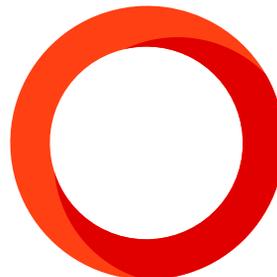
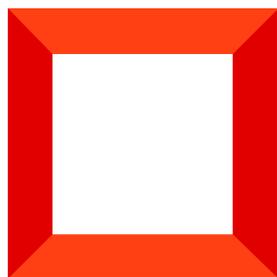
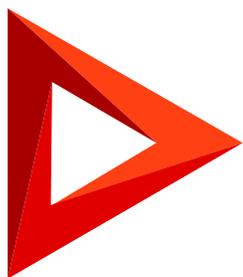


Элементы интерфейса

Коммуникационная панель

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

Коммуникационная панель	4
Структура коммуникационной панели	4
Контейнеры коммуникационной панели	4
Создать пользовательское напоминание или уведомление	5
Создать обращение по сообщению из внутренней ленты другого обращения	7
Создать схему замещающей модели представления страницы обращения	7
Результат выполнения примера	10
Скрыть область ленты в едином окне	11
Создать схему замещающей модели представления раздела	11
Результат выполнения примера	13
Отобразить часовой пояс контакта на вкладку коммуникационной панели	13
1. Создать схему замещающей модели представления страницы найденного абонента	13
2. Добавить стили отображения часового пояса	17
Результат выполнения примера	21
Создать пользовательское напоминание	21
1. Реализовать текст и окно для напоминания	22
2. Реализовать логику отправки напоминания	24
3. Реализовать отображение напоминания	28
Результат выполнения примера	32

Коммуникационная панель



Основы

Коммуникационная панель — инструмент ведения коммуникаций с клиентами и коллегами, а также получения уведомлений в Creatio. Отображается в правой части экрана.

Структура коммуникационной панели

Вкладки коммуникационной панели:

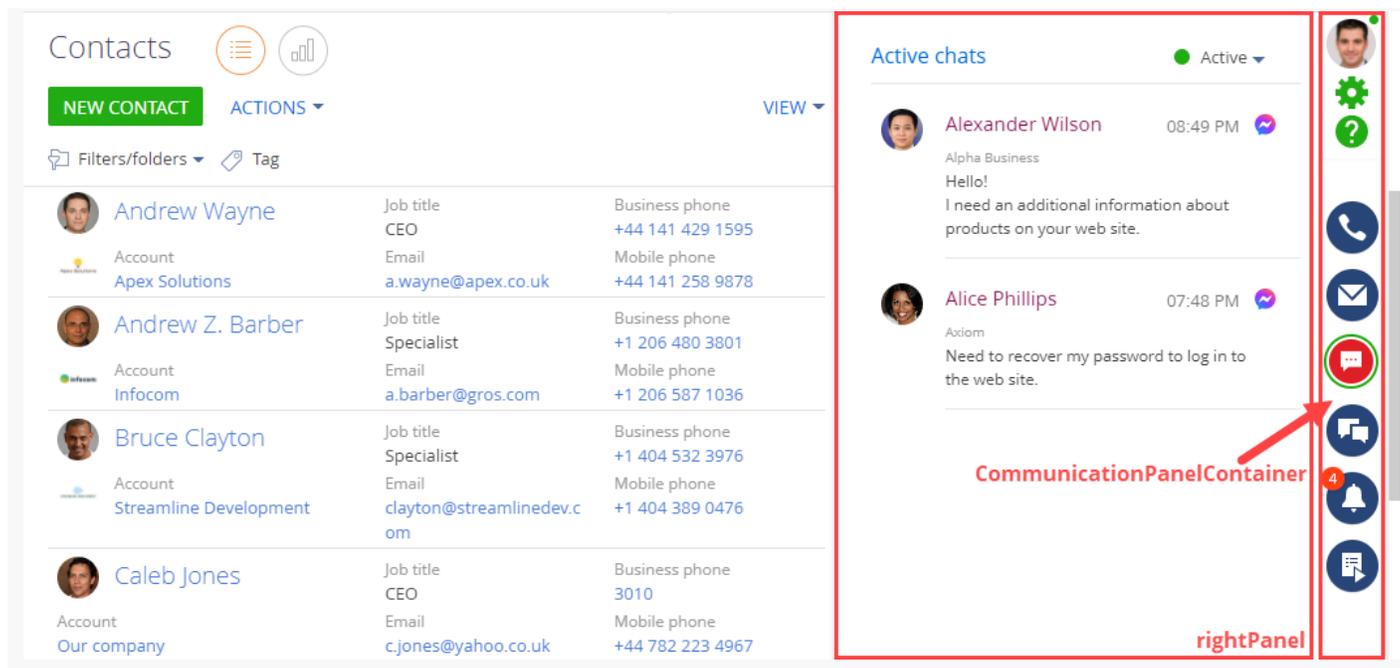
- [*Единое окно оператора*] ([*Operation single window*]) — проведение менеджером банка консультаций клиенту. Вкладка доступна в продукте Financial Services Creatio, customer journey edition. Позволяет воспользоваться различными критериями для поиска клиента банка, начать проведение консультации, отложить консультацию.
- [*Звонки*] ([*CTI panel*]) — прием входящих и выполнение исходящих звонков непосредственно в приложении. Один из инструментов телефонии в Creatio. Подробнее о телефонии читайте в блоке статей [Коннекторы к телефонии](#) и статье [Работа со звонками](#).
- [*Email*] — отправка и получение email-сообщений. Позволяет связывать email-сообщения с другими объектами приложения. Подробнее об управлении почтой читайте в статье [Работа с почтой](#).
- [*Лента*] ([*Feed*]) — отображение сообщений раздела [*Лента*] ([*Лента*]). Позволяет просматривать сообщения тех каналов, на которые вы подписаны, а также добавлять новые сообщений и комментарии. Функциональность вкладки аналогична функциональности раздела [*Лента*] ([*Feed*]).
- [*Центр уведомлений*] ([*Notification center*]) — отображает уведомления о различных событиях в приложении. Подробнее об управлении уведомлениями читайте в статье [Работа с уведомлениями](#).
- [*Задачи по бизнес-процессам*] ([*Business process tasks*]) — отображает невыполненные шаги по запущенным бизнес-процессам. Подробнее об управлении запущенными бизнес-процессами читайте в статье [Работа с уведомлениями](#).

Контейнеры коммуникационной панели

Элементы пользовательского интерфейса приложения, которые относятся к коммуникационной панели, размещены в соответствующих контейнерах. Контейнеры коммуникационной панели конфигурируются в базовой схеме `CommunicationPanel` пакета `UIv2`. Вкладка [*Звонки*] ([*CTI panel*]) конфигурируется в дочерней схеме `CtiPanel` пакета `CTIBase`.

На заметку. В приложении используются мета-имена html-контейнеров. На основании мета-имен приложение формирует фактические идентификаторы соответствующих html-элементов коммуникационной панели.

Основные **контейнеры коммуникационной панели** представлены на рисунке ниже.



- Контейнер кнопок коммуникационной панели (`CommunicationPanelContainer`) — содержит кнопки, которые позволяют открыть вкладки коммуникационной панели.
- Контейнер вкладок коммуникационной панели (`rightPanel`) — содержит контент текущей вкладки коммуникационной панели.

Создать пользовательское напоминание или уведомление

Варианты отображения уведомлений, которые позволяет настроить Creatio:

- Мгновенно.
- В указанное время.

Чтобы **создать пользовательское напоминание или уведомление**:

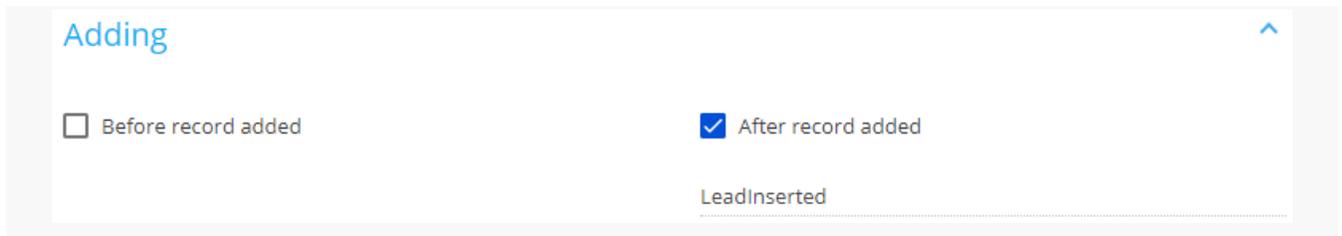
1. Реализуйте **текст и окно для напоминания**.

Создайте схему типа [*Исходный код*] ([*Source code*]). Для этого воспользуйтесь инструкцией, которая приведена в статье [Исходный код \(C#\)](#).

- Создайте класс для формирования текста и окна напоминания.
- В классе реализуйте интерфейс `IRemindingTextFormer` СХЕМЫ `IRemindingTextFormer` пакета `Base`.

2. Реализуйте **логику отправки напоминания**.

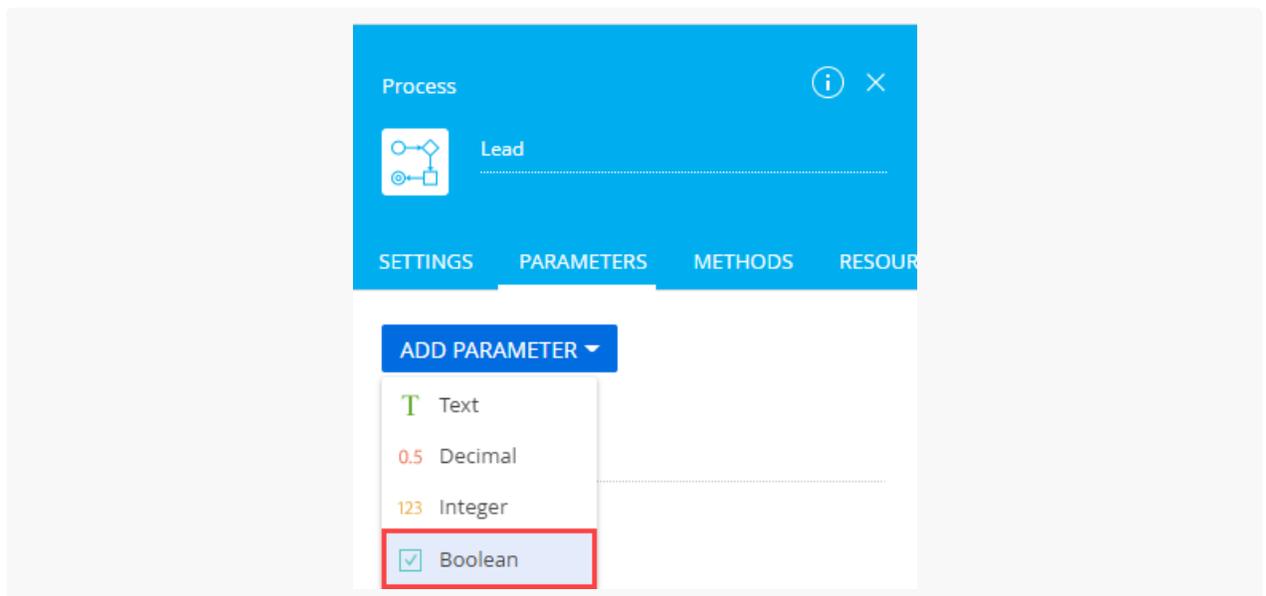
- Создайте замещающий объект, по которому планируется отправлять напоминание. Для этого воспользуйтесь инструкцией, которая приведена в статье [Объект](#).
- Создайте **событие объекта**. Для этого перейдите в узел [*События*] ([*Events*]) и в блоке [*Добавление*] ([*Adding*]) установите признак [*После добавления записи*] ([*After record added*]).



