Creatio Academy

Front-end development Freedom UI

Client schema

Version 8.0

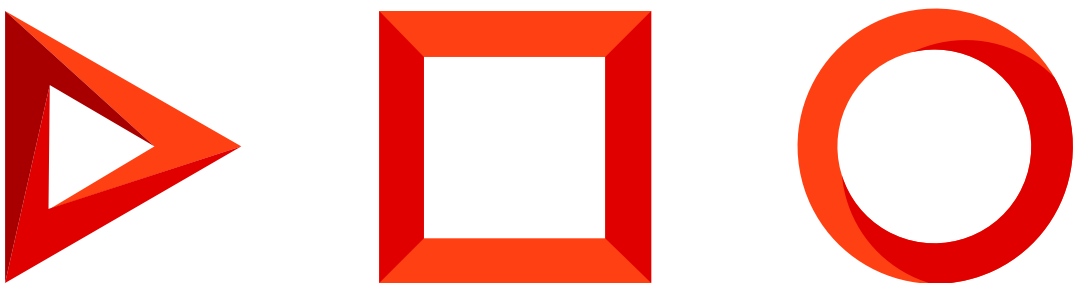


Table of Contents

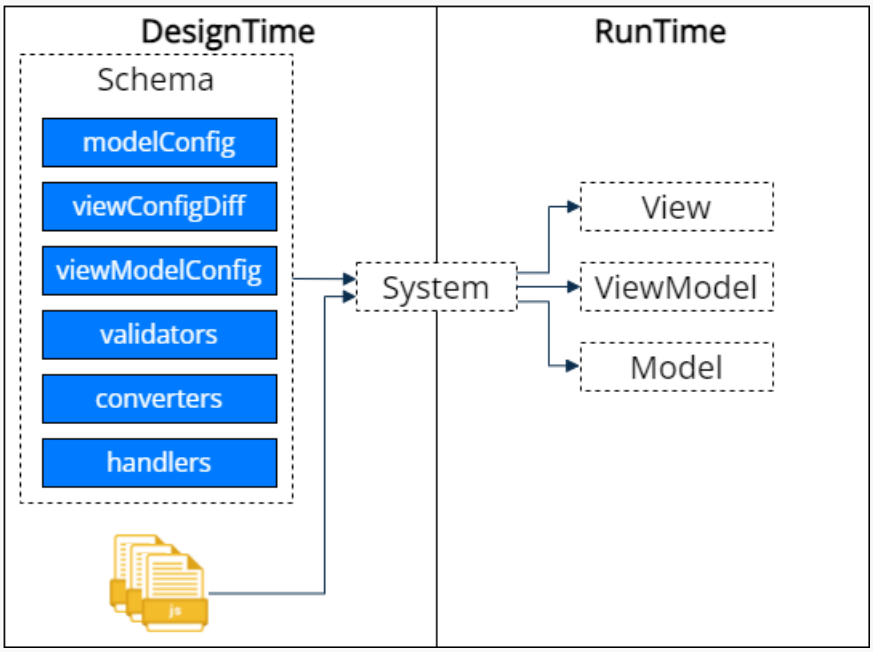
Client schema

Beginner

The **client schema** is a client module schema required to implement the front-end part of Creatio. Client schema is an element of a Creatio package. The **purpose** of the client schema is to save and deploy the metadata of Creatio parts (application, section, pop-up box, etc.). Learn more about module types and their specificities in a separate article: Module types and their specificities.

The [*Freedom UI page*] type of the client module schema represents the client schema for Creatio 8 Atlas and later. Learn more about creating a Freedom UI page in a separate article: Client module. Client schema belongs to the `Schema` layer (the metadata layer) of the `DesignTime` mode in the Creatio platform and contains the settings of the `RunTime` mode layers.

View the structural items of the `Schema` layer and its interaction pattern with other structural items in the Creatio platform in the figure below.



Learn more about `DesignTime` and `RunTime` modes in a separate article: Creatio front-end architecture.

The source code of client schemas has a generic structure available below.

Source code of a client schema

```
define("ExampleSchema", [], function() {
    return {
        modelConfig: /**SCHEMA_MODEL_CONFIG*/{}/**SCHEMA_MODEL_CONFIG*/,
        viewConfigDiff: /**SCHEMA_VIEW_CONFIG_DIFF*/[]/**SCHEMA_VIEW_CONFIG_DIFF*/,
        viewModelConfig: /**SCHEMA_VIEW_MODEL_CONFIG*/{}/**SCHEMA_VIEW_MODEL_CONFIG*/,
```

```
        validators: /**SCHEMA_VALIDATORS*/ {} /**SCHEMA_VALIDATORS*/,
        converters: /**SCHEMA_CONVERTERS*/ {} /**SCHEMA_CONVERTERS*/,
        handlers: /**SCHEMA_HANDLERS*/[] /**SCHEMA_HANDLERS*/
    };
});
```

Token comments of client schema sections are required.

View the **client schema sections** in the table below.

