Cognitum AI Platform

0.16.6 → **0.17.0** для администратора · 13 ноября 2026

Действие — по решению человека

Политика подтверждения вызовов на трёх уровнях и сигнал SDK для процессных агентов. Виджеты MCP Apps, оценки с выгрузкой, маршрут пула и отдельный сервис tool-runner.



Подтверждения
Политика, очередь, SDK

MCP Apps и оценки
Виджеты, выгрузка JSONL

Пул и базы
Отказ, маршрут, таблицы

Поставка
tool-runner, шина, .env

3
уровня политики: сервер, инструмент, агент

1
новый сервис — `tool-runner`

0
правок в агентах на SDK 0.16

Что нового в каждом сервисе — и один новый

Решения о вызовах принимает app. «Офис» и «Сайты» переехали из харнесса в tool-runner, шина — с учётными записями.

Workspace и панель управления

браузер

карточка подтверждения · «Ждут моего решения» · виджеты MCP Apps · лайк / дизлайк · панель плана

Внешние MCP-серверы

Jira, почта, CRM…

аннотации readOnlyHint / destructiveHint · ui://-ресурсы виджетов

HTTP

MCP через app

app

cognitum-app 0.17.0

политика подтверждения и срок решения · очередь «Ждут моего решения» · прокси MCP Apps · оценки и выгрузка JSONL · «Вернуть ошибку» и «Маршрут» пула · сброс пароля · таблицы по строкам · тип параметра knowledge_base

NATS с JetStream учётная запись у каждого сервиса

nocode-harness

0.17.0

План и подсказки, виджеты, подтверждение вызовов. Образ без офиса и браузера.

tool-runner

0.17.0 · НОВЫЙ

«Офис» и «Сайты» отдельным сервисом. Масштабируется: `--scale tool-runner=2`.

Агенты на SDK 0.17

echo-agent и процессные

`ConfirmationRequired`, `wait_approval`, виджеты у MCP-сервера агента.

kb-parser

0.17.0

Таблицы «По строкам»: строка — фрагмент с путём колонки из многоуровневой шапки.

Пароли сервисов

`./deploy.sh nats-passwords` — до пересоздания контейнеров

Внешний агент

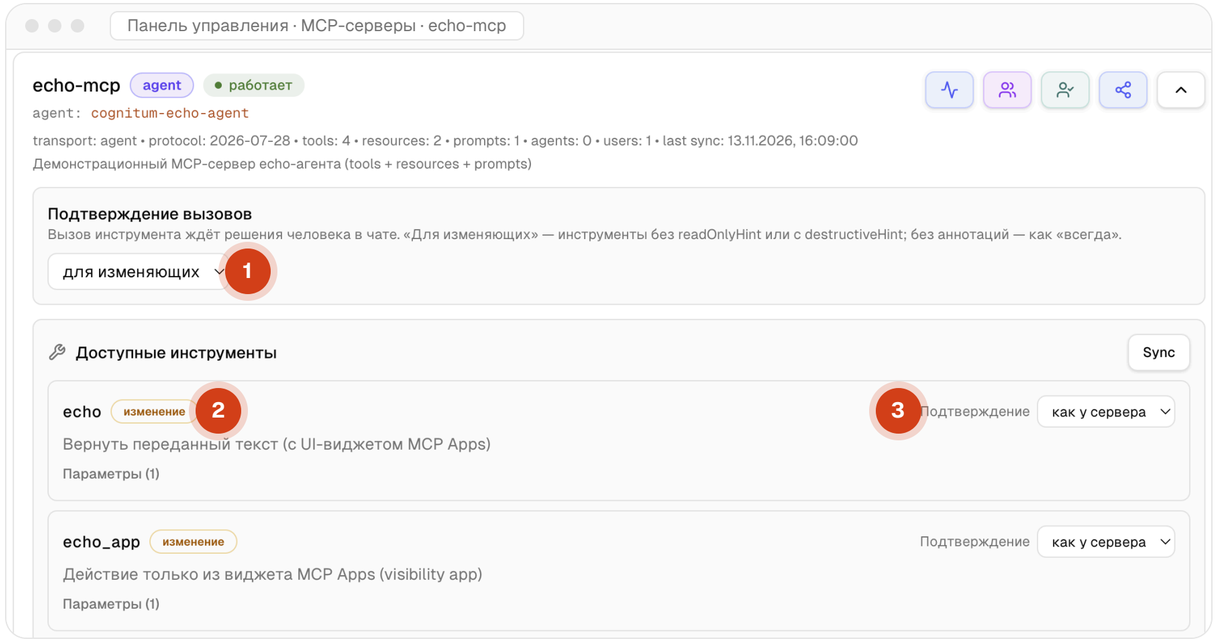
Своя учётная запись одной командой `nats-add-agent.sh`

