

Отчет о пакете «it-conf-2026.lgp» (2026-08-26 14:33:49)

Общая информация о пакете «it-conf-2026.lgp»

• **Версия платформы:** 7.3.1

Описание пакета

Пакет включает шесть сценариев обработки данных с применением REST-запросов, переменных и генеративных моделей. Первый сценарий — стартовый шаблон без логики. Второй — параллельные ветви импорта файлов, работы с JSON, группировки и вызова внешнего API. Третий — подготовка и обогащение данных об объектах культурного наследия с использованием нейросети. Четвёртый — анализ данных общественного питания Москвы с фильтрацией и обогащением через внешний сервис. Пятый — стандартизация нормативно-справочной информации: формирование промптов, примеров и системных запросов для нормализации описаний товарно-материальных ценностей. Шестой — конвейер подготовки данных для модели машинного обучения: автоматическая генерация и расчёт новых признаков с помощью языковой модели. Сценарии покрывают типовые задачи: очистку, преобразование, обогащение данных и подготовку к анализу.

Ссылки на внешние пакеты

1. loginom_silver_kit. ^3.2.0
2. loginom_llm_kit. ^1.1.1

Статистика пакета

- **Количество модулей:** 6
- **Общее количество заметок:** 9
- **Общее количество узлов:** 92
- **Общее количество подмоделей:** 8
- **Общее количество узлов программирования:** 2
- **Общее количество узлов-ссылок:** 0
- **Общее количество производных узлов:** 12

Список модулей

- Модуль 1. «Пример сценария»
- Модуль 2. «[REST-API] Обработка данных о вакансиях с hh.ru»
- Модуль 3. «[ИИ + low-code] data.mos.ru - Объекты культурного наследия»
- Модуль 4. «[ИИ + low-code] data.mos.ru - Общественное питание в Москве»
- Модуль 5. «[ИИ + low-code] Стандартизация нормативно-справочной информации (НСИ) »
- Модуль 6. «[ИИ + low-code] Генерация новых признаков для ML-моделей»

Модуль 1. «Пример сценария»

Описание модуля

Сценарий представляет собой пустую (заготовочную) схему без узлов и связей. В рабочей области размещена только пояснительная заметка с текстом «# Пример сценария», расположенная в верхней части холста. Сама заметка не содержит исполняемой логики и служит лишь маркером-комментарием для пользователя.

Логика подмоделей

Подмодели отсутствуют, связи между узлами не заданы, поэтому описание ограничивается общей характеристикой: сценарий не выполняет никаких действий и предназначен, вероятно, как стартовый шаблон для дальнейшего построения логики обработки данных.

Статистика модуля

- **Глубина вложенности сценария:** 0
- **Количество заметок:** 1
- **Количество узлов:** 0
- **Количество подмоделей:** 0
- **Количество узлов программирования:** None
- **Количество узлов-ссылок:** None
- **Количество производных узлов:** None

Модуль 2. «[REST-API] Обработка данных о вакансиях с hh.ru»

Описание модуля

Сценарий содержит несколько параллельных ветвей обработки данных. Исходные данные поступают из двух файлов Loginom Data, набора переменных и внешнего REST-источника. Файл json_raw_data_loginom.lgd импортируется, преобразуется в дерево JSON, затем разворачивается в таблицы: одна ветвь обрабатывает Root.items, другая — Root. После разворачивания выполняются удаление полей, группировка и фильтрация строк по area.name и salary.to. Файл json_raw_data.lgd проходит похожий путь: импорт, JSON в дерево, таблица Root, удаление и группировка. Отдельная ветвь формирует переменные в таблицу (grant_type, client_id и другие) и передаёт их в REST-запрос через узел-ссылку. Итерационный цикл с фильтром по номеру строки экспортирует данные в Loginom Data файл. Также присутствуют автономный цикл и завершающие узлы обработки деревьев.

Подмодель «REST» содержит REST-запрос и узел «Параметры запроса». Входной порт подмодели передаёт параметры запроса, после выполнения запроса результат через выходной порт возвращается во внешний контур, где соединяется с источником «DREST: https://api.hh.ru/vacancies». Таким образом, подмодель инкапсулирует формирование параметров и выполнение REST-запроса.

Статистика модуля

- Глубина вложенности сценария: 1
- Количество заметок: 0
- Количество узлов: 23
- Количество подмоделей: 1
- Количество узлов программирования: 0
- Количество узлов-ссылок: 0
- Количество производных узлов: 4

Модуль 3. «[ИИ + low-code] data.mos.ru - Объекты культурного наследия»

Описание модуля

Сценарий посвящён обработке датасета «Объекты культурного наследия» с портала открытых данных Правительства Москвы (файл data-530-28-04-2026.csv). Данные проходят через две подмодели — «Подготовка данных» и «Обогащение данных». На входе также подаются переменная api_key (через калькулятор переменных) и таблица «В строки: Долгота (X), Широта (Y)» (переменные в таблицу). Итоговая цепочка: импорт CSV → «Подготовка данных» → фильтр строк «Номер строки» → «Обогащение данных» → фильтр строк «Автор = Шехтель» (терминальный узел). Внутренние узлы подмоделей дополнительно связаны с внешними источниками.

Логика подмодели «Подготовка данных»: входной порт направляет данные в узел «JSON в дерево: Centroid» (производный компонент), затем «Дерево в табл.: Root.Глобальный индекс, Root.type (+1)». Далее выполняется соединение «Дополнять до макс. набора, повторять набор данных», куда также поступает таблица координат из внешнего узла «В строки...». После этого строится кросс-таблица (cols: Имя, rows: Root.Глобальный индекс, cells: Root.coordinates), применяются «Параметры полей» (изменение) и слияние «Левое, Root.Глобальный индекс». Параллельно входной порт через «Параметры полей» (удаление) подаёт данные в калькулятор «ID, Тип объекта», результат которого участвует в том же слиянии. Выходной порт завершает подмодель.

