# Тестовое приложение «Входящий звонок»

# Аннотация

В данном документе описано учебное демонстрационное приложение, а также способ его воспроизведения.

На примере учебного приложения продемонстрированы некоторые возможности FIS Platform.

Оглавление

[Общее описание приложения 1](#_Toc162001898)

[Начало работы 1](#_Toc162001899)

[Замечания к скриптовому языку UDMS 1](#_Toc162001900)

[Выявление сущностей 1](#_Toc162001901)

[Поисковые методы 1](#_Toc162001902)

[Создание процессов 1](#_Toc162001903)

[Процесс «Генерация\_данных» 1](#_Toc162001904)

[Состав процесса 1](#_Toc162001905)

[Настройка скриптов 1](#_Toc162001906)

[Процесс «Генератор\_звонков» 1](#_Toc162001907)

[Настройка атрибутов процесса 1](#_Toc162001908)

[Состав процесса 1](#_Toc162001909)

[Настройка соединений 1](#_Toc162001910)

[Настройка задержки таймеров 1](#_Toc162001911)

[Настройка скриптов 1](#_Toc162001912)

[Подготовка форм 1](#_Toc162001913)

[Форма "Форма\_оператора" 1](#_Toc162001914)

[Форма "БП" 1](#_Toc162001915)

[Форма "Просмотр\_отчетов" 1](#_Toc162001916)

[Форма "Просмотр\_активного\_менеджера" 1](#_Toc162001917)

[Форма "Просмотр\_клиентов" 1](#_Toc162001918)

[Форма "Просмотр\_данных\_клиента" 1](#_Toc162001919)

[Форма «Поиск\_клиента» 1](#_Toc162001920)

[Форма «Создание\_нового\_клиента» 1](#_Toc162001921)

[Форма «Заполнение\_данных\_претензии» 1](#_Toc162001922)

[Форма «Заключение\_ответственного\_исполнителя» 1](#_Toc162001923)

[Форма «Просмотр\_печатной\_формы» 1](#_Toc162001924)

[Форма «Конец» 1](#_Toc162001925)

[Процесс «Главный» 1](#_Toc162001926)

[Настройка атрибутов процесса 1](#_Toc162001927)

[Состав процесса 1](#_Toc162001928)

[Настройка блоков и соединений 1](#_Toc162001929)

[Настройка атрибутов процесса 1](#_Toc162001930)

[Настройка блоков и событий 1](#_Toc162001931)

[Наполнение форм 1](#_Toc162001932)

[Форма "Просмотр\_активного\_менеджера" 1](#_Toc162001933)

[Настройка атрибутов формы 1](#_Toc162001934)

[Настройка элементов 1](#_Toc162001935)

[Форма "Форма\_оператора" 1](#_Toc162001936)

[Настройка атрибутов формы 1](#_Toc162001937)

[Настройка элементов 1](#_Toc162001938)

[Форма «БП» 1](#_Toc162001939)

[Настройка атрибутов формы 1](#_Toc162001940)

[Настройка элементов формы 1](#_Toc162001941)

[Форма «Просмотр\_отчетов» 1](#_Toc162001942)

[Настройка атрибутов формы 1](#_Toc162001943)

[Настройка элементов формы 1](#_Toc162001944)

[Форма «Поиск\_клиента» 1](#_Toc162001945)

[Настройка атрибутов формы 1](#_Toc162001946)

[Настройка элементов формы 1](#_Toc162001947)

[Форма «Создание\_нового\_клиента» 1](#_Toc162001948)

[Настройка атрибутов формы 1](#_Toc162001949)

[Настройка элементов формы 1](#_Toc162001950)

[Форма «Заполнение\_данных\_претензии» 1](#_Toc162001951)

[Настройка атрибутов формы 1](#_Toc162001952)

[Настройка элементов формы 1](#_Toc162001953)

[Форма «Заключение\_ответственного\_исполнителя» 1](#_Toc162001954)

[Настройка атрибутов формы 1](#_Toc162001955)

[Настройка элементов формы 1](#_Toc162001956)

[Форма "Просмотр\_данных\_клиента" 1](#_Toc162001957)

[Настройка атрибутов формы 1](#_Toc162001958)

[Настройка элементов формы 1](#_Toc162001959)

[Форма "Просмотр\_клиентов" 1](#_Toc162001960)

[Настройка атрибутов формы 1](#_Toc162001961)

[Настройка элементов формы 1](#_Toc162001962)

[Форма «Просмотр\_печатной\_формы» 1](#_Toc162001963)

[Настройка атрибутов формы 1](#_Toc162001964)

[Настройка элементов формы 1](#_Toc162001965)

[Форма «Выход» 1](#_Toc162001966)

[Политики запуска 1](#_Toc162001967)

[Процесс "Генерация\_данных" 1](#_Toc162001968)

[Процесс "Генератор\_звонков" 1](#_Toc162001969)

[Процесс "Главный" 1](#_Toc162001970)

[Процесс "Входящий\_звонок" 1](#_Toc162001971)

[Создание отчетов 1](#_Toc162001972)

[Отчет по клиентам 1](#_Toc162001973)

[Настройка событий 1](#_Toc162001974)

[Настройка связи форм «Форма оператора» и «Просмотр активного менеджера» 1](#_Toc162001975)

[Форма "Просмотр\_активного\_менеджера" 1](#_Toc162001976)

[Настройка связи между формами «Просмотр клиентов» и «Просмотр данных клиента» 1](#_Toc162001977)

[Форма "Просмотр\_клиентов" 1](#_Toc162001978)

[Форма "Просмотр\_данных\_клиента" 1](#_Toc162001979)

[Форма "Просмотр\_клиентов" 1](#_Toc162001980)

[Проверка работы приложения 1](#_Toc162001981)

# Общее описание приложения

В качестве примера создадим приложение для приема и обработки претензий.

Большую часть информации и примеров можно взять из официальной документации к FIS Платформе

<https://doc.caseplatform.tech/>

<https://how-to.caseplatform.tech/>

В приложении будут использоваться функциональные **Объекты** - части платформы для разных задач.

<https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/application_objects.html:0:ru>

* **Типы** и **Перечисления** - для описания данных, хранимых в приложении
* **Процессы** - для описания проводимых действий (по сути блоки функций).  
  Один из ключевых элементов, он отвечает за последовательностей выполняемых элементов, передачу параметров внутри приложения, вызов **Форм** и др.
* **Формы** - для визуального отображения информации (как формы в html+css) и события над ними (как js event)
* **Отчеты** - для формирования документа с данными из приложения

Рекомендуется почитать про **Объекты** в документации, чтобы легче ориентироваться внутри приложения.

1. Для эмуляции поступающих звонков создадим отдельный процесс.  
   В дальнейшем мы его вручную запустим перед запуском главного процесса в целях генерации коллекции готовых к обработке входящих звонков.
2. По номеру поступающего звонка система определяет клиента или позволяет добавить нового.
3. Оператор имеет возможность выбрать звонок из списка поступающих (из тех, что запущены нами в генераторе звонков).  
   Так же дадим возможность оператору в любой момент вернуться к выбору звонков и продолжить ранее прерванное заполнение формы претензии.

Таким образом, в нашем приложении нам потребуются следующие процессы:

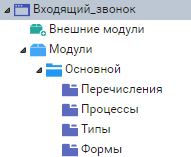
1. Генератор звонков – для создания входящих звонков.
2. Главный – для отображения списка входящих звонков, а также для перехода к приему звонков.
3. Входящий звонок – для приема входящего звонка и обработки претензии.  
   Также в данном процессе создадим возможность добавить нового клиента в базу.

Каждый процесс имеет свой контекстный тип (класс), объекты которого используются для работы бизнес-процесса.

Атрибуты объекта контекстного типа (свойства объекта бизнес процесса) могут быть использованы как входные, так и выходные параметры бизнес-процесса.

# Начало работы

1. Запускаем Конструктор FIS-платформы.
2. Создаем свое Окружение <https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/environment.html:0:ru>   
   2729e708d45924915e8a2dcca8e02e67
3. Внутри Окружения создаём своё приложение <https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/application.html:0:ru>   
   f0945f54bce0aa100670bc3bcbc0b336
4. Открываем появившееся в списке приложение.  
   В панели слева отображается состав приложения - дерево Объектов, а также функциональные кнопки - <https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/application_objects.html:0:ru>
5. fc0ab1c211b55188f49e6459f3205d72Добавьте в модуль **Основной**четыре **Пакета** - Процессы, Типы, Перечисления и Формы



6. Модуль основной должен иметь свойство Привилегированный <https://doc.caseplatform.tech/#1.3.15:0:0:user_guide/application_objects/module/module_operations.html:module-create:ru>

# Замечания к скриптовому языку UDMS

UDMS является интерпретатором JavaScript с дополнительной функциональностью. При выполнении задания необходимо помнить, что язык является регистрозависимым.

JavaScript используется версии 1.5 (~2000 года). Подробнее - <https://doc.caseplatform.tech/#1.3.13:0:0:reference_guide.html:0:ru>

Из JavaScript можно вызывать интерпретатор языка UDML используя оператор **udm:()** . Интерпретатор языка UDML позволяет обращаться к нашим данным, поисковым методам, формам, всем типам и атрибутам которые мы создаем в приложении.

В языке UDML используется Pascal-подобный синтаксис присваивания и сравнения. Оператор присваивания “ **:=** ”, оператор сравнения “ **=** ”.   
В UDML присутствует объект **Этот** , некоторый аналог объекта **this** в других языках, для указания текущего объекта.  
В примере ниже свойствам текущего бизнес процесса разными путями присваиваются значения, также используется оператор **udm:()** .

**JS**

var контекст = udm:(Этот);

udm:(Счетчик := 0); // присваивание свойства текущего объекта через интерпретатор UDM

контекст.Индексы = []; // присваивание через JS оператором “=”

# Выявление сущностей

Выявим сущности (Типы - таблицы базы по сути) и их свойства (атрибуты):

<https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/type_builder.html:0:ru>

* Для **генерации звонков** нам потребуется справочник, содержащий номера телефонов.
* Для обработки **входящих звонков**, нам нужно знать номер звонка, его дату и время, а также статус этого звонка.
* Для обработки данных о **клиенте** нам потребуются его данные, в том числе **мобильный телефон**.
* Для обработки **претензии** нам потребуются данные о **клиенте**, **претензии** и **исполнителе**, принявшем претензию, **реакцию клиента**.

В таблицах ниже описаны Перечисления, Типы и атрибуты, которые необходимо будет создать.

Далее иногда будут использоваться обозначения типов данных, принятые во многих языка программирования:

* **bool** - Логический тип
* **string** - Строка
* **datetime** - Дата и время и т.д.

Добавление Объектов (Типов 459275a1c0a05447a72905429698ee33и Перечислений6e5669a176063f45c476d6e92d5724d3) производится также через панель слева.

Добавление атрибутов производится при выборе добавленного Типа.

Атрибуты вида **class.Основной.Категория\_претензии** и **Основной.Филиал** (ссылающиеся на другой тип) требуют предварительного создания целевого типа.

Атрибуты вида **Основной.Мобильные\_телефоны[]** при создании/изменении помечаются 9e495241666322646d7a7a423610d5cc

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип атрибута** | **Примечание** |
| **Мобильные\_телефоны** | | |
| Номер | Строка |  |
| **Входящий\_звонок** | | |
| Статус | Логическое значение |  |
| Дата\_время | Дата и время |  |
| Номер | Строка |  |
| **Генерация\_номеров** | | |
| Номер | Строка |  |
| **Филиал** | | |
| Название | Строка |  |
| **Исполнитель** | | |
| Фамилия | Строка |  |
| Имя | Строка |  |
| Отчество | Строка |  |
| Доп\_информация | Строка |  |
| Логин | Строка |  |
| ФИО | Строка |  |
| Выбран | Логическое значение |  |
| **Логический\_справочник** | | |
| Значение | Логическое значение |  |
| Название | Строка |  |
| **Претензия** | | |
| Категория | class.Основной.Категория\_претензии |  |
| Направление | class.Основной.Направление\_претензии |  |
| Реакция\_клиента | class.Основной.Реакция\_клиента |  |
| Номер\_телефона | Строка |  |
| Филиал | Основной.Филиал |  |
| Исполнитель | Основной.Исполнитель |  |
| Решение | Логическое значение |  |
| Клиент\_уведомлен | Логическое значение |  |
| Комментарий\_оператора | Строка |  |
| Комментарий\_исполнителя | Строка |  |
| Содержание | Строка |  |
| **Клиент** | | |
| Фамилия | Строка |  |
| Имя | Строка |  |
| Отчество | Строка |  |
| Контактная\_информация | Строка |  |
| Доп\_информация | Строка |  |
| Мобильные\_телефоны | Основной.Мобильные\_телефоны[] | Проставить в свойствах 9e495241666322646d7a7a423610d5cc |
| Звонок | Основной.Входящий\_звонок[] | Проставить в свойствах 9e495241666322646d7a7a423610d5cc |
| Претензия | Основной.Претензия[] | Проставить в свойствах 9e495241666322646d7a7a423610d5cc |
| Дата\_рождения | Дата |  |
| **Печатная форма претензии** | | |
| Дата\_время | Дата и время |  |
| Претензия | Двоичные данные |  |
| ФИО\_клиента | Строка |  |
| **Процесс\_исполнитель** | | |
| Ид\_исполнителя | Строка |  |
| Ид\_процесса | Строка |  |

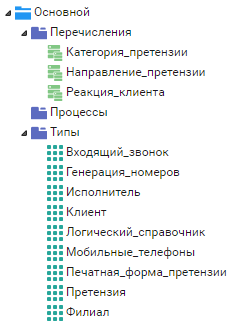
Перечисления:

|  |  |
| --- | --- |
| **Категория\_претензии** | |
| Качество\_обслуживания | Качество обслуживания |
| Списание\_со\_счета | Списание со счета |
| **Направление\_претензии** | |
| Депозиты | Депозиты |
| Кредиты | Кредиты |
| Терминалы | Терминалы |
| **Реакция\_клиента** | |
| Нейтральная | Нейтральная |
| Отрицательная | Отрицательная |
| Положительная | Положительная |

В нашем примере используются перечисления с простыми строчными значениями.

В других проектах перечисления могут быть динамичными, то есть содержать в себе реалтайм вычисляемые значения, используя UDML выражения.

Итоговый вид дерева Объектов:



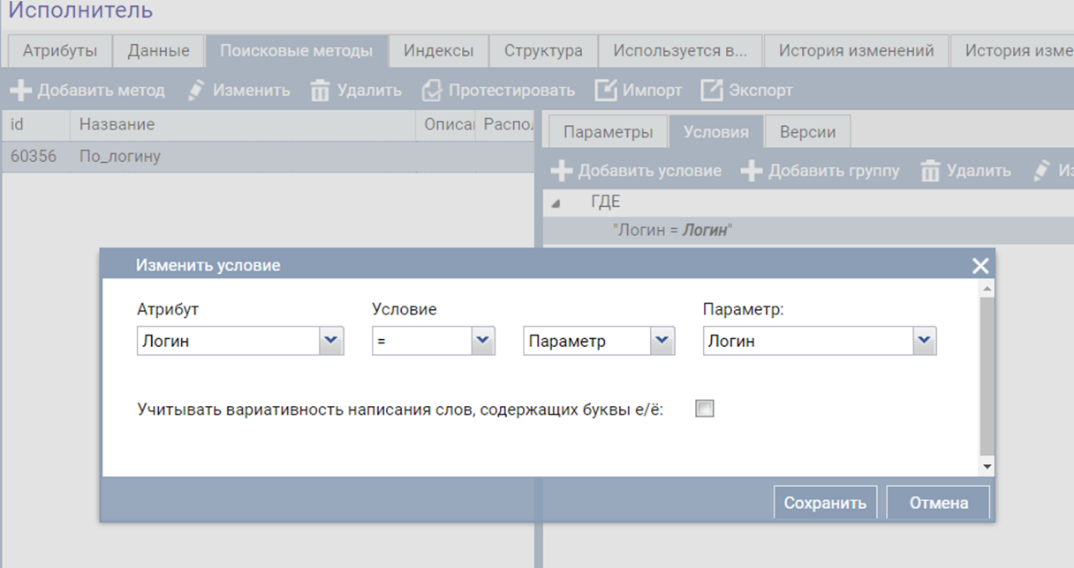
# Поисковые методы

<https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/type_builder/type_search_methods.html:0:ru>

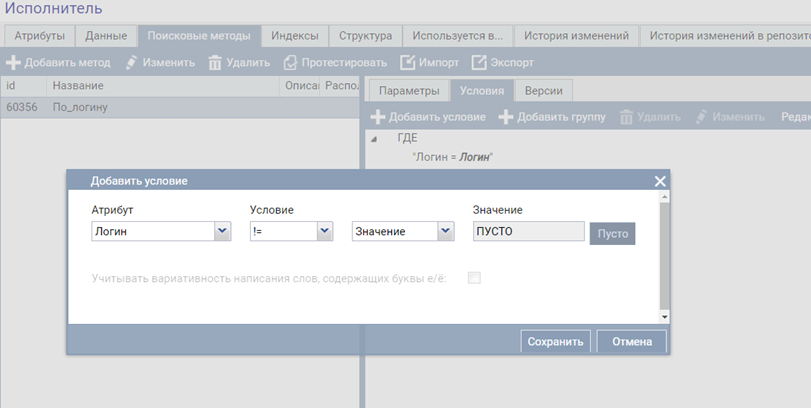
Для некоторых типов нам потребуются поисковые методы, которые используются для выборки коллекций объектов типа с помощью заданных условий или SQL-запросов с возможностью задания параметров поиска.

В нашем случае потребуются поисковые методы, созданные с помощью конструктора запросов.

Условия для запросов:



Для того, чтобы запрос при случайном пустом значении НЕ выдавал весь набор данных добавляем в каждом поисковом проверку на Значение ПУСТО.



И настройка

|  |  |  |  |
| --- | --- | --- | --- |
| **Название типа** | **Название поискового метода** | **Параметр (тип параметра)** | **Условия** |
| Исполнитель | По\_логину | Логин  (строка) | Логин = Логин  И  Логин != ПУСТО |
| По\_фио | ФИО (строка) | ФИО = ФИО  И  ФИО != ПУСТО |
| По\_выбран | Выбран (bool) | Выбран = Выбран  И  Выбран != ПУСТО |
| Клиент | По\_номеру\_телефона | Номер\_телефона  (строка) | Мобильные\_телефоны.Номер = Номер\_телефона  И  Мобильные\_телефоны.Номер != ПУСТО |
| Логический\_справочник | По\_значению | Значение  (Логическое значение) | Значение = Значение  И  Значение != ПУСТО |
| Филиал | По\_названию | Название (строка) | Название = Название  И  Название != ПУСТО |
| Генерация\_номеров | По\_номеру | Номер (строка) | Номер = Номер  И  Номер != ПУСТО |
| Процесс\_исполнитель | По\_идпроцесса | Ид\_процесса (строка) | Ид\_процесса = Ид\_процесса    И    Ид\_процесса != ПУСТО |

# Создание процессов

<https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/process_builder.html:0:ru>

Процесс является важнейшей частью приложения, т.к. задаёт последовательность действий, совершаемых приложением. Например, можно настроить процесс так, чтобы в ходе его выполнения отображались пользовательские формы, рассчитывались скрипты, отправлялись электронные письма или совершались какие-либо другие действия.

