



Настройка Системы для работы Чат-бота и ИИ-ассистента

Памятка Администратора

Версия 5.5 с ИИ

г. Самара, 2026

Оглавление

ВВЕДЕНИЕ	2
1. ТРЕБУЕМЫЕ ДЕЙСТВИЯ ДЛЯ РАБОТЫ ЧАТ-БОТА	3
1.1. Настройка pg_vector	3
1.1.1. Установка pg_vector на Linux	3
1.1.2. Установка pg_vector на ОС Windows.....	5
1.2. Установка Tesseract	5
1.3. Добавление настроек в файлы конфигурации.....	5
1.4. Добавление LLM-модели	10
2. ЧАТ-БОТ.....	11
3. ИИ-АССИСТЕНТ	13

Введение

Установка и настройка системы ТЕЗИС и дополнительного программного обеспечения для работы с искусственным интеллектом описана в [Памятке Администратора Установка ТЕЗИС 5.5 ИИ](#) и предоставляется службой технической поддержки по запросу.



Важно!

Первоначально необходимо произвести установку и настройку по инструкции [Памятки Администратора Установка ТЕЗИС 5.5 ИИ](#) и только потом переходить к дополнительным настройкам, зафиксированным в текущей инструкции.

1. Требуемые действия для работы Чат-бота

`pg_vector` – это расширение PostgreSQL, которое добавляет новый тип данных «vector» и функции для работы с ним (поиск ближайших векторов, сравнение по расстоянию и т.п.). Оно позволяет хранить эмбединги (числовые вектора) в базе данных и искать похожие тексты.

Расширение доступно с 13-ой версии PostgreSQL. В данной инструкции версия PostgreSQL 15 в командах указана в качестве примера.

К существующей БД СЭД на СУБД PostgreSQL можно добавить расширение `pg_vector` или создать отдельную БД под модель и добавить ей расширение вектора.

1.1. Настройка `pg_vector`

1.1.1. Установка `pg_vector` на Linux

Через официальный архив apt PostgreSQL необходимо:

1. Добавить ссылку на репозиторий в отдельный файл: `/etc/apt/sources.list.d/pgdg.list`

```
echo "deb http://apt-archive.postgresql.org/pub/repos/apt buster-  
pgdg main" | sudo tee /etc/apt/sources.list.d/pgdg.list > /dev/null
```

2. Добавить GPG-ключ репозитория.

```
wget --quiet -O -  
https://www.postgresql.org/media/keys/ACCC4CF8.asc | apt-key add -
```

3. Обновить пакеты и установить `pg_vector`.

Выполнить команды:

```
sudo apt update  
sudo apt install postgresql-15-pgvector (можно подставить нужную  
версию, например postgresql-16-pgvector)
```

- 3.1. Установка `pg_vector` из исходников (GitHub).

- 3.1.1. Установить зависимости для PostgreSQL 15.

Выполнить команды:

```
sudo apt update
```

```
sudo apt install -y postgresql-server-dev-15 build-essential
git (подставить нужную версию)
```

3.1.2. Клонировать репозиторий и выбрать нужную версию.

Выполнить команду:

```
cd /tmp
git clone --branch v0.8.1
https://github.com/pgvector/pgvector.git
cd pgvector
```

3.1.3. Выполнить сборку и установку.

Выполнить команды:

```
make
sudo make install
```

4. Подключить расширения к базе.

Выполнить команды:

```
sudo -u postgres psql
\c thesis_ai (или другая бд)

CREATE EXTENSION IF NOT EXISTS vector;
```

5. Выполнить проверку работы.

В той же базе, в psql, выполнить команду:

```
\dx
```

или `SELECT * FROM pg_extension WHERE extname = 'vector';`

В списке должно появиться: «vector | 0.8.1 | public | vector data type for PostgreSQL».

```
thesis_ai=# \dx
```

Name	Version	Schema	Description
plpgsql	1.0	pg_catalog	PL/pgSQL procedural language
vector	0.8.1	public	vector data type and ivfflat and hnsw access methods

(2 rows)

Рисунок 1

1.1.2. Установка pg_vector на ОС Windows

Для установки необходимо:

1. Скачать архив с расширением для подходящей версии PostgreSQL:

https://tezis-doc.ru/materials/pgvector_v0.8.1_pg15.zip

В текущей инструкции в качестве примера рассматривается PostgreSQL 15.

