АЛЬПИНА.ПЛЮС × ЖЕМАЛ ХАМИДУН

ТРАНСКРИПТ

Создание AI-агентов для data-driven решений в бизнесе

2025 · 1 ч 37 мин · С Павлом Дорониным · Бесплатный вебинар

Расшифровка аудио: Deepgram Nova-2, русский язык, уверенность распознавания 97.9%.
Абзацев: 124 · **Длительность:** 0 ч 57 мин

00:00:00 Так, ну все, я тогда вывожу презенташку. Сначала сделаю небольшую вводную и после этого пойдем в практическую работу. У нас сегодня будет такой прям воркшоп-воркшоп, то есть чисто практико-ориентированная работа. Вот Егор пишет: Чаты с ИИ уже не в новинку и вовсю используются, а вот с агентами пока нет работы. Да, спасибо, Егор, действительно.

00:00:22 И агент это такой 1 из трендов 2025 года в мире ИИ и кажется, что это вот самое хайповое, что сейчас в ИИ происходит. Денис пишет: Что такое ИИ агент и как я могу с ним развиваться вместе? Да, супер. Так, сейчас я тогда открою презенташку. Это что, Глеб, мы с тобой в таком жилом режиме диалога, так что любые мысли, когда возникают, ты просто смело вклинивайся.

00:00:52 Так, а я сейчас введу. А вот и Паша. Ага. Вот Паша, отлично, как раз пришел вовремя. Паша, привет.

00:01:05 Как слышно, видно? Мы тебя пока не слышим. Микрофончик надо включить. Ну-ка скажи что-нибудь. Сейчас видимо Паша перезайдет.

00:01:20 Так, ну что, у нас есть помимо того, что мы сегодня расскажем, поделимся в рамках лектория, вот у нас есть еще канал интересный. Часто, кстати, название мне очень нравится. Реально, очень часто дело в промпте. Вот мы так его и назвали. Спасибо большое за это нашей креативной команде.

00:01:38 Я все время с улыбкой вспоминаю название этого канала, когда что-то идет не так, с моделью что-то не получается, все время дело в prompte. Если вы хотите знать, что дело в prompte, тоже подписывайтесь на наш канал. Там много интересного мы разбираем на эту тему. Мы сегодня с вами поговорим про Даже не поговорим, поговорим только про небольшую часть в начале, а остальное время поделаем. Будем создавать и агенты для принятия data-driven решений в бизнесе, для того чтобы опираться на данные.

00:02:10 И сегодня агенты ручками прямо сделаем в течение часа примерно. Вообще Альпина, если вы вдруг не знали, нас раньше первый раз только пришли откуда-то, то мы в рамках Альпины лекторий проводим наш сегодняшний воркшоп, вебинар. Альпина 27 лет уже на рынке новых знаний, в том числе, и в теме искусственного интеллекта у нас тоже много книжек есть по этой теме, поэтому присоединяйтесь к нам, к подписке, к корпоративной библиотеке. У нас есть более, здесь написано 1100 книг, на самом деле 1 0

потерялся. На самом деле, их больше 11 1000 и разных вебинаров, аудио и прочего-прочего.

00:02:52 Вот сегодняшний вебинар, наверняка, вы смотрите либо из корпоративной библиотеки, либо из подписки Альпина Плюс, поэтому вы, соответственно, можете ознакомиться со всем ассортиментом, который у нас есть. У нас есть Alpine GPT это технологичная платформа, агрегаторы разных нейросетей, и у вас будет возможность, все кто с нами будут до конца получат в подарок доступ к этой платформе на 5 дней, где вы сможете пользоваться абсолютно без всяких ограничений всеми нейросетями, включая gpt, glot, gemino и любые другие. Так что welcome in board, будем очень рады вас там видеть. Почему я могу в этом разговоре поучаствовать с вами? Я SQL-pen digital отвечаю за продуктовый портфель, за стратегию развития и многое другое, все, что касается цифровых продуктов.

00:03:34 А еще вот я запускал собственно сам этот агрегатор нейросетей, который изначально был для внутреннего потребления, для ускорения издательских процессов, а вот сейчас мы уже с ним вышли на рынок. Вот, И я также выступаю как эксперт и партнер в iCommunity. Сегодня у нас гости нашего эфира как раз представители iCommunity, которые сейчас расскажут про себя. Ну и меня тоже не миновала участь, я завел свой канал Готовим яичницу. В общем, если вам интересно готовить яичницу, приходите ко мне тоже.

00:04:05 Ну и, Паш, тут тебе тоже мячик перекинул. С нами сегодня еще Паша Доронин. Паша стоял у истоков этой трансформации Альпины, был 1 из первых экспертов, кого мы пригласили, когда решили внедрять искусственный интеллект вообще в работу издательства. И вот Паша приходил, рассказывал, как правильно это делать, строить AI-коммьюнити внутри. Паш, я небольшое интро сделал.

00:04:30 Давай пару слов ты тоже давай. Работает ли микрофон? Да, работает. Ура! Всем привет!

00:04:39 Спасибо за приглашение за такое интересное зажорчество. Рад приветствовать всех всех здесь. Я являюсь таким одновременно и амбассадором в профессиональной среде в тему искусственного интеллекта, в основе смены и в работе. И ещё она, на самом деле, является фанатом внутренним. Я сам являлся таковым.

00:05:02 И на самом деле, не только я. У нас в сообществе много эффектов. Я вспоминаю Радвин детей, и Глеба, нашего инспектора. Надеюсь, что мы сегодня с вами поделимся тем, что вас вдохновит, порадует, и вы не только

таким позитивно заряженным, но ещё и прикладно полезным. Чуть попозже, когда будет возможность, я покажу еще пару слайдов про наш предстоящий сокатон, онлайн сокатон.

00:05:31 Вот то, что вы увидите сегодня, то, что Глеб вам покажет вместе с журналом, то, что мы будем обсуждать, у вас будет возможность получить бесплатно, возможность получить этому то, что вы сегодня увидите. Поэтому будьте внимательны, и мы тоже покажем слайды про этот хакатон, и будет отдельная ссылочка на регистрацию. Спасибо. Да, спасибо, Паш. Ну и Глеб наш нейромаг и волшебник сегодняшнего дня, который покажет сейчас то, как на самом деле все это работает, руками непосредственно пособирает.

00:06:08 Глеб эксперт и комьюнити, и, в общем, Глеб, тебе слово перекидываю тоже. Я в прошлый раз, наверное, представлялся достаточно подробно. Для тех, кто не был на предыдущем вебинаре по этой же теме, ну или из этой же серии, скажу, я в общем довольно давно в эмеле, еще до того, как это стало мейнстримом, еще до того, как его стали называть ИИ, в общем-то, потому что, ну, знаете, между собой мы обычно MLA говорим, а вот как только надо продавать, так мы сразу говорим и я, и все, Как-то вот так вот устроено. Я на самом деле не так давно. Просто в какой-то момент, знаете, стереотипы, что ли, думаешь: Ну, я же точно лучше кот пишу, и я его быстро очень пишу.