Каждый процесс имеет свой контекстный тип (класс), объекты которого используются для работы бизнес-процесса.

Атрибуты объекта контекстного типа могут быть использованы как входные, так и выходные параметры бизнес-процесса.

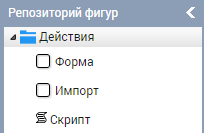
В приложении будут использоваться следующие процессы:

1. Генерация данных
2. Генератор звонков
3. Главный
4. Входящий звонок

## Процесс «Генерация\_данных»

Процесс предназначен для генерации данных при работе с приложением и состоит из последовательных блоков генерации данных для справочников.

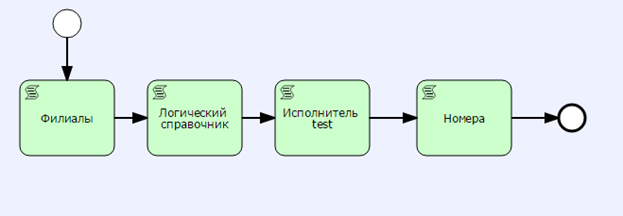
Все блоки в этом процессе скриптовые (иконка в левом верхнем углу).



### Состав процесса

Процесс состоит из следующих блоков:

1. Начальное событие «Старт»
2. Скриптовый блок «Филиалы»
3. Скриптовый блок «Логический справочник»
4. Скриптовый блок «Исполнитель test»
5. Скриптовый блок «Номера»
6. Завершающее событие «Завершение»



Атрибуты fd78cd54e1eaa7f09fbdc4fa264767b3этому процессу не требуются (объект этого процесса не нуждается в своих свойствах).

### Настройка скриптов

В скриптах, кроме обычной логики, используется оператор **udm:()** , который запускает интерпретатор языка UDML для доступа к данным.

**Скриптовый блок Филиалы**

var массив\_филиалов = ["Москва", "Санкт-Петербург", "Казань"];

for (var i=0; i< массив\_филиалов.length; i++){

if (udm:(ЕПУСТО(Основной:Филиал:по\_названию(js:строка(массив\_филиалов[i]))))){

var филиал = udm:(Основной:Филиал:новый());

филиал.Название = массив\_филиалов[i];

}

}

**Скриптовый блок Логический справочник**

if (udm:(ЕПУСТО(Основной:Логический\_справочник:По\_значению(true)))){

var logical = udm:(Основной:Логический\_справочник:Новый());

logical.Значение = true;

logical.Название = "Да";

}

if (udm:(ЕПУСТО(Основной:Логический\_справочник:По\_значению(false)))){

var logical = udm:(Основной:Логический\_справочник:Новый());

logical.Значение = false;

logical.Название = "Нет";

}

**Скриптовый блок Исполнитель test**

if (udm:(ЕПУСТО(Основной:Исполнитель:По\_логину("test")))){

var user = udm:(Основной:Исполнитель:Новый());

user.Фамилия = "Иванов";

user.Имя = "Иван";

user.Отчество = "Иванович";

user.Логин = "test";

}

if (udm:(ЕПУСТО(Основной:Исполнитель:По\_логину("test1")))){

var user = udm:(Основной:Исполнитель:Новый());

user.Фамилия = "Васильев";

user.Имя = "Василий";

user.Отчество = "Васильевич";

user.Логин = "test1";

}

if (udm:(ЕПУСТО(Основной:Исполнитель:По\_логину("test2")))){

var user = udm:(Основной:Исполнитель:Новый());

user.Фамилия = "Петров";

user.Имя = "Петр";

user.Отчество = "Петрович";

user.Логин = "test2";

}

**Скриптовый блок Номера**

var массив\_номеров = [

"+7 100 100 1111",

"+7 200 200 1112",

"+7 300 300 1113",

"+7 400 400 1114",

"+7 500 500 1115",

"+7 600 600 1111",

"+7 700 700 1112",

"+7 800 800 1113",

"+7 900 900 1114",

"+7 110 110 1115",

"+7 000 000 0001"

];

for (var i=0; i< массив\_номеров.length; i++){

if (udm:(ЕПУСТО(Основной:Генерация\_номеров:по\_номеру(js:строка(массив\_номеров[i]))))){

var номер = udm:(Основной:Генерация\_номеров:новый());

номер.Номер = массив\_номеров[i];

}

}

## Процесс «Генератор\_звонков»

Процесс предназначен для генерации звонков.

Процесс сбрасывает «счетчик», а затем с указанной задержкой создает заданное количество звонков.

### Настройка атрибутов процесса

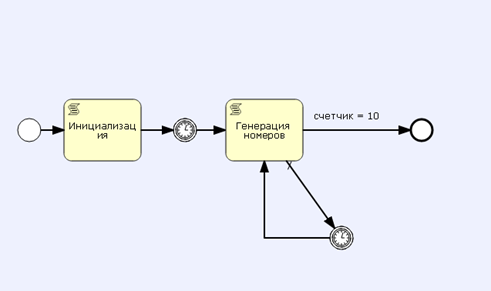
Для работы данному процессу потребуются атрибуты da11d11890708a4b90176baabca99ab2(панель справа):

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| Результат | string | создается автоматически для всех процессов для хранения результата выполнения процесса |
| Индекс | string | для хранения текущего индекса при генерации |
| Телефон | string | для хранения номера телефона, соответствующего индексу |
| Счетчик | float | для указания номера текущей итерации |
| Индексы | string[] | набор неповторяющихся сгенерированных индексов |

### Состав процесса

Процесс состоит из следующих блоков:

1. Начальное событие «Старт»
2. Скриптовый блок «Инициализация»
3. Задержка по таймеру
4. Скриптовый блок «Генерация номеров»
5. Задержка по таймеру
6. Завершающее событие «Завершение»



### Настройка соединений

Важный момент - Соединения <https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/process_builder/process_editor/shape_repository/process_connectors.html:0:ru>

ConditionType  - тип соединения. Значение может быть равно:

• Standart  - это значение по умолчанию, используется для безусловного перехода. В этом случае, даже если указано условие перехода  Condition Expression, оно не проверяется и считается выполненным.

• DefaultFlow  - маршрут по умолчанию, используется для перехода в случае, если условия других переходов не выполнены. В этом случае, даже если указано условие перехода  Condition Expression, оно не проверяется.

• ConditionalFlow  - переход по условию, используется обычно совместно с  [операторами](http://urdc-halva-app-test3.sovcombank.group:8080/doc/shluz_proc.htm). В этом случае должно быть также задано свойство  Condition Expression. Переход совершается при выполнении условия.

В данном Процессе используются:

* **DefaultFlow** - cоединение от Генерации номеров к Задержке по таймеру
* **ConditionalFlow** - соединение от Генерации номеров к Завершению  
  Условие в данном случае задаётся в виде:

**Condition expression для Соединения**

Счетчик == 10

### Настройка задержки таймеров

Для первого таймера задержки установлены свойства:

* прерывающий (boolean) **- true**
* задержка (timeduration) – **РТ4S**   
  т.е. 4 секунды

Для второго таймера:

* прерывающий (boolean) – **true**
* задержка (timeduration) – **РТ2S**   
  т.е. 2 секунды

### Настройка скриптов

После настройки соединений и таймеров можно перейти к наполнению блоков процесса.

**Скриптовый блок Инициализация**

var контекст = udm:(Этот);

udm:(Счетчик := 0);

контекст.Индексы = [];

**Скриптовый блок Генерация номеров**

var номера = udm:(Основной:Генерация\_номеров:все().Номер);

var индекс;

var fl;

var номер;

var индексы = udm:(Индексы);

var i = udm:(Счетчик);

индекс = Math.round(Math.random()\*9);

fl=0;

for (var j=0; j<индексы.length; j++){

if (индексы[j]==индекс){

fl=1;

}

}

if (fl==0){

индексы[i] = индекс;

номер = номера[индекс];

udm:(Индекс := js:String(индекс));

udm:(Телефон := js:String(номер));

//запускаем процесс «Входящий звонок» с полученными данными

udm:(ЗАПУСТИТЬПРОЦЕСС("Основной.Процессы.Входящий\_звонок",["Номер"], Телефон));

i++;

}

//присваиваем значение равное i контекстному типу «счетчик»

udm:(Счетчик:= js:число(i));

## Подготовка форм

Для процесса **Главный** и **Входящий\_звонок** (которые будут добавлены сразу после этого пункта) необходимо ссылаться на Формы - визуальные представления.

<https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/form_builder.html:0:ru>

Форма  - объект, используемый в процессе для взаимодействия с пользователями приложений. Фактически формы формируют интерфейс приложения.

Сейчас в пакете Формы мы будет добавлять новые объекты-формы и их атрибуты.

Для процесса Главный

1. **Форма\_оператора** – для отображения сгенерированных (в нашем случае) звонков.
2. **БП** – отображает формы процесса «входящий звонок», использует виджет «Окно процесса».
3. **Просмотр\_отчетов** – для просмотра отчетов
4. **Просмотр\_активного\_менеджера** - для выбора и просмотра менеджера, который принимает входящий звонок
5. **Просмотр\_клиентов** - для просмотра всех клиентов, которые когда-либо звонили
6. **Просмотр\_данных\_клиента** - для редактирования персональных данных клиента

Для процесса Входящий звонок

1. **Поиск\_клиента** – для поиска клиентов в БД
2. **Создание\_нового\_клиента** – для заведения клиента в базу
3. **Заполнение\_данных\_претензии** – для заполнения данных претензии
4. **Заключение\_ответственного\_исполнителя** – для ввода заключения исполнителя.
5. **Просмотр\_печатной\_формы** – для просмотра печатной формы претензии
6. **Конец** - для завершения процесса обработки звонка

### Форма "Форма\_оператора"

Используется в процессе «Главный» для отображения сгенерированных (в нашем случае) звонков.

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Результат | да | строка |
| Активная\_претензия | да | строка |

### Форма "БП"

Используется в процессе « **Главный** », использует виджет «Окно процесса», который является контейнером для форм процесса «входящий звонок».

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Активная\_претензия | Да | Строка |
| Результат | Да | Строка |

### Форма "Просмотр\_отчетов"

Используется в главном процессе для просмотра любого отчета.

Атрибуты:

| **Атрибут** | **Входной** | **Тип** |
| --- | --- | --- |
| Отчет | Нет | Основной.Печатная\_форма\_претензии |
| Результат | да | Строка |

### Форма "Просмотр\_активного\_менеджера"

Используется в процессе "Главный" для выбора и просмотра менеджера, который принимает входящий звонок.

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Активный\_менеджер | Нет | строка |

### Форма "Просмотр\_клиентов"

Используется в процессе "Главный" для просмотра всех клиентов, которые когда-либо звонили.

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Активный\_клиент | Нет | Основной.Клиент |

### Форма "Просмотр\_данных\_клиента"

Используется в процессе "Главный" для редактирования персональных данных клиента.

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Фамилия | Нет | Строка |
| Имя | Нет | Строка |
| Отчество | Нет | Строка |
| Дата\_рождения | Нет | Строка |
| Ид | Нет | Строка |

### Форма «Поиск\_клиента»

Поиск клиента – используется в процессе «входящий звонок» для поиска клиентов в БД.

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Клиент | да | Основной.Клиент |
| Номер | да | строка |
| Результат | да | строка |

### Форма «Создание\_нового\_клиента»

Создание нового клиента – используется в процессе «входящий звонок»  для заведения клиента.

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Клиент | да | Основной.Клиент |
| Мобильный\_телефон | да | строка |

### Форма «Заполнение\_данных\_претензии»

Заполнение данных претензии – используется в процессе «входящий звонок» для заполнения данных претензии.

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Претензия | да | Основной.Претензия |

### Форма «Заключение\_ответственного\_исполнителя»

Заключение ответственного исполнителя – используется в процессе «входящий звонок», для ввода заключения исполнителя.

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Клиент | да | Основной.Клиент |
| Претензия | да | Основной.Претензия |
| Результат | да | строка |

### Форма «Просмотр\_печатной\_формы»

Используется для просмотра печатной формы претензии.

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Печатная\_форма | да | Основной.Печатная\_форма\_претензии |

### Форма «Конец»

Используется в процессе «входящий звонок» для завершения процесса.

Атрибуты и адаптеры для формы не требуются.

Атрибуты:

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Результат | Да | Строка |

Теперь Формы-заготовки присутствуют в приложении и на них можно ссылаться.

Их наполнением и оформлением нужно будет заняться позже.

## Процесс «Главный»

Выводит форму оператора с таблицей, содержащей текущие входящие звонки.

После выбора звонка запускается процесс **Входящий\_звонок** **.**

После обработки входящего звонка, возможно: либо вернуться к начальной форме, либо закончить работу с приложением.

Для демонстрации добавлена возможность вернуться на главную форму на любом шаге обработки входящего звонка.

### Настройка атрибутов процесса

Для данного процесса установлены следующие атрибуты b6a5e73f98df9e8bdc90c68dfb9c5093:

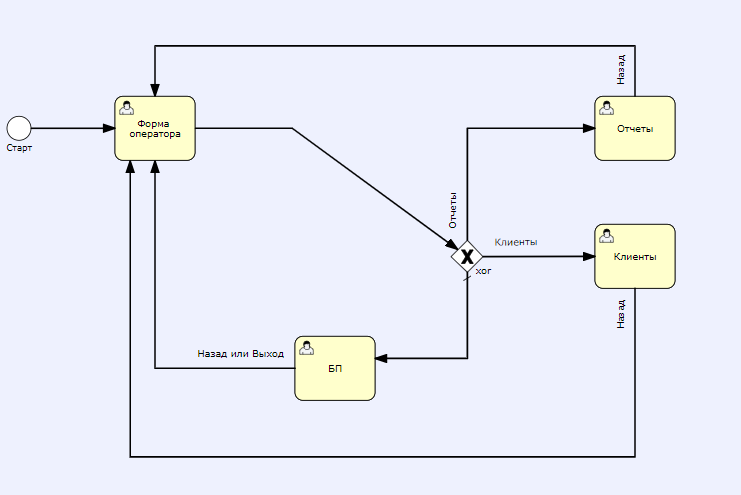
|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| Результат | string | для хранения результата выполнения блока (создается автоматически) |
| Активная\_претензия | string | для хранения id потока процесса «Входящий звонок» |

### Состав процесса

Процесс состоит из следующих блоков:

1. Начальное событие «Старт»
2. Блок Формы «Форма оператора»
3. Блок Оператора исключающее ИЛИ
4. Блок Формы «Отчеты»
5. Блок Формы «БП»
6. Блок Формы "Клиенты"

Обращаем внимание на то, что в данном процессе нет Завершающего события. Процесс цикличен.



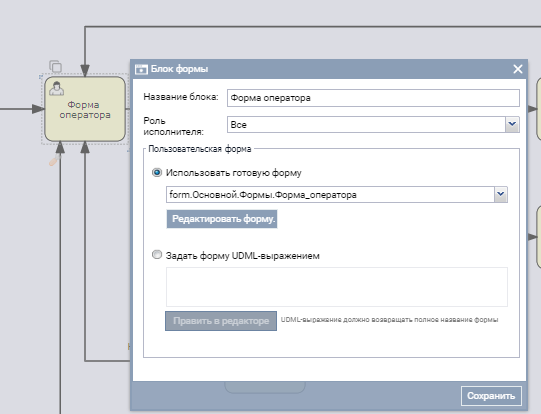
### Настройка блоков и соединений

**Важно:**  Данный процесс, как и процесс " **Входящий\_звонок** " использует элементы фигур " **Форма** " (помимо **Скрипта** ), с которыми необходимо установить связь.

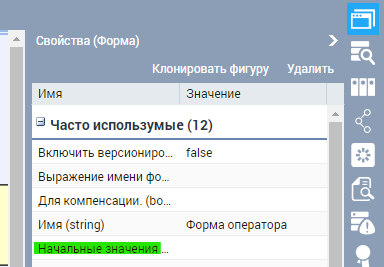
#### ****Блок формы "Форма оператора"****

Для этой формы установлены следующие свойства:

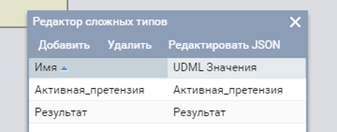
* Имя (string) – Форма оператора.
* Роль исполнителя (string) – Все (или возможно установить требующуюся нам роль, например, Оператор).
* Форма (text) - выбрана форма «Форма оператора».



Далее необходимо задать начальные значения:



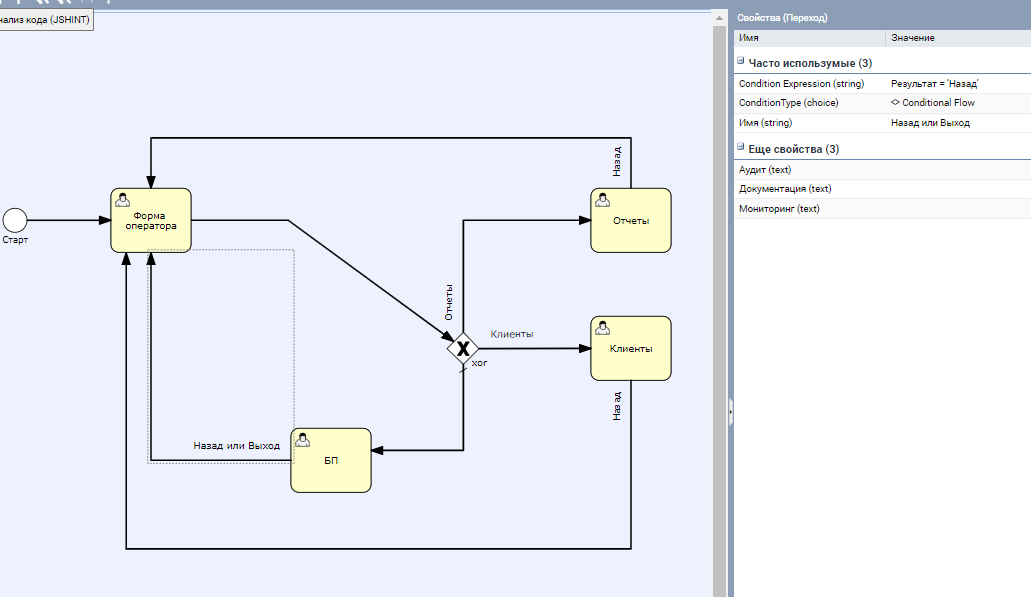
Начальное значение для формы:



Свойства условия перехода к форме «Форма оператора»

Свойства условия перехода «Назад или Выход» от формы «БП» к форме «Форма оператора»

* ConditionType (choice) – Standard
* Имя (string) – Назад или Выход



Свойства условия двух переходов «Назад» от форм «Клиенты» и «Отчеты» к форме «Форма оператора»

* ConditionType (choice) – Standard
* Имя (string) – Назад

#### 2909a80e4900500b849c4a960f8e3632

#### ****Блок формы "БП"****

Для этой формы установлены следующие свойства:

* Имя (string) – БП.
* Роль исполнителя (string) – Все (или возможно установить требующуюся нам роль, например, Оператор).
* Форма (text) - выбрана форма «БП».

