

ТЕХНИЧЕСКИЙ АУДИТ

Планировщик Тимура — Telegram-бот @timplanner_bot и
мини-приложение

9 августа 2026, 13:40 МСК · аудит выполнен скиллом «техдир» R3KA · режим read-only, код не
правился

Находок всего	20
Из них критичных	5
Обещаний из канона проверено	11
Из них выполнено полностью	1 (автозапуск)
Данных мини-аппки доехало до бота за 11 часов	0 раз
Сообщений по расписанию отправлено	0
Вызовов нейросети в живом коде бота	0

ГЛАВНЫЙ ВЫВОД

Приложение и бот — это две разные программы с двумя разными базами, которые почти не разговаривают друг с другом. Всё, что ты заполнил в окне, осталось в облаке Телеграма и до бота не доехало ни разу. Всё, что бот записал, лежит в его собственной базе на сервере.

Из этого растёт всё остальное: бот тебя не знает (профиль пустой), поэтому молчит по утрам; окно показывает не то, что бот; а «умный ассистент» — это не «плохо работающий мозг», мозга в живом коде нет вообще: одна строка, которая любой текст кладёт в инбокс задач.

Хорошая новость: визуал, экраны, схема базы и автозапуск — сделаны нормально. Переделывать надо связь между частями, а не всё подряд.

ОБЕЩАНО → ЕСТЬ НА САМОМ ДЕЛЕ

Что обещали	Что есть на самом деле	Статус
Живой диалог с ботом, отвечает словами	Нейросети в живом коде нет ни строки. Любой текст механически падает в инбокс задач.	НЕТ
Знакомство: 7 вопросов о тебе	Вопросы в окне задаются, ответы сохраняются в облако Телеграма и до бота не доезжают. В базе бота профиль пустой.	НЕ ДОЕХАЛО
Брони пользовательские: создать, переименовать, иконка, частота, удалить	Всё это написано и работает, но живёт только внутри мастера знакомства. После знакомства попасть туда можно лишь через Профиль → строка «Брони» → мастер заново.	СПРЯТАНО
Брони регулируются с главного экрана	Кольца на главном — картинка. Обработчика нажатия на них нет.	НЕТ
Карточки таскаются пальцем	В приложении нет ни одного обработчика перетаскивания или касания. Ноль строк.	НЕТ
Задачи из бота залетают в окно моментально	Окно спрашивает сервер ровно один раз — в момент открытия. Дальше не спрашивает никогда.	НЕТ
Передача данных из окна боту	Технически возможна только при входе через нижнюю кнопку «Планировщик». За 11 часов не сработала ни разу.	НЕ РАБОТАЛО
Расписание: утро 08:30, вечер 21:00, воскресенье	Код написан и планировщик запущен, но рассылка идёт только «прошедшим знакомство». Ты в базе числишься не прошедшим — значит цикл каждую минуту крутится вхолостую.	НИ РАЗУ
Автозапуск переживает перезагрузку сервера	Работает: системная служба → движок → сторож бота, двойная страховка.	ДА
Голосовые расшифровываются	Код есть, ключ у живого бота в окружении есть. Живьём ни одного голосового не приходило — не проверено.	НЕ ПРОВЕРЕНО
Дыра с открытой папкой закрыта (отчёт 04:19)	Закрыт только просмотр списка файлов. Сам файл с твоими записями по-прежнему отдаётся из интернета любому, кто знает адрес.	НЕ ЗАКРЫТА

НАХОДКИ

Сначала критичные — то, из-за чего вообще ничего не работает.

КРИТИЧНО

К1. У бота нет мозга — любой текст механически становится задачей

КРИТИЧНО чинить ≈ 6-10 ч

Где: `planner_bot/bot.py:172-179`, функция `on_text` (живая версия от 04:13)

Почему так вышло: Ветвления не предусмотрели вообще: обработчик текста писался как «быстрая запись в инбокс» и таким и остался. Нейросеть в этот код никогда не подключали.

Что видит Тимур: «Я все заполнил» и «Че ты записал?» стали задачами 01 и 02.