00:06:50 Я в этом бизнесе ого-го, сколько, сколько лет. А потом вот так пробуешь, пробуешь, опа, интересно, а ведь быстрее получается и в общем потом подшлифовать какие-то мелочи быстрее, чем написать самому. И как-то я так в это всё погрузился, мне понравилось. Наверное, от хардкора, которым я занимался раньше, но я не скажу, что я от него отошел, я продолжаю заниматься всем, что связано с промышленностью, с данными дистанционного зондирования. Но раньше это было 60 на 40 промышленности, всякие спутники и прочее, а сейчас, наверное, 45 на 35 на 20, и 20 это вот как раз то, о чём мы сегодня будем говорить в разных форматах и поделать, и поучить, и рассказать.

00:07:43 Глеб, спасибо большое. Так, ну что, давайте двигаться дальше. Мы быстренько сейчас пробежим всю вводную часть, как наш формат сегодняшний воркшоп 1 час, раздел вопросы-ответы, около 30 минут, может быть чуть поменьше. Пишите, пожалуйста, ваши вопросы прямо по ходу дискуссии в чат. У нас будет такой формат демонстрации и живого диалога на фоне того, что мы будем собирать агента, поэтому вы можете в любой момент писать ваши вопросы, я их буду видеть и в рамках модерации буду задавать эти вопросы Флу, Глебу.

00:08:16 Мы будем в таком живом формате двигаться. Кто будет с нами до конца, получит подарки, про это уже говорил. Используете ли вы искусственный интеллект? У нас частично уже в начале было, про агентов тут писали. В общем, тоже будет здорово, если вы поделитесь немножко.

00:08:31 Мы все время делаем этот срез, этот слайд кочует из презентации в презентацию, потому что мы хотим понимать, как вообще тренд выглядит. Когда-то давно, когда мы только начинали эту тему еще в 23 году, почти все писали не используй, но очень хочу. Сейчас уже очень многие пишут каждый день. Прям видно, как меняется этот срез. Наверное, если выгрузить все ответы участника за все это время и проследить динамику, будет очень интересна такая кривая вверх.

00:08:59 Интересно будет сейчас от вас тоже узнать. Напишите, пожалуйста, можно даже цифры просто от 1 до 5, где 1 это каждый день, а 5 вам может быть вообще тема неинтересна. Случайно пришли на название темы. Когда работал, то каждый день. И сейчас пару раз в неделю.

00:09:19 Каждый день 1. В основном, конечно, сейчас уже кажется аудитория. Ну и, наверное, тема агентов она такая в общем, про каждый день использования. Эволюция, если так ее очень кратко проследить, то вообще-то с 1966 года все начало развиваться. Тогда был первый чат бот Элиза.

00:09:37 Она имитировала, эта программа разговор с психотерапевтом, используя простые правила, так что, в общем, все очень давно в этой теме развивается. Ну а сейчас как раз наступила такая эра и ассистентов, и агентов, полноценных систем, которые могут проактивно выполнять разные действия. Для понимания этого сейчас, конечно, как мир устроен уже с точки зрения этих всех навыков, то можно просто на эти цифры посмотреть. Это данные из абк трендов. Там, на самом деле, гораздо больше цифр, которые могут объяснить, почему стоит в эту тему сейчас инвестировать свое время.

00:10:11 92% крупнейших мировых компаний уже используют ChatGPT, и, конечно, все больше и больше компаний пытаются внедрить это в свою работу, но поскольку просто в бизнес чат-боты это не совсем релевантно внедрять, то вот как раз сейчас тема с агентами пошла, и это непосредственно то, что бизнес пытается реализовывать внутри своих компаний. Я знаю много крупных корпоратов, которые уже начали собирать своих агентов прямо во внутреннем контуре. Ну и, чтобы вы понимали, в 10 раз увеличилось количество вакансий со знанием нейросетей, только за прошлый год продолжает эта цифра расти, следует из данных HeadHunter. Зачем нужны EA агенты? Вот этот слайд, наверное, мог бы нам все объяснить.

00:10:52 Вот когда-то были двое изларца. В общем, люди всегда хотели, чтобы какая-то штука что-то делала за них, а они могли ничего не делать. Кажется, это желание нас не покидает никогда за всю историю человечества. Вот мы подошли к тому этапу, когда это становится реалистичным, и действительно и агенты это могут делать. Если посмотреть отличия, то чат-боты это простые такие диалоговые с заранее заданными сценариями, они дают ответы быстрые, но у них очень ограниченная гибкость.

00:11:22 Есть уже такое понятие как ИИ ассистенты, они уже более прокачаны, то есть это умные помощники на основе таких больших языковых моделей. Они понимают контекст, генерируют текст, могут быть кастомизированы под разные атомарные задачи, но они не работают проактивно. Появились и агенты не так давно. Такая сущность, которая работает на базе конструкторов, ну или в некоторых случаях кодом можно ее написать, и они по разным триггерам могут активироваться и уже выполнять действия автономно. То есть, агенты отличаются тем, что это автономные системы, способные выполнять действия, принимать решения и интегрироваться с разными системами, благодаря этому.

00:12:03 Вот у нас был, собственно, кейс альпины запрос бизнеса, ускорение создания коммерческих предложений. Это была первая история, с которой мы начинали нашу серию вебинаров. Сегодня у нас будет несколько другая задача. Мы тоже в эту тему активно идем. Наверное, в этом месте, я думаю, что я прервусь, чтобы нас всякой теорией и базой не забивать, и мы пошли дальше.

00:12:29 Мы сейчас будем это делать на базе конструктора N8N или N8N, как его еще тоже могут иногда называть. Это, по сути, такой визуальный редактор, который позволяет без программирования создавать разные цепочки. Он имеет огромное количество интеграций для подключения к популярным сервисам и программам и имеет встроенные и узлы, то есть, когда вы можете подключить ту или иную нейросеть для связки с какой-то программой. А также можно там использовать разные триггеры: запуск расписанию, какой-то вебхук из другой системы, разные события и так далее, так далее. И, конечно, обработка данных, то есть он может сохранять себе что-то в базе данных и так далее.

00:13:13 Это можно разворачивать как локально внутри своей компании, внутри своего контура или на каком-то своем сервере, так и по подписке, то есть у них есть подписка и приобретая подписку можно дальше собирать там определенные вычисления, но даже бесплатно это тоже работает. В этом месте, наверное, Глеб, я останавливаю свою демонстражку и передаю слово тебе. Ты, наверное, как раз про это подробно сейчас расскажешь. А я вижу

вопрос в чате способность к самообучению не является ли критерием агента. В принципе, в некоторых случаях может являться, но совсем не обязательно, что он будет, например, дообучаться и менять свой сценарий работы.