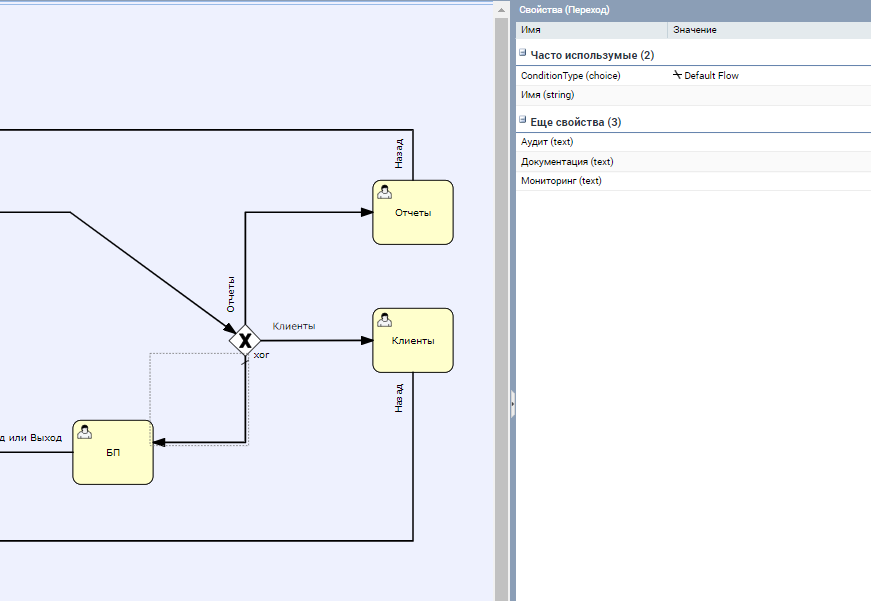


Начальные значения для формы:



Свойства условия перехода к форме «БП».

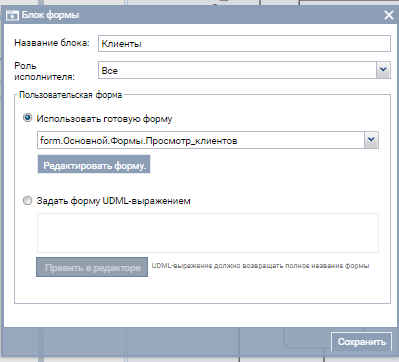
* ConditionType (choice) – Default Flow



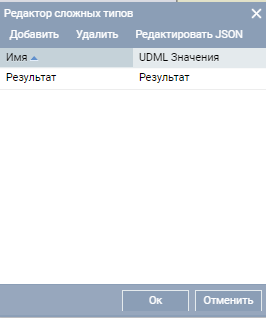
#### ****Блок формы "Клиенты"****

Для этой формы установлены следующие свойства:

* Имя (string) – Клиенты.
* Роль исполнителя (string) – Все (или возможно установить требующуюся нам роль, например, Оператор).
* Форма (text) - выбрана форма «Просмотр\_клиентов».



Начальные значения для формы:



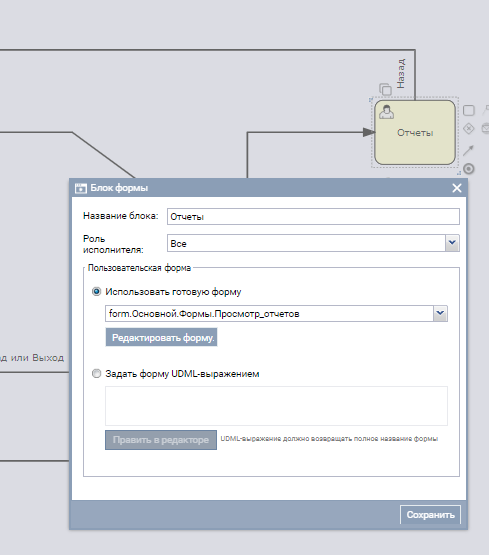
Свойства условия перехода к форме «Клиенты».

* ConditionType (choice) – Default Flow
* Condition Expression (string) – Результат = "Клиенты"

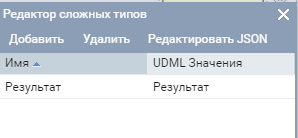
#### ****Блок формы "Отчеты"****

Для этой формы установлены следующие свойства:

* Имя (string) – Отчеты.
* Роль исполнителя (string) – Все (или возможно установить требующуюся нам роль, например Оператор).
* Форма (text) - выбрана форма «Просмотр\_отчетов».

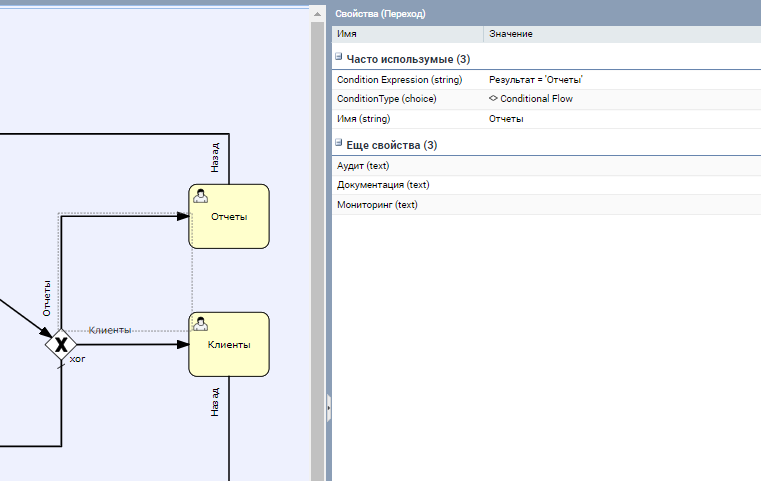


Начальное значение формы “Отчеты”



**Свойства условия перехода к форме «Отчеты».**

* Condition Expression (string) – Результат == "Отчеты"
* ConditionType (choice) – Conditional Flow



**Процесс «Входящий звонок»**

Основной процесс обработки входящего звонка.

При запуске процесс инициализирует звонок и находит клиента в базе по номеру, если он существует.

Затем выводится форма поиска клиента с данными по найденному номеру. Если номер не найден, отображается только номер телефона. Данная форма позволяет оператору:

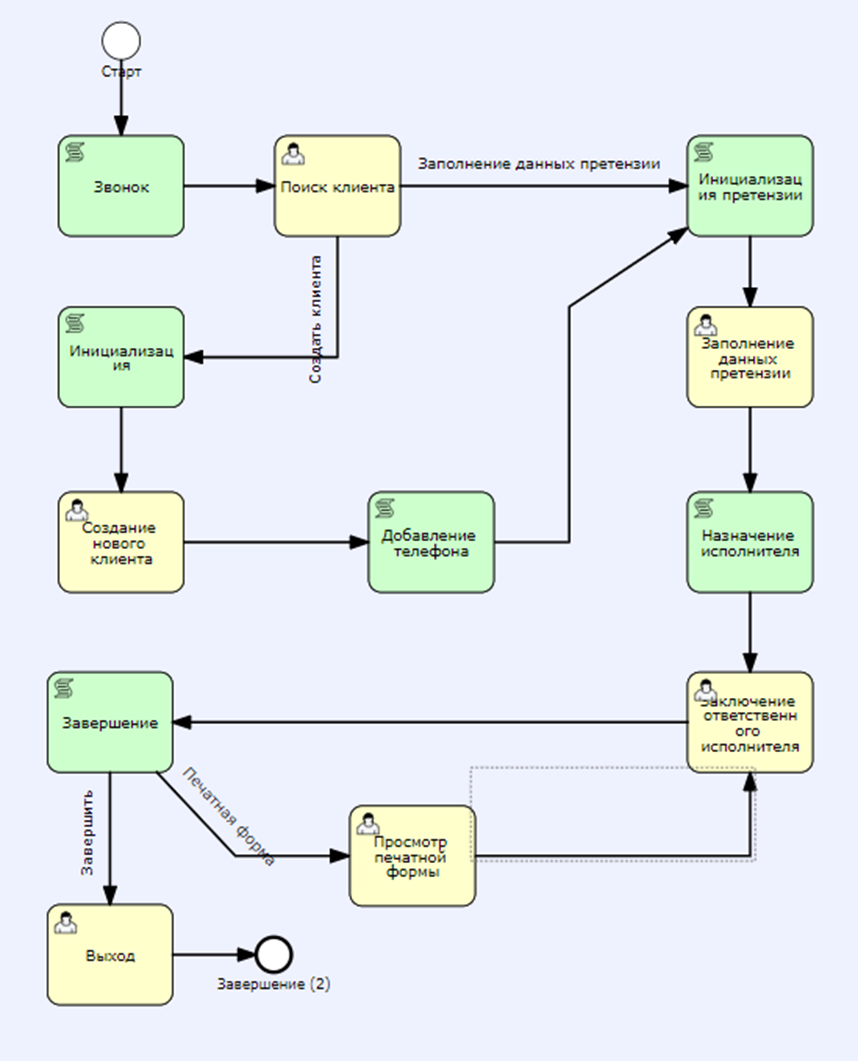
* создать нового клиента;
* перейти к заполнению данных претензии.

### Настройка атрибутов процесса

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| Результат | string | для хранения результата выполнения блока (создается автоматически) |
| Звонок | Основной.Входящий\_звонок | для работы с текущим звонком |
| Номер | string | для хранения  номера текущего звонка |
| Клиент | Основной.Клиент | для работы с текущим клиентом |
| Претензия | Основной.Претензия | для работы с текущей претензией |
| Мобильный\_телефон | Основной.Мобильные\_телефоны | для работы с текущим мобильным телефоном |
| Печатная\_форма | Основной.Печатная\_форма\_претензии | Для работы с печатной формой претензии |
| Ид\_исполнителя | string | Для связи процесса с исполнителем |

Процесс состоит из следующих блоков:

1. Начальное событие «Старт».
2. Скриптовый блок «Звонок».
3. Форма «Поиск клиента» с разделением на две ветки:
   1. Ветка «создание нового клиента»  
      После инициализации выводится форма создания нового клиента. После заполнения полей и нажатия на кнопку «Добавить», выполняется скрипт добавления клиента в БД и переход к заполнению данных претензии.
   2. Ветка «заполнение данных претензии»  
      После инициализации выводится форма заполнения данных претензии. После заполнения полей и нажатия кнопки «Далее», назначается исполнитель и выводится форма заключения ответственного исполнителя. После заполнения полей и нажатия кнопки «Завершить», данные заносятся в БД и процесс завершается
4. Скриптовый блок «Инициализация» (для ветки добавления клиента)
5. Форма «Создание нового клиента»
6. Скриптовый блок «Добавление телефона»
7. Скриптовый блок «Инициализация претензии»
8. Форма «Заполнение данных претензии»
9. Скриптовый блок «Назначение исполнителя»
10. Форма «Заключение ответственного исполнителя»
11. Скриптовый блок «Завершение»
12. Завершающее событие «Завершение»



### Настройка блоков и событий

#### Начальное событие «Старт».

Как и любой процесс, «Входящий звонок» начинается со стартового события. Используются все свойства по умолчанию.

#### Скриптовый блок Звонок

Представляет собой скрипт, который создает новый экземпляр звонка и заполняет его статус, дату\_время и находит клиента по его номеру.

Для этой задачи установлены следующие свойства:

* Имя – «Звонок».
* Тип задачи – Скрипт.

**Скриптовый блок Звонок**

udm:(НАЗНАЧИТЬНАРОЛЬ("Все", текущийпользователь()));

udm:(Звонок := Основной:Входящий\_звонок:Новый());

var контекст = udm:(Этот);

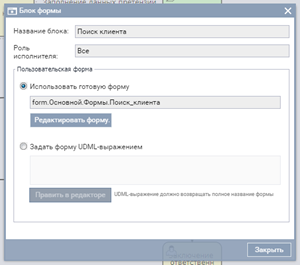
udm:(Звонок.Дата\_время := СЕЙЧАС());

udm:(Звонок.Номер := Номер);

udm:(Клиент := Основной:Клиент:По\_номеру\_телефона(Номер)[0]);

#### Блок формы «Поиск клиента».

Для этой формы установлены следующие свойства:

* Имя – Поиск клиента.
* Роль исполнителя (string) – Все (или возможно установить требующуюся нам роль, например Оператор)
* Форма (text) - выбрана форма «Поиск\_клиента»  
  
* Начальные значения:  
  

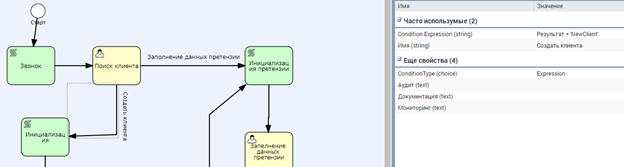
Если клиент найден, выполняется переход к ветке заполнения претензии.

Если клиент не найден, значение атрибута «Результат» контекстного типа становится равным ‘NewClient’ и выполняется переход к ветке добавления клиента.

**Свойства условия перехода к ветке заполнения претензии**

* ConditionType (choice) – Conditional Flow
* Condition Expression (string) **Результат = 'Заполнение данных претензии'**

**Свойства условия перехода к ветке добавления клиента**

* ConditionType (choice) – Conditional Flow
* Condition Expression (string) - **Результат == 'NewClient'**   
  

#### Скриптовый блок «Инициализация» (для ветки добавления клиента)

Задача представляет собой скрипт, создающий новые поля в атрибутах для типов «Клиент» и «Мобильные\_телефоны».

Для этой задачи установлены следующие свойства:

* Имя – «Инициализация».
* Тип задачи – Скрипт.

**Скриптовый блок Инициализация**

udm:(Клиент := Основной:Клиент:Новый());

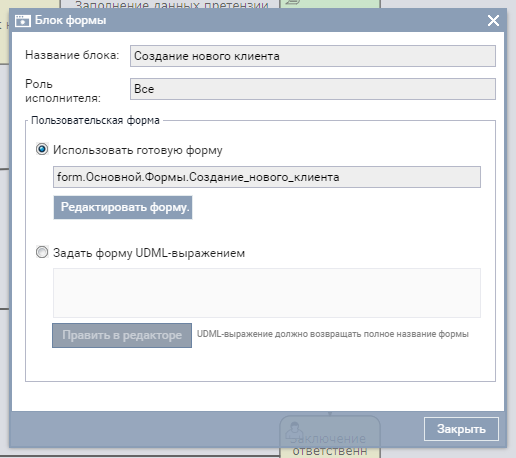
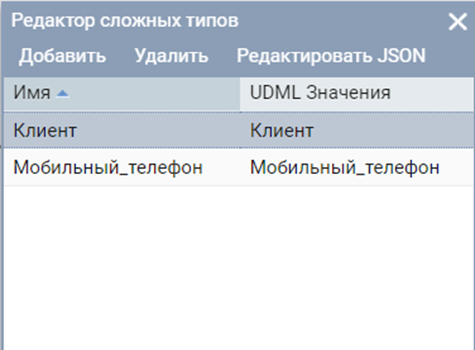
udm:(Мобильный\_телефон := Основной:Мобильные\_телефоны:Новый());

udm:(Мобильный\_телефон.Номер := Номер);

udm:(ДОБАВИТЬ(Клиент.Мобильные\_телефоны, Мобильный\_телефон));

#### Блок формы «Создание нового клиента»

Для этой формы установлены следующие свойства:

* Имя (string) – Создание нового клиента.
* Роль исполнителя (string) – Все (или возможно установить требующуюся нам роль, например Оператор).
* Форма (text) - выбрана форма «Создание\_нового\_клиента».  
  
* Начальное значение формы “Создание нового клиента”  
  

#### Скриптовый блок "Добавление телефона"

Представляет собой скрипт, который добавляет номер телефона в атрибут «Мобильный\_телефон» контекстного типа.

Для этой задачи установлены следующие свойства:

* Имя – «Добавление телефона».
* Тип задачи – Скрипт.

**Скриптовый блок Добавление телефона**

udm:(ДОБАВИТЬ(Клиент.Мобильные\_телефоны, Мобильный\_телефон));

#### Скриптовый блок "Инициализация претензии"

Задача представляет собой скрипт, создающий новые поля в атрибуте «Претензия» контекстного типа.

Для этой задачи установлены следующие свойства:

* Имя – «Инициализация претензии».
* Тип задачи – Скрипт.

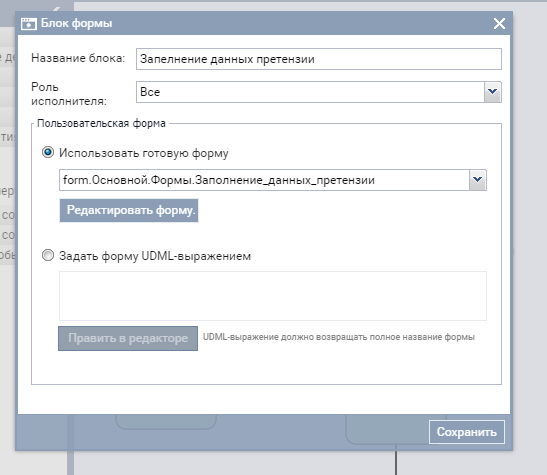
**Скриптовый блок Инициализация претензии**

udm:(Претензия := Основной:Претензия:Новый());

udm:(Претензия.Номер\_телефона := Номер);

#### Блок формы «Заполнение данных претензии».

Для этой формы установлены следующие свойства:

* Имя (string) – Заполнение данных претензии.
* Роль исполнителя (string) – Все (или возможно установить требующуюся нам роль, например Оператор).
* Форма (text) - выбрана форма «Заполнение\_данных\_претензии».  
  
* Начальное значение формы “Заполнение данных претензии”  
  

#### Скриптовый блок "Назначение исполнителя"

Представляет собой скрипт, получающий список исполнителей из контекстного типа и выбирает первого.

Для этой задачи установлены следующие свойства:

* Имя – «Назначение исполнителя».
* Тип задачи – Скрипт.

**Скриптовый блок Назначение исполнителя**

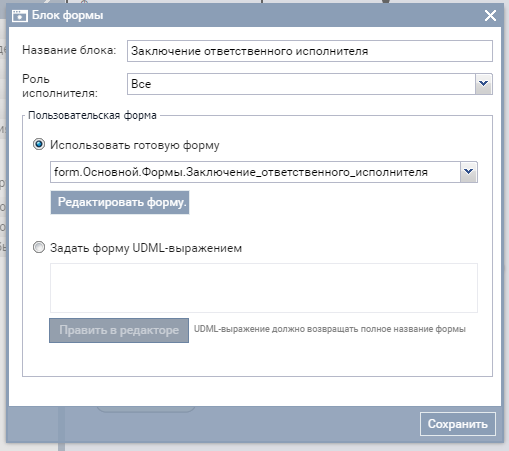
var процесс\_исполнитель = udm:(Основной:Процесс\_исполнитель:По\_идпроцесса(ТЕКУЩИЙПРОЦЕСС().идэкземпляра)[0]);

var ид\_исполнителя = процесс\_исполнитель.Ид\_исполнителя;

udm:(Претензия.Исполнитель := Основной:Исполнитель:поИд(js:string(ид\_исполнителя)));

#### Блок формы «Заключение ответственного исполнителя».

Для этой формы установлены следующие свойства:

* Имя (string) – Заключение ответственного исполнителя.
* Роль исполнителя (string) – Все (или возможно установить требующуюся нам роль, например Оператор).
* Форма (text) - выбрана форма «Заключение\_ответственного\_исполнителя».  
  
* Начальное значение формы “Заключение ответственного исполнителя”  
  

#### Скриптовый блок "Завершение"

Представляет собой скрипт, добавляющий в тип «Клиент» данные о звонке.

Для этой задачи установлены следующие свойства:

* Имя – «Завершение».
* Тип задачи – Скрипт.