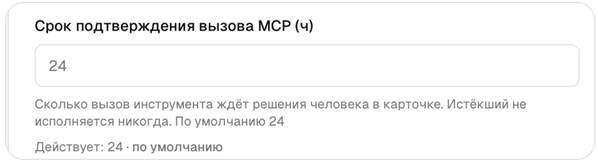
Агенты на других серверах

Порт шины не открыт на хосте — `docker-compose.nats-public.yml`

Что спрашивать — на сервере, инструменте и агенте

По умолчанию подтверждение выключено. Включают его на MCP-сервере; инструмент и агент уточняют правило.





Срок решения — в «Настройках Workspace», по умолчанию **24 часа**. Истёкший вызов не выполнится никогда.

1 **Сервер — «Подтверждение вызовов»**

никогда | для изменяющих | всегда

«Для изменяющих» — без `readOnlyHint` или с `destructiveHint`; без аннотаций — как «всегда»

2 **Инструмент — бейдж и переопределение**

как у сервера | всегда | никогда

3 **Агент — привязка к серверу**

наследовать | для изменяющих | всегда

Только ужесточает: ослабить правило сервера для агента нельзя

Итог: **правило инструмента**, если задано; иначе — **строже** из правил сервера и агента.

Кто и когда решил

В карточке вызова и в журнале аудита

Агент на SDK 0.17

После «Выполнить» повторяет вызов — платформа выполняет его один раз

Процессный агент получает понятный сигнал

ConfirmationRequired: из очереди платформа выполнит вызов сама, из чата — агент повторит его.

SDK 0.17.0 · ПРОЦЕССНЫЙ АГЕНТ

```python
from cognitum_agent import ConfirmationRequired

try:
    res = await agent.invoke_mcp_tool(
        server_id=7, tool="create_issue", arguments=args)
except ConfirmationRequired as exc:
    if exc.queued:   # «Ждут моего решения»
        state = await agent.wait_approval(exc.request_id)
        # платформа выполнила вызов сама: status, result_status
    else:            # чат: карточка у собеседника
        await agent.wait_approval(exc.request_id)
        res = await agent.invoke_mcp_tool(
            server_id=7, tool="create_issue", arguments=args,
            hitl_request_id=exc.request_id)
```

`ConfirmationRequired`

Наследник `RuntimeError`: `request_id`, `expires_at`, `preview`, `queued`. Прежний `except RuntimeError` его ловит

