...
  },
  {
    "operation": "insert",
    "name": "CreateRequestButton",
    "values": {
      ...,
      "clicked": {
        /* Bind the sending of the custom usr.CreateUsrRequestRequest query to the cl
        "request": "usr.CreateUsrRequestRequest"
```

```

    }
  },
  ...
}
]/**SCHEMA_VIEW_CONFIG_DIFF*/,

```

### 3. Implement custom query handlers in the `handlers` schema section:

- `usr.EditContactRequest`
  - Retrieve the instance of the `sdk.HandlerChainService` singleton service that opens pages.
  - Send the `crt.UpdateRecordRequest` system query that opens the page of the contact that has the specified ID. You can view the ID of the contact whose page to open in the browser address bar. For this example, open the page of the Alexander Wilson contact whose ID is "98dae6f4-70ae-4f4b-9db5-e4fcb659ef19."
- `usr.CreateUsrRequestRequest`
  - Retrieve the instance of the `sdk.HandlerChainService` singleton service that opens pages.
  - Send the `crt.CreateRecordRequest` system query that opens the page of a new request. Populate the `[ Name ]` field with the "New request" value.

#### handlers **schema section**

```

handlers: /**SCHEMA_HANDLERS*/[
  {
    request: "usr.EditContactRequest",
    /* Implement the custom query handler. */
    handler: async (request, next) => {
      /* Retrieve the instance of the singleton service that opens pages. */
      const handlerChain = sdk.HandlerChainService.instance;
      /* Send the crt.UpdateRecordRequest system query that opens the page of the conta
      await handlerChain.process({
        type: 'crt.UpdateRecordRequest',
        entityName: 'Contact',
        recordId: '98dae6f4-70ae-4f4b-9db5-e4fcb659ef19',
        $context: request.$context
      });
      /* Call the next handler if it exists and return its result. */
      return next?.handle(request);
    }
  },
  {
    request: "usr.CreateUsrRequestRequest",
    /* Implement the custom query handler. */
    handler: async (request, next) => {
      /* Retrieve the instance of the singleton service that opens pages. */
      const handlerChain = sdk.HandlerChainService.instance;

```