**Скриптовый блок Завершение**

udm:(Звонок.Статус := да);

udm:(ДОБАВИТЬ(Клиент.Звонок, Звонок));

udm:(ДОБАВИТЬ(Клиент.Претензия, Претензия));

var phone\_number = udm:(ТЕКСТ(Номер)).substr(3).split(' ').join('');

if (udm:(Результат = "Печатная форма")){

var пф = udm:(Основной:Печатная\_форма\_претензии :Новый());

пф.ФИО\_клиента =udm:(ТЕКСТ(Клиент.Имя+' '+Клиент.Фамилия+' '+Клиент.Отчество));

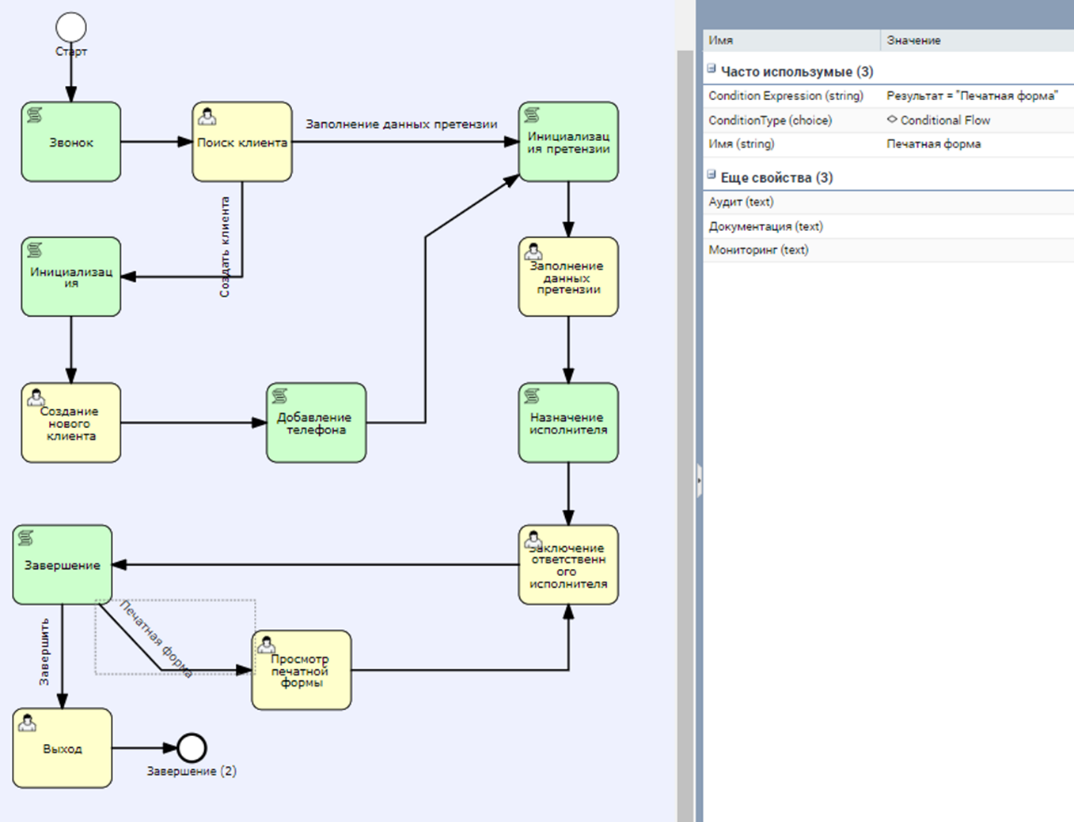
пф.Дата\_время = udm:(СЕЙЧАС());

пф.Претензия = udm:(ОТЧЕТ("Основной.\_ПоУмолчанию.Отчет\_по\_клиентам", Клиент, "PDF")); // P.S. На 10 строке скрипта путь до указанного шаблона скорее всего будет лежать по пути: "Основной.Шаблоны.Отчет\_по\_клиентам"

udm:(Печатная\_форма :=js:Основной:Печатная\_форма\_претензии(пф));

}

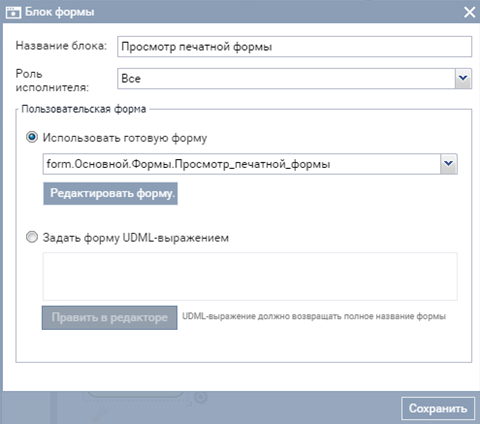
#### Условный переход к "Просмотр печатной формы"

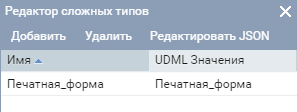
* Condition Expression (string) - **Результат = 'Печатная форма'**
* Condition Type - Conditional Flow
* Имя(string) – Печатная форма  
  

#### Блок формы «Просмотр печатной формы»

Для этой формы установлены следующие свойства:

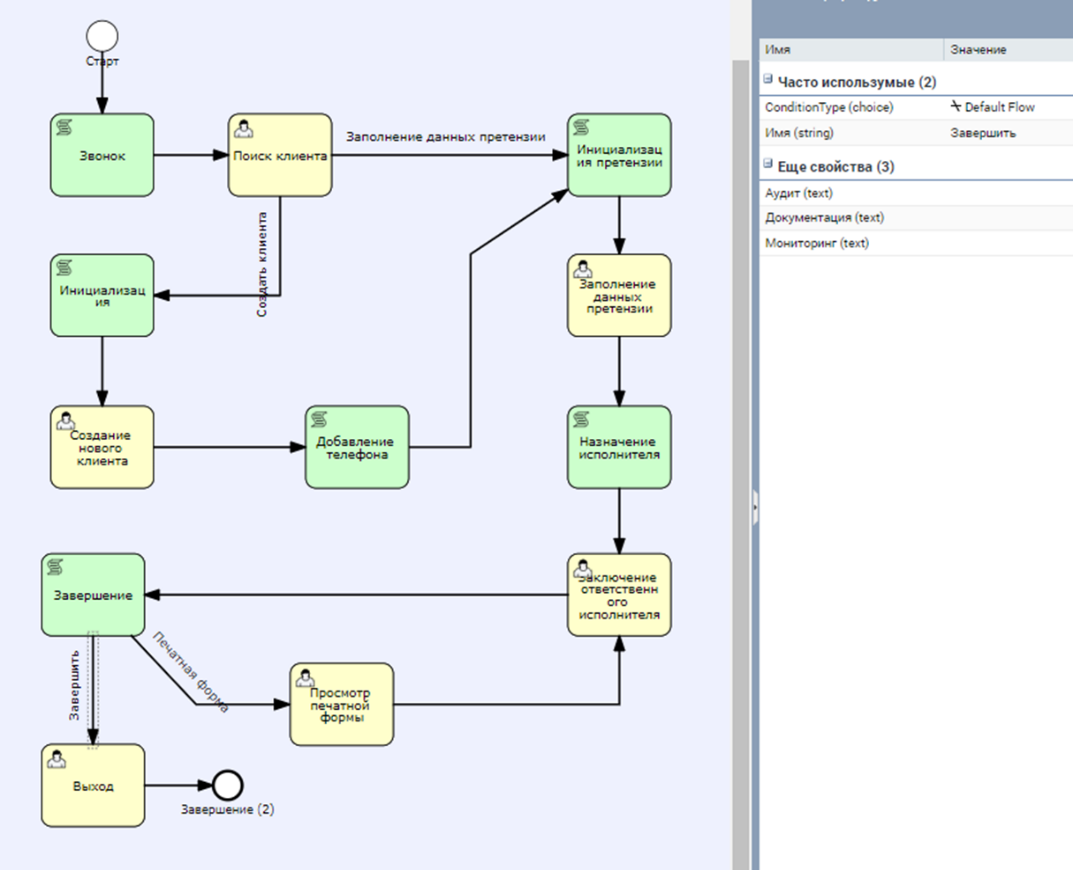
* Имя (string) – Просмотр печатной формы.
* Роль исполнителя (string) – Все (или возможно установить требующуюся нам роль, например Оператор).
* Форма (text) - выбрана форма «Просмотр\_формы\_претензии».



Начальное значение формы “Просмотр печатной формы”:  


#### Переход к форме «Выход».

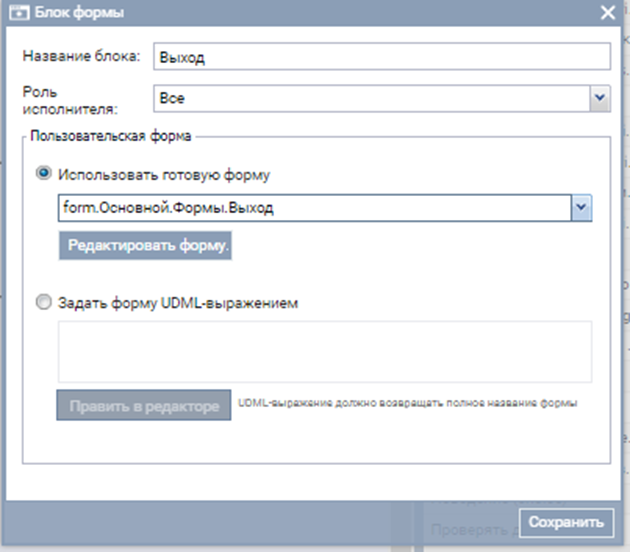
* Condition Type – Default Flow
* Имя – Завершить



#### Блок формы «Выход»

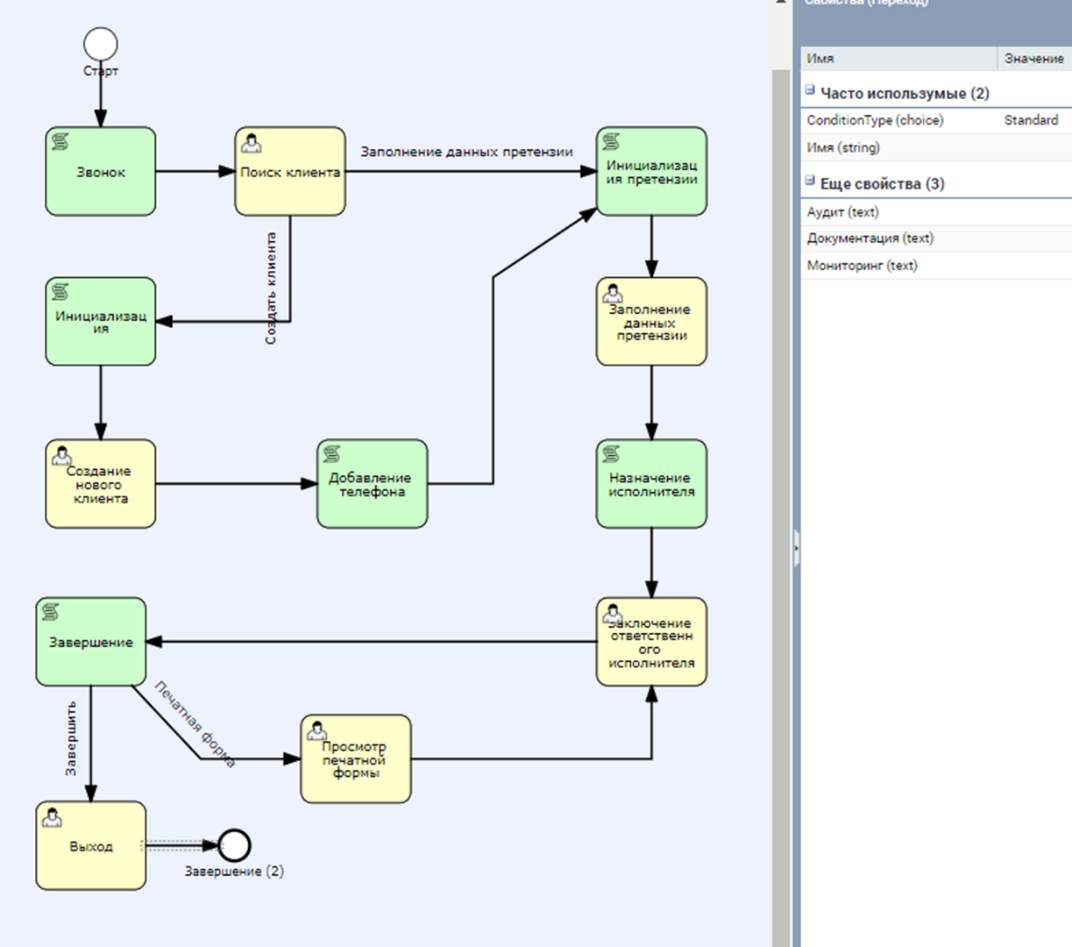
Для этой формы установлены следующие свойства:

* Имя (string) – Выход.
* Роль исполнителя (string) – Все (или возможно установить требующуюся нам роль, например Оператор).
* Форма (text) - выбрана форма «Выход».



#### Переход к блоку «Завершение».

* Condition Type - Standart
* Имя - Завершить



#### Завершающее событие «Завершение».

Как и любой конечный процесс, «Входящий звонок» заканчивается завершающим событием.

Используются все свойства по умолчанию.

# Наполнение форм

<https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/form_builder.html:0:ru>

Форма  - объект, используемый в процессе для взаимодействия с пользователями приложений. Фактически формы формируют интерфейс приложения.

В приложении будут использоваться несколько форм для разных процессов.

Для процесса Главный

1. **Форма\_оператора** – для отображения сгенерированных (в нашем случае) звонков.
2. **БП** – отображает формы процесса «входящий звонок», использует виджет «Окно процесса».
3. **Конец** – для вывода информации о завершении процесса
4. **Просмотр\_отчетов** – для просмотра отчетов

Для процесса Входящий звонок

1. **Поиск\_клиента** – для поиска клиентов в БД
2. **Создание\_нового\_клиента** – для заведения клиента в базу
3. **Заполнение\_данных\_претензии** – для заполнения данных претензии
4. **Заключение\_ответственного\_исполнителя** – для ввода заключения исполнителя.
5. **Просмотр\_печатной\_формы** – для просмотра печатной формы претензии

## Форма "Просмотр\_активного\_менеджера"

### Настройка атрибутов формы

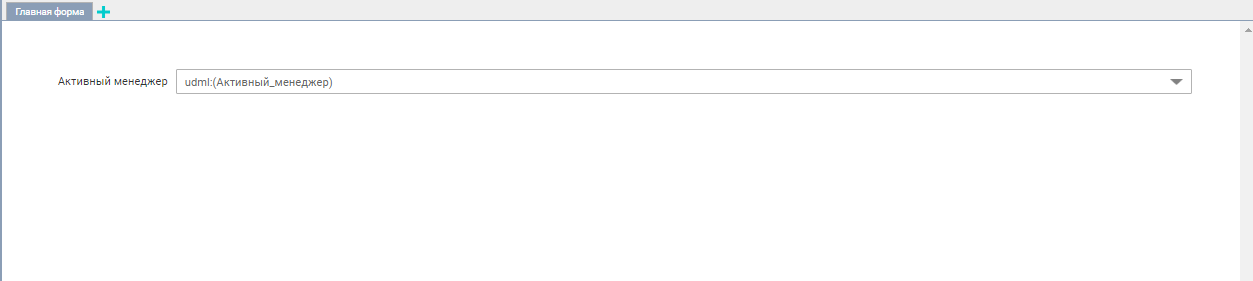
|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Активный\_менеджер | да | строка |

### Настройка элементов

Используются элементы:

* Раскрывающийся\_список\_Менеджер

Общий вид результата:



#### Элемент Раскрывающийся\_список\_Менеджер

|  |  |
| --- | --- |
| **Имя** | **Значение** |
| Надпись | Активный менеджер |
| Источник\_значений | Основной:Исполнитель:все() |
| Шаблон\_значения | строка.ФИО |
| Значение | Активный\_менеджер |
| Выбранное\_значение | строка.ФИО |
| Обязательное | нет |
| Надпись\_положение | слева |

## Форма "Форма\_оператора"

Используется в процессе «Главный» для отображения сгенерированных (в нашем случае) звонков.

### Настройка атрибутов формы

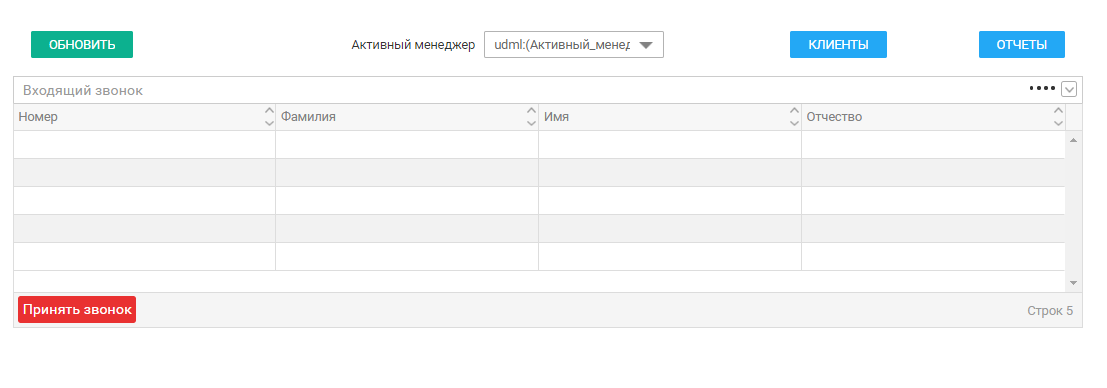
|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Результат | да | строка |
| Активная\_претензия | да | строка |

### Настройка элементов

Используются элементы:

* Таблица\_Входящие\_звонки;
* Кнопка\_Принять\_звонок (как действие внутри Таблица\_Входящие\_звонки);
* Кнопка\_Обновить
* Кнопка\_Клиенты
* Вложенная форма "Просмотр\_активного\_менеджера"
* Кнопка\_Отчеты.

Общий вид результата:



#### Элемент Таблица\_Входящие\_звонки

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Входящие звонки |
| Дополняемый | Нет |
| Тип\_объекта |  |
| Коллекция объектов | Система.Основной.Процессы.$процессы.Входящий\_звонок.$экземплярыпроцессов.$потоки.$экземплярформы |
| Колонки | |  |  | | --- | --- | | Номер | строка.$объект.Номер | | Фамилия | строка.$объект.Клиент.Фамилия | | Имя | строка.$объект.Клиент.Имя | | Отчество | строка.$объект.Клиент.Отчество | |
| Значение | Активная\_претензия |
| Выбранное\_значение | строка.идэкземплярапроцесса |
| Действия | |  |  | | --- | --- | | **Название действия** | **Цвет кнопки** | | Принять звонок | Красный | |

Все остальные свойства этого и следующих элементов оставляем по умолчанию (серым цветом).

