

# Интеграция VseGPT в SaleBot

1

Чтобы применить эту инструкцию, желательно иметь базовый опыт работы с SaleBot. Тогда всё получится с первого раза и с минимальными затратами времени.

Но умение читать и присутствие логики - уже будет достаточно :)

Пример из этой инструкции - телеграм-бот, в котором есть функции в двух форматах:

- Для получения ответа от нейросети в конкретном формате, который вы задаёте
- Для получения ответов от нейросети на любые вопросы, без заданного формата

2

В конструкторе воронок создайте блок "Первостепенная проверка условия"

Через него подписчик вашего бота будет переходить к ИИ-функциям

The screenshot displays the SaleBot funnel builder interface. At the top, there are settings for 'Кейсы' (Cases), 'RU' (language), '(GMT+03:00)' (time zone), 'На смене' (On shift), and 'Инфобиз до 03:18 15.02.2024' (InfoBiz until 03:18 15.02.2024). The main workspace shows a flowchart with two green blocks: 'Что хотите сделать с' (What do you want to do with) and 'Чтобы вернуть в меню' (To return to the menu). Below the first block is a '#(none)' block, and below the second is a 'Написать ещё одну статью' (Write another article) block. On the right, a list of conditions is shown, with 'Первостепенная проверка условия' (Priority condition check) highlighted in blue and circled in orange. Other conditions include 'Начало диалога' (Start of dialog), 'Состояние диалога' (Dialog state), 'Передача данных' (Data transfer), 'Не состояние передача данных' (Not data transfer state), 'Конец сбора данных' (End of data collection), 'Не состояние конец сбора данных' (Not end of data collection state), 'Напоминание' (Reminder), 'Не состояние с условием' (Not state with condition), and 'Не состояние' (Not state).

3

В поле "Условие" напишите текст и/или команды\*, которые должен ввести пользователь, чтобы открыть меню ИИ-функций вашего бота.

В данном примере сообщение с меню ИИ-функций будет приходить при старте диалога с ботом и при вводе команды /menu

Чтобы у подписчиков вашего бота всегда был доступ к ИИ-функциям, можете назначить их вызов на команды и добавить эти команды в меню бота (кнопка в левом нижнем углу чата с ботом). Создать команды можно в BotFather: Edit Bot -> Edit Commands

The screenshot displays the BotFather interface for configuring a bot command. At the top, there are settings for the bot, including the language (RU), time zone (GMT+03:00), and a status indicator (На смене). The main area is divided into two parts: a flowchart on the left and a configuration panel on the right.

The flowchart shows a sequence of actions:

- A green box labeled "Что хотите сделать с" (What do you want to do with) is connected to another green box labeled "Чтобы вернуть в меню" (To return to the menu).
- An arrow points from the "Чтобы вернуть в меню" box to a white box labeled "#[none]".
- Below the "#[none]" box, there is a text input field containing "Написать ещё одну статью" (Write another article).

The configuration panel on the right is titled "Первостепенная провер..." (Priority check...) and includes the following fields:

- Условие** (Condition): A text input field containing "/start;/menu".
- Выбор соответствия** (Match selection): A dropdown menu set to "Полное совпадение" (Exact match).
- Переменная для сравнения** (Comparison variable): A text input field containing "Переменная для сравнения".
- Расширенные настройки** (Advanced settings): A dropdown menu.

Below the configuration panel, there is a text input field containing "Что хотите сделать с помощью нейросети?" (What do you want to do with the help of the neural network?).

4

Введите текст стартового сообщения, который получит подписчик вашего бота

The screenshot shows a bot configuration interface. On the left, a flowchart on a grid background consists of two green nodes: 'Что хотите сделать с' and 'Чтобы вернуть в меню'. A transition labeled 'Вернуться в меню' connects the first node to the second. Below the second node is a white box containing '#[none]'. A transition labeled 'Написать еще одну статью' connects this box to the second node. A dashed transition labeled 'Через 1 секунду' connects the second node to a white box containing '#[error] #[text]'. A transition labeled 'Вернуться в меню' connects this box back to the first node. On the right, a message editor is open with the text '/start;/menu'. Below the text are settings for 'Выбор соответствия' (set to 'Полное совпадение') and 'Переменная для сравнения' (set to 'Переменная для сравнения'). The 'Расширенные настройки' section is expanded, showing a text input field with the text 'Что хотите сделать с помощью нейросети?'. Below the text field are buttons for 'Написать статью' and 'Написать текст медитации'. The character count '39/4000' is visible at the bottom right of the editor.

5

В этом примере у пользователя будет выбор, какими ИИ-функциями воспользоваться. Реализуем их через создание кнопок (здесь - inline-клавиатура, привязанная к сообщению)

The screenshot shows the same bot configuration interface as in step 4, but with the 'Расширенные настройки' section expanded to show keyboard configuration options. The flowchart remains the same. The keyboard configuration panel on the right includes a '+ Добавить кнопку' button, tabs for 'Клавиатура' and 'Кнопки в тексте', and a 'Показывать превью ссылок' toggle. Below these are several buttons for actions: 'Сообщение', 'Вложение', 'Кнопки', 'Действие', 'Условие', 'Калькулятор', 'API-запрос', and 'Аналитика'. The 'Кнопки' button is highlighted with an orange circle. At the bottom of the panel are 'Сохранить' and 'Сохранить и закрыть' buttons.

6

Создайте кнопки по количеству ИИ-функций, которые хотите реализовать, введите текст, который будет отображаться на кнопках.

Здесь кнопки реализованы в виде inline-клавиатуры, т.е. прилеплены к сообщению от бота. Когда подписчик вашего бота нажмёт кнопку, он перейдёт к конкретной ИИ-функции, которую вы настроите дальше.

The screenshot shows a configuration interface for a bot. On the left, a flowchart is displayed on a grid background. It features two green buttons: 'Что хотите сделать с' and 'Чтобы вернуть в меню'. Below the second button is a white box containing '#[none]' and a link icon. A dashed arrow labeled 'Через 1 секунду' points from this box to another white box containing '#[error] #[text]'. A solid arrow labeled 'Написать ещё одну статью' points from the '#[none]' box to the right. A solid arrow labeled 'Вернуться в меню' points from the bottom left back to the first green button. On the right, a keyboard control panel is visible. It has a title 'Кнопки' and a list of buttons: 'Написать статью', 'Написать текст медитации', 'Написать хокку', 'Просто поболтать', and '+ Добавить кнопку'. Below the list are tabs for 'Клавиатура' and 'Кнопки в тексте'. At the bottom of the panel, there are icons for link, gallery, image, document, and music, and a toggle switch labeled 'Показывать превью ссылок' which is currently turned on. A blue bot icon is in the bottom right corner.

7

Когда сделаете всё, что вам нужно, сохраните изменения в этом блоке :)

This screenshot is identical to the one above, showing the same flowchart and keyboard control panel. The main difference is that the blue 'Сохранить' button at the bottom of the keyboard control panel is now highlighted with an orange circle, indicating that the user is ready to save their changes.

