

## Пишем стих.

Ну кто может. Кто нет – перепечатывает.

## Немного теории.

Мобильная аппликация состоит из 3-х взаимосвязанных «слоев»:

1. внешнего вида (пользовательского интерфейса). Его мы набираем из визуальных компонентов в конструкторе, его видит пользователь при работе и с ним взаимодействует.
2. Внутренняя база данных. Структура, отдельная и изолированная для каждого бизнес-процесса и состоящая из набора таблиц, необходимых для работы. Например, таблицы «Товары» и «Штрих-коды» нужны для прайс-чекера, Для инвентаризации могут потребоваться еще «Таблица документов» и «Таблица строк в документах»
3. Инструкции (цепочки инструкций) – они определяют поведение каждой формы, ее реакции на нажатие кнопок, сканирование, они получают, отдают и изменяют данные.

В любом бизнес-процессе, сконструированном в системе, первичны именно данные. Форма отображает их, запускает инструкции. Инструкции модифицируют данные. Форма - показывает результат. **Показали данные -> Изменили данные -> Показали измененные данные.**

В этой части, мы будем выводить на экран Хокку поэта Эномото Кикаку:

Мошек легкий рой  
Вверх летит - плавучий мост  
Для моей мечты.

Помните, что, самовольно изменяя текст, вы действуете на свой страх и риск. Разработчики не несут ответственности за последствия.

## Экранные формы

Для выполнения, будем использовать 2 формы 2-х разных типов: на первой мы будем вносить, редактировать или удалять текст строку за строкой, на второй – любоваться результатом. Первая форма будет иметь тип «Объект» - то есть она оперирует одним конкретным объектом данных, одной записью. Вторая – тип «Список». Выводит список объектов, соответствующих критерию.

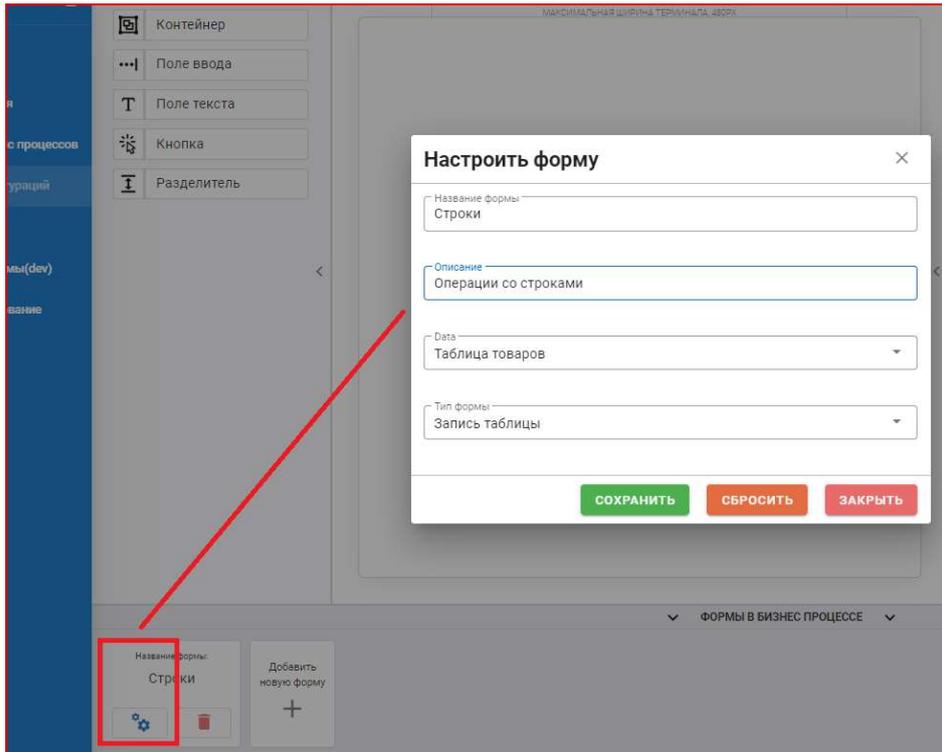
Так как мы для работы не получаем никаких данных из товароучетной системы и ничего не отправляем, берем первую подходящую структуру, из доступных в конструкторе. Пусть это

будет «Таблица товаров», каждой строке нашего Хокку будет соответствовать название «Товара», которое мы создадим в процессе.