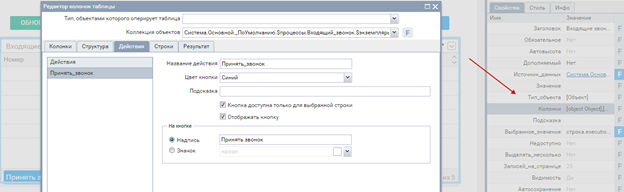
Некоторые пояснения:

* **Система.Основной.Процессы.$процессы.Входящий\_звонок.$экземплярыпроцессов.$потоки.$экземплярформы** – путь к объектам экземпляра процесса
* **строка.$объект** – объекты потока
* **строка.идэкземплярапроцесса** – id потока процесса

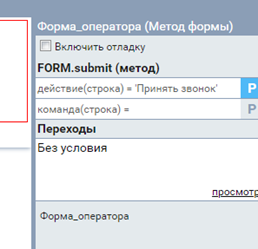
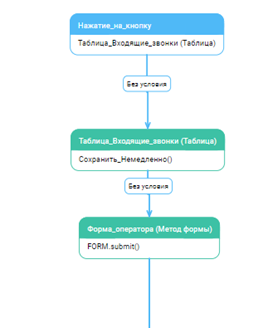
Таким образом, в таблице будут отображены значения  контекстных типов процесса «Входящий\_звонок».  
Чаще более правильным было бы «скормить» источнику данных некий атрибут формы и/или процесса, который в свою очередь настраивать при запуске процесса, но в данном случае нам нужно обновлять список входящих запросов «вживую», поэтому обращаемся напрямую к процессам.

#### Элемент Кнопка\_Принять\_звонок

Кнопка «Принять звонок» добавляется в закладке «Действия» свойства «Колонки».

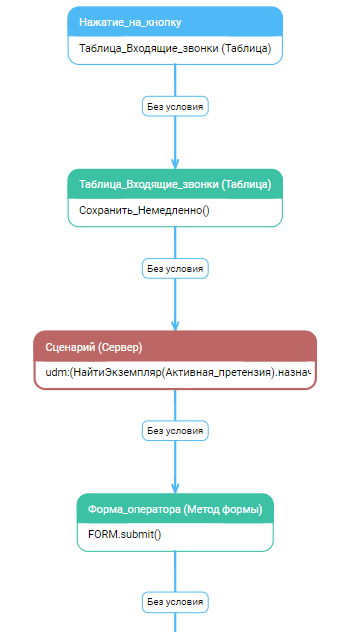


Для работы этой кнопки создадим событие для таблицы Нажатие на кнопку



Для реализации возможности перехвата звонков, созданных другим пользователем нужно добавить ещё один блок сценарий (сервер) в сценарий нажатия на кнопку "Принять звонок".

Дополнительно: пользователю, у которого перехватили заявку, необходимо показать уведомление о перехвате. Дальнейшую работу с формой, ее элементами данный пользователь продолжить не может (скрыть или сделать недоступной всю форму).



В сценарий-сервер нужно добавить следующий код:

**Сценарий (Сервер)**

udm:(НайтиЭкземпляр(Активная\_претензия).назначения.Все :=

НАЙТИПОЛЬЗОВАТЕЛЯ(ТЕКУЩИЙПОЛЬЗОВАТЕЛЬ()));

udm:(НайтиЭкземпляр(Активная\_претензия).$Текущийпользователь :=

НАЙТИПОЛЬЗОВАТЕЛЯ(ТЕКУЩИЙПОЛЬЗОВАТЕЛЬ()));

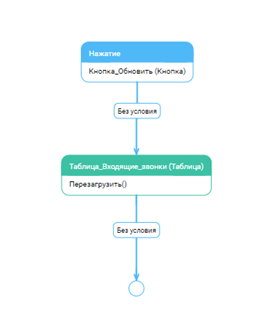
#### Элемент Кнопка\_Обновить

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Обновить |
| Закрыть\_форму | Нет |
| Сохранить\_форму | Нет |
| Стиль\_кнопки | дополнительная |

Все остальные свойства этого и следующих элементов оставляем по умолчанию (серым цветом).

К этой кнопке (ячейка Кнопка\_ Обновить) на событие нажатие нужно построить диаграмму

Данный событие обновляет элемент «Таблица», расположенный в ячейке Таблица\_Входящие\_звонки.



#### Элемент Кнопка\_Отчеты

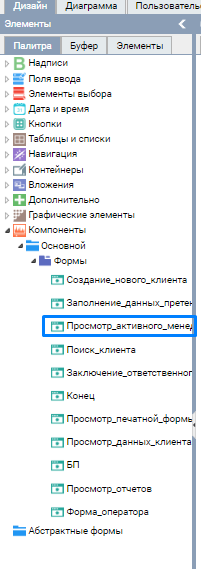
| **Имя** | **Значение** |
| --- | --- |
| Надпись | Отчеты |
| Закрыть\_форму | Применить и закрыть |
| Сохранить\_форму | Нет |
| Результат | Отчеты |
| Расположение | справа |

#### Элемент Кнопка\_Клиенты

|  |  |
| --- | --- |
| **Имя** | **Значение** |
| Надпись | Клиенты |
| Закрыть\_форму | Применить и закрыть |
| Сохранить\_форму | Нет |
| Результат | Клиенты |
| Расположение | справа |

#### ****Элемент вложенная форма «Просмотр\_активного\_менеджера»****

Для того, чтобы вставить вложенную форму, необходимо открыть раздел Компоненты, выбрать форму и перетянуть ее в нужное место.



Все остальные свойства этого и следующих элементов оставляем по умолчанию (серым цветом).

## Форма «БП»

Используется в процессе «**Главный**», использует виджет «Окно процесса», который является контейнером для форм процесса «входящий звонок».

В форме «БП» используются 3 элемента:

* Окно\_процесса\_1
* Кнопка\_Назад
* Кнопка\_Выход

Общий вид результата:



### Настройка атрибутов формы

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Активная\_претензия | Да | Строка |
| Результат | Да | Строка |

### Настройка элементов формы

#### Элемент "Окно\_процесса\_1"

Внутри элемента будут отображаться формы процесса «Входящий звонок».

|  |  |
| --- | --- |
| **Имя** | **Значение** |
| Фильтр | форма.идэкземплярапроцесса == Активная\_претензия |

#### Элемент "Кнопка\_Назад"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Назад |
| Закрыть\_форму | Применить и закрыть |
| Сохранить\_форму | Нет |
| Результат | Назад |

Все остальные свойства этого и следующих элементов оставляем по умолчанию (серым цветом).

#### Элемент "Кнопка\_Выход"

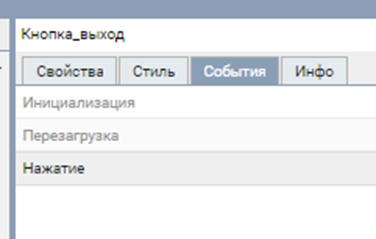
| **Имя** | **Значение** |
| --- | --- |
| Надпись | Выход |
| Закрыть\_форму | Отменить и закрыть |
| Сохранить\_форму | Нет |
| Расположение | справа |
| Результат | Выход |

Кнопка выполняет функцию закрытия процесса обработки звонка в случае если мы дошли до формы «Выход».

Поэтому нам не подходит свойство Закрыть\_форму = «Применить и закрыть», используемое для других кнопок.

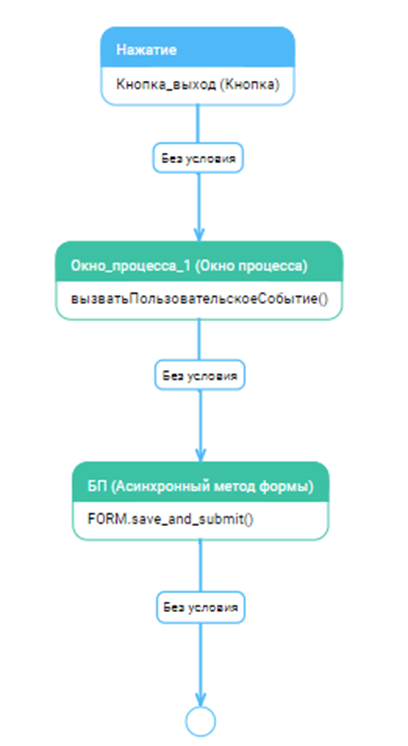
По кнопке «Кнопка\_выход» мы вызовем пользовательскую функцию и реализуем ее только на последней форме «Выход».

Для создания своего обработчика настроим Событие Нажатие.

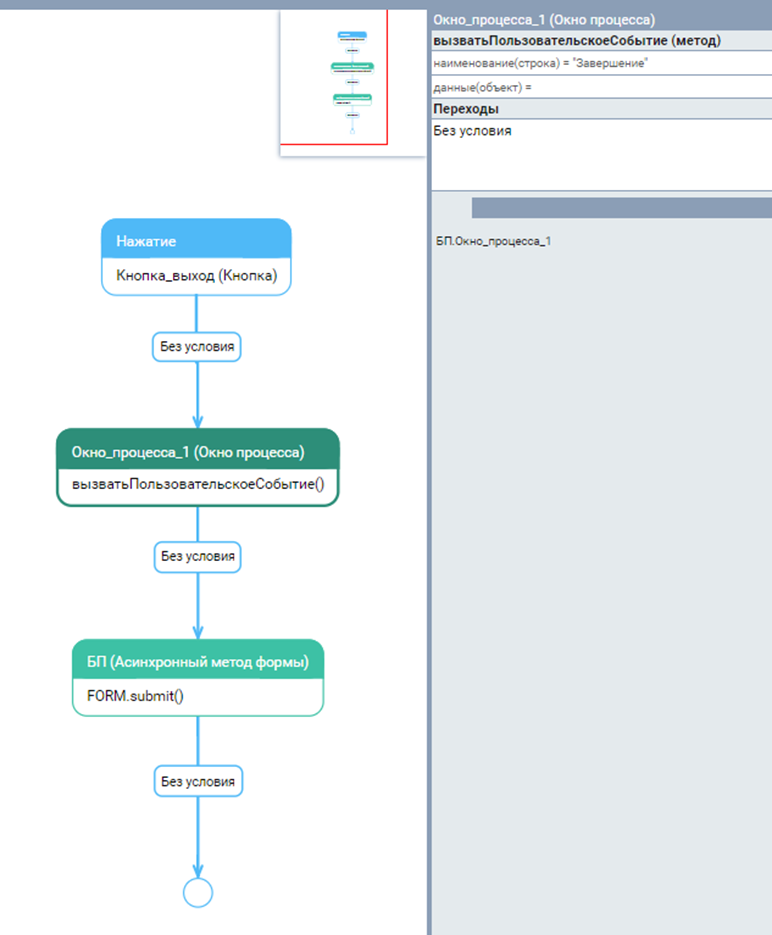


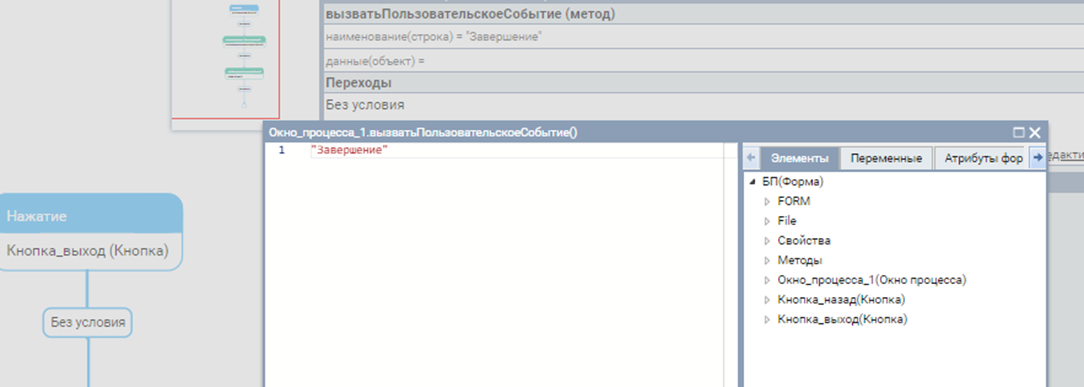
Двойной клик переведет нас в режим диаграммы, где мы настроим такую логику

1. Вызов пользовательского события в элементе Окно\_процесса\_1
2. Submit формы



Настройка вызова пользовательской функции. Параметр вызова наименование(строка) напишем «Завершение»

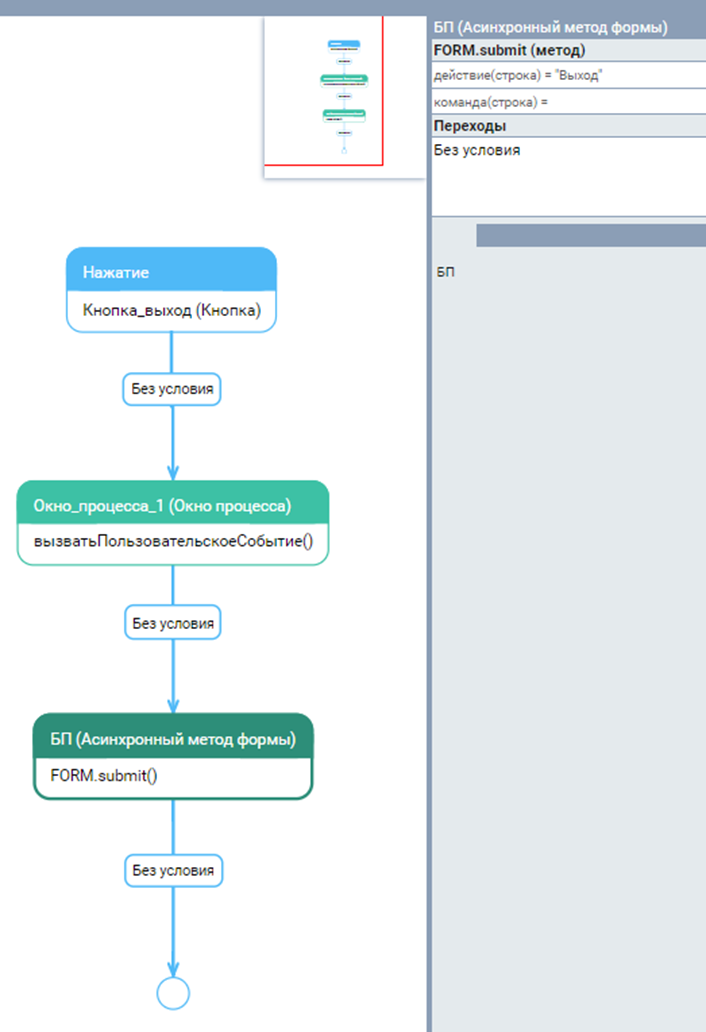




Другой пример настройки Пользовательского события можно посмотреть в документации

<https://doc.caseplatform.tech/#1.3.15:0:0:user_guide/form_builder/form_design/form_widgets/extras_form_widgets/placeholder_widget.html:0:ru>

Настройка Submit



Параметр действие(строка) настроим как «Выход».

В итоге, для всех случаев кнопка должна действовать также как кнопка Назад, но когда работа по звонку закончена по кнопке произойдет завершение процесса «входящий звонок», а потом переход на Форму оператора

## Форма «Просмотр\_отчетов»

Используется в главном процессе для просмотра любого отчета. На форме имеются следующие элементы:

* Таблица\_Отчетов (с кнопками/действиями «Сформировать отчет» и «Скачать»)
* Кнопка\_Назад
* Всплывающее\_окно\_Просмотр\_отчета

Общий вид результата:



### Настройка атрибутов формы

| **Атрибут** | **Входной** | **Тип** |
| --- | --- | --- |
| Отчет | Нет | Основной.Печатная\_форма\_претензии |
| Результат | да | Строка |

### Настройка элементов формы

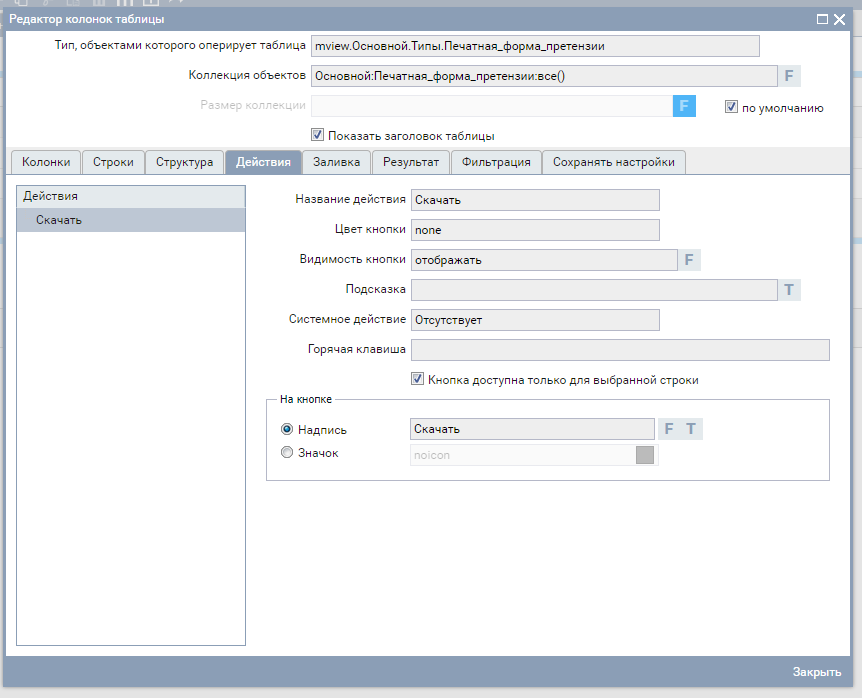
#### Элемент "Таблица\_Отчетов"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Отчеты |
| Дополняемый | Нет |
| Тип, объектами которого оперирует таблица | mview.Основной.Типы.Печатная\_форма\_претензии |
| Коллекция объектов | Основной:Печатная\_форма\_претензии:все() |
| Колонки | |  |  |  | | --- | --- | --- | | **Имя атрибута** | **Значение** | **Тип** | | ФИО | ФИО\_клиента | Строка | | Дата | Дата\_время | Дата\_время | |
| Действие | Скачать |
| Выбранное значение | Строка |
| Авто сохранение | Да |
| Значение | Отчет  Примечание: важно, чтобы в свойствах был явно выбран Атрибут формы. Добавляется в F  fe4ac7f8151eda0bdaa9098798c8d41f |

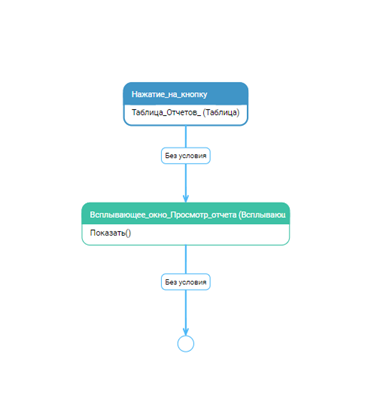
Колонки настраиваются в отдельном окне “Редактор колонок таблицы”:



Кнопки таблицы настраиваются в закладке «Действия» данного окна:



Для работы кнопки таблицы отчетов нам понадобится назначить событие **Нажатие\_на\_кнопку**

****

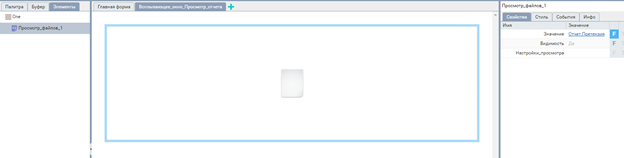
#### Элемент "Кнопка\_Назад"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Назад |
| Закрыть\_форму | Отменить и закрыть |
| Сохранить\_форму | Нет |
| Результат | Назад |

#### Элемент "Всплывающее\_окно\_Просмотр\_отчета"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Просмотр отчета |
| Видимость | Нет |
| Modal | Нет |