с. В схеме объекта настройте **параметр процесса, который генерирует напоминание**.

а. Добавьте **параметр**.

а. На панели свойств перейдите на вкладку [*Параметры*] ([*Parameters*]) и нажмите [*Добавить параметр*] → [*Логическое*] ([*Add parameter*] → [*Boolean*]).



б. В дизайнера объекта заполните **свойства параметра**.

б. Переопределите **методы**.

- Переопределите **метод, который вызывается после сохранения объекта**. Для этого на панели свойств перейдите на вкладку [*Методы*] ([*Methods*]) и добавьте исходный код переопределенного метода.

Способы отображения напоминания:

- Вкладка напоминаний центра уведомлений. В этом случае константе `remindingConfig.NotificationTypeId` присвойте значение `RemindingConsts.NotificationTypeRemindingId`.
- Вкладка служебных сообщений центра уведомлений. В этом случае константе `remindingConfig.NotificationTypeId` присвойте значение `RemindingConsts.NotificationTypeNotificationId`.
- Переопределите **метод, который вызывается перед сохранением объекта**. Для этого на панели свойств перейдите на вкладку [*Методы*] ([*Methods*]) и добавьте исходный код переопределенного метода.

3. Реализуйте **отображение напоминания**.

Создайте замещающую модель представления. Для этого воспользуйтесь инструкцией, которая приведена в статье [Клиентский модуль](#). В качестве родительского объекта выберите

`ReminderNotificationsSchema`.

Создать обращение по сообщению из внутренней ленты другого обращения

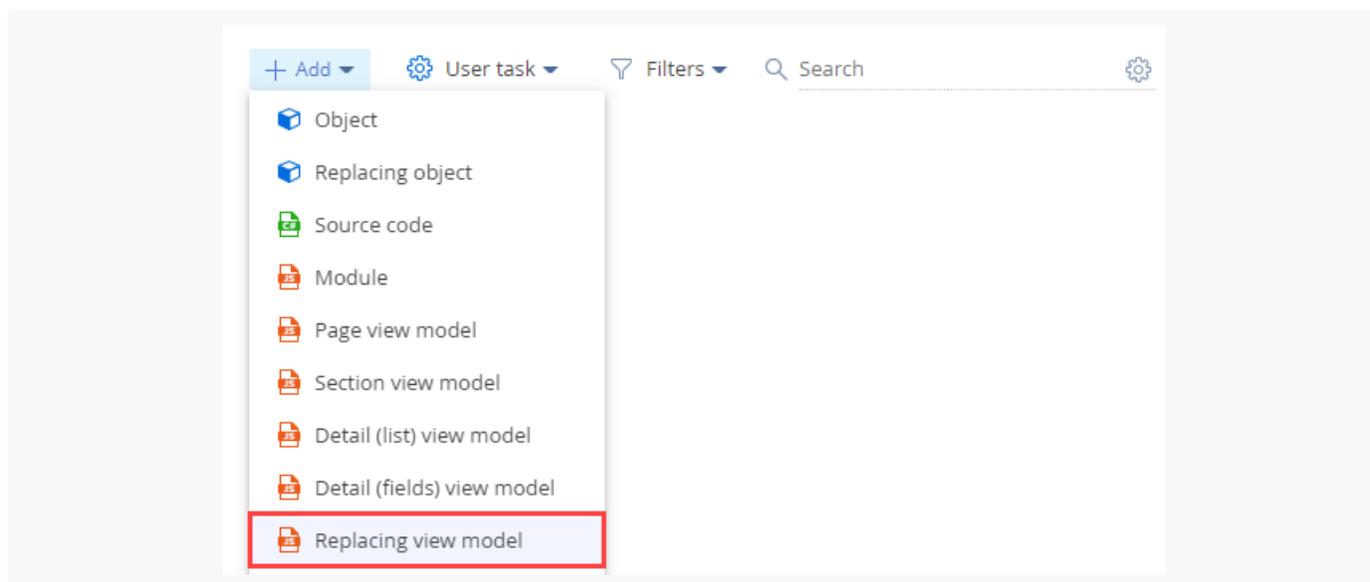


Пример реализован для продуктов линейки Service Creatio.

Пример. На вкладку [*Обработка*] ([*Processing*]) страницы обращения добавить всплывающую кнопку. Кнопка отображается при выделении текста в сообщениях с почты и портала самообслуживания, отправленных с внутренней ленты обращения. При нажатии на кнопку создается новое обращение. Поля [*Тема*] ([*Subject*]) и [*Описание*] ([*Description*]) заполняются автоматически. Значения полей — выделенный текст.

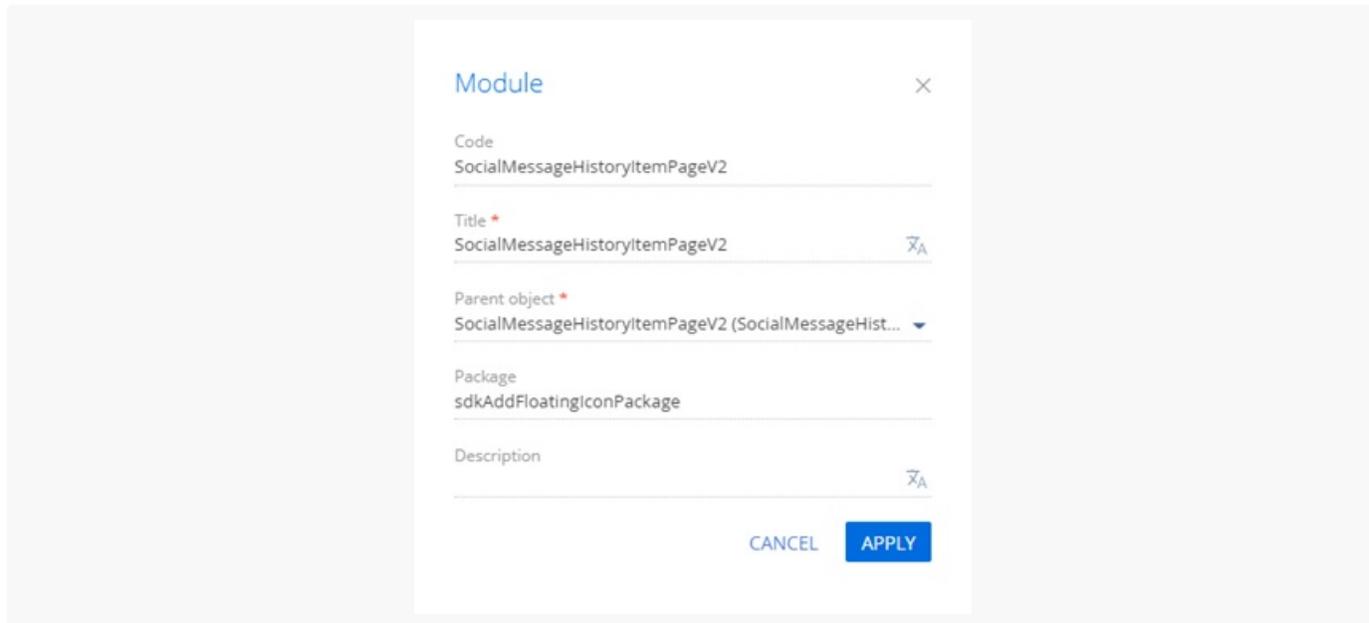
Создать схему замещающей модели представления страницы обращения

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*]) — "SocialMessageHistoryItemPageV2".
- [*Заголовок*] ([*Title*]) — "SocialMessageHistoryItemPageV2".
- [*Родительский объект*] ([*Parent object*]) — выберите "SocialMessageHistoryItemPageV2".



4. Реализуйте **логику работы всплывающей кнопки**.

- В свойстве `methods` реализуйте **методы**:
 - `onSelectedTextChanged()` — передает значение выделенного текста в атрибут `HighlightedHistoryMessage`. Срабатывает при выделении текста.
 - `onSelectedTextButtonClicked()` — создает обращение, тему которого получает из атрибута `HighlightedHistoryMessage`. Логика создания обращения определена в родительской схеме `BaseMessageHistory`. Срабатывает при нажатии на всплывающую кнопку.
 - `getMessageFromHistory()` — переопределенный метод родительской схемы, который получает тему выделенного сообщения.
- В массив модификаций `diff` добавьте конфигурационный объект с настройками элемента `SelectionHandlerMultiLineLabel` пакета `Message`, который реализует логику создания нового обращения на основе выделенного текста.

Исходный код схемы замещающей модели представления страницы обращения представлен ниже.

