



Программное обеспечение AllegroClient. Руководство разработчика.



Содержание

1 Настройка программного обеспечения	5
1.1 Настройка AllegroClient-web расширением 1с для собственной разработки операций	5
1.1.1. Подключение расширения 1с	5
1.1.2. Регистрация HTTP-сервиса 1с	5
1.1.3. Как открыть обработку для настройки	6
1.1.4. Как создать пользователя для подключения к http-сервису	7
1.1.5. Как настроить подключение в клиентском приложении	8
1.2 Настройка AllegroClient – web с 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	11
1.2.5. Настройка в клиентском приложении сервису	11
1.3 Настройка AllegroClient – prof 2.0.....	11
1.3.1. Установка Сервера ТСД	11
1.3.2. Настройка Сервера ТСД	11
1.3.3. Выгрузка Форм	13
1.3.4. Как настроить подключение в клиентском приложении	14
2 Конфигурирование логики работы с операциями в режиме онлайн	16
2.1 Разработка макета форм.....	16
2.1.1. Разработка макетов форм для AllegroClient-prof 2.0 и для AllegroClient- web.....	16
2.1.2. Разработка макетов форм для расширения AllegroCount	16
2.2. Описание серверной функции для обработки событий от клиента	16
2.2.1. Функция для AllegroClient-prof 2.0 и для AllegroClient-web с расширением 1с для собственной разработки операций	16
2.2.2 Функция для расширения AllegroCount для собственной разработки операций	16
2.3 Параметр «Код сотрудника»	16
2.4 Параметр «Событие»	17
2.5 Параметр «Наименование формы».....	17
2.6 Параметр «Данные»	17
2.7 Параметр «Массив с контролами»	17
2.8 Работа с элементами форм	17

2.8.1	Общее описание	18
2.8.2	Элемент управления «Надпись».....	19
2.8.3	Элемент управления «Поле ввода».....	19
2.8.4	Элемент управления «Список»	19
2.8.5	Элемент управления «Список выбора».....	20
2.8.6	Элемент управления «Флажок».....	20
2.8.7	Элемент управления «Табличное поле».....	21
2.8.8	Элемент управления «Кнопка»	23
2.8.9	Элемент управления «Изображение».....	23
2.9	Команды	24
2.9.1	Сохранение данных формы.....	24
2.9.2	Настройка «Пропускать при вводе»	25
2.9.3	Зарезервированное слово Null	26
2.9.4	Управление экранной клавиатурой	26
2.9.5	Команда «RepeatCall»	26
2.9.6	Команда «SUCCESS»	26
2.10	Диалоговые окна	27
2.10.1	Диалоговое окно с ошибкой	27
2.10.2	Диалоговое окно с вопросом	27
2.10.3	Диалоговое окно с сообщением	27
2.11	Сканирование штрих-кодов	27
2.11.1	Сканирование камерой устройства	27
2.11.2	Сканирование в разрыв клавиатуры	27
2.11.3	Сканирование в режиме BroadCast	28
2.12	Эмулятор ТСД	28
3	Конфигурирование в режиме офлайн	30
3.1	Режим офлайн общие принципы	30
3.1.1	Общее описание выполнения кода на клиенте	30
3.1.2	Общие процедуры и функции для работы с JavaScript	31
3.1.3	Выгрузка JavaScript в клиентское приложение	34
3.2	Работа со справочниками в режиме офлайн	38

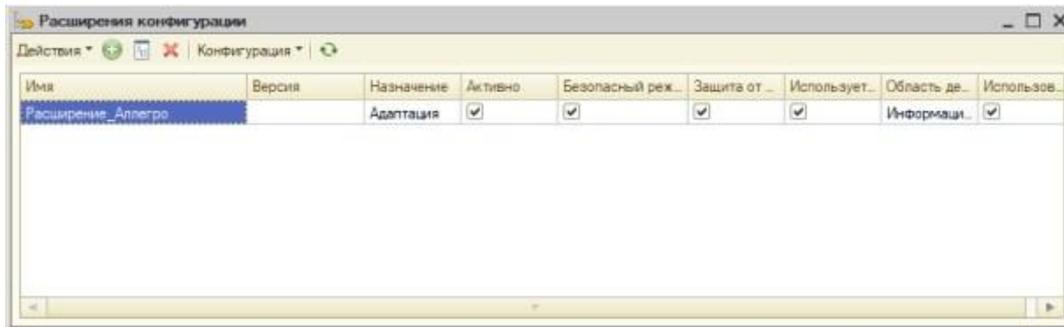
3.2.1	Общее описание работы	38
3.2.2.	Загрузка справочников из 1С	39
3.2.3.	Загрузка справочников на клиенте	40
3.2.4.	Работа с данными	41
3.3	Работа с документами в режиме офлайн	41
3.3.1	Структура таблиц документов	42
3.3.2	Работа с таблицами документов	43
3.3.3	Команды для загрузки документов с севера	45
4	Настройка сканирования на ТСД для работы с AllegroClient	47
4.1	Общее описание	47
4.1.1.	Пример настройки для терминалов сбора данных ATOL серии Smart Lite и Slim	47
4.1.2.	Пример настройки сканирования для терминала сбора данных Urovo	48
4.1.3.	Пример настройки сканирования для терминала сбора данных Honeywell	50
4.1.4.	Пример настройки сканирования для терминала сбора данных Mertech	52
4.1.5.	Пример настройки сканирования для терминала сбора данных iData	54

1. Настройка программного обеспечения

1.1 Настройка AllegroClient – web с расширением 1с для собственной разработки операций

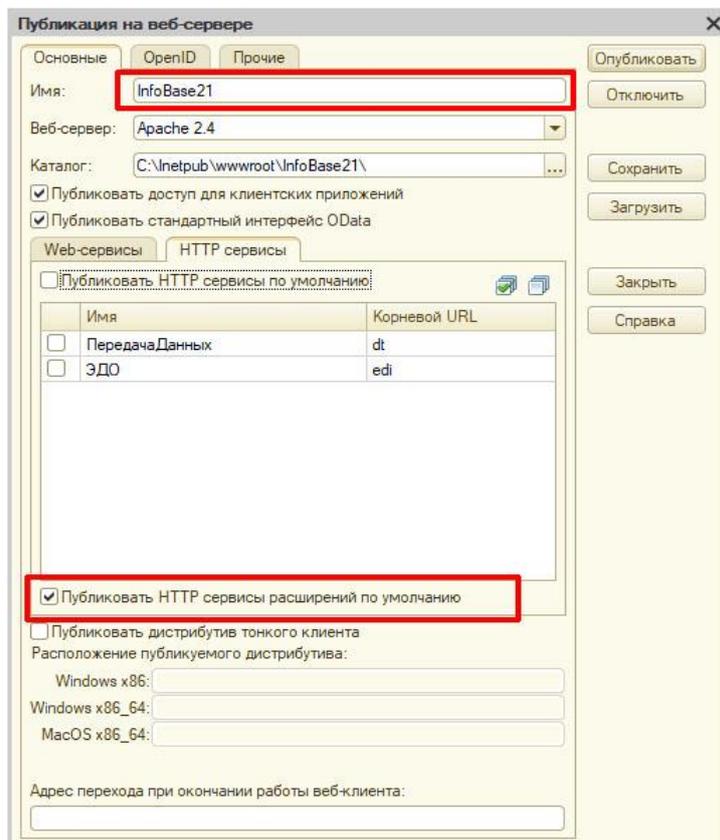
1.1.1. Подключение расширения 1с.

Подключить расширение конфигурации, идущее в поставке следующим образом.



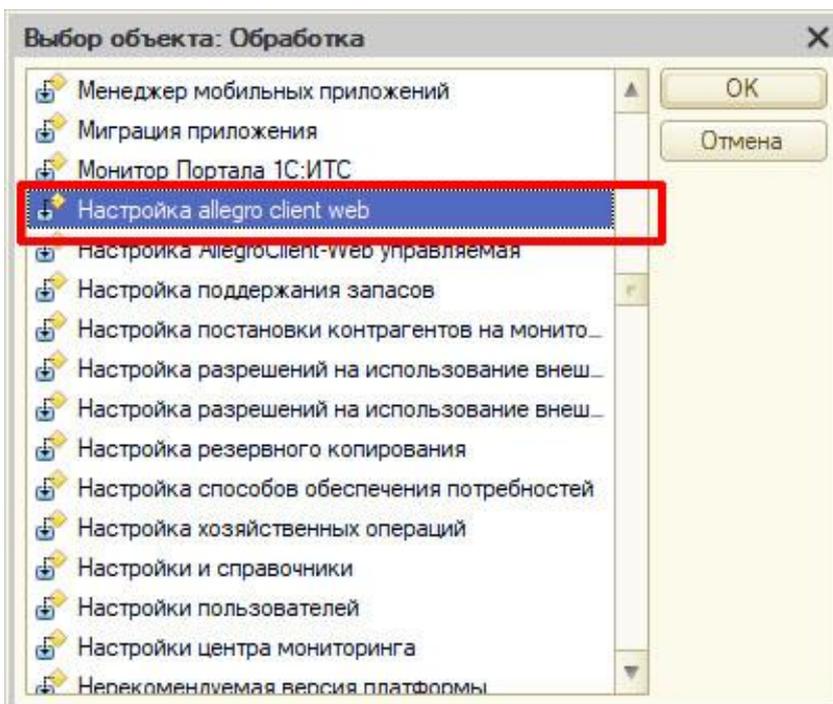
1.1.2. Регистрация HTTP-сервиса.

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.

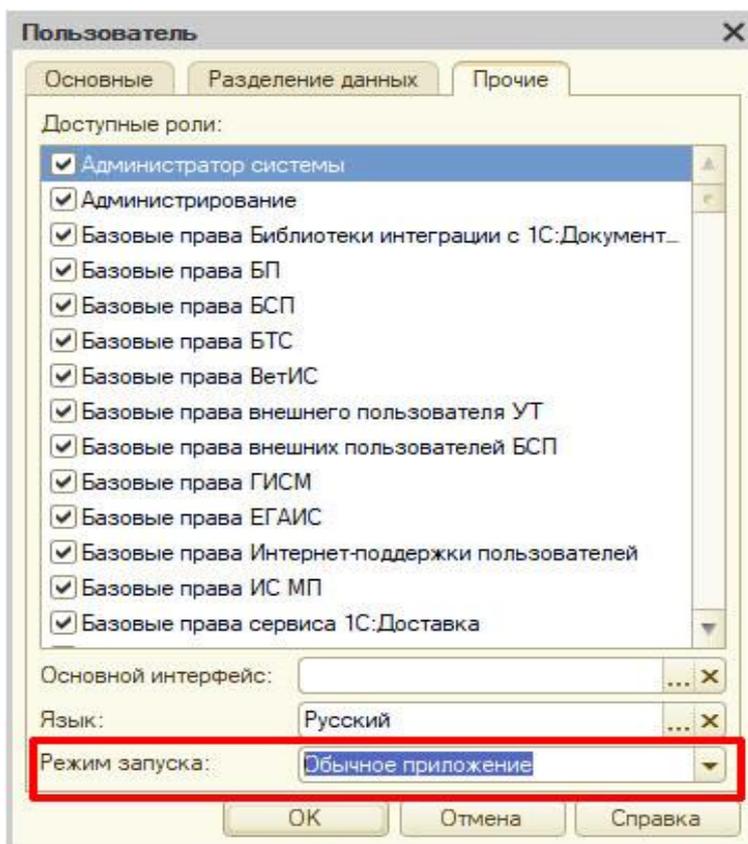


1.1.3. Как открыть обработку для настройки

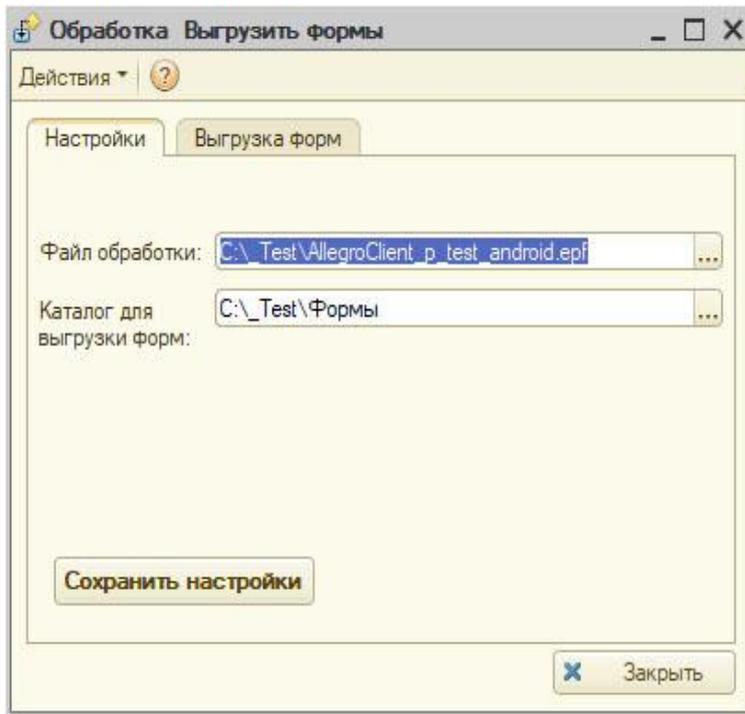
Необходимо открыть 1С в режиме предприятия и открыть обработку «Настройка AllegroClient web».