В само Окно мы добавляем элемент просмотр файлов



Атрибуты:

| **Имя** | **Значение** |
| --- | --- |
| Значение | Отчет.Претензия |

## Форма «Поиск\_клиента»

Поиск клиента – используется в процессе «входящий звонок» для поиска клиентов в БД. Форма содержит следующие элементы:

* Поле\_ввода\_Определенный\_номер
* Поле\_ввода\_Фамилия
* Поле\_ввода\_Имя
* Поле\_ввода\_Отчество
* Элемент Дата\_рождения
* Кнопка\_Создать\_клиента
* Кнопка\_Далее

Общий вид результата:



### Настройка атрибутов формы

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Клиент | да | Основной.Клиент |
| Номер | да | строка |
| Результат | да | строка |

### Настройка элементов формы

#### Элемент "Поле\_ввода\_Определенный\_номер"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Определенный номер |
| Значение | Номер |

#### Элемент "Поле\_ввода\_Фамилия"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Фамилия |
| Значение | Клиент.Фамилия |
| Не\_редактируется | Да |

#### Элемент "Поле\_ввода\_Имя"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Имя |
| Значение | Клиент.Имя |
| Не\_редактируется | Да |

#### Элемент "Поле\_ввода\_Отчество"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Отчество |
| Значение | Клиент.Отчество |
| Не\_редактируется | Да |

#### Элемент "Дата\_Рождения"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Дата рождения |
| Значение | Клиент.Дата\_рождения |
| Не\_редактируется | Да |

#### Элемент "Кнопка\_Создать\_клиента"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Создать клиента |
| Закрыть\_форму | Применить и закрыть |
| Результат | NewClient |
| Сохранить\_форму | Нет |
| Недоступно | **текст**(Клиент.Имя+Клиент.Фамилия+Клиент.Отчество) != '' |

#### Элемент "Кнопка\_Далее"

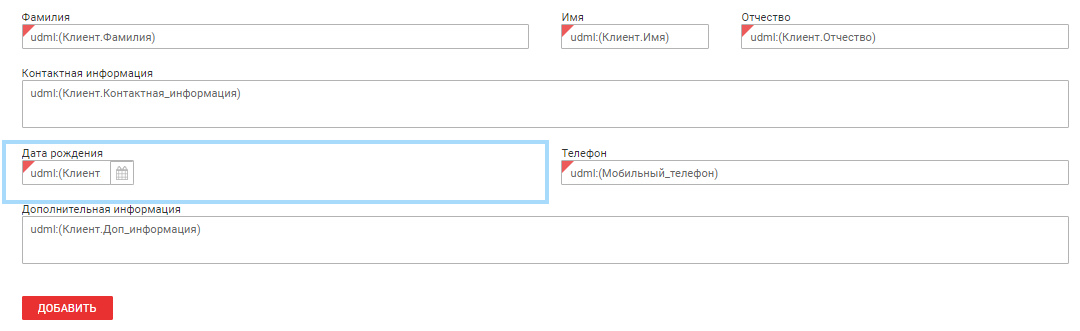
| **Имя** | **Значение** |
| --- | --- |
| Надпись | Далее |
| Закрыть\_форму | Применить и закрыть |
| Результат | Заполнение данных претензии |
| Расположение | справа |
| Недоступно | **текст**(Клиент.Имя+Клиент.Фамилия+Клиент.Отчество) == '' |

## Форма «Создание\_нового\_клиента»

Создание нового клиента – используется в процессе «входящий звонок»  для заведения клиента. Форма содержит следующие элементы:

* Поле\_ввода\_Фамилия
* Поле\_ввода\_Имя
* Поле\_ввода\_Отчество
* Многострочное\_поле\_ввода\_Контактная\_информация
* Поле\_ввода\_Телефон
* Элемент Дата\_рождения
* Многострочное\_поле\_ввода\_Дополнительная\_информация
* Кнопка\_Добавить

Общий вид результата



### Настройка атрибутов формы

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Клиент | да | Основной.Клиент |
| Мобильный\_телефон | да | строка |

### Настройка элементов формы

#### Элемент "Поле\_ввода\_Фамилия"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Фамилия |
| Значение | Клиент.Фамилия |
| Обязательное | Да |

#### Элемент "Поле\_ввода\_Имя"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Имя |
| Значение | Клиент.Имя |
| Обязательное | Да |

#### Элемент "Поле\_ввода\_Отчество"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Отчество |
| Значение | Клиент.Отчество |

#### Элемент "Многострочное\_поле\_ввода\_Контактная\_информация"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Контактная информация |
| Значение | Клиент.Контактная\_информация |

#### Элемент "Поле\_ввода\_Телефон"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Телефон |
| Значение | Мобильный\_телефон.Номер |
| Обязательное | Да |

#### Элемент Дата\_Рождения

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Дата рождения |
| Значение | Клиент.Дата\_рождения |
| Обязательное | Да |

#### Элемент "Многострочное\_поле\_ввода\_Дополнительная\_информация"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Дополнительная информация |
| Значение | Клиент.Доп\_информация |

#### Элемент "Кнопка\_Добавить"

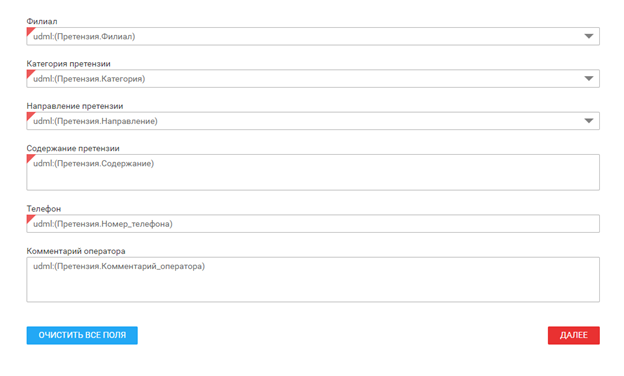
| **Имя** | **Значение** |
| --- | --- |
| Надпись | Добавить |
| Закрыть\_форму | Применить и закрыть |

## Форма «Заполнение\_данных\_претензии»

Заполнение данных претензии – используется в процессе «входящий звонок» для заполнения данных претензии. Форма содержит следующие элементы:

* Раскрывающийся\_список\_Филиал
* Раскрывающийся\_список\_Категория\_претензии
* Раскрывающийся\_список\_Направление\_претензии
* Многострочное\_поле\_ввода\_Содержание\_претензии
* Поле\_ввода\_Телефон
* Многострочное\_поле\_ввода\_Комментарий\_оператора
* Кнопка\_Очистить\_все\_поля
* Кнопка\_Далее

Общий вид результата:



### Настройка атрибутов формы

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Претензия | да | Основной.Претензия |

### Настройка элементов формы

#### Элемент "Раскрывающийся\_список\_Филиал"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Филиал |
| Обязательное | Да |
| Источник\_значений | Основной:Филиал:все() |
| Шаблон\_значения | строка.Название |
| Выбранное\_значение | строка |
| Значение | Претензия.Филиал |

#### Элемент "Раскрывающийся\_список\_Категория\_претензии"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Категория претензии |
| Источник\_значений | Основной:Категория\_претензии:все() |
| Значение | Претензия.Категория |
| Обязательное | Да |

#### Элемент "Раскрывающийся\_список\_Направление\_претензии"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Направление претензии |
| Источник\_данных | Основной:Направление\_претензии:все() |
| Значение | Претензия.Направление |
| Обязательное | Да |

#### Элемент "Многострочное\_поле\_ввода\_Содержание\_претензии"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Содержание претензии |
| Значение | Претензия.Содержание |
| Обязательное | Да |

#### Элемент "Поле\_ввода\_Телефон"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Телефон |
| Обязательное | Да |
| Значение | Мобильный\_телефон.Номер |

#### Элемент "Многострочное\_поле\_ввода\_Комментарий\_оператора"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Дополнительная информация |
| Значение | Клиент.Доп\_информация |

#### Элемент "Кнопка\_Очистить\_все\_поля"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Добавить |
| Закрыть\_форму | Нет |
| Сохранить\_форму | Нет |
| Стиль\_кнопки | отменяющая |

К этой кнопке на событие Нажатие создать диаграмму. После нажатия на эту кнопку будут очищены поля «Содержание претензии», «Телефон», «Комментарий оператора».



**Код сценария**

Многострочное\_поле\_ввода\_Содержание\_претензии.Значение = '';

Поле\_ввода\_Телефон.Значение = '';

Многострочное\_поле\_ввода\_Комментарий\_оператора.Значение = '';

#### Элемент <Кнопка\_Далее>

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Далее |
| Закрыть\_форму | Применить и закрыть |
| Расположение | справа |

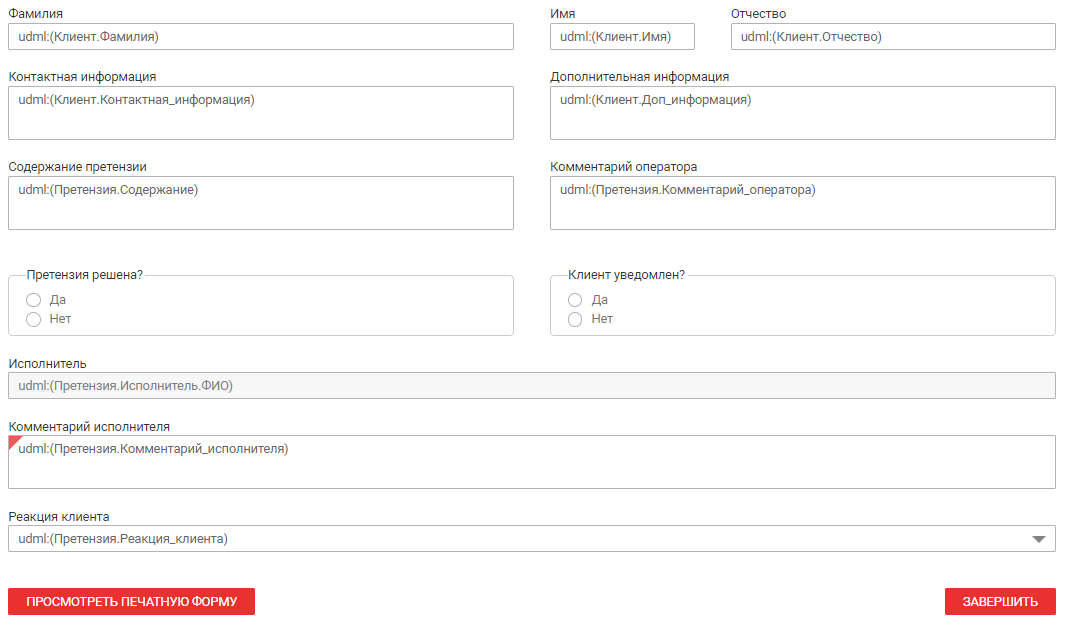
Проверка на обязательность полей при уходе с формы регулируется свойством Кнопки Проверять\_значения должно быть «Да»

## Форма «Заключение\_ответственного\_исполнителя»

Заключение ответственного исполнителя – используется в процессе «входящий звонок», для ввода заключения исполнителя. Форма содержит следующие элементы:

* Поле\_ввода\_Фамилия
* Поле\_ввода\_Имя
* Поле\_ввода\_Отчество
* Многострочное\_поле\_ввода\_Контактная\_информация
* Многострочное\_поле\_ввода\_Дополнительная\_информация
* Многострочное\_поле\_ввода\_Содержание\_претензии
* Многострочное\_поле\_ввода\_Комментарий\_оператора
* Поле\_ввода\_Исполнитель
* Группа\_переключателей\_Претензия\_решена
* Группа\_переключателей\_Клиент\_уведомлен
* Многострочное\_поле\_ввода\_Комментарий\_исполнителя
* Раскрывающийся\_список\_Реакция\_клиента
* Кнопка\_Просмотреть\_печатную\_форму
* Кнопка\_Завершить

Общий вид результата:



### Настройка атрибутов формы

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Клиент | да | Основной.Клиент |
| Претензия | да | Основной.Претензия |
| Результат | да | строка |

### Настройка элементов формы

#### Элемент "Поле\_ввода\_Фамилия"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Фамилия |
| Значение | Клиент.Фамилия |

#### Элемент "Поле\_ввода\_Имя"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Имя |
| Значение | Клиент.Имя |

#### Элемент "Поле\_ввода\_Отчество"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Отчество |
| Значение | Клиент.Отчество |

#### Элемент "Многострочное\_поле\_ввода\_Контактная\_информация"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Контактная информация |
| Значение | Клиент.Контактная\_информация |

#### Элемент "Многострочное\_поле\_ввода\_Дополнительная\_информация"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Дополнительная информация |
| Значение | Клиент.Доп\_информация |

#### Элемент "Многострочное\_поле\_ввода\_Содержание\_претензии"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Содержание претензии |
| Значение | Претензия.Содержание |

#### Элемент "Многострочное\_поле\_ввода\_Комментарий\_оператора"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Комментарий оператора |
| Значение | Претензия.Комментарий\_оператора |

#### Элемент "Поле\_ввода\_Исполнитель"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Исполнитель |
| Значение | Претензия.Исполнитель.ФИО |
| Не\_редактируется | Да |

#### Элемент "Группа\_переключателей\_Претензия\_решена"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Претензия решена? |
| Значение | Претензия.Решение |
| Обязательное | Да |
| Источник\_данных | Основной:Логический\_справочник:все() |
| Шаблон\_значения | строка.Название |
| Выбранное\_значение | строка.Значение |

#### Элемент "Группа\_переключателей\_Клиент\_уведомлен"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Клиент уведомлен? |
| Значение | Претензия.Клиент\_уведомлен |
| Источник\_данных | Основной:Логический\_справочник:все() |
| Шаблон\_значения | строка.Название |
| Выбранное\_значение | строка.Значение |
| Обязательное | Нет |

#### Элемент "Многострочное\_поле\_ввода\_Комментарий\_исполнителя"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Комментарий исполнителя |
| Значение | Претензия.Комментарий\_исполнителя |
| Обязательное | Да |

#### Элемент "Раскрывающийся\_список\_Реакция\_клиента"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Реакция клиента |
| Источник\_данных | Основной:Реакция\_клиента:все() |
| Значение | Претензия.Реакция\_клиента |

#### Элемент "Кнопка\_Просмотреть\_печатную\_форму"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Просмотреть печатную форму |
| Закрыть\_форму | Применить и закрыть |
| Результат | Печатная форма |

#### Элемент "Кнопка\_Завершить"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Завершить |
| Закрыть\_форму | Применить и закрыть |
| Расположение | справа |
| Результат | Завершить |

Проверка на обязательность полей при уходе с формы регулируется свойством Кнопки Проверять\_значения должно быть «Да»

## Форма "Просмотр\_данных\_клиента"

На форме используются элементы:

* Заголовок\_второго\_уровня
* Поле\_ввода\_Фамилия
* Поле\_ввода\_Имя
* Поле\_ввода\_Отчество
* Дата\_Рождения
* Кнопка сохранить

Общий вид результата:



### Настройка атрибутов формы

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Фамилия | Нет | Строка |
| Имя | Нет | Строка |
| Отчество | Нет | Строка |
| Дата\_рождения | Нет | Строка |
| ид | Нет | Строка |

### Настройка элементов формы

#### Элемент "Заголовок\_второго\_уровня"

|  |  |
| --- | --- |
| **Имя** | **Значение** |
| Надпись | Персональные данные клиента |

#### Элемент "Поле\_ввода\_Фамилия"

|  |  |
| --- | --- |
| **Имя** | **Значение** |
| Надпись | Фамилия |
| Значение | Фамилия (атрибут формы) |

#### Элемент "Поле\_ввода\_Имя"

|  |  |
| --- | --- |
| **Имя** | **Значение** |
| Надпись | Имя |
| Значение | Имя (атрибут формы) |

#### Элемент "Поле\_ввода\_Отчество"

|  |  |
| --- | --- |
| **Имя** | **Значение** |
| Надпись | Отчество |
| Значение | Отчество (атрибут формы) |

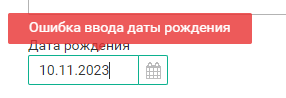
#### Элемент "Дата\_Рождения"

|  |  |
| --- | --- |
| **Имя** | **Значение** |
| Надпись | Дата рождения |
| Значение | Дата\_рождения (атрибут формы) |

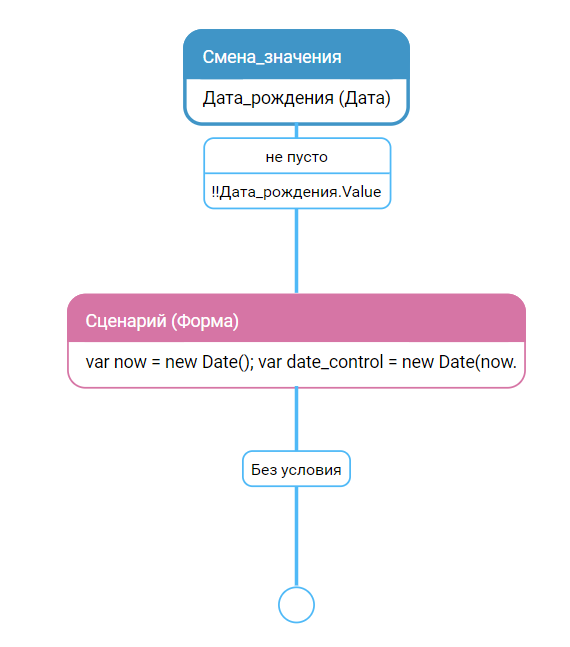
#### Элемент "Кнопка\_Сохранить"

|  |  |
| --- | --- |
| **Имя** | **Значение** |
| Надпись | Сохранить |
| Сохранить\_форму | Да |
| Закрыть\_форму | Применить без закрытия |

Добавим проверку на возраст. Если при смене даты рождения клиенту окажется меньше 18 лет, должно всплыть предупреждающее окно.



Для этого нужно выбрать поле Дата\_Рождения, перейти во вкладку События и выбрать Смена\_значения.



1. Добавим условие перехода, что поле с датой рождения не пустое

**Переход**

!!Дата\_Рождения.Value

      2.  Добавим сценарий (форма) со следующим скриптом:

**Сценарий (форма)**

var now = new Date();

var date\_control = new Date(now.getFullYear() - 18, now.getMonth(), now.getDate());

var db = Дата\_Рождения.Value;

db = new Date(db.getFullYear(), db.getMonth(), db.getDate());

if (db > date\_control) {

Дата\_Рождения.show\_balloon('Клиенту должно быть больше 18 лет', 'error', true);

} else {

Дата\_Рождения.hide\_balloon('Клиенту должно быть больше 18 лет', 'error');

}

## Форма "Просмотр\_клиентов"

Используется в Главном процессе для просмотра всех клиентов, которые когда-либо звонили, и для редактирования их персональных данных.