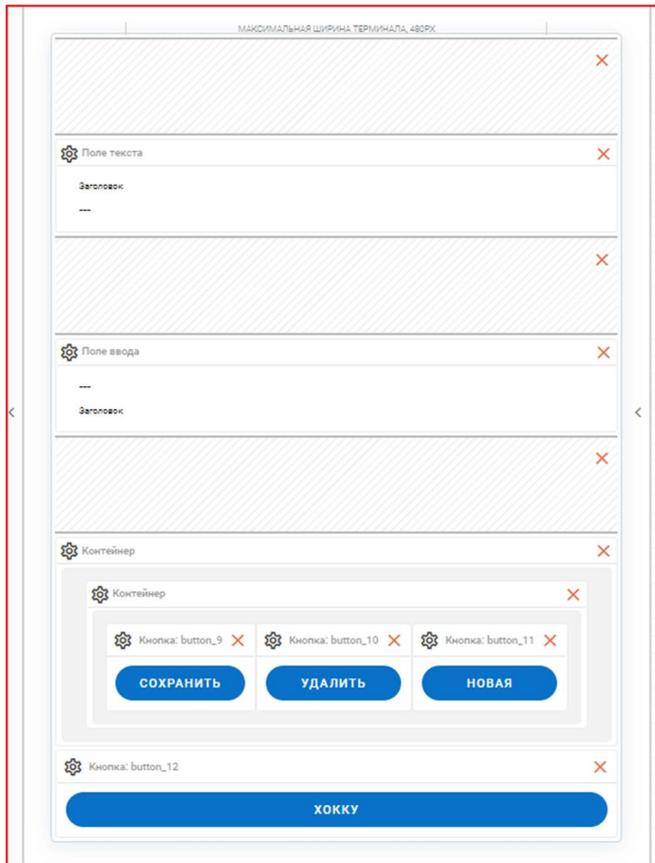
Создаем новый бизнес-процесс, называем его «Хокку» и заходим в него.

Объектная форма «Строки»

Создаем первую форму и устанавливаем ее параметры:

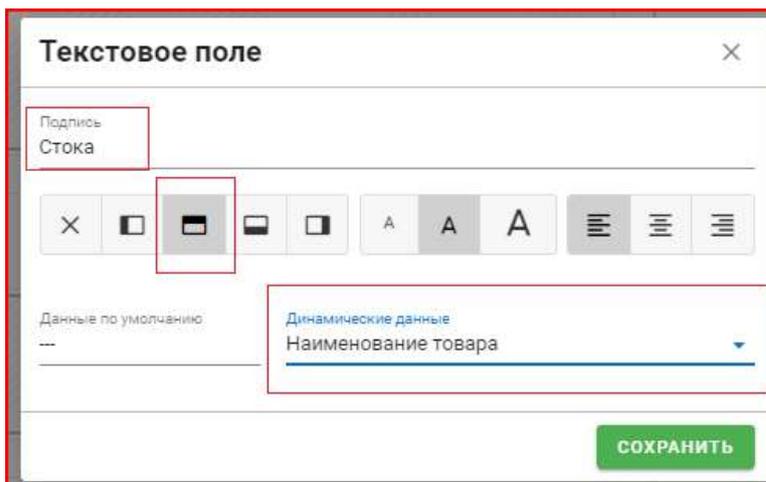


Добавляем нужные нам элементы на форму:



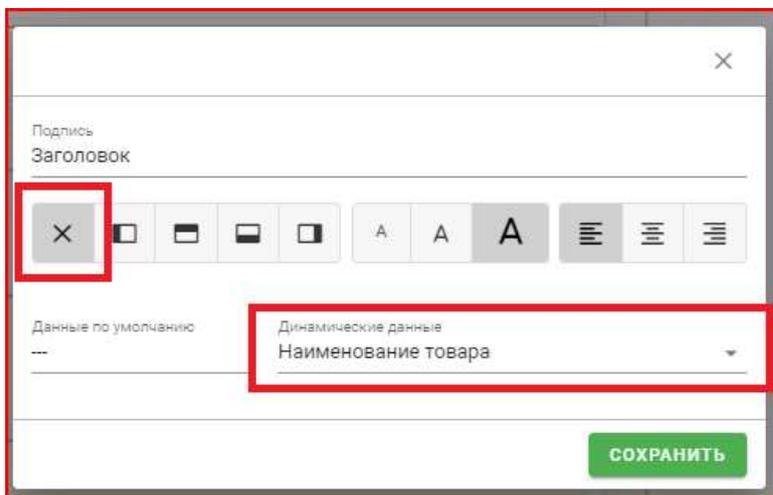
**Поле текста** – в нем мы будем отображать текст после того, как наберем его и сохраним в базе данных.