00:13:52 То есть он может быть довольно заскриптованным с точки зрения того, как он работает, а может, например, накапливать базу данных и на основе накопленной базы данных как-то дальше менять свой промт или подход к работе. Ну тут Глеб, наверное, тоже сможешь чуть-чуть дополнить меня. Да, смогу и так, наверное, попробую приземлить этот вопрос. Я вообще не люблю все эти споры о терминологии, потому что понятно, что здесь терминология, мягко выражаясь, плавает еще. Мы определяем все, как хотим, что такое агент, что такое ассистент, что такое чат-бот, чем вот они все друг от друга отличаются.

00:14:29 И в конечном счете, ну это такое очень плавающее понятие, а ИИ это настоящее, это что такое? Это спор филологический, его можно где-то вести в каком-то контексте, но мне кажется, нужно практические задачи решать, лучше так и подходить. Здесь, наверное, небольшое отступление сделаю по поводу обучения, самообучения. Многие, наверное, уже заметили, что FineTuning куда-то ушёл очень далеко на задворки. Вот когда модели только появились более или менее такие работоспособные, большие языковые, чтобы под что-то свое их подтюнить, нужно было их дотренировать.

00:15:09 Это долго, дорого, сложно. Потом все сказали параллельно с этим кто-то говорил, надо промтить правильно, надо данный поиск подавать. Потом появился RUC, то есть Travel agmented generation, то есть когда у нас есть база документов векторизованная, мы вытаскиваем похожие документы, подаем их, добавляем их, вернее, в контекст промта и говорим модели отвечать по тому, что мы ей дали, а не вообще. И оказалось, что для большинства задач вот это вот лучше, чем FineTuning работает, и при этом оно намного дешевле, проще и легче организовать. Тоже это не тривиальная задача, но это достаточно обозримая задача по сравнению с fine-tuning.

00:16:00 Фантюнинг это вот прям не хорошее дело. Иногда нужно, но зависит очень сильно. И вот тут вопрос самообучения, что тут понимать самообучения. Ну, то есть эти модели настолько большие, что они для обучения требуют каких-то колоссальных ресурсов. Они же не будут под вас 1 самообучаться, дообучаться.

00:16:20 Поэтому мы можем как-то это эмулировать, мимикрировать, что она вроде бы вас знает. Но в конечном счете это просто сохранение истории переписки и добавление ее в контекст текущего разговора, по большому счету. Вот и

все самообучение. Выглядеть оно может действительно, что она вас узнает. Но контекст у этих моделей не бесконечный, даже у самых больших.

00:16:43 Поэтому в какой-то момент она забудет, о чем вы говорили, какое-то время назад. То есть может казаться, что она вроде бы как самообучается, тюнится, но на самом деле технически это не так. Так, ну чего, наверное, поедем, да? Да, давай за что. Как в прошлый раз я рассказываю.

00:17:08 А я буду, говоря себе. Ты смотришь чат, потому что я в него смотреть не успеваю и прерываешь меня, когда есть какие-то хорошие вопросы. Да, Паша, Паша тоже что-то говорил в моменте, но мы его перебили, мне кажется, Паша. Лев говорит, ну что, я буду как обычно делать, я буду как обычно комментироваться, я буду как обычно внимательно наблюдать, слушаться. У нас отличная команда.

00:17:39 Я сразу скажу, я немножко простужен, поэтому я иногда могу кашлять, я тогда буду микрофон выключать, чтобы откашляться. Это может случиться, ничего необычного в этом нет. Окей, давайте я сначала чуть-чуть слова какие-то приговорю, вообще о чём сегодня будет идти речь, что мы тут называем datadriven, и не слишком ли громкие слова. Вот смотрите, когда вы заходите в ChatGPT или в Клод, любую другую модель с помощью браузера и начинаете с ней переписываться, она ничего не может сама сделать. Она может сгенерировать новый текст по тому тексту, который вы ей написали и в вашей предыдущей переписке.

00:18:21 Она не может пойти там куда-то в базу данных, где-то что-то взять по какому-то API, в общем, ничего она не может. Максимум, что она может это воспользоваться поиском. Большинство моделей поиск уже встроен. Плюс вы спарсите какие-то документы, которые вы загрузите. Для ситуации, когда это ручками делается, это вроде как бы ок, потому что если вам нужно какие-то данные просуммировать, вы их сами руками вытащили, сами в окошко браузера каким-то образом подали, неважно каким, в виде файлика загруженного или прям скопировали текстом, не суть важно.

00:18:53 Как бы, ну лишняя работа, но ничего страшного. А вот если вам нужно это автоматизировать, то тут начинаются проблемы, потому что все это большие языковые модели. Они в базе умеют чего делать? Вы им дали 1 текст, они его продолжили другим текстом по определенным принципам, ну там чтобы это правдоподобно звучало, это продолжение и так далее, так далее. Они не знают, что такое пойти к базе данных обратиться, пойти по API обратиться.

00:19:20 Они этого всего не понимают сами по себе. Поэтому для создания агентов, то есть систем автономных, которые чего-то делают, к ним что-то прилетело,

какой-то запрос, какие-то данные, еще что-то, они должны на основании этих данных что-то сделать, причем очень часто решить, что сделать. Здесь вот мы в прошлый раз об этом не говорили, мы создали такой pipeline, который в общем не подразумевает вариабельности каких-то разветвлений и так далее, причем разветвлений, основанных на решении самой модели. Очень часто они могут решать, что делать: пойти туда или пойти сюда, сделать вот это или вот это. И вот тогда их агентская сущность проявляется наиболее ярко.

00:20:03 Так вот, если у нас автономная система, нам нужно каким-то образом дать ей возможность общаться с внешним миром. Самой что-то делать, самой пойти взять где-то какие-то данные, самой какую-то страничку там распарсить, получить из неё информацию, самой куда-то отправить какое-то сообщение, создать какую-то заявку, там задачку куда-нибудь бросить в Trello, Asana или кто, что любит. Записать в Google Doc что-нибудь или в Google Sheets, или еще куда-нибудь. Обычный чат, то, что называется ChatGPT, такая еще вечная путаница, если говорят ChatGPT, ChatGPT, но ChatGPT это же вот то, что мы в браузере видим, а сами-то модели, они же не ChatGPT, а не просто GPT. Они доступны для автоматизированного использования без всякого пользовательского чата.

00:20:53 Вот они с некоторого времени и GPT, и Cloud, и DeepSig, и большинство других современных моделей имеют специальную, грубо говоря, розетку, куда вы ей можете подключить что-то, что позволит ей выполнять какие-то операции во внешнем мире. В общем, это называется tooling или function holeing, в зависимости от терминологии в разных, в зависимости от того, на каком уровне абстракции что ли мы это рассматриваем. То есть это некоторые, грубо говоря, устройства, которые говорят: Вот я умею делать то-то, то-то, вызывай меня, когда тебе нужно то-то, то-то. И модель умеет это делать. Вы подаете prompt, она говорит ага, а я ответить на это сама не могу, мне для этого нужно, ага, вот у меня есть для этого инструмент.