На форме используются элементы:

* Заголовок\_Список клиентов
* Таблица\_Клиенты
* Панель\_1
* Вложенная форма "Просмотр данных клиента"

Общий вид результата:



### Настройка атрибутов формы

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Активный\_клиент | Нет | Основной.Клиент |

### Настройка элементов формы

#### Элемент Заголовок\_Список\_клиентов

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Список клиентов |

#### Элемент Таблица\_Клиенты

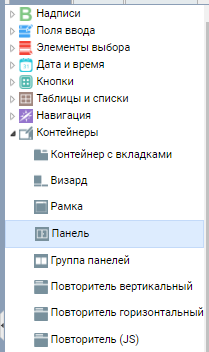
| **Имя** | **Значение** |
| --- | --- |
| Надпись | Список клиентов |
| Заголовок | Клиенты |
| Автовысота | Нет |
| Значение | Активный\_клиент |
| Колонки | | **Имя атрибута** | **Значение** | **Тип** | | --- | --- | --- | | Фамилия | Фамилия | Строка | | Имя | Имя | Строка | | Отчество | Отчество | Строка | | Дата\_рождения | Дата\_рождения (формат вывода - [dd.mm](http://dd.mm).yyyy) | Дата | |
| Выбранное\_значение | строка |

#### Элемент "Кнопка\_Назад"

| **Имя** | **Значение** |
| --- | --- |
| Надпись | Назад |
| Сохранить форму | Нет |
| Закрыть форму | Отменить и закрыть |

#### Элемент "Панель\_1"

Панель находится в разделе Контейнеры.



|  |  |
| --- | --- |
| **Имя** | **Значение** |
| Заголовок | Панель клиента |
| Сворачиваемая | Да |
| Свернута | Нет |
| Выдвижная панель | ab956b84b40f925530f667aa13e9e927 |

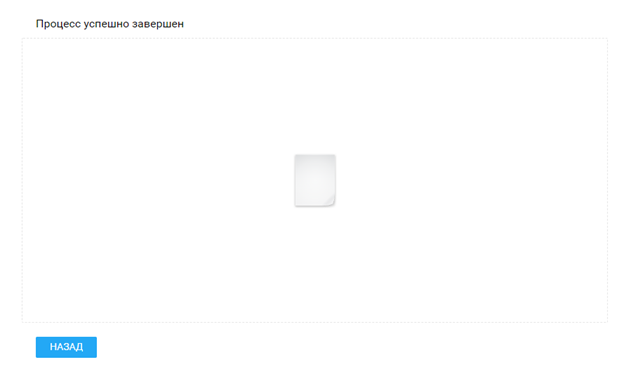
Вложенную форму "Просмотр\_данных\_клиента" необходимо вставить внутрь панели. Для этого 2 раза щелкните по панели, откройте раздел "Компоненты", выберите нужную форму и перенесите ее на панель.

## Форма «Просмотр\_печатной\_формы»

Используется для просмотра печатной формы претензии. Форма содержит следующие поля:

* Заголовок\_второго\_уровня\_1
* Просмотр\_PDF\_документа\_1
* Кнопка\_Назад

Общий вид результата



### Настройка атрибутов формы

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Печатная\_форма | да | Основной.Печатная\_форма\_претензии |

### Настройка элементов формы

#### Элемент "Заголовок\_второго\_уровня\_1"

|  |  |
| --- | --- |
| Имя | Значение |
| Надпись | Процесс успешно завершен |
| Видимость | ЕНД(Печатная\_форма) |

#### Элемент "Просмотр\_PDF\_документа\_1"

|  |  |
| --- | --- |
| Имя | Значение |
| Значение | Печатная\_форма.Претензия |
| Видимость | Да |

#### Элемент "Кнопка\_Назад"

| Имя | Значение |
| --- | --- |
| Надпись | Назад |
| Закрыть\_форму | Отменить и закрыть |
| Сохранить\_форму | Нет |
| Результат | Назад |

## Форма «Выход»

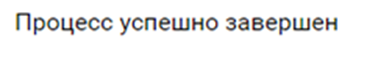
Используется в процессе «**входящий звонок**» для вывода информации о завершении процесса.

Атрибуты

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Входной** | **Тип** |
| Результат | нет | строка |

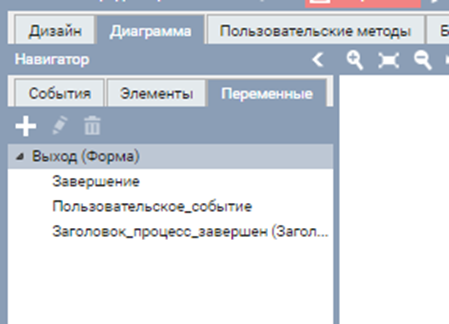
Форма содержит следующие элемент Заголовок\_второго\_уровня\_1;

Общий вид:

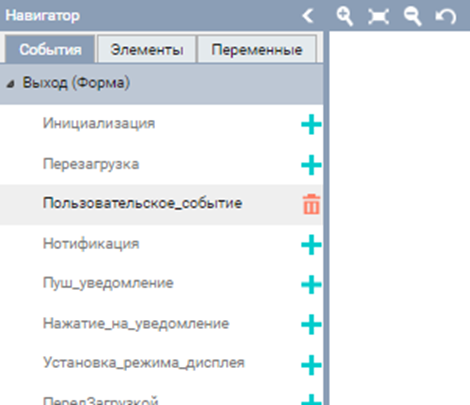


Перейдем в Диаграммы и настроим Пользовательское событие

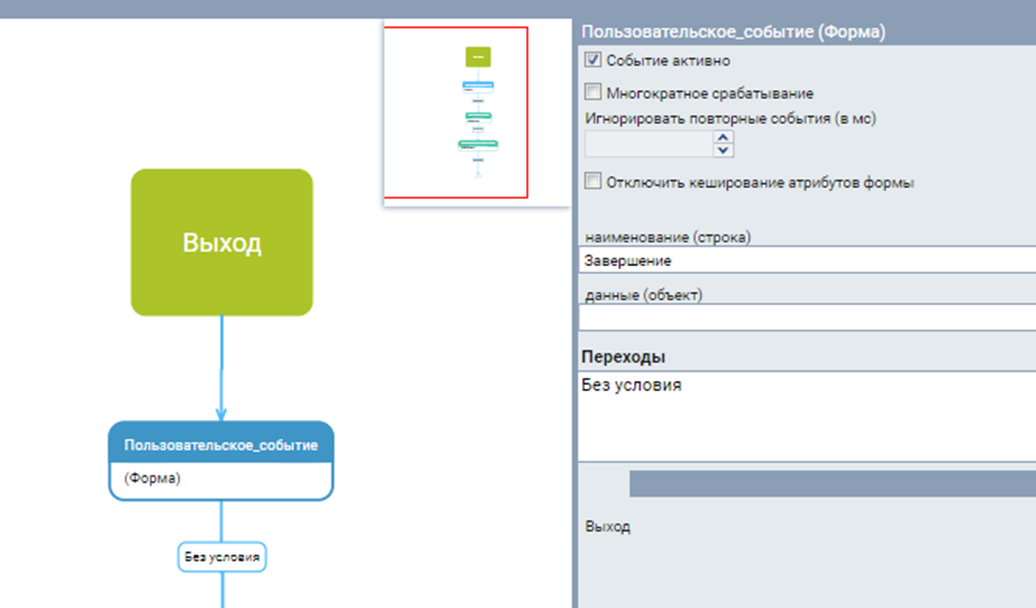
1. Сначала создадим переменную Завершение



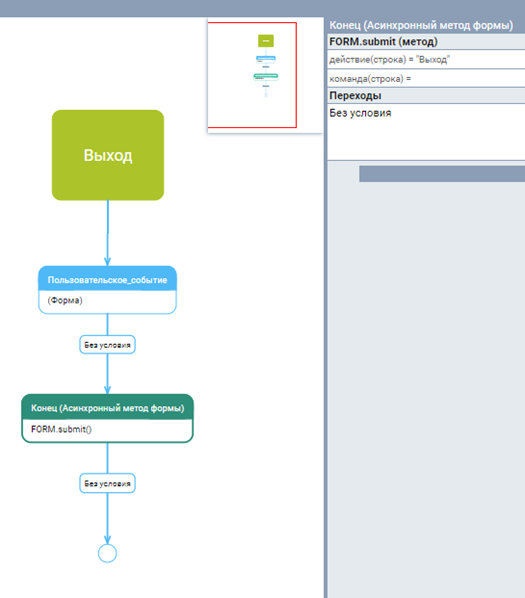
2. Создадим пользовательское событие формы



3.  Свойства события: наименование (строка) – выбираем Завершение, остальные свойства стандартные



4. Добавим элемент Submit настроим его действие(строка) = «Выход»:



Таким образом, нажимая на форме БП кнопку «Выход» происходит попытка вызвать пользовательское событие.

Если оно есть, а оно есть только на Форме «Выход», то происходит закрытие Формы «Выход». Тем самым завершается экземпляр «входящий звонок» для обработанного звонка.

# Политики запуска

Бизнес-процессы по объектам может запускаться различными способами:

* автозапуск
* запуск при создании сессии пользователя

Запуск по политикам:

* запуск при входе в класс
* перезапуск по завершению
* а также, завершение при входе

Запуск процесса вручную доступен только пользователям с соответствующей ролью.

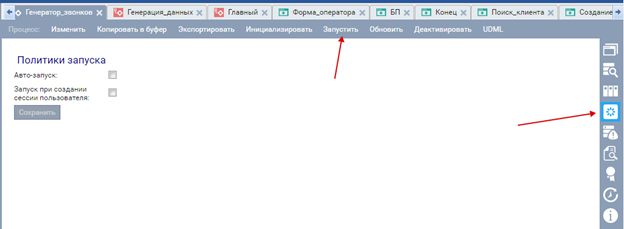
Запуск процесса автоматически производится согласно настроенным политикам: запускать при входе в класс, перезапускать завершившийся процесс до выхода из класса, завершать при выходе из класса.

## Процесс "Генерация\_данных"

Запускается вручную.

## Процесс "Генератор\_звонков"

Запускается вручную.



## Процесс "Главный"

Запускается при инициализации сессии пользователя.



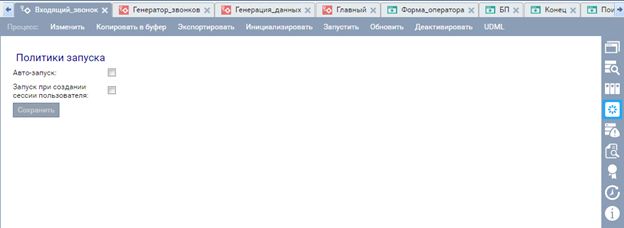
## Процесс "Входящий\_звонок"

Запускается в процессе «Генератор звонков». Запускается методом «**ЗАПУСТИТЬПРОЦЕСС**».

**Запуск процесса Входящий\_звонок внутри процесса Генератор\_звонков**

**Запуск процесса Входящий\_звонок внутри процесса Генератор\_звонков**

udm:(ЗАПУСТИТЬПРОЦЕСС("Основной.Процессы.Входящий\_звонок",["Номер"], Телефон));



# Создание отчетов

## Отчет по клиентам

Шаблоны печатных форм в документации: <https://doc.caseplatform.tech/#1.3.13:0:0:user_guide/report_builder.html:0:ru>

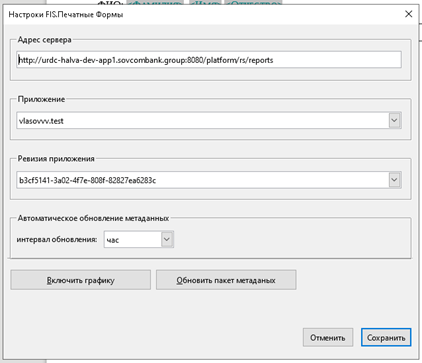
Для начала создания отчета заходим в LibreOffice и создаем новый текстовый документ, называем Отчет.

Далее устанавливаем необходимые плагины для работы с FIS Platform.

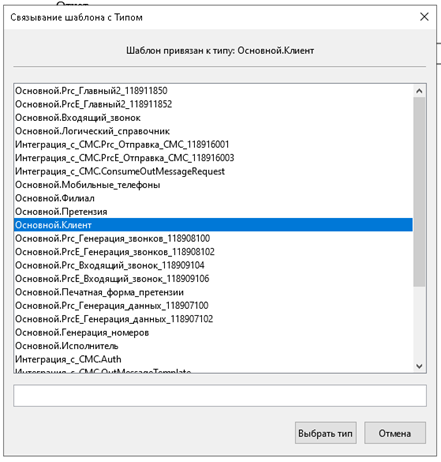
Для этого заходим в Сервисы –> Управление расширениями -> Добавить

Выполняем настройку плагина 9fb798897fee26dff5bc88ed4689ca30

В поле приложение выбрать свое приложение:

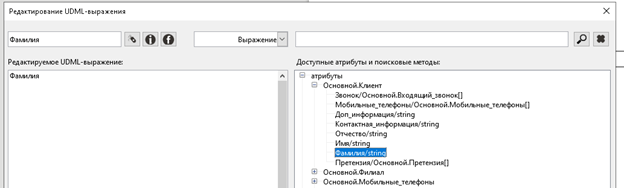


Связываем шаблон с типом 57797a95c7002ccbd6ba3982e486f787

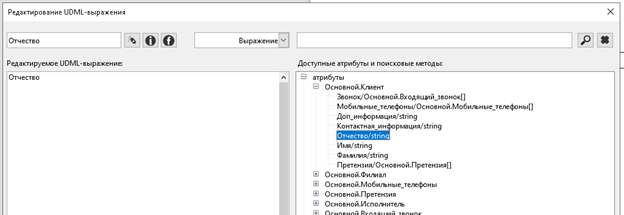


Вставляем необходимые поля 8a8e7d5e38a4f849aed2032856473a71



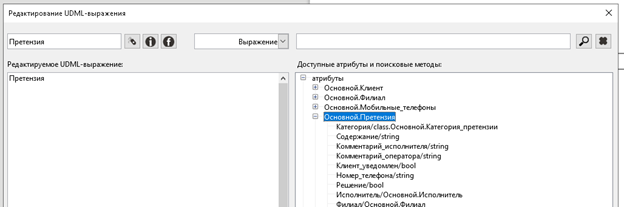


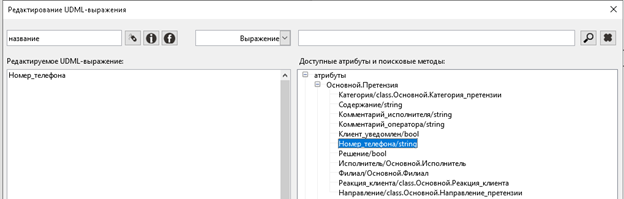




Создаем таблицу и в нее вставляем необходимые значения:

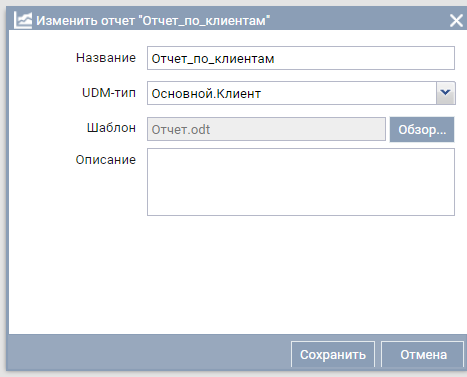
49d946273e1beb0851f3ebeeb6ad9177





Далее в FIS создаем e05565bfa50863699668ae4d12110726**Отчет\_по\_клиентам**

Как шаблон вставляем созданный файл:



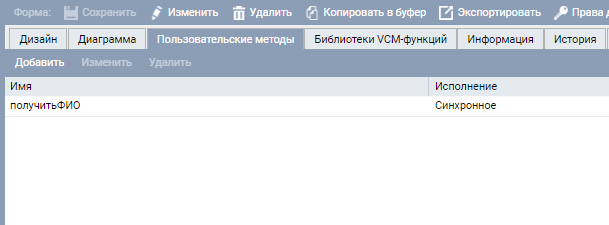
# Настройка событий

## Настройка связи форм «Форма оператора» и «Просмотр активного менеджера»

### Форма "Просмотр\_активного\_менеджера"

Чтобы принять звонок, необходимо для начала выбрать активного менеджера. Раскрывающийся список со значениями активного менеджера находится внутри дочерней формы, поэтому для того, чтобы сохранить и использовать в дальнейшем ФИО выбранного менеджера, необходимо настроить передачу значения с дочерней формы на главную. Для этого создаем пользовательский метод на форме «Просмотр активного менеджера»:

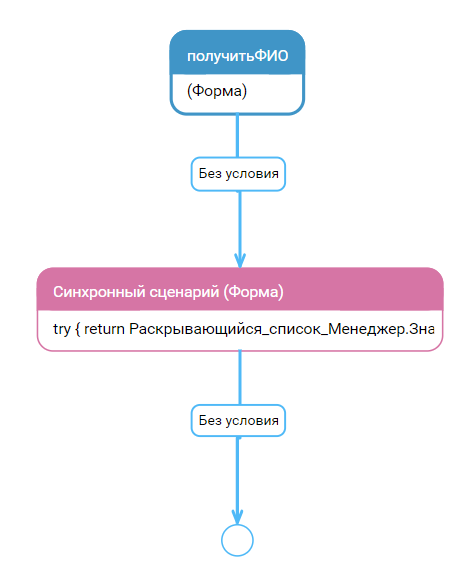
1. На форме переходим во вкладку "Пользовательские методы"



       2. Нажимаем кнопку добавить и задаем следующее

|  |  |
| --- | --- |
| Название | получитьФИО |
| Синхронное исполнение | Да |

Общий вид пользовательского метода:



В синхронный сценарий (форма) добавляем следующий скрипт, который будет возвращать значение раскрывающегося списка:

try {

    return Раскрывающийся\_список\_Менеджер.Значение;

} catch(ex) {

    return ex;

}

Чтобы при переключении между формами выбранный менеджер сохранялся на форме «Просмотр\_активного\_менеджера», реализуем еще одно событие.

1. На вкладке События выбираем ПередЗагрузкой
2. Добавляем следующий скрипт:

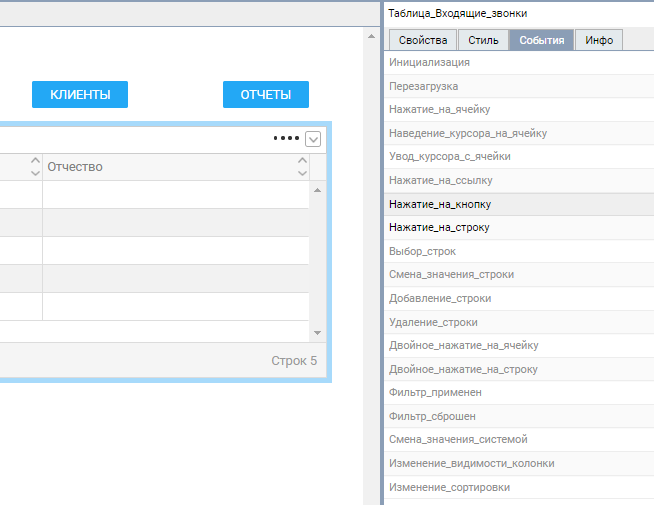
**ПередЗагрузкой**

udm:(Активный\_менеджер := Основной:Исполнитель:По\_выбран(true).ФИО);

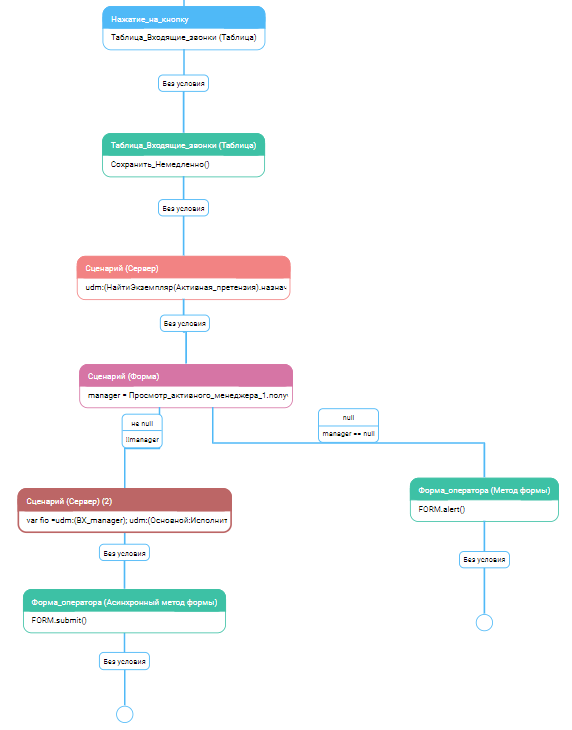
**Форма "Форма\_оператора"**

При нажатии на кнопку принять звонок нам необходимо получить ФИО активного менеджера, пометить, что он выбран, и добавить связку выбранного входящего звонка и активного менеджера.

