

# 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

<b>Коммуникационная панель</b>	<b>4</b>
Структура коммуникационной панели	4
Контейнеры коммуникационной панели	4
<b>Создать обращение по сообщению из внутренней ленты другого обращения</b>	<b>5</b>
Создать схему замещающей модели представления страницы обращения	5
Результат выполнения примера	8
<b>Скрыть область ленты в едином окне</b>	<b>9</b>
Создать схему замещающей модели представления раздела	9
Результат выполнения примера	11
<b>Отобразить часовой пояс контакта на вкладку коммуникационной панели</b>	<b>12</b>
1. Создать схему замещающей модели представления страницы найденного абонента	12
2. Добавить стили отображения часового пояса	16
Результат выполнения примера	20

# 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:

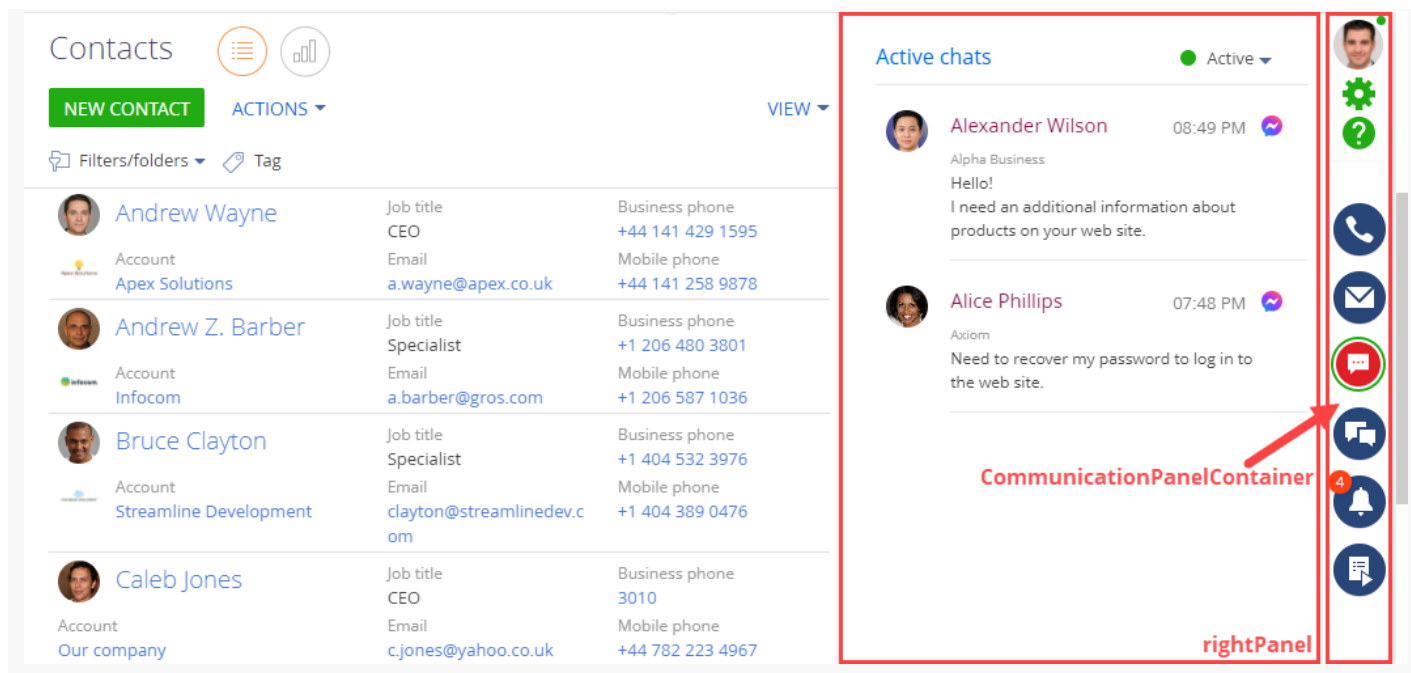
- [ *Единое окно оператора* ] ([ *Operation single window* ]) — проведение менеджером банка консультаций клиенту. Вкладка доступна в продукте Financial Services Creatio, customer journey edition. Позволяет воспользоваться различными критериями для поиска клиента банка, начать проведение консультации, 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 обращение по сообщению из внутренней ленты другого обращения