Настраиваем.



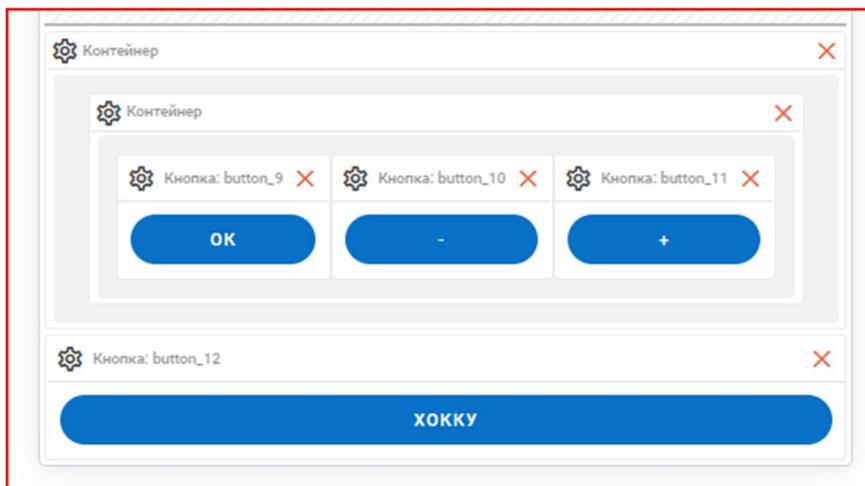
Устанавливаем текст и расположение подписи, указываем, из какого поля базы данных мы хотим видеть информацию.

Поле ввода – в него мы будем набирать каждую строку текста.



Отключаем отображение подписи и также указываем связь с полем базы данных.

Блок кнопок управления:



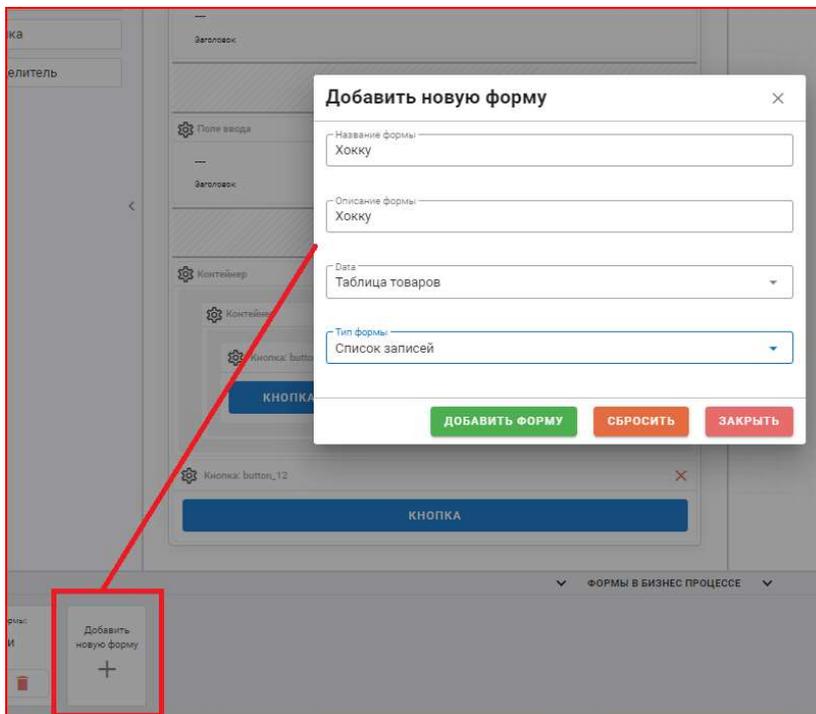
**Внимание!** Подписи кнопок **button\_X** — это системные имена, они могут отличаться от тех, которые будут в вашем конструкторе бизнес-процесса!

