

Клиентская схема

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

Клиентская схема	4
Секция validators	6
Базовые валидаторы	6
Секция converters	7
Базовые конвертеры	7

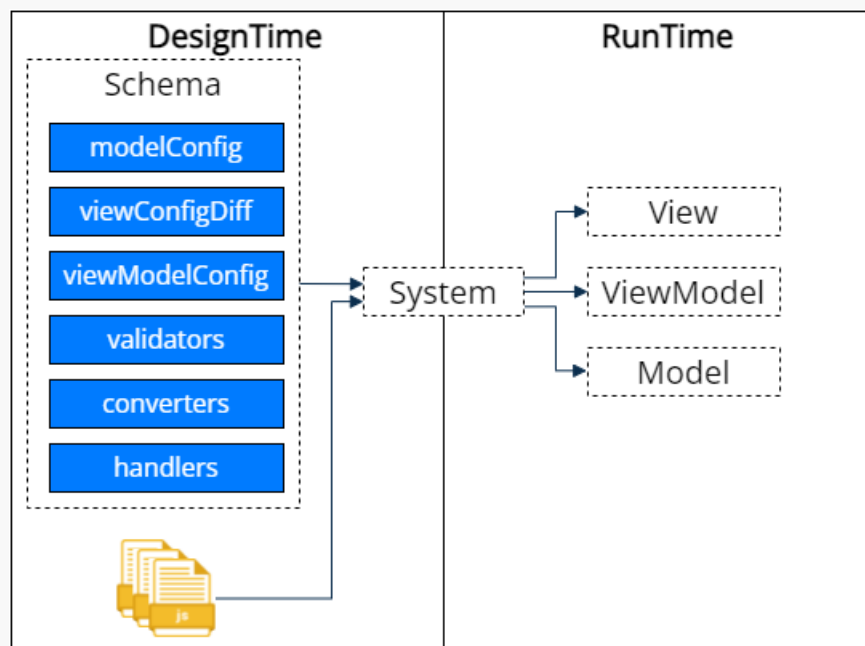
Клиентская схема

Основы

Клиентская схема — это схема клиентского модуля, с помощью которой реализуется front-end часть приложения. Элемент пакетной подсистемы Creatio. **Назначение** клиентской схемы — сохранение и поставка метаданных части системы (приложения, раздела, модального окна и т. д.). Типы модулей и их особенности описаны в статье [Виды модулей](#).

Клиентская схема принадлежит к слою `Schema` (слой метаданных) режима `DesignTime` платформы Creatio и содержит настройки слоев режима `RunTime`, которые доступны к изменению путем использования по-`code` инструментов.

Схема слоя `Schema` и его взаимодействие с другими структурными элементами платформы Creatio представлены на рисунке ниже.



Подробнее о режимах `DesignTime` и `RunTime` читайте в статье [Front-end архитектура Creatio](#).

Исходный код клиентских схем имеет общую структуру, которая представлена ниже.

Структура исходного кода клиентской схемы

```
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*/
  };
});

```

Маркерные комментарии к свойствам клиентской схемы обязательны к использованию.

Описание **свойств** конфигурационного объекта клиентской схемы представлено в таблице ниже.

Свойства клиентской схемы

Свойство	Способ разработки	Описание
<code>modelConfig</code>	Кастомизация	Содержит описание источников данных.
<code>viewConfigDiff</code>	Кастомизация	Отвечает за формирование слоя <code>View</code> .
<code>viewModelConfig</code>	Кастомизация	Отвечает за формирование слоя <code>ViewModel</code> — бизнес-логики взаимодействия слоев <code>View</code> и <code>Model</code> .
<code>validators</code>	Замещение	Валидаторы.
<code>converters</code>	Замещение	Конвертеры.
<code>handlers</code>	Замещение	Обработчики.

Свойства клиентской схемы реализуются в формате `JSON`. В `JSON`-массиве свойств `validators`, `converters`, `handlers` может быть использован `JavaScript`.