 Сложный

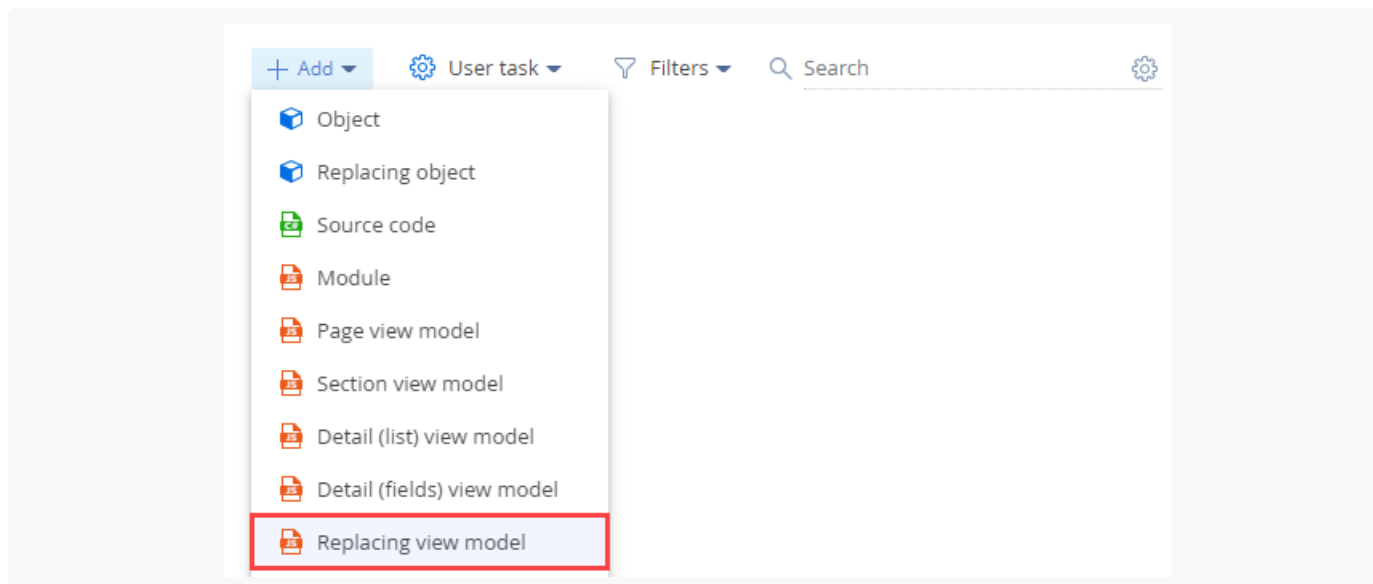
Пример реализован для продуктов линейки Service Creatio.

