# 

# Программное обеспечение AllegroClient. Руководство разработчика



Allegrosoft 2025. Версия 2.3.0

Содержание

**1 Настройка программного обеспечения5**

1.1Настройка AllegroClient-web расширением 1с для собственной разработки операций 5

1.1.1. Подключение расширения 1с5

1.1.2. Регистрация HTTP-сервиса 1с5

1.1.3. Как открыть обработку для настройки 6

1.1.4. Как создать пользователя для подключения к http-сервису 8

1.1.5. Как настроить подключение в клиентском приложении 8

1.2 Настройка AllegroClient – web c расширением AllegroCount……………………………………………………9

1.2.1. Публикация расширения 9

1.2.2. Регистрация HTTP-сервиса 9

1.2.3. Выгрузка форм 9

1.2.4. Создание пользователя для подключения к http-сервису 14

1.2.5. Настройка в клиентском приложении 14

1.2.6. Описание формата JSON выгруженной формы**.**14

1.3 Настройка AllegroClient – prof 2.018

1.3.1. Установка Сервера ТСД18

1.3.2. Настройка Сервера ТСД 19

1.3.3. Выгрузка Форм 20

1.3.4. Как настроить подключение в клиентском приложении 21

**2 Конфигурирование логики работы с операциями в режиме онлайн23**

2.1 Разработка макета форм23

2.1.1. Разработка макетов форм для AllegroClient-prof 2.0 и для AllegroClient- web23

2.1.2. Разработка макетов форм для расширения AllegroCount 23

2.2. Описание серверной функции для обработки событий от клиента 23

2.2.1. Функция для AllegroClient-prof 2.0 и для AllegroClient-web с расширением 1с для

собственной разработки операций 23

2.2.2 Функция для расширения AllegroCount для собственной разработки операций 23

2.3 Параметр «Код сотрудника»23

2.4 Параметр «Событие»24

2.5 Параметр «Наименование формы»24

2.6 Параметр «Данные»24

2.7 Параметр «Массив с контролами»24

2.8 Работа с элементами форм 24

2.8.1 Общее описание 25

2.8.2. Элемент управления «Надпись»26

2.8.3. Элемент управления «Поле ввода»26

2.8.4. Элемент управления «Список»26

2.8.5. Элемент управления «Список выбора»27

2.8.6. Элемент управления «Флажок»27

2.8.7. Элемент управления «Табличное поле»28

2.8.8. Элемент управления «Кнопка»32

2.8.9. Элемент управления «Изображение»32

2.9 Команды 33

2.9.1 Сохранение данных формы33

2.9.2. Настройка «Пропускать при вводе»34

2.9.3. Зарезервированное слово Null 35

2.9.4. Управление экранной клавиатурой 35

2.9.5. Команда «RepeatCall»35

2.9.6. Команда «SUCCESS»35

2.9.7. Команда «ShowToast»35

2.9.8. Команда «PlayErrorSound»35

2.9.9. Команда «PlayBeep»36

2.9.9.1 Команда «AttachIdleHandler»36

2.9.9.2 Команда «setEventAsync»36

2.10 Диалоговые окна 36

2.10.1. Диалоговое окно с ошибкой 36

2.10.2. Диалоговое окно с вопросом 36

2.10.3. Диалоговое окно с сообщением 36

2.11 Сканирование штрих-кодов 37

2.11.1 Сканирование камерой устройства 37

2.11.2 Сканирование в разрыв клавиатуры 37

2.11.3 Сканирование в режиме BroadCast 37

2.12 Создание фотографий………………………………………………………………………………………………………………37

2.13 Вывод табличного документа на экране устройства………………………………………………………………38

2.14 Печать на Bluetooth принтер……………………………………………………………………………………………………39

2.14 Назначение кнопкам аппаратных клавиш………………………………………………………………………………41

2.15 Эмулятор ТСД 42

**3 Конфигурирование в режиме офлайн44**

3.1 Режим офлайн общие принципы 44

3.1.1. Общее описание выполнения кода на клиенте 44

3.1.2. Общие процедуры и функции для работы с JavaScript 45

3.1.3 Глобальная переменная AllegroModule.JSStatus…………………………………………..…………………..49

3.1.4. Выгрузка JavaScript в клиентское приложение 49

3.1.5 Формирование шаблона JavaScript…………………………………………………………………………………….51

3.2 Работа со справочниками в режиме офлайн 52

3.2.1 Общее описание работы 52

3.2.2. Загрузка справочников из 1С 52

3.2.3. Загрузка справочников на клиенте 54

3.2.4. Работа с данными справочников……………………………………………………………………………………..54

3.3 Работа с документами в режиме офлайн 56

3.3.1 Структура таблиц документов 56

3.3.2 Работа с таблицами документов 57

3.3.3 Команды для загрузки документов с сервера 59

**4 Работа с веб сокетами** **60**

**5 Настройка сканирования на ТСД для работы с AllegroClient** **61**

4.1 Общее описание 61

4.1.1.Пример настройки для терминалов сбора данных ATOL серии Smart Lite и Slim 61

4.1.2. Пример настройки сканирования для терминала сбора данных Urovo 62

4.1.3. Пример настройки сканирования для терминала сбора данных Honeywell 64

4.1.4. Пример настройки сканирования для терминала сбора данных Mertech 64

4.1.5. Пример настройки сканирования для терминала сбора данных iData 68

4.1.6. Пример настройки сканирования для терминала сбора данных Unitech 70

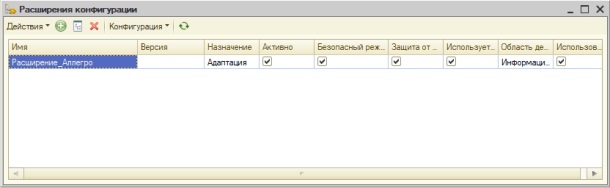
4.1.7. Пример настройки сканирования для терминала сбора данных Zebra 71

## Настройка программного обеспечения

## *Настройка AllegroClient – web c расширением 1с для собственной разработки операций*

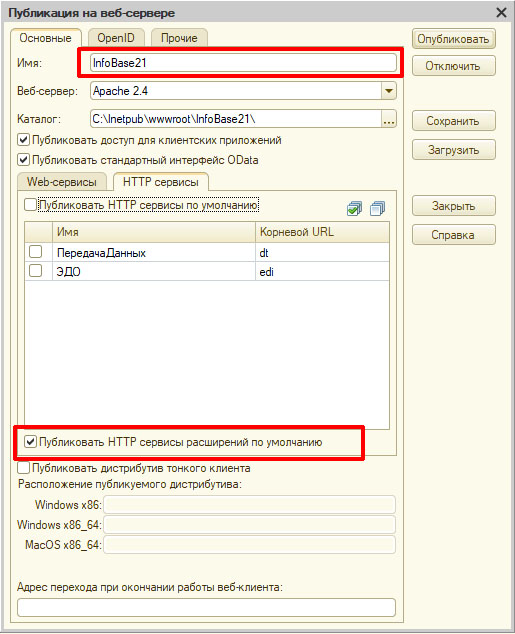
**1.1.1. Подключение расширения 1с.**

Подключить расширение конфигурации, идущее в поставке следующим образом.



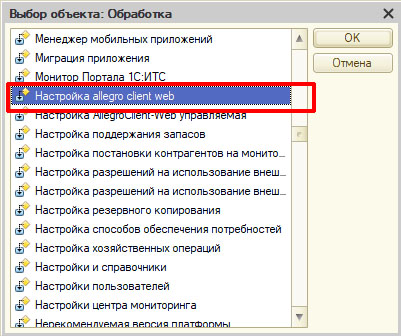
**1.1.2. Регистрация HTTP-сервиса.**

Зарегистрировать HTTP-сервис из расширения конфигурации. Для этого необходимо поставить флажок «Публиковать HTTP сервисы расширений по умолчанию». Имя сервиса (в нашем случае «InfoBase21») будет использоваться для заполнения URL в настройках на клиенте.

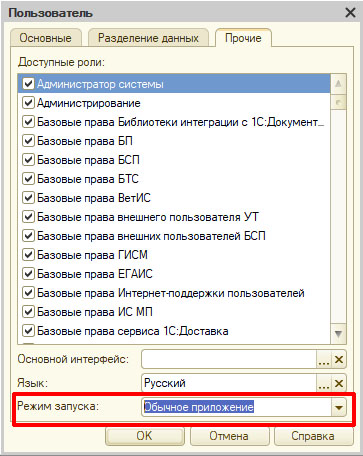


**1.1.3. Как открыть обработку для настройки**

Необходимо открыть 1С в режиме предприятия и открыть обработку «Настройка AllegroClient web».

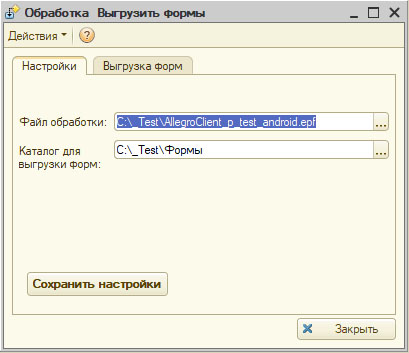


**Важно!**  Систему 1С необходимо открыть обычном в режиме. У пользователя должен стоять режим «Обычное приложение».

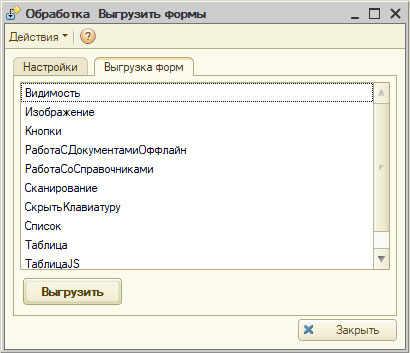


В обработке настроить путь к обработке, в которой разрабатывается логика работы на клиенте. И произвольный каталог, в который будут выгружаться формы (перечень форм с размерами, расположением, оформлением элементов управления).

В обработке настроить путь к обработке, в которой разрабатывается логика работы на клиенте. И произвольный каталог, в который будут выгружаться формы (перечень форм с размерами, расположением, оформлением элементов управления).

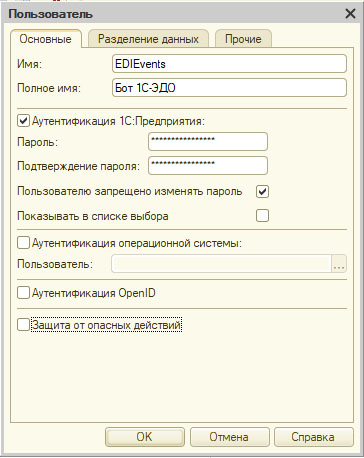


Выгрузить формы. Увидим перечень выгруженных форм.



**1.1.4. Как создать пользователя для подключения к http-сервису.**

Создать пользователя для подключения к http-сервису. **Важно!** Пользователь должен иметь права администратора.

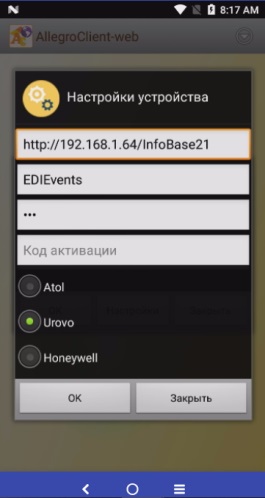


**1.1.5. Как настроить подключение в клиентском приложении.**

На клиентском приложении заполнить настройки.

* URL – доменное имя сервера (IP) + имя http-сервиса
* Пользователя – имя созданного пользователя для подключения

Пароль - пароль пользователя для подключения



## *Настройка AllegroClient – web c расширением AllegroCount*

**1.2.1. Публикация расширения.**

Публикация расширения описано в пункте 1.1.1.

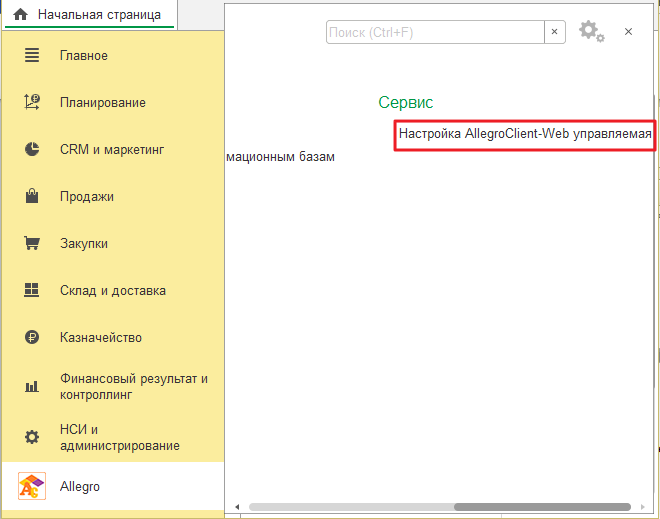
**1.2.2. Регистрация HTTP-сервиса.**

Регистрация HTTP-сервиса описано в пункте 1.1.2

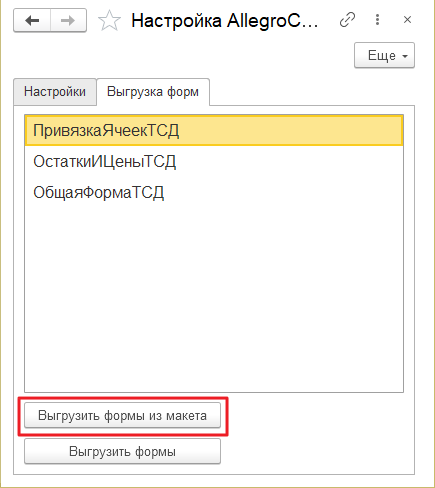
**1.2.3. Выгрузка форм.**

Выгрузка форм . 3 варианта:

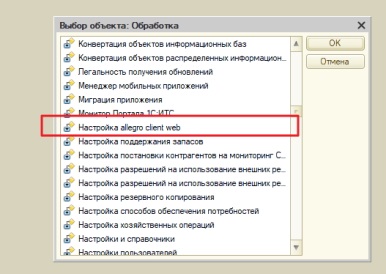
a) Формы не меняем, используем штатные. Открываем обработку в системе Allegro.



В закладке «Выгрузка форм» выгружаем формы из макета.

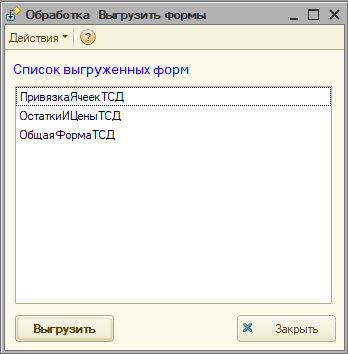


б) После изменения штатных форм (общие формы в расширении). Формы необходимо выгрузить с помощью обработки:



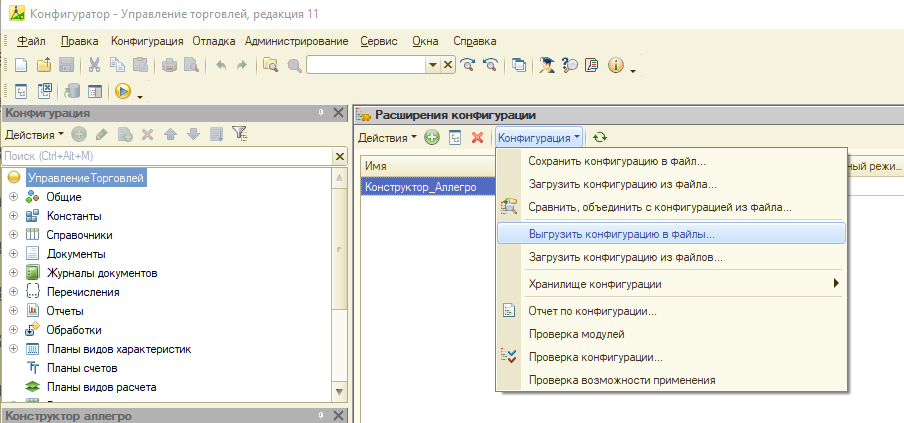
**Важно!**  Систему 1С необходимо открыть обычном в режиме. У пользователя должен стоять режим «Обычное приложение».

Выгрузить формы. Увидим перечень выгруженных форм.

