

Профиль связанной сущности

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

Профиль связанной сущности	4
Добавить профиль связанной сущности	4
Добавить пользовательский профиль связанной сущности на страницу записи	7
1. Создать схему модели представления связанного профиля контрагента	7
2. Создать схему замещающей модели представления страницы контакта	9
Результат выполнения примера	12
Схема BaseProfileSchema	12
Атрибуты	13
Сообщения	13
Методы	14
Схема BaseMultipleProfileSchema	14
Атрибуты	15
Сообщения	15
Схема BaseRelatedProfileSchema	15
Сообщения	16

Профиль связанной сущности



Профиль связанной сущности — элемент управления, который по умолчанию представляет собой информационный блок, наполняемый при загрузке страницы записи информацией о связанной сущности. В приложении элемент используется как профиль связанной записи на странице записи раздела. Например, при открытии страницы контакта в контейнере левой части страницы записи (`LeftModulesContainer`) отображается профиль контакта и информация о связанном с ним контрагенте. Контрагент связанный с контактом по колонке [*Account*] объекта [*Contact*]. Профиль записи и профиль связанной записи подробно описаны в статье [Страницы записей](#).

Составляющие, которые реализуют функциональность профиля связанной сущности:

- Класс `Profile` .
- `BaseProfileSchema` — базовая схема для создания профиля связанной сущности. Позволяет отобразить любой набор полей по связанной сущности, а также любое количество модулей. Является родительским классом для класса `Profile` . Все схемы профилей связанных сущностей должны наследовать схему `BaseProfileSchema` .
- `BaseMultipleProfileSchema` — базовая схема для создания профиля связанной сущности, который содержит в себе несколько профилей и свободно между ними переключается, пользуясь логикой выбора значений из справочников. Основное **отличие** от базового профиля — возможность встраивать другие профили в текущий профиль. При этом встроенные профили могут взаимодействовать друг с другом посредством сообщений. Профили `BaseMultipleProfileSchema` должны наследовать базовую схему `BaseRelatedProfileSchema` , которая реализует профили, зависящие или встроенные в другие профили.

Добавить профиль связанной сущности

1. Создайте схему модели представления профиля связанной сущности. Для этого воспользуйтесь инструкцией, которая приведена в статье [Разработка конфигурационных элементов](#).
2. В схеме модели представления настройте **профиль связанной сущности**.
 - a. В свойство `mixins` добавьте МИКСИН `ProfileSchemaMixin` .
 - b. В массив модификаций `diff` добавьте конфигурационный объект с настройками связанного пользовательского профиля.
3. Создайте схему замещающей модели представления страницы записи, на которой размещено поле. Для этого воспользуйтесь инструкцией, которая приведена в статье [Разработка конфигурационных элементов](#).
4. В схеме замещающей модели представления **добавьте связанный пользовательский профиль на страницу записи**.
 - a. В свойство `modules` добавьте модуль связанного пользовательского профиля контрагента. В


СВОЙСТВО `masterColumnName` СВОЙСТВА `viewModelConfig` добавьте название колонки, по которой выполняется связь связанного профиля с основной схемой страницы записи. Опираясь на значение этой колонки, класс `Profile` загружает данные.

- b. В массив модификаций `diff` добавьте конфигурационный объект с настройками расположения связанного пользовательского профиля контрагента.

По значению колонки, которая указана в свойстве `masterColumnName`, класс `Profile` загружает данные. На этапе инициализации объекта профиля класс `Profile`:

- Отправляет сообщение `GetColumnInfo` для получения дополнительной информации о колонке, по которой он связан (фильтры, заголовок и т. д.).
- Запрашивает значение колонки, по которой он связан.
- Если **связанный профиль не пустой** (т. е. в поле связи выбрана запись), то по этой записи инициализируются данные.
- Если **связанный профиль пустой** (т. е. в поле связи не выбрана запись), то на месте связанного профиля контрагента отображаются:
 - Название поля, по которому выполняется связь. Например, на странице контакта отображается связанный профиль контрагента, который связан с контактом по полю [*Контрагент*] ([*Account*]).
 - [*Добавить контрагент*] ([*New account*]) — создать новую запись в справочнике поля связи.]
 - [*Выбрать*] ([*Search*]) — выбрать существующую запись из списка доступных.

Профиль контакта, у которого присутствует связанный профиль контрагента, представлен на рисунке ниже.