**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

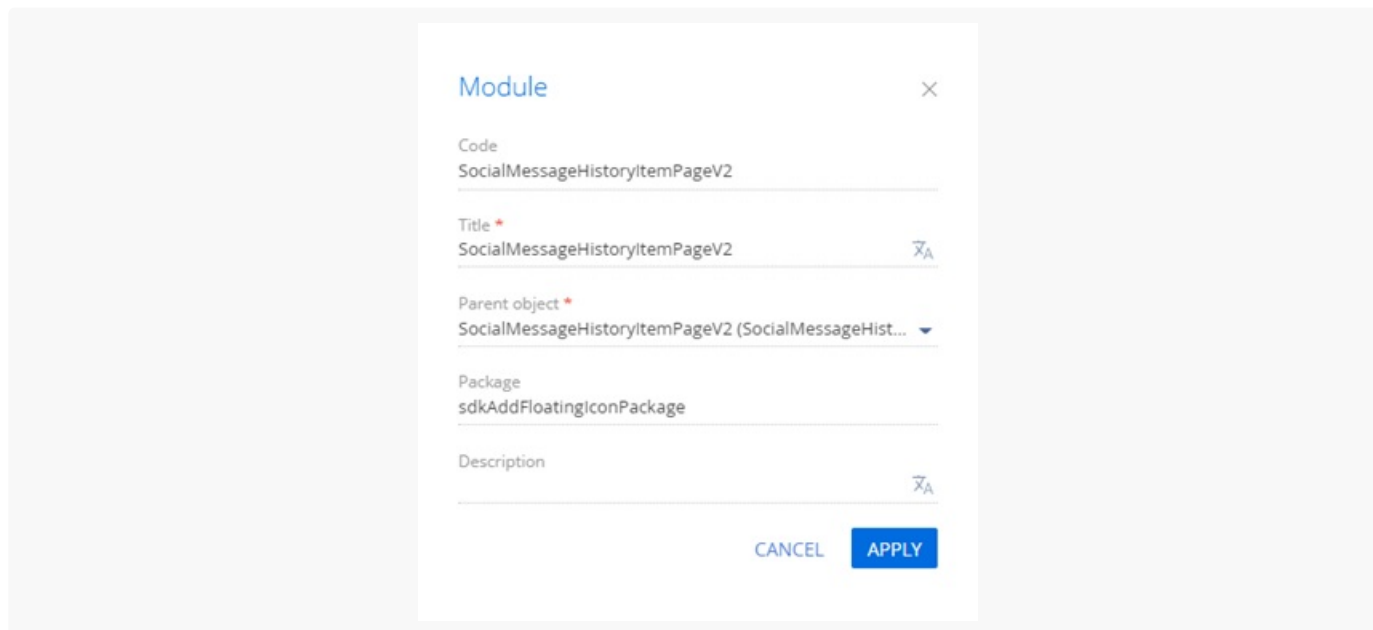
1. [Перейдите в раздел \[ \*Конфигурация\* \]](#) ( [ *Configuration* ] ) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ *Добавить* ] —> [ *Замещающая модель*

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.
  - `onSelectedTextButtonClick()` — 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.



Cases

NEW CASE ACTIONS

Show closed cases Assignee Filters/folders Tag

Number	Subject	Category	Assignee	Status	Resolution time
SR00000048	Consultation on functionality	Service request	Marina Kysla	Waiting for response	11/25/2021 10:00 PM
SR00000047	Unable to create an account on the site	Service request	William Walker	Resolved	11/27/2021 12:05 PM
SR00000046	What's your e-mail address?	Service request	Jason Robinson	New	11/25/2021 3:00 PM
SR00000044	Recovering password to login to the system	Service request	Marina Kysla	In progress	11/25/2021 1:14 AM
SR00000041	Missing data on hard drive	Service request	Marina Kysla	In progress	11/23/2021 1:00 AM
SR00000038	Color laserjet pro CP1025nw - printing issues	Service request	Marina Kysla	In progress	11/24/2021 12:21 PM
SR00000037	Consultation on settings	Service request	Marina Kysla	Waiting for response	11/23/2021 10:00 PM
SR00000036	An unknown error when the computer starts	Incident	Megan Lewis	In progress	11/26/2021 3:00 AM
SR00000030	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

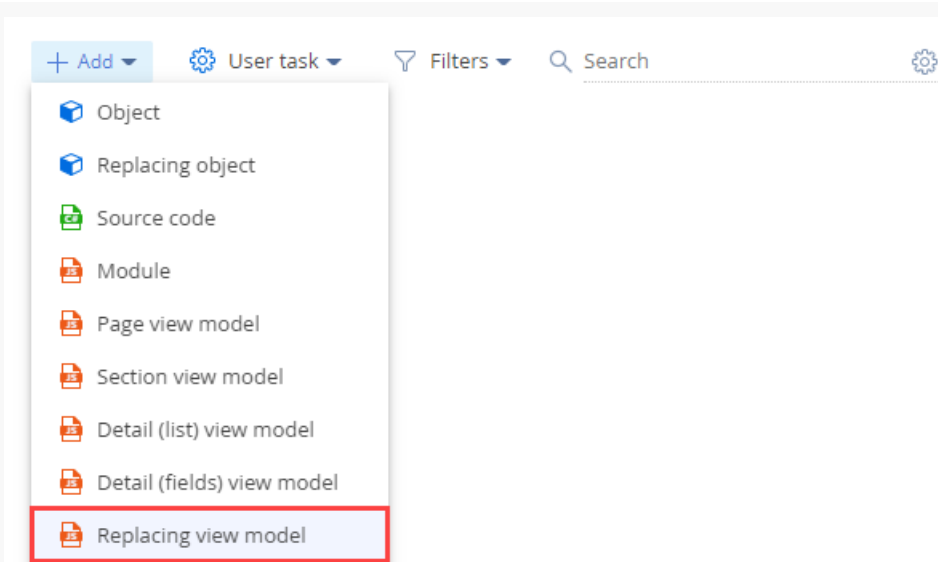
Средний

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 `ENSFeedModule`.
- 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*/
};
});