8

В свободном месте на поле конструктора, создайте ещё один блок "Первостепенная проверка условия"

Здесь в поле с условиями, нужно ввести текст, который вы написали на одной из кнопок в прошлом блоке. И, если нужно, команду, чтобы эта функция также сразу открывалась из меню.

The screenshot displays a chatbot builder interface. At the top, there are settings for 'Кейсы' (Cases), language 'RU', time zone '(GMT+03:00)', a status 'На смене' (On shift), and a user profile 'МА' with a session timer 'Инфобиз до 03:18 15.02.2024' and a counter '2000 / 65 / 2 / 4'. The main workspace is a grid where a flowchart is being built. It features two green blocks with the text 'Чтобы вернуть в меню' (To return to the menu). The left block is connected to a white block containing '#{none}'. The right block is connected to another white block containing '#{none}'. A search icon is visible in the top right of the workspace. On the right side, a configuration panel for the selected block is open, titled 'Первостепенная провер...' (Primary check...). The panel includes: 'Условие' (Condition) with a text input field containing 'Написать хокку;/haiku'; 'Выбор соответствия' (Match selection) with a dropdown menu set to 'Полное совпадение' (Exact match); 'Переменная для сравнения' (Comparison variable) with a text input field containing 'Переменная для сравнения'; and 'Расширенные настройки' (Advanced settings) with a dropdown arrow. Below the configuration panel, there is a preview of the chatbot's response: 'Чтобы вернуть в меню введите команду /menu' and 'Напишите тему, на которую хотите получить текст хокку:'.

9

Напишите нужный текст сообщения, который бот отправит подписчику при переходе к этой ИИ-функции вашего бота

Чтобы вернуть в меню

#{none}

Через 1 секунду

Написать ещё одно хокку

Чтобы вернуть в меню

#{none}

Через 1 секунду

Полное совпадение

Переменная для сравнения ?

Переменная для сравнения

Расширенные настройки

Чтобы вернуть в меню введите команду /menu

Напишите тему, на которую хотите получить текст хокку

98/4000

Показывать превью ссылок

Сообщение

Вложение

Кнопки

Лействие

10

Протяните стрелку связи от предыдущего блока. Автоматически создастся следующий пустой блок. Никакие настройки этой стрелки-перехода менять не нужно. Эта стрелка означает, что после того как подписчик вашего бота напишет запрос к нейросети, произойдёт переход к обработке запроса.

Чтобы вернуть в меню

Чтобы вернуть в меню

Чтобы вернуть в меню

Написать ещё один текст ме

#{none}

Через 1 секунду

Написать ещё одно хокку

#{none}

Через 1 секунду

11

Теперь перейдите в настройки содержания создавшегося на конце стрелки пустого блока. Оставьте в тексте сообщения `#{none}`. Это переменная, которая означает, что при переходе в этот блок, подписчик не получит никакого сообщения от бота

The screenshot shows a chatbot configuration interface. At the top, there are settings for 'Кейсы' (Cases), 'RU', '(GMT+03:00)', 'На смене' (On shift), 'МА', and 'Инфобиз до 03:18 15.02.2024 2000 / 65 / 2 / 4'. The main area is a flowchart on a grid. A green box contains the text 'Чтобы вернуть в меню' (To return to the menu). Below it, a blue box contains the text `#{none}`. A transition arrow labeled 'Написать ещё одно хокку' (Write another haiku) points to another blue box with `#{none}`. A second green box also contains 'Чтобы вернуть в меню'. To the right, a settings panel is open for a block. The 'Состояние диалога' (Dialog state) is `#25543620`. The 'Тип запроса' (Request type) is 'POST-json'. The 'URL запроса' (Request URL) is `https://api.vsegpt.ru/v1/chat/completions`. The 'Сохраняемые значения' (Saved values) are `error->error; choices[0]|message|content->text;`.

12

В вариантах содержания/функций блока, выберите "API-запрос"

С этого момента начинается самая магия :)

The screenshot shows the same chatbot configuration interface. The flowchart shows a transition from a blue box with `#{none}` to another blue box with `#{none}` via the transition 'Написать ещё одно хокку'. Below this, a transition labeled 'Через 1 секунду' (After 1 second) leads to a blue box with `#{error} #{text}`. The settings panel on the right is open to the 'API-запрос' (API request) option. The 'content' field contains `"content": "#{question}"`. The 'max\_tokens' is 4096 and the 'temperature' is 0.7. The 'Показывать превью ссылок' (Show link preview) toggle is turned on. The 'API-запрос' option is highlighted with an orange circle. At the bottom, there are buttons for 'Сохранить' (Save) and 'Сохранить и закрыть' (Save and close).

### 13 В типе запроса, выберите POST-json

The screenshot shows a chatbot configuration interface. At the top, there are settings for 'Кейсы' (Cases), 'RU' (language), '(GMT+03:00)' (time zone), 'На смене' (On shift), and user information 'МА' (name) and 'Инфобиз до 03:18 15.02.2024 2000 / 65 / 2 / 4' (phone number). The main area is a flowchart with two green nodes labeled 'Чтобы вернуть в меню' (To return to the menu) and a central node 'Написать ещё одно хокку' (Write another haiku). Below the flowchart is a text input field containing '#{none}'. To the right, a configuration panel for the request is shown. The 'Тип запроса' (Request type) is set to 'POST-json'. The 'URL запроса' (Request URL) is 'https://api.vsegpt.ru/v1/chat/completions'. The 'Сохраняемые значения' (Saved values) field contains 'error->error; choices[0]|message|content->text;'. The 'Заголовок запроса' (Request header) field contains '{ "Content-Type": "application/json", "Accept": "application/json",'. An orange circle highlights the 'POST-json' dropdown menu.

### 14 В URL запроса добавьте эндпоинт из документации VseGPT.

Например: <https://api.vsegpt.ru/v1/chat/completions>

This screenshot is similar to the previous one, but the 'URL запроса' (Request URL) field is now being edited. The text 'https://api.vsegpt.ru/v1/chat/completions' is entered, and an orange circle highlights the end of the URL. The rest of the configuration panel, including the request type 'POST-json', saved values, and headers, remains the same as in the previous screenshot.