```

    /* Send the crt.CreateRecordRequest system query that opens the page of a new req
await handlerChain.process({
  type: 'crt.CreateRecordRequest',
  entityName: 'UsrAppRequests',
  defaultValues: [{
    attributeName: 'UsrName',
    value: 'New request'
  }],
  $context: request.$context
});
/* Call the next handler if it exists and return its result. */
return next?.handle(request);
}
}
] /**SCHEMA_HANDLERS*/,

```

[Complete source code of the page schema](#)

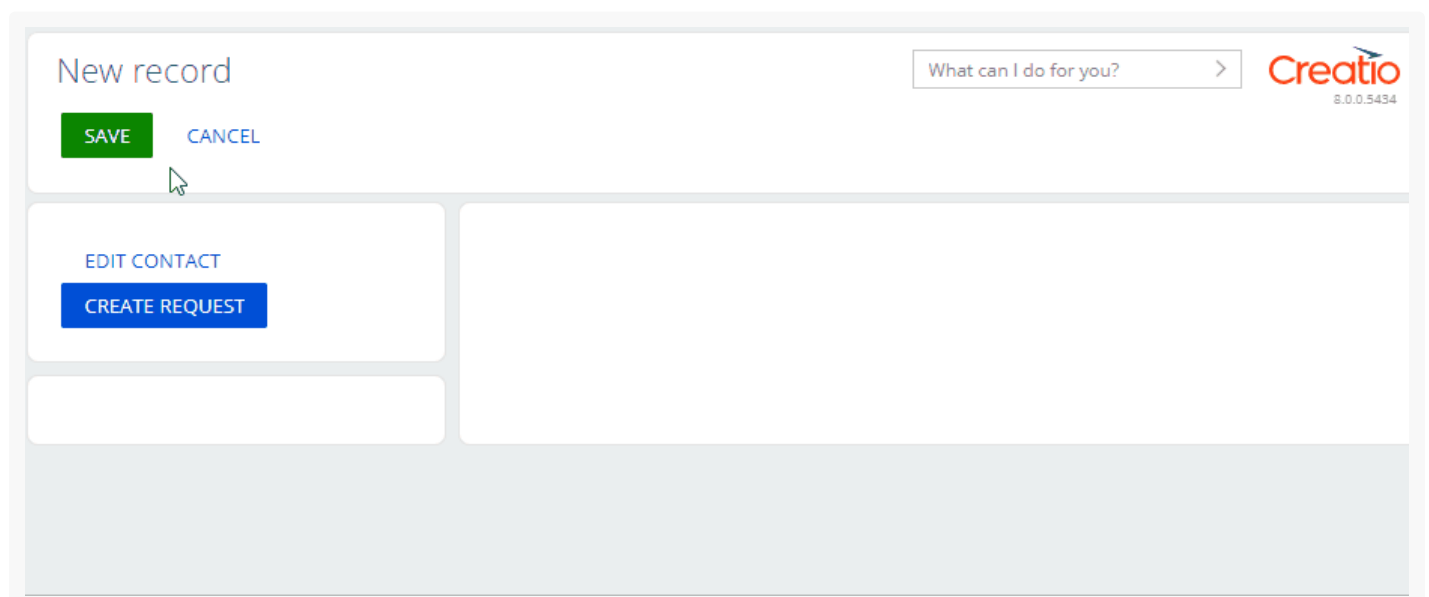
4. Click [ Save ] on the Client Module Designer's toolbar.

## Outcome of the example

To **view the outcome of the example that opens the contact page**:

1. Open the `Handler Chain Service` app page and click [ *Run app* ].
2. Click [ *New* ] on the `Handler Chain Service` app toolbar.
3. Click [ *Edit contact* ] on the record page of the custom `Handler Chain Service` section.

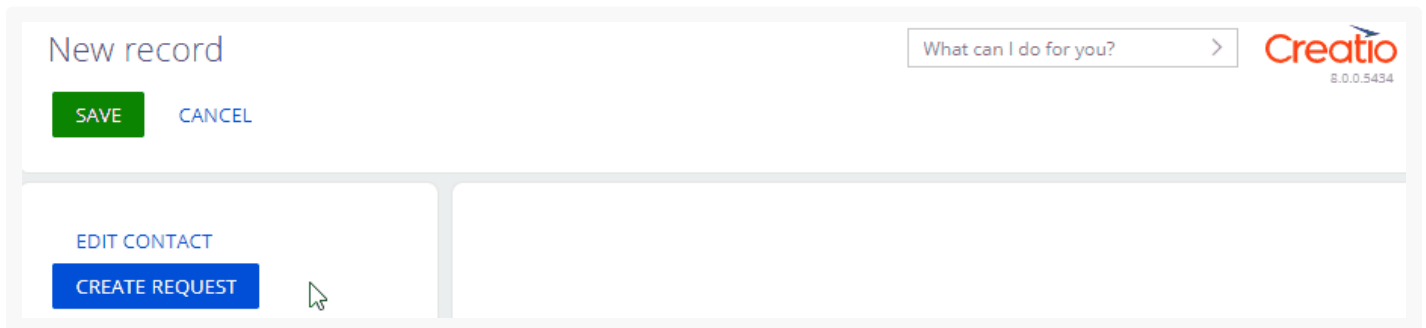
As a result, Creatio will open the page of the Alexander Wilson contact whose ID is "98dae6f4-70ae-4f4b-9db5-e4fcb659ef19."



To **view the outcome of the example that opens the request page and populates the field**:

1. Refresh the `Handler Chain Service` app page.
2. Click [ *New* ] on the `Handler Chain Service` app toolbar.
3. Click [ *Create request* ] on the record page of the custom `Handler Chain Service` section.

As a result, Creatio will open the page of a new request in the custom [ *Requests* ] section. The [ *Name* ] field will be populated with the "New request" value.




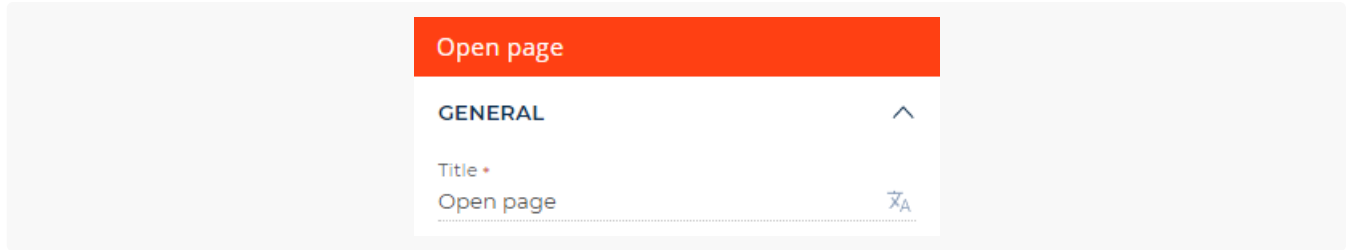
# Open a Freedom UI page from a custom handler

 Medium

**Example.** Add an [ *Open page* ] button to the record page of the custom [ *Handler Chain Service* ] section. The button must open the `StudioHomePage` Freedom UI page.

## 1. Set up the page UI

1. Create a custom `Handler Chain Service` app based on the [ *Records & business processes* ] template. To do this, follow the guide in the user documentation: [Create a custom app](#).
2. Open the [ *Handler Chain Service form page* ] page in the working area of the `Handler Chain Service` app page.
3. Add a **button** that opens the `StudioHomePage` page.
  - a. Add a [ *Button* ] type component to the toolbar of the Freedom UI Designer.
  - b. Click  in the action panel of the Freedom UI Designer and set the [ *Title* ] button property in the setup area to "Open page."



4. Click  in the action panel of the Freedom UI Designer. After you save the page settings, Creatio opens the source code of the Freedom UI page.

## 2. Set up the way the Freedom UI page opens

Configure the business logic in the Client Module Designer. For this example, set up the way the Freedom UI page opens.

1. Enable the `sdk.HandlerChainService` service that opens pages. To do this, add `@creatio-devkit/common` to the AMD module as a dependency.

### AMD module dependencies

```
/* Declare the AMD module. */
define("UsrAppHandlerChainSe_FormPage", /**SCHEMA_DEPS*/["@creatio-devkit/common"] /**SCHEMA_
    ...
    });
});
```

2. Change the `clicked` property value for the `OpenPageButton` element in the `viewConfigDiff` schema section to `usr.OpenUsrTestPageRequest`. The `clicked` property handles the action executed on button click.

### viewConfigDiff schema section

```
viewConfigDiff: /**SCHEMA_VIEW_CONFIG_DIFF*/[
    {
        "operation": "insert",
        "name": "OpenPageButton",
        "values": {
            ...,
            "clicked": {
                /* Bind the sending of the custom usr.OpenUsrTestPageRequest query to the cli
                "request": "usr.OpenUsrTestPageRequest"
            }
        },
        ...
    }
]/**SCHEMA_VIEW_CONFIG_DIFF*/,
```

3. Implement the `usr.OpenUsrTestPageRequest` custom query handler in the `handlers` schema section.
  - a. Retrieve the instance of the `sdk.HandlerChainService` singleton service that opens pages.
  - b. Send the `crt.OpenPageRequest` system query that opens the `StudioHomePage` page.

#### handlers schema section

```
handlers: /**SCHEMA_HANDLERS*/[
  {
    request: "usr.OpenUsrTestPageRequest",
    /* Implement the custom query handler. */
    handler: async (request, next) => {
      /* Retrieve the instance of the singleton service that opens pages. */
      const handlerChain = sdk.HandlerChainService.instance;
      /* Send the crt.OpenPageRequest system query that opens the Freedom UI page. */
      await handlerChain.process({
        type: 'crt.OpenPageRequest',
        schemaName: 'StudioHomePage',
        $context: request.$context
      });
      /* Call the next handler if it exists and return its result. */
      return next?.handle(request);
    }
  }
] /**SCHEMA_HANDLERS*/,
```

[Complete source code of the page schema](#)

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

## Outcome of the example

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

1. Open the `Handler Chain Service` app page and click [ *Run app* ].
2. Click [ *New* ] on the `Handler Chain Service` app toolbar.
3. Click [ *Open Page* ] on the record page of the custom `Handler Chain Service` section.

As a result, Creatio will open the `StudioHomePage` Freedom UI page.