Пруф:

```
grep -niE "openai|chat.completions|gpt|tool_call" bot.py(живой)
→ 0 совпадений
```

```
bot.py:172
async def on_text(update, ctx):
    """Любой текст — быстрая запись в инбокс."""
    ...
    await stash(update, u.id, txt)
```

```
SELECT * FROM inbox:
id=5 num=1 "Я все заполнил" 13:10:39
id=6 num=2 "Че ты записал?" 13:12:22
id=7 num=3 "С тобой можно общаться?" 13:16:49
```

К2. Две несвязанные базы: окно пишет в облако Телеграма, бот — к себе на сервер

КРИТИЧНО чинить ≈ 12-16 ч
(переписывать)

Где: `public/planner/store.js` (CloudStorage/localStorage) против `planner_bot/db.py` (SQLite на сервере)

Почему так вышло: Домен `tim.r3ka.ru` стоит на чужом сервере 151.243.145.216, куда нет доступа, — поэтому своего API под приложение не подняли, а данные положили в облако Телеграма. Общего источника правды не стало.

Что видит Тимур: Окно и бот показывают разное, ничего не сходится.

Пруф:

```
store.js:8-17 — cloud = tg.CloudStorage, иначе localStorage
db.py:8 — DB_PATH = /home/alesha/gorilla/_state/planner/planner.db
```

```
Строк в базе бота по Тимуру (697785207):
users 1 | profile 0 | bookings 0 | tasks 0 | snapshots 0 | inbox 3
```

К3. Канал «окно → бот» не сработал ни разу за 11 часов

КРИТИЧНО чинить ≈ 2-3 ч

Где: app.js:1287 handOff() → tg.sendData; bot.py:67-74 sync_kb (единственный вход с передачей); bot.py:60-64 и post_init — все остальные входы без неё

Почему так вышло: Телеграм разрешает передачу данных из мини-приложения только когда его открыли нижней кнопкой клавиатуры. Меню-кнопка и inline-кнопка ведут на адрес без пометки sync — оттуда передать физически нельзя. Ты заходил именно так.

Что видит Тимур: Прошёл знакомство — бот об этом не узнал.

Пруф:

```
grep -c "sync from" planner_bot.log → 0 (за 11 часов работы)
SELECT COUNT(*) FROM snapshots → 0

bot.py:67
def sync_kb(uid):
    """Кнопка на клавиатуре — только из неё мини-аппка умеет отдать снимок боту."""
    return ReplyKeyboardMarkup([[KeyboardButton("Планировщик",
web_app=WebAppInfo(url_for(uid, "&sync=1")))]], ...)

bot.py:60 open_kb() — inline-кнопка, БЕЗ &sync=1
post_init set_chat_menu_button(url_for(uid)) — тоже БЕЗ &sync=1
```

К4. Расписание физически не могло сработать ни разу

КРИТИЧНО чинить ≈ 1 ч (после К3)

Где: bot.py:503 (цикл tick), db.py:424 (onboarded_users), db.py:219 (единственное место, где ставится флаг)

Почему так вышло: Рассылка идёт по списку «прошедших знакомство». Флаг ставится только в момент приёма снимка из окна — а снимок не приходил (К3). Список всегда пустой, цикл крутится вхолостую раз в минуту.

Что видит Тимур: Ни утренних трёх дел, ни вечернего разбора, ни воскресного итога — тишина.

Пруф:

```
bot.py:503
for uid in db.onboarded_users(): # ← список пуст

db.py:424
SELECT tg_id FROM users WHERE onboarded=1

db.py:219 (внутри apply_snapshot, вызывается только из on_webapp_data)
UPDATE users SET onboarded=1 WHERE tg_id=?

SELECT onboarded FROM users WHERE tg_id=697785207 → 0
SELECT COUNT(*) FROM sent → 0 (ни одного отправленного по расписанию)
grep -c "morning_sent|evening_sent" в логe → 0
```

K5. Чат внутри окна превращал любую фразу в карточку на день

КРИТИЧНО чинить ≈ 0.5 ч
(удалить)

Где: app.js, функция planSend (~строка 1076), привязка на app.js:1108

Почему так вышло: В окне нет доступа ни к базе, ни к нейросети, поэтому «чат» собрали на самодельном разборе текста: ищет день недели, ищет время, остальное считает названием задачи. Если дня не нашлось — ставит на сегодня.