15

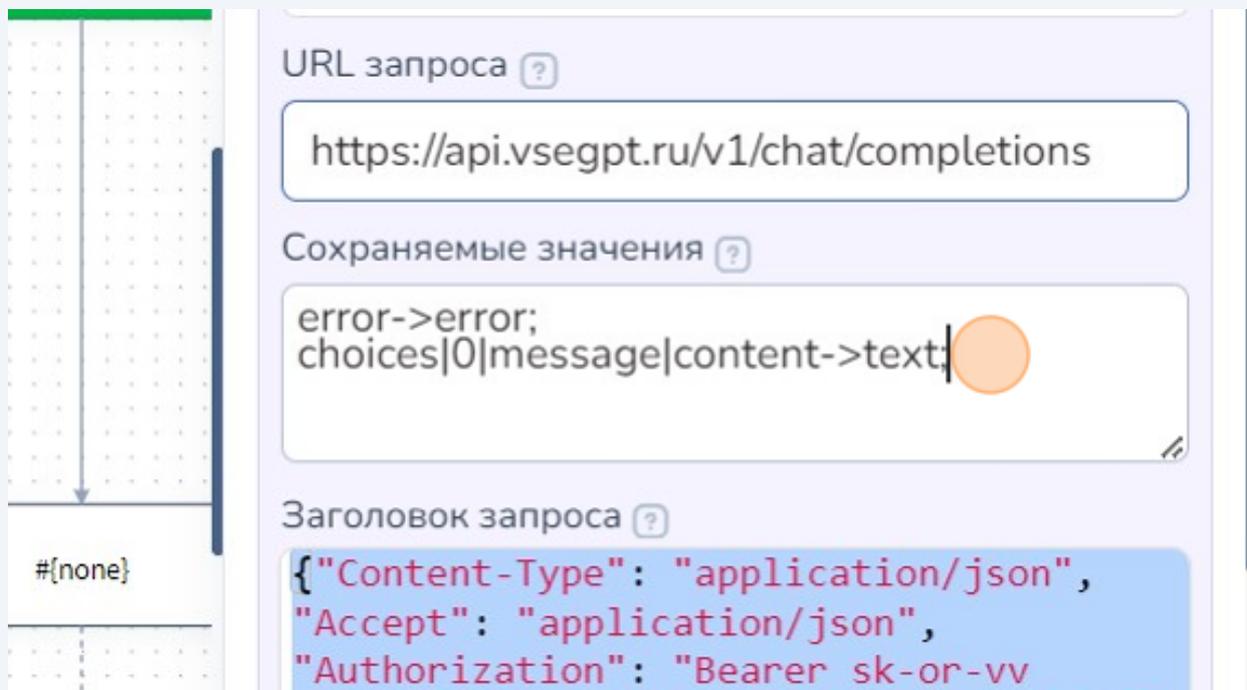
Сохраняемые значения - здесь вы указываете, в какие переменные будут сохраняться конкретные данные из ответа сервера.

Чтобы бот работал, достаточно указать:

**choices|0|message|content->text;**

Это значит, что содержимое ответа "content" будет сохраняться в переменную "text". Название для переменной вы можете выбрать сами.

В примере ниже, также сохраняется значение переменной "error". Если хотите, чтобы подписчики бота получали коды и текст ошибок при обработке запросов к серверу/нейросети, сделайте также.



The screenshot shows a configuration interface for a chatbot. On the left, there is a vertical grid of dots with a blue arrow pointing down to a box containing the text `#{none}`. The main configuration area is divided into three sections:

- URL запроса** (?): A text input field containing the URL `https://api.vsegpt.ru/v1/chat/completions`.
- Сохраняемые значения** (?): A text input field containing the configuration `error->error;` and `choices|0|message|content->text;`. An orange circle is positioned at the end of the second line.
- Заголовок запроса** (?): A text input field containing the JSON headers: `{"Content-Type": "application/json", "Accept": "application/json", "Authorization": "Bearer sk-or-vv"}`.

16

В поле "Заголовок запроса" добавьте:

```
{"Content-Type": "application/json", "Accept": "application/json",  
"Authorization": "Bearer ваш_токен_от_VseGPT", "X-Title": "SaleBot"}
```

X-Title - это опциональный заголовок. В данном примере он добавлен, чтобы в статистике использования VseGPT можно было увидеть, какие запросы делались именно из этого бота.

Workflow diagram showing two parallel paths:

- Path 1: Trigger "Написать ещё одно хокку" (node: `#{none}`) → Delay "Через 1 секунду" → Node `#[error] #[text]` → Action "Чтобы вернуть в меню".
- Path 2: Trigger "Написать ещё одно хокку" (node: `#{none}`) → Delay "Через 1 секунду" → Node `#[error] #[text]` → Action "Чтобы вернуть в меню".

REST Client interface details:

- Сохраняемые значения: `error->error; choices[0]|message|content->text;`
- Заголовок запроса: 

```
{"Content-Type": "application/json",  
"Accept": "application/json",  
"Authorization": "Bearer sk-or-vv  
[redacted]", "X-Title":  
"SaleBot"}
```
- JSON параметры: 

```
{  
  "model": "openai/gpt-3.5-turbo-0125",  
  "messages": [  
    {  
      "role": "system",  
      "content": "Ты - поэт, который  
пишет хокку. Напиши хокку, на тему:"  
    },  
    {  
      "role": "user",  
      "content": ""  
    }  
  ]  
}
```

## 17 Поле JSON параметры:

```
{
  "model": "openai/gpt-3.5-turbo-0125",
  "messages": [
    {
      "role": "system",
      "content": "Ты - поэт, который пишет хокку. Напиши хокку, на тему:"
    },
    {
      "role": "user",
      "content": "#{question}"
    }
  ],
  "max_tokens": 4096,
  "temperature": 0.7
}
```

**model** - модель из документации VseGPT

**role** - в данном примере есть две роли:

1. **system** - в ней вы задаёте системный промпт для нейросети, инструкции, которым она будет сделать при ответе на вопрос/запрос пользователя
2. **user** - здесь вы задаёте сам текст запроса от пользователя. Так как SaleBot по умолчанию сохраняет последнее сообщение подписчиков бота в переменную **question**, то для роли user в контенте нужно указать содержание этой переменной: **#{question}**

**max\_tokens** - здесь общая рекомендация, не превышать максимальный объем текста в одном сообщении, даже если выбранная модель это позволяет. Для телеграма без сбоев работает 4096

**temperature** - как использовать этот и другие параметры, можно посмотреть в документации VseGPT. В большинстве случаев для моделей от OpenAI наилучшие результаты будут получаться при значениях температуры 0.5-0.8

На этом самое главное для работы вашего нейро-бота через SaleBot сделано, не забудьте сохранить изменения в блоке.

```

"model": "openai/gpt-3.5-turbo-0125",
"messages": [
  {
    "role": "system",
    "content": "Ты - поэт, который пишет хокку. Напиши хокку, на тему:"
  },
  {
    "role": "user",
    "content": "#{question}"
  }
],
"max_tokens": 4096,
"temperature": 0.7
}

```

18

От блока с API-запросом проведите стрелку связи, снова создается пустой блок

19

В этот раз настройки стрелки нужно поменять. Для этого наведите курсор на стрелку, нажмите шестерёнку и в поле "Задержка перед ответом" выберите 1 секунда.

В большинстве случаев с такой минимальной задержкой всё работает хорошо.

Если вы заметите, что иногда бот отвечает на прошлый, а не текущий вопрос пользователя, можете увеличить задержку (оптимально - 3-5 сек.)