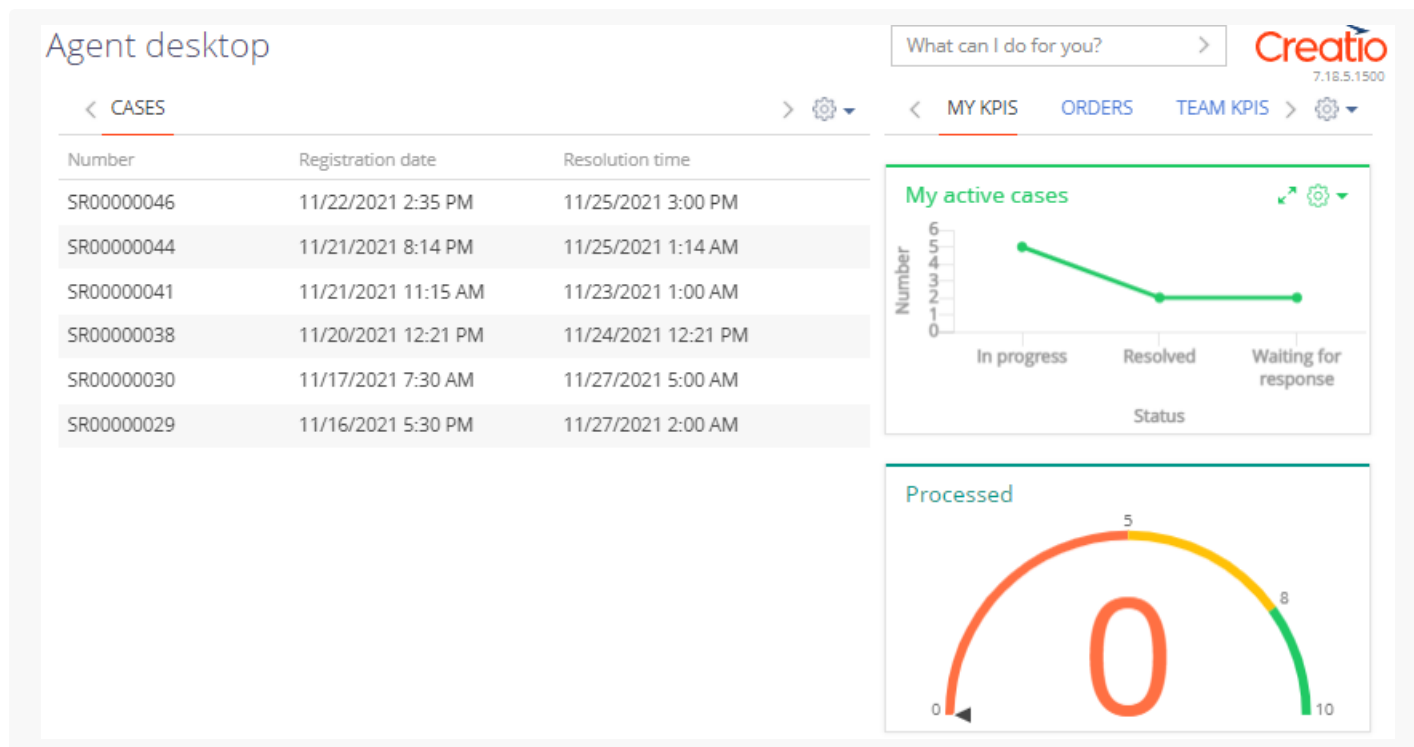
```

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.



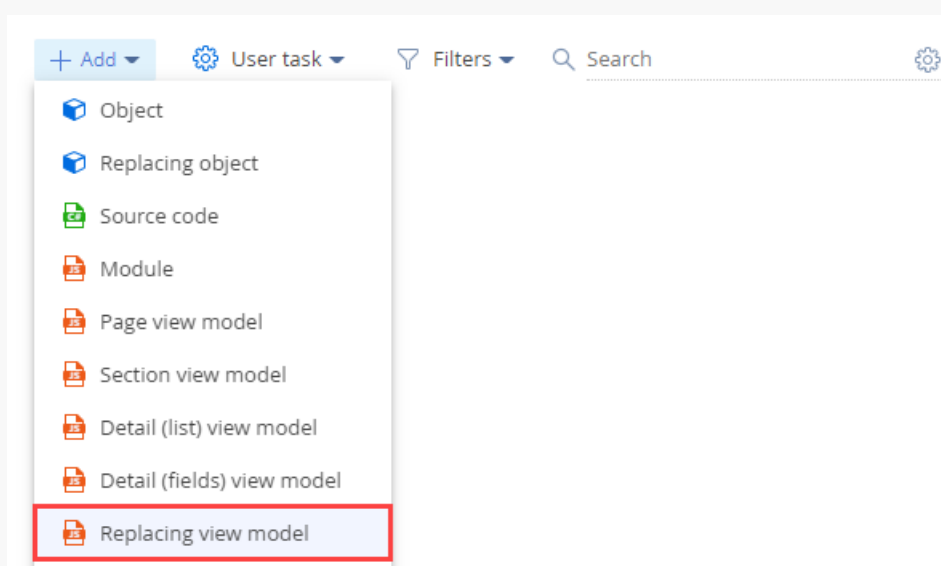
## Отобразить часовой пояс контакта на вкладку коммуникационной панели

 Сложный

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

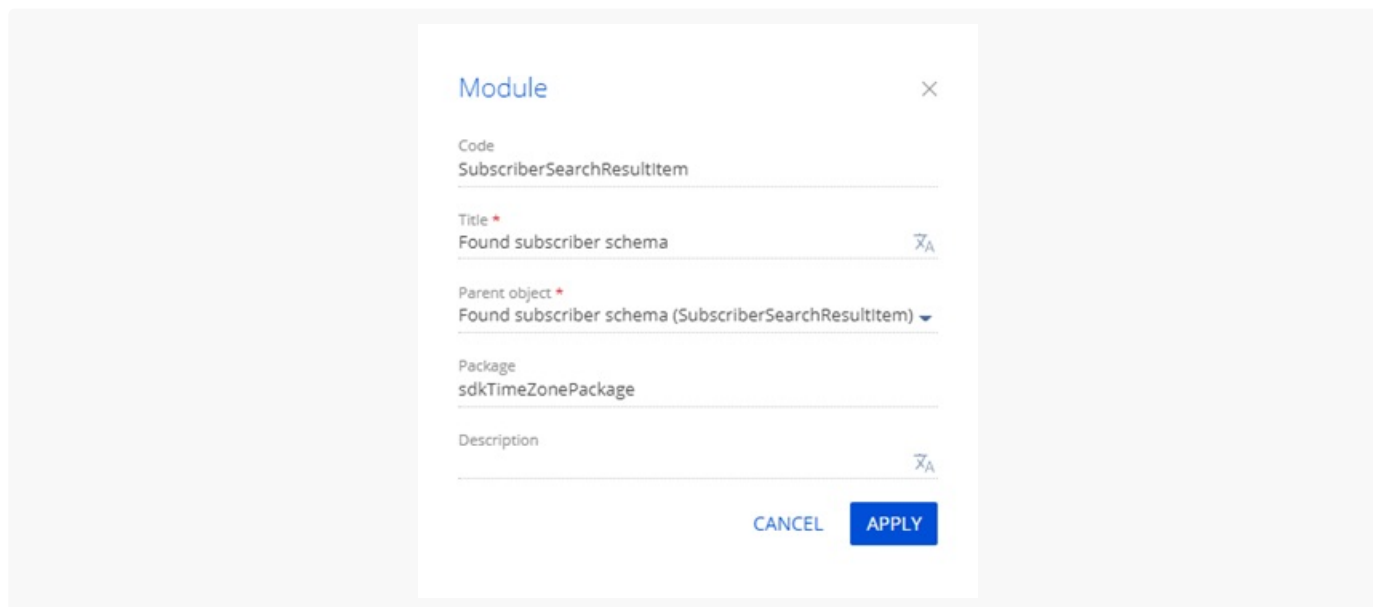
### 1. Создать схему замещающей модели представления страницы найденного абонента

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. Миксин `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 временной зоны контакта и текущее время.