SocialMessageHistoryItemPageV2

```
define("SocialMessageHistoryItemPageV2", ["SocialMessageConstants", "css!SocialMessageHistory
  return {
    /* Название схемы объекта страницы записи. */
    entitySchemaName: "BaseMessageHistory",
```

```

/* Детали модели представления страницы записи. */
details: /**SCHEMA_DETAILS*/{/**SCHEMA_DETAILS*/,
/* Методы модели представления страницы записи. */
methods: {
    /* Переопределение базового метода. Получает тему для выделенного сообщения. */
    getMessageFromHistory: function() {
        var message = this.get("HighlightedHistoryMessage");
        if (this.isHistoryMessageEmpty(message)) {
            message = this.get("[Activity:Id:RecordId].Body");
        }
        return message;
    },
    /* Обработчик события выделения текста. */
    onSelectedTextChanged: function(text) {
        this.set("HighlightedHistoryMessage", text);
    },
    /* Обработчик нажатия всплывающей кнопки. */
    onSelectedTextButtonClick: function() {
        /* Подготовка данных по обращению из истории. */
        this.prepareCaseDataFromHistory();
    }
},
/* Отображение кнопки на странице записи. */
diff: /**SCHEMA_DIFF*/[
    /* Метаданные для добавления на страницу всплывающей кнопки. */
    {
        /* Выполняется операция изменения существующего элемента. */
        "operation": "merge",
        /* Мета-имя изменяемого компонента. */
        "name": "MessageText",
        /* Свойства, передаваемые в конструктор элемента. */
        "values": {
            /* Свойства генератора представления. */
            "generator": function() {
                return {
                    /* Значение HTML-тега id. */
                    "id": "MessageText",
                    /* Значение маркера. */
                    "markerValue": "MessageText",
                    /* Название класса компонента. */
                    "className": "Terrasoft.SelectionHandlerMultilineLabel",
                    /* Настройка CSS-стилей. */
                    "classes": {
                        "multilineLabelClass": ["messageText"]
                    },
                    /* Заголовок. */
                    "caption": {"bindTo": "Message"},
                    "showLinks": true,
                    /* Привязка события изменения выделенного текста к методу-обработ

```

```

        "selectedTextChanged": {"bindTo": "onSelectedTextChanged"},
        /* Привязка события нажатия всплывающей кнопки выделенного текста
        "selectedTextHandlerButtonClick": {"bindTo": "onSelectedTextButto
        /* Признак видимости всплывающей кнопки. */
        "showFloatButton": true
    };
};
}
}
}
]/**SCHEMA_DIFF*/
};
});

```

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

Результат выполнения примера

Чтобы **посмотреть результат выполнения примера**:

1. Обновите страницу раздела [Обращения] ([Cases]).
2. Откройте страницу обращения.

В результате выполнения примера на вкладку [Обработка] ([Processing]) страницы обращения добавлена всплывающая кнопка. Кнопка отображается при выделении текста в сообщениях с почты и портала самообслуживания, отправленных с внутренней ленты обращения. При нажатии на кнопку создается новое обращение. Поля [Тема] ([Subject]) и [Описание] ([Description]) заполняются автоматически. Значения полей — выделенный текст.

Number	Subject	Category	Assignee	Status	Resolution time
SR0000048	Consultation on functionality	Service request	Marina Kysla	Waiting for response	11/25/2021 10:00 PM
SR0000047	Unable to create an account on the site	Service request	William Walker	Resolved	11/27/2021 12:05 PM
SR0000046	What's your e-mail address?	Service request	Jason Robinson	New	11/25/2021 3:00 PM
SR0000044	Recovering password to login to the system	Service request	Marina Kysla	In progress	11/25/2021 1:14 AM
SR0000041	Missing data on hard drive	Service request	Marina Kysla	In progress	11/23/2021 1:00 AM
SR0000038	Color laserjet pro CP1025nw - printing issues	Service request	Marina Kysla	In progress	11/24/2021 12:21 PM
SR0000037	Consultation on settings	Service request	Marina Kysla	Waiting for response	11/23/2021 10:00 PM
SR0000036	An unknown error when the computer starts	Incident	Megan Lewis	In progress	11/26/2021 3:00 AM
SR0000030	The customer cannot create an account on the site.	Service request	Marina Kysla	In progress	11/27/2021 5:00 AM

Скрыть область ленты в едином окне

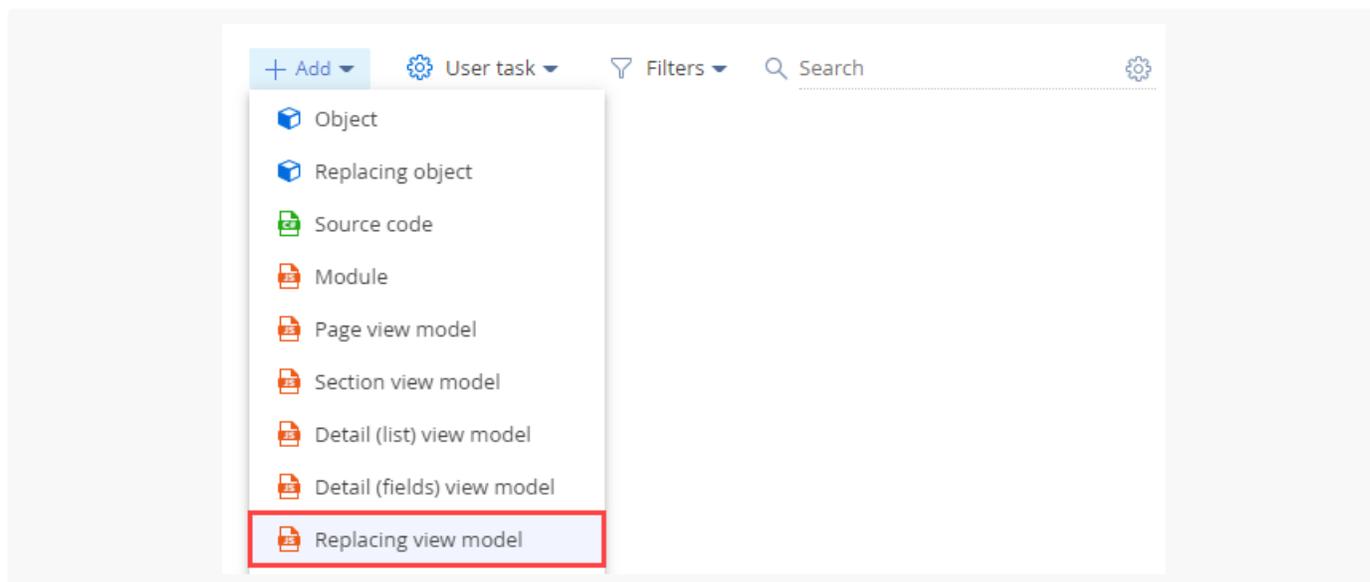
 Средний

Пример реализован для рабочего места [Контакт центр] ([Contact center]).

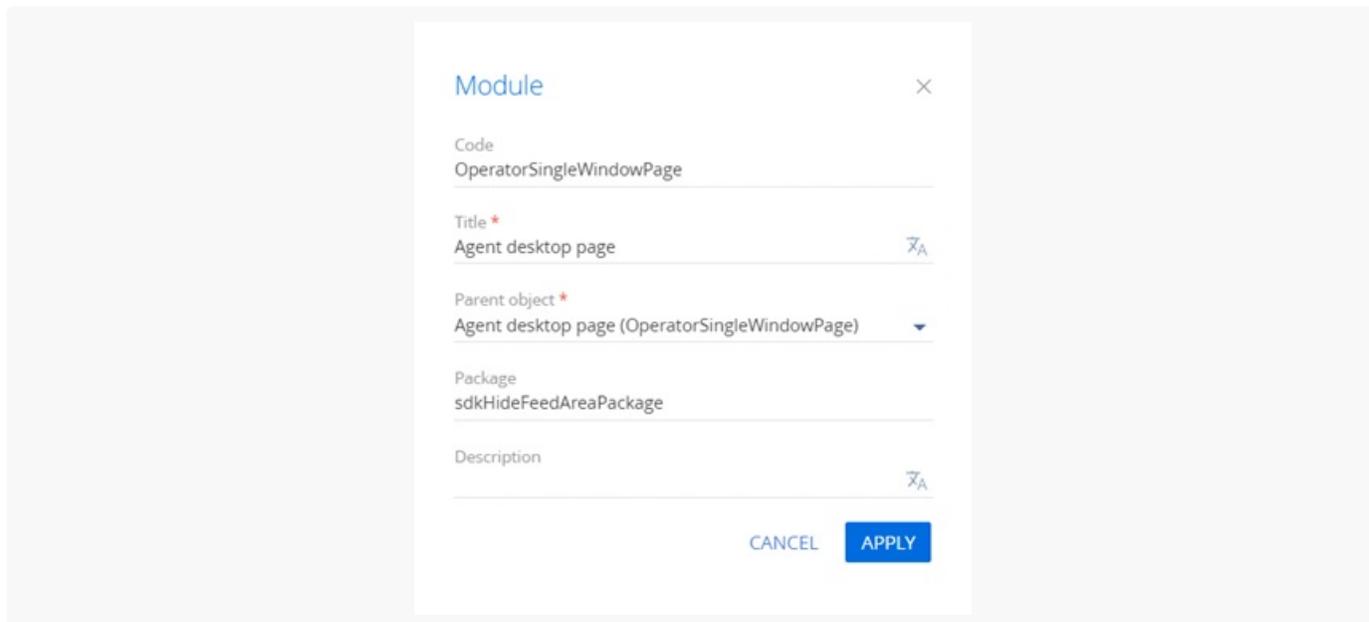
Пример. В разделе [Единое окно] ([Agent desktop]) скрыть область ленты.

Создать схему замещающей модели представления раздела

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]) — "OperatorSingleWindowPage".
 - [Заголовок] ([Title]) — "Страница единого окна оператора" ("Agent desktop page").
 - [Родительский объект] ([Parent object]) — выберите "OperatorSingleWindowPage".



4. Реализуйте **логику скрытия области ленты**.

- В свойстве `methods` реализуйте метод `loadContent()` — переопределенный базовый метод, который исключает из перечня загружаемых модулей модуль ленты `ESNFeedModule`.
- В массив модификаций `diff` добавьте конфигурационный объект, который удаляет элемент со страницы.

Исходный код схемы замещающей модели представления раздела представлен ниже.

OperatorSingleWindowPage

```
define("OperatorSingleWindowPage", [], function() {
    return {
        /* Методы модели представления раздела. */
        methods: {
            /* Замещает базовый метод для исключения из состава загружаемых модулей модуля ле
            loadContent: function() {
                /* Поскольку контейнер centerContainer удален, то модуль ESNFeedModule загруж
                this.loadModule("ESNFeedModule", "centerContainer"); */
                /* Загрузка модулей. */
                this.loadModule("SectionDashboardsModule", "rightContainer");
                this.loadModule("OperatorQueuesModule", "leftContainer");
            }
        },
        /* Отображение контейнера в разделе. */
        diff: /**SCHEMA_DIFF*/[
            /* Метаданные для удаления контейнера из раздела. */
            {
                /* Выполняется операция удаления существующего элемента. */
                "operation": "remove",
                /* Мета-имя удаляемого компонента. */
            }
        ]
    }
});
```

```

        "name": "centerContainer"
      }
    ]/**SCHEMA_DIFF*/
  };
});

```

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

Результат выполнения примера

Чтобы **посмотреть результат выполнения примера**, обновите страницу раздела [Единое окно] ([Agent desktop]).

В результате выполнения примера в разделе [Единое окно] ([Agent desktop]) скрыта область ленты.