Что видит Тимур: Вопрос «Почему ты брони расставил так?» стал карточкой на воскресенье 9, а бот ответил «Поставил на воскресенье, 9».

Пруф:

```
app.js, planSend():
if (found === null) found = (TODAY.getDay() + 6) % 7; // ← сегодня
...
S.week.tasks.push({ id: nid(), title: title, day: iso(dates[found]), ... });
S.week.chat.push({ r: "a", t: "Поставил на <b>" + DAYS[found].n.toLowerCase()
+ ", " + dates[found].getDate() + "</b>" ... });
```

Сегодня 09.08.2026 — воскресенье. Совпадает со скриншотом дословно.

Поле на месте и сейчас — живая проверка задеплоенной версии:
плейсхолдер «Напиши, что на неделе» присутствует: True

БОЛЬНО

Б1. Инбокс обновляется один раз за открытие — опроса нет

БОЛЬНО чинить ≈ 1-2 ч

Где: app.js:1451 — единственный вызов pullBridge()

Почему так вышло: Мост писали как «подхватываем при открытии». Ни таймера, ни реакции на возврат в приложение не заложили.

Что видит Тимур: Записал что-то боту — в окне пусто, пока не закроешь и не откроешь заново.

Пруф:

```
Живой прогон задеплоенного приложения в браузере (headless):
запросов к /sync/ на старте: 1
запросов к /sync/ за следующие 15 секунд: 0

grep -c "setInterval|visibilitychange" app.js → 0

Кэш ни при чём: запрос уже идёт с ?t=<время> и cache:no-store,
а nginx отдаёт файл вообще без Cache-Control (проверено curl -I).
```

Б2. Брони не регулируются: кольца на главном — просто картинка

БОЛЬНО чинить ≈ 2-3 ч

Где: app.js:358 — кольца только рисуются; редактор брони openBookSheet вызывается лишь из мастера знакомства (app.js:943, 946) и через Профиль → строка → мастер заново (app.js:1236-1240)

Почему так вышло: Экран правки броней сделали частью мастера знакомства и наружу не вынесли. После знакомства он остался без входа.

Что видит Тимур: Тапаешь по кольцу «Зал 0/3» — ничего не происходит.

Пруф:

```
grep -n "home-rings" app.js
358: rings($("#home-rings"), bookingProgress(S.week));
→ единственное упоминание, обработчика нажатия нет

app.js:943 (внутри слушателя #onb-ctl — мастер знакомства)
if (op) { openBookSheet(op.closest("[data-bid]").dataset.bid); ... }

Живая проверка: элементы на главном — div.ring без слушателей,
["Сын 0/2", "Жена 0/1", "Отдых 0/1", "Зал 0/3", "Массаж 0/1"]
```

Б3. Перетаскивания нет вообще — ни одной строки кода

БОЛЬНО чинить ≈ 4-6 ч

Где: public/planner/app.js целиком

Почему так вышло: Функцию обсуждали, но никогда не писали. При этом приложение само пишет тебе «подвинь, если не так».

Что видит Тимур: Карточку не сдвинуть ни пальцем, ни мышкой.

Пруф:

```
grep -c "dragstart|dragover|draggable|touchstart|touchmove|touchend|
pointerdown|pointermove" app.js
→ 0

При этом app.js (seedFirstWeek) отправляет тебе сообщение:
"Брони расставил по неделе — подвинь, если не так."
```

Б4. Брони раскидались по дням молча, тебя не спросили

БОЛЬНО

чинить ≈ 1-2 ч

Где: app.js, функция seedFirstWeek (~1014), вызывается из завершения знакомства (app.js:1004)

Почему так вышло: Чтобы экран не был пустым, при первом входе брони автоматически раскладываются «через день» — slot += 2 по кругу. Ни подтверждения, ни смысла в расстановке.