Логика подмодели «Обогащение данных»: входной порт передаёт данные в производный компонент «Pmeta-Промпты. Датасет», затем в «Pmeta-GigaChat» (сюда же поступает api_key из внешнего калькулятора). Ответ обрабатывается узлом «Парсинг строки» (JavaScript) и через соединение «Дополнять до макс. набора» выводится в выходной порт. Исходные данные из входного порта также напрямую подаются в это соединение для дополнения набора. Результат фильтруется по автору «Шехтель».

Статистика модуля

- Глубина вложенности сценария: 1
- Количество заметок: 1
- Количество узлов: 19
- Количество подмоделей: 2
- Количество узлов программирования: 1
- Количество узлов-ссылок: 0
- Количество производных узлов: 3

Модуль 4. «[ИИ + low-code] data.mos.ru - Общественное питание в Москве»

Описание модуля

Сценарий начинается с импорта csv-файла с данными об общественном питании Москвы и переменной ключа доступа. Данные поступают в подмодель «Подготовка данных», затем в «Обогащение данных», после чего проходят фильтр «Подходит для деловых встреч» и завершают обработку.

Логика подмоделей:

«Подготовка данных»: на входе фильтрует записи по условию «НАО», удаляет лишние поля через «Удаление» и возвращает очищенный набор через выходной порт.

«Обогащение данных»: входной поток разделяется. Одна ветка идёт сразу в соединение «Дополнять до макс. набора». Другая ветка проходит через компонент промптов, затем через GigaChat, куда подаётся внешний ключ доступа. Ответ GigaChat фильтруется узлом «Ответ», дополнительно обрабатывается скриптовым узлом, и результаты соединяются через «Дополнять до макс. набора». Далее данные объединяются, сортируются по возрастанию идентификатора и через второе соединение «Дополнять до макс. набора» возвращаются на выходной порт подмодели. Внешняя переменная ключа доступа также подаётся в GigaChat.

Статистика модуля

- Глубина вложенности сценария: 1
- Количество заметок: 1
- Количество узлов: 15
- Количество подмоделей: 2
- Количество узлов программирования: 1
- Количество узлов-ссылок: 0
- Количество производных узлов: 2

Модуль 5. «[ИИ + low-code] Стандартизация нормативно-справочной информации (НСИ) »

Описание модуля

Сценарий автоматизирует стандартизацию нормативно-справочной информации на примере нормализации описаний ТМЦ. Основные входные данные: файлы ТМЦ, примеров и атрибутов, а также переменные (api_key, бренды СНГ, шаблон запроса и др.). Поток: из ТМЦ формируется датасет промптов, обогащается примерами и системным запросом, затем через GigaChat преобразуется в дерево и таблицу. Итог выводится терминальным узлом «Дерево в таблицу».

Подмодель «Примеры»: получает данные из Excel-файла примеров, внутри преобразует их в пары «пользователь/ассистент», транспонирует столбцы и корректирует поля, возвращая обработанные примеры через выходной порт.

Подмодель «Дополнительные инструкции»: использует переменные (бренды СНГ и калькулятор переменных), преобразует их в таблицы, вычисляет страну происхождения, объединяет результаты и подаёт как дополнительные инструкции. Входной порт не задействован — данные формируются из внутренних источников.

Подмодель «Системный запрос»: принимает данные атрибутов из Excel, внутри группирует и сливает их. Затем с помощью калькулятора и шаблона запроса формирует системный запрос, который через выходной порт уходит в основной поток. Дополнительные инструкции перед этим проходят через калькуляторы и группировку, связываясь с системным запросом.

Статистика модуля

- Глубина вложенности сценария: 1
- Количество заметок: 1
- Количество узлов: 31
- Количество подмоделей: 3
- Количество узлов программирования: 0
- Количество узлов-ссылок: 0
- Количество производных узлов: 3

Модуль 6. «[ИИ + low-code] Генерация новых признаков для ML-моделей»

Описание модуля

Сценарий представляет собой конвейер подготовки данных для ML-модели. Исходный набор данных загружается через узел импорта Loginot, параллельно активируется подключение к Ollama. Оба этих узла являются стартовыми и подают данные в узел «LLM. Генерация признаков». Далее сформированные признаки передаются в завершающий узел «LLM. Обогащение данных», который выполняет их расчёт.

Сценарий не содержит вложенных подмоделей, логика распределена по отдельным узлам обработки.

По заметкам: «Исходные данные» относятся к узлу импорта, «Подключение» — к узлу Ollama. «Запрос 1» соответствует первому обращению к LLM для генерации признаков, «Запрос 2» — второму этапу обогащения. Основная задача, зафиксированная в описании, — автоматически сгенерировать 5 новых признаков для оценки вероятности отклика на предложение по банковскому продукту. При генерации учитываются только структура датасета и первая строка данных. «LLM. Генерация признаков» обращается к LLM с подготовленным промптом и формирует признаки, а «LLM. Обогащение данных» рассчитывает их средствами Python и библиотек pandas и numpy.

Статистика модуля

- Глубина вложенности сценария: 0
- Количество заметок: 5
- Количество узлов: 4
- Количество подмоделей: 0
- Количество узлов программирования: 0
- Количество узлов-ссылок: 0
- Количество производных узлов: 0