

JS-вайбкодинг в LogiNot:

Повышаем гибкость сценариев,
не погребая себя под непонятным кодом



Евгений Стучалкин

dm-lab.pro

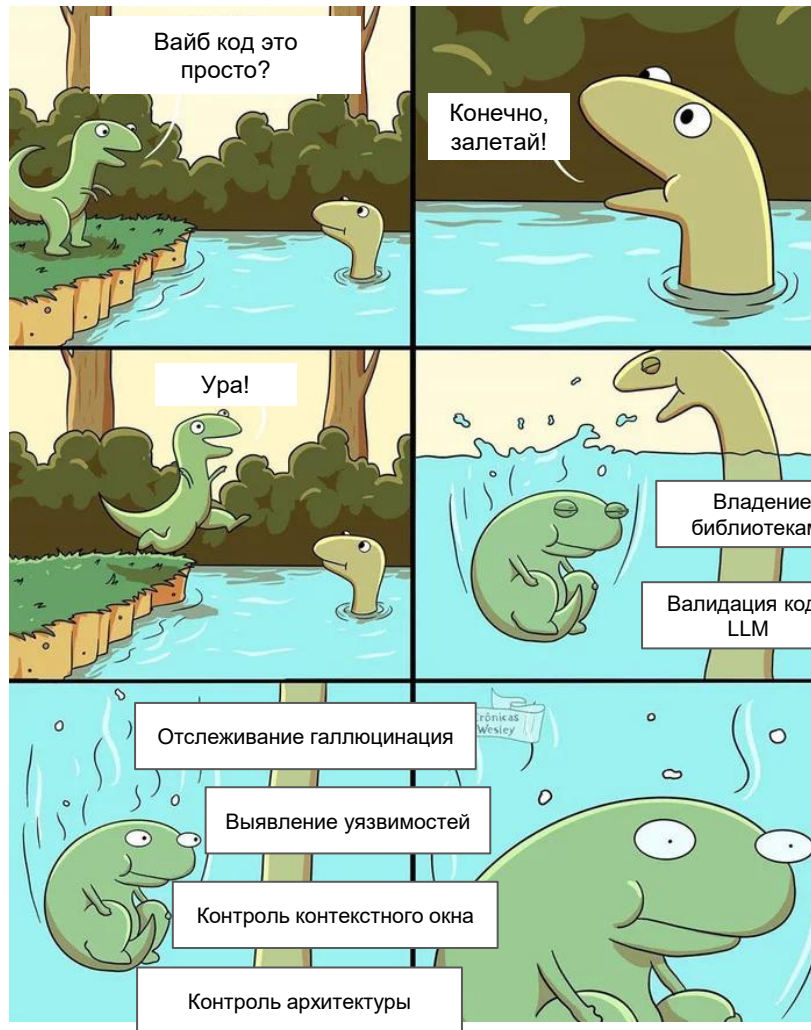
11+ лет опыта в VI и КХД проектах

50+ лично реализованных проектов

7+ лет подтвержденный срок актуальности архитектуры в развивающихся проектах

**ДОКЛАД - ПРОСТ
ПОТОМУ - КРАСИВ**





Почему именно JavaScript?

1. Нативно встроен в Loginom

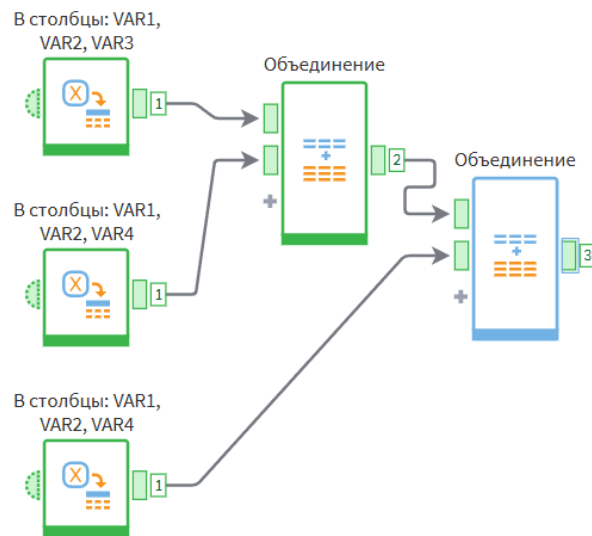
(не надо отдельно ставить на сервер, работает в песочнице, не надо согласовывать);

1. Фокус сценариев на “утилиты”, а не обработку данных.

Пример “Универсальное объединение”

В таблицах 2 и 3 есть общие поля, которых нет в таблице 1. А значит, одним объединением их не собрать.