00:21:47 Она идет в этот инструмент, дает в него что-то на основании вашего промта. Инструмент выполняет свою работу, дает какой-то результат, этот результат добавляется к промту, но мы это увидим все. Я сейчас как бы на словах рассказываю, это все увидим вживую. Добавляет это к промпту и потом уже генерирует финальный ответ для пользователя. Модель может сходить в базу данных это то, чего ChatGPT не может и никогда не сможет просто по определению того, что это такое.

00:22:15 Вот Когда вы в браузере это делаете, это невозможно в принципе сделать. Обратиться куда-то по API, где-то создать В общем, по API большинство сервисов позволяют делать практически все: вытащить звонки, создать встречу с зуме, ну, в общем, практически любую операцию можно сделать.

Модель может решить, что ей нужно это сделать. Для этого ей просто нужно знать, какие инструменты у нее есть, по большому счету, и что каждый инструмент делает, и как с ним обращаться. И вот на самом деле все, что мы сегодня будем делать, будет крутиться в основном вокруг этого.

00:22:52 Как нам научить модель общаться с какой-то внешней средой? У нас качестве внешней среды будет реляционная база данных, прям простая, там 1 табличка всего, мы сейчас это всё разберём. Сначала мы разберём сам туллинг, а потом уже как бы большую задачу сделаем. На самом деле, финальный workflow в n8 будет очень простой, потому что я сегодня не столько n8 сам по себе и построение workflow в нём хочу проиллюстрировать, сколько вот эту вот главную ноду EA Agent, и как она вот работает потому что у нее там 3 гнезда 1 для модели другое для памяти это нам пока не интересно это мы там где-нибудь на 1 из следующих вебинаров разберем и третье для инструментов И вот нас будет интересовать, в прошлый раз нас интересовало первое гнездо, куда мы модельку подключаем, а сейчас нас будет интересовать гнездо, куда мы инструменты подключаем. Вот, это у кого-то, микрофон включен, пошло это самое.

00:23:52 Да-да-да, это я в моменте включил. Сейчас плохо стало, да? Сейчас стало эхо. Да нет, сейчас нормально. Нормально нормализовалось, окей.

00:24:00 Вот, ну на самом деле, я тут что могу только добавить, что, конечно, то, что мы сейчас покажем, это только как бы 1 из граней агентов. То есть, мы покажем просто базовый workflow, что есть такой инструмент, что там можно делать вот так, но дальше сфера, вот, как бы, где вы можете это применить, она, ну, конечно, не безгранична, но близко к этому, с учетом того, что появляется все больше и больше разных коннекторов, все больше появляется разных уже готовых сконструированных агентов кем-то другим, т. Е. Есть целый большой маркет уже заранее созданных агентов, всегда можете зайти, скачать эту ноду, посмотреть, как она была сделана и сделать что-то похожее для себя, переиспользовать частично. В общем, это открывает очень большие возможности по реальному применению искусственного интеллекта в бизнесе.

00:24:51 Но я думаю, Глеб, наверное, не будем томить, мы уже максимально интерес и фокус внимания, думаю, собрали, теперь надо показать. Давайте, наверное, я расшарю экран и поедем. Так, весь экран. Да, ну, и напомню всей аудитории, что вы можете по ходу писать, задавать ваши вопросы в чатик. Я их буду видеть, если что, задавать их по ходу Глебу.

00:25:18 Вот и в общем-то ваши любые мысли опишите. Ну и напомню всем, кто до конца с нами будет, получит еще подарок доступ к платформе нейросетей. Сейчас должно быть видно мой экран, правильно? Мы начнем прям с очень простой, прям вот игрушечной-игрушечной штуки. Я хочу, чтобы мы сегодня сделали что-то небольшое, но функциональное и очень понятное, что вот прям любой из тех, кто слушает, мог взять да повторить.

00:25:46 Я не хочу тут городить что-то очень сложное, что потом там полдня разбираться, что там такое нагородили. Я сначала объясню, как туллинг работает вот прямо на супер простом примере на примере генерации случайных чисел. Я хочу модели сказать дай мне случайное число в интервале от и до, и она должна мне его дать. В принципе, большинство моделей с этим и так справляются без проблем более или менее. Но мы это сделаем как бы так, чтобы это было наверняка.

00:26:15 То есть мы добавим инструмент, который генерирует и который модель будет понимать, что надо использовать, когда вот такой запрос от пользователей пришел. Так, я создал новый workflow. Давайте его как-нибудь обзовем. Alpino номер 2 Random Numbers. Вот, например, так.

00:26:36 Как и в прошлый раз, мы будем использовать встроенный чат. Просто для простоты можно Telegram подрубить, еще что-нибудь подрубить. Не в этом суть, поэтому использовать мы будем просто встроенный чат OnChat Message. Для тех, кто в прошлый раз пропустил, я напомню, что в natn всё устроено на соединении разных нод, как они называются. У ноды есть вход и есть выход.

00:27:02 Вход это обычно выход какой-то другой ноды и, соответственно, выход ноды становится входом какой-то следующей или нескольких следующих. У входа и у выхода есть 2 части просто JSON, то бишь какие-то данные структурированные и бинарные данные. Сейчас бинарных данных не будет, забыли о них. У нас всё только JSON. Это как бы совсем база.

00:27:23 Единственная нода, у которой нет входа, это триггерная нода, которая стартует весь workflow. У нас будет он стартовать от сообщения в чате. Окей, всё вроде пока понятно. Главное, что мы должны добавить это, собственно, наш агент. Вот тут есть такая замечательная нода под названием AI Agent.

00:27:43 И вот она-то и будет тут всем заведовать. Давайте посмотрим. Да, кстати, по ходу пока ты как раз все это добавляешь, задали вопрос, объясните, где мы находимся, в какой программе. Мы находимся в конструкторе N8N, это конструктор, который вы можете в принципе, он доступен из России, можно зайти и посмотреть. Единственное, есть некоторое ограничение на то, чтобы

пользоваться платной версией, оно заключается в том, что вы для того, чтобы оплатить эту программу конструктор, нужно иметь зарубежную банковскую карту, но есть и опция развертывания на своих серверах.

00:28:23 Если вы представитель компании, хотите внедрить это в свой бизнес, то вы можете скачать некую серверную версию и разместить ее у себя. Так что в этом смысле то, что показывает Глеб, это развернуто непосредственно на сервере AI комьюнити. Сейчас я тут как раз вот об этом хотел сказать. Я сейчас локально это делаю, так получилось, AI community вот такой вот. Если чего пишите, разберемся, если вы захотите.