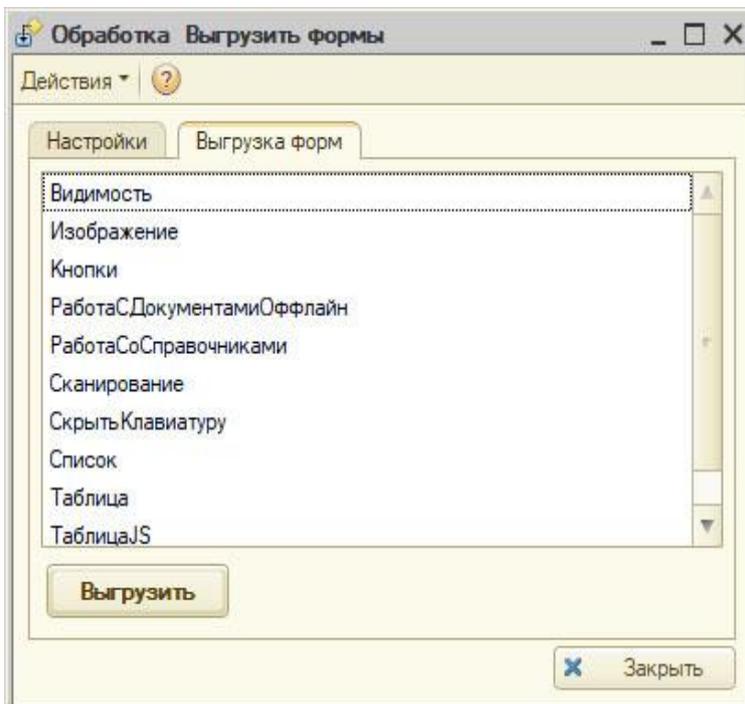
Важно! Систему 1С необходимо открыть обычном в режиме. У пользователя должен стоять режим «Обычное приложение».



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).

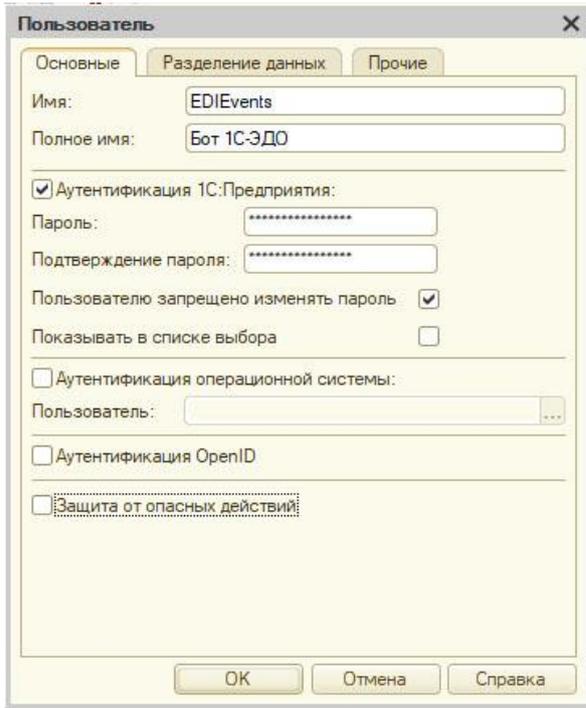


Выгрузить формы. Увидим перечень выгруженных форм.



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

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

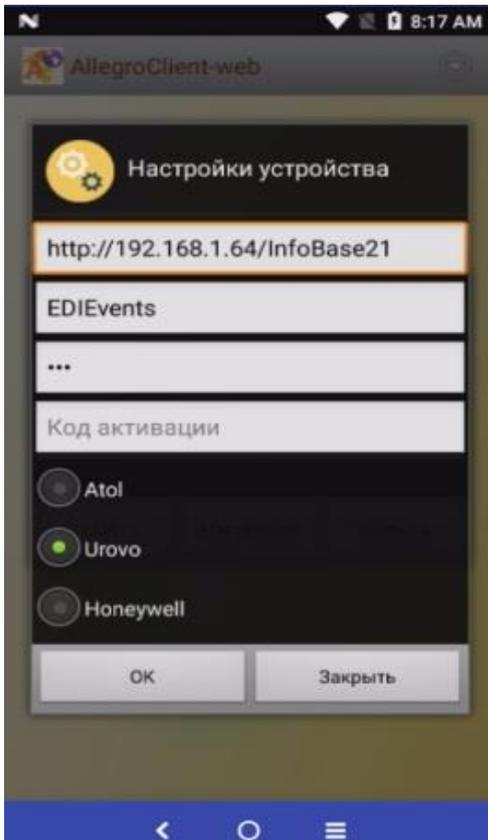


1.1.5. Как настроить подключение в клиентском приложении.

На клиентском приложении заполнить настройки.

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

Пароль - пароль пользователя для подключения



1.2 Настройка AllegroClient – web с расширением AllegroCount

1.2.1. Публикация расширения.

Публикация расширения описано в пункте 1.1.1.

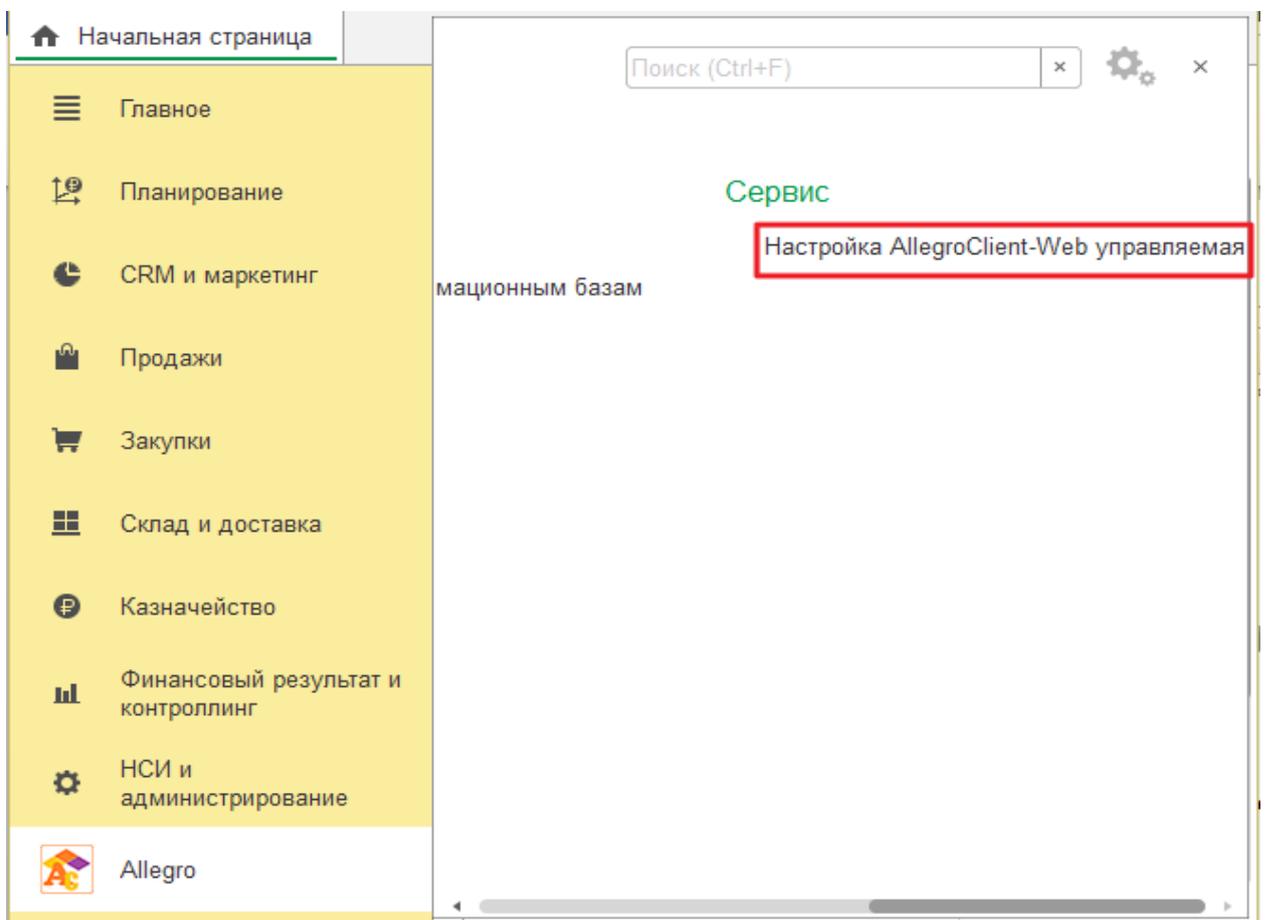
1.2.2. Регистрация HTTP-сервиса.

Регистрация HTTP-сервиса описано в пункте 1.1.2

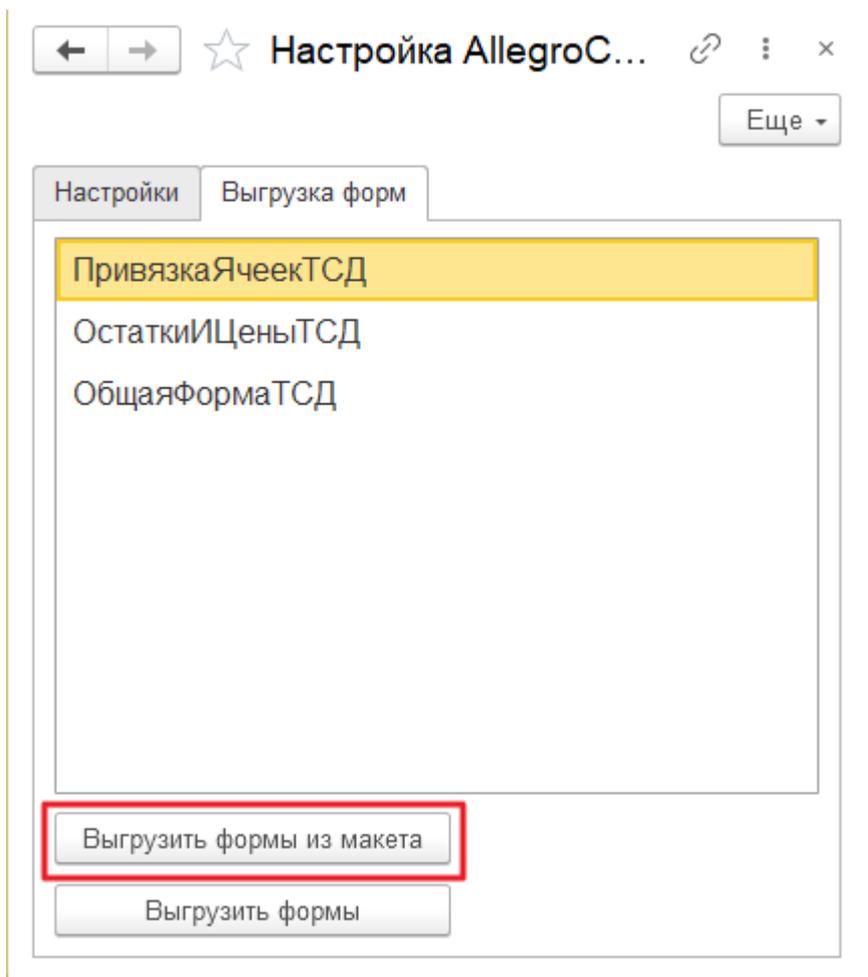
1.2.3. Выгрузка форм.