Таблица 1			
Поле 1	Поле 2	Поле 3	
Таблица 2			
	Поле 2	Поле 3	Поле 4
Таблица 3			
	Поле 2		Поле 4



Пример “Универсальное объединение”

Общая таблица			
Поле 1	Поле 2	Поле 3	Поле 4
Таблица 1			
Поле 1	Поле 2	Поле 3	
Таблица 2			
	Поле 2	Поле 3	Поле 4
Таблица 3			
	Поле 2		Поле 4

Вот бы у нас была общая таблица со всем набором полей

Открываем ссылку, качаем промпт

<https://loginom.ru/blog/loginom-js-ai-generation>

Подготовлены два файла-шаблона запросов (prompt), предназначенные для разных версий платформы:

- Loginom 7.2: [скачать](#);
- Loginom 7.3: [скачать](#).

Погружаем LLM в контекст использования JS в Loginom

Ставим задачу LLM

The image shows a screenshot of a chat interface with a sidebar on the left and a main chat area on the right. The sidebar contains a search bar, a 'New chat' button, and a list of chat sessions. The main chat area shows a conversation with a file upload, a task description, and a code output.

Файл с промптом → loginom_js_llm_system... MD 85.5KB

Описываю задачу →

- 1. Без данных
- 2. Состоящая из уникальных полей всех входных таблиц, с сохранением имени, метки на основе таблицы которая подавалась первой, с сохранением типа данных на основе таблицы, которая сохранялась первой.
- 3. Если будут поля с одинаковыми названиями но разными типами на входных портах, вывести сообщение об ошибке с указанием этих полей и номеров портов.

Результат →

Вот реализация компонента для Loginom, который объединяет метаданные всех входных таблиц и проверяет конфликты типов:

```
javascript Copy Download

import { InputTables, OutputTable, DataType } from "builtin/Data";

// Хранище информации о полях
const fieldMap = new Map(); // key: имя поля, value: { displayName, dataType, sourcePort }

// функция для получения DataType в виде строки для сообщения об ошибке
function getDataTypeName(dataType) {
  const names = {
    [DataType.None]: "None",
    [DataType.Boolean]: "Boolean",
    [DataType.DateTime]: "DateTime",
    [DataType.Float]: "Float",
  };
}
```

Message DeepSeek

DeepThink Search

AI-generated, for reference only

Добавляем JavaScript-узел в сценарий, разрешаем создавать выходные столбцы

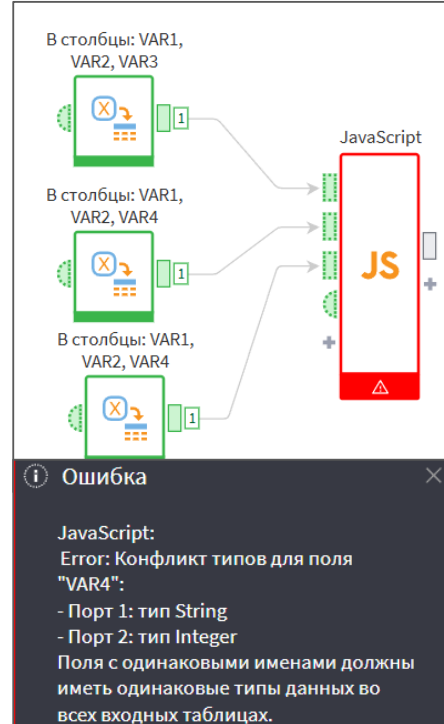
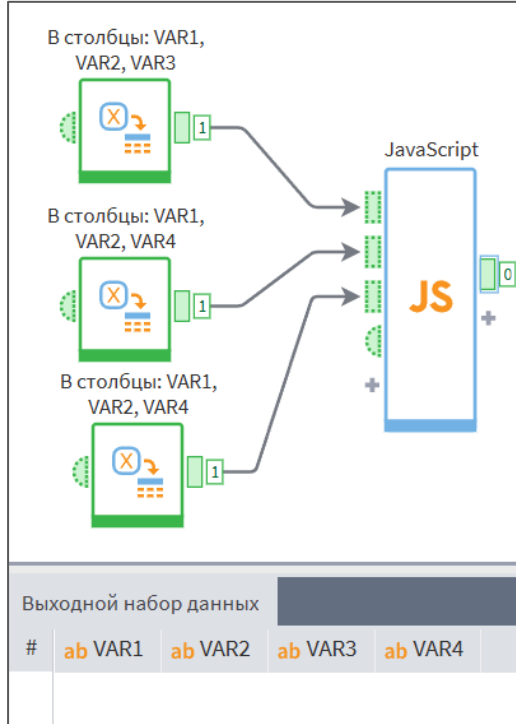