00:28:52 Да, кстати, перед тем случаем В рамках Cathona, который будет в ближайшее время, вот про который Паша рассказывал, там будет предоставляться сервер N8n, который есть у AI комьюнити, и вы сможете там все это собирать. В рамках Katon мы даем это сервер бесплатно. Я скинул ссылочку в чат, мы не только делаем обучение, но еще и сам сервис даем. Так, Глеб. Едем дальше.

00:29:21 В прошлый раз, для тех, кто был в прошлый раз, для тех, кто не был, сейчас я объясню. Мы использовали не эту ноду, мы использовали Basic LLM Chain. Это та нода, которая просто умеет выйти EEPROMPT, она вам ответ из какой-то модели, больше ничего она не умеет. В NEITEN есть вот такая большая большая нода с кучей выходов под названием AI Agent. Модель вот здесь Chat Model это только 1 её элемент.

00:29:46 Здесь ещё есть память, о ней пока не говорим, и вот второй выход, который нас интересует это tool. Значит, чего мы хотим? Мы отправили сообщение: Дай мне случайное число от и до И вот этот агент должен всё это вместе скоординировать, то есть понять, что Ага, мне нужен инструмент. Вызвать этот инструмент правильно, потому что от и до это диапазон. Его нужно вытащить из промта 2 числа.

00:30:13 Вызвать инструмент, взять его ответ и потом сформировать ответ для пользователя, который мы вот здесь в чате и увидим как ответ. Давайте первым делом модельку добавим. Здесь я буду использовать вот такой вот сервис под названием Open-Router. Это не принципиально, но я просто хочу показать, что для прототипирования такая штука есть. В сущности, это интерфейс, куча-куча-куча разных моделей.

00:30:41 Там цена как бы зависит дешевле дороже, то есть дешевле, чем напрямую запрашивать какие-то дороже. Но там, во-первых, есть много бесплатных моделей, мы сейчас как раз 1 из них и используем. А во-вторых, у вас 1 точка

входа. Это сильно упрощает жизнь. Там опять же есть какие-то способы правильно его пополнять.

00:31:04 Я о них не буду говорить. Это, наверное, не то место, где о них стоит говорить. И в конечном счете вам это просто упрощает жизнь. Вы можете 1 модель заменить на другую. Но сейчас увидим, как модельки выбирать.

00:31:16 И главное, что он внутри ещё умеет каскадом выбирать модели. Вы можете отранжировать несколько моделей и он будет выбирать сначала самую дешёвую, если она недоступна более дорогую, если она недоступна ещё более дорогую и так далее. В общем, это какой-то такой способ немножко может быть сэкономить в каких-то нагруженных пайплайнах. Вот я буду использовать его. У него точно так же, как у OpenAI или Cloud есть ключ API, который позволяет по известному протоколу к нему обращаться, по OpenAI, собственно, протоколу.

00:31:50 Поэтому подключим мы здесь вот эту модель или напрямую OpenAI Chat Module, или напрямую Untropic Chat Module. В общем, как бы работают они все одинаково. Просто Open-Router это как бы хаб, куча разных моделей. Почему я его использую? Ну, просто, чтобы показать, что вот какие-то довольно большие, довольно разумные модели можно даже использовать бесплатно.

00:32:12 Здесь в Credential, опять же, просто ключ API, больше ничего не нужно, как и в OpenAI. И понятно, что он у меня уже заведен, чтобы не копировать его и не светить ключ на всех. Нам нужно только модельку выбрать. Вот давайте выберем модельку. Ну вот те, которые просто, они стоят каких-то денег.

00:32:31 Мы сейчас сделаем такую интересную вещь. Мы посмотрим, а что тут есть бесплатного. И возьмём, ну, вот, например, DeepSeek 3.1 бесплатный. Он может иногда глючить, возможно, мы на это даже нарвёмся. Надеюсь, что нет, он не так часто глючит, но в остальном он бесплатен, то есть вы ничего за него платить не будете.

00:32:56 Есть провайдер, который его хостит. Это, в сущности, тот же DeepSig, который вы видите в чате. Не Reasoner, а обычный. Здесь, в принципе, всё. Пока что.

00:33:09 Пока что здесь всё. Мы потом, наверное, добавим системный промпт. Вот это мы тоже сразу соединим. И давайте вот сейчас пока возьмем и просто потестируем, а что получится. Переключимся на русский.

00:33:31 От 10 до 20, например. Он чего-то там подумает-подумает, и он на самом деле справится. Большинство моделей с такой простой задачкой

справляются, но на этой задачке очень легко обкатать, что такое туллинг и как он работает. Давайте сделаем так, чтобы не он сам, Deep Six 3.1 придумывал случайное число, а мы его придумывали и модели в ответ давали, а она уж формировала ответ. То есть мы хотим сделать так, чтобы она решала: Ага, мне нужен вот этот инструмент.

00:34:06 Этот инструмент имеет то-то, то-то, обратиться к нему нужно так-то, так-то. Окей, я его вызываю, я получаю ответ, я отвечаю пользователю. И мы эту цепочку прямо вот здесь вот увидим и посмотрим, как она на самом деле устроена. Вот в этом ключевая сущность этого примера. Если мы кликнем вот туда вот на плюсик выхода tool, нам NATN предложит какое-то большое количество тулов.

00:34:33 Основные 3 встроенных это вызвать другой workflow в самом annate. То есть мы можем вызвать какой-то большой сложный workflow в самом annate. Ru и забрать из него результат. Это нам не нужно. Ну, то есть, наверное, я немножко дополню, что можно по сути собирать такой каскадный конструктор, где 1 кусочек в 1 workflow собрали, workflow такой рабочий процесс получается, бизнес-процесс.

00:34:58 И таким образом бизнес-процессы можно друг с другом связывать, где у вас 1 часть 1 месте, потом вы к ней коннектор другой подвязали и так далее, и так далее. Прелесть в том, что Тулом может быть другой ИИ агент. То есть, у вас может быть каскады ИИ агентов. Это, наверное, не будет супер быстро зависеть от модели, конечно, но это возможно. Ну, когда сложное какое-то дерево принятия решений, его сложно в 1 Е агент добавить, но можно как бы действительно Да, но у меня, кстати, на эту тему, прям пример даже есть готовый.

00:35:36 Вот мы делали бота библиотекаря, и у нас там такая нетривиальная задачка была сделать так, чтобы бот библиотекарь не придумывал книги, которых нет. И валидацией ответа служит, по сути, другая модель, которая проверяет, действительно ли соответствует этот ответ базе данных. То есть, грубо говоря, есть агент-контроллер, который проверяет, что другой агент говорит. Вот примерно так. Ну, и там могут быть разные другие сценарии, но, по сути, там 1 модель контролирует другую.

00:36:08 Примерно так. Это, на самом деле, паттерн в общем даже описанный. Да. Особенно в применениях. Вот мы дальше, когда будем основной пайплан делать, там у нас будут запасы к базе данных и там вот это уже становится критичным.