Выгрузка форм . 2 варианта:

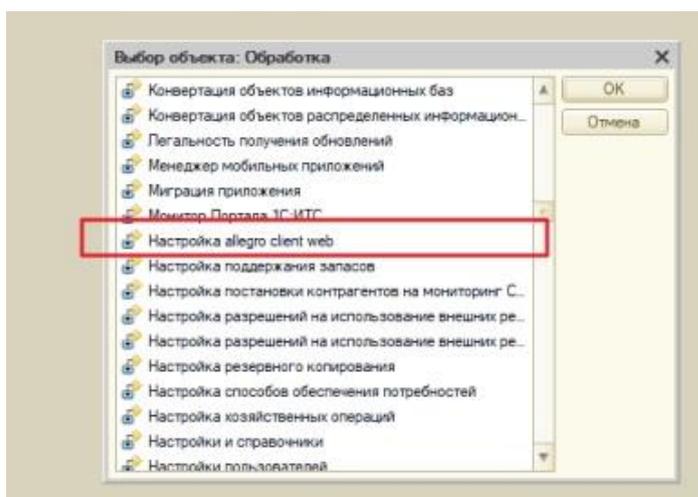
а) Формы не меняем, используем штатные. Открываем обработку в системе Allegro.



В закладке «Выгрузка форм» выгружаем формы из макета.

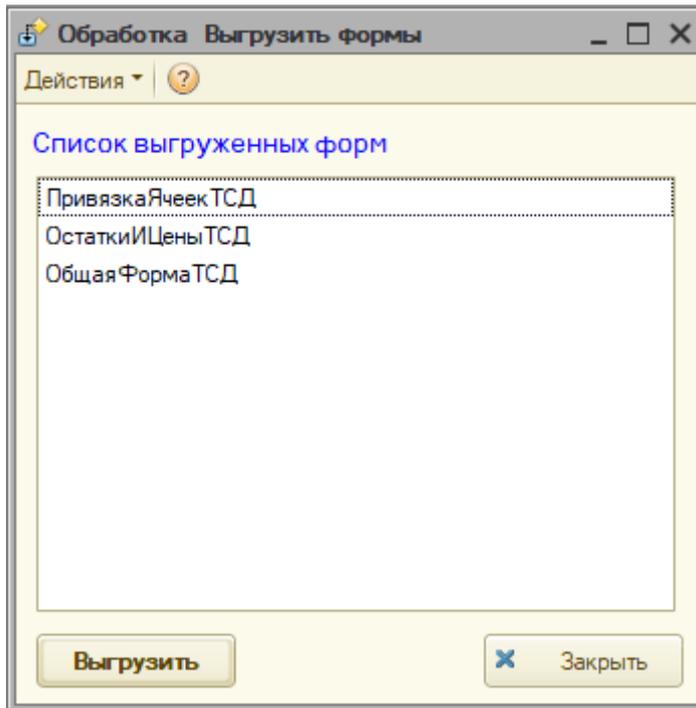


б) После изменения штатных форм (общие формы в расширении). Формы необходимо выгрузить с помощью обработки:



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

Выгрузить формы. Увидим перечень выгруженных форм.



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.

1.3 Настройка AllegroClient – prof 2.0

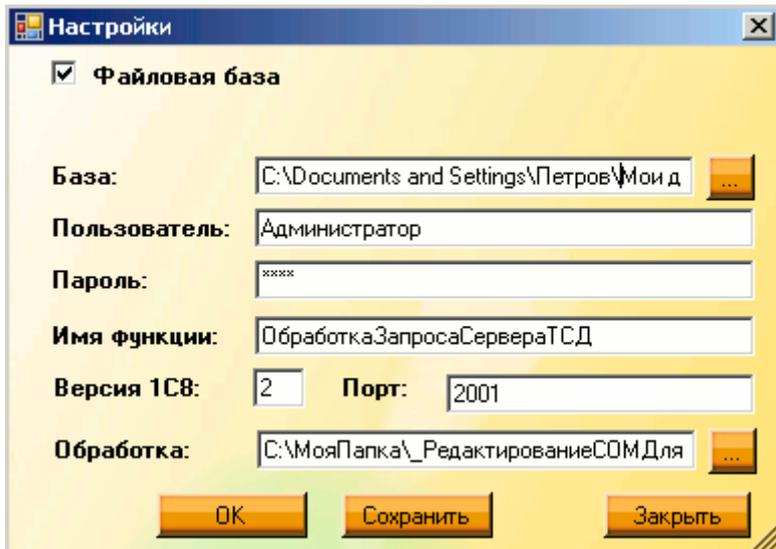
1.3.1. Установка Сервера ТСД

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

1.3.2 Настройка Сервера ТСД

Настройка Сервера ТСД :

а) 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.ерf" (обработка содержится в поставке «AllegroClient»)

b) Настройка для серверной базы 1С.

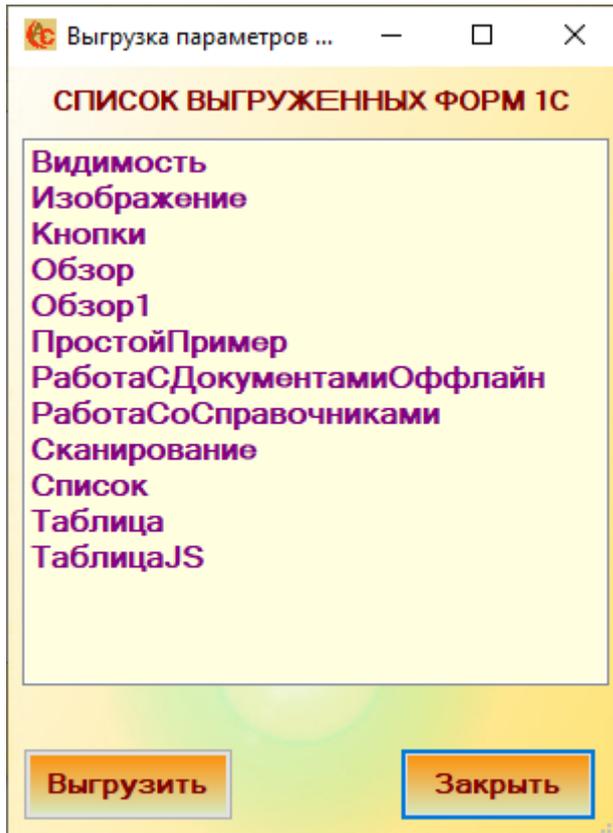
Важно! Пользователь 1С, которого указываете в настройках сервера, должен иметь режим запуска «Обычное приложение».

- 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)

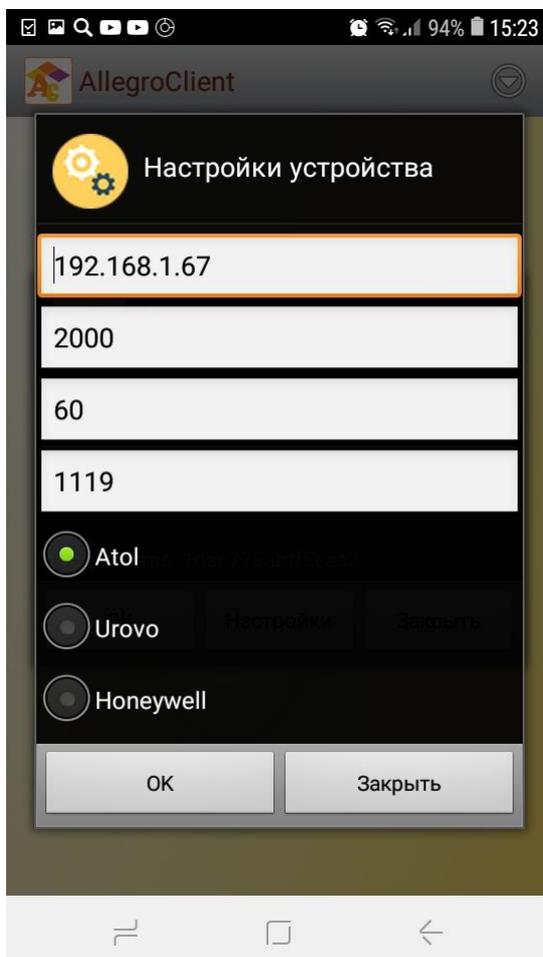
Обработка - путь к обработке " AllegroКлиент.erf" (обработка содержится в поставке «AllegroClient»)

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

2 Конфигурирование логики работы с операциями в режиме онлайн

2.1 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 с 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 находится 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 содержатся 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.

Товар	Характеристика	Количество
ab <Поле _	ab <Поле _	ab <Пол.

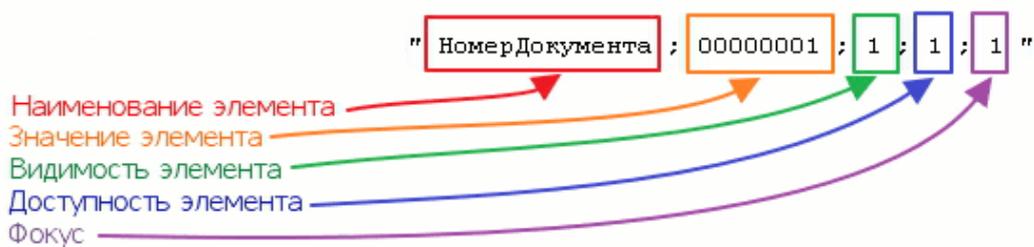
Отображение формы на устройстве

Ботинки	Синие	2
Шарф	зеленый	5
Пальто	черное	10

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.

```
МассивДляОтправки.Добавить ("СписокДокументов; " ; ; " ) ;
```

Пустое значение, очищает список

```
МассивДляОтправки.Добавить ("СписокДокументов; 00000001 ; ; " ) ;
МассивДляОтправки.Добавить ("СписокДокументов; 00000002 ; ; " ) ;
```

Значения для добавления в список

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 отдельной колонки можно установить следующие настройки:

- Размер шрифта
- Жирность шрифта
- Цвет шрифта

Так же можно сгруппировать колонки следующим образом:

Колонка1	
Колонка2	
Колонка3	
ab <Поле ввода>	
ab <Поле ввода>	
ab <Поле ввода>	

Важно! Можно так реализовать только в первой колонке!

Важно! Максимальное количество колонок для отдельного табличного поля не более 9.