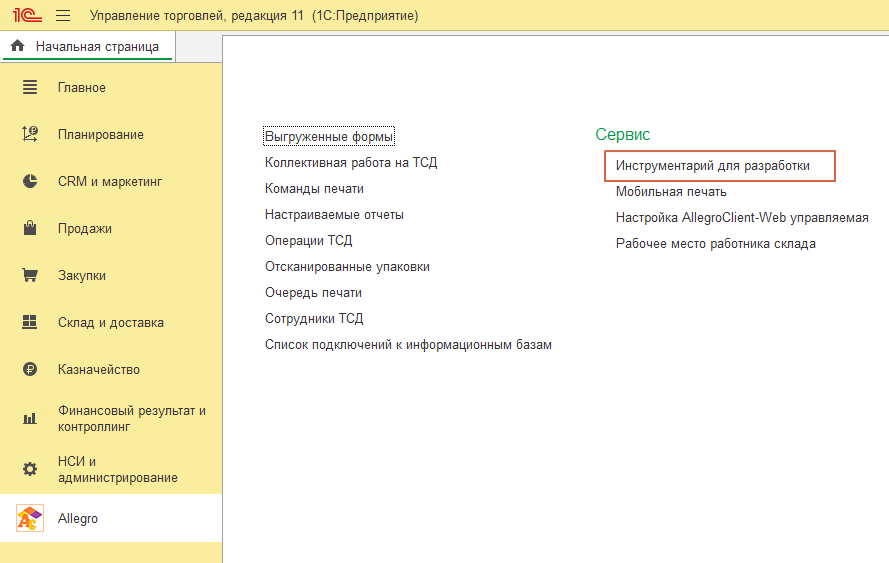


в) После изменения штатных форм (общие формы в расширении).

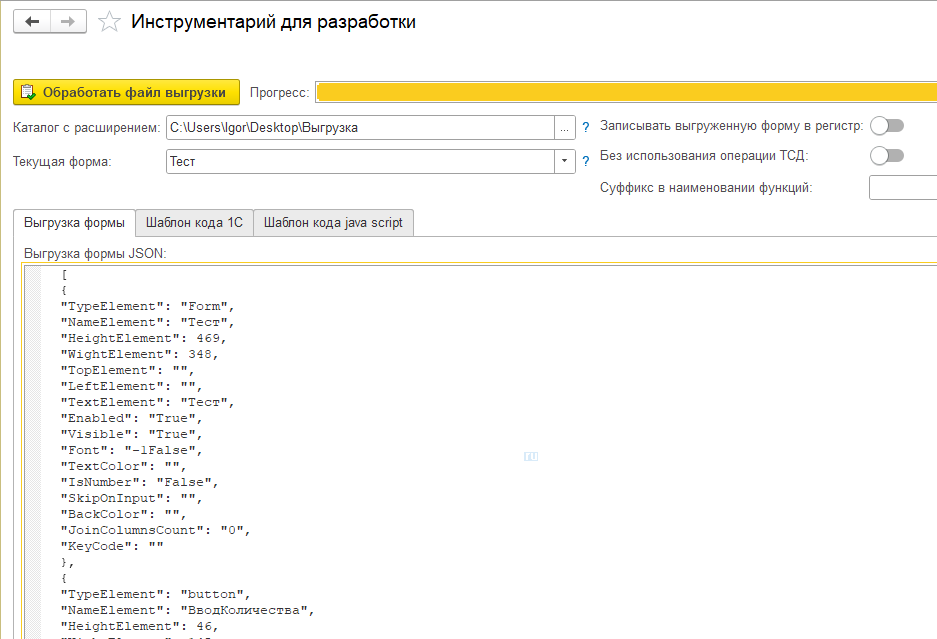
Выгружаем расширение в файлы в отдельный каталог.



В режиме исполнения 1с открываем обработку «Инструментарий для разработки».

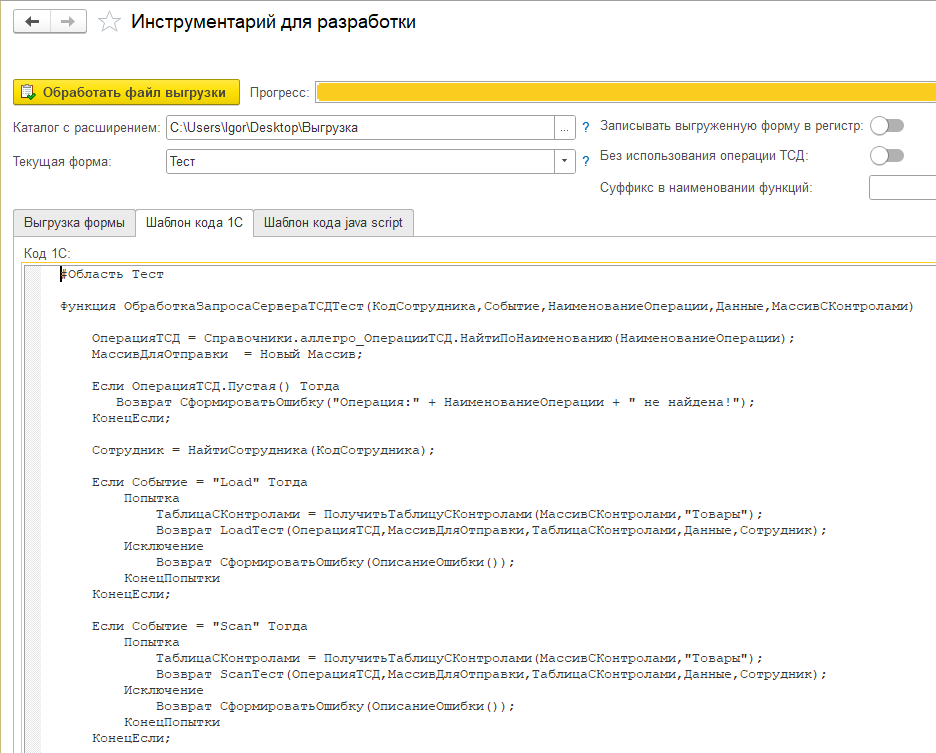


В обработке выбираем каталог с выгруженным расширением. Выбираем форму, которую необходимо выгрузить. И нажать «обработать файл выгрузки».



Будет сформирован Json (выгрузка формы), который необходимо поместить в общий макет в расширении, который связан выгружаемой формой.

Обработка так же формирует шаблон кода 1С и JavaScript (шаблон JavaScript будет описан в разделе разработки офлайн событий). В шаблоне формируются пустые обработчики нажатия кнопок и предопределенные события формы, процедуры для управления видимостью элементов. Шаблон кода значительно сокращает время разработки.



**1.2.4. Создание пользователя для подключения к http-сервису**

Создать пользователя для подключения к http-сервису. Описано в пункте 1.1.4.

**1.2.5. Настройка в клиентском приложении.**

Описано в пункте 1.1.5.

**1.2.6. Описание формата JSON выгруженной формы.**

*Параметры формы:*

FormName – имя формы

Height – высота формы

Width – ширина формы

*Общие параметры все всех элементов управления:*

NameElement – наименование элемента

TypeElement – тип элемента, ниже соответствие элементам 1с

* Label - Надпись
* Button -Кнопка
* TextBox - Поле ввода
* ComboBox – Выпадающий список
* CheckBox – Флажок
* ListBox – Список
* DateTimePicker – Поле ввода с типом дата
* ImageView – Картинка
* DataGrid – Табличное поле
* SpreadSheetDocument – Табличный документ

Height – высота элемента, тип число

Width - ширина элемента, тип число

Top- отступ сверху, тип число

Left- отступ слева, тип число

Enabled- использование, значение строка «True» или «False»

Visible – видимость, значение строка «True» или «False»

SkipOnInput- значение, значение строка «True» или «False», значение элемента не будет передаваться в 1С (на сервер)

Page – страница на которой расположен элемент

Value- значение. К примеру, текст кнопки, надписи, поля ввода.

*Дополнительные параметры для элементов, кроме DataGrid,* *ImageView, SpreadSheetDocument*

Font – шрифт в формате XXFalse или XXTrue, тип строка, где True/False – Bold, XX – размер шрифта, если значение меньше 10, обязателен лидирующий 0.

FontColor – цвет шрифта в формате RGB через «/», тип строка, пример - 0/0/0.

BackColor- цвет фона в формате RGB через «/», тип строка, пример - 0/0/0.

*Дополнительный параметр для TextBox:*

IsNumber – числовое поле , значение строка «True» или «False»

*Параметры колонок для DataGrid:*

TypeElement – тип элемента: Label, Image, CheckBox

Width – ширина колонки (вес)

Font - шрифт в формате XXFalse или XXTrue, тип строка, где True/False – Bold, XX – размер шрифта, если значение меньше 10, обязателен лидирующий 0.

FontColor – цвет шрифта в формате RGB через «/», тип строка, пример - 0/0/0.

Italic – наклон шрифта, тип строка, pyfxtybz False или True

HorizontalAlign – расположение по горизонтали значения: Left, Center, Right

Location – расположение колонки, число, значения 0,1,2.

* 0 – новая колонка
* 1 – на следующей строке
* 2 – в той же колонке

Примеры:

**С TextBox:**

{

"Width": 350,

"Height": 500,

"FormName": "Новая форма",

"Controls": [

{

"TypeElement": "TextBox",

"NameElement": "TextBoxEx\_1",

"Page": "Страница 1",

"Top": 238,

"Left": 125,

"Width": 100,

"Height": 24,

"Enabled": "True",

"Visible": "True",

"SkipOnInput": "False",

"Font": "09False",

"FontColor": "0/0/0",

"BackColor": "255/255/255",

"Value": "Текст",

"IsNumber": "False"

}

]

}

**С DataGrid:**

{

"Width": 350,

"Height": 500,

"FormName": "Новая форма",

"Controls": [

{

"TypeElement": "DataGrid",

"NameElement": "DataGridView\_1",

"Page": "Страница 1",

"Top": 175,

"Left": 25,

"Width": 300,

"Height": 150,

"Enabled": "True",

"Visible": "True",

"SkipOnInput": "False",

"JoinColumnsCount": "-1",

"Columns": [

{

"TypeElement": "Label",

"Width": 50,

"Font": "09,False",

"FontColor": "",

"Italic": "False",

"HorizontalAlign": "Left",

"Location": 0

},

{

"TypeElement": "Label",

"Width": 50,

"Font": "09,False",

"FontColor": "",

"Italic": "False",

"HorizontalAlign": "Left",

"Location": 0

}

]

}

]

}

## *Настройка AllegroClient – prof 2.0*

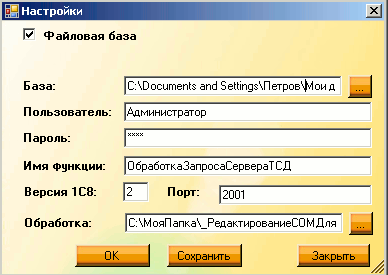
**1.3.1. Установка Сервера ТСД**

Установить сервер ТСД. Для этого нужно запустить файл SetupAllegroClient.msi и следовать инструкциям мастера установки.

**1.3.2 Настройка Сервера ТСД**

Настройка Сервера ТСД :

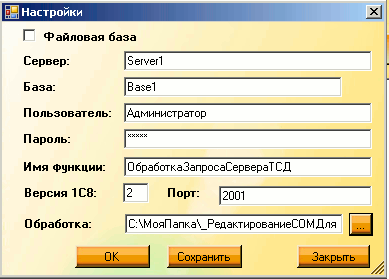
a) Настройка для файловой базы 1С.



**Важно!** Пользователь 1С, которого указываете в настройках сервера, должен иметь режим запуска «Обычное приложение».

* База - путь к базе 1С.
* Пользователь - указываем пользователя 1С (должен иметь права администратора)
* Пароль- пароль пользователя 1С
* Имя функции - указываем имя функции «ОбработкаЗапросаСервераТСД», данная функция содержится в обработке, которая имеется в комплекте поставки.
* Версия 1С8 - для 1С версии 8.2 ставим 2 для 8.3 соответственно 3
* Порт - любой, укажем 2001 (в дальнейшем этот же порт установим в настройках клиентского приложения)
* Обработка - путь к обработке "АллегроКлиент.epf" (обработка содержится в поставке «AllegroClient»)

b) Настройка для серверной базы 1С.



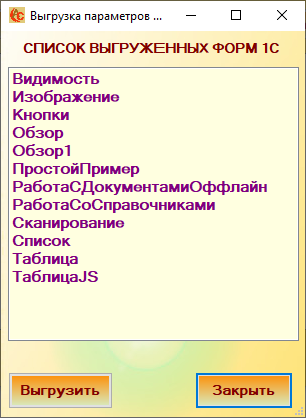
**Важно!** Пользователь 1С, которого указываете в настройках сервера, должен иметь режим запуска «Обычное приложение».

* Для серверной системы 1С, снимаем флажок «Файловая база», заполняем следующие настройки:
* Сервер - наименование сервера 1С.
* База - наименование базы 1С.
* Пользователь - указываем пользователя 1С (должен иметь права администратора)
* Пароль- пароль пользователя 1С
* Имя функции- указываем имя функции «ОбработкаЗапросаСервераТСД».
* Версия 1С8 - для 1С версии 8.2 ставим 2 для 8.3 соответственно 3
* Порт - любой, укажем 2001 (в дальнейшем этот же порт установим в настройках клиентского приложения)

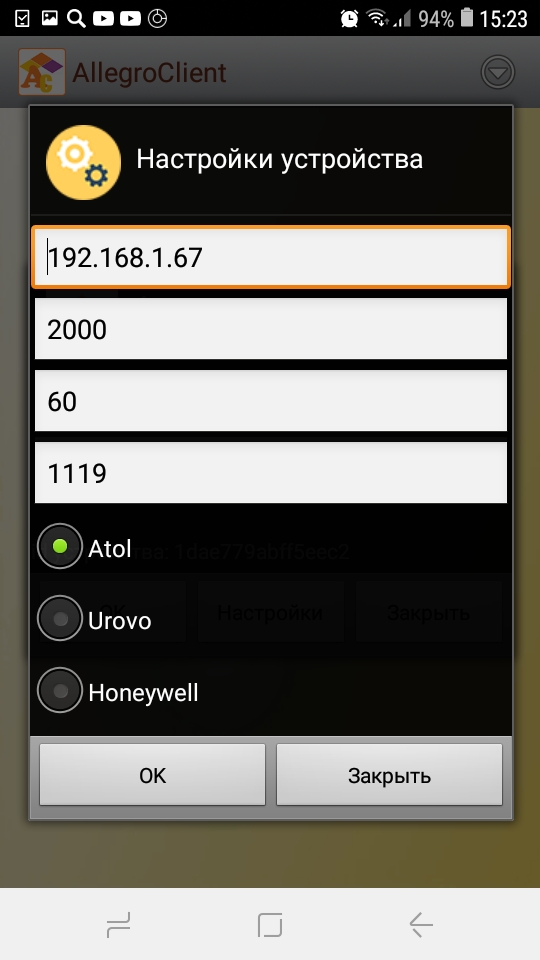
Обработка - путь к обработке " АллегроКлиент.epf" (обработка содержится в поставке «AllegroClient») 

**1.3.3. Выгрузка Форм**

Выгрузить формы. В меню Сервера ТСД нужно выбрать Сервис -> Выгрузка параметров форм



**1.3.4. Как настроить подключение в клиентском приложении.**



В настройках необходимо указать:

* IP адрес сервера — IP адрес ПК, на котором установлен «Сервер ТСД».
* Порт – порт, который был указан в настройках «Сервера ТСД»
* Время ожидания — время ожидания ответа от сервера в секундах, рекомендуем установить 30 секунд

## Конфигурирование логики работы с операциями в режиме онлайн

### *Разработка макета форм*

**2.1.1. Разработка макетов форм для AllegroClient-prof 2.0 и для AllegroClient- web**

Разработка макетов форм для AllegroClient-prof 2.0 и для AllegroClient- web c расширением 1с для собственной разработки операций. Разработка форм осуществляется во внешней обработке, отдельная форма является отдельной операцией.

**2.1.2. Разработка макетов форм для расширения AllegroCount**

Разработка макетов форм для расширения AllegroCount осуществляется в самом расширении путем добавления/изменения общих форм. Так как решение является гибким для настройки различных операций, предполагается, что необходимости в добавлении новых форм нет. Вполне достаточно будет отредактировать отдельные формы из существующего списка.