Настройка столбцов выходной таблицы JavaScript

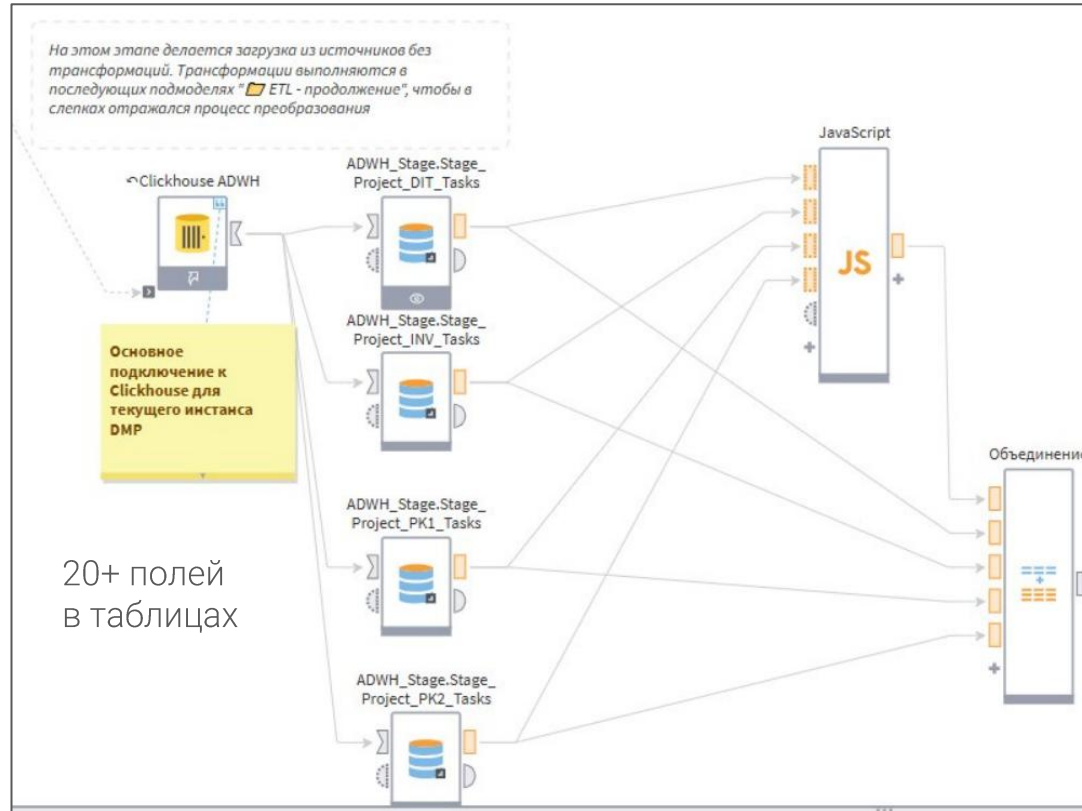
Разрешить формировать выходные столбцы из кода



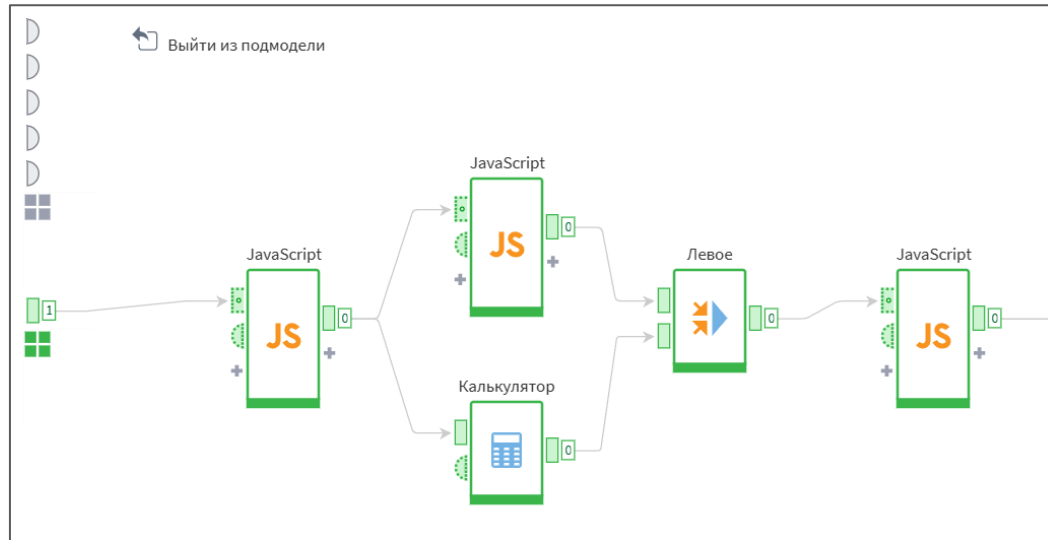
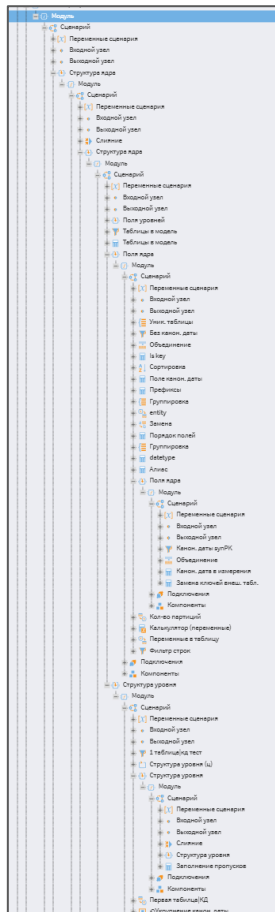
Вставляем код, наслаждаемся результатом