The screenshot shows a chatbot configuration interface. At the top, there is a header with 'эронек', 'Кейсы', 'RU', '(GMT+03:00)', 'На смене', 'МА', and 'Инфобиз до 03:18 15.02.2024 2000 / 65 / 2 / 4'. The main area contains a flowchart on a grid background. The flowchart consists of three green boxes labeled 'Чтобы вернуть в меню' connected by arrows. Below each green box is a white box labeled '#(none)'. Arrows point from the white boxes back to the green boxes. Labels 'Написать еще один текст ме' and 'Написать еще одно хокку' are positioned between the white boxes. Below the white boxes, there are labels 'Через 1 секунду' with vertical dashed lines. On the right side, a settings panel titled 'Соединение' is open. It has a search icon and a 'Задержка перед ответом' field with the value '1' and a unit dropdown set to 'Секунд'. Below this are three toggle switches: 'Отменить если покинул блок' (off), 'Отменить сообщения с таймером' (on), and 'Не отменять' (off). There are also fields for 'Время отправки' and 'Дата отправки', and a toggle for 'Отсылать если дата прошла' (on). At the bottom of the settings panel, there is a 'Условие' field, a 'Выбор соответствия' dropdown set to 'Игнорируя ошибки и неточности', and a 'Расширенные настройки' dropdown.

20

В блоке, к которому ведёт эта стрелка подписчики бота будут получать ответы от нейросети. В данном примере это содержание переменной `#{text}`

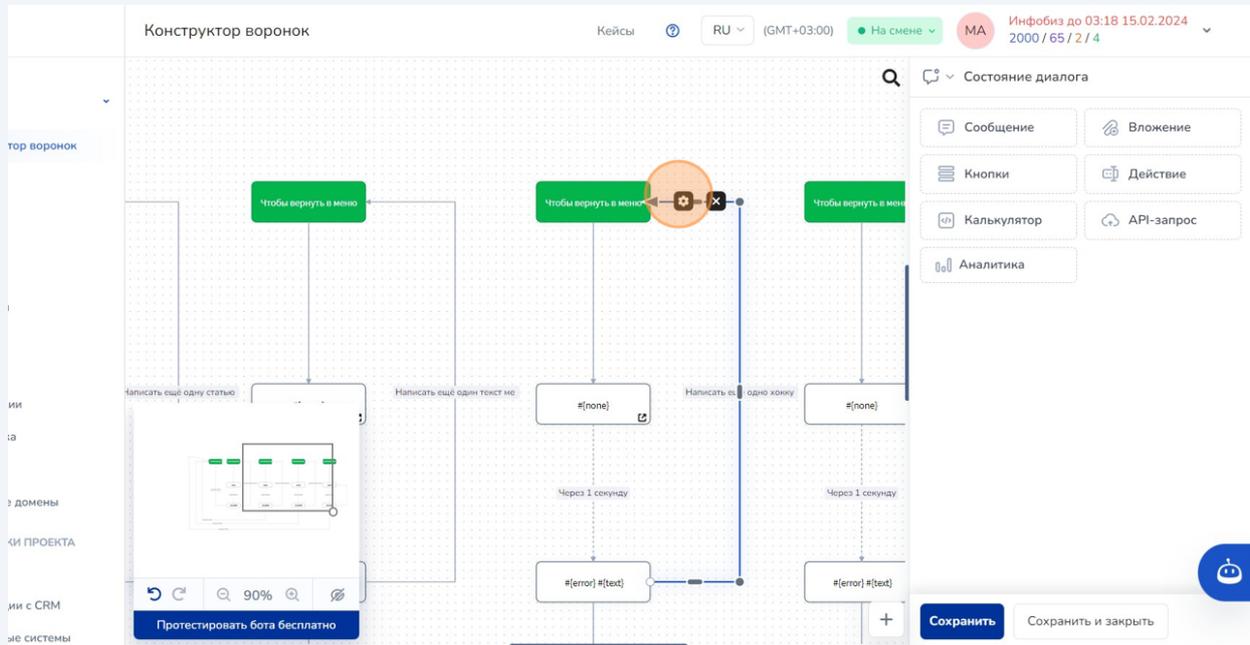
\*на скриншоте показан пример, с показам пользователю также сообщения об ошибках, если они будут возникать. Если вы создавали переменную для сохранения текста ошибок, можете сделать также. Тогда в этом блоке подписчик бота будет получать либо текст с ошибкой, либо ответ от нейросети.

The image shows a screenshot of a Telegram bot interface. On the left, there is a workflow diagram on a grid background. It starts with a trigger event 'Текст мне' (Text to me) leading to a block with the variable `#{none}`. A transition labeled 'Написать ещё одно хокку' (Write another haiku) leads to another `#{none}` block. From there, a transition labeled 'Через 1 секунду' (After 1 second) leads to a block containing `#{error} #{text}`. A second path from the first `#{none}` block also leads to this `#{error} #{text}` block. On the right, there is a chat window for a user named 'Инфобиз до 03:18 15.02.2024'. The chat input field contains the text `#{error} #{text}`. Below the input field are various chat controls, including a toggle for 'Показывать превью ссылок' (Show link preview) and buttons for 'Сообщение' (Message), 'Вложение' (Attachment), 'Кнопки' (Buttons), and 'Действие' (Action).

21

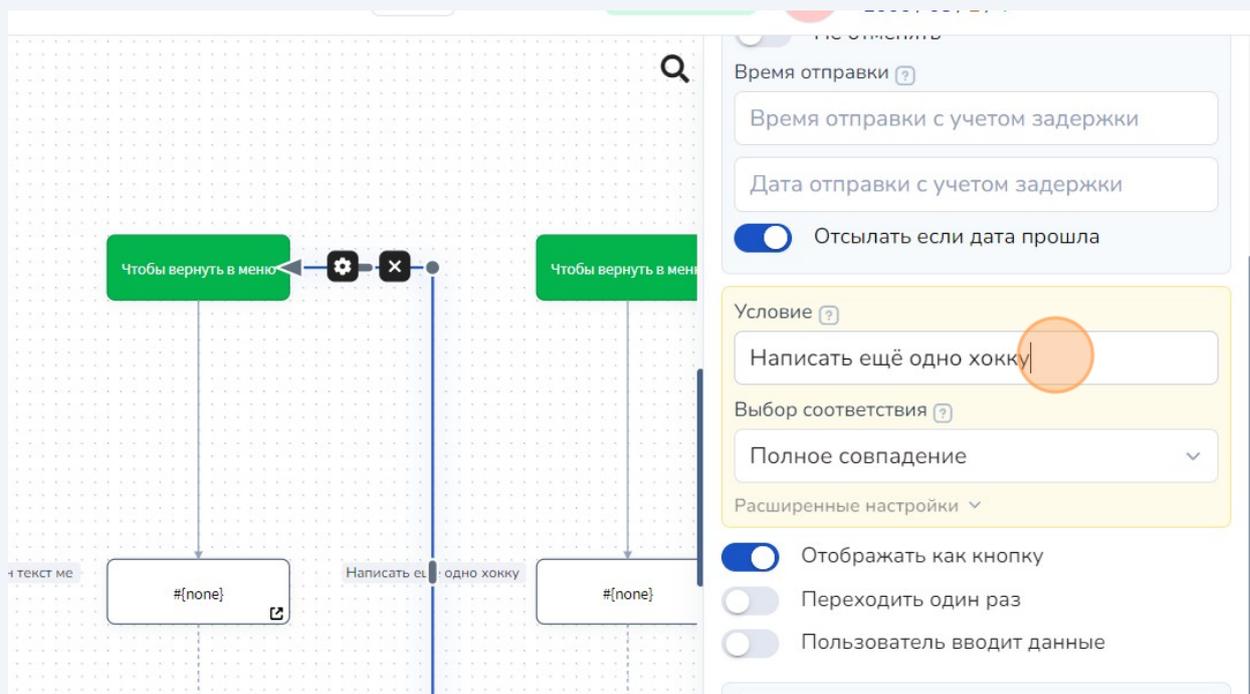
В данном примере, чтобы сделать взаимодействие подписчиков с ИИ-функциями бота удобнее, создадим 2 герлу-кнопки (клавиатура, которая открывается под окном чата в телеграм)