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:

```
ДобавитьЭлементДляОбработкиТаблицы(МассивДляОтправки, "ТабличноеПоле1", "Delete ", "1", "1", "1", "0");
Возврат ОтправитьОтвет(МассивДляОтправки);
```

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:

```

ИмяфайлаКартнки = КаталогВременныхфайлов () + "\Картинка1.jpg";
ДвоичныеДанныеКартинки = ПолучитьМакет ("Картинка");
ДвоичныеДанныеКартинки.Записать (ИмяфайлаКартнки);
Картинка = Новый Картинка (ИмяфайлаКартнки);
Бэйз64Строка = Base64Строка (ДвоичныеДанныеКартинки);
Бэйз64Строка = СтрЗаменить (Бэйз64Строка, Символы.VK, "");
Бэйз64Строка = СтрЗаменить (Бэйз64Строка, Символы.ПС, "");

МассивДляОтправки.Добавить ("Изображение;" + Бэйз64Строка + ";1;1;1");

Возврат ОтправитьОтвет (МассивДляОтправки);

```

2.9 Команды

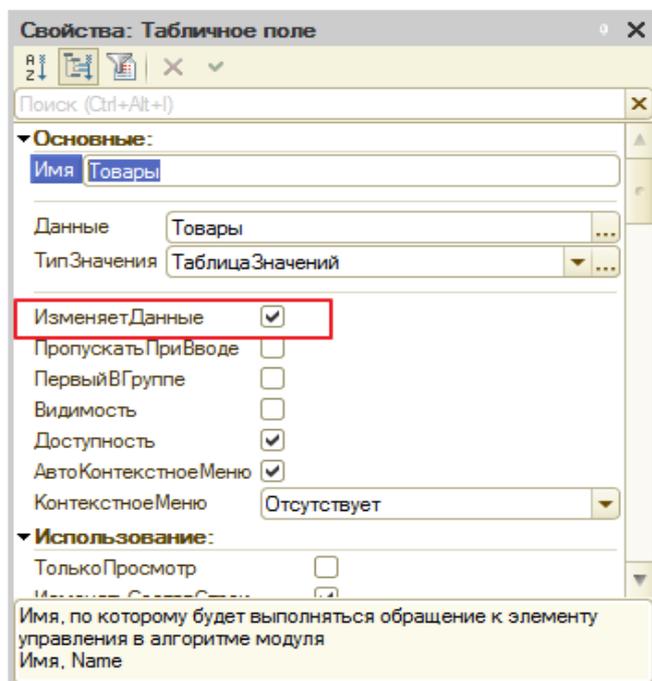
2.9.1 Сохранение данных формы

Значения элементов формы можно сохранить в xml 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

Для того чтобы значения элемента формы были сохранены в файл, в свойствах элемента необходимо установить флажок «ИзменяетДанные».



Важно! В отдельной форме можно сохранить значения только одного элемента «Табличное поле».

Данные элементов формы сохраняются в два различных xml 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;

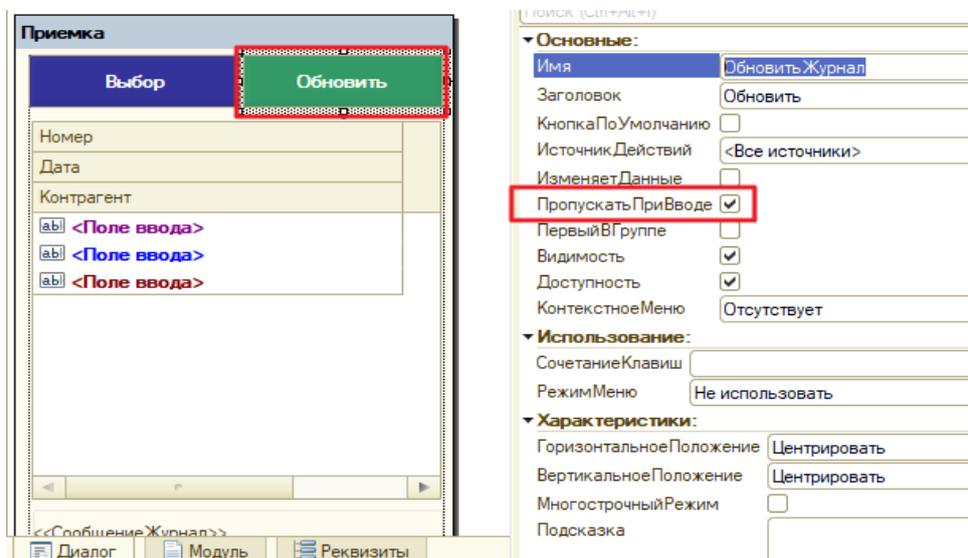
Для загрузки сохраненных данных используется команда LoadDocument, имеет следующий формат:

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 вернуть com-массив с элементом «SUCCESS,»

2.10 Диалоговые окна

2.10.1. Диалоговое окно с ошибкой.

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

```
МассивДляОтправки.Добавить ("ERROR, Документ не найден!");
```

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 вернуть один элемент com-массива в следующем формате –«ShowMessage;Текст сообщения».

2.11 Сканирование штрих-кодов



2.11.1 Сканирование камерой устройства

Сканирование штрих-кодов производится 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.

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 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!

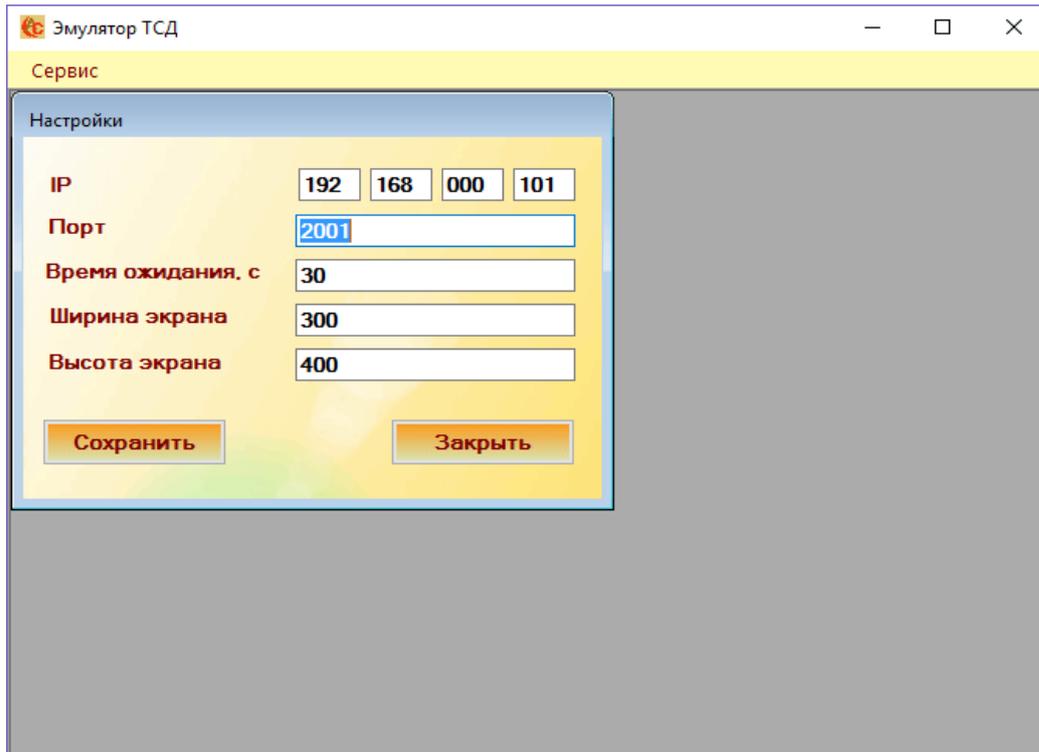
Настройки эмулятора содержатся в выпадающем меню «Сервис». Настройки имеют следующий вид:

Для веб соединения.

Настройки	
Url	<input type="text" value="http://192.168.1.67/Globus"/>
Пользователь	<input type="text" value="tsd"/>
Пароль	<input type="password" value="..."/>
Ширина экрана	<input type="text" value="400"/>
Высота экрана	<input type="text" value="600"/>
<input type="button" value="Сохранить"/> <input type="button" value="Закрыть"/>	

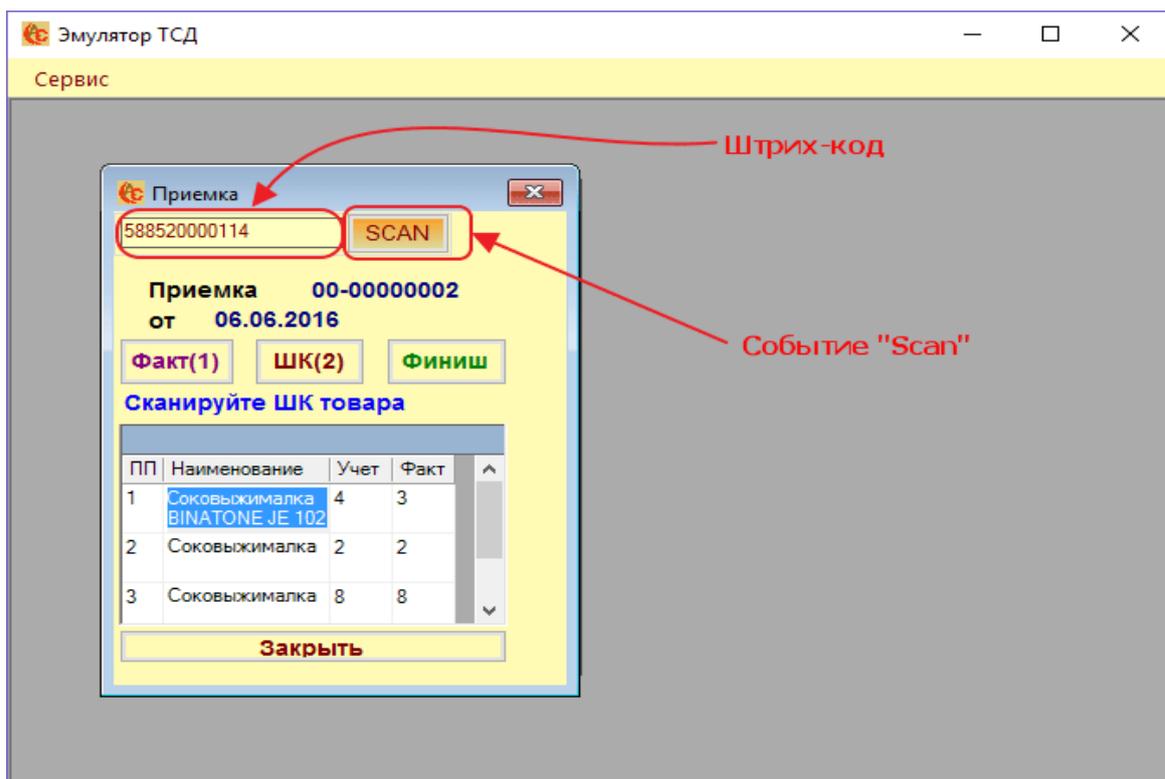
Как настроить описано в пункте 1.1.5.

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.



3 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-Prof под андроид используется JavaScript. Принцип работы аналогичен работе в 1С.

Есть функция, на входе у которой массив строк с элементами формы через разделитель «;» (переменная массива строк называется args).

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):

```
List<String> process(String [] args, String Form,String EventName,String Barcode)
{
    if ((Form.indexOf("Приемка") != -1) & (EventName.indexOf("ОбновитьЖурнал") != -1))
    {
        ListWithControls = new ArrayList<String>();
        AllegroModule.GenerateError(ListWithControls,"Ошибка");
        return ListWithControls;
    }
    if ((Form.indexOf("Приемка") != -1) & (EventName.indexOf("Scan") != -1))
    {
        ListWithControls = new ArrayList<String>();
        AllegroModule.GenerateError(ListWithControls,Barcode);
        return ListWithControls;
    }

    return null;
}
```

Параметры :

1. args — Массив строк с элементами формы
2. Form — Наименование формы(операции)
3. EventName — Наименование события
4. Barcode — 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).

```
Товары;ИндексСтроки;Колонка1;Колонка2;КолонкаN;1
Товары;ИндексСтроки;Колонка1;Колонка2;КолонкаN;0
Товары;ИндексСтроки;Колонка1;Колонка2;КолонкаN;0
```

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

```
List<String> ListWithControls = new ArrayList<String>();
String str = "";
```

3.1.2. Общие процедуры и функции для работы с JavaScript

Для упрощения работы с JavaScript предусмотрены predefined процедуры и функции.