Бизнес-логика реализуется в схемах страниц Freedom UI с использованием no-code и low-code инструментов. Подробнее читайте в статье [Front-end архитектуры Creatio](#).

Компоненты бизнес-логики, JS-реализацию которых может содержать клиентская схема:

- Валидаторы.
- Конверторы.
- События и обработчики событий.

Бизнес-логика, которой можно управлять в компонентах:

- Видимость элементов.
- Блокировка элементов.
- Обязательность элементов.
- Фильтрация элементов.
- Заполнение полей.
- Обращение к данным Creatio.

- Отправка http-запросов.
- Переход по страницам.

Также Creatio предоставляет возможность выстраивать обработчики запросов в цепочки выполнения, например, чтобы по запросу сохранения записи сначала отработал базовый обработчик сохранения, а потом выполнялась пользовательская логика страницы.

Секция validators

Основы

Валидаторы — функции проверки корректности значения атрибута `ViewModel`. Например, проверка значения поля записи на соответствие установленным условиям. Пример использования валидатора приведен в статье [Реализовать валидацию значения поля на странице](#).

Базовые валидаторы, которые предоставляет Creatio 8 Atlas, представлены ниже.

Базовые валидаторы

`crd.Required`

Устанавливает обязательность заполнения поля.

`crd.MinLength`

Устанавливает минимально допустимое значение длины строки.

Параметры

`minLength`

INTEGER

Минимально допустимое значение длины строки.

`crd.MaxLength`

Устанавливает максимально допустимое значение длины строки.

Параметры

`maxLength`

INTEGER

Максимально допустимое значение длины строки.

`crd.Min`

Устанавливает минимально допустимое значение.

Параметры

min

INTEGER

Минимально допустимое значение.

crt.Max

Устанавливает максимально допустимое значение.

Параметры

max

INTEGER

Максимально допустимое значение.

crt.EmptyOrWhiteSpace

Устанавливает обязательность заполнения поля. Не допускается заполнение поля пробелами.

Секция converters

Основы

Конвертеры — функции модификации значения атрибута `viewModel`, который привязан к свойству визуального компонента, в другое значение. Пример использования конвертера приведен в статье [Реализовать конвертацию значения поля на странице](#).

Базовые конвертеры, которые предоставляет Creatio 8 Atlas, представлены ниже.

Базовые конвертеры

crt.ToBoolean

Конвертирует значения других типов в тип `BOOLEAN`.

Значения

value	ANY	Входящее значение. Значение, которое не является типом <code>BOOLEAN</code> .
result	BOOLEAN	Исходящее значение. Конвертированное значение типа <code>BOOLEAN</code> .

crt.InvertBooleanValue

Конвертирует значение типа `BOOLEAN` в противоположное значение типа `BOOLEAN`. Например, если входящее значение — `true`, то возвращается значение `false` и наоборот.

Значения

value	BOOLEAN	Входящее значение.
result	BOOLEAN	Исходящее значение.

`crt.ToEmailLink`

Конвертирует адрес электронной почты в ссылку, добавляя в начало адреса префикс `mailto:` .

Значения

value	TEXT	Входящее значение. Адрес электронной почты.
result	TEXT	Исходящее значение. Ссылка на адрес электронной почты.

`crt.ToObjectProp`

Получает значение указанного свойства объекта.

Значения

value	CUSTOM_OBJECT	Входящее значение. Имя свойства объекта.
result	ANY	Исходящее значение. Значение свойства объекта.

Параметры

prop required	TEXT	Имя свойства, которое необходимо получить.
defaultValue optional	ANY	Значение, которое возвращается, если свойства не существует или в ответе возвращено значение <code>false</code> .

`crt.ToPhoneLink`

Конвертирует номер телефона в ссылку, добавляя в начало адреса префикс `tel:` .

Значения

value	TEXT	Входящее значение. Номер телефона.
result	TEXT	Исходящее значение. Ссылка на номер телефона.