Протяните стрелку связи от блока, где подписчик бота получал ответ от нейросети к блоку "Первостепенная проверка условия", где вы предлагали подписчику сделать запрос к текущей ИИ-функции бота



22

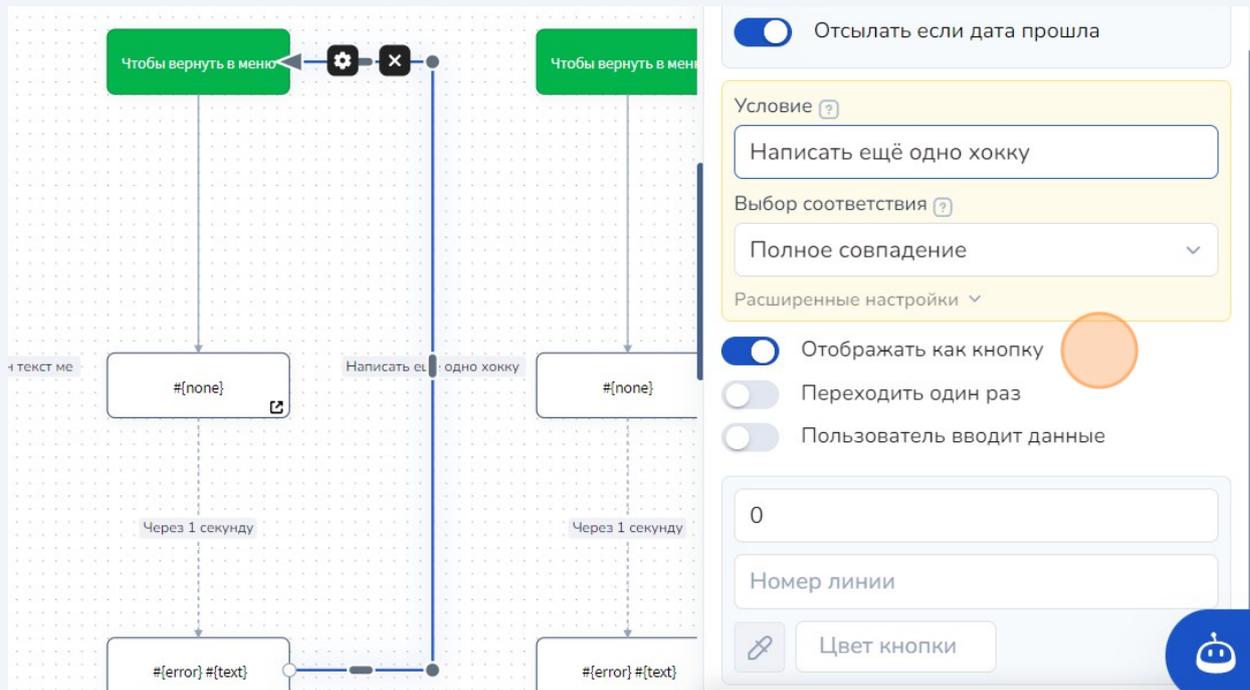
В настройках стрелки добавьте нужный текст в поле условие



## 23 Включите отображение этой стрелки-перехода как кнопки.

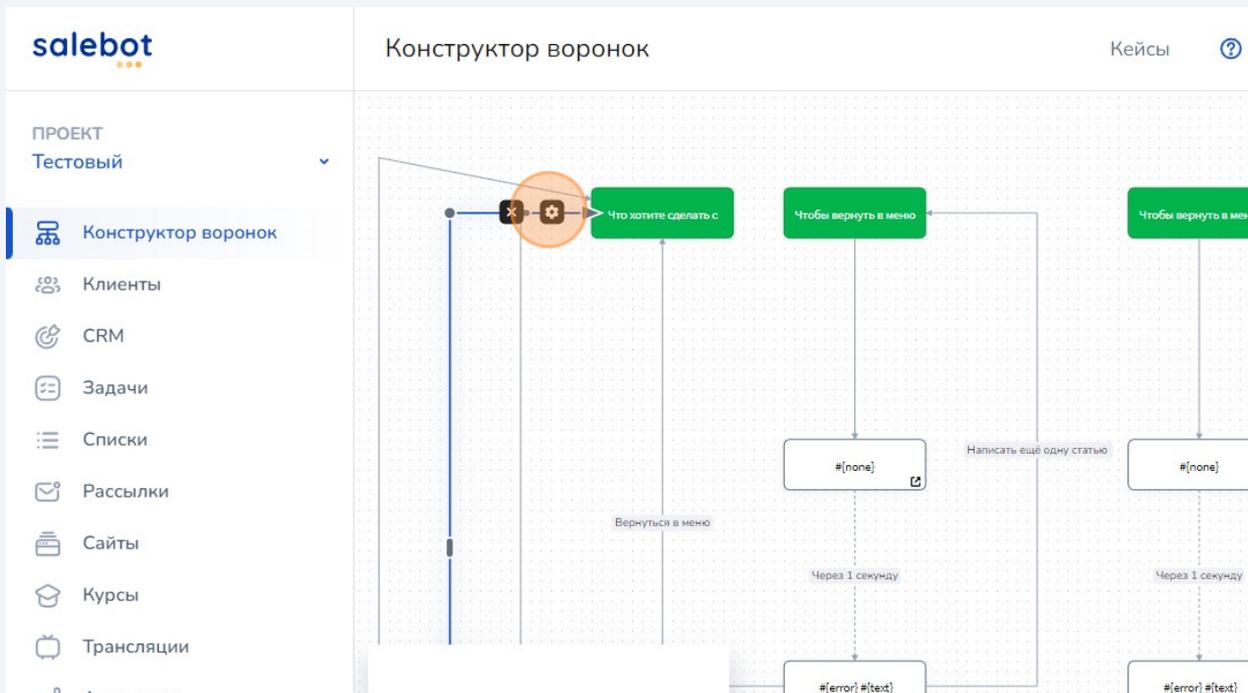
Теперь, вместе с сообщением с ответом от нейросети, у подписчика вашего бота будет появляться кнопка для повторного использования текущей ИИ-функции. В данном примере: "Написать ещё одно хокку".