AllegroModule.AddControl — процедура для добавления элемента в массив для дальнейшей обработки в приложении.

```
public static void AddControl( List<String> ListWithControls,
                               String Control,
                               String Value,
                               String Visible,
                               String Available,
                               String Focus )
```

AllegroModule.AddElementForProcessingTable — процедура для добавления строки таблицы в массив для дальнейшей обработки в приложении.

```
public static void AddElementForProcessingTable(List<String> ListWithControls,
                                             String Table,
                                             String Command,
                                             String Visible,
                                             String Available,
                                             String Focus,
                                             String FieldIndex)|
```

AllegroModule.GenerateError — процедура для формирования ошибки.

```
public static void GenerateError(List<String> ListWithControls, String ErrorText)
```

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

```
public static List<ArrayList<String>> GetListOfControls(String[] ArrWithControls)
```

AllegroModule.GetValueOfControlByName – получить значение элемента по имени.

```
public static String GetValueOfControlByName(List<ArrayList<String>> ListOfControls,
                                             String ControlName)
```

AllegroModule.GetTableRows – получение строк таблицы по наименованию таблицы.

```
public static List<ArrayList<String>> GetTableRows(List<ArrayList<String>> ListOfControls,
                                                    String TableName)
```

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

```
public static List<ArrayList<String>> FindTableRowsByValue(List<ArrayList<String>> ListOfControls,
                                                         String TableName,
                                                         String Value,
                                                         int Index)
```

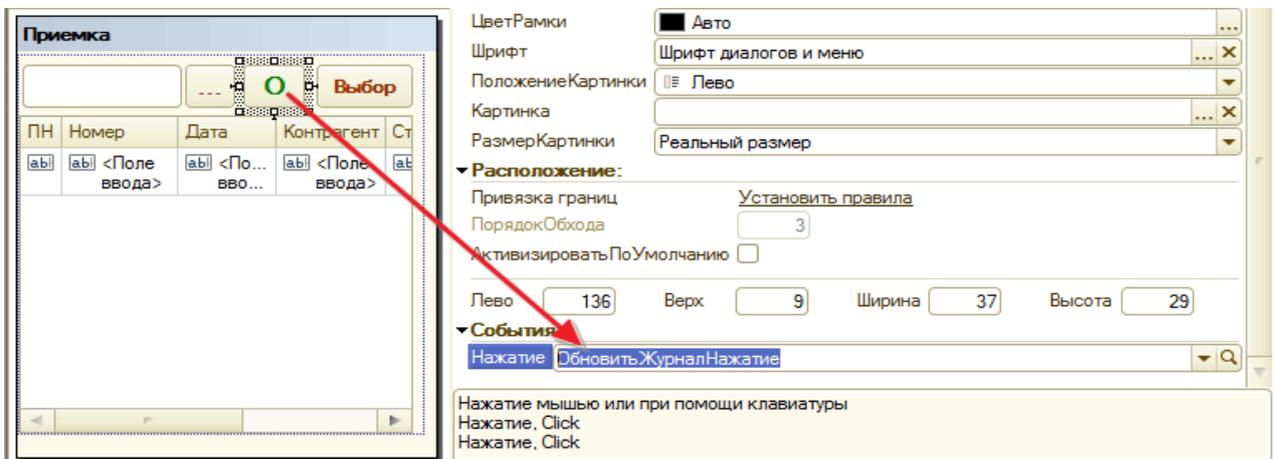
AllegroModule.FindTableRowByValue – поиск строки таблицы по значению, нужно указать значение поиска и колонку поиска(Index), возвращает строку, если найдено или пустую строку, если не найдено.

```
public static String FindTableRowByValue(List<ArrayList<String>> ListOfControls,
                                         String TableName,
                                         String Value,
                                         int Index)
```

AllegroModule.GetValueOfSelectedItemByName – возвращает выделенную строку в таблице или пустую строку.

```
public static String GetValueOfSelectedItemByName(List<ArrayList<String>> ListOfControls,
                                                 String TableName)
```

JavaScript можно выполнить для обработчиков событий элементов формы: кнопка и поле ввода.

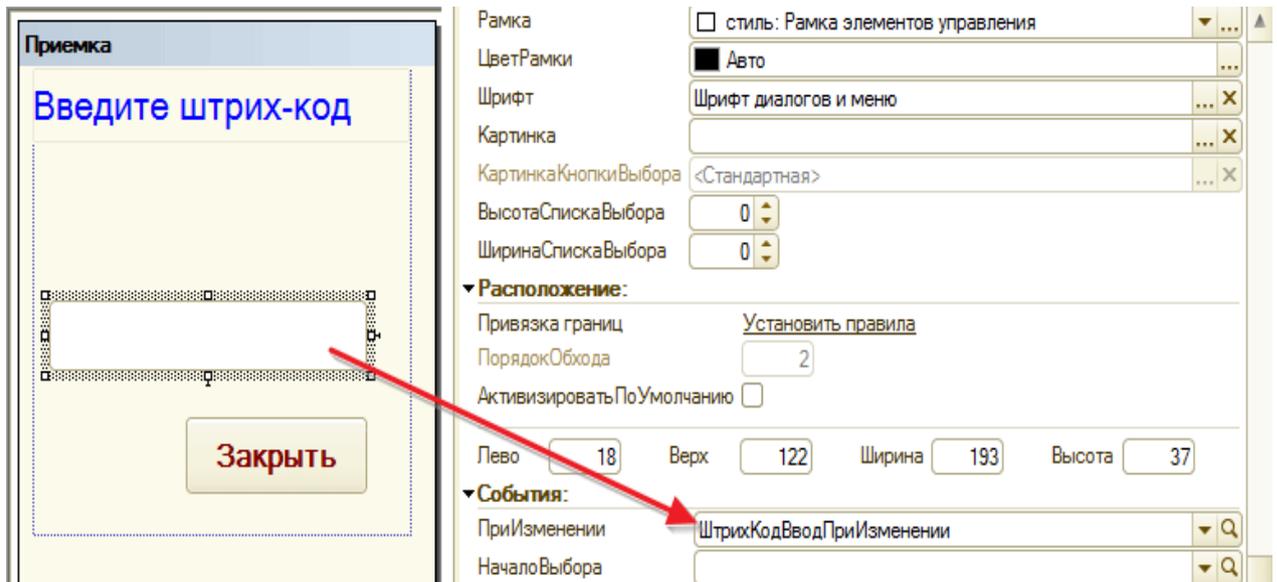


Процедура ОбновитьЖурналНажатие (Элемент)

```
js = " ListWithControls = new ArrayList<String>();
      | AllegroModule.GenerateError(ListWithControls, ""Ошибка"");
      | return ListWithControls; ";
```

КонецПроцедуры

Для поля ввода используется событие «При изменении».



```

Процедура ШтрихКодВводПриИзменении (Элемент)
    js = " ArrWithControls = AllegroModule.GetListOfControls (args);
    | NumberDoc = AllegroModule.GetValueOfControlByName (ArrWithControls, "НомерДокумента");
    | ListWithControls = new ArrayList<String>();
    | AllegroModule.GenerateError (ListWithControls, "Номер документа:" + NumberDoc);
    | return ListWithControls; ";
КонецПроцедуры
  
```

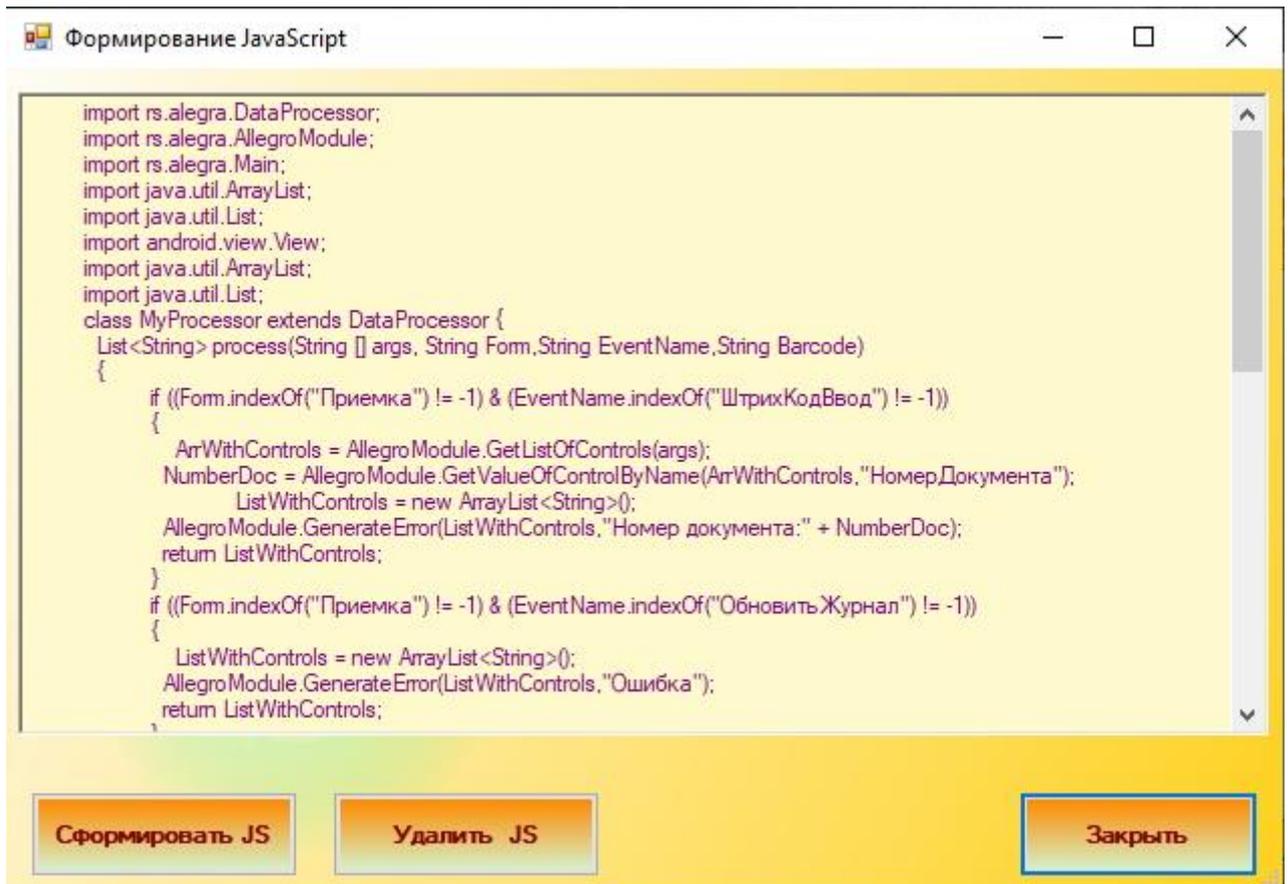
Так же скрипт можно использовать для события сканирования, для этого нужно в модуле формы создать процедуру «Scan», в которой разместить скрипт. Штрих-код передается в predeterminedенную переменную «Barcode», значение которой можно использовать в коде.

```

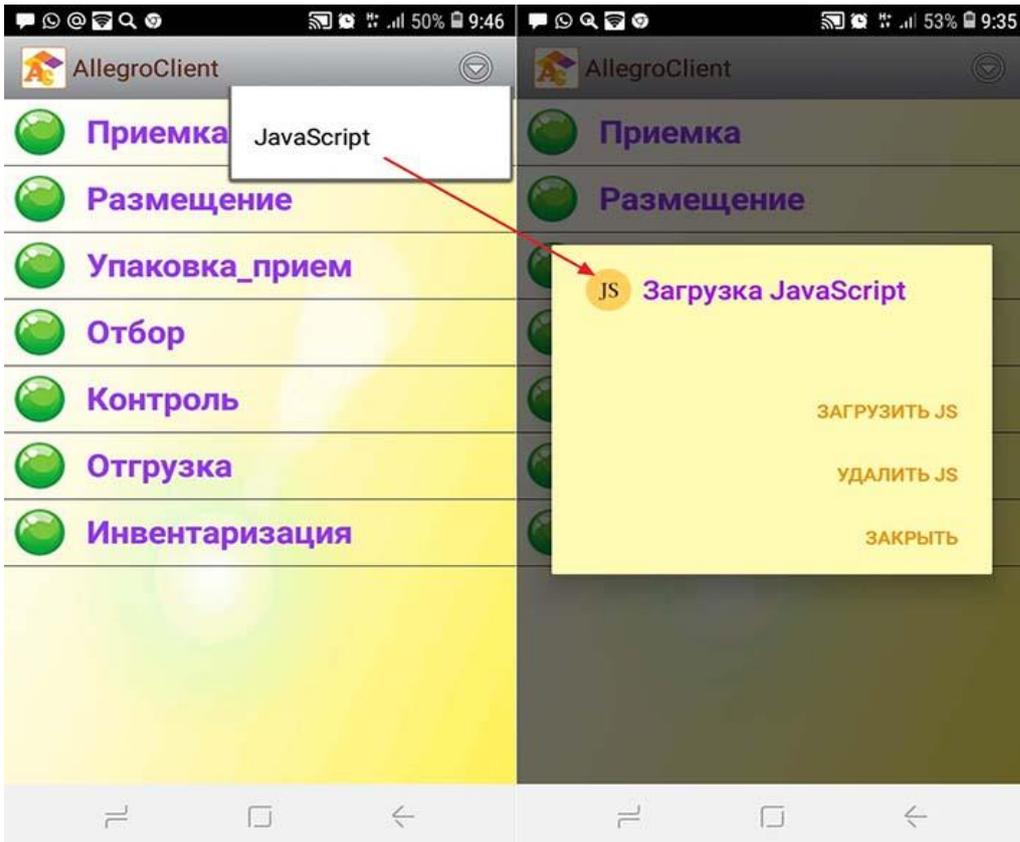
Процедура Scan()
    js = " ListWithControls = new ArrayList<String>();
    | AllegroModule.GenerateError (ListWithControls, Barcode);
    | return ListWithControls; ";
КонецПроцедуры
  