The screenshot shows the 'Agent desktop' interface. On the left, there is a table with the following data:

Number	Registration date	Resolution time
SR00000046	11/22/2021 2:35 PM	11/25/2021 3:00 PM
SR00000044	11/21/2021 8:14 PM	11/25/2021 1:14 AM
SR00000041	11/21/2021 11:15 AM	11/23/2021 1:00 AM
SR00000038	11/20/2021 12:21 PM	11/24/2021 12:21 PM
SR00000030	11/17/2021 7:30 AM	11/27/2021 5:00 AM
SR00000029	11/16/2021 5:30 PM	11/27/2021 2:00 AM

On the right side, there are two charts:

- My active cases:** A line chart showing the number of cases in different statuses. The y-axis is labeled 'Number' and ranges from 0 to 6. The x-axis is labeled 'Status' and has three categories: 'In progress', 'Resolved', and 'Waiting for response'. The data points are approximately: In progress (5), Resolved (2), and Waiting for response (2).
- Processed:** A gauge chart showing the number of processed cases. The gauge has a scale from 0 to 10, with a current value of 0. The gauge is divided into segments of different colors: red (0-5), yellow (5-8), and green (8-10).

Отобразить часовой пояс контакта на вкладку коммуникационной панели

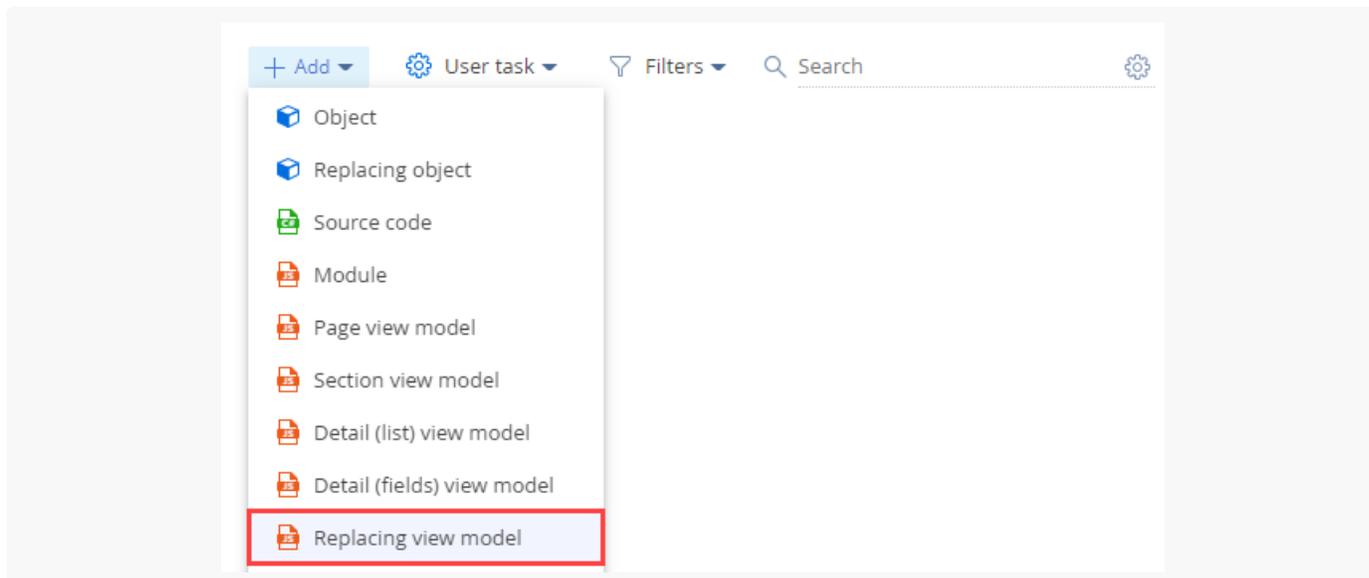
 Сложный

Пример. При поиске контакта на вкладке [Звонки] ([CTI panel]) коммуникационной панели отображать его часовой пояс. Использовать текущее время контакта.

1. Создать схему замещающей модели представления

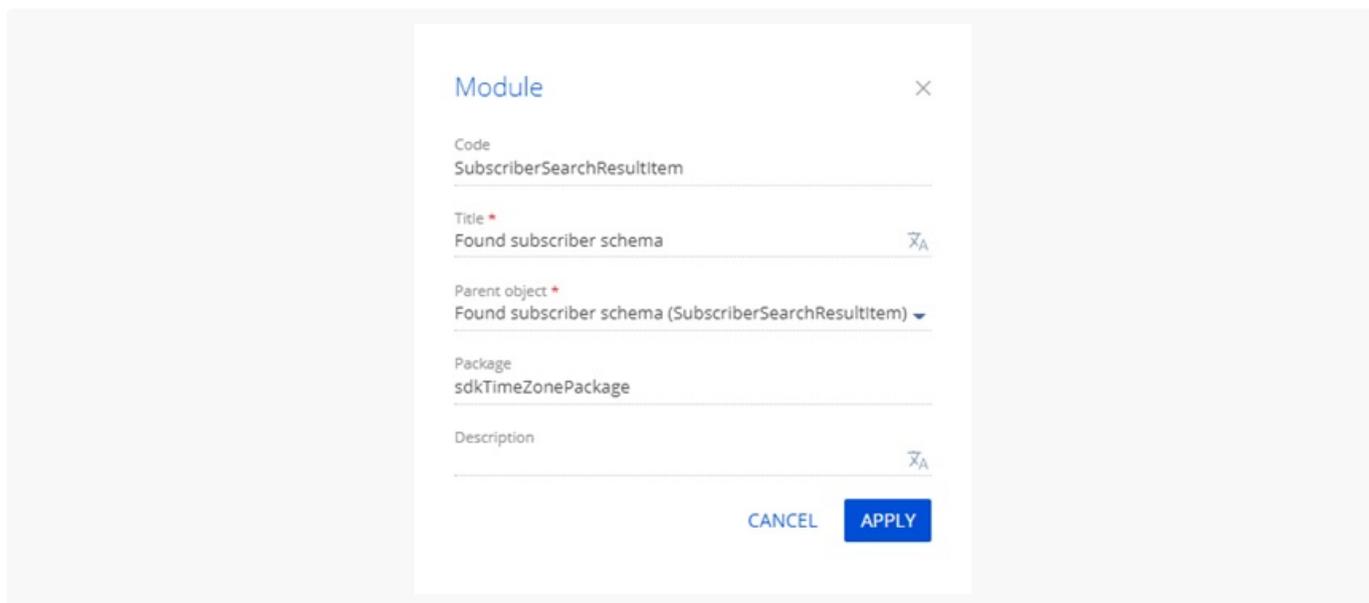
страницы найденного абонента

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]) — "SubscriberSearchResultItem".
- [Заголовок] ([Title]) — "Схема найденного абонента" ("Found subscriber schema").
- [Родительский объект] ([Parent object]) — выберите "SubscriberSearchResultItem".



4. В объявлении класса модели представления в качестве зависимостей добавьте модули `TimezoneGenerator` и `TimezoneMixin`. Модуль `TimezoneGenerator` формирует элемент отображения

часового пояса контакта. Модуль `TimezoneMixin` выполняет поиск часового пояса контакта.

5. Реализуйте логику отображения часового пояса.

- В свойстве `attributes` добавьте атрибут `IsShowTimeZone`, который отвечает за состояние отображения элемента часового пояса.
- В свойстве `mixins` добавьте миксин `TimezoneMixin`. Для запуска поиска часового пояса контакта в метод `init()` миксина `TimezoneMixin` передайте уникальный идентификатор контакта. **Атрибуты**, которые будут установлены в результате выполнения:
 - `TimezoneCaption` — название часового пояса контакта и текущее время.
 - `TimezoneCity` — название города, который находится в часовом поясе.
- В свойстве `methods` реализуйте **методы**:
 - `constructor()` — конструктор класса.
 - `isContactType()` — возвращает признак, который указывает, что абонент является контактом.
- В массив модификаций `diff` добавьте конфигурационный объект с настройками отображения часового пояса контакта.
 - Свойство `index` — настройка позиционирования элемента.

Элементы контейнера `SubscriberSearchResultItemContainer`:

- Индекс 0 — фотография абонента.
- Индекс 1 — информация об абоненте.
- Индекс 2 — телефоны абонента.

Свойству `index` массива модификаций присвойте значение 2, чтобы отобразить часовой пояс контакта между данными абонента и списком телефонных номеров.

- Свойство `wrapClass` — управление стилями. Свойство предоставляет генератор элемента. Стили текстовых элементов в схеме определяются CSS-классом `subscriber-data`.

Исходный код схемы замещающей модели представления страницы найденного абонента представлен ниже.

SubscriberSearchResultItem

```
define("SubscriberSearchResultItem", ["TimezoneGenerator", "TimezoneMixin"], function() {
    return {
        /* Атрибуты модели представления страницы. */
        attributes: {
            /* Название атрибута, который отвечает за состояние отображения элемента часового
            "IsShowTimeZone": {
                /* Тип данных колонки модели представления. */
                "dataValueType": Terrasoft.DataValueType.BOOLEAN,
                /* Тип атрибута. */
                "type": Terrasoft.ViewModelColumnType.VIRTUAL_COLUMN,
```

```

        /* Значение по умолчанию. */
        "value": true
    }
},
/* Миксины модели представления страницы. */
mixins: {
    /* Подключение миксина. */
    TimezoneMixin: "Terrasoft.TimezoneMixin"
},
/* Методы модели представления страницы. */
methods: {
    /* Конструктор класса. */
    constructor: function() {
        /* Вызов базового конструктора. */
        this.callParent(arguments);
        /* Признак того, что абонент является контактом. */
        var isContact = this.isContactType();
        /* Если абонент – контакт, то элемент отображается. */
        this.set("IsShowTimeZone", isContact);
        /* Если абонент – контакт. */
        if (isContact) {
            /* Идентификатор контакта. */
            var contactId = this.get("Id");
            /* Поиск часового пояса контакта. */
            this.mixins.TimezoneMixin.init.call(this, contactId);
        }
    },
    /* Возвращает признак того, что абонент – контакт. */
    isContactType: function() {
        /* Тип абонента. */
        var type = this.get("Type");
        /* Возвращает результат сравнения. */
        return type === "Contact";
    }
},
/* Массив модификаций модели представления страницы. */
diff: [
    {
        /* Выполняется операция добавления элемента на страницу. */
        "operation": "insert",
        /* Мета-имя родительского контейнера, в который добавляется элемент. */
        "parentName": "SubscriberSearchResultItemContainer",
        /* Элемент добавляется в коллекцию элементов родительского элемента. */
        "propertyName": "items",
        /* Мета-имя добавляемого элемента. */
        "name": "TimezoneContact",
        /* Свойства, передаваемые в конструктор элемента. */
        "values": {
            /* Тип элемента. */

```

```

        "itemType": Terrasoft.ViewItemType.CONTAINER,
        /* Для формирования конфигурации представления вызывается метод генератор
        "generator": "TimezoneGenerator.generateTimeZone",
        /* Видимость контейнера привязывается к атрибуту. */
        "visible": {"bindTo": "IsShowTimeZone"},
        /* Имя CSS класса. */
        "wrapClass": ["subscriber-data", "timezone"],
        /* Привязка заголовка к атрибуту. */
        "timeZoneCaption": {"bindTo": "TimeZoneCaption"},
        /* Привязка города к атрибуту. */
        "timeZoneCity": {"bindTo": "TimeZoneCity"}
    },
    /* Позиция элемента в родительском контейнере. */
    "index": 2
}
]
};
});

```

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

В результате в приложении отображается текущее время контакта и его город.