- `TimeZoneCity` — название города, для которого определена временная зона.
- 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.

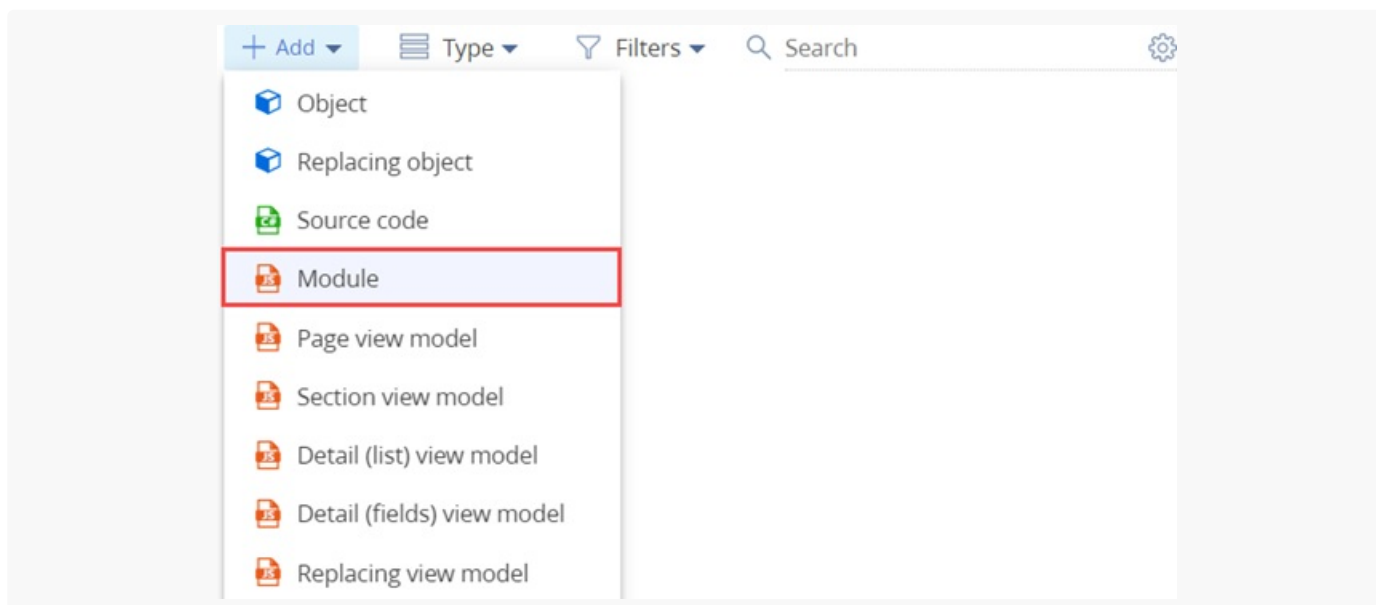
## 2. Добавить стили отображения часового пояса

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".

The screenshot shows a 'Module' configuration window with the following fields:

- Code \***: UsrSubscriberSearchResultItemCSS
- Title \***: SubscriberSearchResultItemCSS
- Package**: sdkTimeZonePackage
- Description**: (empty)

Buttons: CANCEL, APPLY

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**:

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

The screenshot shows a CRM interface with two main sections. The left section, titled 'Accounts', displays details for 'Vertigo Systems', including its website, primary phone number, and address. The right section, titled 'Bruce Clayton', displays his profile information, including his title 'Streamline Develop...' and 'Specialist'. A red box highlights the current time and location '3:30 AM, Atlanta' on the contact card.

- Откройте вкладку [ Звонки ] ([ CTI panel ]) коммуникационной панели.
- Выполните поиск абонента.