После нажатия на эту кнопку, бот снова предложит подписчику написать запрос к ИИ.



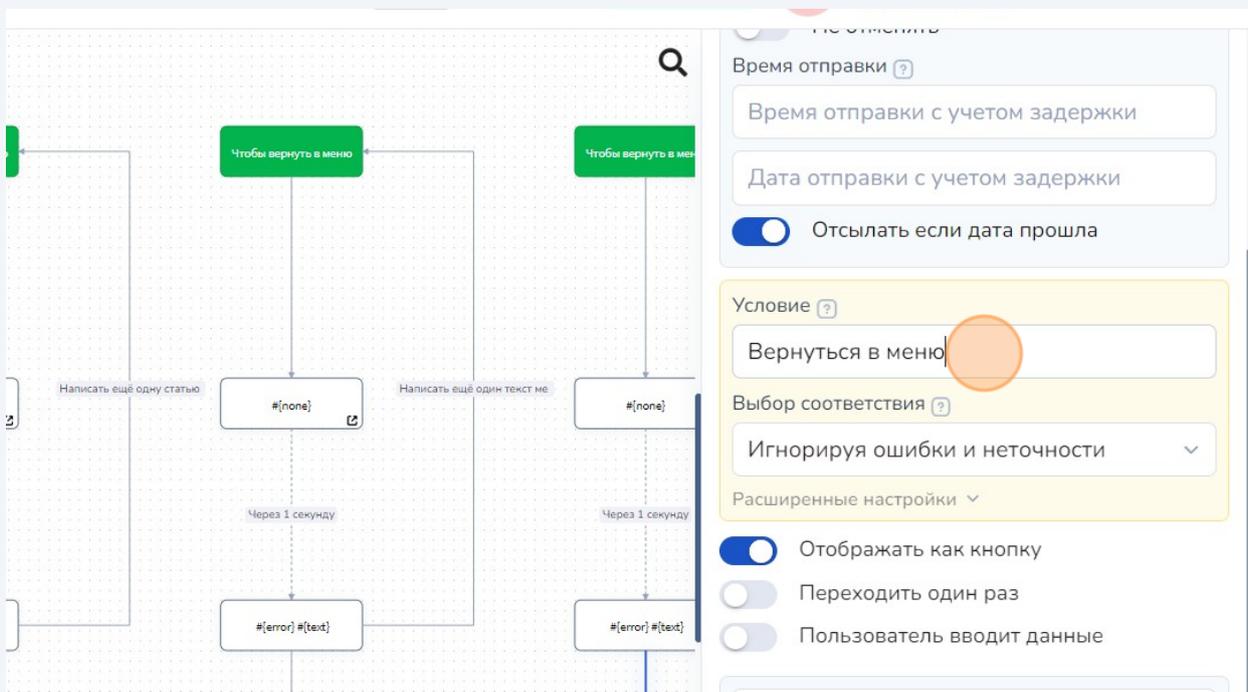
24

Таким же образом сделайте стрелку-связь с самым первым блоком "Первоначальная проверка условия" для создания второй кнопки (она будет возвращать подписчика в стартовое сообщение с набором ИИ-функций)



25

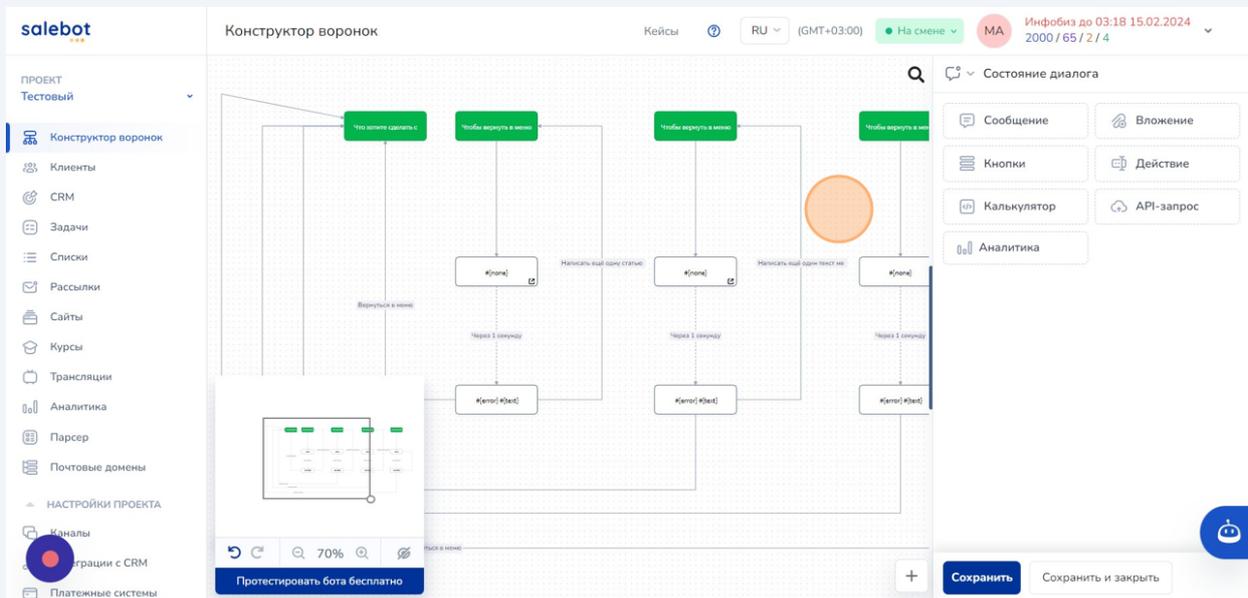
Аналогично первой стрелке, в настройках заполните поле "Условие" и включите отображение этой связи как кнопки.