```

3.1.3. Выгрузка JavaScript в клиентское приложение

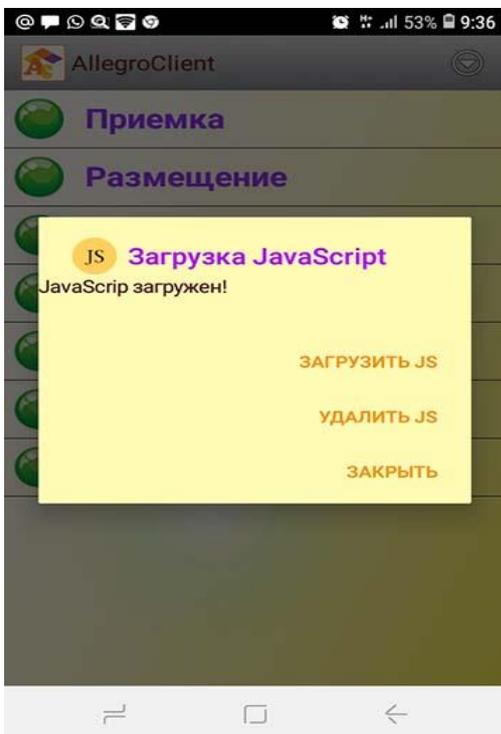
В серверном приложении открываем пункт меню «Выгрузить JavaScript» и формируем JavaScript.



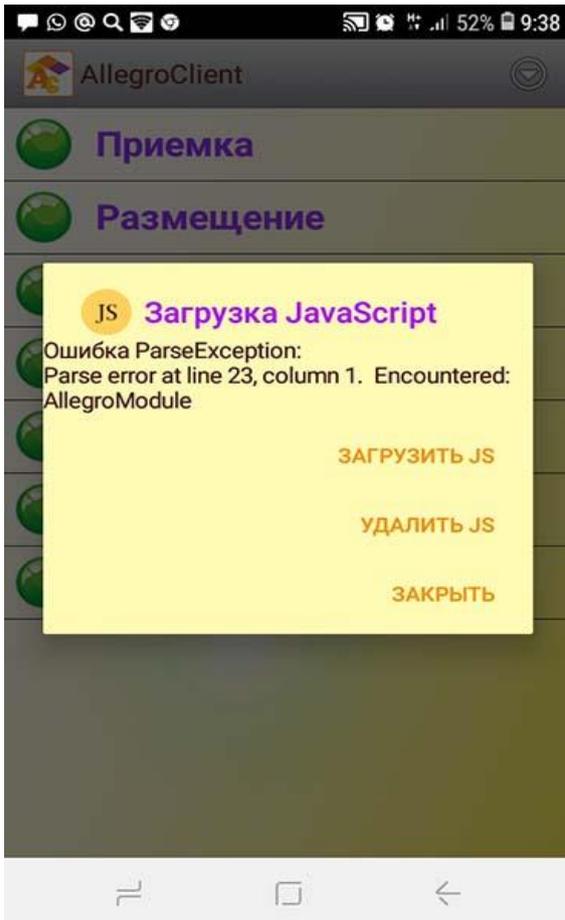
В клиентском приложении открываем пункт меню «JavaScript».



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 серверном приложении и найти ошибку.



Видим, что пропущена точка с запятой.

Исправляем в 1С и выгружаем скрипт заново.

```

Процедура ОбновитьЖурналНажатие (Элемент)
    js = " ListWithControls = new ArrayList<String> ();
        | AllegroModule.GenerateError(ListWithControls, ""Ошибка"");
        | return ListWithControls; ";
КонецПроцедуры
  
```

После выполнения клиентского кода можно выполнить серверный код, для этого нужно использовать ключевое слово **ExecuteServerCode**.

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 light.

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» — «storageCells»
- 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

Данные загружаются частями по 9000 строк, т.е. если справочник имеет список более 9000, то событие по загрузке справочника будет выполняться несколько раз.

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:

Если Событие = "ЗагрузкаНоменклатуры" Тогда

```

КодНоменклатуры = СокрЛП (мРазбивкаСтроки (Данные) );
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ ПЕРВЫЕ 9000
| Номенклатура.Код,
| Номенклатура.Наименование,
| Номенклатура.ШтрихКод
|ИЗ
| Справочник.Номенклатура КАК Номенклатура
| " + ? (ПустаяСтрока (КодНоменклатуры), "", " ГДЕ Номенклатура.Код > &КодСправочника ") + "
| УПОРЯДОЧИТЬ ПО Номенклатура.Код";

Запрос.УстановитьПараметр ("КодСправочника", КодНоменклатуры);
РезЗапроса = Запрос.Выполнить ();
Если РезЗапроса.Пустой () Тогда
    МассивДляОтправки.Добавить ("NEXTCATALOG");
    Возврат ОтправитьОтвет (МассивДляОтправки);
КонецЕсли;

РезЗапросаВыборка = РезЗапроса.Выбрать ();
Пока РезЗапросаВыборка.Следующий () Цикл
    МассивДляОтправки.Добавить (Строка (РезЗапросаВыборка.Код) + ";"
        + Строка (РезЗапросаВыборка.Наименование) + ";"
        + Строка (РезЗапросаВыборка.ШтрихКод) + ";;;");
КонецЦикла;

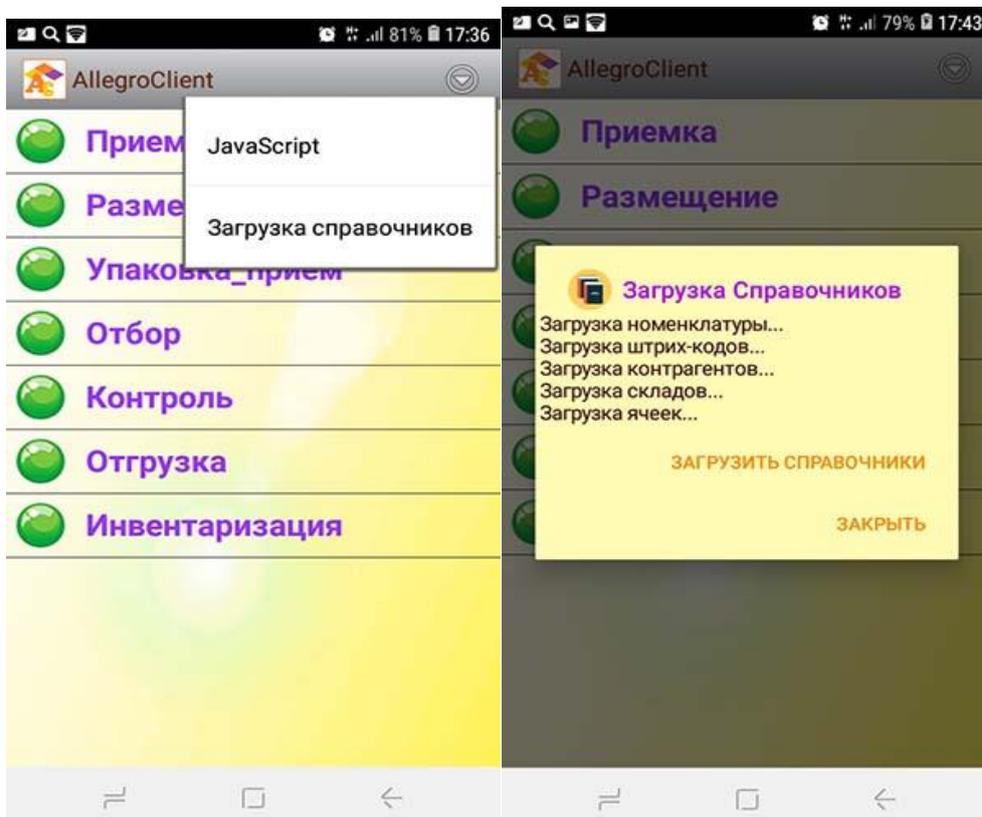
Если РезЗапросаВыборка.Количество () < 9000 Тогда
    МассивДляОтправки.Добавить ("NEXTCATALOG");
КонецЕсли;

Возврат ОтправитьОтвет (МассивДляОтправки);

```

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 – 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 – 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 – 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:

```

js = " ListWithControls = new ArrayList<String>();
| filter = new HashMap();
| filter.put(0, ""2538093214"");
| filter.put(1, ""Бакалея"");
| _List = AllegroModule.GetTableRows ("Контрагенты", filter);
|
| if (_List==null)
|     AllegroModule.GenerateError(ListWithControls, ""Ошибка"");
| else {
|     String[] ss = _List.get(0).split(";"");
|
|     AllegroModule.GenerateError(ListWithControls, ""Наименование контрагента:""
|                                     +ss[1].toString());
| }
| return ListWithControls; ";

```

Важно! Параметр TableName может иметь следующие значения:

- ШтрихКоды (barCodes)
- Номенклатура (nomenclature)
- Контрагенты(contractors)
- Склады(warehouses)
- Ячейки(storageCells)
- 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-prof 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 — Добавляет значения в таблицу «table_doc_header». В параметре необходимо передавать строку со значениями, которые разделены «;».

public static void AddDocumentHeader(String values)

AllegroModule.AddDocumentRows— Добавляет значения в таблицу «table_doc_table». В параметре необходимо передавать строку со значениями, которые разделены «;».

public static void AddDocumentRows(String values)

AllegroModule.DeleteDocumentHeader — 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 — 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 — удаляет все значения из таблиц «table_doc_header» и «table_doc_table» по типу документа

public static void DeleteDocuments(String DocType)

AllegroModule.GetDocumentHeader— 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 — 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 — 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 — 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 севера

ADDDOCUMENT — для загрузки данных в таблицу «table_doc_header». Данные для записи идут после команды через разделитель «;»

«ADDDOCUMENT;000000001;Дос;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;Дос;1;0001;Комбайн MOULINEX A77 4С;2;0»

DELETEDOCUMENTHEADER — удаление данных из таблиц «table_doc_header». После команды через разделитель «;» идут данные с номером документа и типом документа.

«DELETEDOCUMENTHEADER;000000001;Дос»

DELETEDOCUMENTROWS — удаление данных из таблиц «table_doc_table». После команды через разделитель «;» идут данные с номером документа и типом документа.

