

Audion Hub Manager — история

[English](#) · [О программе](#) · [Руководство](#) · [Решения](#)

Содержание

- [1.10.0](#)
- [1.9.0](#)
- [1.8.0](#)
- [1.7.3](#)
- [1.7.2](#)
- [1.7.1](#)
- [1.7.0](#)
- [Ранние версии](#)

История нужна не ради счёта версий, а чтобы не чинить дважды одно и то же. Здесь записано, что менялось и почему.

1.10.0

Испорченный реестр больше не мешает запуску. Одна лишняя запятая в

`config/projects.json` роняла программу с трассировкой ещё до появления окна: реестр читался строгим разбором на этапе загрузки. Теперь чтение настроек при запуске откатывается к умолчанию, сохраняет уцелевшие записи и говорит о проблеме в окне вывода. Пустой или нечитаемый реестр даёт временный проект, указывающий на саму программу, — окно открывается, и настройки правятся изнутри.

План зеркалирования перепроверяется перед записью. Применение доверяло корням, записанным в файле плана, — устаревший или отредактированный план мог удалить по пути, пересекающемуся с исходником. Профиль и запрет пересечения проверяются заново, прежде чем что-либо будет записано или удалено.

Изменение поведения: ошибка копирования безусловно запрещает удаление. Прежде настройка `delete_after_successful_copy: false` позволяла удалять после сбоя — ровно в том случае, когда зеркало способно потерять файлы, существующие только в нём. Настройка осталась в конфигурации, но отключить эту защиту больше не может.

Команды через движок, а не через строку оболочки. Создание пакета и приём перемоткой собирали командную строку в окне, подставляя в неё путь файловой системы: имя папки с `&` или кавычкой ломало команду. Теперь обе идут списком доводов. Нужные функции движка существовали всё это время и просто не вызывались.

Опознание в панели задач. Имя, уже начинающееся с марки, больше не получает второй приставки: программа зовётся `Audion.Hub.Manager`, а не `Audion.Tools.Audion.Hub.Manager`. Windows группирует окна именно по этому имени.

Новый значок — белый трёхлучевой знак на почти чёрной плитке со скруглением, в размерах от 16 до 256 точек; ниже 32 точек упрощённый вариант, где три отдельных узла сливаются. Прежний значок сохранён.

Убраны два помощника обращения к серверу, которыми ничто не пользовалось.

1.9.0

Токен заработал сразу. Прежде каждая кнопка читала токен обратно из хранилища — и введённый в поле не делал ничего, пока не был сохранён. Получалось, что сохранение было условием работы, а не удобством. Теперь введённый токен используется немедленно, а «запомнить токен» делает ровно то, что обещает названием: помнит между запусками. Строка состояния называет, каким именно токеном воспользуются кнопки.

Забывание чистит оба места сразу — и поле, и запомненную копию в хранилище учётных данных.

1.8.0

Клонирование появилось в панели удалённых, где не было вовсе. Берёт адрес из поля, клонирует рядом с папкой активного проекта, отказывается клонировать поверх существующей папки и запускает `git` списком доводов, а не через оболочку.

Клонирование не виснет. Оно идёт с запретом на запрос пароля: запрос из окна всё равно некому увидеть, и вместо ожидания в пустоту программа сразу предупреждает, что Git читает хранилище, а не форму.

Панель удалённых помнит выбор — платформу и вид адреса между запусками, и открывается там, где её оставили, а не сбрасывается на GitHub.

1.7.3

Области действия токена следуют маршруту, а не объекту. Проверка на живом сервере Forgejo 15 показала: создание репозитория лежит под `/api/v1/user/*` и потому требует права на пользователя, а не на репозиторий. Git по HTTPS оценивается отдельно и требует права на репозиторий в любом случае — одним набором закрыть и обращения к серверу, и Git нельзя.