`wait_approval`

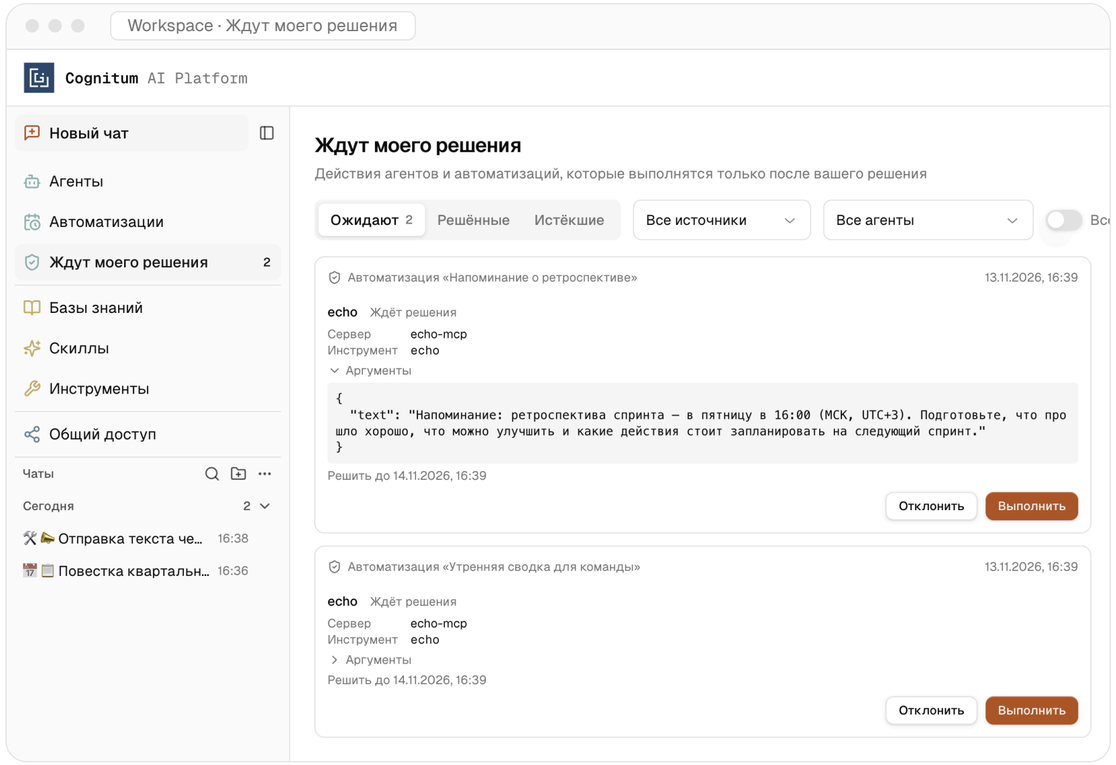
Ждёт события, пропущенное добирает опросом раз в `poll_interval`; по `timeout` — `asyncio.TimeoutError`

`run_with_mcp_tools`

Отдаёт модели статус `confirmation_queued` или `confirmation_required`, а не ошибку

Совместимость

Агенты на SDK 0.17.0 требуют платформу 0.17.0



Кто решает

Владелец автоматизации; у процессного агента — сотрудник, от чьего имени идёт работа, иначе владелец агента

Как узнают

Счётчик в меню и колокольчике, письмо — если настроена почта. Запуск автоматизации — «Частично»

Код — по sdk/README.md, идентификаторы условные; скриншот со стенда

Виджет — это ресурс ui:// у инструмента

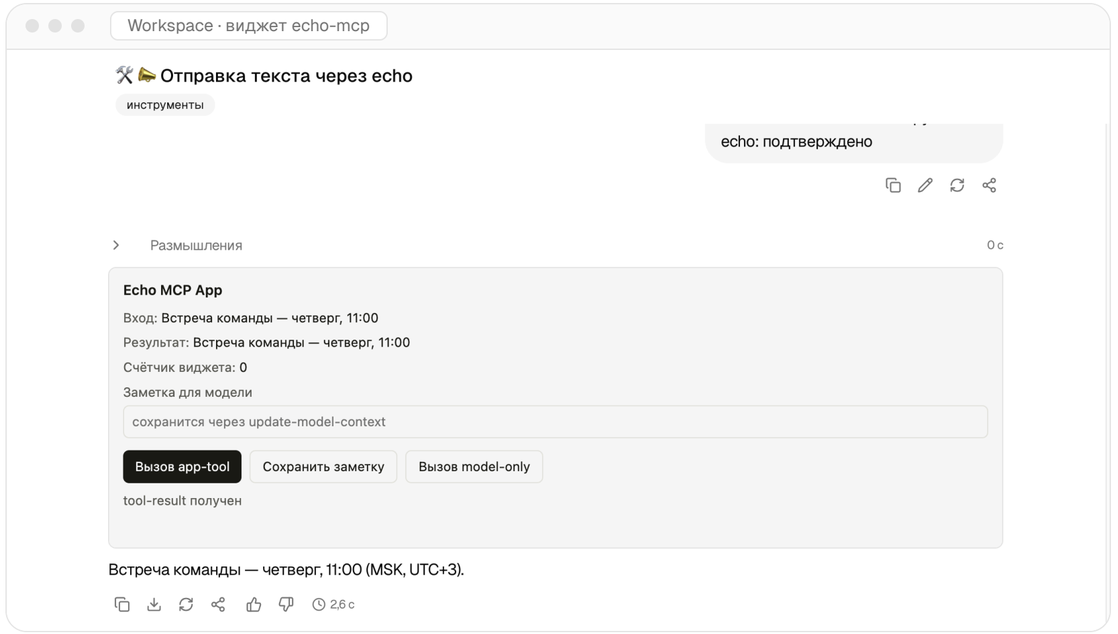
Инструмент ссылается на HTML-ресурс сервера, платформа показывает его в чате в изолированной области.

