Документация по LLMTracer

Модуль **LLMTracer** предназначен для трассировки запросов к LLM, а также для сбора статистики по токенам, времени выполнения и ошибкам. Все данные записываются в таблицу `llm_tracer`.

Класс: Session

Подключается через

```
let LLMTracer = require("Common.AIHelpers.LLMTracer")
```

createSession(options)

Создаёт новую сессию трассировки.

Пример:

```
let session = LLMTracer.createSession({
  business_id: "123",
  bot_id: "456",
  lead_id: "789",
  script_id: "script_1",
  command_id: "cmd_1",
  agent_name: "MyAgent",
  provider: "OpenAI",
  model: "gpt-4"
})
```

Параметры:

Имя	Тип	Описание
business_id	String	Идентификатор бизнеса
bot_id	String	Идентификатор бота
lead_id	String	Идентификатор лида
script_id	String	Идентификатор скрипта
command_id	String	Идентификатор команды
agent_name	String	Имя агента
provider	String	Название провайдера LLM
model	String	Название модели LLM

Return
Session — Экземпляр сессии трассировки.

---

getCurrentSession()

Получает текущую сессию из атрибута лида.

Пример:

```
let session = LLMTracer.getCurrentSession()
```

Параметры:
Нет

Return
Session | null — Текущий объект сессии или null, если не найден.

---

prepareRecordData(data)

Преобразует данные в безопасный формат для записи в таблицу.

Пример:

```
let safeData = LLMTracer.prepareRecordData({ message: "Тест", status: "ok" })
```

Параметры:

Имя	Тип	Описание

data	Object	Объект с исходными данными

Return
Object — Объект с преобразованными значениями.

---

recordTrace(traceData)

Записывает трассировку в таблицу `llm_tracer`.

Пример:

```
LLMTracer.recordTrace({
  agent_name: "MyAgent",
  outclient_name: "ClientA",
  step: "step1",
  outclient_time: 120,
  provider: "OpenAI",
  model: "gpt-4",
  type: "info",
  message: "Запрос выполнен",
  record: {},
  params: {},
  status: "ok",
  input_tokens: 100,
  output_tokens: 50
})
```

Параметры:

Имя	Тип	Описание
agent_name	String	Имя агента
outclient_name	String	Имя внешнего клиента
step	String	Название шага
outclient_time	Number	Время выполнения шага (мс)
provider	String	Название провайдера LLM
model	String	Название модели LLM
type	String	Тип трассировки (info, error и др.)
message	String	Сообщение

Имя	Тип	Описание
record	Object	Дополнительные данные
params	Object	Параметры запроса
status	String	Статус выполнения
input_tokens	Number	Количество входных токенов
output_tokens	Number	Количество выходных токенов

Return
Boolean — true при успехе, false при ошибке.

---

recordTraceLLM(llm, timerCtrl, traceData)

Записывает трассировку по активному LLMClient.

Пример:

```
LLMTracer.recordTraceLLM(llmClient, "START_TIME", { message: "Старт" })
```

Параметры:

Имя	Тип	Описание
llm	Object	Экземпляр LLMClient
timerCtrl	String	Контроллер времени ("", "START_TIME", "END_TIME")
traceData	Object	Дополнительные параметры трассировки

Return
Boolean — true при успехе, false при ошибке.

---

recordTraceQuery(llm, timerCtrl, traceData)

Записывает трассировку для запросов к внешним сервисам.

Пример:

```
LLMTracer.recordTraceQuery(llmClient, "END_TIME", { message: "Финиш" })
```

Параметры:

Имя	Тип	Описание
llm	Object	Экземпляр LLMClient
timerCtrl	String	Контроллер времени ("", "START_TIME", "END_TIME")
traceData	Object	Дополнительные параметры трассировки

Return
Boolean — true при успехе, false при ошибке.

---

info(message, data)

Записывает информационную трассировку.

Пример:

```
LLMTracer.info("Запрос выполнен", { step: "step1" })
```

Параметры:

Имя	Тип	Описание
message	String	Текст сообщения
data	Object	Дополнительные данные

Return
Boolean — true при успехе, false при ошибке.

---

error(message, data)

Записывает ошибку в трассировку и отправляет сообщение админу.

Пример:

```
LLMTracer.error("Ошибка запроса", { code: 500 })
```

Параметры:

Имя	Тип	Описание
message	String	Текст ошибки
data	Object	Дополнительные данные

Return
Boolean — true при успехе, false при ошибке.

---

countTokens(dateRange)

Подсчитывает количество токенов за указанный период.

Пример:

```
LLMTracer.countTokens({ from: "2024-06-01 00:00:00", to: "2024-06-30 23:59:59" })
```

Параметры:

Имя	Тип	Описание
from	String	Начальная дата ("YYYY-MM-DD HH:MM:SS")
to	String	Конечная дата

Return
Object — { input_tokens, output_tokens, total_tokens }

---

getActiveSessionsCount(minutes)

Подсчитывает количество активных сессий за последние N минут.

Пример:

```
LLMTracer.getActiveSessionsCount(10)
```

Параметры:

Имя	Тип	Описание

minutes	Number	Глубина окна в минутах (по умолчанию 5)

Return
Number — Количество активных сессий.

Быстрый старт

1. Создайте сессию: `LLMTracer.createSession({ ... })`
2. Запишите шаг: `LLMTracer.recordTrace({ ... })`
3. Для LLM: используйте `recordTraceLLM()`
4. Для ошибок: используйте `error()`
5. Для статистики: используйте `countTokens()`

Версия #2
Павел Борисов создал 2025-11-06 11:21:33 UTC
Павел Борисов обновил 2025-11-12 14:03:47 UTC