Записано, чем плох адрес с учётными данными: Git пишет такой токен открытым текстом в `.git/config` рабочей копии, где он переживает каждый резерв и каждое зеркало этой папки. Та же семья ошибок, что секрет в параметрах адреса, и полностью обходится через помощника учётных данных.

1.7.2

Закрыт путь внедрения команды. Разбор адреса сохраняет `;`, `&&` и `$(...)` внутри имени хоста, а это имя подставлялось в пробу `ssh -T`, которую окно запускает через оболочку. Файл `forgejo_hosts.json`, приехавший с клонированным проектом, мог таким образом выполнить команду на первой же пробе. Имя хоста, пользователь и порт проверяются там, где сходятся все пути; враждебные записи отбрасываются вместо загрузки; адреса со встроенными учётными данными отвергаются сразу.

Отвергаются адреса, заставляющие Git выполнить команду (`ext::`, `fd::`), адреса, начинающиеся с дефиса, — Git прочтёт их как ключ, — и адреса с управляющими символами. Проверка идёт и при сохранении из панели, и при применении файла настроек, чтобы конфигурация из чужого проекта не протащила такой адрес.

Предупреждение о простом HTTP перед отправкой токена: для сервера в домашней сети это законно, но токен пойдёт по проводу открытым.

1.7.1

Сервер, введённый в форму, больше не теряется. Запись настроек шла через функцию, которая выходила раньше времени, если учётная запись уже совпадала, — и хост не попадал в файл, так что при следующем запуске его приходилось вводить заново.

Разбор настроек стал устойчивым к ошибкам. Файл, поправленный руками с опечаткой, прежде ронял обработчик события: панель читает его при каждом переключении платформы и каждом нажатии клавиши в поле адреса.

Список репозиториев больше не обрезается молча. Перебор страниц останавливается на тысяче записей, и достижение потолка теперь называется вслух, а не выдаётся за полный список.

Негодный порт отвергается, а не подменяется тихо на двадцать второй — это строило правдоподобный адрес, который отказывал потом, при отправке.

Адрес со встроенным логином или токеном не сохраняется в `config/remotes.json` — это файл, попадающий в коммит, и секрет был бы опубликован следующей записью.

Удаление токена требует названной учётной записи: хранилища учётных данных различают записи по хосту и имени, поэтому пустое имя не совпадало ни с чем, сообщая при этом об успехе.

1.7.0

Поддержка своих серверов Forgejo и Gitea. Файл `config/forgejo_hosts.json` описывает сервера — адрес, пользователя и порт SSH, предпочтительный вид адреса, запомненную учётную запись — и не содержит секретов.

Принят обычный договор учётной записи: личный токен доступа, проверяемый запросом к серверу и передаваемый помощнику учётных данных. В настройки проекта не пишется ничего, а обычная отправка по HTTPS подхватывает тот же токен.

Вход через OAuth2 рассмотрен и отклонён: в Forgejo для него не реализованы области действия, такие токены несут административные права учётной записи, быстро истекают и сами по себе отpravку не обслуживают.

403 отделено от 401 в каждом обращении. Недостаточное право теперь сообщает: «токен верный, но создан без права, которое нужно этому вызову», — называет требуемое право и предлагает рабочий набор, вместо голого кода состояния рядом с токеном, который на самом деле в порядке.

Текст ошибки сервера затирается перед показом. Forgejo цитирует неизвестный токен обратно в сообщении — и отозванный или истёкший токен появился бы в окне вывода дословно, а оттуда в снимках экрана и скопированном тексте.

Записан случай 413 для серверов за прокси или туннелем с ограничением на размер запроса: упирается в него именно первая отправка истории, а обход — отправка по SSH, в том числе через проброс порта.

Ранние версии

0.2.x — планирование: договоры безопасности зеркала, разделение слоёв, профили фильтрации. **v0.1.0** — скелет приложения.

Из этого периода в программу перешли правила, изложенные в [Решениях](#): они собраны по следам настоящих поломок Disk Auditor, и повторять их заново не нужно.