MCP-СЕРВЕР ПРОЦЕССНОГО АГЕНТА · ECHO-AGENT

```python
@agent.mcp_tool(
    "echo-mcp", name="echo", input_schema=schema,
    meta={"ui": {"resourceUri": "ui://echo/widget",
                 "visibility": ["model", "app"]}},
)
async def mcp_echo(arguments): ...

@agent.mcp_resource(
    "echo-mcp", uri="ui://echo/widget",
    name="Echo MCP App", mime_type="text/html",
)
async def widget(uri: str):
    return Path("widgets/echo-mcp-app.html").read_text()
```

`["model", "app"]`	**echo**	Вызывает модель, виджет показывает результат
`["app"]`	**echo_app**	Только из виджета — модели не виден
`["model"]`	**echo_model**	Только модель — из виджета отказ



Изоляция и состояние

Виджет ходит только к своему серверу через платформу; состояние хранится с перепиской, модель видит его актуальным

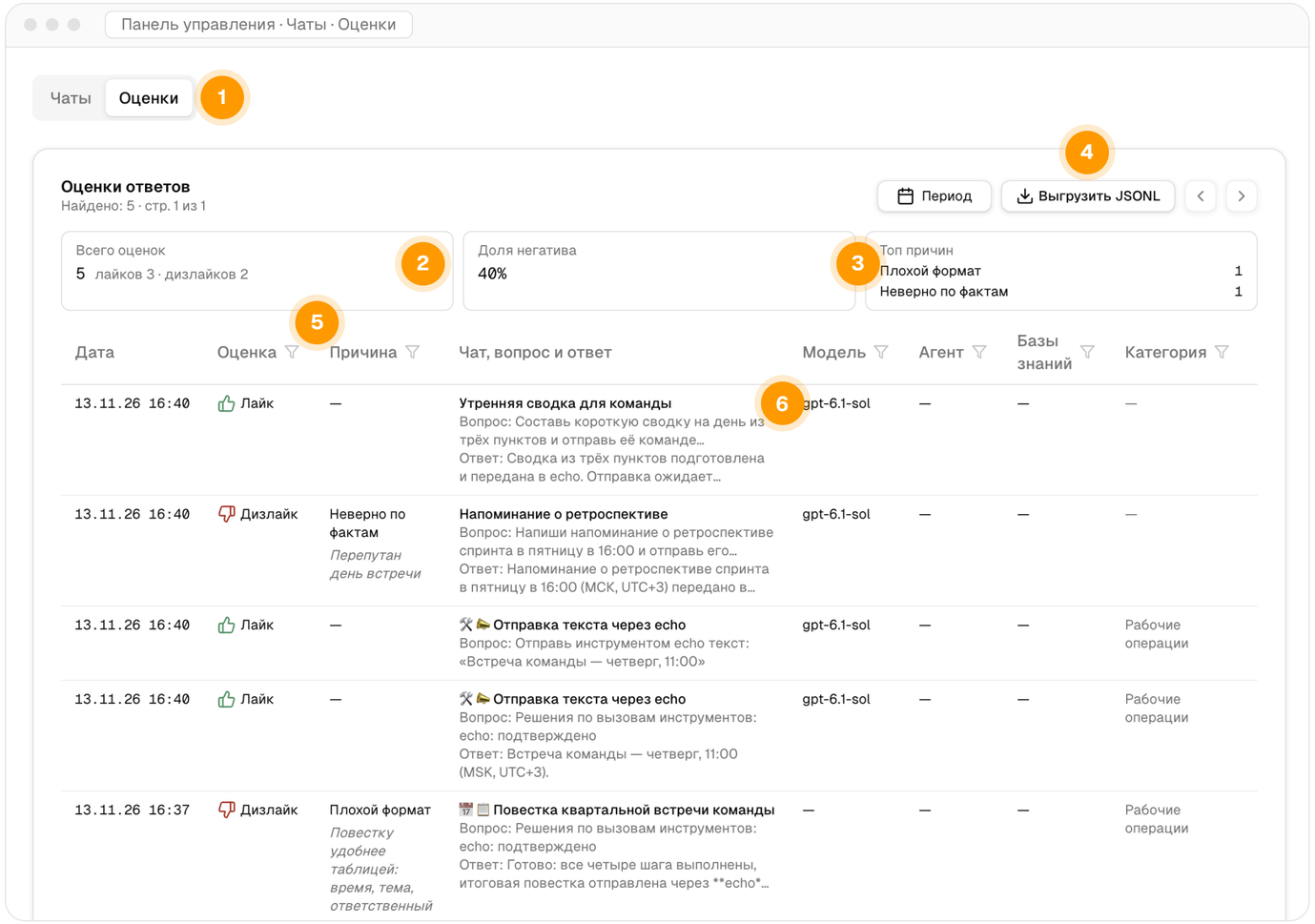
Бейдж `agent`

MCP-сервер процессного агента подключается к NoCode-агенту без личного подключения; агент остановлен — «недоступен»

Код — по agents/echo-agent, сокращён; скриншот со стенда

Где агент ошибается — сводка и выгрузка

«Чаты» → «Оценки»: доля негатива, топ причин, фильтры и файл JSONL для разбора и дообучения.



1. **Вкладка «Оценки»**
   Рядом с «Чатами»; в списке чатов — число дизлайков