100%

2:43 PM,
London

Full name*
Caleb Jones

Full job title
Managing Director

Mobile phone
+44 782 223 4967

Business phone
3010

Email
c.jones@yahoo.co.uk

Account

Our company

Type

Our company

Owner

John Best

NEXT STEPS (2)

Discuss contract 102-01 with Caleb

11/11/2021 | Symon Clarke

Prepare new customer report

11/15/2021 | John Best

CONTACT INFO
CONNECTED TO
MAINTENANCE
TIMELINE
ENGAGEMENT
WEBSITE EVENTS

Type Employee
Title Mr.
Recipient's name Jones
Age 42

Owner John Best
Gender Male
Preferred language

Communication options +

Business phone 3010
Email c.jones@yahoo.co.uk
Do not use fax ☒

Addresses +

Address type	Primary	Address	City	Country	ZIP/postal...
Home	Yes	153 Sunsine Street	London	United Kingdom	

Профиль контакта, у которого отсутствует связанный профиль контрагента, представлен на рисунке ниже.

NEXT STEPS (2)

- Call back to the client. Inform about case resolution
11/9/2021 | Megan Lewis
- Specify contract terms
11/17/2021 | John Best

CONTACT INFO | CONNECTED TO | MAINTENANCE | TIMELINE | ENGAGEMENT | WEBSITE EVENTS

Type: Customer | Owner: John Best
 Title: Ms. | Gender: Female
 Recipient's name: Nelson | Preferred language:
 Age: 39

Communication options + [Facebook] [Twitter]

Mobile phone: +44 (788) 247 1010 | Business phone: +44 (20) 3488 6553
 Email: christine@noveltycorpo... | Email: christine@novcorp.co.uk

Addresses + :

Address type	Primary	Address	City	Country	ZIP/postal...
Actual	Yes	1537 Green Lane	London	United Kingdom	

ACCOUNT
 New account
 Search

При очистке поля или изменении значения в объекте профиля выполняется переинициализация данных. При выборе существующей связанной сущности на справочник накладывается бизнес-логика, которая определена в странице записи и связана с атрибутом, который ссылается на эту сущность. Т. е. сохраняется фильтрация, настройки колонок запроса и т. д. Удаление атрибута из схемы страницы записи приведет к удалению бизнес-логики.

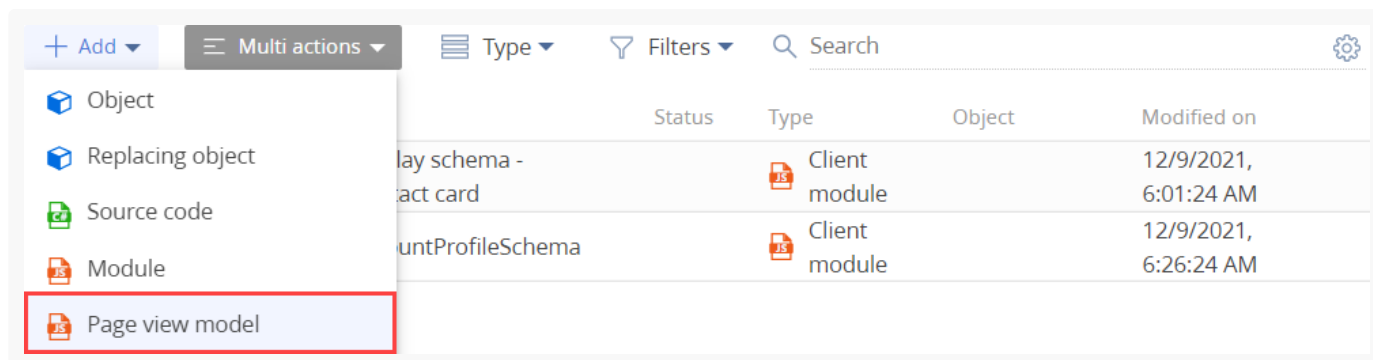
Добавить пользовательский профиль связанной сущности на страницу записи

Сложный

Пример. Добавить связанный пользовательский профиль контрагента на страницу контакта.