26

На текущем этапе, ваш бот уже будет работать так (на данном примере):

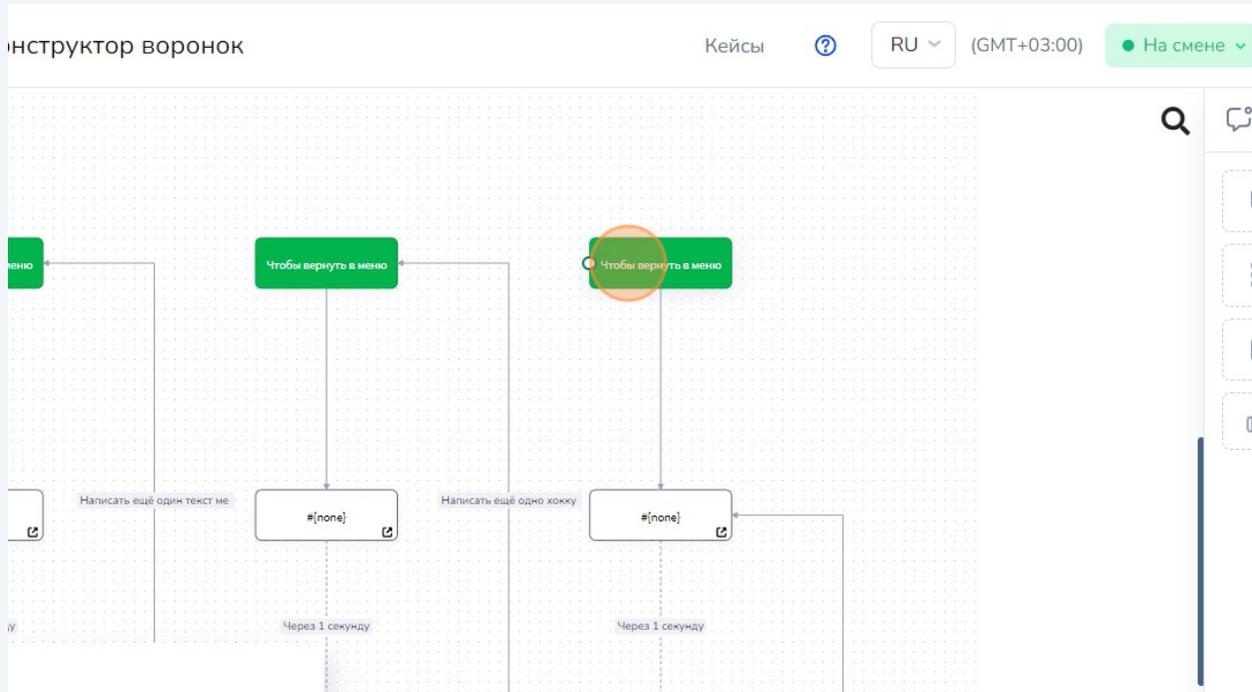
1. Подписчик заходит в бота, запускает диалог с ним (срабатывает команда /start)
2. Подписчику присылается сообщение с вопросом, что он хочет сделать и кнопка "Написать хокку"
3. После нажатия на кнопку, подписчику присылается сообщение с предложением, написать тему для хокку
4. Подписчик пишет тему, бот делает API-запрос к VseGPT с заданными вами параметрами
5. Бот присылает текст ответа подписчику. Либо текст ошибки.



27

Теперь добавим ещё одну ИИ-функцию для бота, чтобы стартовое меню было хотя бы 2 кнопки :)

Это будет функция без строго-заданного формата ответа. Подписчик сможет делать любые запросы к ИИ.



28

В свободном месте поля конструктора воронок создайте ещё один блок "Первоначальная проверка условия"

Далее большая часть действий, которые вы уже делали. Поэтому без подробностей.

"Просто поболтать" - это условие, которое соответствует тексту второй кнопки первоначального меню ИИ-функций бота.

The screenshot displays a chatbot funnel builder interface. At the top, there are navigation elements: "Кейсы", a help icon, "RU", "(GMT+03:00)", "На смене", a user profile "МА", and a session timer "Инфобиз до 03:18 15.02.2024 2000 / 65 / 2 / 4".

The main workspace shows a flowchart on a grid. A green block with the text "Чтобы вернуть в меню" is connected to a white block labeled "#[поле]". A dashed line below the white block indicates a delay of "Через 1 секунду".

On the right, a configuration panel for a condition block is open. The title is "Первостепенная провер... #25798410". The configuration includes:

- Условие**: "Просто поболтать;/dialog" (highlighted with an orange circle).
- Выбор соответствия**: "Полное совпадение".
- Переменная для сравнения**: "Переменная для сравнения".
- Расширенные настройки**: "Чтобы вернуть в меню со всеми функциями, введите команду /menu" and "Напишите свой запрос к нейросети:".

29

В случае, когда вы даёте подписчику бота задавать любые вопросы ИИ и не требуется определенный стиль и формат ответов, роль "system" можно не задавать.

Тогда JSON параметры будут выглядеть так:

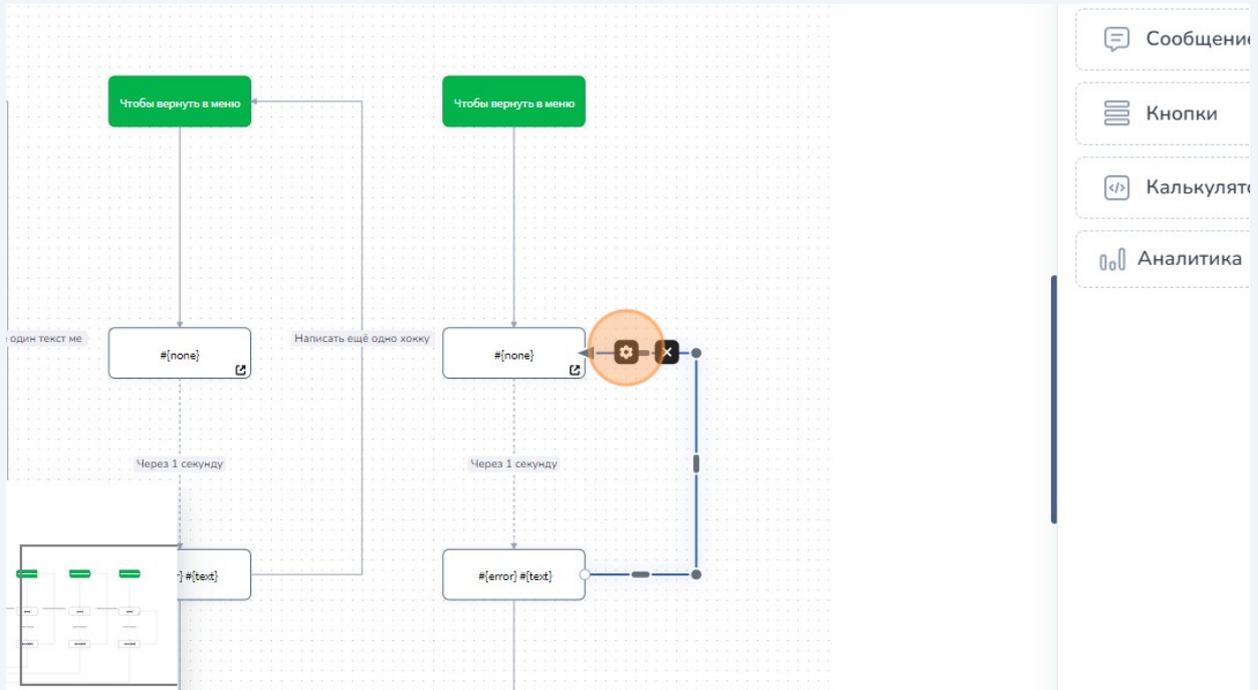
```
{
  "model": "openai/gpt-3.5-turbo-0125",
  "messages": [
    {
      "role": "user",
      "content": "#{question}"
    }
  ],
  "max_tokens": 4096,
  "temperature": 0.7
}
```

The image shows a Telegram bot interface. On the left, a configuration window displays a JSON object for the bot's settings. The JSON is identical to the one shown in the text above. On the right, a chat window titled "Состояние диалога" (Dialog State) shows a system message: "Внимательно следуй инструкциям пользователя и отвечай в дружелюбном тоне" (Carefully follow the user's instructions and answer in a friendly tone). Below this is a user message: "#{question}". The JSON configuration is visible in the chat window's input area, with a red circle highlighting the closing curly brace. The chat window also shows various interaction buttons like "Сообщение", "Вложение", "Кнопки", "Действие", "Калькулятор", and "API-запрос".