«DELETEDOCUMENTROWS;000000001;Дос»

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».

4 Настройка сканирования на ТСД для работы с AllegroClient

4.1 Общее описание

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

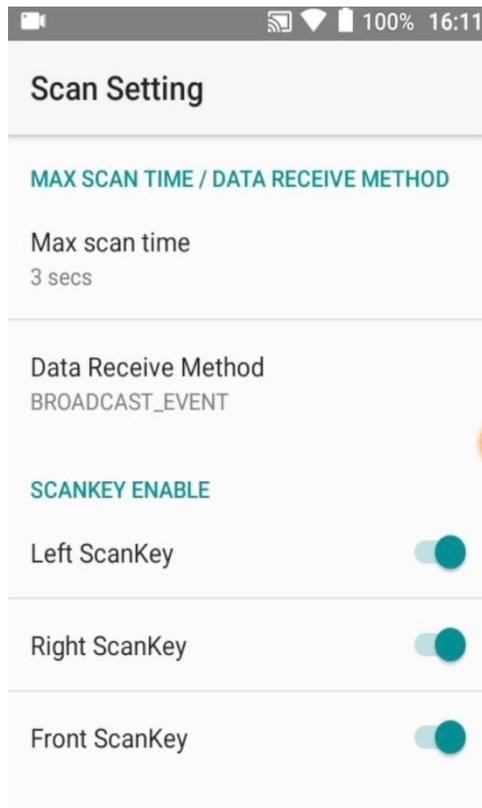
Ниже приведены настройки для самых распространенных ТСД:

- *ATOL Smart Lite u SmartSlim*
- *ТСД UROVO*
- *ТСД Honeywell*
- *ТСД Mertech*

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



Достаточно настроить один параметр, в поле **Data Receive Method** выбрать **BROADCAST_EVENT**



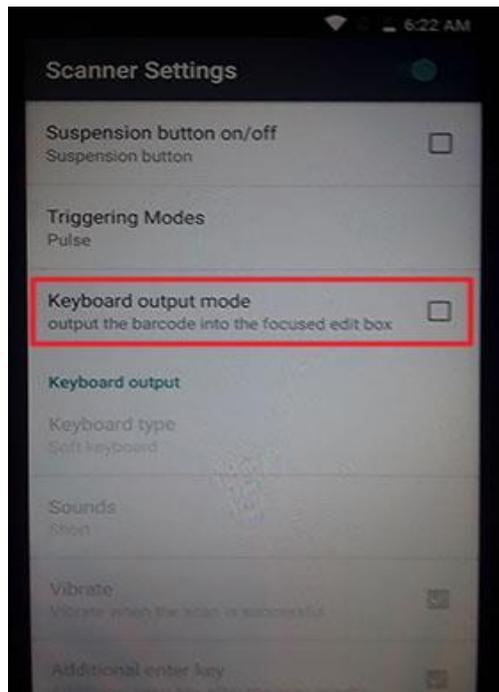
Инструкция по настройке:

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*.

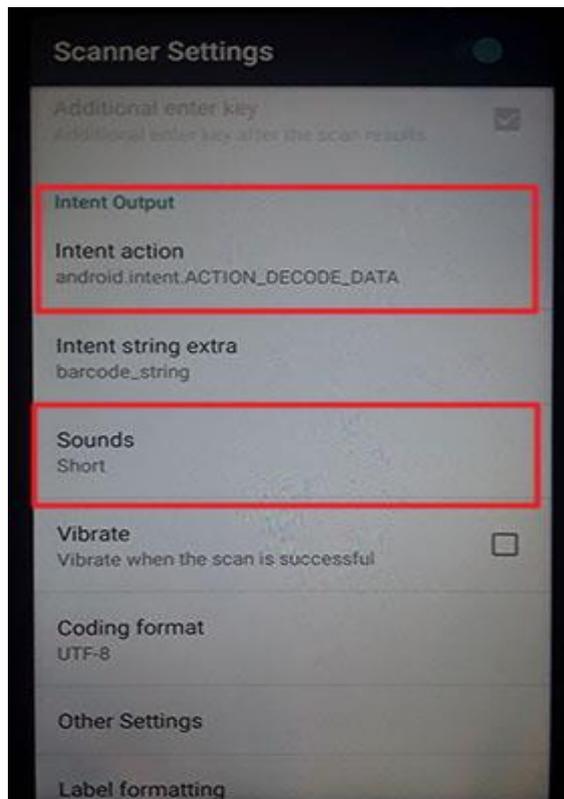


Снять флажок **Keyboard output mode**.



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.

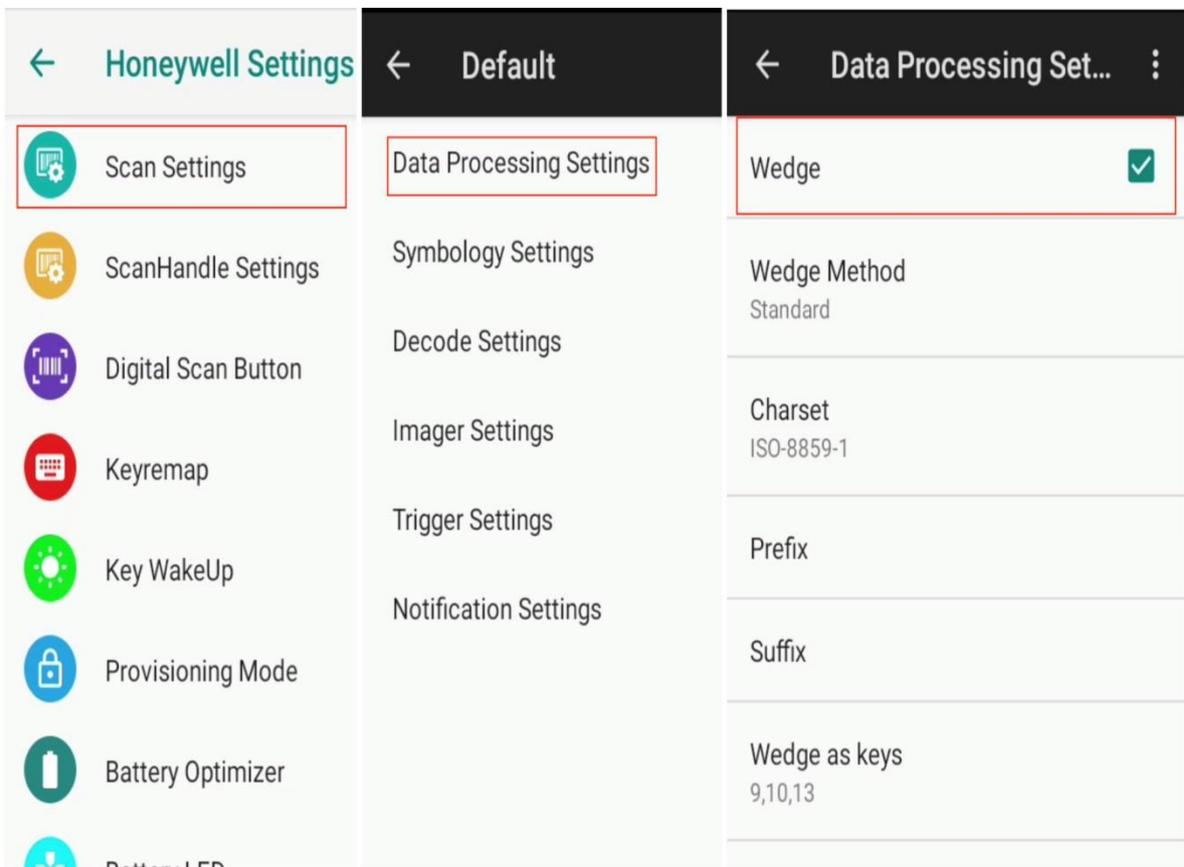


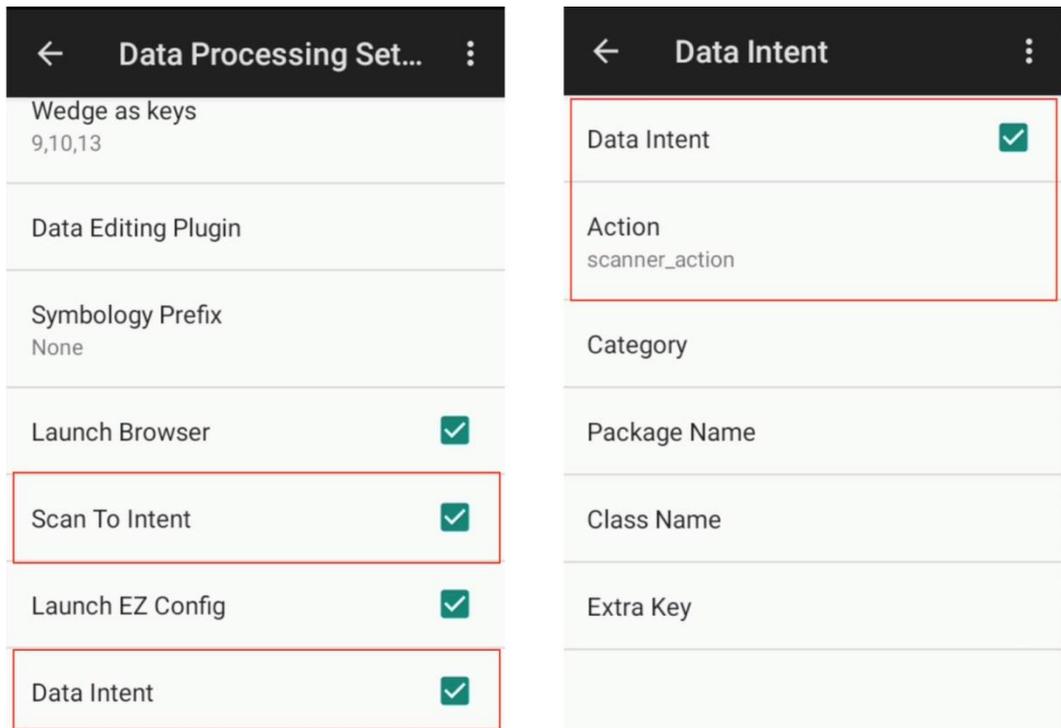
4.1.3. Пример настройки сканирования для терминала сбора данных *Honeywell*.