00:36:22 Можно же напрямую SQL генерировать, это вот не совсем Я, кстати, Глеб, периодически, если позволишь, буду иногда чуть-чуть переводить, потому что вот я тоже вижу, нам в чатике пишут, потому что, пожалуйста, объяснять понятным языком. Но будем еще максимально, насколько это вообще возможно, учитывая довольно сложную техническую тему. Ну, здесь сложно всегда попасть в аудиторию, потому что заранее ведь неизвестно, где мы окажемся. Поэтому я вроде стараюсь, но, видимо, надо еще чуть-чуть больше. Я иногда буду помогать.

00:36:57 Окей. Давайте просуммируем, что у нас есть. У нас есть наш E-агент, который триггерится по нашему сообщению во встроенном чате. Встроенный чат исключительно для тестирования нужен больше ни для чего. Он может вызывать модель известную.

00:37:11 Модель мы берем из хаба, который предоставляет возможность ко многим моделям обращаться и выбирать модели. Не отдельно вот к OpenAI, отдельно к кладу, отдельно ещё кому-нибудь, а в общем провайдер 1 и тот же, просто мы меняем название модельки. И теперь мы хотим добавить инструмент для этой модели. Нам здесь подойдет вот этот вот, который кодовый. Мы прямо сейчас быстренько на JavaScript накидаем, вернее, он уже накидан, я не буду на это тратить время.

00:37:41 Я только скажу, что этих инструментов здесь ого-го сколько, то есть взаимодействие практически с любым сервисом, и мы на самом деле увидим вот как бы в основном примере, как это с реляционными базами данных работает. Ну, в общем, как бы из того, что тут есть, можно собрать ну прям очень-очень много всего. Очень много всего. То есть, если ваша модель решила отправить e-mail, ну, вон там есть Gmail tool, допустим, где он висит и как GVD это должен быть. Вот он, да.

00:38:09 Если она решила на GitHub сделать, вот тут есть GitHub tool. Ключевой вопрос в том, что она должна решить использовать этот инструмент. Она это делает сама. И это как бы тоже такой тонкий вопрос решит она или не решит, потому что точность использования инструментов она не 100%. Иногда модель может не понять.

00:38:29 Ну как-то так промпт написан, она может не вычленить из него, что Ага, мне нужен вот этот инструмент. Так бывает. Ну, как бы, здесь всё не стопроцентной точностью обладает, поэтому с этим просто надо смириться, принять какие-то меры, обработку ошибок и так далее. Окей, добавляем этого товарища. Здесь давайте вот потихонечку, спокойно.

00:38:54 Вот он наш tool. Это не просто кодовая нода, вот как есть в Initem, где вы какой-то вход получаете и какой-то выход даёте, а посередине у вас JavaScript или Python. Эта штука чуть сложнее. Во-первых, ей нужно описание. Почему?

00:39:10 Потому что модель на основании описания будет понимать, а что эта штука делает. И на основании схемы входов здесь вот есть такая штука Specify input scheme, что по-русски означает задать схему, собственно, входных данных. Мы её зададим. Если мы вот это всё не заполним, то как там оно выберет это такой вопрос. Я, конечно же, немножечко схитрю, потому что у меня уже всё это написано, я не хочу на это время тратить, я просто возьму вот так вот скопирую отсюда описание.

00:39:48 Оно отформатировано, вызывая этот tool для получения случайного числа на ход, подавая объект с полями Low и High. Всё как бы предельно просто. Язык оставляем JavaScript и сам JavaScript опять же тоже написан, но он прям супер примитивный, ясное дело, тут ничего сложного не нужно. Если вы не знаете JavaScript, скорее всего большинство не знает, просто забейте, оно даёт случайное число целое в интервале от Low до High включительно. Всё, больше ничего знать не надо.

00:40:20 Преобразование в строку на самом деле не обязательно, поэтому его можно и убрать. Теперь давайте вход зададим. Здесь есть 2 варианта. Можно задать прямо вот честно, а можно задать пример, и оно там дальше само разберётся. Давайте мы зададим пример, чтобы оно не так.

00:40:44 Допустим, здесь будет 10 и 100. Вот этого в принципе достаточно уже. То есть оно скажет: Ага, я поняла, у тебя значит на входе должен быть объект с 2 полями и оба поля числа. Окей, я поняла, как с этим работать. В принципе инструмент готов.

00:41:10 Из чего он состоит? Из описания, из описания входа и, собственно, из его функциональности. В данном случае это какой-то код. Вот теперь давайте смотреть, получится ли у нас эту всю штуку запустить. В самом акинте нам нужно понять, должны ли мы что-то менять.

00:41:32 Я здесь добавлю системное сообщение. Давайте что-нибудь разумное напишем. Умею генерировать случайные целые числа по запросу пользователя? Думая, что у него можно охватить. Да, тут, наверное, важно сказать, что то, что вот ты сейчас пишешь, это по сути инструкция для модели.

00:42:01 То есть это некая инструкция по тому, кем ей быть и что уметь делать. Отступление как бы на шаг или даже на 2 шага назад, системный промпт это общие некие инструкции о том, кто она такая, в какой роли она выступает, каким правилам ей следовать и так далее. Инструкции не обязательно привязаны к этому конкретному промту исключительно, а как бы вообще. А дальше вы можете с ней уже общаться несколькими сообщениями. И вот этот промпт, пока он в контексте, по крайней мере, он будет её корректировать в её поведении.

00:42:41 Наверное, мы не будем о промптинге говорить, потому что это такая большая тема. Можно У нас на эту тему, пользуясь случаем, много вебинаров в вебинарах. А, ну тем более. Заходите, смотрите, там все про это подробно работает. Отлично.

00:42:51 Вот теперь давайте смотреть. Так, оно сохраняется само. Давайте смотреть, получится ли это вот в таком простом подходе, по идее должно. Сейчас внимательно-внимательно вот за этими штуками смотрим, потому что когда он запускается, там видно, кто работает. Вот мы отправили сообщение, работает я.

00:43:12 Вернее, кружочки. Включился Code Tool, вернулся снова к EA Агенту. Результат. Ваше случайное число 13. Вот теперь давайте разберем, что произошло.

00:43:23 Я сейчас отодвину вот эту нижнюю панельку. И здесь мы видим последовательность событий в EA Агенте. Сначала сама модель. Она получила свой системный промпт, system умею генерировать случайные числа по запросу. Human это наш промпт, что мы у неё попросили.

00:43:41 Дай мне случайное число от 10 до 20. Она там внутри, зная, какие у неё есть инструменты, решила: Ага, мне нужно вот этот инструмент использовать. Окей. Вызвала его. Посмотрите, правильно распарсила, нашла, что 10.2 действительно.

00:43:58 Всё хорошо. Мы могли там, кстати, схему не совсем передавать, но как бы мы это увидим дальше. Вернулась сюда. Теперь смотрим, как теперь input выглядит. Системный промпт, наш запрос и дальше вот такая штучка tool.