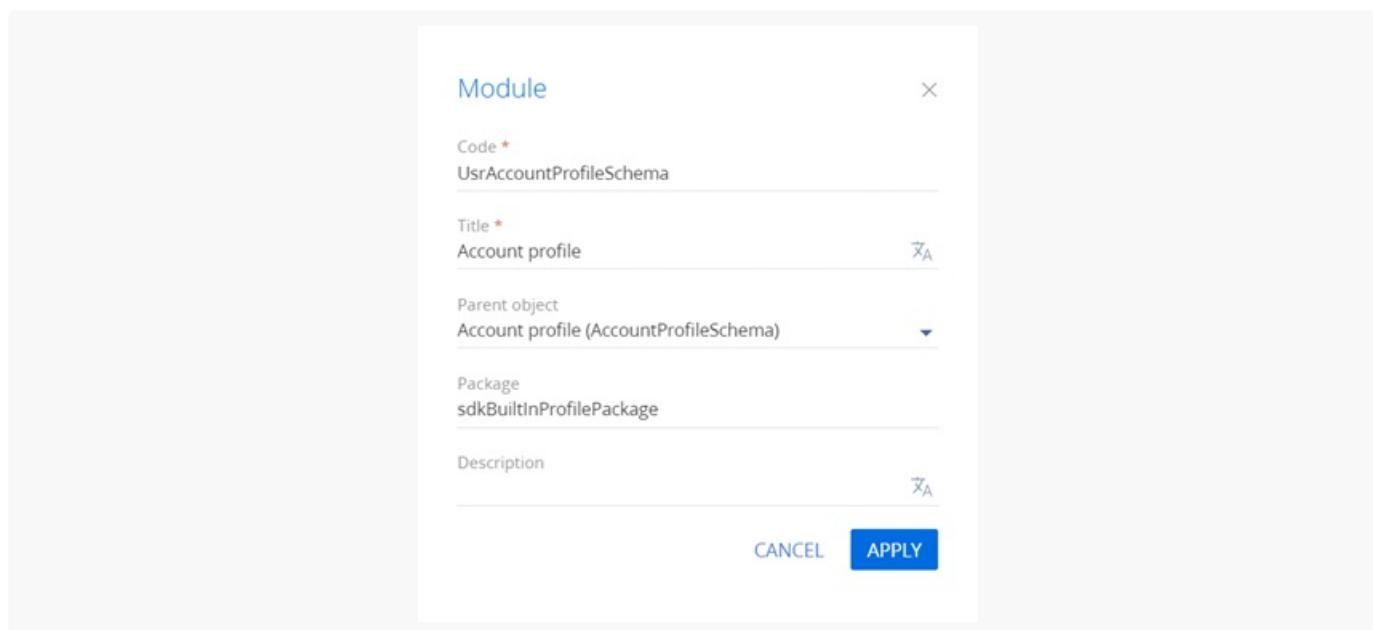
1. Создать схему модели представления связанного профиля контрагента

1. [Перейдите в раздел \[Конфигурация \]](#) ([Configuration]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [Добавить] —> [Модель представления страницы] ([Add] —> [Page view model]).



3. Заполните **свойства схемы**.

- [Код] ([Code]) — "UsrAccountProfileSchema".
- [Заголовок] ([Title]) — "Профиль контрагента" ("Account profile").
- [Родительский объект] ([Parent object]) — выберите "AccountProfileSchema".



4. В объявлении класса модели представления в качестве зависимостей добавьте миксин `ProfileSchemaMixin`.

5. Настройте **связанный пользовательский профиль контрагента**.

- В свойство `mixins` добавьте миксин `ProfileSchemaMixin`.
- В массив модификаций `diff` добавьте конфигурационный объект с настройками связанного пользовательского профиля контрагента.

Исходный код схемы модели представления связанного профиля контрагента представлен ниже.