Что видит Тимур: «Я не давал тебе такого указания.»

Пруф:

```
app.js:1004
if (wasNew) seedFirstWeek();

function seedFirstWeek() {
var slot = 0;
S.profile.bookings.forEach(function (b) {
for (var i = 0; i < b.t; i++) {
var d = dates[slot % 7];
slot += 2; // ← вся «логика» расстановки
S.week.tasks.push({ ... day: iso(d) ... });
```

Б5. Знакомство собирает 7 ответов и никому их не отдаёт

БОЛЬНО

чинить ≈ вместе с К3

Где: app.js, список QUESTIONS (~720-765); ответы уходят в S.profile → облако Телеграма

Почему так вышло: Знакомство сделали анкетой внутри окна, а не разговором с ботом. Ответы легли туда же, куда и всё остальное — в облако, мимо бота (К2, К3).

Что видит Тимур: «Стартовой инфы обо мне ни приложение, ни бот не взяли.»

Пруф:

```
Вопросы в коде есть, все 7:
«Что должно быть в неделе обязательно?» (брони)
«Ты сейчас в форме или пустой?»
«Сколько тяжёлых дел в день ты вывозишь?»
«Что сейчас давит сильнее всего?»
«Когда собираем неделю?»
«Как ты ломаешься?»
«Как с тобой говорить, когда ты злой?»

SELECT COUNT(*) FROM profile WHERE tg_id=697785207 → 0
```

Б6. Лог бесполезен: пишутся только тики таймера, входящие сообщения не логируются

БОЛЬНО

чинить ≈ 0.5 ч

Где: /home/alesha/gorilla/_logs/planner_bot.log

Почему так вышло: Логирование входящих не сделали. Когда что-то ломалось, смотреть было не на что — отсюда и диагностика «по памяти».

Что видит Тимур: Ничего. Но из-за этого поломки ловил ты, а не мы.

Пруф:

```
Всего строк в логе: 1348, из них подавляющее большинство —
"Running job tick ... executed successfully" (раз в минуту)
```

Твоих трёх сообщений (13:10, 13:12, 13:16) в логе НЕТ ни одной строки.

Б7. Живой бот работает не тем кодом, что лежит на диске

БОЛЬНО

чинить $\approx$ решение, не
часы

Где: процесс 693643 против файлов `planner_bot/*.py`

Почему так вышло: Ночная переделка правил файлы, не перезапуская бота. Сегодняшняя (остановленная по твоей команде) успела переписать их ещё раз и оборвалась на полпути.

Что видит Тимур: Пока ничего. Но если бот упадёт — сторож поднимет его уже на недописанном коде.

Пруф:

```
Процесс бота запущен: 09.08 04:18:21 (работает 9 ч)
bot.py на диске изменён: 09.08 13:30:57
agent.py (нейросеть, 622 строки) создан: 09.08 13:29:11 – в памяти его нет

Файлы синтаксически целы (проверено), но задача была оборвана на середине –
что дописано, а что нет, гарантировать нельзя.
```

Б8. Тесты гонялись прямо по боевой базе

БОЛЬНО

чинить $\approx$ 0.5 ч

Где: `planner_bot/test_bot.py:45, 57-63`

Почему так вышло: Отдельной тестовой базы для части прогонов не завели — подключались к рабочей, писали туда тестового пользователя и потом удаляли.

Что видит Тимур: Ничего не потерялось, но следы в твоей базе остались.