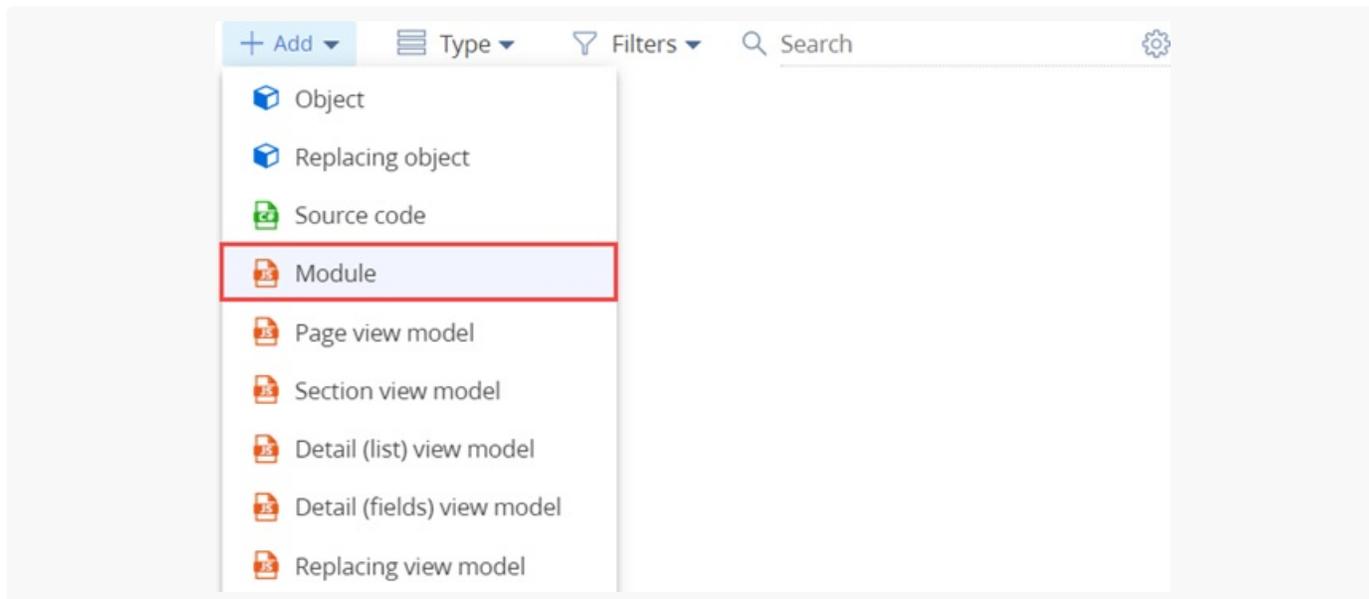
2. Добавить стили отображения часового пояса

Поскольку в схеме модели представления страницы найденного абонента невозможно задать стили для отображения, необходимо:

1. Создать схему модуля, в которой определить стили.
2. Добавить модуль со стилями в зависимости страницы найденного абонента.

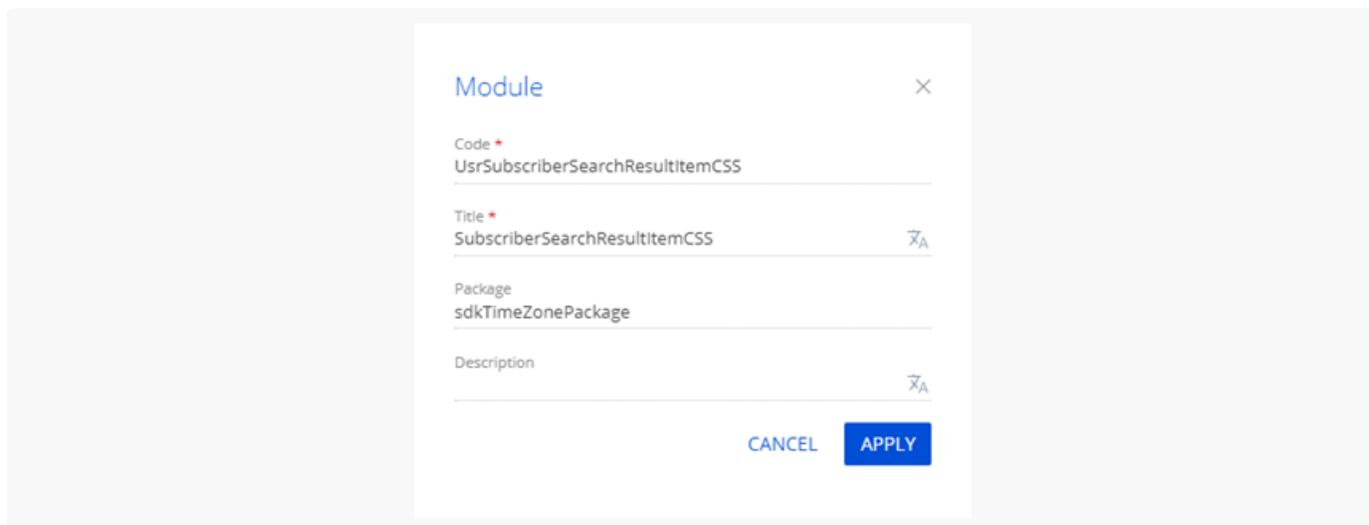
1. Создать схему модуля

1. [Перейдите в раздел \[Конфигурация \]](#) ([*Configuration*]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [*Добавить*] → [*Модуль*] ([*Add*] → [*Module*]).



3. Заполните **свойства схемы**:

- [Код] ([Code]) — "UsrSubscriberSearchResultItemCSS".
- [Заголовок] ([Title]) — "SubscriberSearchResultItemCSS".



Для применения заданных свойств нажмите [Применить] ([Apply]).

4. Перейдите в узел [LESS] структуры объекта и задайте необходимые стили отображения часового пояса.

Настройка стилей отображения часового пояса

```
/* Настройка стилей для отображения добавляемого элемента. */
.ctiPanelMain .search-result-items-list-container .timezone {
  /* Отступ сверху. */
  padding-top: 13px;
  /* Смещение снизу. */
}
```

```

margin-bottom: -10px;
}
/* Настройка стилей для отображения времени контакта. */
.ctiPanelMain .search-result-items-list-container .timezone-caption {
  /* Отступ слева. */
  padding-left: 10px;
  /* Цвет текста. */
  color: rgb(255, 174, 0);
  /* Шрифт текста – жирный. */
  font-weight: bold;
}
/* Настройка стилей для отображения города контакта. */
.ctiPanelMain .search-result-items-list-container .timezone-city {
  /* Отступ слева. */
  padding-left: 10px;
}

```

5. Перейдите в узел [JS] структуры объекта и добавьте код модуля.

UsrSubscriberSearchResultItemCSS

```

define("UsrSubscriberSearchResultItemCSS", [], function() {
  return {};
});

```

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

2. Модифицировать схему модели представления страницы найденного абонента

Чтобы **использовать созданный модуль и его стили** в схеме страницы найденного абонента:

1. Откройте схему `SubscriberSearchResultItem` модели представления страницы найденного абонента.
2. В зависимости схемы `SubscriberSearchResultItem` добавьте модуль `UsrSubscriberSearchResultItemCSS`.

Исходный код модифицированной схемы страницы найденного абонента представлен ниже.

SubscriberSearchResultItem

```

define("SubscriberSearchResultItem", ["TimezoneGenerator", "TimezoneMixin", "css!UsrSubscribe
  return {
    /* Атрибуты модели представления страницы. */
    attributes: {
      /* Название атрибута, который отвечает за состояние отображения элемента часового
      "IsShowTimeZone": {
        /* Тип данных колонки модели представления. */

```

```

        "dataValueType": Terrasoft.DataValueType.BOOLEAN,
        /* Тип атрибута. */
        "type": Terrasoft.ViewModelColumnType.VIRTUAL_COLUMN,
        /* Значение по умолчанию. */
        "value": true
    }
},
/* Миксины модели представления страницы. */
mixins: {
    /* Подключение миксина. */
    TimezoneMixin: "Terrasoft.TimezoneMixin"
},
/* Методы модели представления страницы. */
methods: {
    /* Конструктор класса. */
    constructor: function() {
        /* Вызов базового конструктора. */
        this.callParent(arguments);
        /* Признак того, что абонент является контактом. */
        var isContact = this.isContactType();
        /* Если абонент – контакт, то элемент отображается. */
        this.set("IsShowTimeZone", isContact);
        /* Если абонент – контакт. */
        if (isContact) {
            /* Идентификатор контакта. */
            var contactId = this.get("Id");
            /* Поиск часового пояса контакта. */
            this.mixins.TimezoneMixin.init.call(this, contactId);
        }
    },
    /* Возвращает признак того, что абонент – контакт. */
    isContactType: function() {
        /* Тип абонента. */
        var type = this.get("Type");
        /* Возвращает результат сравнения. */
        return type === "Contact";
    }
},
/* Массив модификаций модели представления страницы. */
diff: [
    {
        /* Выполняется операция добавления элемента на страницу. */
        "operation": "insert",
        /* Мета-имя родительского контейнера, в который добавляется элемент. */
        "parentName": "SubscriberSearchResultItemContainer",
        /* Элемент добавляется в коллекцию элементов родительского элемента. */
        "propertyName": "items",
        /* Мета-имя добавляемого элемента. */
        "name": "TimezoneContact",
    }
]

```

```

/* Свойства, передаваемые в конструктор элемента. */
"values": {
  /* Тип элемента. */
  "itemType": Terrasoft.ViewItemType.CONTAINER,
  /* Для формирования конфигурации представления вызывается метод генератор
  "generator": "TimeZoneGenerator.generateTimeZone",
  /* Видимость контейнера привязывается к атрибуту. */
  "visible": {"bindTo": "IsShowTimeZone"},
  /* Имя CSS класса. */
  "wrapClass": ["subscriber-data", "timezone"],
  /* Привязка заголовка к атрибуту. */
  "timeZoneCaption": {"bindTo": "TimeZoneCaption"},
  /* Привязка города к атрибуту. */
  "timeZoneCity": {"bindTo": "TimeZoneCity"}
},
/* Позиция элемента в родительском контейнере. */
"index": 2
}
]
};
});

```

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

Результат выполнения примера

Чтобы **посмотреть результат выполнения примера**:

1. Откройте вкладку [*Звонки*] ([*CTI panel*]) коммуникационной панели.
2. Выполните поиск абонента.