2. Распаковать содержимое архива в папку с установленным PostgreSQL (например, D:\PostgreSQL\15) и перезапустить СУБД.
3. Для подключения расширения выполнить запрос в базе:

```
CREATE EXTENSION vector
```

4. Для проверки работы выполнить:

```
SELECT * FROM pg_extension WHERE extname = 'vector'
```

extname name	extowner oid	extnamespace oid	extrelocatable boolean	extversion text
vector	10	2200	true	0.8.1

Рисунок 2

1.2. Установка Tesseract

Установка Tesseract OCR описана в [Памятке Администратора Установка и настройка Tesseract OCR версия 5.4.](#) и предоставляется службой техподдержки по запросу.

1.3. Добавление настроек в файлы конфигурации

Ниже приведен пример минимальных настроек для работы текущей конфигурации.

Для файла «tomcat\conf\app-core\local.app.properties»:

```
thesis.vectorSearch.enabled=true
thesis.vectorSearch.embeddingsProvider.type=infinity
thesis.vectorSearch.embeddingsProvider.url=https://localhost:7997/embeddings
```

```
thesis.vectorSearch.embeddingsProvider.apiKey=API_KEY  
thesis.vectorSearch.embeddingsProvider.model=bge-m3  
thesis.vectorSearch.vectorStore.dimensions=1024
```

Для файла «tomcat\conf\app\local.web-app.properties»:

```
thesis.vectorSearch.enabled=true
```

Для применения настроек потребуется перезапуск службы «tomcat» на сервере.

Описание доступных конфигураций для настройки LLM модели представлено в таблице [ниже](#) (добавляется «tomcat\conf\app-core\local.app.properties»).

Таблица 1

Параметр	Значение (примеры)	Описание
thesis.chatBot.enabled	true	Включает работу Чат-бота в Системе
thesis.vectorSearch.enabled	true	Активирует модуль векторного поиска (semantic search)
thesis.vectorSearch.embeddingsProvider.type	infinity, yandex, mts	Провайдер эмбедингов (в данном случае – MTS AI)
thesis.vectorSearch.embeddingsProvider.maxBatchSize	32	Максимальное количество текстов, обрабатываемых за один запрос к API
thesis.vectorSearch.embeddingsProvider.maxFragment Length	8000	Максимальная длина текста (в символах или токенах), отправляемого на обработку
thesis.vectorSearch.embeddingsProvider.model	bge-m3 (использовалось для mts)	Название модели, используемой для генерации эмбедингов (векторных представлений текста)
thesis.vectorSearch.embeddingsProvider.url	(url)	URL API для получения эмбедингов

Параметр	Значение (примеры)	Описание
thesis.vectorSearch.vectorStore.dimensions	1024	Размерность (длина) векторного пространства – количество чисел в одном эмбединге
thesis.vectorSearch.embeddingsProvider.apiKey	(apiKey)	Ключ API для доступа к сервису генерации эмбедингов (секретный параметр, не хранить в открытом виде)
thesis.vectorSearch.embeddingsProvider.passagePrefix	(пусто)	Префикс, добавляемый к фрагментам текста перед генерацией эмбедингов (может использоваться для настройки контекста)
thesis.vectorSearch.embeddingsProvider.queryPrefix	(пусто)	Префикс для поисковых запросов при генерации эмбедингов
thesis.vectorSearch.minIndexableTextLength	100	Минимальная длина текста, который можно индексировать (тексты меньшей длины не добавляются в базу)
thesis.vectorSearch.vectorStore.prefix	ts_	Префикс таблиц/индексов в хранилище векторов

Параметр	Значение (примеры)	Описание
thesis.vectorSearch.vectorStore.type	pg_vector	Тип хранилища векторов – в данном случае PostgreSQL с расширением pg_vector
thesis.vectorSearch.splitter.minChunkSize	2000	Минимальный размер фрагмента
thesis.vectorSearch.splitter.maxChunkSize	3000	Максимальный размер фрагмента
thesis.vectorSearch.splitter.minOverlapFraction	0.3	Минимальная доля перекрытия между соседними фрагментами (в диапазоне 0.0-1.0)
thesis.vectorSearch.splitter.maxOverlapFraction	0.5	Максимальная доля перекрытия между соседними фрагментами (в диапазоне 0.0-1.0)

1.4. Добавление LLM-модели

Для корректной работы функций интеллектуального поиска и анализа текста необходимо указать параметры подключения к провайдеру эмбедингов (LLM-модели), который будет использоваться для генерации векторных представлений текста.