UsrAccountProfileSchema

```
/* В качестве зависимостей укажите миксин ProfileSchemaMixin. */
define("UsrAccountProfileSchema", ["ProfileSchemaMixin"], function () {
```



```

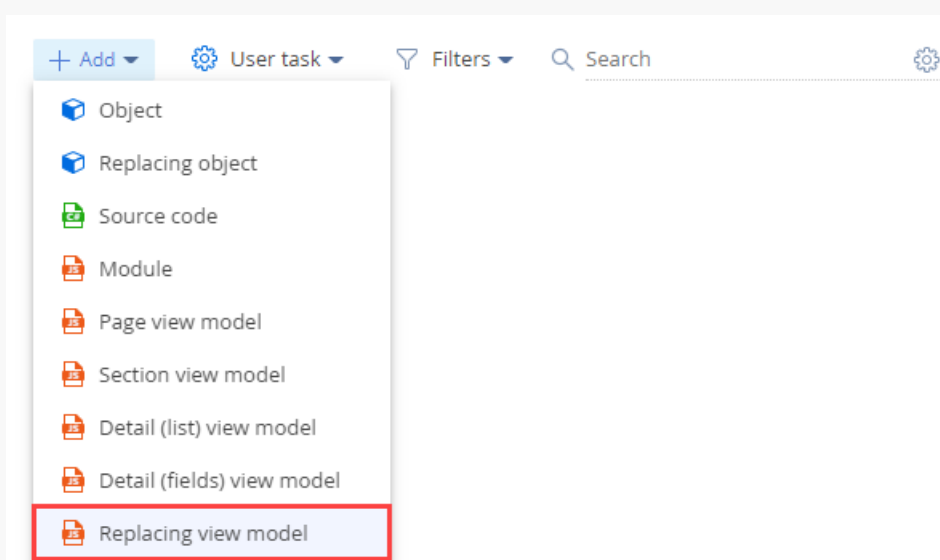
return {
    /* Название схемы объекта. */
    entitySchemaName: "Account",
    /* Миксины. */
    mixins: {
        /* Миксин, который содержит функции для получения иконок и картинок профиля. */
        ProfileSchemaMixin: "Terrasoft.ProfileSchemaMixin"
    },
    /* Массив модификаций diff. */
    diff: /**SCHEMA_DIFF*/[
        {
            /* Операция добавления. */
            "operation": "insert",
            /* Имя сущности. */
            "name": "Contact",
            /* Имя родительского элемента, в который выполняется вставка. */
            "parentName": "ProfileContentContainer",
            /* Свойство элемента родителя, с которым выполняется операция. */
            "propertyName": "items",
            /* Значение добавляемого элемента. */
            "values": {
                /* Привязка к значению свойства Account объекта Contact. */
                "bindTo": "Account",
                /* Конфигурация разметки. Позиционирование элемента. */
                "layout": {
                    "column": 3,
                    "row": 10,
                    "colSpan": 19
                }
            }
        }
    ]/**SCHEMA_DIFF*/
};
});

```

6. На панели инструментов дизайнера нажмите [*Сохранить*] ([*Save*]).

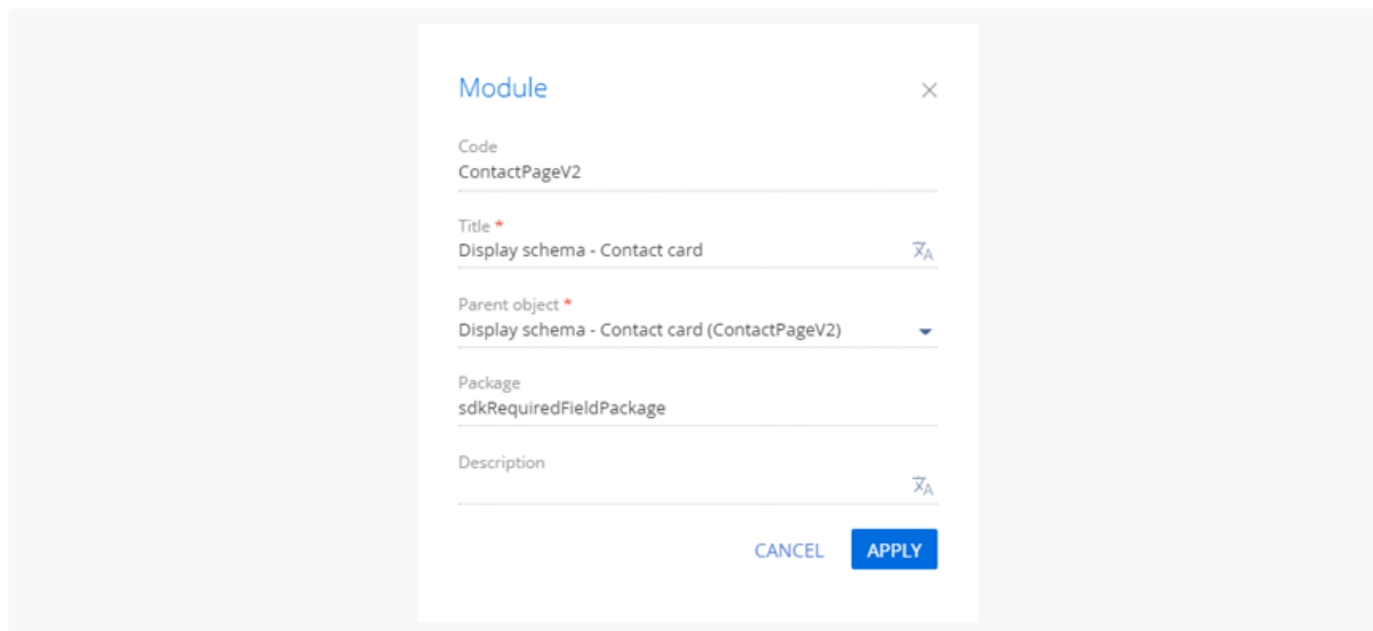
2. Создать схему замещающей модели представления страницы контакта