2. **Сводка и доля негатива**
   Всего оценок, лайков и дизлайков
3. **Топ причин**
   Шесть причин дизлайка
4. **«Период» и «Выгрузить JSONL»**
   Выгрузка фиксируется в аудите
5. **Фильтры**
   Оценка, причина, модель, агент, база знаний, категория чата
6. **Переход к ответу**
   Чат, вопрос и ответ в строке

Неверно по фактам · Не нашёл в базе · Не по делу · Плохой формат · Отказался отвечать · Другое

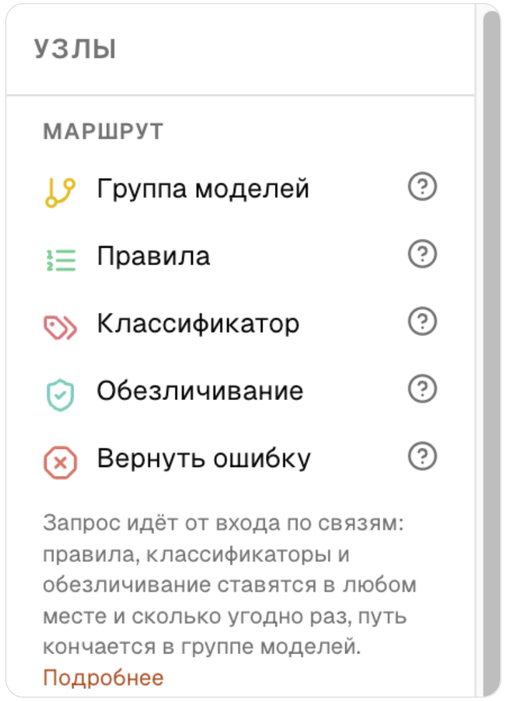
Оценка хранится **вместе с диалогом**: администратор видит причину, комментарий и сам ответ.

Скриншот со стенда: оценки поставлены тестовой учёткой

«Вернуть ошибку» — отказ без повторов

Узел на выходе «Ошибка обезличивания»: запрос никуда не уходит, а путь каждого вызова виден в «Маршруте».

ПАЛИТРА УЗЛОВ



МАРШРУТ С ОТКАЗОМ

Вход
чат-пул «default»

Обезличивание
заменяет персональные данные метками

«Обезличено» →

Группа «Внешние»
внешняя · только обезличенные

«Ошибка обезличивания» →

Вернуть ошибку
Не обезличили — ошибка без повторов

`Пул «default»: запрос не отправлен – причина` — 503, если сервис обезличивания не запущен или не ответил, 400 — если обезличивание запрещено.

Чат, агенты и NoCode-харнесс ошибку не повторяют; failover нет. Альтернатива на том же выходе — локальная группа.