Один UNION, чтобы объединить всех



Динамическая генерация SQL-запросов. Десятки узлов становятся простой схемой

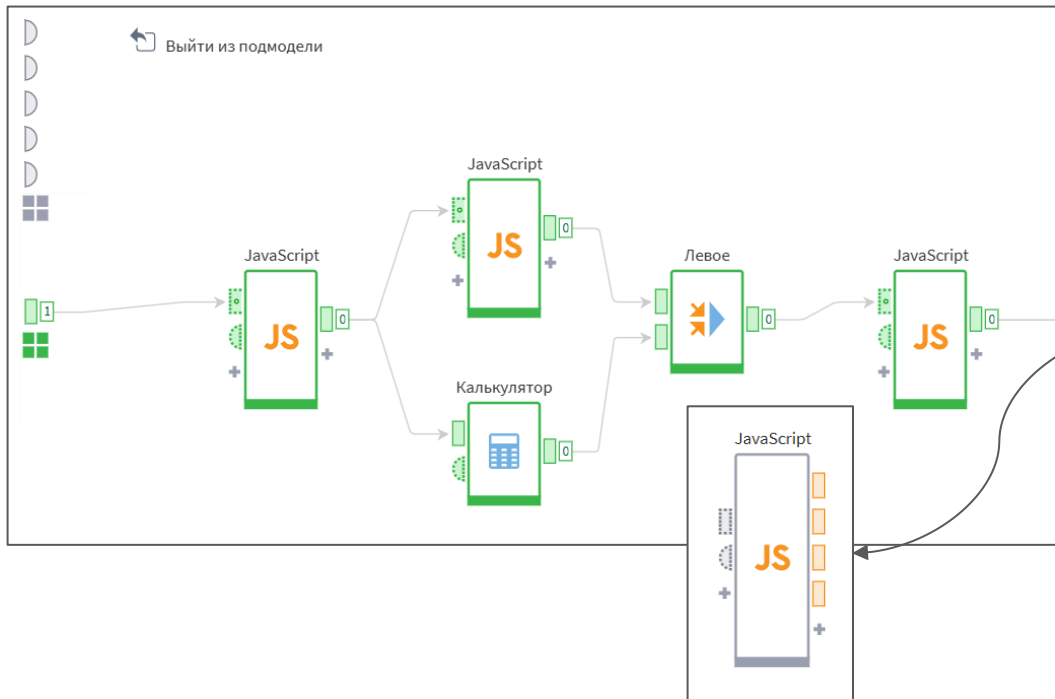


Ограничения применимости

Главный водораздел: объемы обрабатываемых данных;

- **Сотни строк, сложные преобразования** - однозначно да;
- **Десятки тыс. строк** - скорее да;
- **Сотни тыс. строк** - возможно да;
- **Миллионы строк** - лучше не стоит

Как избежать потери контроля над сценарием



1. Не пишем один гига-компонент;
2. Строго формализуем входные поля и переменные;
3. Разделяем логику на потоки;
4. Создаем отладочные выходы с промежуточными результатами при сложных трансформациях;
5. Сложные места, требующие точности, реализуем стандартными компонентами

Почему взлетело

1. Работает сразу, без кучи сервисов и настроек инфраструктуры;
2. Фокус на решении задачи, а не освоении новых подходов к разработке
3. Разбивка на блоки помогает сохранить контроль и прозрачность;
4. Возможность “доноукодить” свое, если нужно иметь полный контроль над логикой.

Ваши вопросы