

Freedom UI page customization

Send a web service request and handle the response

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

Send a web service request and handle the response	4
Send a request to an external web service and handle its result on a page	4
1. Set up the page UI	5
2. Send the web service request and handle its results	6
Outcome of the example	8

Send a web service request and handle the response

Beginner

Creatio 8 Atlas uses the `sdk.HttpClientService` service to send web service requests.

To **send a web service request and handle the response on the page**:

1. Add a page inscription to display the handled result of a web service request at step 1 of the [Freedom UI page customization procedure](#) if needed.
2. Set up **how to send a web service request and handle the response on the page** at step 2 of the [Freedom UI page customization procedure](#).
 - a. Enable the `sdk.HttpClientService` service that sends HTTP requests. Enable the service similarly to the [display procedure for the value of system variables](#).
 - b. Add an attribute that stores data to the `viewModelConfig` schema section. Add the attribute similarly to the [setup procedure for the field display condition](#).
 - c. Bind the `caption` property to the corresponding model attribute in the `viewConfigDiff` schema section. Property binding is similar to that described in the [setup procedure for the field display condition](#). Instead of the `visible` property, use the `caption` property responsible for the text displayed in the element.
 - d. Add a custom implementation of the `ctrl.HandlerViewModelInitRequest` system query handler to the `handlers` schema section. The handler is executed when the `View` model is initialized.
 - a. Instantiate the HTTP client from `@creatio-devkit/common`.
 - b. Specify the URL to get the required information. If a web service request is sent using a non-absolute path (without `https://` or `http://` prefixes), this is a request to an internal Creatio web service. In that case, Creatio automatically adds the address of the current app to the link.
 - c. Send a GET request.
 - d. Retrieve the required values from the response and write them to the corresponding attributes.

View an example of the `ctrl.HandlerViewModelInitRequest` request handler that sends a request to the `https://SomeUrlValue` web service, receives the `someValue` parameter from the response body, and writes the parameter to the `SomeAttributeName` attribute, below.

`handlers` [schema section](#)


View a detailed example that sets up a web service request and handles the response in a separate article: [Send a request to an external web service and handle its result on a page](#).

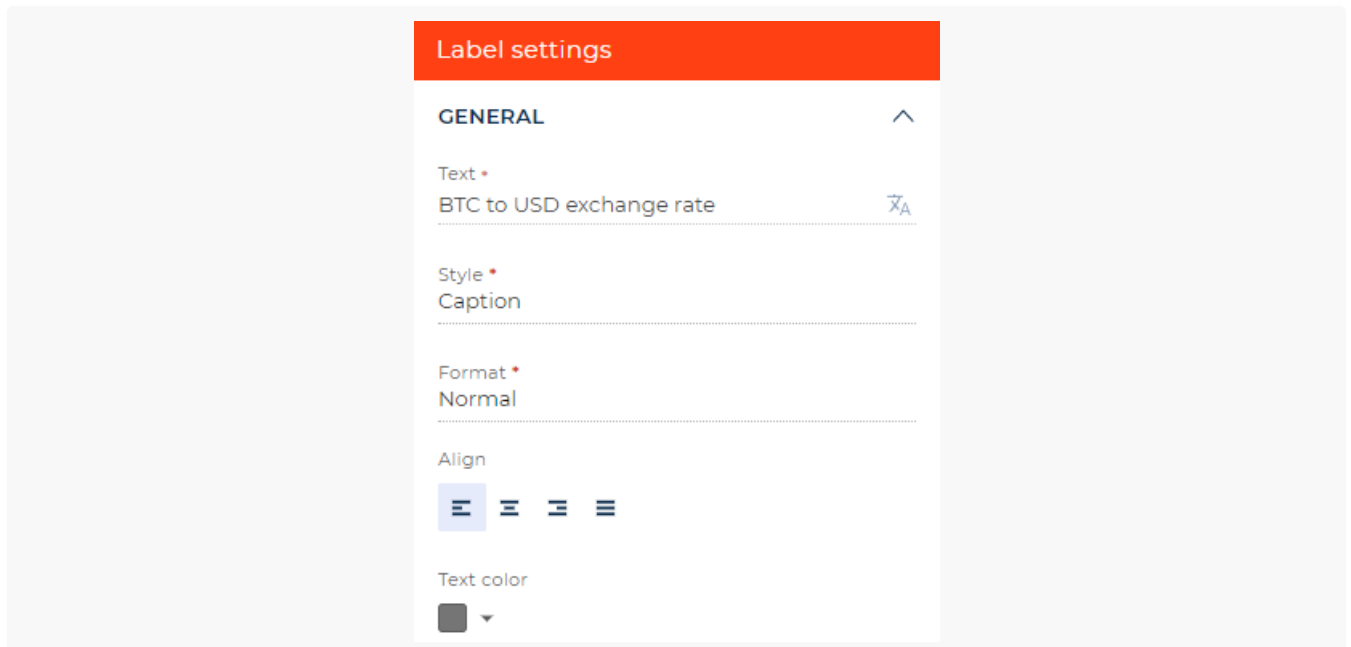
Send a request to an external web service and handle its result on a page




Example. Call the `coindesk.com` external web service from the record page of the [*Http Client Service*] custom section. Display the current BTC to the USD exchange rate.

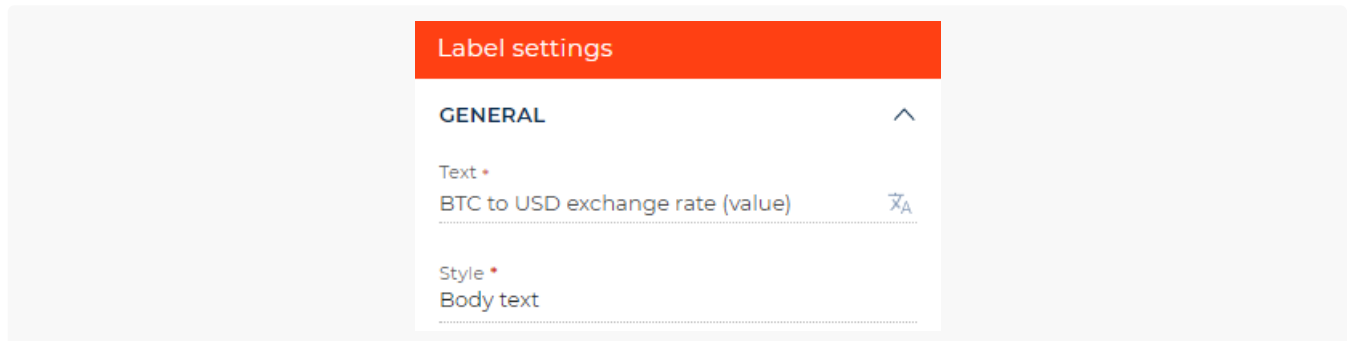
1. Set up the page UI

1. Create a custom `Http Client 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 [*Http Client Service form page*] page in the working area of the `Http Client Service` app page.
3. Delete the [*Name*] field the [*Http Client Service form page*] page includes by default.
4. Add a **label of the BTC to the USD exchange rate**.
 - a. Add a [*Label*] type component to the working area of the Freedom UI Designer.
 - b. Click  in the action panel of the Freedom UI Designer and fill out the **label properties** in the setup area.
 - Set [*Title*] to "BTC to USD exchange rate."
 - Select "Caption" in the [*Style*] property.
 - Select gray in the [*Text color*] property.



5. Add a **label that contains the value of the BTC to the USD exchange rate**.
 - a. Add a [*Label*] type component to the working area of the Freedom UI Designer.
 - b. Click  in the action panel of the Freedom UI Designer and fill out the **label properties** in the setup area.

- Set [*Title*] to "BTC to USD exchange rate (value)."
- Select "Body text" in the [*Style*] property.



6. 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. Send the web service request and handle its results

Configure the business logic in the Client Module Designer. For this example, send the web service request and handle its results.

1. Enable the `sdk.HttpClientService` service that sends HTTP requests. To do this, add `@creatio-devkit/common` to the AMD module as a dependency.

AMD module dependencies

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

2. Add the `BtcToUsd` attribute that stores data of the bitcoin to dollar exchange rate to the `viewModelConfig` schema section.

viewModelConfig schema section

```
viewModelConfig: /**SCHEMA_VIEW_MODEL_CONFIG*/{
    "attributes": {
        ...,
        /* The attribute that stores the bitcoin to dollar exchange rate. */
        "BtcToUsd": {}
    }
}/**SCHEMA_VIEW_MODEL_CONFIG*/,
```

3. Bind the `caption` property of the `BtcToUsdExchangeRateValue` element to the `$BtcToUsd` model attribute in the

`viewConfigDiff` schema section. The `caption` property handles the text contained in the element.

`viewConfigDiff` schema section

```
viewConfigDiff: /**SCHEMA_VIEW_CONFIG_DIFF*/[
  ...,
  {
    "operation": "insert",
    "name": "BtcToUsdExchangeRateValue",
    "values": {
      ...,
      /* Bind the BtcToUsd attribute to the caption property. */
      "caption": "$BtcToUsd",
      ...
    },
  },
  ...
]/**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 Creatio initializes the `View` model.
 - a. Create an instance of the HTTP client from `@creatio-devkit/common`.
 - b. Specify the URL to retrieve the current rate. Use the `coindesk.com` external web service.
 - c. Send a GET request.
 - d. Retrieve the rate from the response and specify the rate in the `BtcToUsd` attribute.

`handlers` schema section

```
handlers: /**SCHEMA_HANDLERS*/[
  {
    request: "crt.HandlerViewModelInitRequest",
    /* The custom implementation of the system query handler. */
    handler: async (request, next) => {
      /* Create an instance of the HTTP client from @creatio-devkit/common. */
      const httpClientService = new sdk.HttpClientService();
      /* Specify the URL to retrieve the current rate. Use the coindesk.com external se
      const endpoint = "https://api.coindesk.com/v1/bpi/currentprice/USD.json";
      /* Send a GET request. The HTTP client converts the response body from JSON to a
      const response = await httpClientService.get(endpoint);
      /* Retrieve the rate from the response and specify the rate in the BtcToUsd attri
      request.$context.BtcToUsd = response.body.bpi.USD.rate;
      /* Call the next handler if it exists and return its result. */
      return next?.handle(request);
    },
  },
],
```

```
] /**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. Open the `Http Client Service` app page and click [*Run app*].
2. Click [*New*] on the `Http Client Service` app toolbar.

As a result, Creatio will display the current BTC to the USD exchange rate on the record page of the [*Http Client Service*] custom section. The value will be retrieved from the `coindesk.com` external web service.