00:44:12 Это то, что наш код вернул. И вот после этого, вот из этого всего она наконец-то сгенерировала наш ответ. Ключевое здесь, что вот эта вот последовательность событий, которая происходит, Модель смотрит на исходный промпт, ну, естественно, на свой системный и на тот, который от пользователя пришел. Решает, на основании того, что она знает о своих

инструментах, надо ли ей какой-то использовать или какие-то или не надо. Если надо, то она его использует с теми данными, которые она получила от пользователя, берет ответ, добавляет его к промту и дальше выдает финальный результат.

00:44:50 Вот в этом фишка. Что здесь в инструменте находится? Это дело совершенно десятое. В принципе, там может быть какая-то довольно сложная штука, которая в том числе завязана на физический мир, например. То есть, в принципе, многие физические объекты имеют некие API и так далее.

00:45:08 И, в принципе, модель может ими управлять в целом. Насколько это безопасно или небезопасно это нас сейчас не интересует, но, в принципе, может. Вот эта вот последовательность это главное, что я в этом игрушечном совсем примере хотел объяснить, как в принципе это всё работает, и что это за тулинг. Потому что сейчас-то вот как бы что-то начинаешь говорить, а видно, что не укладывается схема в голове, непонятно, откуда эти инструменты берутся, в какой момент они вызываются, что вообще происходит, какая-то магия вроде бы. Я надеюсь, что вот так оно больше не выглядит как магия.

00:45:39 Вот, по крайней мере, вот глядя на вот этот input, системный промпт, наш промпт. Затем смотрите, AI пустой. То есть он решил ничего не отвечать, решил вызвать инструмент, получил ответ от инструмента и затем только ответил нам. Да, это очень важный момент. По сути, ответ на вопрос, на дискуссию, которая сейчас в чате развернулась, значит, ее смысл заключается в том, что поскольку AI склонен, сами модели склонны к тому, чтобы галлюцинировать и иногда выдавать какую-то несуществующую информацию, то вопрос возникает, как сделать так, чтобы агент как бы работал с достоверной информацией и не придумывал.

00:46:20 Ведь вот в твоем кейсе он сейчас мог придумать число. Придумать, да. Понимал, что к нему подключен конкретный tool и вызвал его для решения этой задачи, потом сообщил уже конкретное число, которое было именно случайно сгенерировано. Я сейчас давай озвучу как раз дискуссию, которая там есть. Следующем.

00:46:43 Началась она с того, что модель может придумывать ответы на запросе к реальной базе данных. Почему это происходит? У нее уже задан конкретный источник, и если там данных нет, то вам нужно просто сообщить об отсутствии информации, а не придумывать ответ. Вот. Модель решит, модели неуправляемы.

00:46:58 Как она может сама решить, если ей заранее не оговорили, в каких случаях ей написать письмо. Я как раз это прокомментировал, что в силу специфики модели иногда могут эволюционировать, но если мы даем ей как бы базу данных и довольно широкий спектр диапазон, в котором она может отвечать, т. Е, например, она должна с пользователем вести диалог более естественный, а не просто служить передатчиком информации из базы к пользователю, то тогда возникает риск того, что там, ну, допустим, мы у модели просим конкретную книжку, книжка есть в базе данных, он ее сообщает, а, допустим, а следующим этапом пользователь спрашивает: а какого года выпуска эта книга? И этой информации, например, в базе нет. Но модель, поскольку она должна что-то пользователю ответить и вести с ним диалог как представитель, там не знаю, книжный бот, то она может придумать в теории эту информацию.

00:47:48 И вот задача контролирующего бота агента, она в том, чтобы он уже довольно жестко просто сверял этот ответ соответствует базе данных или нет. Ну, как бы, и не пропускал его, если это необходимо. Вот, ну и дальше тут, то есть, задается вопрос нужно ли проверять по кругу 1 агента за другим, так как они все могут нагаллюцинировать. Ну, значит, много дискуссий, сейчас все-все не буду прописывать. Я могу коротко на этот счет прокомментировать.

00:48:19 Смотрите, вот то, что мы сейчас рисуем, это же просто. То есть у этой ноды может быть ошибочный вывод. Мы можем его добавить, если она глюкнула, и что такое глюкнула, мы в том числе можем задать. Мы идём в какую-то другую ветку и эту ошибку обрабатываем. То есть в принципе понятно, что то, что мы сейчас рисуем это не то, что можно отправлять в production.

00:48:41 Нужно добавлять обработку ошибок, перезапуск, если нужно и много-много чего ещё, кучу разных проверок и так далее. Поэтому понятно, что вот мы это немножечко убираем в сторонку. Для всего этого есть решения, которые всё это улучшают, начиная от промтинга, типа не делай вот этого. То есть если ты вот этого не нашла, вот так скажи, что ты не нашла. Не выдумывай.

00:49:06 Это прямо в промте можно написать и в общем это довольно неплохо работает обычно. И заканчивая какими-то вот такими жесткими проверками Во-первых. Во-вторых, мы будем сейчас делать уже как раз с запросами к БД, мы увидим, что мы запросы не генерируем в самой модели. Запросы у нас шаблонизированы. Мы генерируем, что в эти шаблоны подставить.

00:49:31 И это сразу повышает, потому что понятно, модель там такого может в запросе к базе данных нагенерировать, чтобы потом базу данных не найдешь, где она там вообще была. Поэтому все эти вопросы валидные, но

они решаются не здесь. Они решаются на архитектурном уровне. Когда планируется эта система, когда прописывается архитектура, обработка ошибок, логирование, перезапуск каких-то частей или всего пайплайна может быть, если он не супер большой и так далее. Всё это верно, и галлюцинации это верно, всё это нужно учитывать.

00:50:02 Вызовет или не вызовет инструмент нужно учитывать. В следующем мы прямо напишем, когда вызывать, когда не вызывать. В следующем примере основном. Поэтому всё это так, Но на самом деле люди не идеальны. Всё, что нам надо, чтобы вот эта система была хотя бы настолько же не идеальной, как человек в среднем.

00:50:23 Вот и всё. Потому что она в дальнейшем будет стоить дешевле и работать скорее всего быстрее. Не ходить на больничный и так далее. В отпуск и прочее, прочее, прочее. Все про цену.

00:50:34 Вот еще вопрос задают. В ChatGPT есть режим агентов. Мы тоже говорим про агентов. Можем, пожалуйста, объяснить разницу, значит, коротко. Ну, я со своей стороны могу сказать, что тот агент, который в чат gpt, он все-таки не обладает коннекторами к конкретным базам данных или конкретным toolam, то есть приложением внешним, другим источником.