1. [Перейдите в раздел \[Конфигурация \]](#) ([*Configuration*]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [*Добавить*] —> [*Замещающая модель представления*] ([*Add*] —> [*Replacing view model*]).



3. Заполните **свойства схемы**.

- [Код] ([Code]) — "ContactPageV2".
- [Заголовок] ([Title]) — "Схема отображения карточки контакта" ("Display schema - Contact card").
- [Родительский объект] ([Parent object]) — выберите "ContactPageV2".



4. В объявлении класса модели представления в качестве зависимостей добавьте модули

BaseFiltersGenerateModule, BusinessRuleModule, ContactPageV2Resources, ConfigurationConstants, ContactCareer, DuplicatesSearchUtilitiesV2, UsrAccountProfileSchema.

5. Добавьте **связанный пользовательский профиль контрагента на страницу контакта**.

- В свойство `modules` добавьте модуль связанного пользовательского профиля контрагента.
- В массив модификаций `diff` добавьте конфигурационный объект с настройками расположения связанного пользовательского профиля контрагента.

Исходный код схемы замещающей модели представления страницы контакта представлен ниже.

ContactPageV2

```

/* Определение схемы страницы записи и ее зависимостей. */
define("ContactPageV2", ["BaseFiltersGenerateModule", "BusinessRuleModule", "ContactPageV2Res
return {
    entitySchemaName: "Contact",
    /* Модули. */
    modules: /**SCHEMA_MODULES*/{
        /* Модуль профиля контрагента. */
        "AccountProfile1": {
            /* Конфигурация профиля. */
            "config": {
                /* Название схемы. */
                "schemaName": "UsrAccountProfileSchema",
                /* Признак, который сообщает об инициализации конфигурации схемы. */
                "isSchemaConfigInitialized": true,
                /* Признак, который сообщает, что не используется HistoryState. */
                "useHistoryState": false,
                /* Параметры профиля. */
                "parameters": {
                    /* Конфигурация модели представления. */
                    "viewModelConfig": {
                        /* Название колонки связанной сущности. */
                        masterColumnName: "Account"
                    }
                }
            }
        }
    }
}/**SCHEMA_MODULES*/,
/* Массив модификаций. */
diff: /**SCHEMA_DIFF*/[
    {
        /* Операция добавления. */
        "operation": "insert",
        /* Имя родительского элемента, в который выполняется вставка. */
        "parentName": "LeftModulesContainer",
        /* Свойство элемента родителя, с которым выполняется операция. */
        "propertyName": "items",
        /* Имя сущности. */
        "name": "AccountProfile1",
        /* Значение добавляемого элемента. */
        "values": {
            /* Тип элемента – модуль. */
            "itemType": Terrasoft.ViewItemType.MODULE
        }
    }
]

```

```

    ]/**SCHEMA_DIFF*/
  };
});

```

6. На панели инструментов дизайнера нажмите [Сохранить] ([Save]).

Результат выполнения примера

Чтобы **посмотреть результат выполнения примера**, обновите страницу раздела [Контакты] ([Contacts]).

В результате выполнения примера на страницу контакта добавлен связанный пользовательский профиль контрагента.