Пруф:

```
test_bot.py:45 con = sqlite3.connect(db.DB_PATH) # боевая база
test_bot.py:60-63
for t in ["users", "profile", "bookings", "tasks", "inbox", ...]:
    con.execute(f"DELETE FROM {t} WHERE tg_id=?", (UID,))

Следы в базе Тимура: счётчики sqlite_sequence tasks=19, snapshots=2
при нуле живых строк – то есть 19 задач и 2 снимка были созданы и удалены.
```

МЕЛОЧЬ

Что	Где	Пруф
Ошибки глотаются молча — сломанная операция моста просто исчезает	app.js, applyOps	try { applyOp(op) } catch (e) {}
Падение планировщика при старте движка глотаётся	engine/bot.py:1392	except Exception as _pe: sys.stderr.write(...)
Два бота одновременно — было живьём	лог, строка 77	telegram.error.Conflict: terminated by other getUpdates request
Обработчика ошибок у бота нет	лог, строка 49	No error handlers are registered, logging exception
Бот перезапускался 12 раз за ночь	лог	grep -c "starting long polling" → 12
Расписание в старом коде было на 21:30, в новом 21:00 — расхождение с каноном	лог строка 8 против bot.py:39	jobs scheduled: evening 21:30 / EVENING_DEFAULT = 21:00

БЕЗОПАСНОСТЬ

Статус находки, объявленной закрытой в отчёте 04:19: закрыта частично, по сути — нет.

Твои личные записи лежат в открытом интернете и отдаются любому, кто знает адрес. Никакой проверки, что это ты, нет — вместо пароля работает случайный кусок в адресе. В 04:08 закрыли только просмотр списка файлов в папке; сами файлы как отдавались, так и отдаются.

```
curl -I https://tim.r3ka.ru/planner/sync/zYfhYHw7B_PThVTSiTtX21KT.json
HTTP/2 200
content-type: application/json
content-length: 401

curl https://tim.r3ka.ru/planner/sync/zYfhYHw7B_PThVTSiTtX21KT.json
{"v":1,"ops":[{"id":3,"kind":"inbox_add","data":{"title":"Я все заполнил"...
{"id":4,... "title":"Че ты записал?"...
{"id":5,... "title":"С тобой можно общаться?"...

curl https://tim.r3ka.ru/planner/sync/ → заглушка «Здесь ничего нет»
(это и есть то, что закрыли в 04:08 — только листинг)
```

Насколько это страшно: адрес из 24 случайных символов не подберут перебором, и в логах сервера он не светится (проверено). Но ключ вечный, не меняется, уходит в адресную строку мини-приложения и живёт в истории Телеграма. Утечёт один раз — открыт навсегда.

Правильно чинить не заплаткой, а вместе с корневой причиной K2: когда данные переедут в нормальную базу за авторизацией, публичный файл обмена исчезнет сам. Отдельно — 1-2 часа.

КОРНЕВЫЕ ПРИЧИНЫ

20 находок растут из трёх решений. Чинить надо их, а не 20 багов.

1. Два хранилища вместо одного

Приложение хранит всё в облаке Телеграма, бот — в своей базе на сервере. Между ними — узкая односторонняя щель: файл обмена в одну сторону и ручная кнопка «отдать боту» в другую. Отсюда: пустой профиль, мёртвое расписание, разъезжающийся инбокс, «бот меня не знает».

Почему так вышло: домен tim.r3ka.ru стоит на чужом сервере, куда нет доступа, поэтому свой API под приложение поднимать не стали и выкрутились облаком Телеграма.

2. Окну дали роль мозга, не дав ему мозга

В мини-приложении нет ни базы, ни нейросети — а чат туда всё равно поставили. Пришлось изображать понимание самодельным разбором текста: «ищем день недели, остальное — название задачи». Это и есть тот чат, который превращал твои вопросы в карточки.

3. Приёмка проверяла функции в обход настоящего пути

Всё, что нельзя проверить снаружи Телеграма, объявлялось рабочим по результату прямого вызова функции. Поломка сидела ровно в том звене, которое ни один тест не трогал.

ПОЧЕМУ ОТЧЁТЫ ГОВОРИЛИ «ГОТОВО»

Шесть отчётов подряд заявляли, что всё работает. Вот механика — без оправданий.