В результате выполнения примера при поиске контакта на вкладке [*Звонки*] ([*CTI panel*]) коммуникационной панели отображается его часовой пояс. Используется текущее время контакта.

The screenshot displays a CRM interface with two main sections. On the left, the 'Accounts' section shows details for 'Vertigo Systems', including the website 'www.vertigosys.com', address '83 Ashton Street, London, United Kingdom', and primary contact 'Peter Moore'. On the right, the contact details for 'Bruce Clayton' are shown, including his title 'Streamline Develop... Specialist' and phone numbers. A red box highlights the time zone information '3:30 AM, Atlanta' next to a clock icon.

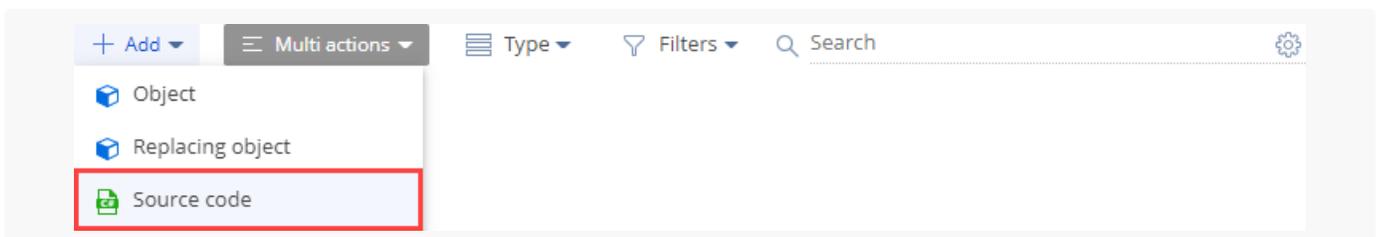
Создать пользовательское напоминание

 Средний

Пример. Создать пользовательское напоминание о дате актуализации продажи в лиде. Дата указана на вкладке [*Информация о сделке*] ([*Opportunity info*]) в поле [*Дата следующей актуализации*] ([*Next actualization date*]). Напоминание отобразить на вкладке напоминаний центра уведомлений.

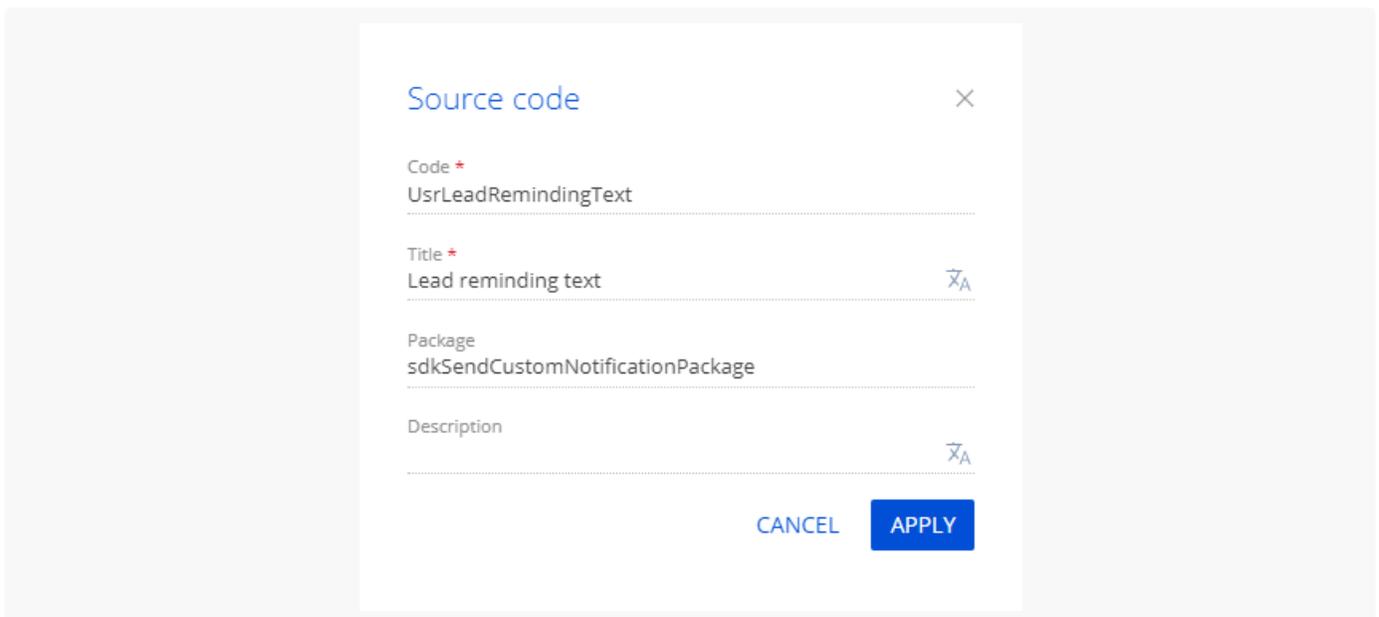
1. Реализовать текст и окно для напоминания

1. [Перейдите в раздел \[*Конфигурация* \]](#) ([*Configuration*]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [*Добавить*] → [*Исходный код*] ([*Add*] → [*Source code*]).



3. В дизайнера исходного кода заполните **свойства схемы**:

- [*Код*] ([*Code*]) — "UsrLeadRemindingText".
- [*Заголовок*] ([*Title*]) — "Текст напоминания для лида" ("Lead reminding text").



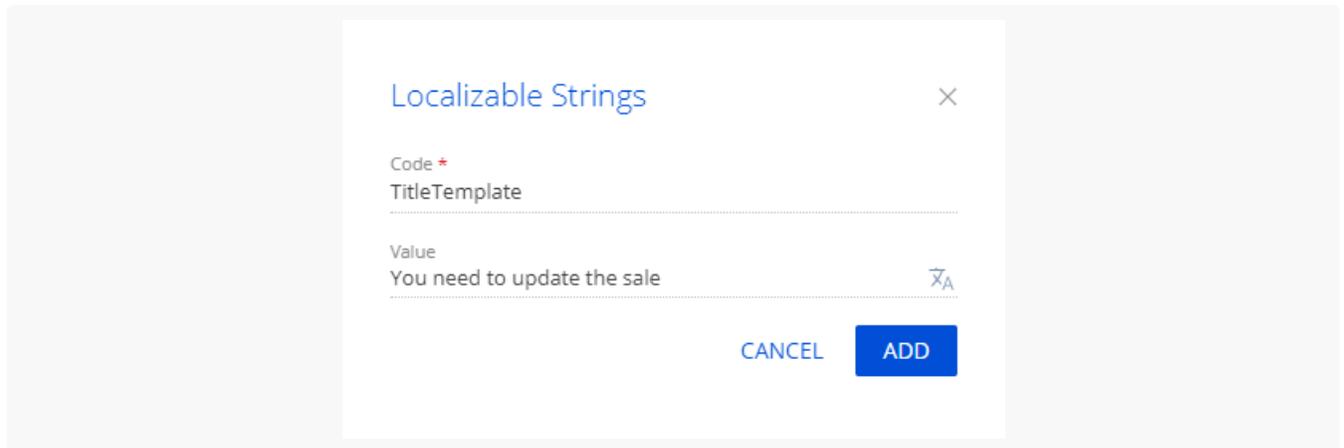
Для применения изменений свойств нажмите [*Применить*] ([*Apply*]).

4. Добавьте **локализуемую строку** с заголовком окна напоминания.

- a. В контекстном меню узла [*Локализуемые строки*] ([*Localizable strings*]) нажмите кнопку **+**.

b. Заполните **свойства локализуемой строки**:

- [Код] ([Code]) — "TitleTemplate".
- [Значение] ([Value]) — "Необходимо актуализировать продажу" ("You need to update the sale").



е. Для добавления локализуемой строки нажмите [*Добавить*] ([Add]).

5. Аналогично добавьте **локализуемую строку** с текстом напоминания.

Свойства локализуемой строки, которую необходимо добавить, приведены в таблице ниже.

Значения свойств локализуемой строки

[Код] ([Code])	[Значение] ([Value])
"BodyTemplate"	"Лид {0} требует актуализации информации по продаже" ("Lead {0} requires update of sales information")

6. Реализуйте **текст и окно для напоминания**.

Исходный код схемы исходного кода представлен ниже.

UsrLeadRemindingText

```
namespace Terrasoft.Configuration
{
    using System.Collections.Generic;
    using Terrasoft.Common;
    using Terrasoft.Core;

    public class UsrLeadRemindingText : IRemindingTextFormer
    {
        private const string ClassName = nameof(UsrLeadRemindingText);
        protected readonly UserConnection UserConnection;

        public UsrLeadRemindingText(UserConnection userConnection) {
```

```

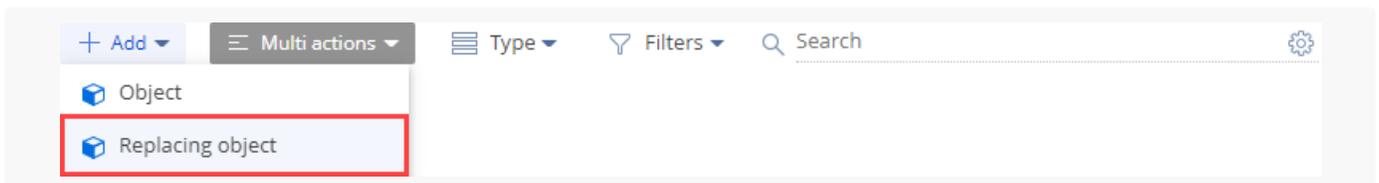
        UserConnection = userConnection;
    }
    /* Формирует текст напоминания из коллекции входящих параметров и локализуемой строки
    public string GetBody(IDictionary<string, object> formParameters) {
        formParameters.CheckArgumentNull("formParameters");
        var bodyTemplate = UserConnection.GetLocalizableString(ClassName, "BodyTemplate");
        var leadName = (string)formParameters["LeadName"];
        var body = string.Format(bodyTemplate, leadName);
        return body;
    }
    /* Формирует заголовок напоминания из имени класса и локализуемой строки TitleTemplat
    public string GetTitle(IDictionary<string, object> formParameters) {
        return UserConnection.GetLocalizableString(ClassName, "TitleTemplate");
    }
    }
}

```

7. На панели инструментов дизайнера исходного кода нажмите [*Сохранить*] ([*Save*]) для временного сохранения изменений в метаданных схемы.
8. На панели инструментов дизайнера исходного кода нажмите [*Опубликовать*] ([*Publish*]) для выполнения изменений на уровне базы данных.

2. Реализовать логику отправки напоминания

1. [Перейдите в раздел \[*Конфигурация* \]](#) ([*Configuration*]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [*Добавить*] → [*Замещающий объект*] ([*Add*] → [*Replacing object*]).