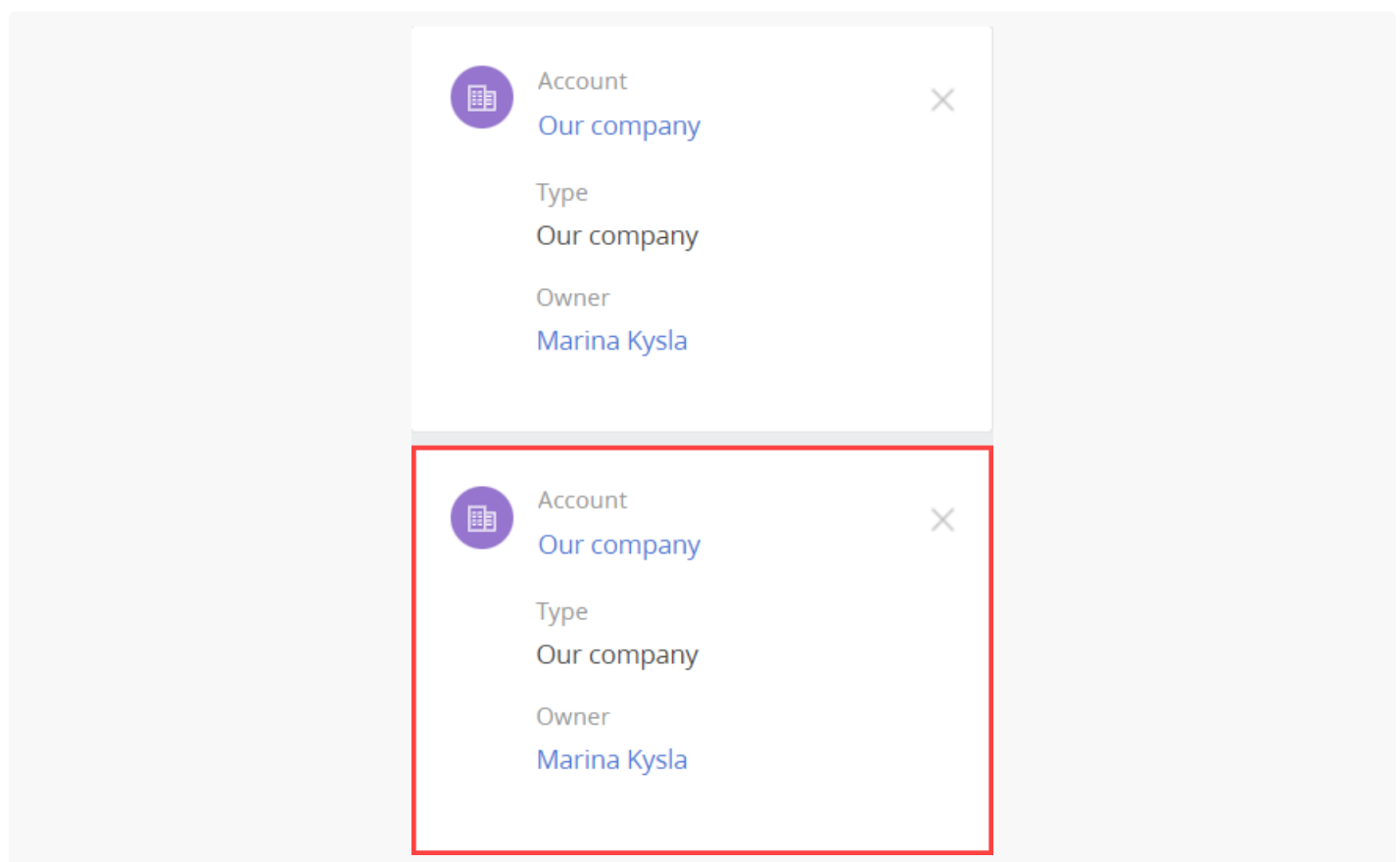


Схема BaseProfileSchema JS

 Средний

BaseProfileSchema — базовая схема для создания профиля связанной сущности. Позволяет отобразить любой набор полей по связанной сущности, а также любое количество модулей. Реализована в пакете **nui**. Схема является схемой модели представления. Описание свойств схемы содержится в статье [Клиентская схема](#). Все схемы профилей связанных сущностей должны наследовать схему **BaseProfileSchema**.

Атрибуты

MasterColumnValue GUID

Значение главной колонки.

MasterColumnInfo CUSTOM_OBJECT

Информация о главной колонке.

DataItemMarkerTpl TEXT

Шаблон маркера элемента данных.

Сообщения

Сообщения базовой схемы

Название	Режим	Направление	Описание
Entity Initialized	Широковещательное	Подписка	Событие инициализации основной сущности.
GetEntityColumn Changes	Широковещательное	Подписка	Обрабатывает изменения колонки сущности.
GetColumnsValues	Адресное	Публикация	Возвращает запрошенные значения колонки.
GetLookupQuery Filters	Адресное	Публикация	Получает справочник фильтров запроса.
GetColumnInfo	Адресное	Публикация	Возвращает информацию колонки.
UpdateCard Property	Адресное	Публикация	Изменяет значение карточки модели.
OpenCard	Адресное	Публикация	Открывает карточку.
CardModule Response	Адресное	Подписка	Ответное сообщение карточки модуля.