#### 2.2. Описание серверной функции для обработки событий от клиента

**2.2.1 . Функция для AllegroClient-prof 2.0 и для AllegroClient-web с расширением 1с для собственной разработки операций.**

Вся логика работы разрабатывается в той же обработке что и проектировка форм. Разработка осуществляется в модуле обработки в экспортной функции (имя функции указывали ранее в настройках «Сервера ТСД» - ОбработкаЗапросаСервераТСД).

Функция содержит 5 входящих параметров:

* КодСотрудника
* Событие
* НаименованиеФормы
* Данные
* МассивСКонтролами

**2.2.2 Функция для расширения AllegroCount**  **для собственной разработки операций.**

Функция находиться в http-сервисе ServiceAllegro, называется SeverExchangePOST. Получает от клиента пакет json с такой же структурой параметров, который описан в пункте 2.2.1.

* Id (КодСотрудника)
* Event (Событие)
* Form (НаименованиеФормы)
* Data (Данные)
* Arrcontrols (МассивСКонтролами)

#### 2.3 Параметр «Код сотрудника»

**КодСотрудника** – данный параметр необходим для авторизации сотрудника на клиенте, можно использовать любой реквизит, однозначно идентифицирующий пользователя 1С, лучше использовать короткий набор символов для быстрого набора в поле авторизации ТСД.

#### 2.4 Параметр «Событие»

**Событие** – наименование события от клиента. Необходимо выделить два типа событий зарезервированные системой и не зарезервированные.

***Зарезервированные события***

Identification – событие срабатывает при авторизации на клиенте. В этом событии необходимо сделать проверку параметра «КодСотрудника».

Load - событие при открытии формы отдельной операции (формы).

Scan – событие срабатывает при сканировании. Параметр «Данные», при срабатывании события, возвращает отсканированный штрих-код.

***Не зарезервированные события***

Данные события возникают при нажатии кнопок на ТСД, а так же клавиши «Enter» на поле ввода. Событие имеет наименование такое же, как наименование кнопки или поля ввода, присвоенное при разработке форм.

#### 2.5 Параметр «Наименование формы»

**НаименованиеФормы** - наименование текущей формы, в которой происходит работа на ТСД. Возвращается при любом событии кроме «Identification».

#### 2.6 Параметр «Данные»

**Данные** – возвращает штрих-код при событии «Scan».

#### 2.7 Параметр «Массив с контролами»

**МассивСКонтролами** – Сом-массив (обычный массив для AllegroClient-web) в строках которого содержаться все наименования элементов формы текущей формы с их текущими значениями, значениями видимости, доступности, наличием фокуса через разделитель «;». Строка массива имеет следующий формат: **НаименованиеЭлемента;ЗначениеЭлемента;**

Наименование элемента – наименование элемента, которое было задано элементу в конфигураторе.

Значение элемента – строковое представление значения элемента формы

#### 2.8 Работа с элементами форм

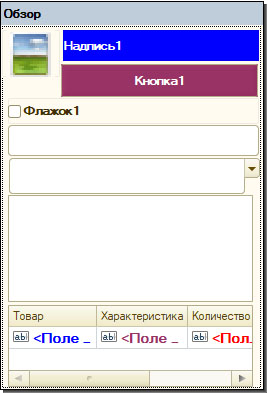
**2.8.1 Общее описание.**

Элементы форм динамически выстраиваются при открытии формы, которые были заранее выгружены в файлы. В выгрузке формируется структура элементов форм, которая определяет порядок и вид элементов, которые будут отображены на форме. Выгрузка из себя представляет структуру в формате json или xml для AllegroClient-prof 2.0.

Элементы формы, которые можно использовать:

* Надпись
* Поле ввода
* Список
* Список выбора
* Флажок
* Кнопка
* Табличное поле
* Картинка

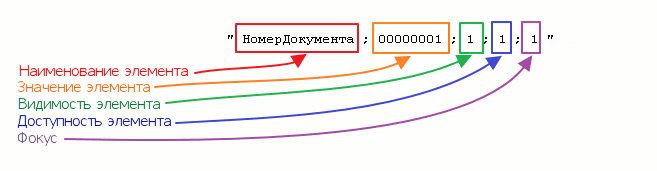
Спроектированная форма в 1с. Отображение формы на устройстве

В момент срабатывания события мобильный клиент отправляет пакет с данными, описанный в пункте 2.2.1 и 2.2.2, содержащий Массив с контролами. Массив содержит весь перечень элементов с текущими значениями этих элементов. Задача программиста - обработать данный массив с элементов и сформировать новый массив для отправки на мобильный клиент, чтобы обновить значения элементов на форме. Массив для отправки на мобильное устройство выглядит следующим образом:

**Наименование;Значение;Видимость,Доступность;Фокус;**

* Наименование – наименование элемента
* Значение – устанавливает новое значение элемента
* Видимость – задает видимость элемента. Значения 1 , 0 или пустое значение (не менять значение).
* Доступность - задает доступность элемента. Значения 1, 0 или пустое значение (не менять значение).
* Фокус – устанавливает фокус на элемент. Значения 1 или 0 или пустое значение (не менять значение).



**2.8.2. Элемент управления «Надпись».**

Для элемента можно установить следующие настройки в макете формы 1с:

* Видимость
* Доступность
* Размер шрифта
* Жирность шрифта
* Цвет шрифта
* Фон элемента

**2.8.3. Элемент управления «Поле ввода»**

Для элемента можно установить следующие настройки в макете формы 1с:

* Видимость
* Доступность
* Размер шрифта
* Жирность шрифта
* Цвет шрифта
* Фон элемента
* Тип значения. Может быть: Строка, Дата, Число.

Элемент «Поле ввода» с типом «Дата» - «xx.xx.xx», где через точку день, месяц, год.

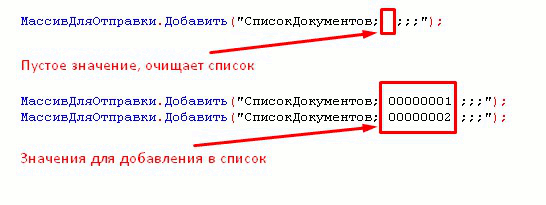
**2.8.4. Элемент управления «Список»**

Для элемента можно установить следующие настройки в макете формы 1с:

* Видимость
* Доступность
* Размер шрифта
* Жирность шрифта
* Цвет шрифта

Для элемента «Поле списка» имеются особенности формата получаемых данных. Мобильный клиент возвращает значения выделенной строки в формате: НаименованиеЭлемента;ЗначениеВСтроке;

Для элемента «Поле списка» имеются особенности формата возвращаемых данных. Так как это список, можно передать несколько значений для заполнения списка, каждое значение отдельным элементом массива. Для очистки списка, в значении элемента нужно использовать пустую строку.



**2.8.5. Элемент управления «Список выбора»**

Для элемента можно установить следующие настройки в макете формы 1с:

* Видимость
* Доступность
* Размер шрифта
* Жирность шрифта
* Цвет шрифта

Заполнение списка так же как, для элемента «Список», пункт 3.6.4.

**2.8.6. Элемент управления «Флажок»**

Для элемента можно установить следующие настройки в макете формы 1с:

* Видимость
* Доступность
* Размер шрифта
* Жирность шрифта
* Цвет шрифта

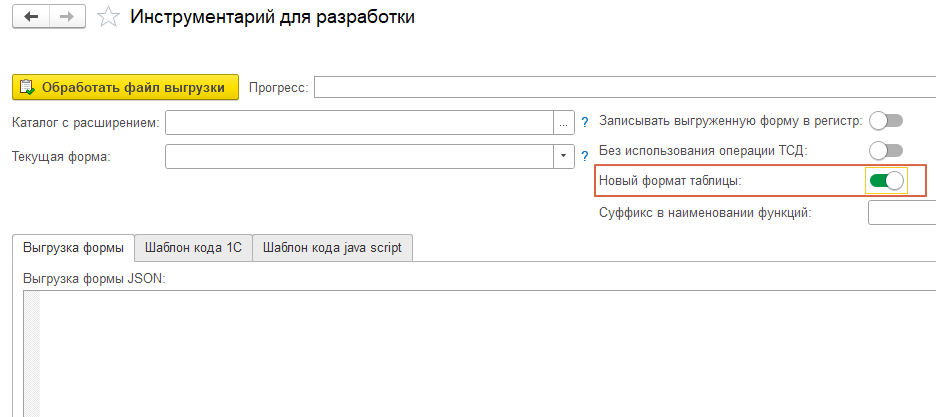
Для элемента можно установить следующие значения:

* 1 – флажок установлен
* 0 – флажок снят
* Пустое значение – не менять значение

**2.8.7. Элемент управления «Табличное поле»**

Для элемента можно установить следующие настройки в макете формы 1с:

* Видимость
* Доступность

Для элемента реализован старый и новый формат таблицы, определяющий принцип формирования элемента и внешний вид таблицы. Для формирования нового формата таблицы необходимо использовать обработку «Инструментарий для разработки» ( инструмент был описан в пункте 1.2.3 (в)) и использовать выключатель «Новый формат таблицы».

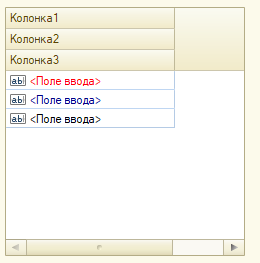
Рассмотрит два эти формата таблицы.

***Старый формат таблицы***

Для отдельной колонки можно установить следующие настройки:

* Размер шрифта
* Жирность шрифта
* Цвет шрифта

Так же можно сгруппировать колонки следующим образом:



**Важно!** Можно так реализовать только в первой колонке!

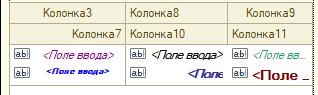
**Важно!** Типовое решение AllegroCount реализовано с использованием старого формата таблицы, при доработке типовых операций необходимо учитывать это.

***Новый формат таблицы***

Группировать колонки можно без ограничений.

Для отдельной колонки можно установить следующие настройки:

* Размер шрифта
* Жирность шрифта
* Цвет шрифта
* Наклон шрифта
* Выравнивание по горизонтали



Также можно вывести изображение в колонке, для этого в имени колонки должен быть префикс/суффикс «Image». Можно вывести флажок в колонке, для этого в имени колонки должен быть префикс/суффикс «CheckBox».

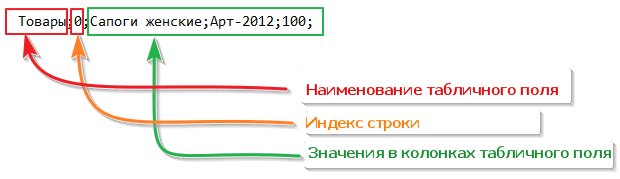


**Важно!** Максимальное количество колонок для отдельного табличного поля не более **9**, это относится к обеим форматам таблицы.

Формат получаемых данных от мобильного клиента имеет свои особенности. С клиента возвращается выделенная строка в следующем формате:

НаименованиеЭлемента ; ИндексСтроки ; Значение в колонке1; Значение в колонке2;…

Индекс строки – номер строки по порядку, нумерация начинается с 0.  
После индекса строки перечисляются все значения в колонках по порядку.



Если необходимо получить все строки с мобильного клиента, то нужно поступить следующим образом. Необходимо создать кнопку и сформировать имя кнопки следующим образом:

ИмяКнопки+ ИмяТабличногоПоля

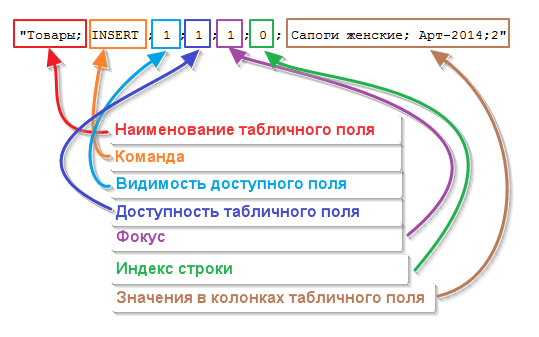
При нажатии данной кнопки клиент возвратит все строки табличного поля, имя которого содержится в наименовании кнопки.

Для элемента «Табличное поле» в формате возвращаемых данных используются команды. Использование команд позволяет управлять заполнением и изменением состава строк в «Табличном поле» на стороне ТСД.

Список команд:

* Add – добавляет строку в «Табличное поле»
* Update – обновляет конкретную строку в «Табличном поле»
* Delete – удаляет конкретную строку в «Табличном поле»
* Insert – вставляет конкретную строку в «Табличное поле»
* Clear – очищает строки в «Табличном поле»
* SelectItem – перевод курсора на конкретную строку «Табличного поля»

Формат строки выглядит следующим образом:  
Наименование табличного поля; Команда; Видимость; Доступность;Фокус; ИндексСтроки; Значение колонки 1; Значение колонки 2; Значение колонки N;…



Пример:



Команда «setColorToEqualValue» - сравнивает значения в двух колонках указанных в команде, если значения равны, то меняет цвет текста в сравниваемой колонке (цвет также указывается в команде). Использовать необходимо при открытии формы в событии «Load». Обычно используется для выделения фактического количества, если плановое и фактическое количество равны.

Пример:



В данном примере сравнивается колонка за номером 5 с колонкой 4, если значения в колонках равны, цвет текста в колонке 5 меняется на rgb(0,128,0).

Команда «setTableClickable» - устанавливает кликабельность для таблицы. При клике на таблице отрабатывает событие равное наименованию таблицы. Использовать необходимо при открытии формы в событии «Load».

Пример:



**2.8.8. Элемент управления «Кнопка».**

Для элемента можно установить следующие настройки в макете формы 1с:

* Видимость
* Доступность
* Размер шрифта
* Жирность шрифта
* Цвет шрифта
* Фон элемента

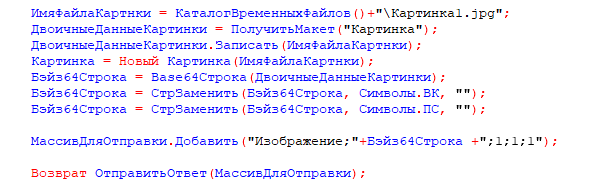
**2.8.9. Элемент управления «Изображение»**

Для элемента можно установить следующие настройки в макете формы 1с:

* Видимость
* Доступность

Для того чтобы отправить изображение на мобильный клиент, необходимо двоичные данные изображения преобразовать в формат Base64. Для очистки изображения необходимо отправить пустую строку.

Пример:



#### 2.9 Команды

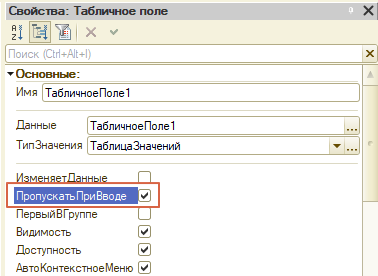
**2.9.1 Сохранение данных формы**

Значения элементов формы можно сохранить в текстовый файл на ТСД, и при необходимости загрузить сохраненные значения обратно в элементы формы.

Перечень элементов, значения которых можно сохранить в файл:

* Поле ввода
* Надпись
* Кнопка
* Табличное поле

По умолчанию все элементы формы сохраняются в файл, но, если необходимо, для отдельного элемента сохранение можно отключить. Для этого необходимо установить флаг «**Пропускать при вводе**».