Карточка вызова · Маршрут макет

ПУТЬ

1. Фильтры: отсеяно моделей — 0, уцелело групп — 2
2. Правила: в группу «Внешние»
3. Обезличивание: запрос обезличен
4. Модель «gpt-6.1-sol», запасные: claude-opus-5.5

ОБЕЗЛИЧИВАНИЕ

140 мс — обезличено

`PERSON × 2` `PHONE × 1`

Исходный | **Ушло в модель** | до восстановления значений

```
Подготовь письмо <PERSON_1>: договор согласован, подписание у <PERSON_2>, телефон <PHONE_1>.
```

Трейс чата · отказ макет

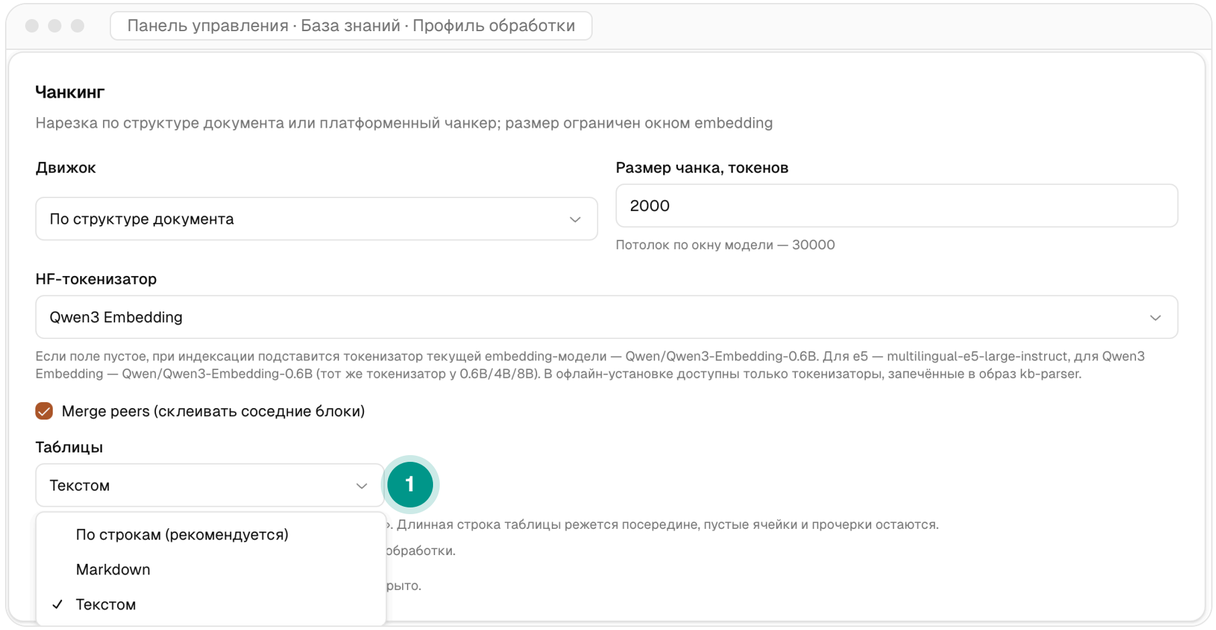
1. Обезличивание: выход «Ошибка обезличивания» — чувствительный запрос
2. Пул «default»: запрос не отправлен — чувствительный запрос

Обезличить не удалось — во внешнюю модель запрос не ушёл

Палитра — скриншот со стенда; маршрут, карточка вызова и трейс — макет, данные вымышлены

Таблицы по строкам и параметр knowledge_base

Режим нарезки таблиц — в профиле обработки. Разработчик агента объявляет параметр «база знаний» одной строкой схемы.



1 **«Таблицы»: «По строкам (рекомендуется)», «Markdown», «Текстом»**
Новые базы — по строкам сразу. Существующие остаются в прежнем режиме; новый действует после переобработки

ФРАГМЕНТ «ПО СТРОКАМ»

Склад №2 — **III квартал / План, шт.:** 1 200; **III квартал / Факт, шт.:** 1 135; **Выполнение, %:** 94,6

SETTINGS_SCHEMA АГЕНТА · SDK 0.17.0

```
{
    "name": "practice_kb",
    "type": "knowledge_base",
    "description": "База практики",
    "required": True
}

kb_name = agent.get_setting("practice_kb")
hits = await agent.search_knowledge(kb_name, query)
```

Настройки агента макет

База практики *

Архив-2025

База «Архив-2025» не привязана к агенту

Список и проверка
Только активные базы, привязанные к агенту. Непривязанную сохранить нельзя — ответ 400 с именем поля

Переход со `string`
Сменить `type` — сохранённое значение остаётся, код агента не меняется

Профиль — скриншот со стенда; схема — по sdk/docs/AGENT_CONFIG.md; поле и фрагмент — макет

Что поменялось в .env и «Настройках»

Обязательна одна новая переменная — образ tool-runner. Пароли шины задаёт одна команда, остальное — по желанию.