Передачу данных «проверяли», минуя Телеграм

Тест сам собирал объект с данными и напрямую звал обработчик бота. Записалось в базу — значит «канал работает». А сломан был не обработчик, а доставка: Телеграм разрешает передачу только из нижней кнопки, и до обработчика ничего не долетало.

```
test_full.py:211
mw.web_app_data = type("W", (), {"data": json.dumps(snap)}]()
await B.on_webapp_data(upd, FakeCtx(fb))
check("задачи записались", s["tasks"] == 4) → OK
check("брони записались", s["bookings"] == 2) → OK
← «доказано» на данных, которые тест сам себе и подsunул
```

Мост «бот → окно» проверяли на файле, который тест сам написал

```
test_e2e.py:44-50
def write_bridge(ops):
with open(os.path.join(d, KEY + ".json"), "w") as f:
json.dump({"v": 1, "ops": ops}, f)
← потом тест открывает приложение и радуется, что данные подхватились
```

Приложение гоняли в обычном браузере, где Телеграма нет вовсе

Тесты поднимают локальный http-сервер и открывают приложение на 127.0.0.1. В таком режиме нет ни передачи данных, ни облачного хранилища — ровно тех двух механизмов, которые и сломаны. Проверялась только внешняя часть.

```
test_e2e.py:22 BASE = "http://127.0.0.1:8765/"
test_e2e.py:38 [sys.executable, "-m", "http.server", "8765", "-d", ROOT]
```

Проверки не были проверками

В пяти тестовых файлах на 1200 строк — один `assert`. Всё остальное — печать «ОК» или «ПРОВАЛ» в консоль. Прогон печатает провалы и спокойно идёт дальше, а в отчёт попадает строчка «прогон выполнен».

```
grep -c "assert" test_bot.py test_app.py test_e2e.py test_bookings.py test_full.py
test_bot.py:1 test_app.py:0 test_e2e.py:0 test_bookings.py:0 test_full.py:0
```

Ни одной проверки живого разговора

Ни разу не было прогона вида «написали боту в настоящем Телеграме — посмотрели, что он ответил, и что легло в базу». Поэтому первым это увидел ты.

Вывод

Проверялось «функция при правильном входе делает правильное». Не проверялось «вход вообще доезжает». Именно там и была поломка — во всех шести случаях.

ЧТО ДЕЛАТЬ

Этап	Что	Часы
0	Решить судьбу недописанного кода на диске: доделать или откатить к рабочему. Пока не решено — бота лучше не перезапускать.	0.5
1	Один источник правды. Поднять нормальный сервер данных, приложение читает и пишет туда. Развилка: либо API на нашем сервере с отдельным адресом, либо переносить мини-приложение к себе целиком. Это главный этап, без него остальное — латание.	12-16
2	Мозг в боте: нейросеть с инструментами, вопрос — ответ словами, задача — заведение с подтверждением. Начато (agent.py), надо доделать и проверить живьём.	6-10
3	Знакомство разговором в чате, а не анкетой в окне.	2-3
4	Руки в окне: перетаскивание пальцем, правка брони прямо с главного, удаление фальшивого чата.	6-9
5	Расписание: утро, вечер, воскресенье — и проверить на живом непустом профиле.	1-2
6	Новая приёмка: настоящий разговор в Телеграме, реальные числа из базы до и после, замер задержек.	2

Итого 30-42 часа честной работы.

Что предлагаю переписать с нуля, а не латать

Переписать	Почему
store.js + весь мост (bridge.py, файлы обмена, кнопка «отдать боту»)	Это протез вокруг отсутствующего сервера. Когда появится нормальная база — исчезает целиком. Латать его — вкладываться в то, что всё равно снесём.
Чат в окне (planSend)	Изображает понимание там, где его нет. Удалить, оставить кнопку «обсудить с ботом».
Тесты	Сейчас они создают ощущение проверки, не проверяя. Хуже, чем их отсутствие.

Что трогать не надо — сделано нормально

Визуал и все пять экранов, схема базы с миграциями, автозапуск с двойным сторожем (пережитие перезагрузки сервера проверено по конфигу), расшифровка голоса, идеология и тексты бота.

ПОДОЗРЕНИЯ — НЕ ПОДТВЕРЖДЕНО

Сюда честно вынесено то, что не доказано. В находки выше это не пошло.