00:50:57 Единственное в этом смысле, что он может сделать, если мы подключили, например, ему почту свою, дали почтовый сервис или календарь или отправили его в интернет поискать нам билеты, например, на самолет, то вот это он может сделать. То есть, для него открыт интернет и он выполняет задачу, идет каскадно, пошагово пытается ее выполнить, добиться нужного результата. Но для бизнеса, когда мы говорим про интеграцию с какими-то внутренними системами или под очень узкоспециализированные задачи, это не самый рабочий и подходящий вариант. Поэтому это не работает. Много еще вопросов я сейчас еще прокомментирую.

00:51:36 Вместо того, чтобы запретить выдумывать модели и говорить, что нет данных или книги нет, мы дадим ненужные ветви диалога про несуществующие книги, проверки другими ботами. Вот не совсем так. То есть, это просто сделано для того, чтобы бот вел более естественный диалог с пользователем, обращаясь к базе данных. То есть, он ходит в базу данных, оттуда вытаскивает книгу, но у него могут спросить: расскажи мне, а почему мне стоит эту книгу почитать сегодня вечером? Или, например, а как эту книгу можно было бы рассказать на книжном клубе?

00:52:07 Если мы слишком жестко его запромтим и этой информации естественно нет в базе данных, то тогда ответ бота будет: извините, я не знаю эту

информацию, извините, я не умею, извините, извините, он как бы все время это будет говорить. Поэтому мы оставляем некоторую гибкость этому агенту, чтобы он все-таки мог довольно широко общаться по тем книгам, которые есть в базе, но при этом контролируем его внешним агентом, который проверяет, например, ну не выдал ли он информацию о конкретной книге, которая в реальности не существует. То есть, это не то, чтобы плодить лишние ветки диалога, это скорее оставлять гибкость, при этом проверяя какие-то контрольные точки, которые являются критичными, там да, например, не знаю, имя автора, чтобы он не перепутал, да. Вот, что тут еще, наверное, прокомментировать? Выживает такая дискуссия в чате.

00:52:57 Пам-пам-пам-пам-пам. Сразу вместо внутри основного агента нельзя настроить его шаг работы, создать шаг сверки с базой. Да, можно, конечно. Мы вот сейчас покажем это тоже как раз, как это делает. И был задан вопрос про несуществующий билет.

00:53:12 Он вам не купит случайно? Нет, не существующий билет не купит, потому что его нет. В этом смысле он просто будет нажимать на кнопки, если мы говорим про агента ChatGPT, который внутри самого ChatGPT. GPT, он просто выполнит, ну, как бы, он видит экран, если можно применить слово видит в этом контексте, да, и там понимает, что надо нажать на эту кнопку, на эту или на эту. Вот, как-то так.

00:53:33 Я постарался прокомментировать, чтобы закрыть часть вопросов. Давайте, наверное, дальше двинемся. Да, давайте. Маленький совсем комментарий. Мне кажется, вот этот разговор про галлюцинации, он немножечко по инерции идет, потому что ситуация сильно улучшилась с последними моделями, прямо скажем.

00:53:54 Сильно улучшилась с каким-то более изощренным промтингом. Я не могу сказать, что это прямо катастрофически, как это бывало раньше. С этим можно бороться и в большинстве случаев успешно. То, что какой-то процент ошибок остаётся, оно остаётся, а он у людей остаётся. И у кода без языковых моделей тоже остаётся.

00:54:16 Поэтому я бы не сказал, что это ключевая проблема сейчас. Вот так давайте сформулируем. Почему-то об этом до сих пор много говорят, несмотря на то, что всё стало лучше по разным причинам. Из нескольких источников эти улучшения идут, но они есть. Окей, едем дальше.

00:54:36 Давайте теперь быстренько, вот мы сейчас прям, мы поняли, как это сделать, поэтому мы, я думаю, основной пайплайн сделаем очень быстро. Так, этот я сохраню, вернусь и Кстати, пока ты сохраняешь, как раз тоже

прокомментируешь, что в целом, когда вы создали какой-то процесс, вы можете ее сохранить в виде так называемой, это называется нодами, да, по сути это некий json файл, кусочек кода. То есть, вы как бы этого агента можете сохранить как кусочек кода и кому-то отправить, передать, расшарить и так далее. И многие, как я говорил, повторюсь, агенты, которых уже создавали профессионалы по всему миру, они доступны для скачивания на специальном маркете NE10, и вы можете посмотреть, как они собраны, как они работают. Есть очень изощренные конструкции, которые, вот, не знаю, чтобы собрать, нужно месяц сидеть и ковырять.

00:55:32 Скорее всего, собрать-то быстро, а вот потом вычистить-вылезать, может быть, довольно долго. Окей, давайте возьмемся за основную задачу. Мы там что-то про DataDreven говорили, давайте чуть-чуть попробуем DataDreven сделать. Смотрите, вот прям опять же это всё ещё игрушечный пример, но он будет иллюстрировать уже что-то более или менее реалистичное. То есть мы базу данных сделаем и будем к ней обращаться на естественном языке и получать ответы на естественном языке.

00:55:59 Представьте себе, что вы языка запросов в SQL не знаете. Вот не знаете. BI ваш он там загружен, вся компания его терроризирует, поэтому он не успевает на запрос отвечать. Мы как бы редуцируем этот дань на самом деле до нескольких запросов всего. Но конкретно.

00:56:18 И научим, в промте, вернее, напишем, как их различать и так далее. Но всё равно это какая-то попытка сделать более-менее реалистичной. Стартуем мы вот с этого файлика. Здесь как бы сделки отдельные. Для каждой сделки указана сумма, указан менеджер по продажам.

00:56:38 Регион здесь только Центральный Федеральный округ. Просто для простоты, чтобы не было слишком много и сложно понять, насколько оно разумно выдает, неразумно. Потому что когда все регионы, иногда просто так сразу глазами непонятно, оно вообще реалистично или не очень. То есть здесь у нас регионов меньше, сколько 18, да, по-моему, в Центральном федеральном округе и как-то проще на всё это смотреть. Ну и дата собственно сделки.

00:57:02 Все сделки здесь за август. Данные синтетические, имена синтетические суммы, синтетические даты синтетические, но они реалистичные в том смысле, что там сделаны поправки на валовый региональный продукт, на количественность населения и всякие такие штуки. То есть сами данные выглядят очень реалистично, но они не настоящие. То есть они синтезированные. Я прям посидел ручками аккуратно все это попытался смоделировать.

00:57:30 Вот это наш файлик. Мы его хотим куда-нибудь запихнуть, потому что вот сюда мы точно напрямую обращаться не сможем. Можно, на самом деле, но это будет совсем некрасиво. Какие варианты? Берем Docker, берем Postgress, разворачиваем все как бы жестко и на этом моменте все, кто не разработчики, с вебинара уходят и все заканчивается.

00:57:51 Поэтому так мы делать не будем, мы сделаем чуть хитрее. Опять же мы 1 сервис обсудили openrouter. Сейчас