3. В дизайнере объекта заполните свойство схемы [*Родительский объект*] ([*Parent object*]) — выберите "Lead".

General

Code *
Lead

Title *
Lead

Package
sdkSendCustomNotificationPackage

Description

Inheritance

Parent object *
Lead

Replace parent

4. Создайте **событие объекта**.

- Перейдите в узел [События] ([Events]).
- В блоке [Добавление] ([Adding]) установите признак [После добавления записи] ([After record added]).

Adding

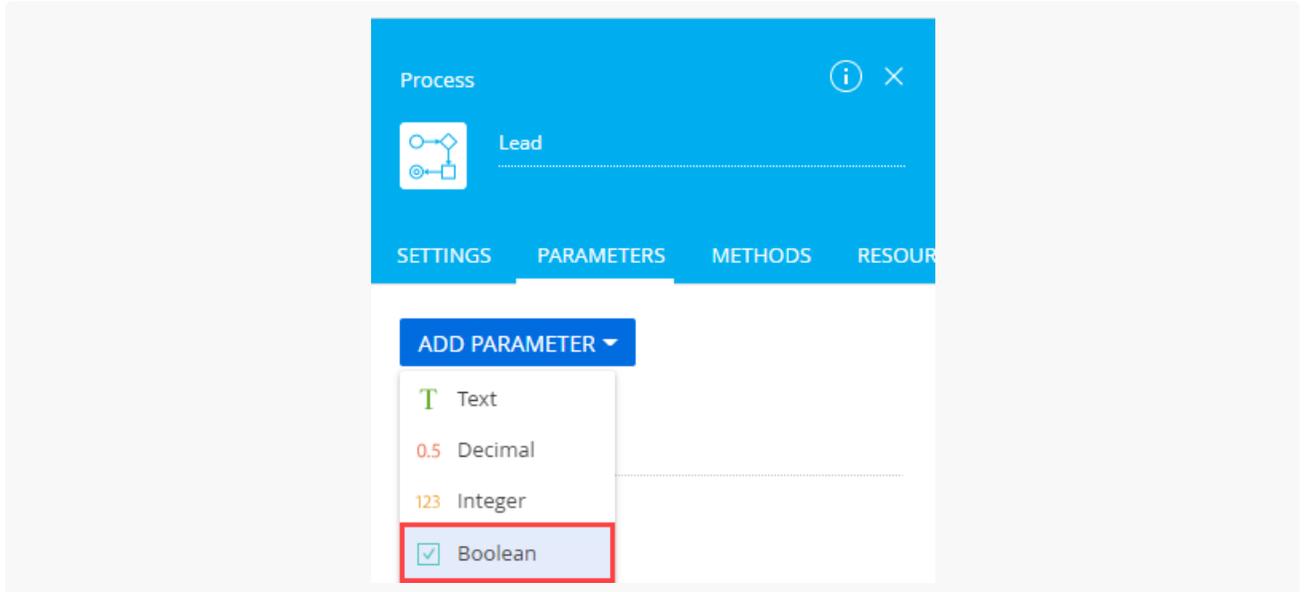
Before record added

After record added

LeadInserted

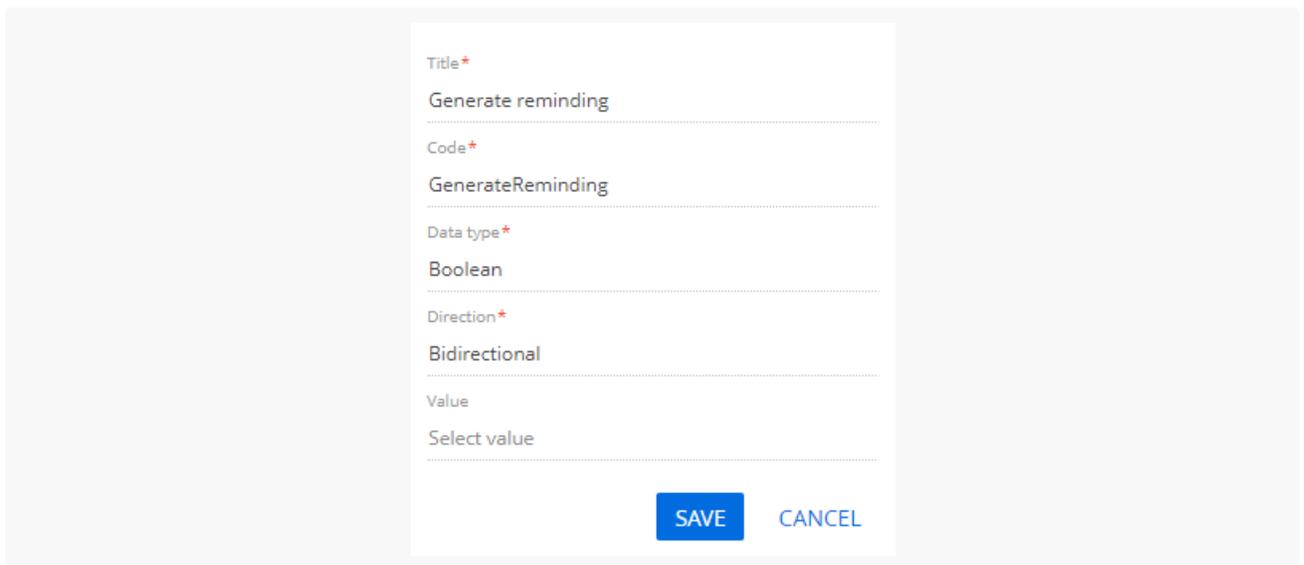
5. В схеме объекта настройте **параметр процесса, который генерирует напоминание**.

- На панели инструментов дизайнера объекта нажмите [Сохранить] ([Save]).
- На панели инструментов дизайнера объекта нажмите [Открыть процесс] ([Open process]).
- Добавьте **параметр**.
 - На панели свойств перейдите на вкладку [Параметры] ([Parameters]) и нажмите [Добавить параметр] → [Логическое] ([Add parameter] → [Boolean]).



b. В дизайнера объекта заполните **свойства параметра**:

- [Заголовок] ([Title]) — "Сгенерировать напоминание" ("Generate reminding").
- [Код] ([Code]) — "GenerateReminding".



e. Нажмите [Сохранить] ([Save]).

d. Переопределите **методы**.

- Переопределите **метод, который вызывается после сохранения объекта**. Для этого на панели свойств перейдите на вкладку [Методы] ([Methods]) и добавьте исходный код переопределенного метода `LeadSaved()`.

`LeadSaved()`

```
public override void LeadSaved() {
    base.LeadSaved();
}
```

```

/* Проверки необходимость генерации напоминания. */
if (!GenerateReminding) {
    return;
}
DateTime remindTime = Entity.NextActualizationDate;
if (Entity.OwnerId.Equals(Guid.Empty) || remindTime.Equals(default(DateTime))) {
    return;
}
/* Создает экземпляр класса UserLeadRemindingText. */
IRemindingTextFormer textFormer = ClassFactory.Get<UserLeadRemindingText>(new Constru
/* Получает название лида. */
string leadName = Entity.LeadName;
/* Получает текст напоминания. */
string subjectCaption = textFormer.GetBody(new Dictionary<string, object> {
    {"LeadName", leadName}
});
/* Получает заголовок напоминания. */
string popupTitle = textFormer.GetTitle(null);
/* Конфигурирует напоминание. */
var remindingConfig = new RemindingConfig(Entity);
/* Автор сообщения – текущий контакт. */
remindingConfig.AuthorId = UserConnection.CurrentUser.ContactId;
/* Целевой получатель – ответственный лида. */
remindingConfig.ContactId = Entity.OwnerId;
/* Тип – напоминание. */
remindingConfig.NotificationTypeId = RemindingConsts.NotificationTypeRemindingId;
/* Дата отправки напоминания – дата актуализации продажи в лиде. */
remindingConfig.RemindTime = remindTime;
/* Текст напоминания. */
remindingConfig.Description = subjectCaption;
/* Заголовок напоминания. */
remindingConfig.PopupTitle = popupTitle;
/* Создает утилитный класс напоминания. */
var remindingUtilities = ClassFactory.Get<RemindingUtilities>();
/* Создает напоминание. */
remindingUtilities.CreateReminding(UserConnection, remindingConfig);
}

```

Значение `RemindingConsts.NotificationTypeRemindingId` КОНСТАНТЫ

`remindingConfig.NotificationTypeId` позволяет отображать напоминание на вкладке напоминаний центра уведомлений. Если для константы указать значение `RemindingConsts.NotificationTypeNotificationId`, то напоминание будет отображено на вкладке служебных сообщений центра уведомлений.

- Переопределите **метод, который вызывается перед сохранением объекта**. Для этого на панели свойств перейдите на вкладку [*Методы*] ([*Methods*]) и добавьте исходный код переопределенного метода `LeadSavingMethod()`.

LeadSavingMethod()

```

public override void LeadSavingMethod() {
    /* Вызов базовой реализации метода. */
    base.LeadSavingMethod();
    /* Получает идентификатор ответственного. */
    var oldOwnerId = Entity.GetTypedOldColumnValue<Guid>("OwnerId");
    /* Получает дату следующей актуализации. */
    DateTime oldRemindDate = Entity.GetTypedOldColumnValue<DateTime>("NextActualizationDate");
    /* Сравнивает старое и новое значения идентификаторов ответственного. */
    bool ownerChanged = !Entity.OwnerId.Equals(oldOwnerId);
    /* Сравнивает старое и новое значения даты актуализации. */
    bool remindDateChanged = !Entity.NextActualizationDate.Equals(oldRemindDate);
    /* Параметру процесса GenerateReminding присваивает true, если были изменены ответственные. */
    GenerateReminding = ownerChanged || remindDateChanged;
}

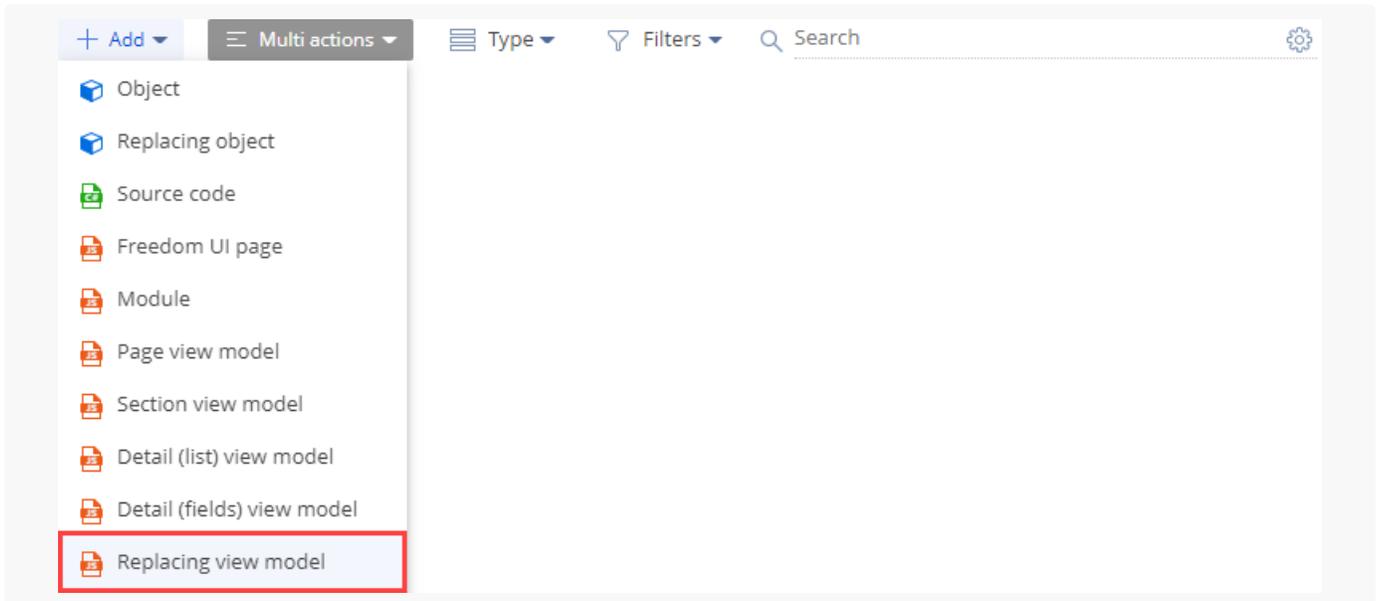
```