ПЕРЕМЕННАЯ	ЗАЧЕМ	ПО УМОЛЧАНИЮ
`TOOL_RUNNER_IMAGE`	Образ сервиса `tool-runner`; из реестра — `…/cognitum-tool-runner:0.17.0`	`cognitum-tool-runner:latest`
`NATS_*_PASSWORD`	Пароли `app`, `nocode-harness`, `tool-runner`, `echo-agent` на шине — `./deploy.sh nats-passwords`	пусто
`NATS_ALLOW_ANONYMOUS`	Вход без пароля — для агентов на SDK до 0.17.0, обезличивания и классификатора пула. В 0.17.0 оставьте `true`	`true`
`KB_INGEST_THREADS`	Потоки обработки документов баз знаний — чтобы разбор больших файлов не тормозил платформу	`2`
`LOOP_LAG_WARN_S`	Задержка обработки запросов `app` дольше порога — предупреждение в лог	`1.0`
`NATS_MONITORING_PORT`	Больше не публикуется на хост; строку можно удалить	удалена

«НАСТРОЙКИ» ПЛАТФОРМЫ

«Повтор при немом соединении»
«Немое» соединение к модели обрывается и повторяется

«Таймаут нестриминговых вызовов API»
Свой таймаут для вызовов `stream: false`

«Срок подтверждения вызова MCP (ч)»
Сколько вызов ждёт решения; по умолчанию 24

Размышления «нет»
Выключают их и у моделей с флагом `enable_thinking`

ВНЕШНИМ СКРИПТАМ API

`422`
JSON без `Content-Type: application/json`

`401` **вместо** `403`
Запрос без учётных данных

0.16.6 → 0.17.0: агенты на SDK 0.16 — без правок

Шаги — по инструкции обновления, она подходит и для 0.16.5. Миграции базы применяются сами при старте app.

1 Файлы из дистрибутива

`docker-compose.yml`, `nats-config.conf`, `nats-entrypoint.sh`, `nats-add-agent.sh`, `docker-compose.nats-public.yml`, `nats-agents/`

2 Образы 0.17.0 и `TOOL_RUNNER_IMAGE`

Новый образ `cognitum-tool-runner` заменяет `nocode-tools`; `kb-parser` 0.17.0

```
docker compose pull
```

3 Пароли шины — до пересоздания

```
./deploy.sh nats-passwords
```

4 Пересоздать контейнеры

```
docker compose up -d --force-recreate --remove-orphans
```

5 Проверить «Самопроверку»

Строка `tool-runner`; «Шина» жёлтая, пока есть агенты на старом SDK, обезличивание и классификатор

6 Агенты на других серверах

Подключить `docker-compose.nats-public.yml`; учётная запись — `nats-add-agent.sh`

`platform`
0.16.6 → **0.17.0**
политика, очередь, MCP Apps, оценки, пул

`sdk`
0.16.6 → **0.17.0**
ConfirmationRequired, виджеты, «база знаний»

`nocode-harness`
0.16.6 → **0.17.0**
план, подсказки; без офиса и браузера

`tool-runner`
новый → **0.17.0**
«Офис» и «Сайты», свой пайплайн

`deploy`
0.16.6 → **0.17.0**
шина с паролями, порт не на хосте

`echo-agent`
0.16.6 → **0.17.0**
SDK 0.17.0, демо-виджет

Резервная копия

- Пароли `NATS_*_PASSWORD` — в `.env`, попадают в копию вместе с ним
- Учётные записи внешних агентов — `config/nats-agents/nats-agents.conf`: без копии их придётся завести заново

Совместимость

SDK 0.16.x — пока NATS_ALLOW_ANONYMOUS=true · SDK 0.17.0 → платформа 0.17.0 · Реплики агента — на одной версии SDK · PostgreSQL 12+ · NATS с JetStream · Docker 24+ / Compose V2