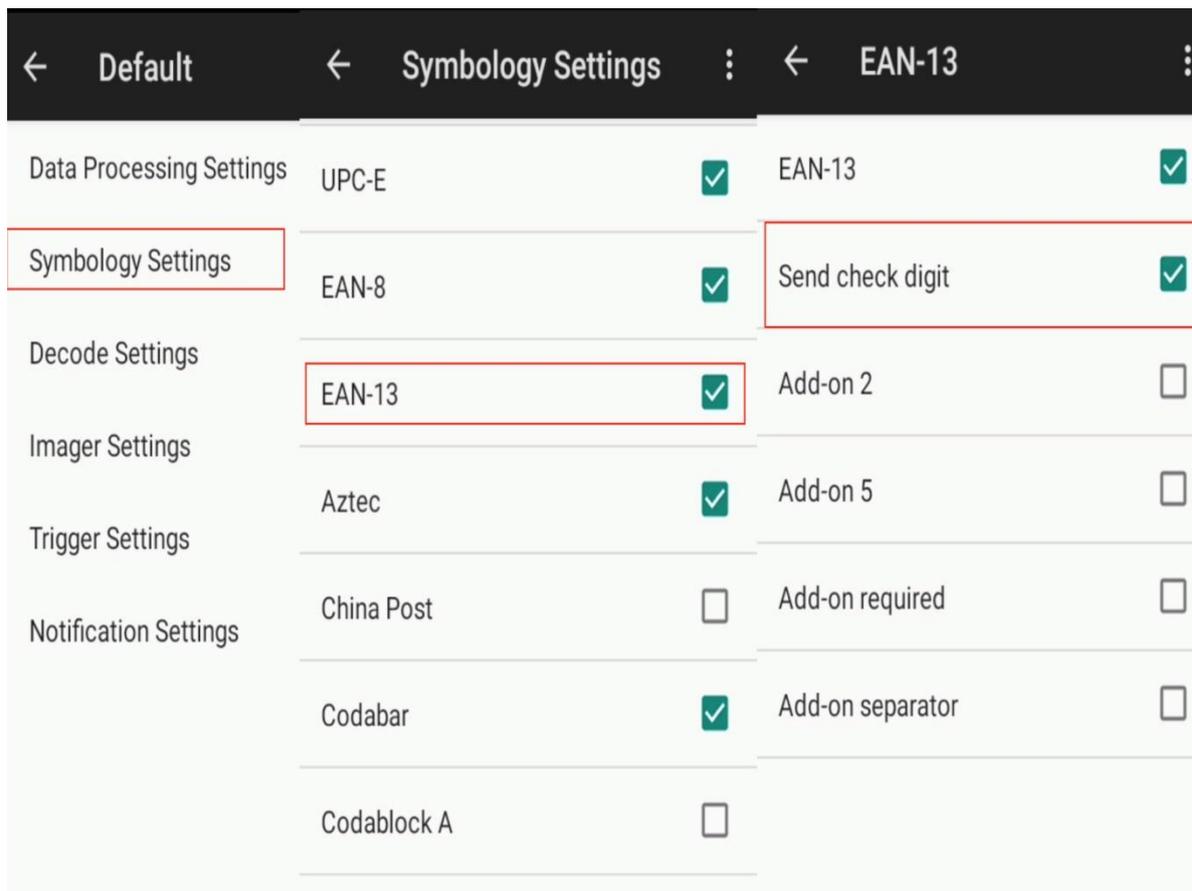
1. Открыть настройки сканирования (Scan Settings) - **Data Processing Settings** - Активировать **Wedge** - Активировать **Scan To Intent**.

Перейти в меню **Data Intent** и в поле **Action** ввести "**scanner_action**"



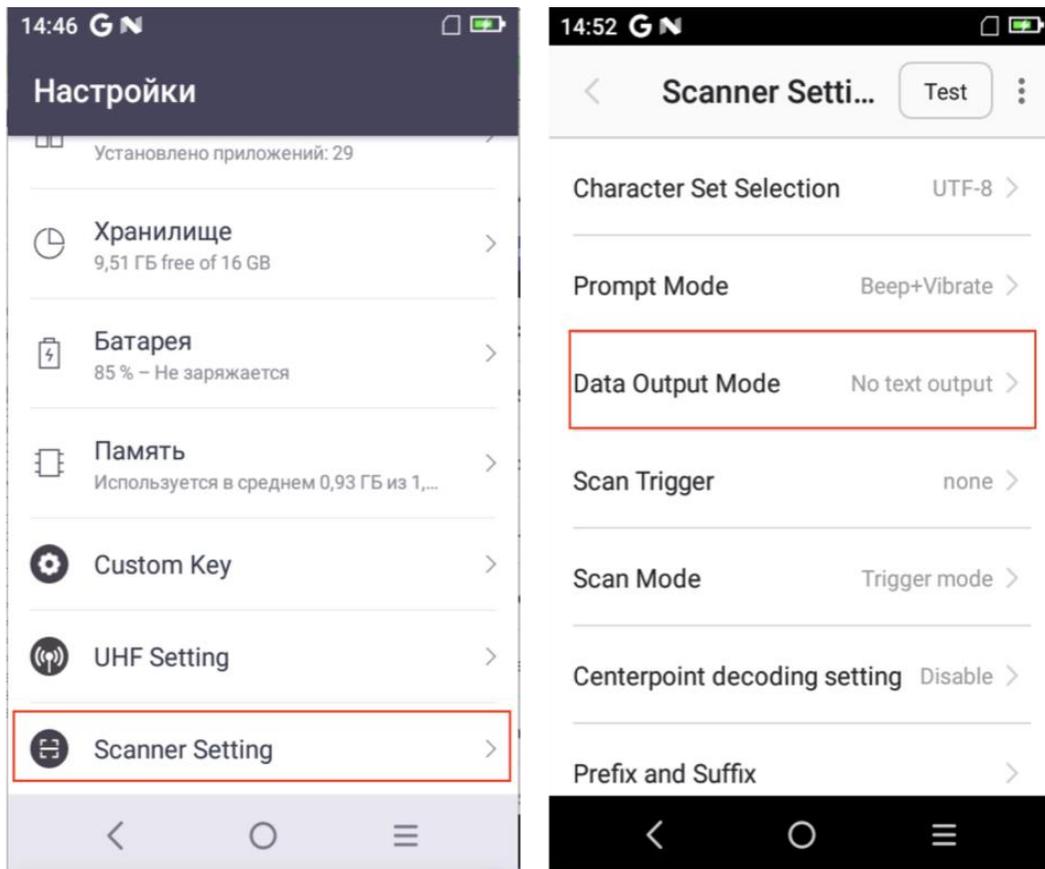


2. Открыть настройки сканирования (**Scan Settings**) - **Symbology Settings** - выбрать необходимый формат кода и в нем активировать **Send check digit**

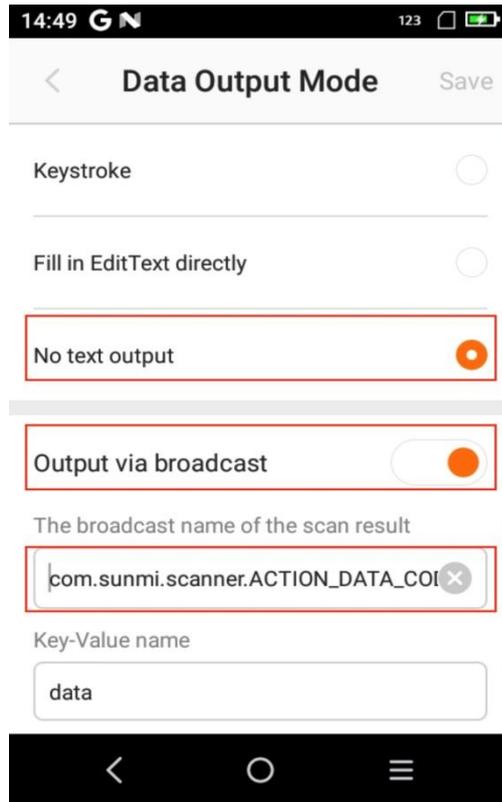


4.1.4. Пример настройки сканирования для терминала сбора данных *Mertech*.

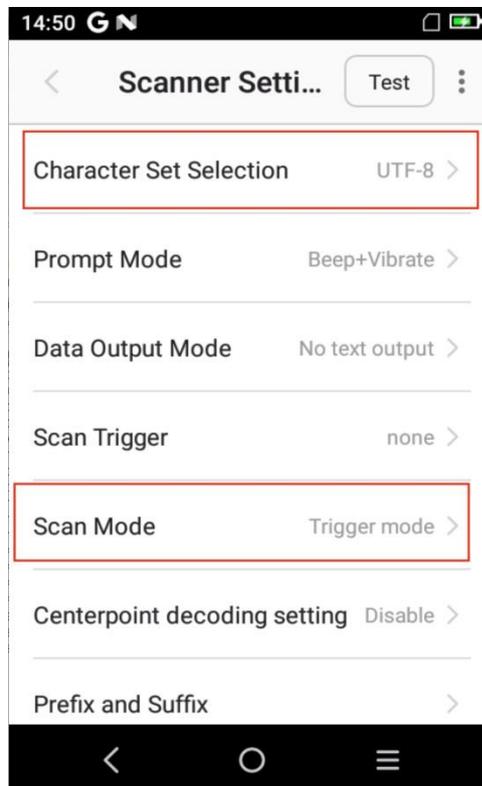
Настройки выбрать пункт "Scanner Setting" - Data Output Mode.



Указать: "No text output", Output via broadcast. Убедиться, что поле "The broadcast name" заполнено "com.sunmi.scanner.ACTION_DATA_CODE_RECEIVED"

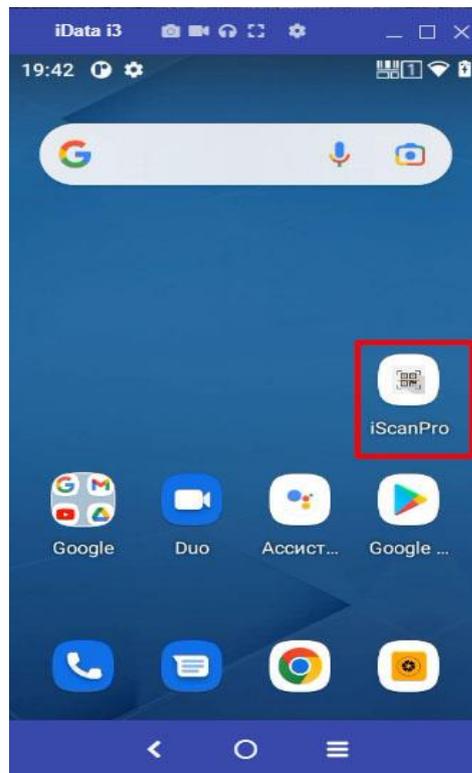


Далее, вернуться в меню **Scan Settings** и убедиться, что **Character Set Selection** выбрано **UTF-8**.
Scan Mode - Trigger mode

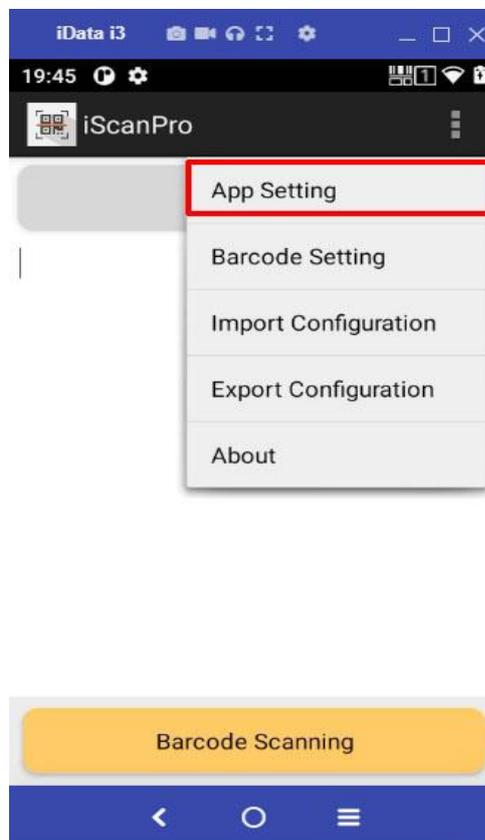


4.1.5. Пример настройки сканирования для терминала сбора данных *iData*.

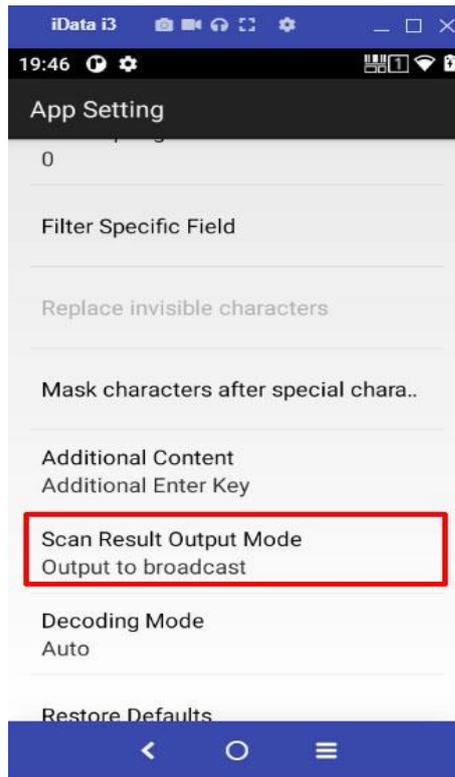
Открыть приложение **iScanPro**.



Выбрать **App Setting**.



Выбрать пункт **Scan Result Output Mode**.



Выбрать **Output to broadcast**.