Выбор модели, её настройки, как и настройки ИИ-операций, клиент выполняет самостоятельно.

Добавить модель можно через раздел меню «Администрирование» – «Базовые настройки» – «Языковые модели» – «Создать».

Пример добавления в UI представлен на рисунке [ниже](#).

The screenshot shows a web interface for editing a language model. At the top, there are tabs for 'System parameters' and 'Editing language model'. Below the tabs, there's a breadcrumb trail: 'Language models > Editing language model'. Two sub-tabs are visible: 'Basic parameters' (active) and 'Chat bot settings'. The form fields are as follows:

- Название ***: Text input with value 'MWS GPT (bge-m3)'. There is a small 'x' icon to the right.
- Тип модели ***: Dropdown menu with 'Vllm' selected and a downward arrow.
- Адрес API ***: Text input with value 'https://api.gpt.mws.ru/v1/embeddings'.
- Модель ***: Text input with value 'bge-m3'.
- API-ключ**: Text input with a masked value 'sk-...'. There is a small 'x' icon to the right.
- Температура**: Text input with value '0'.
- Системный промпт**: Text area with placeholder 'Введите текст'. There is a 'Развернуть' (Expand) link to the right.
- Описание**: Text area with value 'Модель векторизации. Возвращает векторное представление текста (embedding)'. There is a 'Развернуть' (Expand) link to the right.

Рисунок 3

2. Чат-бот

Открыть Чат-бот в интерфейсе Системы можно по кнопке  (см. рисунок [ниже](#)).

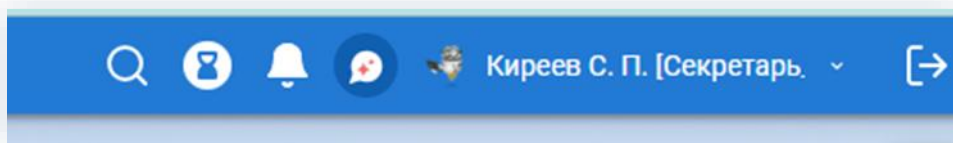


Рисунок 4

Ниже перечислены роли, необходимые для работы Чат-бота:

- «ChatBotAccess» – позволяет взаимодействовать с встроенным в Систему Чат-ботом.

В Чат-боте доступны при этом следующие режимы работы: «Без поиска», «Глобальный поиск», «Поиск по сущностям», а также кнопка .

- «ChatBotGlobalSearch» – позволяет выполнять семантический поиск по всей Системе с помощью встроенного Чат-бота.

Данная роль предоставляет доступ к режиму «Глобальный поиск» в Чат-боте.

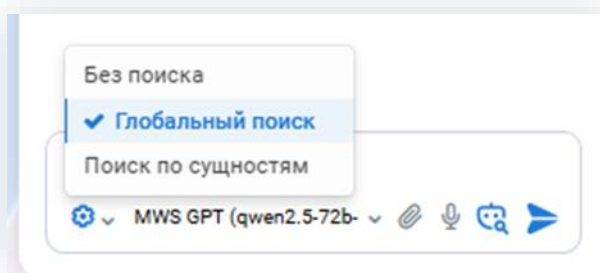


Рисунок 5

- «ChatBotSmartSearch» – позволяет выполнять умный поиск по Системе с помощью встроенного Чат-бота.

Настройки Чат-бота доступны в меню «Администрирование» – «Системные параметры» – «Чат-бот».

Системные параметры

Интеграция с УЦ Контур

Сервер ЭП

Интеграция с УС

Интеграция с ЭДО

Active Directory

Служба поддержки

Параметры

Почтовый клиент

Чат-бот

Инструкции

☐ Использовать ранкер

Количество фрагментов до ранкера *

30

Количество фрагментов RAG *

5

Макс. длина сообщения пользователя (в символах) *

4 000

Макс. длина ответа (в токенах) *

1 000

Доступные экраны (умный поиск)

Выберите значение

Очистить

df\$Contract.browse (Список договоров)

×

df\$SimpleDoc.browse (Список документов)

×

tm\$Task.browse (Список задач)

×

Доступные папки действий (умный поиск)

Выберите значение

Очистить

AppFolder_businessTrip.approval (Согласование)

×

AppFolder_summaryVacationSchedule.approval (Утверждение)

×

Доступные папки поиска (умный поиск)

Выберите значение

Очистить

Макс. размер истории для подготовки запроса (умный поиск) *

4 000

Макс. длина информации о найденных записях (умный поиск) *

5 000

☐ Передавать историю чата в служебные этапы (умный поиск)

Скрипт сообщения пользователя

```

1 package com.haulmont.thesis.core.app.chatbot
2
3 import com.haulmont.cuba.core.global.AppBeans
4 import com.haulmont.cuba.core.global.MetadataTools
5 import com.haulmont.cuba.core.global.TimeSource
6 import com.haulmont.cuba.core.global.UserSessionSource
7 import com.haulmont.thesis.core.macro.chatbot.ChatBotConversation
8 import com.haulmont.thesis.core.macro.chatbot.FileBotItem
9 import com.haulmont.thesis.core.macro.chatbot.VectorSearchHitDataItem
10
11 import java.text.SimpleDateFormat
12
13 UserSessionSource userSessionSource = AppBeans.getUserSessionSource(NAME)
14 MetadataTools metadataTools = AppBeans.get(MetadataTools.NAME)
15 TimeSource timeSource = AppBeans.get(TimeSource.NAME)
16
17 String originalMessage = binding.getVariable('originalMessage')
18 List<VectorSearchHitDataItem> vectorSearchHits = (List<VectorSearchHitDataItem>) binding.getVariable('vectorSearchHits')
19 List<FileBotItem> files = (List<FileBotItem>) binding.getVariable('files')
20 List<String> additionalData = (List<String>) binding.getVariable('additionalData')
21 ChatBotConversation conversation = (ChatBotConversation) binding.getVariable('conversation')
22
23 boolean hasVectorSearchHits = vectorSearchHits != null && !vectorSearchHits.isEmpty()
24 boolean hasFiles = files != null && !files.isEmpty()
25 boolean hasAdditionalData = additionalData != null && !additionalData.isEmpty()
26 String userName = metadataTools.getInstanceName(userSessionSource.userSession.currentSubstitutedUser)
27 String dateStr = new SimpleDateFormat("dd.MM.yyyy HH:mm:ss").format(timeSource.currentTimestamp())
28
29 """
30 s!{hasVectorSearchHits} : "" : ""
31 Ответ на вопрос пользователя по документам, хранящимся в СЭД ТЕЗИС, или по самой СЭД ТЕЗИС, на основании переданных тебе фрагментов текста.
32 не говори "на основе переданных фрагментов текста" или "на основе найденной информации", отвечай сразу по сути.
33 Наиболее релевантные фрагменты идут вначале.
34 каждый фрагмент заключен в xml-подобный тег в формате <rag_fragment id="id фрагмента">, после которого идет текст фрагмента.
35 Исходное сообщение пользователя:
36 $originalMessage
37
38 Ссылайся на фрагменты в формате <quote id="id фрагмента"/> (внутри тега quote ничего не пиши)
39
40 Информация, которая может быть полезной:
41 - текущий пользователь: ${userName}
42 - текущая дата и время: ${dateStr}
43
44 Вот фрагменты:
45
46 - ${vectorSearchHits}
47

```

Промпт подготовки вопроса (умный поиск)

Промпт открытия экрана (умный поиск)

Промпт применения фильтров (умный поиск)

Промпт финальной стадии (умный поиск)

Рисунок 6

12

3. ИИ-ассистент

Создание ИИ операций происходит через раздел меню «Администрирование» – «Базовые настройки» – «ИИ операции». Отображаться ИИ операции могут в текстовых полях (в т.ч. в Чат-боте) и вложениях.

Запускать ИИ операции может пользователь с ролью «AIOperationRole».

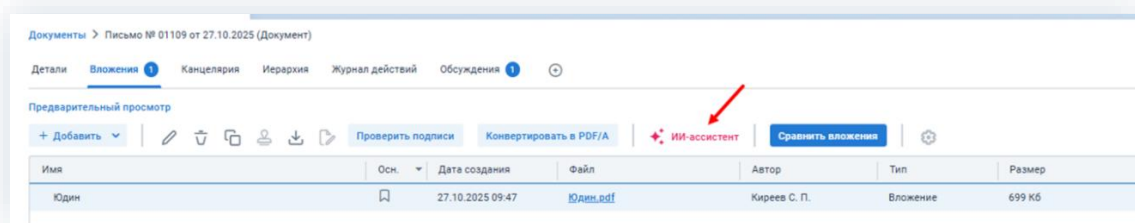


Рисунок 7