Данные элементов формы сохраняются в два различных текстовых файла. В один из файлов сохраняются данные элемента «Табличное поле», в другой файл сохраняются данные всех возможных элементов, перечисленных выше. Для сохранения данных элемента «Табличное поле» используется команда SaveDocumentTable, имеет следующий формат:

**SaveDocumentTable;Уникальный набор символов;**

Для сохранения данных других элементов используется команда SaveDocumentHeader, имеет следующий формат:

**SaveDocumentHeader;Уникальный набор символов;**

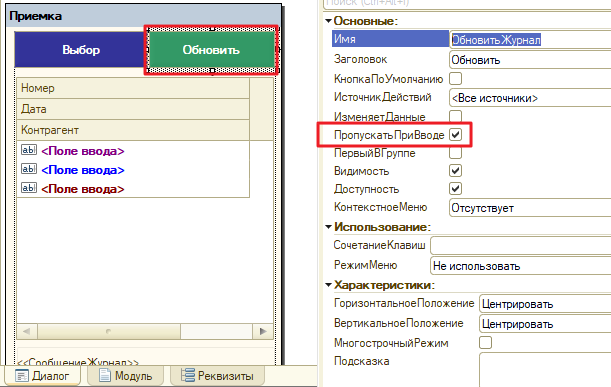
Для загрузки сохраненных данных используется команда LoadDocument, имеет следующий формат:

**LoadDocument;Уникальный набор символов;**

Для удаления файлов используется команда DeleteDocumentBackup, имеет следующий формат:

**DeleteDocumentBackup;Уникальный набор символов;**

**2.9.2. Настройка «Пропускать при вводе»**



Если у элемента формы установлена настройка «ПропускатьПриВводе», то значение данного элемента не будет передаваться на сервер.

Исключения:­

* Табличное поле
* Поле списка
* Поле выбора

**2.9.3. Зарезервированное слово Null**

Бывают ситуации, когда нужно изменить видимость/доступность, при этом нет необходимости изменять значение элемента. В данной ситуации можно использовать зарезервированное слово «Null». В строку передаваемых данных вместо значения элемента необходимо указать «Null».

**2.9.4**. **Управление экранной клавиатурой**

Для того чтобы скрыть клавиатуру, нужно использовать команду:

**hideSoftInput;;**

для того, что бы открыть клавиатуру, нужно использовать команду:

**showSoftInput;ПолеВвода1;**

где нужно указать наименование элемента, для которого нужно открыть клавиатуру.

**2.9.5. Команда «RepeatCall»**

Команда предназначена для вызова любого события указанного в параметре команды.

**RepeatCall;ИмяСобытия;**

**2.9.6. Команда «SUCCESS»**

Так же, для того чтобы не менять элементы формы, можно возвратить com-массив с элементом «SUCCESS,»

**2.9.7. Команда «ShowToast»**

Вывод всплывающего сообщения.



**2.9.8. Команда «PlayErrorSound»**

Воспроизведение звука уведомления (настраивается в настройках устройства).



**2.9.9. Команда «PlayBeep»**

Воспроизведение звука beep



**2.9.9.1 Команда «AttachIdleHandler»**

Обработчик ожидания. В параметрах команды указывается событие и время в секундах через которое сработает событие.



В данном примере событие «ВыполнитьОбработчикОжидания» сработает через 5 секунд.

**2.9.9.2 Команда «setEventAsync»**

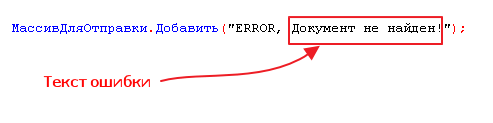
По умолчанию все события работают синхронно с блокировкой интерфейса, для перевода отдельно взятого события в режим асинхронной работы необходимо использовать команду «setEventAsync». Команду необходимо выполнять при открытии формы в событии «Load».



#### 2.10 Диалоговые окна

**2.10.1. Диалоговое окно с ошибкой.**

Для того что бы отобразить ошибку на ТСД в виде предупреждения, необходимо возвратить один элемент com-массива в следующем формате –«ERROR,Текст ошибки».



**2.10.2. Диалоговое окно с вопросом.**

Если необходимо вызвать «Диалог вопроса», то необходимо в возвращаемый com-массив добавить элемент -«DoQueryBox; Текст вопроса;», на ТСД, после нажатия «Yes/No» отработает событие и отдельные параметры будет иметь следующие данные:

* Событие – наименование предыдущего события.
* МассивСКонтролами – все элементы формы плюс «DoQueryBox» со значением «1», если была нажата клавиша «Yes», значение «0», если была нажата клавиша «No».
* Данные – штрих-код, если предыдущие событие было «Scan».

**2.10.3. Диалоговое окно с сообщением.**

Для того чтобы вывести окно с информационным сообщением , необходимо возвратить один элемент com-массива в следующем формате –«ShowMessage;Текст сообщения».

#### ****2.11 Сканирование штрих-кодов****

https://allegrosoft.ru/storage/app/media/uploaded-files/ic_launcher.png

**2.11.1 Сканирование камерой устройства**

Сканирование штрих-кодов производиться с помощью камеры устройства, для этого нужно установить приложение Zxing для сканирования штрих-кодов с телефона. Запуск операции сканирования происходит при нажатии на кнопку с наименованием «Scan». При сканировании штрих-кода срабатывает событие «Scan», которое необходимо обработать на стороне 1С. Штрих-код передается с входящим параметром «Данные».

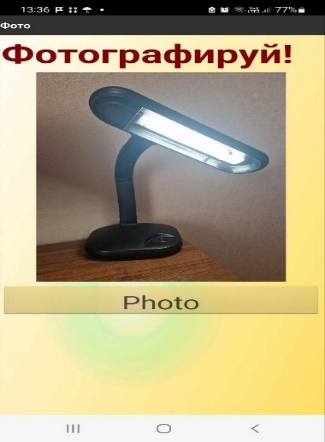
**2.11.2 Сканирование в разрыв клавиатуры**

Для терминалов сбора данных необходимо перевести сканер в режим эмуляции клавиатуры. В настройках сканера добавить суффикс «Enter». Сканирование необходимо производить в поле ввода на форме, т.е. фокус при сканировании всегда должен быть в поле ввода.  **Важно!** Наименование поля ввода должно быть «BARCODEBOX». На стороне 1С будет срабатывать событие «BARCODEBOX».

**2.11.3 Сканирование в режиме BroadCast**

Терминал необходимо перевести в режим BroadCast. При сканировании штрих-кода срабатывает событие «Scan», которое необходимо обработать на стороне 1С. Настройка режима BroadCast для различных моделей ТСД подробно описана **в разделе 4**.

### 2.12 Создание фотографий



Запуск камеры устройства для создания фотографий происходит при нажатии на кнопку с наименованием «Photo». При фотографировании срабатывает событие «Photo», которое необходимо обработать на стороне 1С. Сфотографированное изображение передается с входящим параметром «Данные» в формате Base64. Преобразовать полученные данные можно следующим образом:

ИзображениеНоменклатуры = Новый Картинка(Base64Значение(Данные));

По умолчанию создается изображение в разрешении 600х800. Командой «**resizePhoto**» можно изменить размер фотографии, принимаемые значения 1-150 процентов. Пример:

МассивДляОтправки.Добавить("resizePhoto;150;");

В данном случае фотография будет увеличена в 1,5 раза.

Так же, можно сделать поворот фотографии командой «**rotatePhoto**», принимаемые значения 1-360 градусов. Пример:

МассивДляОтправки.Добавить("rotatePhoto;90;");

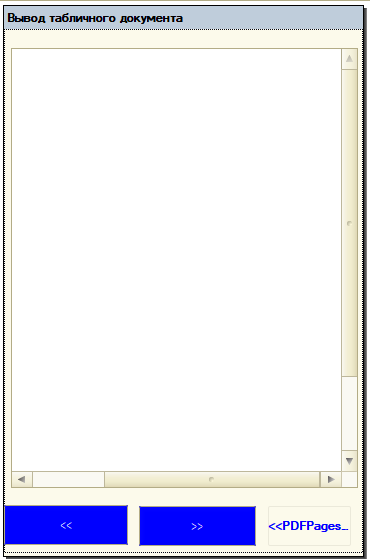
В данном случае фотография будет повернута на 90 градусов.

Данные команды лучше использовать в событии «Load», чтобы при открытии операции задать необходимые значения.

### 2.13 Вывод табличного документа на экране устройства.

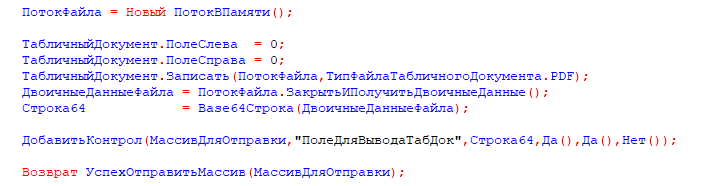
Для вывода табличного документа на форму устройства необходимо подготовить форму. На форме необходимо разместить следующие элементы:

* Поле табличного документа (для отображения табличного документа). Присвоим имя «ПолеДляВыводаТабДок».
* Кнопку с именем «NextPdfPage» для пролистывания на следующую страницу.
* Кнопку с именем «LastPdfPage» для пролистывания на предыдущую страницу.
* Надпись с именем «PDFPagesInfo» для отображения номера текущей страницы



В коде 1С табличный документ сохраняется в поток памяти с типом файла табличного документа «PDF». Поток в памяти преобразуется в формат Base64. Полученная строка отправляется на клиент.

Пример кода:



### 2.14 Печать на Bluetooth принтер.

Для печати на Bluetooth принтер используются специальные команды.

**Печать табличного документа на принтер с поддержкой языка ZPL**

Команда «PrintPdfFileZPL» позволяет отправить на печать Табличный документ.

Формат команды:

МассивДляОтправки.Добавить("PrintPdfFileZPL;ТабличныйДокументВBase64;ШиринаИзображения;ВысотаИзображения;КоличествоКопий;");

Описание параметров команды:

ТабличныйДокументВBase64 - Табличный документ переведенный в формат Base64, пример преобразования описан выше в пункте 2.13.

Ширина изображения – ширина изображения на печать, получение данного значения будет описано ниже.

Высота изображения – высота изображения на печать, получение данного значения будет описано ниже.

КоличествоКопий – количество копий на печать.

**Печать табличного документа на принтер с поддержкой языка TSPL**

Команда «PrintPdfFileTSPL» позволяет отправить на печать Табличный документ.

Формат команды:

МассивДляОтправки.Добавить("PrintPdfFileTSPL;ТабличныйДокументВBase64;ШиринаИзображения;ВысотаИзображения;КоличествоКопий;");

Описание параметров команды:

ТабличныйДокументВBase64 - Табличный документ переведенный в формат Base64, пример преобразования описан выше в пункте 2.13.

Ширина изображения – ширина изображения на печать, получение данного значения будет описано ниже.

Высота изображения – высота изображения на печать, получение данного значения будет описано ниже.

КоличествоКопий – количество копий на печать.

**Печать с использованием языка TSP или ZPL**

Команда «PrintFormattedTextTsplZpl»

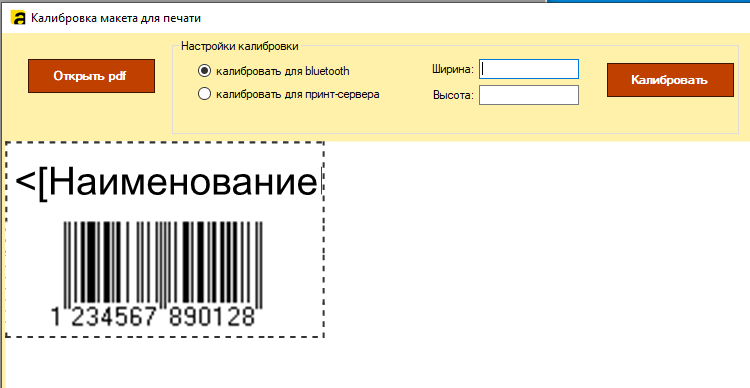
Формат команды:

МассивДляОтправки.Добавить("PrintFormattedTextTsplZpl;КодНаЯзыкеTsplZolВBase64);

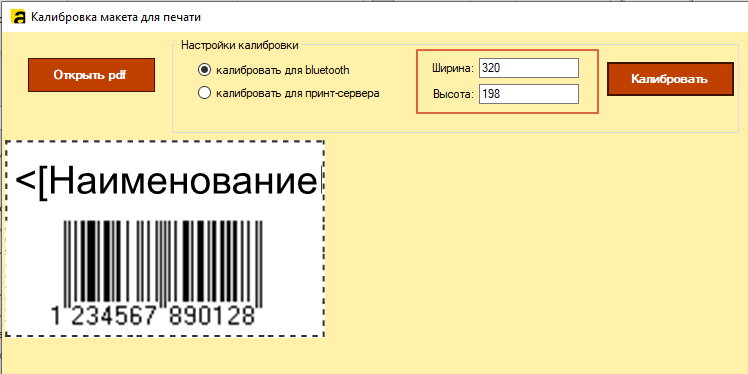
КодНаЯзыкеTsplZplВBase64 – листинг кода на языке ZPL илиTSPL переведенный в Base64.

**Определение высоты и ширины изображения на печать.**

Предварительно необходимо сохранить табличный документ в pdf файл. Открыть Эмулятор ТСД, в эмуляторе открыть сохраненный pdf файл через Сервис -> калибровка макета печати.



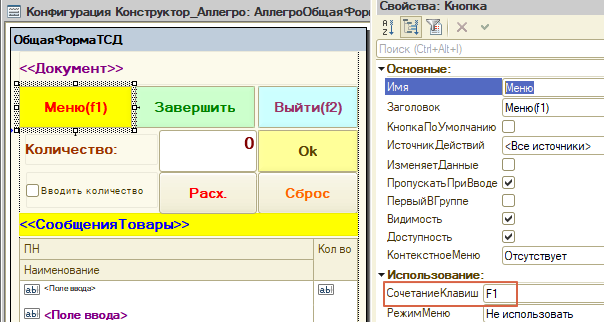
Выставить ширину и высоту по размеру этикетки.



Полученные значения ширины и высоты использовать в команде печати.

#### 2.15 Назначение кнопкам аппаратных клавиш

Аппаратная клавиша устанавливается в свойствах элемента «Кнопка» в пункте «Сочетание клавиш».



Можно назначить кнопкам цифры 1-9, буквы, функциональные клавиши F1 -F12, стрелки.

Есть особенности для стрелок:

Стрелка вверх – Num 8

Стрелка вправо – Num 6

Стрелка влево – Num 4

Стрелка вниз – Num 2

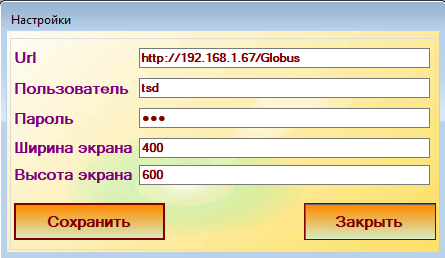
#### 2.15 Эмулятор ТСД

**Эмулятор ТСД –**приложение, эмулирующее работу приложения на терминале сбора данных. Программа упрощает отладку конфигурируемого приложения, и позволяет настраивать логику приложения без использования ТСД.

**Важно!** Эмулятор можно использовать только для разработки в режиме онлайн!

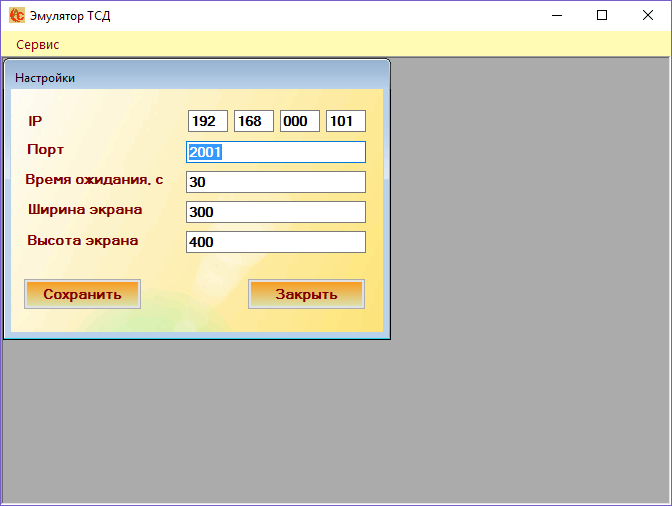
Настройки эмулятора содержатся в выпадающем меню «Сервис».  Настройки имеют следующий вид:

Для веб соединения.



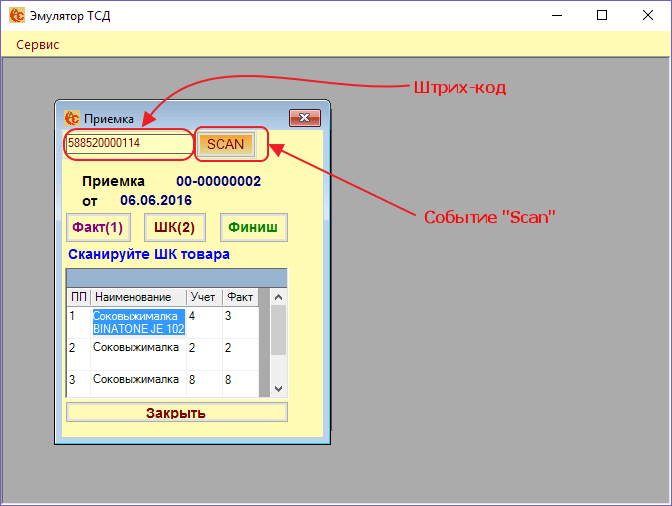
Как настроить описано в пункте 1.1.5.

Для tcp соединения.



Как настроить описано в пункте 1.3.4.

Принцип работы с эмулятором аналогичен работе на клиенте (ТСД), который  будет описан в дальнейшем, за исключением операции сканирования, которая обрабатывается при нажатии на кнопку “Scan”.  Штрих-код (набор символов)  берется из поля ввода,  которое находится  рядом с кнопкой, обрабатывающей событие сканирования.



## Конфигурирование в режиме офлайн

## 3.1 Режим офлайн общие принципы

**3.1.1. Общее описание выполнения кода на клиенте.**

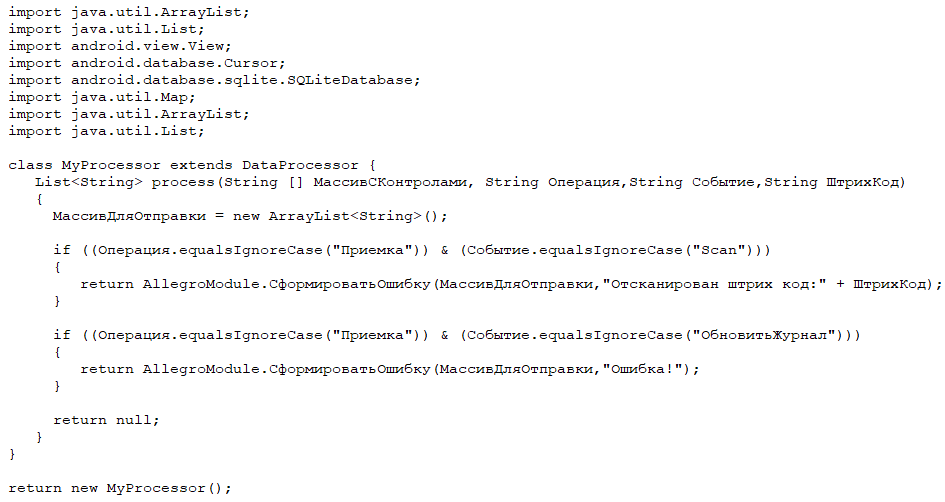
## Для чего необходим режим офлайн на ТСД?

Нередки случаи, когда wi-fi сеть не покрывает полностью рабочую зону работу с терминалом сбора данных (ТСД). В таких случаях используется режим работы с устройством — Оффлайн, когда код выполняется в клиентском приложении без выполнения онлайн запросов в учетную систему.

Для исполнения кода на клиенте в ПО AllegroClient под андроид используется JavaScript. Принцип работы аналогичен работе в 1С.

Предварительно необходимо сформировать листинг с JavaScript с исполняемым кодом для обработки событий. Для обработки событий в режиме офлайн используется функция, на входе у которой  массив строк с элементами формы через разделитель «;» (переменная массива строк называется МассивСКонтролами). В теле функции необходимо сформировать массив строк с элементами формы и вернуть для обработки в клиентское приложение.

Пример функции (рассматривается 1 форма и 2 события):



Параметры :

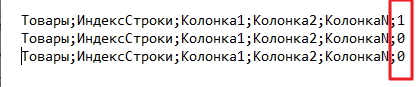
1. МассивСКонтролами — Массив строк с элементами формы

2. Операция — Наименование формы(операции)

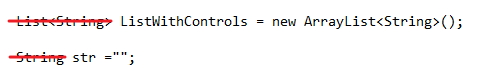
3. Событие — Наименование события

4. ШтрихКод — Штрих-код  (только при событии Scan)

**Важно!** В массив строк добавляются все строки таблиц в отличии при работе с 1С (онлайн режим) и последним символом добавляется 1 или 0 (выделена строка или нет).



Так как используется библиотека BeanShell, строгая типизация переменных не нужна.



По умолчанию все события работают в режиме онлайн, для перевода отдельно взятого события в режим работы офлайн необходимо использовать команду «setEventOffline». Команду необходимо выполнять при открытии формы в событии «Load».

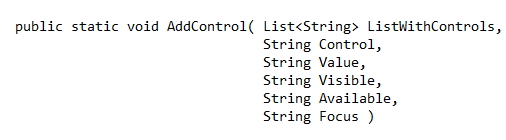
Пример:



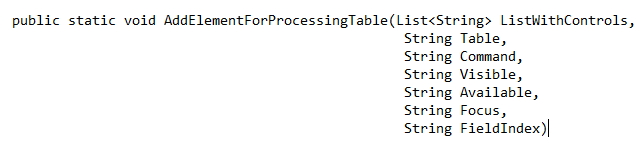
## 3.1.2. Общие процедуры и функции для работы с JavaScript

Для упрощения работы с JavaScript предусмотрены предопределенные процедуры и функции.

**AllegroModule.AddControl (ДобавитьКонтрол)**— процедура для добавления элемента в массив для дальнейшей обработки в приложении.



**AllegroModule.AddElementForProcessingTable (ДобавитьЭлементДляОбработкиТаблицы)** — процедура для добавления строки таблицы в массив для дальнейшей обработки в приложении с использованием команд см. пункт 2.8.7



**AllegroModule.ShowError (СформироватьОшибку)**— функция для формирования ошибки.



**AllegroModule.ShowMessage (СформироватьСообщение)** — выводит диалоговое окно с сообщением.

**AllegroModule.ShowQuestion (СформироватьВопрос)** — выводит диалоговое окно с вопросом.



**AllegroModule.GetListOfControls(ПолучитьМассивСКонтролами)** – копирует массив строк в двухмерный массив, где первый элемент — наименование элемента, второй – значение.

https://allegrosoft.ru/storage/app/media/uploaded-files/getlistofcontrolsjs-1.jpg

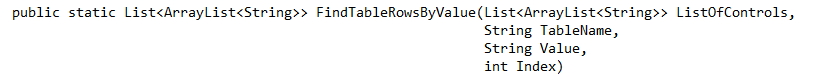
**AllegroModule.GetValueOfControlByName (ПолучитьЗначениеКонтролаПоИмени)** – получить значение элемента по имени.

https://allegrosoft.ru/storage/app/media/uploaded-files/GetValueOfControlByNameJS.jpg

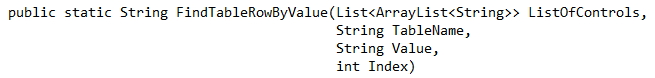
**AllegroModule.GetTableRows(ПолучитьСтрокиТаблицы)** – получение строк таблицы по наименованию таблицы.

https://allegrosoft.ru/storage/app/media/uploaded-files/gettablerowsjs-1.jpg

**AllegroModule.FindTableRowsByValue(НайтиСтрокуТаблицыПоЗначению)** – поиск строк таблицы по значению, возвращает двухмерный массив, нужно указать значение поиска и колонку поиска(Index).



**AllegroModule.FindTableRowByValue(НайтиСтрокуТаблицыПоЗначению )** – поиск строки таблицы по значению, нужно указать значение поиска и колонку поиска(Index), возвращает строку, если найдено или пустую строку, если не найдено.



**AllegroModule.GetValueOfSelectedItemByName(ПолучитьВыделеннуюСтрокуВТаблице)** – возвращает выделенную строку в таблице или пустую строку.

https://allegrosoft.ru/storage/app/media/uploaded-files/getvalueofselecteditembynamejs-1.jpg

**AllegroModule.parseMark (РазобратьКодМаркировки)**

– возвращает массив из 3 элементов, 1 – разобранный код маркировки, 2- разобранный код маркировки со скобками, 3 - gtin. Если не удалось, то null.



**AllegroModule** **isMark (ЭтоКодМаркировки)** – проверяет, что штрих-код является маркировкой .



**AllegroModule.escapeMark (ЭкранироватьМаркировку)** — экранирует символ «;» в коде маркировки.



**AllegroModule.decodeMark (ДекодироватьМаркировку)** — Декодирует символ «;».



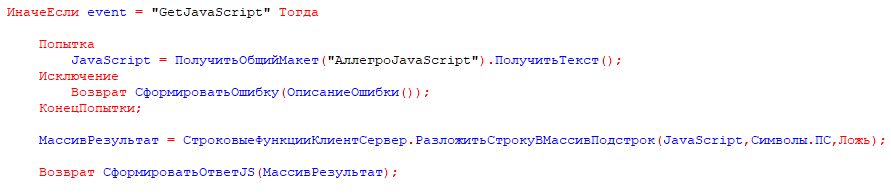
**AllegroModule.GetSubstring (РазбивкаСтроки)** — возвращает подстроку до разделителя «;».



JavaScript можно выполнить для обработчиков событий элементов формы: кнопка и поле ввода». Так же скрипт можно использовать для события сканирования – «Scan». Штрих-код передается в предопределенную переменную «ШтрихКод», значение которой можно использовать в коде.

Для отправки листинга JavaScript в клиентское приложение используется событие «GetJavaScript». Предварительно необходимо сформировать массив строк листинга JavaScript, который будет отправлен в клиентское приложение.

Пример:



В данном примере получаем листинг JavaScript из общего макета, формируем массив строк и отправляем в клиентское приложение.

* + 1. **Глобальная переменная AllegroModule.JSStatus**

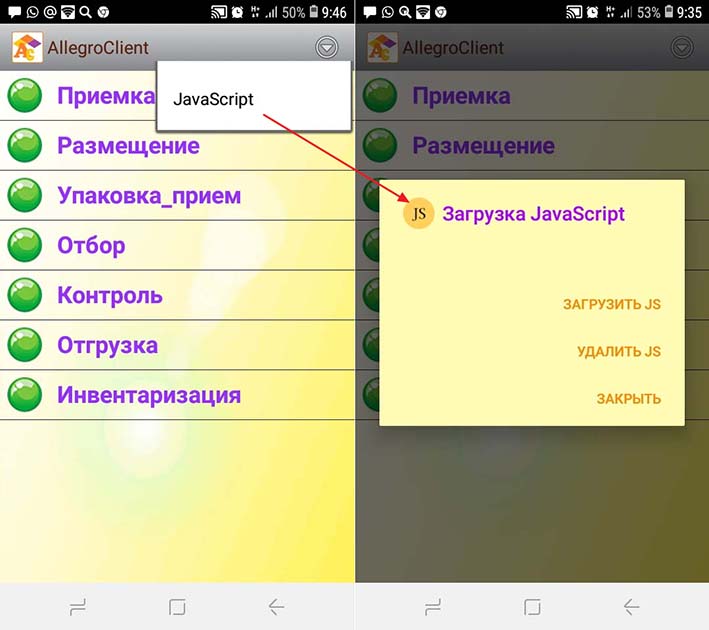
**Изменение значения переменной JSStatus определяет поведение приложения после выполнения JavaScript кода, по умолчанию значение 0 ( обычный режим ).**

**AllegroModule.JSStatus = 1 – после выполнения JavaScript и обновления формы сформированный массив отправляется на сервер.**

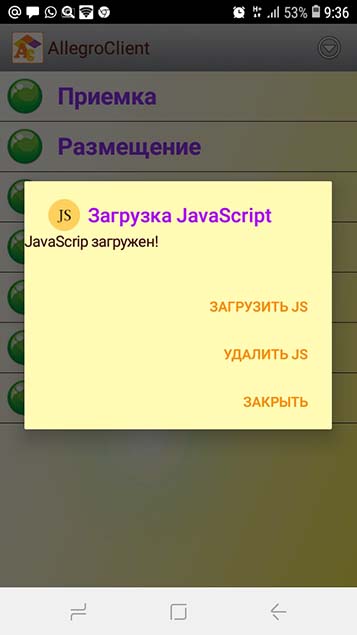
**AllegroModule.JSStatus = 2 – сформированный массив кодом JavaScript сразу отправляется на сервер.**

## 3.1.4. Загрузка JavaScript в клиентское приложение

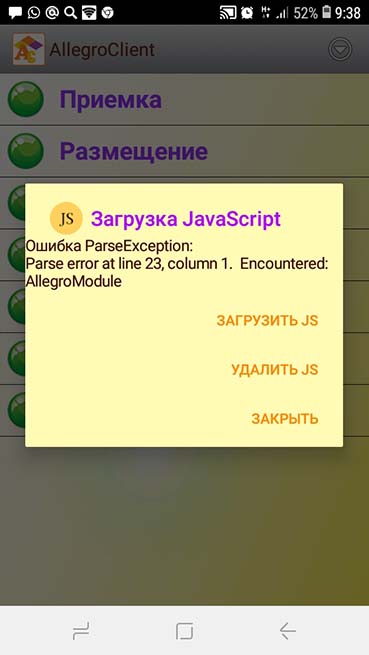
В клиентском приложении открываем пункт меню «JavaScript».



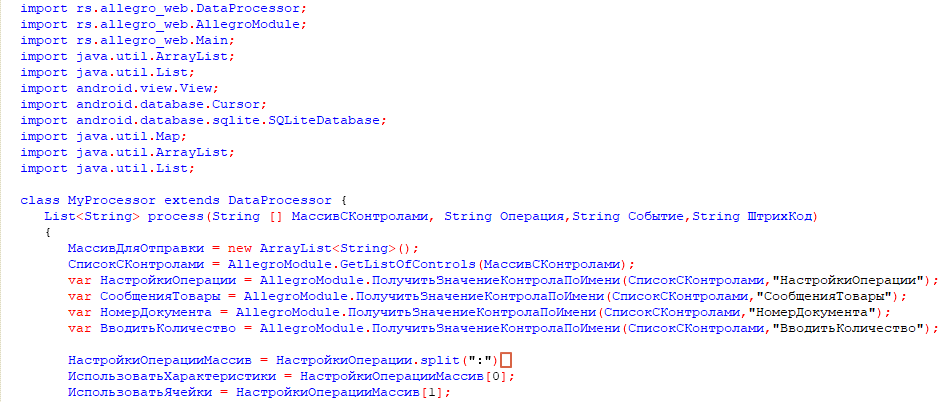
И загружаем JavaScript.



В случае, если  в JavaScript есть ошибка, то при загрузке появится сообщение об ошибке с указанием номера строки ошибки.



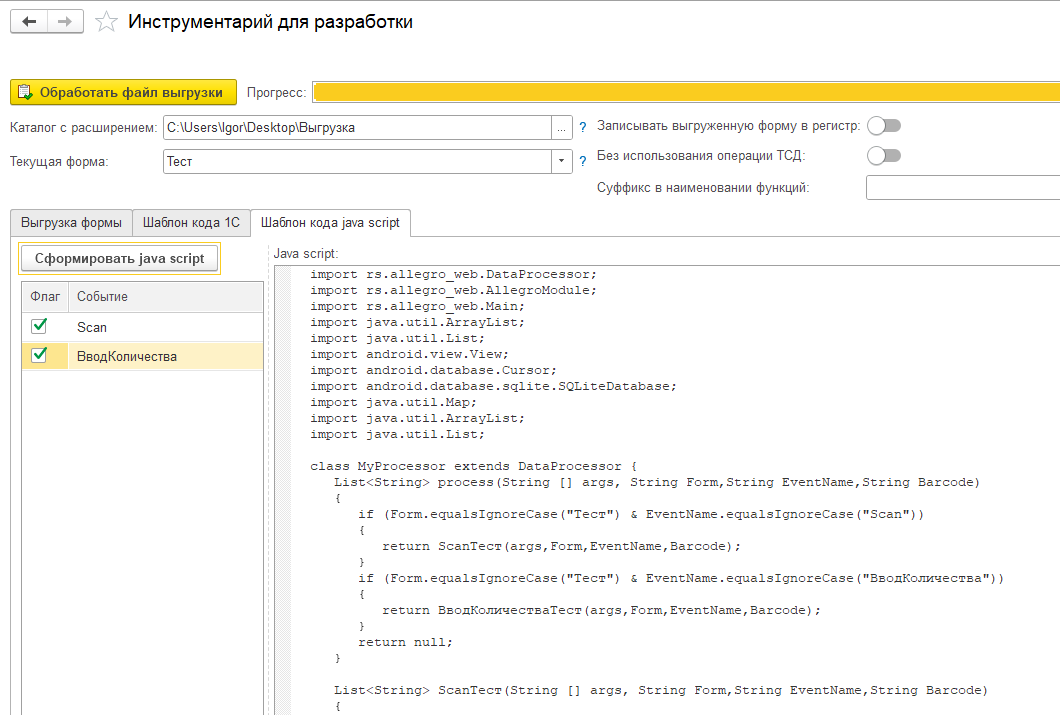
Необходимо посмотреть скрипт в 1С (к примеру, в общем макете) и найти ошибку.



Видим, что пропущена точка с запятой. Исправляем в 1С и выгружаем скрипт заново.

## 3.1.5 Формирование шаблона JavaScript

Для быстрой разработки листинга JavaScript, можно воспользоваться готовым сервисом формирования шаблона JavaScript. Для этого необходимо использовать обработку «Инструментарий для разработки» (выше было описана использование обработки для выгрузки форм). Необходимо проделать действия из пункта 1.2.3 (в), открыть закладку «Шаблон кода JavaScript», выбрать события, которые будут работать в режиме офлайн и сформировать код.



## 3.2 Работа со справочниками в режиме офлайн

## 3.2.1 ****Общее описание работы.****

Для работы со справочниками в ПО AllegroClient под андроид используется база данных SQL ligt.

База данных имеет наименование — «allegroDb». В базе содержится  перечень готовых таблиц  для хранения справочной информации.

**Перечень таблиц и их наименование:**

* Справочник «ШтрихКоды» — «barCodes»
* Справочник «Номенклатура» — «nomenclature»
* Справочник «Контрагенты» — «contractors»
* Справочник «Склады» —  «warehouses»
* Справочник «Ячейки» —  «storageСells»
* Справочник «Короба» —  «boxes»
* Справочник «ТипыЦен» — «priceTypes»

**Структура таблиц и перечень полей таблиц:**

Все таблицы база данных имеют одинаковую структуру.

Перечень полей и наименование: ·

* code – тип String·
* name– тип String·
* attribute1– тип String·
* attribute2– тип String·
* attribute3– тип String

## 3.2.2. ****Загрузка справочников из 1С****

Загрузка справочников в клиентское приложение осуществляется по беспроводной сети wi-fi. Протокол обмена используется такой же, как для режима работы on-line, и логика загрузки реализуется во внешней обработке в общем модуле (там же, где реализована логика работы в режиме on-line) .

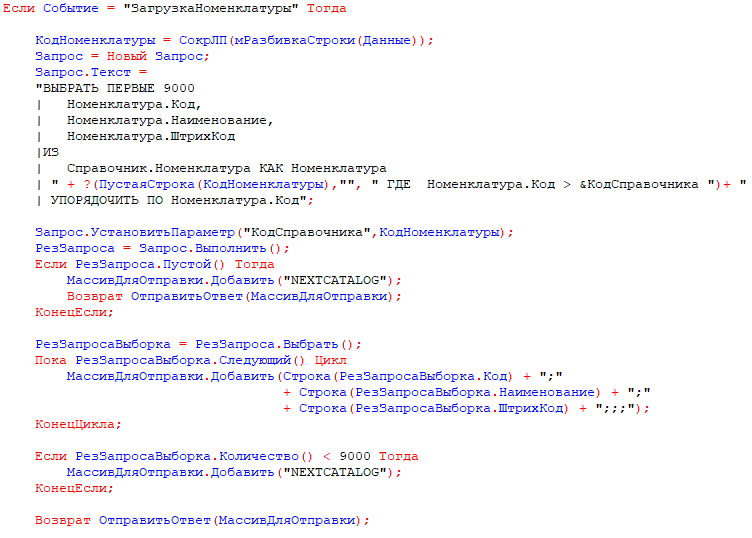
### ****Для загрузки справочников используются следующие события:****

* ЗагрузкаШтрихКодов
* ЗагрузкаНоменклатуры
* ЗагрузкаКонтрагентов
* ЗагрузкаСкладов
* ЗагрузкаЯчеек
* ЗагрузкаКоробов
* ЗагрузкаТиповЦен

Если справочник имеет большой объем, его можно загружать частями.  Далее, в примере использования, будет приведен способ загрузки справочника номенклатуры частями.

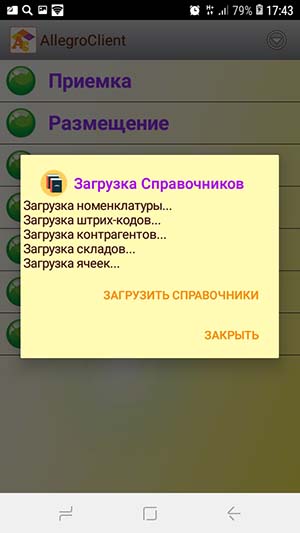
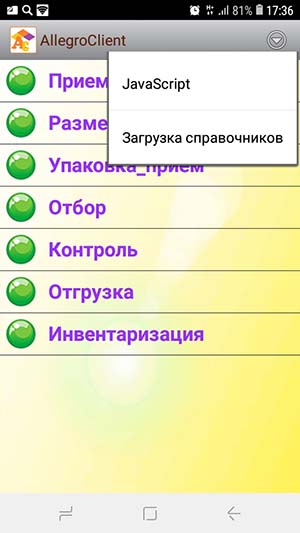
В параметр «Данные» передается последняя загруженная строка справочника  с полями разделенными символом  «;».   Последняя загруженная строка является маркером, опираясь на который,  можно загрузить следующие не загруженные строки справочника.  Для перехода к загрузке следующего справочника (или пропуска загрузки определенного справочника) используется ключевое слово «**NEXTCATALOG**».

Пример использования:



## 3.2.3. ****Загрузка справочников на клиенте****

Для загрузки справочников нужно выбрать пункт меню «Загрузка справочников» и загрузить справочники.



## 3.2.4. ****Работа с данными справочников****

Для работы с таблицами справочников используется JavaScript.

 Для работы с данными таблиц можно использовать произвольный код поддерживаемый JavaScript. Для получения ссылки на базу данных необходимо использовать  статическую переменную  —  **Main**.**dbAllegro**. Для упрощения поиска данных  в справочниках  предусмотрены предопределенные функции.

**AllegroModule.** **GetAllTableRows (ПолучитьВсеСтрокиТаблицы)**

**–**формирует список строк всего справочника, поля в строке разделены символом «;». В параметре необходимо указать наименование таблицы (справочника).

**public** **static** List GetAllTableRows(String TableName)

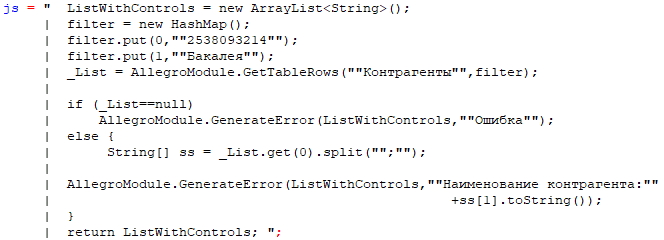
**AllegroModule.** **GetTableRows(ПолучитьСтрокиТаблицы)  –**формирует список найденных  строк  по значению в колонке, поля в строке разделены символом «;». В параметрах необходимо указать наименование таблицы (справочника), номер колонки (нумерация с  0) , по которой нужен поиск и значение поиска.

**public static**List GetTableRows(String TableName,**int**NumberColumn,String \_value)

**AllegroModule.** **GetTableRows(ПолучитьСтрокиТаблицы)  –**формирует список найденных  строк  по значениям в колонках, поля в строке разделены символом «;».В параметрах необходимо указать наименование таблицы (справочника),  и сформировать HashMap (структура) в которой указывается номер колонки(нумерация с  0)   и значение поиска. Таким образом, можно указать множественные значения поиска по нескольким значениям в колонках.

**public static**List GetTableRows(String TableName,  Map filters)

Пример использования:



**Важно!**Параметр TableName может иметь следующие значения:·

* ШтрихКоды (barCodes)
* Номенклатура (nomenclature)
* Контрагенты(contractors)
* Склады(warehouses)
* Ячейки(storageСells)
* Короба(boxes)
* ТипыЦен(priceTypes)

Работа со справочниками представлена в обучающей обработке.

Данный функционал позволяет настроить и автоматизировать с помощью терминала сбора данных работу с любыми данными в 1С.

## 3.3 Работа с документами в режиме офлайн

**3.3.1 Структура таблиц документов**

Для работы с документами в ПО AllegroClient-web под андроид используется база данных SQL lite.

База данных имеет наименование — «allegroDb». В базе содержится перечень готовых таблиц для хранения документов и справочной информации.

### ****Перечень таблиц и их наименование:****

* Основная таблица документа — «table\_doc\_header»
* Табличная часть документа — «table\_doc\_table»

### ****Структура таблиц и перечень полей таблиц:****

Перечень полей и наименование таблицы «table\_doc\_header»:

* «doc\_number» – тип String
* «doc\_type»– тип String
* «doc\_date»– тип String
* «attribute1″– тип String
* «attribute2″– тип String
* «attribute3″– тип String
* «attribute4″– тип String
* «attribute5″– тип String

Перечень полей и наименование таблицы «table\_doc\_table»:

* «doc\_number»– тип String
* «doc\_type»– тип String
* «row\_number»– тип String
* «attribute1″– тип String
* «attribute2″– тип String
* «attribute3″– тип String
* «attribute4″– тип String
* «attribute5″– тип String
* «attribute6″– тип String
* «attribute7″– тип String

## ****3.3.2 Работа с таблицами документов.****

Для работы с таблицами справочников используется JavaScript.

 Для работы с данными таблиц можно использовать произвольный код поддерживаемый JavaScript. Для получения ссылки на базу данных необходимо использовать статическую переменную — **Main.dbAllegro**. Для упрощения поиска, добавления, изменения данных в таблицах предусмотрены предопределенные функции.

**AllegroModule.AddDocumentHeader(ДобавитьРеквизитыДокумента)** — Добавляет значения в таблицу «table\_doc\_header». В параметре необходимо передавать строку со значениями, которые разделены **«;»**.

**public static** void AddDocumentHeader(String values)

**AllegroModule.AddDocumentRows (ДобавитьСтрокиДокумента)**— Добавляет значения в таблицу «table\_doc\_table». В параметре необходимо передавать строку со значениями, которые разделены **«;»**.

**public static** void AddDocumentRows(String values)

**AllegroModule.DeleteDocumentHeader (УдалитьРеквизитыДокумента)** — Удаляет значения из таблицы «table\_doc\_header». В параметрах необходимо передавать номер документа и тип документа.

**public static** void DeleteDocumentHeader(String NumberDoc,String DocType)

**AllegroModule.DeleteDocumentRows(УдалитьСтрокиДокумента)** — Удаляет значения из таблицы «table\_doc\_table». В параметрах необходимо передавать номер документа и тип документа.

**public static** void DeleteDocumentRows(String NumberDoc,String DocType)

**AllegroModule.DeleteDocuments (УдалитьДокументы)** — удаляет все значения из таблиц «table\_doc\_header» и «table\_doc\_table» по типу документа

**public static** void DeleteDocuments(String DocType)

**AllegroModule.GetDocumentHeader (ПолучитьРеквизитыДокумента)**— получить строки из таблицы «table\_doc\_header». В параметрах указывается HashMap (структура) в которой указывается номер колонки(нумерация с 0) и значение поиска. Таким образом, можно указать множественные значения поиска по нескольким значениям в колонках. И так же указывается поле сортировки.

**public static** List GetDocumentHeader(Map filters,String sort)

**AllegroModule.GetDocumentRows (ПолучитьСтрокиДокумента)** — получить строки из таблицы «table\_doc\_table». В параметрах указывается HashMap (структура) в которой указывается номер колонки(нумерация с 0) и значение поиска. Таким образом, можно указать множественные значения поиска по нескольким значениям в колонках. И так же указывается поле сортировки.

**public static** List GetDocumentRows(Map filters,String sort)

**AllegroModule.UpdateDocumentHeader (ОбновитьСтрокиДокумента)** — изменить строки в таблице «table\_doc\_header» (реквизиты документа). В параметрах указывается две HashMap (структуры). В первой структуре указывается номер колонки(нумерация с 0) и значение для вставки в изменяемые строки. Во второй указывается фильтр по строкам, которые будут изменены, первое значение номер колонки(нумерация с 0) и значение отбора.

**public static** boolean UpdateDocumentHeader(Map fields,Map filters)

**AllegroModule.UpdateDocumentHeader (ОбновитьРеквизитыДокумента)** — изменить строки в таблице «table\_doc\_table» (строки документа). В параметрах указывается две HashMap (структуры). В первой структуре указывается номер колонки(нумерация с 0) и значение для вставки в изменяемые строки. Во второй указывается фильтр по строкам, которые будут изменены. Первое значение номер колонки(нумерация с 0) и значение отбора.

**public static** boolean UpdateDocumentRows(Map fields,Map filters)

**3.3.3 Команды для загрузки документов с сервера**

**ADDDOCUMENT** — для загрузки данных в таблицу «table\_doc\_header». Данные для записи идут после команды через разделитель «;»

**«ADDDOCUMENT;000000001;Doc;22.07.2020;Создан;НТЦ Нанотроника»**

**ADDDOCUMENTROW** — для загрузки данных в таблицу «table\_doc\_table». Данные для записи идут после команды через разделитель «;»

**«ADDDOCUMENTROW;000000001;Doc;1;0001;Комбайн MOULINEX A77 4C;2;0»**

**DELETEDOCUMENTHEADER** — удаление данных из таблиц «table\_doc\_header». После команды через разделитель «;» идут данные с номером документа и типом документа.

**«DELETEDOCUMENTHEADER;000000001;Doc»**

**DELETEDOCUMENTROWS** — удаление данных из таблиц «table\_doc\_table». После команды через разделитель «;» идут данные с номером документа и типом документа.

**«DELETEDOCUMENTROWS;000000001;Doc»**

**DELETEDOCUMENT** — удаление данных из таблиц «table\_doc\_header» и «table\_doc\_table». После команды через разделитель «;» идут данные с номером документа и типом документа.

**«DELETEDOCUMENT;000000001;Doc»**

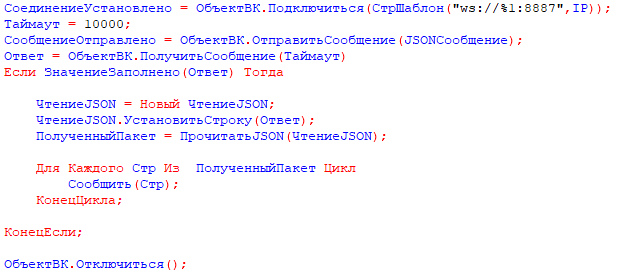
Работа с документами представлена в обучающей обработке в операции «РаботаСДокументамиОффлайн».

## Работа с веб сокетами

Механизм работы с веб сокетами разработан в первую очередь для отправки уведомлений на мобильное устройство. Принцип работы: на мобильном устройстве запускает веб сервер, который слушает сообщения от учетной системы. Веб сервер запускается на порту 8887. Для запуска веб сервера на устройстве при событии **«Identification»** в массиве, который будет отправлен на устройство», в первом элементе отправить значение «1». Для отправки сообщения на мобильное устройство с использованием веб сокетов необходимо сформировать JSON массив, где 1 элемент массива – событие(команда), следующие элементы массива – тело сообщения. Для отправки уведомление на устройство нужно сформировать следующий массив: первый элемент массива - команда «**DisplayNotification**», второй элемент - текст уведомления. Так же можно отправить сообщение для отработки события оффлайн на устройстве. Принцип работы оффлайн был описан выше. Структура массива JSON будет следующая: первый элемент массива – имя события (произвольное ), последующие элементы будут переданы в тело функции для обработки с помощью Java script. Структура параметров которая будет сформирована для передачи в функцию «process» для обработки с помощью javaScript:

* МассивСКонтролами-отправленный массив без первого элемента
* Операция – предопределенное значение «WebSocket»
* Событие – первый элемент переданного массива
* ШтрихКод – пустое значение «»;

В теле функции нужно сформировать массив (обновление элементов формы с использованием массива не реализовано, так как контекст рабочей Activity может быть неактивным, что приведет к ошибке при обновлении элементов ). В 1С будет получен ответ (сформированный массив) в виде JSON массива. Пример отправки сообщения в коде 1С (реализован с помощью внешней компоненты, в последних версиях платформы 1С можно использовать внутренний синтаксис):



Где IP – ip адрес устройства.

## 5 Настройка сканирования на ТСД для работы с AllegroClient

### Общее описание

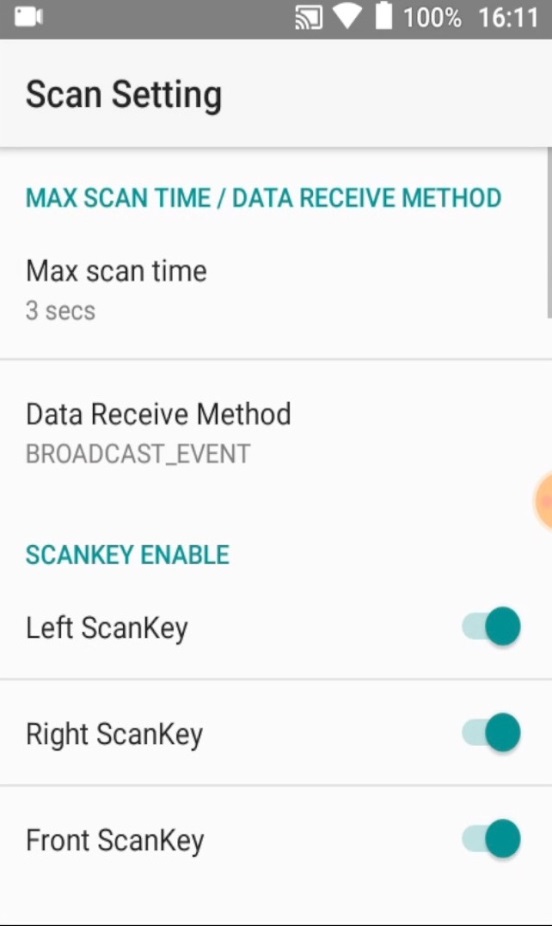
Для корректной работы сканера на терминале сбора данных ( в режиме Broadcast ) с программным обеспечением AllegroCount и AllegroClient необходимо произвести насторойку устройства.

Ниже приведены настройки для самых распространенных ТСД:

* ***ATOL Smart Lite и SmartSlim***
* **ТСД*UROVO***
* **ТСД*Honeywell***
* **ТСД*Mertech***
  + 1. **Пример настройки для терминалов сбора данных ATOL серии Smart Lite и Slim**

**

Достаточно настроить один параметр, В поле **Data Receive Method** выбрать **BROADCAST\_EVENT**



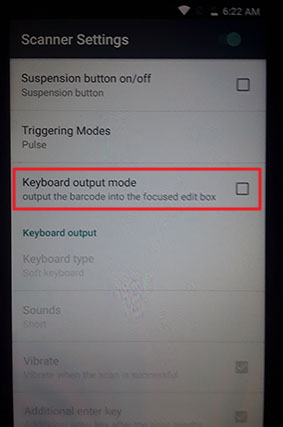
### Инструкция по настройке:

1. Зайти в главное меню и выбрать пункт "Barcode Utility"
2. Далее, Scan Setting
3. В разделе Data Receive Method указать " BROADCAST\_EVENT"

**4.1.2. Пример настройки сканирования для терминала сбора данных Urovo.**

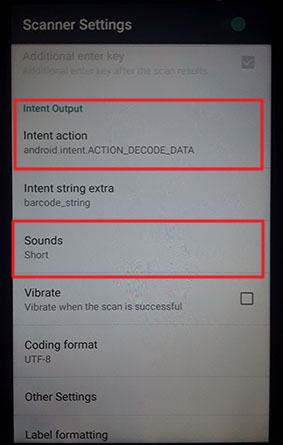


Снять флажок **Keyboard output mode**.



В поле **Intent action** должна быть строка **android.intent.ACTION\_DECODE\_DATA** (стоит по умолчанию).

И выбрать в поле **Sound** звук, который будет звучать при сканировании.

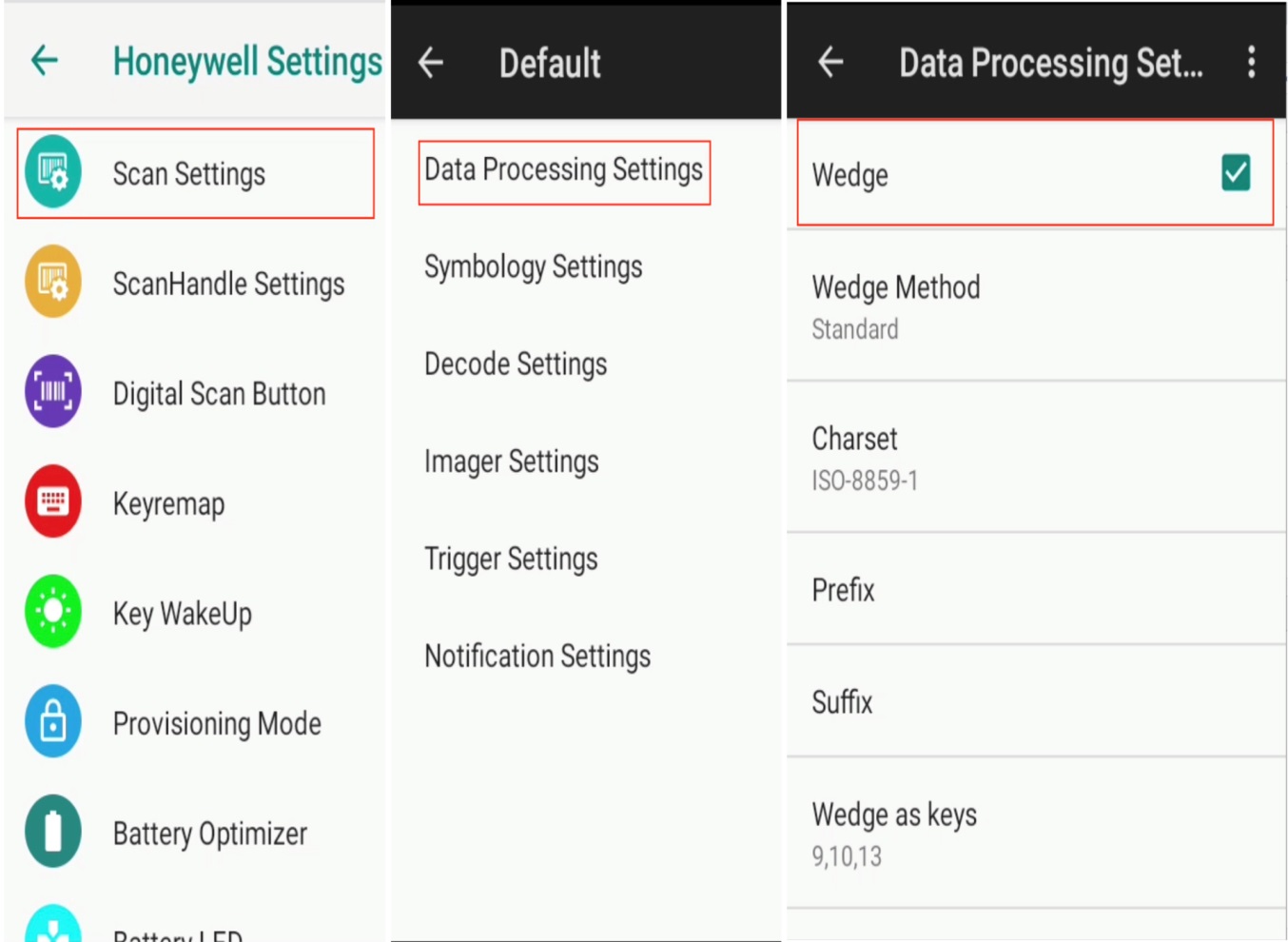


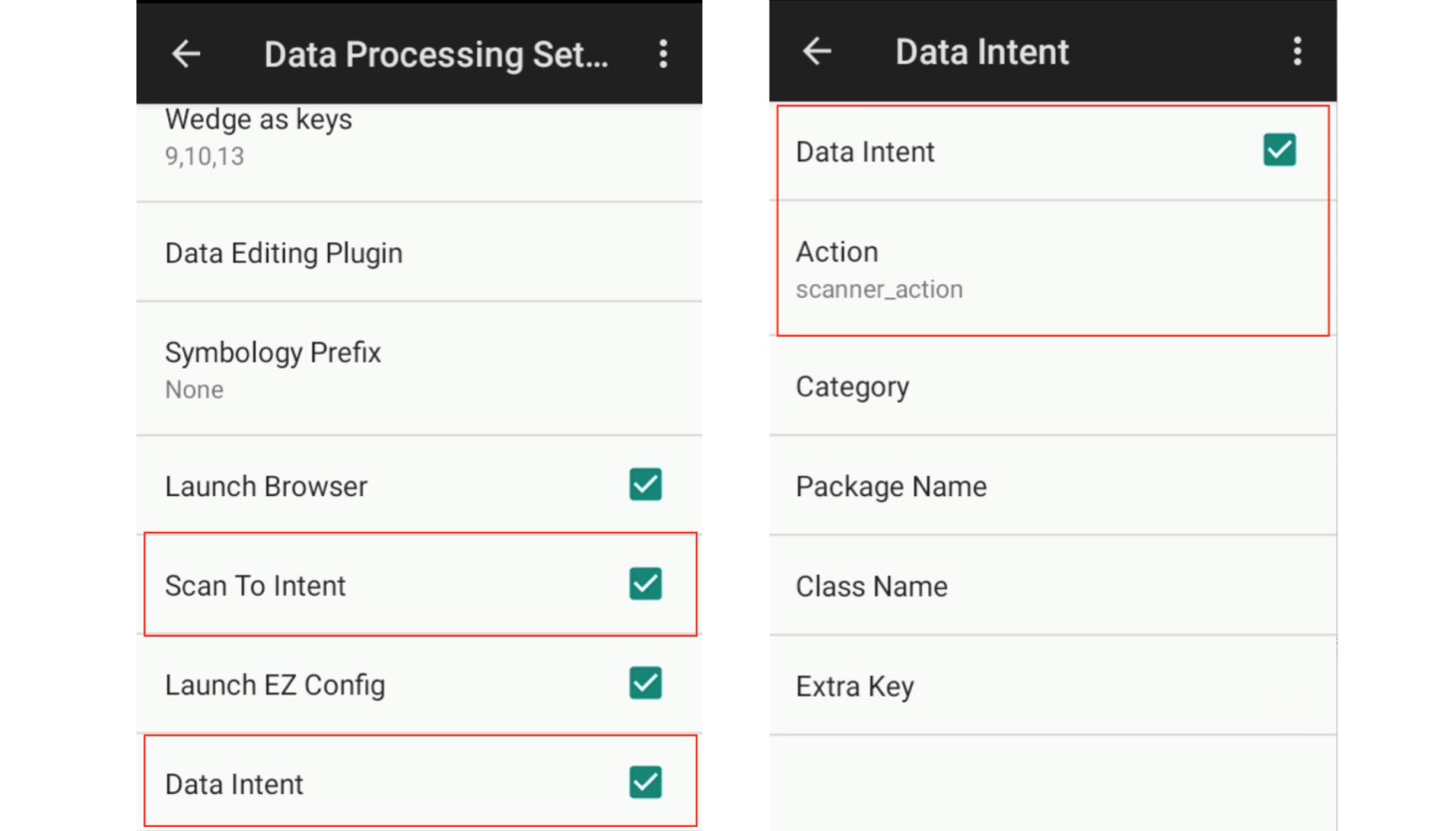
## 4.1.3. Пример настройки сканирования для терминала сбора данных Honeywell.



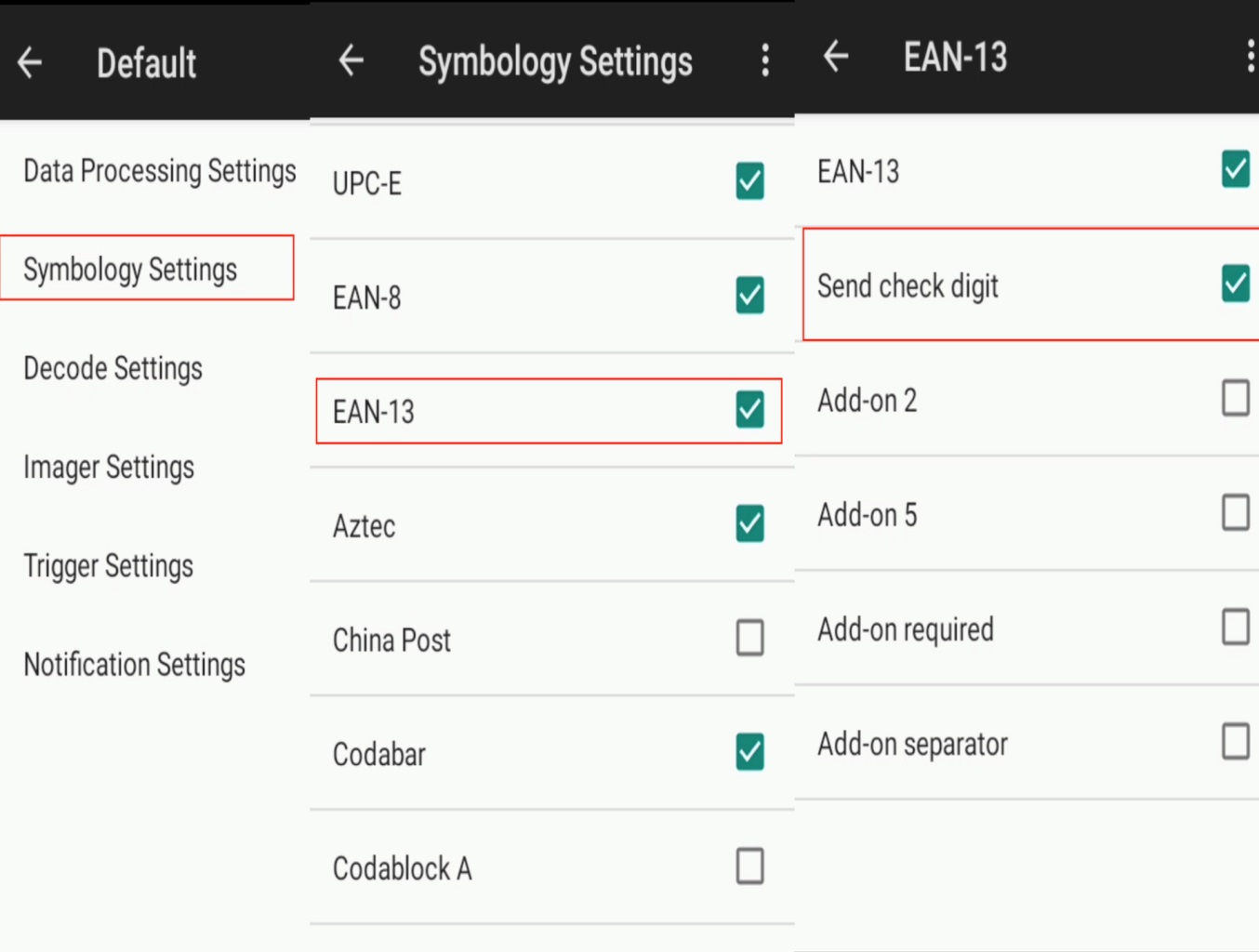
1. Открыть настройки сканирования (Scan Settings) **- Data Processing Settings -**  Активировать **Wedge**- Активировать **Scan To Intent.**

Перейти в меню **Data Intent**и в поле **Action** ввести **"scanner\_action”**



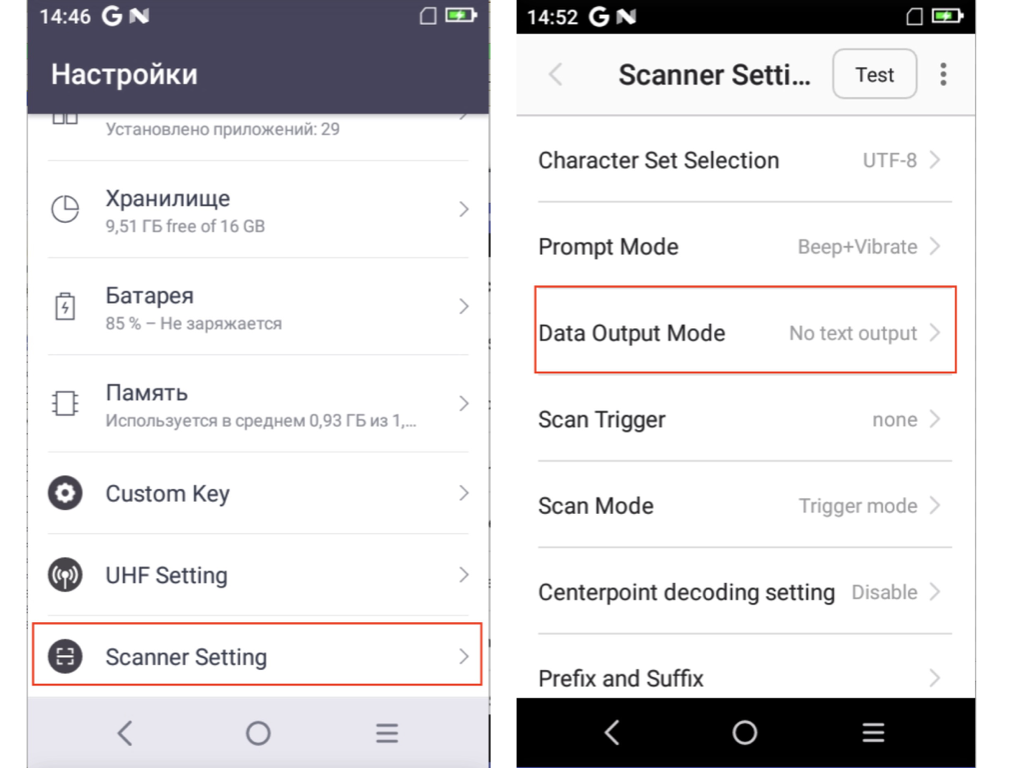


2. Открыть настройки сканирования **(Scan Settings**) - **Symbology Settings** - выбрать необходимый формат кода и в нем активировать  **Send check digit**

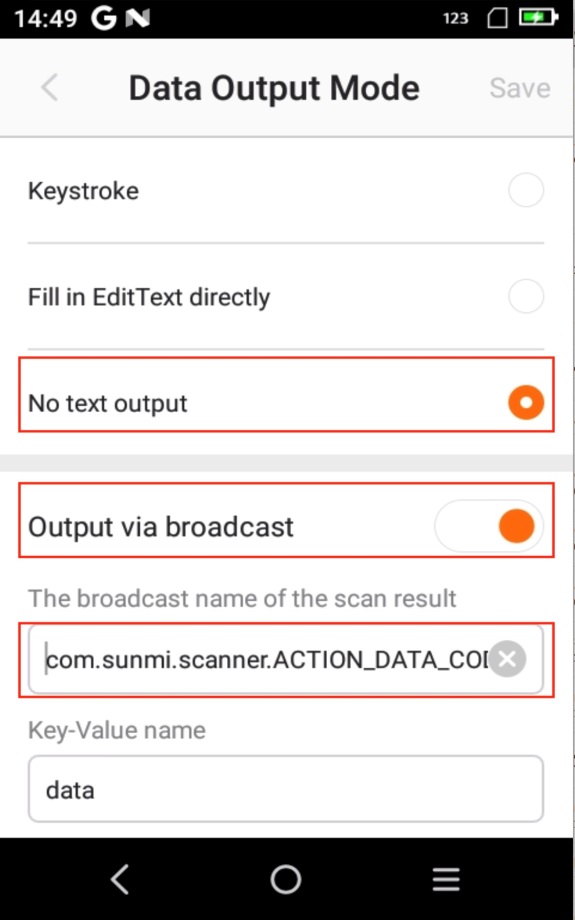
****

**4.1.4. Пример настройки сканирования для терминала сбора данных Mertech.**

Настройки выбрать пункт "Scanner Setting" - Data Output Mode.

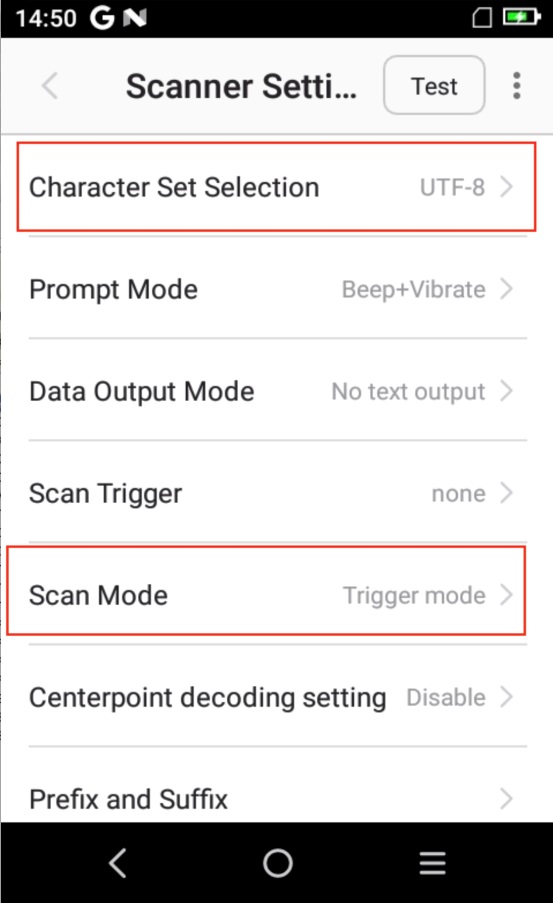


Указать: "No text output", Output via broadcast. Убедиться, что поле "The broadcast name" заполнено "com.sunmi.scanner.ACTION\_DATA\_CODE\_RECEIVED"



Далее, вернуться в меню **Scan Settings** и убедиться, что **Character Set Selection** выбрано **UTF-8.**

**Scan Mode - Trigger mode**

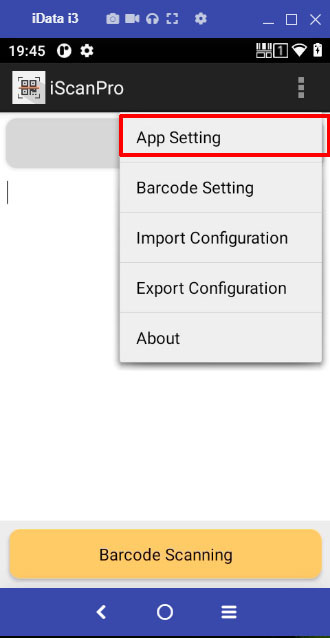
****

**4.1.5. Пример настройки сканирования для терминала сбора данных iData.**

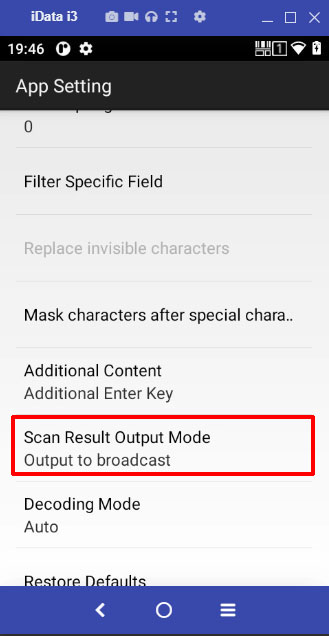
Открыть приложение **iScanPro.**



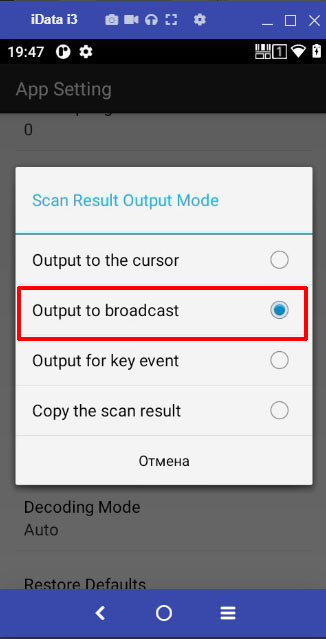
Выбрать **App Setting.**



Выбрать пункт **Scan Result Output Mode.**

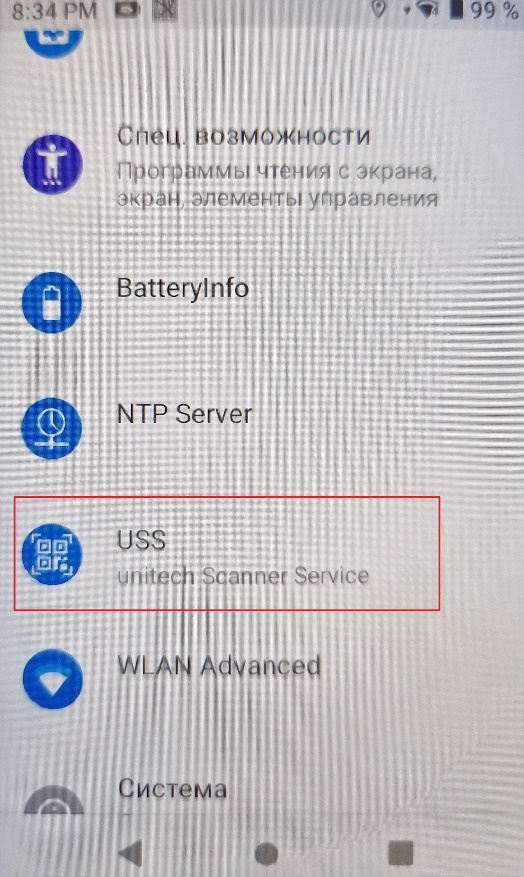


Выбрать **Output to broadcast.**

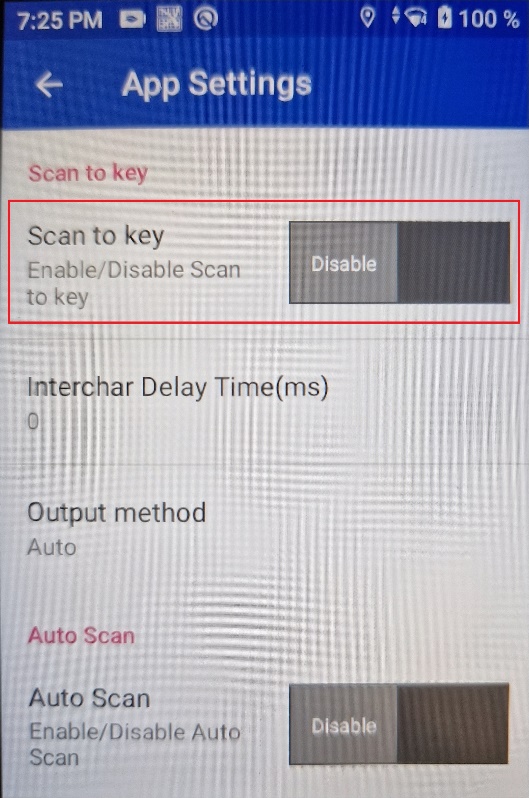


**4.1.6. Пример настройки сканирования для терминала сбора данных *Unitech*.**

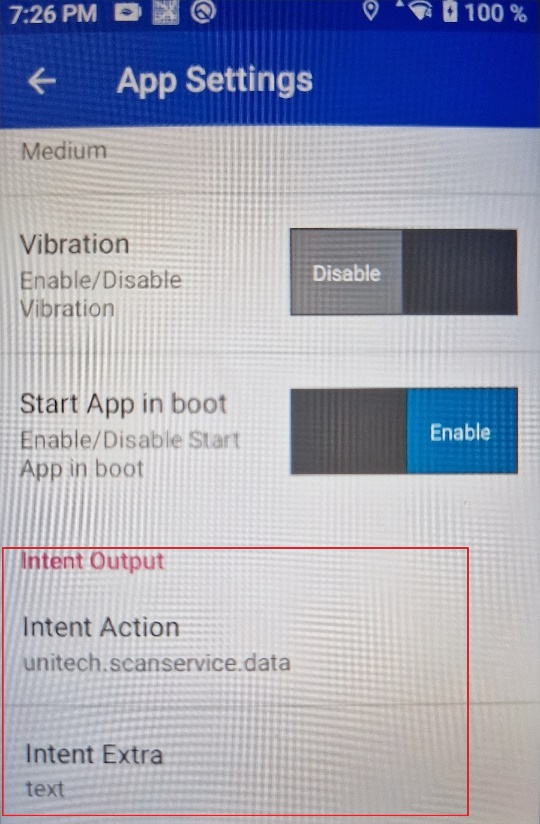
В настройках открываем пункт **UUI**.



Открываем **App Settings**. **Scan to key** устанавливаем Disable.



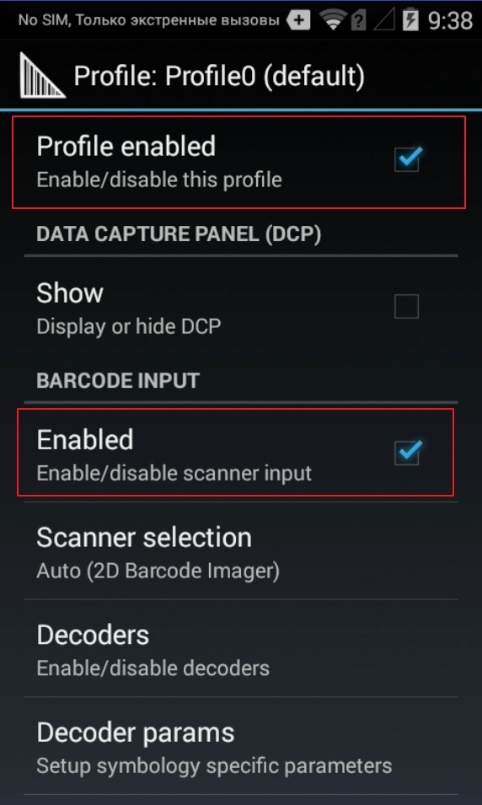
В **Intent Output** должны быть следующие настройки (по умолчанию):

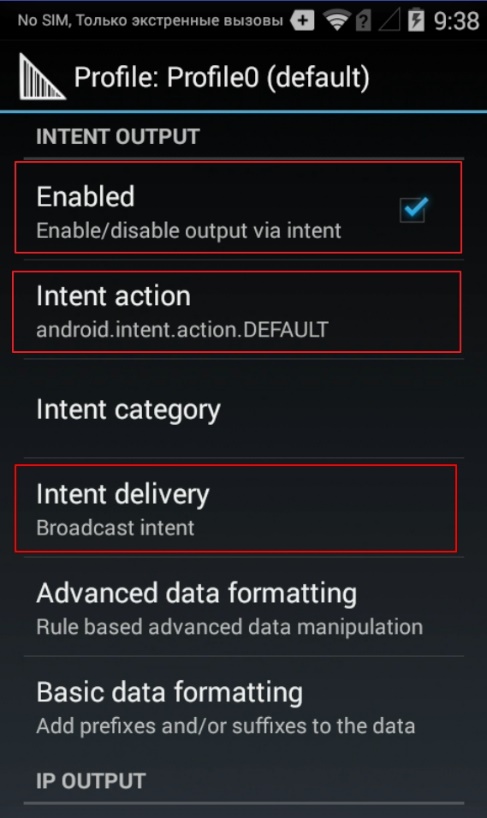


**4.1.7. Пример настройки сканирования для терминала сбора данных *Zebra*.**

Запустите приложение DataWedge. Используйте в списке профилей - Profile0 (профиль по умолчанию, используемый всеми приложениями, которым явно не назначен профиль).

Указать следующие настройки:





**Важно!** В пункте **Intent action** указать android.intent.action.DEFAULT.