Состоит из вложенных один в другой контейнеров с разными режимами размещения содержимого. Общий для этого блока контейнер распределяет свое содержимое (кнопку и второй контейнер) в столбец, вложенный контейнер с 3-мя кнопками - в строку.

Кнопка «Ок» будет сохранять данные с экрана в БД, «-» - удалять из БД и с экрана строку, «+» - добавлять новую пустую запись в БД и отображать ее на экране для редактирования.

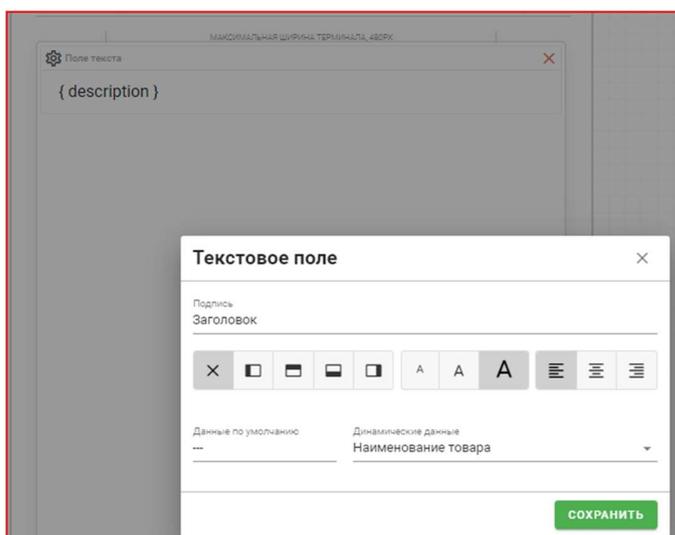
## Списочная форма «Хокку»

Добавляем форму и устанавливаем ее параметры:



Эта форма также будет оперировать данными из Таблицы товаров, но отображать эти данные форма будет в виде списка записей.

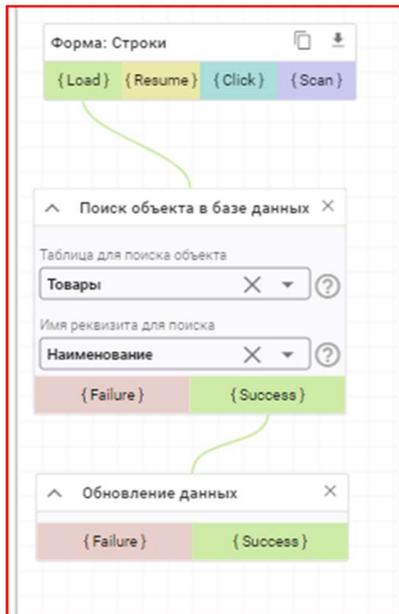
Все, что нам потребуется на этой форме, это – текстовое поле, которое будет показывать «Наименование товара». Строка за строкой. Без заголовка, чтобы ничто не отвлекало.



## Логика поведения.

### Форма «Строки»

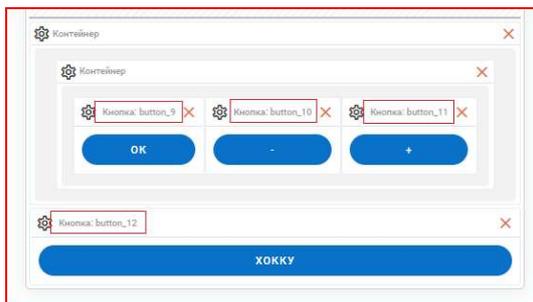
При открытии формы нам надо продолжать творить, если в прошлом подходе муза внезапно нас покинула. Поэтому, загружая форму, выводим первую запись из базы данных на экран:



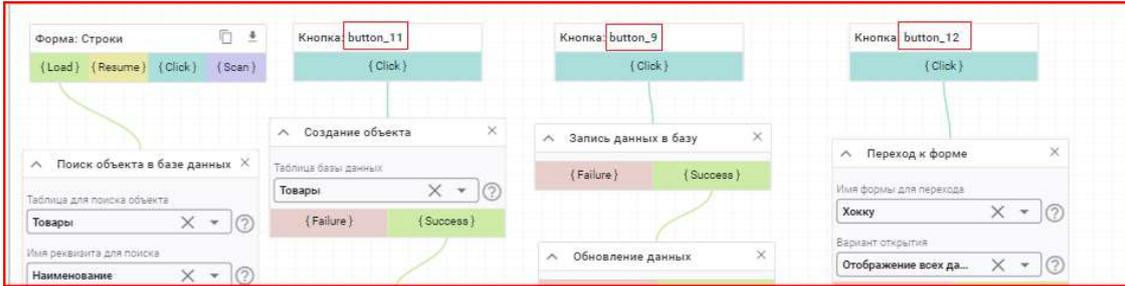
Если на вход инструкции «Поиск объектов в базе данных» мы не передаем значение, которое надо найти – инструкция нам покажет первую запись из таблицы.

### Немного про кнопки.

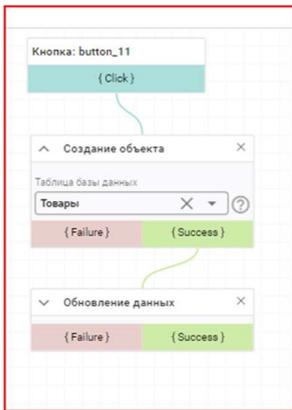
Каждая помещенная на форму кнопка, имеет свой уникальный идентификатор на этой форме:



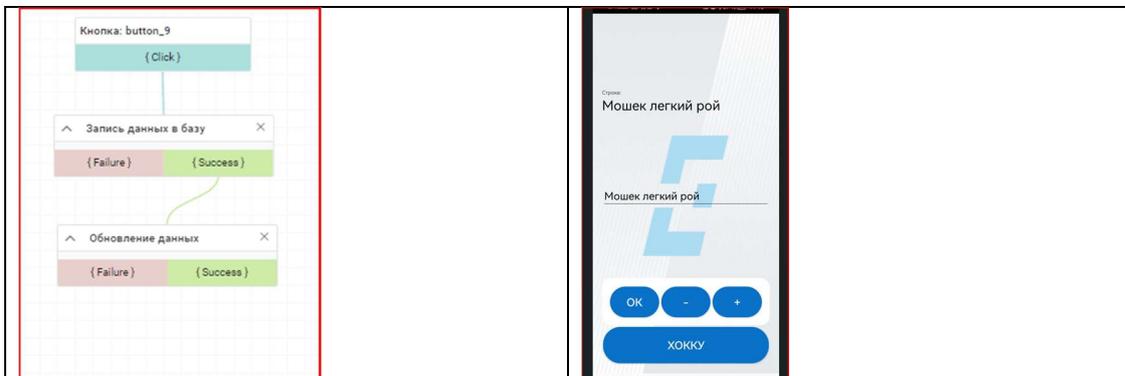
Точно такие же идентификаторы имеют инструкции на холсте действий. С этих инструкций всегда начинаются цепочки действий, которые программа выполняет в ответ на нажатие кнопки:



Для того, чтобы начинать творить, нам нужна пустая запись в базе данных, в которую мы запишем нетленку. Это должно произойти при нажатии на кнопку «+». Рисуем, не забываем обновить связь между данными и формой!