Client schema sections

Section	Description
`modelConfig`	Handles the description of the `Model` layer's data sources.
`viewConfigDiff`	Handles the generation of the `View` layer.
`viewModelConfig`	Handles the generation of the `ViewModel` layer. The layer contains the business logic for interaction of the `View` and `Model` layers.
`validators`	Functions that check whether the `ViewModel` attribute value is correct.
`converters`	Functions that modify the value of the `ViewModel` attribute bound to a property of the visual component.
`handlers`	Items of the `HandlerChain` mechanism which enables you to describe business logic as an action request and a chain of handlers.

Learn more about layers in a separate article: Front-end Creatio architecture.

Describe the client schema sections in `JSON`. Implement the behavior of the `validators`, `converters`, `handlers` schema sections in JavaScript.

Learn more about page customization in a separate article: Page customization.

converters schema section

Beginner

Converters are functions that modify the value of the `ViewModel` attribute bound to a property of the visual component. View an example of container use in a separate article: Implement the field value conversion on a page.

View the **base converters** that Creatio 8 Atlas provides below.

Base converters

`crt.ToBoolean`

Converts the values of other types to the `BOOLEAN` type.

Value

`value`	`ANY`	The incoming value. The value whose type is not `BOOLEAN`.
`result`	`BOOLEAN`	The outgoing value. The converted `BOOLEAN` type value.

`crt.InvertBooleanValue`

Converts the `BOOLEAN` type value to the opposite `BOOLEAN` type value. For example, if the incoming value is `true`, the converter returns `false`, and vice versa.

Value

`value`	`BOOLEAN`	The incoming value.
`result`	`BOOLEAN`	The outgoing value.

`crt.ToEmailLink`

Converts an email address to a link by adding the `mailTo:` prefix.

Value

`value`	`TEXT`	The incoming value. The email.
`result`	`TEXT`	The outgoing value. The link to the email.

`crt.ToObjectProp`

Retrieves the value of the specified object property.

Value

`value`	`CUSTOM_OBJECT`	The incoming value. The name of the object property.
`result`	`ANY`	The outgoing value. The value of the object property.

Parameters

`prop` **required**	`TEXT`	The name of the property to retrieve.
`defaultValue` **optional**	`ANY`	The value to return if the property does not exist or the response is `false`.

`crt.ToPhoneLink`

Converts the phone number to a link by adding the `tel:` prefix.

Value

`value`	`TEXT`	The incoming value. The phone number.
`result`	`TEXT`	The outgoing value. A link to the phone number.

validators schema section JS

Beginner

Validators are functions that check whether the `ViewModel` attribute value is correct. For example, whether the record field value matches the specified conditions. View an example of validator use in a separate article: Implement the field value validation on a page

View the **base validators** that Creatio 8 Atlas provides below.

Base validators

`crt.Required`

Flags the field as required.

`crt.MinLength`

Sets the minimum allowed string length.

Parameters

`minLength`	`INTEGER`	The minimum allowed string length.

`crt.MaxLength`

Sets the maximum allowed string length.

Parameters

`maxLength`	`INTEGER`	The maximum allowed string length.

`crt.Min`

Sets the minimum allowed value.

Parameters

`min`	`INTEGER`	The minimum allowed value.

`crt.Max`

Sets the maximum allowed value.

Parameters

`max`	`INTEGER`	The maximum allowed value.

`crt.EmptyOrWhiteSpace`

Flags the field as required. Field values must not consist of spaces only.

handlers schema section

Beginner

Query handlers are items of the `HandlerChain` mechanism which enables you to describe business logic as an action request and a chain of handlers. Examples of handler invocation are available in separate articles: Page customization.

Generic query handlers include actions to open pages, work with data on pages, etc.

Generic query handlers available in Creatio 8 Atlas are available below.

Generic query handlers

`crt.CreateRecordRequest`

Creates a record.

`crt.UpdateRecordRequest`

Updates a record.

`crt.OpenPageRequest`

Opens a page.

`crt.SaveRecordRequest`

Saves data.

`crt.CancelRecordChangesRequest`

Cancels a record update.

`crt.RunBusinessProcessRequest`

Runs a business process.

`crt.ClosePageRequest`

Closes a page.

`crt.HandleViewModelInitRequest`

Initializes a view model instance.

`crt.HandleViewModelResumeRequest`

Adds a `View` model to a container using the `attach` operation. Available in Creatio 8.0.6 Atlas and later.

Creatio calls the handler when you open a module that has the `viewModel` parameter.

`crt.HandleViewModelAttributeChangeRequest`

Changes an attribute of a view model instance.

`crt.HandleViewModelPauseRequest`

Removes the `View` model of the current module from a container using the `detach` operation. Available in Creatio 8.0.6 Atlas and later.

Creatio calls the handler after calling the `HandleViewModelResumeRequest` handler when you switch to another module and the current module has the `viewModel` parameter. Creatio does not call the handler if you open a Freedom UI page and switch to another module before executing the `attach` operation.

`crt.HandleViewModelDestroyRequest`

Destroys a view model instance. At this stage, execute only synchronous code that destroys resources accumulated as part of runtime.