30

Единственное отличие в логике реализации функции "Просто поболтать" в том, что после получения ответа от нейросети, подписчик вашего бота может сразу писать следующий вопрос. Ему будет показываться только кнопка "Вернуться в меню" (если вы создадите соответствующую связь, по аналогии с предыдущими инструкциями).

Вторую стрелку-связь от блока с ответом нейросети нужно провести к блоку с API-запросом (никакие изменения в настройки стрелки вносить не нужно)

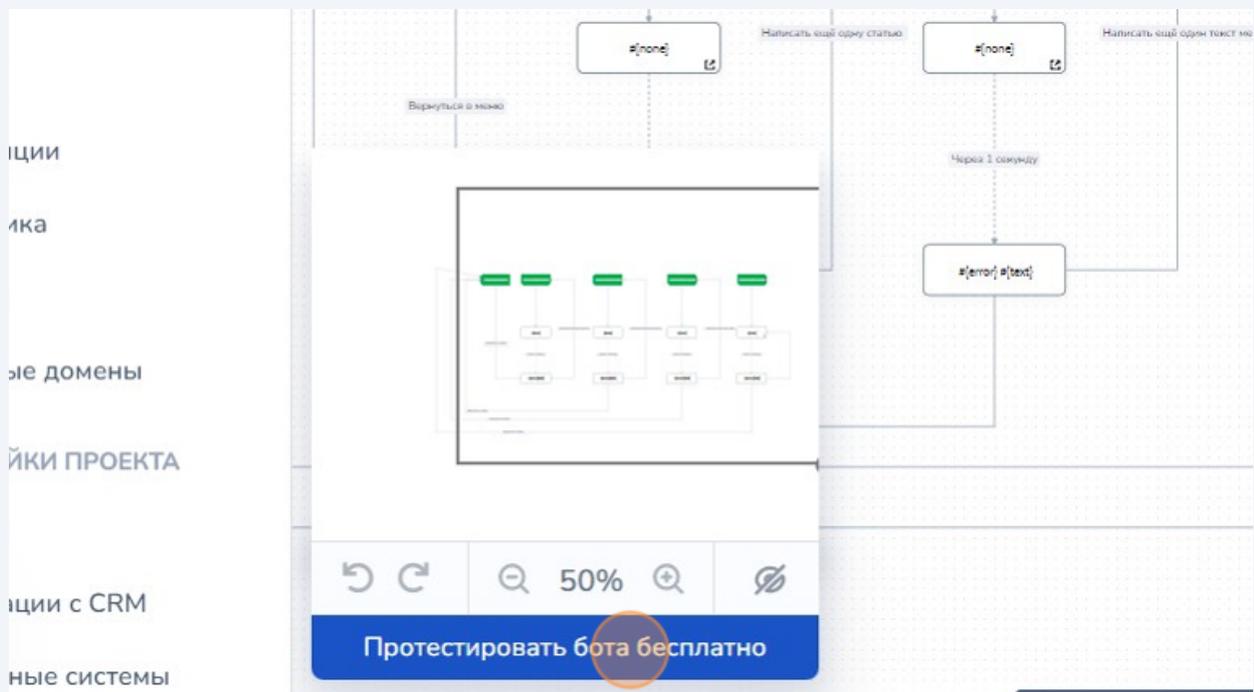


31

Если вы сделали всё по этой инструкции, то теперь ваш бот будет выглядеть и работать так:

1. Подписчик заходит в бота, запускает диалог с ним (срабатывает команда /start)
2. Подписчику присылается сообщение с вопросом, что он хочет сделать и 2 кнопки: "Написать хокку", "Просто поболтать"
3. Подписчик нажимает на кнопку и этим выбирает функцию
4. Подписчику присылается сообщение с предложением задать тему хокку или написать запрос к ИИ
5. Подписчик пишет тему/запрос, бот делает API-запрос к VseGPT с параметрами, заданными в выбранной подписчиком функции
6. Бот присылает текст ответа подписчику. Либо текст ошибки.
7. Под текстом ответа подписчику показываются кнопки:
  - либо "Написать ещё одно хокку" и "Вернуться в меню"
  - либо "Вернуться в меню"

Теперь вы можете протестировать работу бота внутри SaleBot. Для этого нажмите "Протестировать бота бесплатно"



32

Чтобы вернуть в меню введите команду /menu

Напишите тему, на которую хотите получить статью:

**/start**

Что хотите сделать с помощью нейросети?

- Написать статью
- Написать текст медитации
- Написать хокку
- Просто поболтать

Powered by salebot ...

Введите сообщение и нажмите Enter

33

**/start**

Что хотите сделать с помощью нейросети?

- Написать статью
- Написать текст медитации
- Написать хокку
- Просто поболтать

**Написать хокку**

Чтобы вернуть в меню введите команду /menu

Напишите тему, на которую хотите получить текст хокку:

Powered by salebot ...

Введите сообщение и нажмите Enter

34

Для запуска бота введите текст, на который должен реагировать бот. Если это стартовая кнопка, то введите ее название.

Если вы не знаете как создавать бота, то рекомендуем почитать статьи [по ссылке](#)

/start

Что хотите сделать с помощью нейросети?

Написать статью

Написать текст медитации

Написать хокку

Просто поболтать

Написать хокку

Чтобы вернуть в меню введите команду /menu

Напишите тему, на которую хотите получить текст хокку:

Powered by salebot ...

Котики

35

Сам бот работает, нейросеть отвечает

Написать хокку

Чтобы вернуть в меню введите команду /menu

Напишите тему, на которую хотите получить текст хокку:

Котики

Мурлыкают вдвоем,  
на лунном свете играют -  
трепет души моей.

Powered by salebot ...

Написать ещё одно хокку

Вернуться в меню

Введите сообщение и нажмите Enter

36

Теперь можно смело открывать телеграм-бота, которого вы добавили в SaleBot, смотреть, как всё выглядит и работает там. И добавлять ИИ-функции для подписчиков, конечно ☑