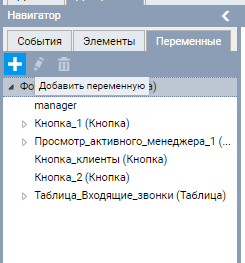
 На форме необходимо нажать на таблицу, перейти на вкладку События и выбрать Нажатие\_на\_кнопку.



Общий вид события "Нажатие\_на\_кнопку":



Добавляем глобальную переменную на форму с названием manager, для этого в Навигаторе переходим на вкладку Переменные, нажимаем на "Форма\_оператора" и плюс. Задаем имя переменной - manager



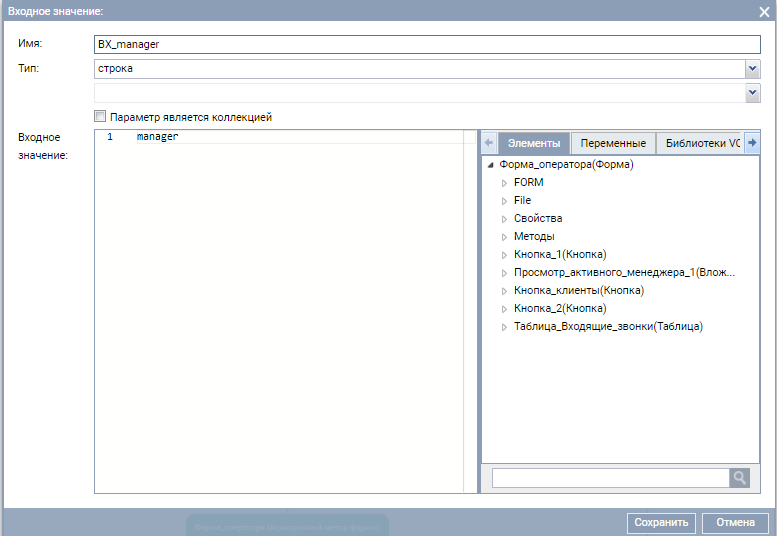
Действие Сохранить\_немедленно() и первый сценарий (сервер) у нас уже есть, поэтому после сценария (сервер) добавляем сценарий (форма),  где получаем ФИО активного менеджера и записываем в переменную manager:

|  |
| --- |
| manager = Просмотр\_активного\_менеджера\_1.получитьФИО(); |

После сценария (форма) идет разветвление. Условие для первого перехода:

|  |
| --- |
| !!manager |

После перехода добавляем сценарий (сервер) и задаем ему **входной**параметр. Необходимо нажать на сам блок сценарий (сервер) и в правой панели нажать на кнопку Добавить параметр.



Внутрь сценария (сервер) добавляем следующий скрипт:

var fio =udm:(ВХ\_manager);

udm:(Основной:Исполнитель:все().выбран := false);

var manager = udm:(Основной:Исполнитель:По\_фио(js:string(fio)))[0];

manager.Выбран = true;

udm:(Процесс\_исполнитель := Основной:Процесс\_исполнитель:НОВЫЙ());

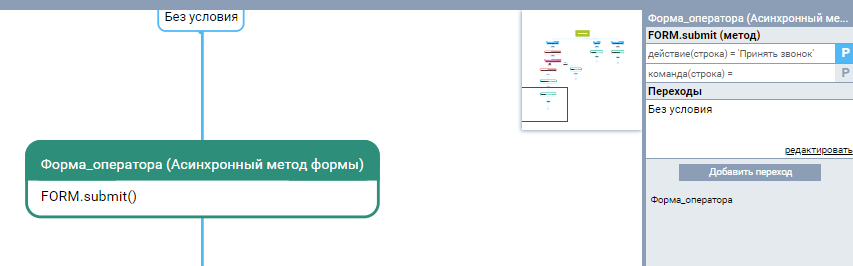
udm:(Процесс\_исполнитель.Ид\_исполнителя := ИДОБЪЕКТА(js:Основной:Исполнитель(manager)));

udm:(Процесс\_исполнитель.Ид\_процесса := Активная\_претензия);

Получаем ФИО активного менеджера, находим нужного исполнителя в базе данных и меняем ему атрибут «Выбран» на true, чтобы при переключении между формами у нас сохранялся выбранный менеджер. Чтобы не было проблем с тем, что у нас выбраны несколько менеджеров, проходимся по всем записям в таблице и обнуляем значение их атрибута «Выбран».

Создаем связку текущего процесса Входящий звонок и активного менеджера и добавляем ее в таблицу «Процесс\_исполнитель», это пригодится нам на форме «Заключение ответственного исполнителя».

Добавляем FORM.submit() с действием «Принять звонок», чтобы перейти дальше по процессу.



Условие для второго перехода:

|  |
| --- |
| manager == null |

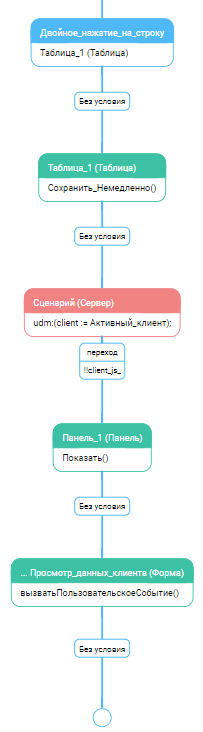
Затем добавляем FORM.alert() с заголовком «Выберите менеджера!».

## Настройка связи между формами «Просмотр клиентов» и «Просмотр данных клиента»

При двойном нажатии на строку с клиентом открывается дочерняя форма, где можно изменить данные клиенты. Затем при нажатии на кнопку сохранить на дочерней форме таблица с клиентами обновляется. Для реализации данной логики необходимо сделать следующее

### Форма "Просмотр\_клиентов"

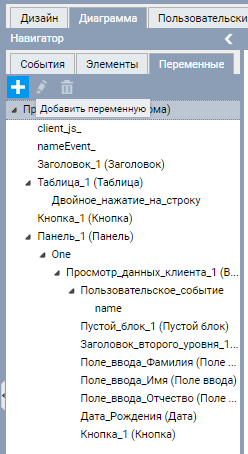
Выбираем таблицу с клиентами, переходим на вкладку События и выбираем действие Двойное\_нажатие\_на\_строку. Общий вид события:



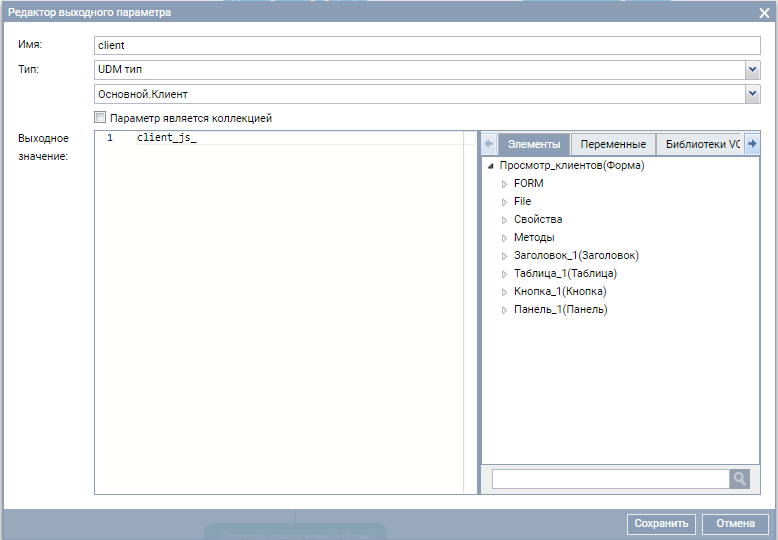
Добавляем сценарий (сервер) со следующим скриптом:

|  |
| --- |
| udm:(client := Активный\_клиент); |

На форме создаем глобальную переменную client\_js\_



Для сценария (сервер) также добавим выходной параметр.



После сценария сервера прописываем условие перехода:

|  |
| --- |
| !!client\_js\_ |

Вызываем у панели метод Показать(). Добавляем вызов пользовательского события на дочерней форме

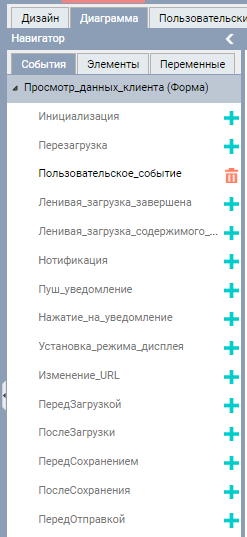


|  |  |
| --- | --- |
| Наименование (строка) | "получитьДанныеКлиента" |
| Данные | client\_js\_ |

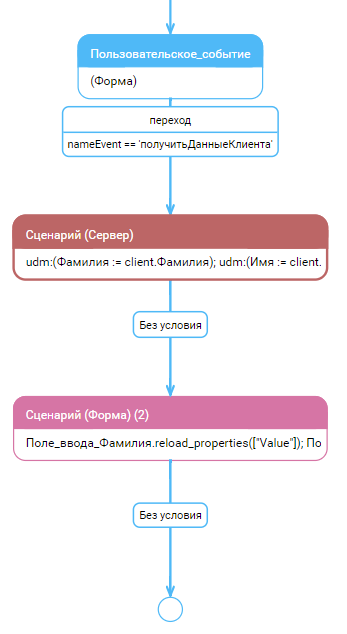
### Форма "Просмотр\_данных\_клиента"

Переходим на форму «Просмотр\_данных\_клиента» для реализации пользовательского события.

Во вкладке События нажимаем на плюс возле Пользовательское\_событие



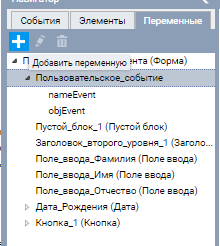
Общий вид Пользовательского события:



Для этого события нам необходимо создать 2 переменные nameEvent и objEvent.

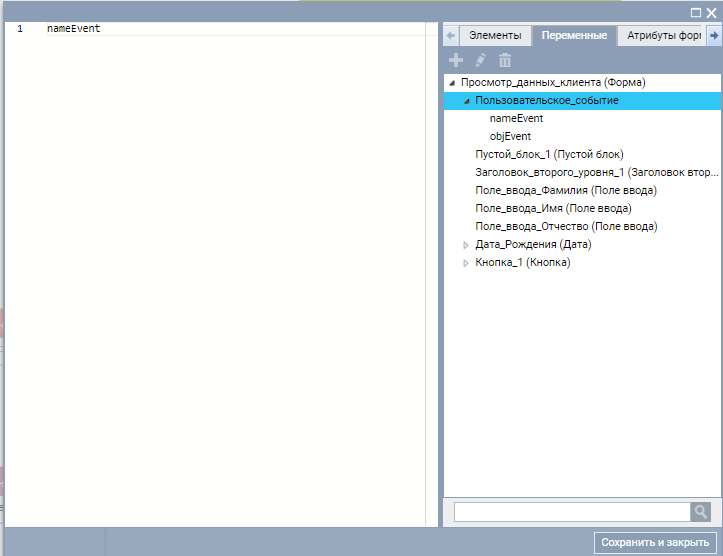
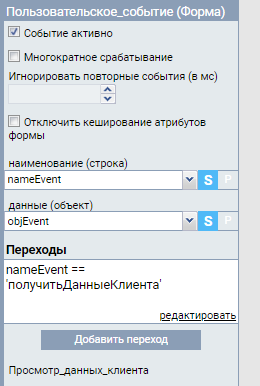
* Способ 1:

Перейти во вкладку Переменные, выбрать Пользовательское\_событие и нажать на плюс



* Способ 2:

Возле строки с выбором переменной для «Наименования» или «Данных» пользовательского события нажать на Р. В открывшемся окне выбрать пользовательское событие и нажать на плюс.



У пользовательского события в графе «наименование (строка) выбираем nameEvent, в графе «данные (объект)» выбираем objEvent.

Добавляем условие для перехода

|  |
| --- |
| nameEvent == 'получитьДанныеКлиента' |

Добавляем сценарий сервер с входным параметром:

**Сценарий (сервер)**

udm:(Фамилия := client.Фамилия);

udm:(Имя := client.Имя);

udm:(Отчество := client.Отчество);

udm:(Дата\_рождения := client.Дата\_рождения);

udm:(ид := ИДОБЪЕКТА(client));

Добавляем сценарий (форма) со скриптом:

**Сценарий (форма)**

Поле\_ввода\_Фамилия.reload\_properties(["Value"]);

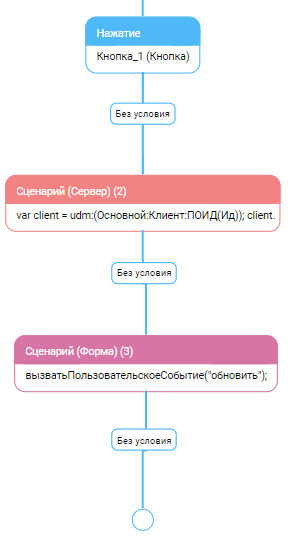
Поле\_ввода\_Имя.reload\_properties(["Value"]);

Поле\_ввода\_Отчество.reload\_properties(["Value"]);

Дата\_Рождения.reload\_properties(["Value"]);

После изменения данных клиента нажимаем на кнопку Сохранить, обновляем данные клиента в базе и обновляем саму таблицу на форме. Для реализации этой логики добавляем Событие "Нажатие\_на\_кнопку" для кнопки Сохранить.

Общий вид события "Нажатие\_на\_кнопку"



Добавляем сценарий (сервер) со скриптом:

**Сценарий (сервер)**

var client = udm:(Основной:Клиент:ПОИД(Ид));

client.Фамилия = udm:(Фамилия);

client.Имя = udm:(Имя);

client.Отчество = udm:(Отчество);

client.Дата\_рождения = udm:(Дата\_рождения);

udm:(ФИКСИРОВАТЬ());

Добавляем сценарий (форма), где вызываем пользовательское событие формы «Форма оператора».

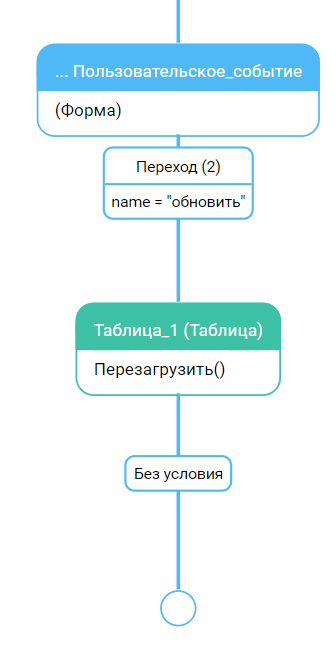
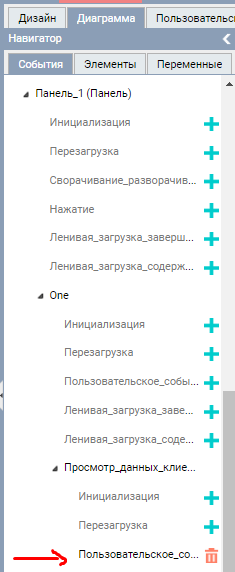
**Сценарий (форма)**

вызватьПользовательскоеСобытие("обновить");

Переходим обратно на форму «Форма оператора» для реализации этого пользовательского события.

### Форма "Просмотр\_клиентов"

Во вкладке События выбираем пользовательское событие внутри панели внутри дочерней формы.



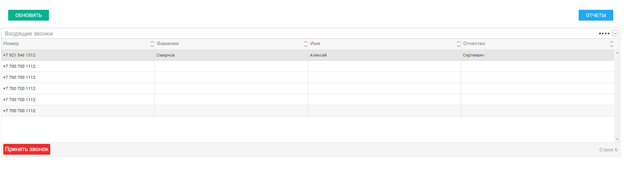
Создаем переменную внутри этого пользовательского события с названием name. Добавляем условие для перехода:

|  |
| --- |
| name == "обновить" |

Добавляем действие Таблица перезагрузить().

# Проверка работы приложения

Открываем АРМ Пользователя, авторизуемся, выбираем наше приложение

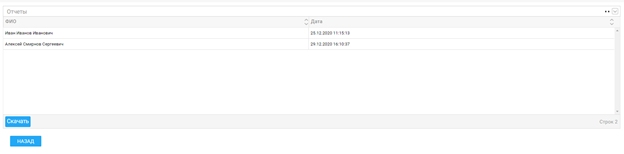


Убедимся, что приложение запускается и работает корректно.

Чтобы загрузить звонки в таблицу входящий звонок нам нужно:

* запустить процесс **Генерация\_данных** (выполняется 1 раз. заполнит таблицы начальными значениями - номера, филиалы, и т.д.), а далее **Генератор\_звонков** (запустит процессы **Входящий\_звонок** для разных номеров).
* или отправить запрос на веб-сервер с соответствующими данными.

При нажатии кнопки Отчеты мы попадаем на форму **Отчеты** , с помощью которой мы можем их просматривать и скачивать нужный.



При выборе звонка в таблице Входящие звонки на начальной форме есть кнопка Принять звонок, по нажатии на которую будет выбор:

1. Если данные о выбранном клиенте существуют, то сразу переходим к заполнению данных о претензии (форма **Заполнение\_данных\_претензии** **)**
2. Если данных нет, то заполняем форму Создание нового клиента (форма **Создание\_нового\_клиента** )

Поиск клиента:



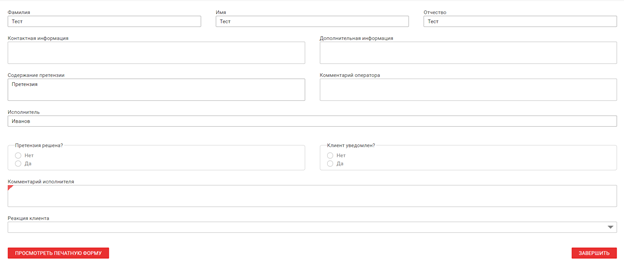
Создание нового клиента:



Заполнение данных претензии:



Заключение ответственного исполнителя:



На последней форме мы можем просмотреть и распечатать форму или завершить звонок.

Вы, вероятно, всё сделали правильно, если:

1. приложение запустилось и отобразились основные элементы
2. в таблице Входящих звонков отобразились данные, созданные при генерации
3. после заполнения претензии на шаге "Заключение ответственного исполнителя" вы можете Просмотреть печатную форму (вылезет окно с подготовленным документом из LibreOffice)
4. на основном экране приложения вы можете открыть список отчетов (кнопка Отчеты) и найти подготовленный под свои данные