Полный исходный код встроенного процесса объекта

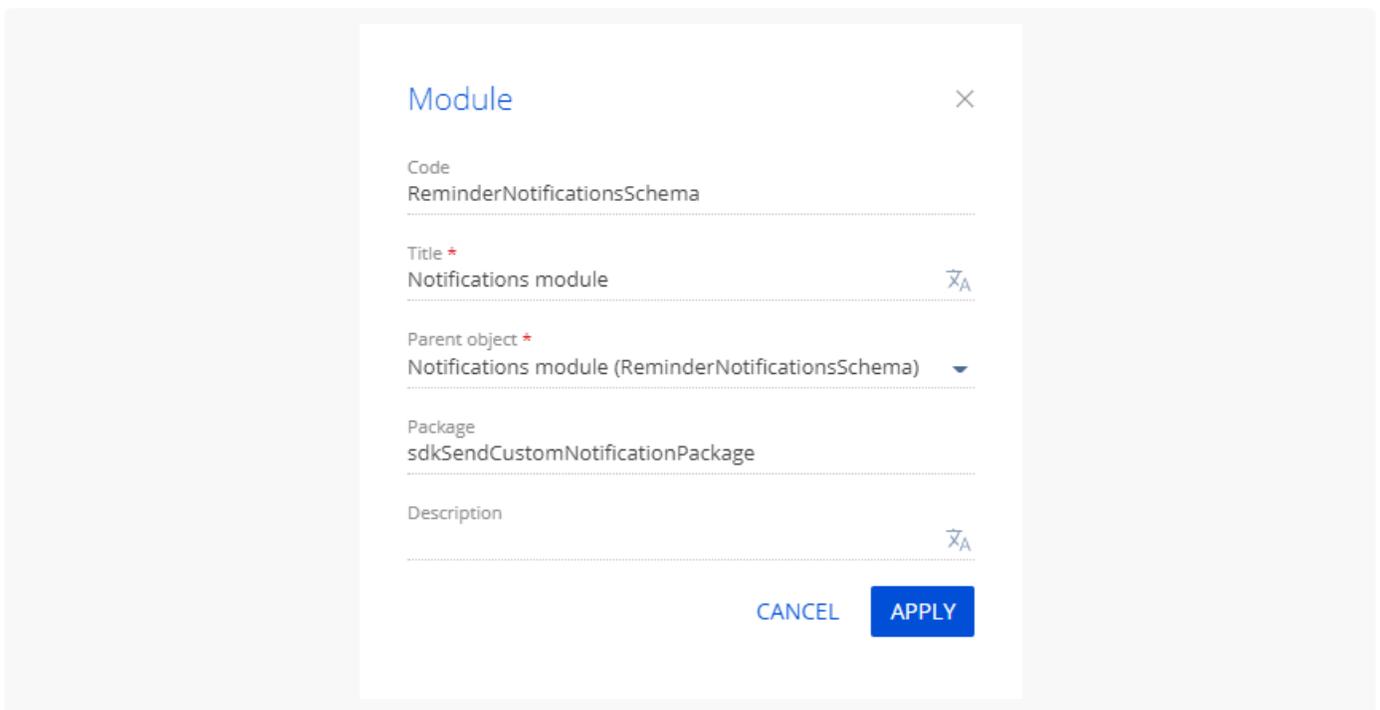
- g. На панели инструментов дизайнера объекта нажмите [*Сохранить*] ([*Save*]), а затем [*Опубликовать*] ([*Publish*]).
 - h. На панели инструментов дизайнера объекта нажмите [*Заккрыть*] ([*Cancel*]).
6. На панели инструментов дизайнера объекта нажмите [*Опубликовать*] ([*Publish*]) для создания соответствующей таблицы в базе данных.

3. Реализовать отображение напоминания

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. В дизайнере модуля заполните свойство схемы [Родительский объект] ([*Parent object*]) — выберите "ReminderNotificationsSchema".



Для применения заданных свойств нажмите [Применить] ([*Apply*]).

4. Реализуйте **отображение напоминания**.

a. В свойстве `methods` реализуйте **методы**:

- `getIsLeadNotification()` — метод, который определяет, относится ли напоминание к лиду.
- `getNotificationSubjectCaption()` — метод, который возвращает заголовок напоминания.

d. В массив модификаций `diff` добавьте **конфигурационные объекты с настройками**

расположения:

- Контейнера на странице.
- Контейнера для заголовка.
- Изображения.
- Даты.
- Текста напоминания.

Исходный код схемы замещающей модели представления представлен ниже.

ReminderNotificationsSchema

```
define("ReminderNotificationsSchema", ["ReminderNotificationsSchemaResources"], function() {
  return {
    /* Название схемы объекта. */
    entitySchemaName: "Reminding",
    /* Методы модели представления. */
    methods: {
      /* Определяет, относится ли напоминание к лиду. */
      getIsLeadNotification: function() {
        return this.get("SchemaName") === "Lead";
      },
      /* Возвращает заголовок напоминания. */
      getNotificationSubjectCaption: function() {
        var caption = this.get("Description");
        return caption;
      }
    },
    /* Отображение напоминания. */
    diff: [
      /* Метаданные для добавления контейнера. */
      {
        /* Выполняется операция добавления элемента на страницу. */
        "operation": "insert",
        /* Мета-имя добавляемого контейнера. */
        "name": "NotificationleadItemContainer",
        /* Мета-имя родительского контейнера, в который добавляется текущий контейнер */
        "parentName": "Notification",
        /* Контейнер добавляется в коллекцию элементов родительского элемента. */
        "propertyName": "items",
        /* Свойства, передаваемые в конструктор элемента. */
        "values": {
          "itemType": Terrasoft.ViewItemType.CONTAINER,
          "wrapClass": [
            "reminder-notification-item-container"
          ],
        },
        /* Отображается только для лидов. */
      }
    ]
  };
});
```

```

        "visible": {"bindTo": "getIsLeadNotification"},
        "items": []
    }
},
/* Метаданные для добавления контейнера для заголовка. */
{
    "operation": "insert",
    "name": "NotificationItemleadTopContainer",
    "parentName": "NotificationleadItemContainer",
    "propertyName": "items",
    "values": {
        "itemType": Terrasoft.ViewItemType.CONTAINER,
        "wrapClass": ["reminder-notification-item-top-container"],
        "items": []
    }
},
/* Метаданные для добавления изображения. */
{
    "operation": "insert",
    "name": "NotificationleadImage",
    "parentName": "NotificationItemleadTopContainer",
    "propertyName": "items",
    "values": {
        "itemType": Terrasoft.ViewItemType.BUTTON,
        "className": "Terrasoft.ImageView",
        "imageSrc": {"bindTo": "getNotificationImage"},
        "classes": {"wrapClass": ["reminder-notification-icon-class"]}
    }
},
/* Метаданные для даты. */
{
    "operation": "insert",
    "name": "NotificationDate",
    "parentName": "NotificationItemleadTopContainer",
    "propertyName": "items",
    "values": {
        "itemType": Terrasoft.ViewItemType.LABEL,
        "caption": {"bindTo": "getNotificationDate"},
        "classes": {"labelClass": ["subject-text-labelClass"]}
    }
},
/* Метаданные для текста напоминания. */
{
    "operation": "insert",
    "name": "NotificationleadSubject",
    "parentName": "NotificationItemleadTopContainer",
    "propertyName": "items",
    "values": {
        "itemType": Terrasoft.ViewItemType.LABEL,

```

```

    "caption": {"bindTo": "getNotificationSubjectCaption"},
    "click": {"bindTo": "onNotificationSubjectClick"},
    "classes": {"labelClass": ["subject-text-labelClass", "label-link", "label-link"], "linkClass": ["subject-link-labelClass", "label-link"], "textClass": ["subject-text-labelClass", "label-link"], "buttonClass": ["subject-button-labelClass", "label-link"], "iconClass": ["subject-icon-labelClass", "label-link"]};
  });
});

```

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

Результат выполнения примера

Чтобы **посмотреть результат выполнения примера**:

1. Обновите страницу раздела [Лиды] ([Leads]).
2. Откройте страницу лида и перейдите на вкладку [Информация о сделке] ([Opportunity info]).
3. В поле [Дата следующей актуализации] ([Next actualization date]) выберите необходимую дату и время актуализации продажи.
4. В поле [Ответственный] ([Owner]) выберите контакт своего пользователя.

The screenshot displays the CRM interface for a lead named 'Bulk email management system / Ronald Gittens'. The interface includes a search bar, navigation buttons (CLOSE, ACTIONS, QUALIFY), and a 'DUPLICATES' section. The main content area shows a progress bar with stages: Qualification, Nurturing, Handoff to sales, Awaiting sale, and Satisfied. Below this, there are communication icons and a message: 'You don't have any tasks yet'. The 'OPPORTUNITY INFO' tab is selected, showing details such as Budget (0.00), Opportunity/Order, and Deal owner. A red box highlights the 'Next actualization date' field, which is set to 4/8/2022 at 7:36 AM.

В результате выполнения примера на вкладке напоминаний центра уведомлений отображается напоминание о дате актуализации продажи в лиде.