Режимы сообщений представлены перечислением `Terrasoft.core.enums.MessageMode`, а направления сообщений — перечислением `Terrasoft.core.enums.MessageDirectionType`. Перечисление `MessageMode` описано в [Библиотеке JS классов](#). Перечисление `MessageDirectionType` описано в [Библиотеке JS классов](#).

Методы

`getMiniPageConfig(options)`

Возвращает конфигурацию мини-карточки.

Параметры

<code>{Object} options</code>	Свойства мини-карточки.
-------------------------------	-------------------------

`getLookupConfig(config)`

Возвращает справочник конфигурации открытого справочника.

Параметры

<code>{Object} config</code>	Конфигурация открытой справочной страницы.
------------------------------	--

`getVisibleBlankSlate()`

Возвращает тег видимости контейнера `BlankSlateContainer`.

`getVisibleContent()`

Возвращает тег видимости контейнера `ProfileContentContainer`.

`getClearButtonHint()`

Возвращает всплывающую подсказку к кнопке [Очистить] ([Clear]).

`getUpdateCardPropertyConfig(response)`

Получает свойства конфигурации обновления карточки при ее сохранении.

Параметры

<code>{Object} response</code>	Ответ сервера.
--------------------------------	----------------

Схема BaseMultipleProfileSchema

 Средний

`BaseMultipleProfileSchema` — базовая схема для создания профиля связанной сущности, который содержит в себе несколько профилей и свободно между ними переключается, пользуясь логикой выбора значений из справочников. Реализована в пакете `nui`. Схема является схемой модели представления. Описание свойств схемы содержится в статье [Клиентская схема](#). Профили `BaseMultipleProfileSchema` должны наследовать базовую схему `BaseRelatedProfileSchema`, которая реализует профили,зависимые или встроенные в другие профили.

Атрибуты

`EditColumnName` `STRING`

Название колонки свойства карточки.

`MasterColumnNames` `CUSTOM_OBJECT`

Массив имен основных колонок модулей.

Сообщения

Сообщения базовой схемы

Название	Режим	Направление	Описание
<code>ProfileEntityColumnChanges</code>	Широковещательное	Публикация	Обрабатывает изменения колонки сущности профиля.
<code>GetProfileEntityColumnChanges</code>	Адресное	Подписка	Возвращает запрошенные значения колонки.
<code>ProfileOpenCard</code>	Адресное	Подписка	Отправляет запрос открытой карточки с конфигурацией.

Режимы сообщений представлены перечислением `Terrasoft.core.enums.MessageMode`, а направления сообщений — перечислением `Terrasoft.core.enums.MessageDirectionType`. Перечисление `MessageMode` описано в [Библиотеке JS классов](#). Перечисление `MessageDirectionType` описано в [Библиотеке JS классов](#).

Схема BaseRelatedProfileSchema

 Средний

`BaseRelatedProfileSchema` — базовая схема, которая реализует профили,зависимые или встроенные в другие профили. Реализована в пакете `nui`. Схема является схемой модели представления. Описание свойств схемы содержится в статье [Клиентская схема](#). Является родительской схемой для базовой

схемы `BaseMultipleProfileSchema` создания профиля связанной сущности, который содержит в себе несколько профилей и свободно между ними переключается, пользуясь логикой выбора значений из справочников.

Сообщения

Сообщения базовой схемы

Название	Режим	Направление	Описание
<code>ProfileEntityColumnChanges</code>	Широковещательное	Подписка	Обрабатывает изменения колонки сущности профиля.
<code>GetProfileEntityColumnChanges</code>	Адресное	Публикация	Отправляет запрошенные значения колонки.
<code>ProfileOpenCard</code>	Адресное	Публикация	Отправляет запрос открытой карточки с конфигурацией.

Режимы сообщений представлены перечислением `Terrasoft.core.enums.MessageMode`, а направления сообщений — перечислением `Terrasoft.core.enums.MessageDirectionType`. Перечисление `MessageMode` описано в [Библиотеке JS классов](#). Перечисление `MessageDirectionType` описано в [Библиотеке JS классов](#).