Что	Почему не доказано
Почему у тебя инбокс был пустым при открытии окна	В моей живой проверке все три записи подтянулись нормально. Скорее всего ты смотрел на уже открытое окно (Telegram держит его живым и заново не загружает) — но на твоём телефоне я это проверить не могу.
Голосовые	Ключ у живого бота есть, код есть. Ни одного голосового ты не присылал — вживую расшифровка не гонялась.
Новый «умный» бот (agent.py)	Написан наполовину и остановлен по твоей команде «отложи коддинг». На шести фразах не гонял — врать про это не буду.
Полный список 17 проектов из ночного анализа GitHub	Со ссылками и звёздами не показывался. Висит непроверенным с 08.08.

ПРИЛОЖЕНИЕ: СЫРЫЕ ВЫВОДЫ

База бота, пользователь 697785207 (Тимур)

```
users 1 onboarded = 0
profile 0 ← знакомство не доехало
bookings 0 ← брони не доехали
tasks 0
snapshots 0 ← окно не отдавало данные НИ РАЗУ
dialog 0
sent 0 ← ни одного сообщения по расписанию
inbox 3 ← три твоих реплики, ставшие задачами
events 4 ops 3

SELECT * FROM inbox:
5 | 1 | Я все заполнил | open | 09.08 13:10:39
6 | 2 | Че ты записал? | open | 09.08 13:12:22
7 | 3 | С тобой можно общаться? | open | 09.08 13:16:49
```

Доставка

```
curl -I https://tim.r3ka.ru/planner/
HTTP/2 200 | nginx/1.24.0 | last-modified: 09.08 10:21:49 GMT

curl -I .../planner/sync/zYfhYHw7B_PThVTSiTTx21KT.json
HTTP/2 200 | application/json | 401 байт
Cache-Control отсутствует — кэш не при чём

getMe → ok=True, timplanner_bot, id 8738883023
```

Живая проверка задеплоенного приложения (браузер, 390x844, touch)

```
загрузка: успешно, ошибок в консоли нет
инбокс подтянулся: ["Я все заполнил", "Че ты записал?", "С тобой можно общаться?"]
брони: ["Сын", "Жена", "Отдых", "Зал", "Массаж"] ← насыпаны сами, без спроса
onboarded: false
запросов к /sync/ при старте: 1
запросов к /sync/ за следующие 15 сек: 0 ← опроса нет
плейсхолдер «Напиши, что на неделе»: присутствует ← фальшивый чат жив
обработчиков перетаскивания: 0
```

Автозапуск (единственное, что работает как обещано)

```
systemctl is-enabled alesha.service → enabled
Restart=always, User=alesha
engine/bot.py:1387 → запускает planner_bot/ensure.sh при старте
ensure.sh → supervise.sh (следит за ботом, раз в 10 сек)
→ guard.sh (следит за supervise.sh, раз в 15 сек)
Живые процессы: 691402 supervise, 691410 guard, 693643 bot
```

Оговорка: systemd --user не используется, linger выключен — всё держится на системной службе alesha.service.

Что было остановлено в начале аудита

Задача «пересборка взаимодействия» шла ЧЕТЫРЬМЯ параллельными процессами, переписывавшими одни и те же файлы. Остановлены все четыре. Движок перезапускал их автоматически — пришлось держать сторожа:

```
13:30:59 killed 748605 | 13:30:59 killed 748631 | 13:30:59 killed 748635
13:31:14 killed 749838 | 13:31:19 killed 749893 | 13:31:19 killed 749897
13:31:35 killed 749999
```

Файлы, которые они успели изменить (не откатывались):
planner_bot/agent.py создан 13:29:11 (31 065 байт, нейросеть)
planner_bot/bot.py изменён 13:30:57 (было 24 555 → стало 36 766)
planner_bot/db.py изменён 13:27:04
public/planner/app.js изменён 13:23:31 — УЖЕ ЗАДЕПЛОЕН

база: накатили схему v3 (добавлена таблица dialog), данные не тронуты

Аудит выполнен скиллом «техдир» R3КА в режиме read-only: код не правился, база только читалась, живой бот не перезапускался.