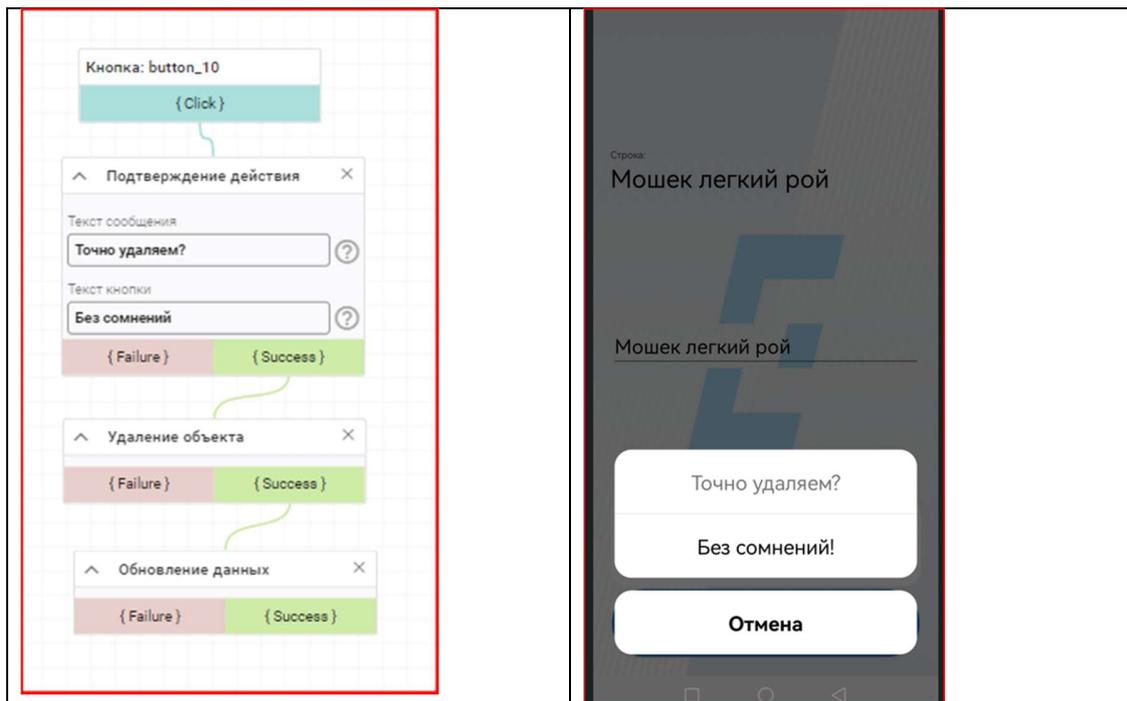


После окончания ввода, строку надо сохранить в базе данных. После выполнения этой инструкции, введенная строка будет сохранена и отобразится в текстовом поле в верхней части формы. Рисуем:

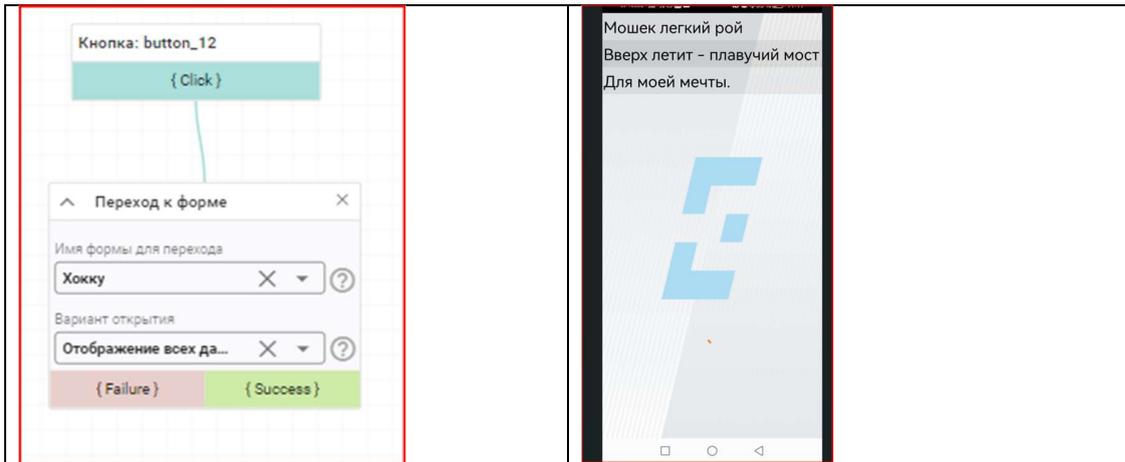


Данные (текст в нашем случае) можно редактировать сколько угодно раз, не забывая нажимать ОК для сохранения. Мы работаем с конкретной записью, и пока мы с нее не ушли кнопкой «удалить» или «создать» - эта запись будет у нас перед глазами.

Если то, что мы видим перед собой на экране, не удовлетворяет нас, безжалостно удаляем запись.



Для того, чтобы прочитать получившийся шедевр, переходим к форме «Хокку», отображая на ней все доступные данные, без дополнительной фильтрации:



### Форма «Хокку»

Так как форма предназначена только для отображения результата наших усилий, нам потребуется единственная возможность – редактировать ошибки ввода. Для этого, при клике на любую строку, мы открываем предыдущую форму и загружаем эту строку в нее:

