



Руководство по установке сервера S-INFO

Версия ПО 4.2.0

2026

Оглавление

1. Установка серверной части S-INFO	5
2. Поддерживаемые платформы.....	6
3. Состав серверной части.....	7
4. Что требуется для работы	8
5. База данных, файлы и конфигурация.....	9
6. Резервное копирование	10
7. Обновление серверной части	11
8. Лицензия.....	13
9. Почтовые уведомления.....	14
10. Файлы клиента и автообновление.....	15
11. Проверка доступности сервера.....	16
12. Основной порядок установки.....	18
13. Обозначения в командах.....	19
14. Установка на Ubuntu	20
14.1. Проверка версии Ubuntu	21
14.2. Обновление списка пакетов	22
14.3. Обновление системы.....	23
14.4. Удаление неиспользуемых пакетов	24
14.5. Проверка необходимости перезагрузки.....	25
14.6. Установка базовых утилит.....	26
14.7. Проверка текущего времени.....	29
15. Локализация и ICU.....	30
16. Системные пакеты для работы серверной части.....	33
17. Основные пакеты.....	34
18. Дополнительные рекомендуемые пакеты	36
19. Проверка шрифтов и отчетов.....	39
19.1. Проверка Zstandard.....	40
19.2. Проверка PostgreSQL-утилит	41
19.3. Если пакет не найден	42
20. Установка и первичная настройка PostgreSQL.....	43
20.1. Установка PostgreSQL.....	44
20.2. Создание пользователя и базы данных	45
20.3. Проверка подключения к базе данных	47
20.4. Настройка параметров PostgreSQL для S-INFO.....	48
20.5. Где находятся основные файлы PostgreSQL.....	50
20.6. Если PostgreSQL установлен на отдельном сервере.....	51
20.7. Оптимизация PostgreSQL.....	52
20.8. Резервное копирование PostgreSQL.....	53
21. Размещение файлов серверной части.....	54
21.1. Создание папки	55
21.2. Копирование файлов.....	56

21.3. Настройка прав.....	57
22. Конфигурационные файлы backend.....	58
23. Настройка конфигурации backend.....	60
23.1. Настройка подключения к PostgreSQL.....	61
23.2. Настройка системных уведомлений.....	63
23.3. Настройка публичного адреса сервера.....	64
23.4. Настройка адреса веб-клиента.....	65
23.5. Настройка порта backend.....	66
23.6. Минимальный набор обязательных настроек.....	68
24. Дополнительные настройки и папка configs.....	69
24.1. SMTP.....	71
24.2. Очередь отправки email.....	72
24.3. Системные фоновые задачи.....	73
24.4. Общие правила работы с папкой configs.....	74
25. Включение резервного копирования файловой папки.....	75
26. Первый запуск backend и получение ключа для лицензии.....	77
27. Важные условия выпуска лицензии.....	79
28. Проверка журналов при первом запуске.....	80
29. Настройка systemd-службы.....	81
29.1. Важное условие по пользователю.....	82
29.2. Проверка прав на папку backend.....	83
29.3. Создание файла службы.....	84
29.4. Применение службы.....	86
29.5. Просмотр журналов службы.....	87
29.6. Проверка доступности.....	88
29.7. Управление службой.....	89
29.8. Если служба не запускается.....	90
30. Загрузка файлов S-INFO Desktop на сервер.....	91
30.1. Способ 1. Загрузка через команду ConsoleTools.....	93
30.1.1. Возможная ошибка libzstd.....	95
30.2. Способ 2. Загрузка через интерактивный режим ConsoleTools.....	96
30.3. Проверка после загрузки.....	98
30.4. Важно при обновлении backend.....	99
31. Ручное резервное копирование через ConsoleTools.....	100
32. Текущий этап установки.....	102
33. Настройка Nginx как обратного прокси.....	103
33.1. Что важно учесть.....	104
33.2. Установка Nginx.....	105
33.3. Подготовка HTTPS-сертификата.....	106
33.4. Создание конфигурации сайта.....	108
33.5. Если S-INFO WEB не используется.....	113
33.6. Включение сайта.....	114
33.7. Настройка appsettings.Production.json.....	115

33.8. Проверка с сервера.....	116
33.9. Проверка с другого компьютера	117
33.10. Проверка SignalR.....	118
33.11. Проверка журналов Nginx.....	119
33.12. Если используется межсетевой экран	120
33.13. Типовые проблемы.....	121
34. Установка S-INFO WEB.....	122
34.1. Загрузка файлов S-INFO WEB.....	123
34.2. Создание папки.....	124
34.3. Копирование файлов веб-клиента.....	125
34.4. Настройка прав для Nginx.....	126
34.5. Вариант для загрузки файлов через SFTP	127
34.6. Проверка доступа Nginx к файлам	128
34.7. Настройка config.json.....	129
34.8. Проверка через браузер.....	131
35. Финальный чек-лист установки	132
35.1. Проверка службы backend.....	133
35.2. Проверка локальной доступности backend	134
35.3. Проверка доступности через Nginx	135
35.4. Проверка S-INFO WEB	136
35.5. Проверка S-INFO Desktop.....	137
35.6. Проверка файлов клиента на сервере.....	138
35.7. Проверка SMTP	139
35.8. Проверка резервного копирования.....	140
35.9. Проверка журналов.....	141
35.10. Проверка прав доступа	142
35.11. Проверка режима работы	143
35.12. Итог	144
36. Обновление S-INFO.....	145
36.1. Важное условие по лицензии.....	146
36.2. Что скачать	147
36.3. Что обязательно сохранить.....	148
36.4. Общий порядок обновления backend.....	149
36.5. Обновление файлов S-INFO Desktop на сервере.....	151
36.6. Обновление S-INFO WEB.....	152
36.7. Что проверить после обновления.....	154
36.8. Если обновление не прошло	155
37. Частые проблемы и способы диагностики	156
37.1. Основные журналы Журналы backend	157
37.2. Журналы S-INFO Desktop	158
37.3. Журналы Nginx	159
37.4. Журналы PostgreSQL.....	160
38. Что отправлять в техническую поддержку	161

39. Backend не запускается	162
40. Ошибка лицензии или backend выводит новый ключ пользователя	164
41. Backend локально работает, но недоступен с других компьютеров.....	165
42. 502 Bad Gateway.....	166
43. 413 Request Entity Too Large	167
44. Долгие запросы обрываются	168
45. Не работают обновления данных в реальном времени	169
46. По /api/ не открывается health-статус.....	170
47. Backend видит HTTP вместо HTTPS	171
48. Backend работает не в режиме Production	172
49. Ошибка подключения к PostgreSQL.....	173
50. Ошибки прав PostgreSQL при миграциях.....	174
51. Ошибки на больших базах или при миграциях	175
52. При первом запуске видны ошибки __EFMigrationsHistory.....	176
53. Нет прав на папку Data.....	177
54. Не работает восстановление пароля или почтовые уведомления	178
55. Не отображается версия клиента или не работает автообновление S-INFO Desktop	179
56. Кнопка «Войти» в клиенте не активна	180
57. Ошибка Unable to load shared library 'libzstd'	181
58. Ошибки Globalization, CultureInfo, ICU или Invariant mode.....	182
59. Ошибки No usable version of libssl was found.....	183
60. Ошибки отчетов, PDF или шрифтов	184
61. Не открывается S-INFO WEB.....	186
62. Ошибки CORS в S-INFO WEB.....	188
63. Старый index.html, config.json или service worker после обновления S-INFO WEB	189
64. После обновления S-INFO WEB потерялись настройки	190
65. При обновлении backend потерялись настройки.....	191
66. Ошибки при резервном копировании базы данных.....	192
67. Не выполняются фоновые задачи	193
68. Недостаточно места на диске	194
69. Предупреждение HTTP/2 is not enabled for 127.0.0.1:5000.....	195
70. Предупреждение No XML encryptor configured.....	196
71. Особенности ALT Linux, Astra Linux и других Linux-дистрибутивов.....	197
72. Общий порядок диагностики.....	199
73. Тезаурус.....	201

1. Установка серверной части S-INFO



Руководство описывает установку серверной части S-INFO на сервер заказчика.



Основной рекомендуемый вариант установки — Linux-сервер. Для инфраструктуры на Windows также поддерживается установка на Windows Server.



В примерах установки для Linux используется Ubuntu LTS. На других Linux-дистрибутивах общий порядок установки сохраняется, но названия пакетов и расположение отдельных конфигурационных файлов могут отличаться.

2. Поддерживаемые платформы

Серверная часть поставляется отдельными сборками под Linux и Windows.

Операционная система	Архитектура	Рекомендуемый способ запуска
Linux	x64	systemd-служба
Windows Server	x64	IIS

Таблица 1



Для Linux рекомендуется Ubuntu LTS. Работа серверной части также проверена на российских Linux-дистрибутивах, включая Astra Linux и ALT Linux.



Для Windows рекомендуется использовать актуальную поддерживаемую версию Windows Server x64. В Windows-среде серверная часть размещается через IIS.

3. Состав серверной части



Серверная часть S-INFO — это приложение на ASP.NET Core 8 / .NET 8.

Сборка является самодостаточной: необходимые компоненты .NET включены в поставку приложения. В обычном сценарии на сервер не требуется отдельно устанавливать .NET Runtime или ASP.NET Core Runtime.



Исключение — установка на Windows через IIS. Для такого варианта требуется ASP.NET Core Hosting Bundle, так как он устанавливает модуль интеграции ASP.NET Core с IIS.



К серверной части подключаются настольное приложение S-INFO Desktop и веб-приложение S-INFO WEB. Без доступа к серверной части они не смогут работать.

4. Что требуется для работы

Для работы серверной части нужны:

Компонент	Назначение
S-INFO Backend	Серверное приложение
PostgreSQL 14 или новее	Основная база данных
Папка хранения файлов	Документы, модели, файлы проектов и другие пользовательские данные
Nginx	Публикация сервера по HTTP/HTTPS на Linux
IIS	Публикация сервера по HTTP/HTTPS на Windows

Таблица 2



Для базового запуска серверной части достаточно PostgreSQL, папки хранения файлов.

SMTP, расширенное логирование и фоновые задачи настраиваются отдельно при необходимости.



Для Windows-установки через IIS необходимо включить поддержку WebSocket, так как серверная часть использует SignalR.

5. База данных, файлы и конфигурация

S-INFO хранит данные в нескольких местах:

Элемент	Назначение
PostgreSQL	Основные данные системы
Data (папка) или другая папка из настроек	Загруженные файлы, документы, модели, файлы проектов и служебные данные
appsettings.Production.json	Основные настройки сервера
Configs (папка)	Дополнительные конфигурации серверной части
server_license.json	Файл лицензии
logs	Журналы работы серверной части

Таблица 3



Путь к папке хранения файлов задается в настройках приложения. По умолчанию используется папка Data.

Пользователь операционной системы, от имени которого работает серверная часть, должен иметь права на чтение, создание, изменение и удаление файлов в папке хранения. Если прав недостаточно, серверная часть не сможет корректно сохранять документы, модели и другие пользовательские файлы.



При обновлении серверной части папку хранения файлов нельзя удалять или перезаписывать.

6. Резервное копирование

Для полного резервного копирования сервера необходимо сохранять:

- базу данных PostgreSQL;
- папку хранения файлов;
- конфигурационные файлы приложения;
- файл лицензии;
- папку configs;
- журналы работы, если они нужны для диагностики.



Резервное копирование можно настроить средствами операционной системы, PostgreSQL, средствами виртуализации или внешней системой резервного копирования.



В S-INFO также есть встроенный механизм резервного копирования через фоновые задачи по расписанию. По умолчанию включается резервное копирование базы данных.



Резервное копирование файловой папки включается отдельно в конфигурации. Эта задача копирует папку хранения файлов в указанное место на диске и затем актуализирует копию по расписанию. Измененные файлы помещаются в отдельную папку-корзину. По умолчанию такие файлы хранятся 14 дней, срок хранения можно изменить в настройках.



Перед обновлением серверной части всегда необходимо сделать резервную копию базы данных и папки хранения файлов.

7. Обновление серверной части



Подробный порядок обновления описывается отдельно. В этом разделе указаны только основные правила.



Перед обновлением необходимо сделать резервную копию базы данных PostgreSQL. Серверная часть при запуске автоматически выполняет миграции базы данных, поэтому резервная копия нужна для отката при нештатной ситуации.

При замене файлов серверной части нельзя терять или перезаписывать:

- папку хранения файлов Data или другую папку, указанную в appsettings.Production.json;
- appsettings.Production.json;
- папку configs;
- server_license.json;
- локальные настройки, измененные администратором.



Новые исполняемые файлы и библиотеки приложения можно заменять файлами из новой поставки. Пользовательские данные, конфигурация и лицензия должны сохраняться.

Новые исполняемые файлы и библиотеки приложения заменяются файлами из новой поставки. Пользовательские данные, конфигурация и лицензия должны сохраняться.

Общий порядок обновления:

1. Остановить серверную часть S-INFO.
2. Сделать резервную копию базы данных, папки хранения файлов, конфигурации и лицензии.
3. Скопировать файлы новой поставки в папку серверной части с заменой исполняемых файлов и библиотек.

4. Проверить, что папка хранения файлов, `appsettings.Production.json`, папка `configs` и `server\_license.json` не были перезаписаны.
5. Запустить backend вручную и проверить старт приложения, подключение к базе данных и выполнение миграций.
6. Остановить ручной запуск backend.
7. Запустить серверную часть через службу операционной системы.
8. Проверить журналы работы и доступность сервера.
9. Загрузить новые файлы клиентского приложения на сервер через консольную утилиту.
10. Проверить автообновление клиента с другого компьютера и доступность сервера для пользователей.

8. Лицензия

Для запуска серверной части требуется файл лицензии:

`server_license.json`



Файл должен находиться в папке приложения рядом с исполняемым файлом серверной части.



Лицензия привязывается к серверу и пользователю операционной системы, от имени которого запускается сервер S-INFO (backend). Поэтому системного пользователя для запуска службы необходимо определить до получения лицензии.

Обычно порядок следующий:

1. Настроить сервер и папку приложения.
2. Запустить серверную часть без лицензии для получения данных, необходимых для выпуска лицензии.
3. Получить файл `server_license.json` через техническую поддержку.
4. Поместить файл лицензии в папку приложения.
5. Запустить серверную часть повторно.



При обновлении приложения файл лицензии необходимо сохранять.

9. Почтовые уведомления



Для полноценной работы системы рекомендуется настроить SMTP.



Почта используется для отправки уведомлений, служебных писем и восстановления пароля. Если SMTP не настроен, серверная часть может работать, но отправка писем и сценарии, завязанные на электронную почту, будут недоступны.



Для настройки необходимо подготовить почтовый ящик и указать параметры подключения в конфигурации SMTP.

10. Файлы клиента и автообновление



После установки серверной части необходимо загрузить файлы клиентского приложения на сервер.



Файлы клиента загружаются через отдельную консольную утилиту, входящую в поставку. Подробный порядок загрузки описывается в отдельном разделе инструкции.



Этот шаг важен для работы автообновления. Когда пользовательский клиент подключается к серверу, он получает обновления именно с этого сервера. Если файлы клиента не загружены, автообновление клиентского приложения работать не будет.

11. Проверка доступности сервера



После запуска серверной части можно проверить ее доступность через браузер.

Откройте адрес backend-сервера:

`https://<DOMAIN>/`



Если сервер опубликован через Nginx или IIS, необходимо открывать внешний адрес, который проксируется на backend.



При успешном запуске сервер вернет health-статус в формате JSON. Этот ответ показывает, что серверная часть запущена, подключилась к базе данных, загрузила конфигурацию и готова принимать запросы.

Пример ответа:

```
{
  "Name": "S-INFO Backend",
  "Status": "Healthy",
  "Mode": "Production",
  "BackendVersion": "4.1.1.0",
  "ClientVersion": "4.1.1.0",
  "WebApplication": "https://test.ru",
  "Timezone": "(UTC+03:00) Moscow Time",
  "Timestamp": "2026-04-29T08:25:43.9273691Z",
  "ActiveConnectionsCount": 2
}
```

Основные признаки корректной работы:

- "Status": "Healthy" — сервер работает штатно;
- "Mode": "Production" — сервер запущен в рабочем режиме;
- "BackendVersion" — версия серверной части;
- "ClientVersion" — версия клиентского приложения, загруженная на сервер;
- "License" — информация о примененной лицензии;
- "ActiveConnectionsCount" — количество активных подключений к серверу.

Такую проверку рекомендуется выполнить:

1. На самом сервере.
2. С другого компьютера внутри сети.
3. С внешнего адреса, если сервер должен быть доступен извне.



Если страница не открывается, необходимо проверить запуск службы, настройки Nginx или IIS, доступность порта, DNS-запись, SSL-сертификат и правила межсетевого экрана.

12. Основной порядок установки

Общий порядок установки:

1. Проверить версию операционной системы.
2. Обновить операционную систему и установленные пакеты.
3. Установить PostgreSQL.
4. Для Linux — установить и настроить Nginx.
5. Для Windows — установить и настроить IIS, включая поддержку WebSocket.
6. Создать базу данных и пользователя PostgreSQL.
7. Подготовить папку для серверной части S-INFO.
8. Настроить владельца и права доступа к папке приложения.
9. Скопировать файлы backend в папку приложения.
10. Настроить appsettings.Production.json и дополнительные конфигурационные файлы.
11. Запустить backend вручную для первичной проверки и получения данных для лицензии.
12. Получить и добавить файл server_license.json.
13. Повторно запустить backend вручную.
14. Проверить, что подключение к базе данных выполнено успешно и миграции прошли без ошибок.
15. Для Linux — зарегистрировать backend как systemd-службу.
16. Для Windows — завершить настройку приложения в IIS.
17. Настроить внешний доступ через Nginx или IIS.
18. Настроить SMTP при необходимости отправки почты.
19. Загрузить файлы клиента на сервер через консольную утилиту.
20. Проверить доступность сервера с клиентского компьютера.

13. Обозначения в командах

В командах используются условные значения:

Обозначение	Что заменить
<APP_DIR>	Папка установки серверной части
<APP_USER>	Пользователь операционной системы, от имени которого работает серверная часть
<DB_NAME>	Имя базы данных PostgreSQL
<DB_USER>	Пользователь PostgreSQL
<DB_PASSWORD>	Пароль пользователя PostgreSQL
<DOMAIN>	Доменное имя сервера
<BACKEND_PORT>	Внутренний порт серверной части
<DATA_DIR>	Папка хранения файлов

Таблица 4



Перед выполнением команд значения в угловых скобках необходимо заменить на значения, принятые в инфраструктуре заказчика.

14. Установка на Ubuntu



В этом разделе приведен пример установки серверной части S-INFO на Ubuntu 24.04 LTS.

Команды выполняются от имени пользователя с правами администратора. В примерах используется `sudo`.

14.1. Проверка версии Ubuntu

Перед установкой рекомендуется проверить версию операционной системы:

```
lsb_release -a
```

Дополнительно можно посмотреть сведения из системного файла:

```
cat /etc/os-release
```

Пример ожидаемого результата для Ubuntu 24.04 LTS:

```
Distributor ID: Ubuntu
```

```
Description:    Ubuntu 24.04.4 LTS
```

```
Release:        24.04
```

```
Codename:       noble
```

14.2. Обновление списка пакетов

Сначала необходимо обновить локальный список доступных пакетов:

```
sudo apt update
```



Эта команда не устанавливает обновления, а только получает актуальную информацию из подключенных репозиториях Ubuntu.

14.3. Обновление системы

После обновления списка пакетов установите актуальные версии системных пакетов:

```
sudo apt full-upgrade -y
```



`full-upgrade` обновляет установленные пакеты и при необходимости корректно обрабатывает изменения зависимостей.



На чистом сервере это рекомендуемый вариант подготовки системы перед установкой PostgreSQL, Nginx и серверной части S-INFO.

14.4. Удаление неиспользуемых пакетов

После обновления можно удалить пакеты, которые больше не требуются системе:

```
sudo apt autoremove -y
```

Затем очистить локальный кэш загруженных пакетов:

```
sudo apt clean
```

14.5. Проверка необходимости перезагрузки



После обновления ядра или системных библиотек Ubuntu может потребовать перезагрузку.

Проверьте это командой:

```
if [ -f /var/run/reboot-required ]; then cat /var/run/reboot-required;  
else echo "Reboot is not required"; fi
```

Если система сообщает, что перезагрузка требуется, выполните:

```
sudo reboot
```



После перезагрузки подключитесь к серверу повторно и продолжите установку.

14.6. Установка базовых утилит

Установите системные пакеты, необходимые для работы серверной части, отчетов, архивов, локализации и обслуживания базы данных:

```
sudo apt install -y \  
    ca-certificates \  
    curl \  
    wget \  
    gnupg \  
    lsb-release \  
    unzip \  
    zip \  
    p7zip-full \  
    gzip \  
    zstd \  
    rsync \  
    nano \  
    htop \  
    jq \  
    tzdata \  
    locales \  
    libicu74 \  
    libssl3 \  
    libc6 \  
    libstdc++6 \  
    libgcc-s1 \  
    fontconfig \  
    libfreetype6 \  
    libgdipplus \  
    fonts-dejavu-core \  

```

fonts-liberation \
fonts-noto-core \
postgresql-client

Назначение основных пакетов:

Пакет	Назначение
ca-certificates, curl, wget, gnupg	Работа с HTTPS, загрузка файлов и подключение внешних репозитория
unzip, zip, p7zip-full, gzip, zstd	Работа с архивами и сжатыми файлами
rsync	Копирование файлов и каталогов, в том числе для резервного копирования
nano, htop, jq	Редактирование конфигурации, просмотр нагрузки, работа с JSON
tzdata, locales, libicu74	Часовые пояса, локали и корректная работа локализации .NET
libssl3	Поддержка OpenSSL для защищенных соединений и криптографических операций
libc6, libstdc++6, libgcc-s1	Базовые runtime-библиотеки Linux, необходимые для native-компонентов
fontconfig, libfreetype6	Поиск, настройка и отрисовка шрифтов
fonts-dejavu-core, fonts-liberation, fonts-noto-core	Базовые шрифты для формирования отчетов и PDF
libgdiplus	Совместимость для компонентов, которые могут обращаться к System.Drawing
postgresql-client	Клиентские утилиты PostgreSQL, включая psql, pg_dump и pg_restore

Таблица 5

14.7. Проверка текущего времени



Для серверной части, лицензии, журналов и фоновых задач важно, чтобы на сервере было корректное время.

Проверьте текущую дату, время и часовой пояс:

```
timedatectl
```

Если часовой пояс указан неверно, его можно изменить. Например, для Москвы:

```
sudo timedatectl set-timezone Europe/Moscow
```

Для просмотра доступных часовых поясов:

```
timedatectl list-timezones
```



На этом базовая подготовка Ubuntu завершена.

15. Локализация и ICU



Серверная часть S-INFO использует механизмы локализации .NET для работы с датами, временем, числами, строками и региональными форматами.

На Linux для этого используется системная библиотека ICU и установленные в системе локали. В большинстве установок необходимые компоненты уже есть. Если система установлена в минимальном составе, при запуске backend может появиться ошибка, связанная с Globalization, CultureInfo, ICU или Invariant mode.

Для Ubuntu 24.04 установите необходимые пакеты:

```
sudo apt install -y libicu74 locales
```

Сгенерируйте UTF-8 локаль:

```
sudo locale-gen ru_RU.UTF-8
```

```
sudo locale-gen en_US.UTF-8
```

При необходимости назначьте системную локаль по умолчанию:

```
sudo update-locale LANG=ru_RU.UTF-8
```

Проверьте текущую локаль:

```
locale
```

Проверьте наличие ICU-библиотек:

```
ldconfig -p | grep icu
```



На Astra Linux названия пакетов обычно близки к Debian/Ubuntu, но могут отличаться в зависимости от версии системы и подключенных репозиториев.

Если пакет `libc74` недоступен, найдите доступный ICU-пакет:

```
apt-cache search libc74
```

Затем установите найденный runtime-пакет ICU и пакет локалей:

```
sudo apt install -y locales <ICU_PACKAGE>
```



Для ALT Linux используется `apt-get`.

Перед установкой обновите сведения о пакетах:

```
sudo apt-get update
```

Найдите доступный ICU-пакет:

```
apt-cache search libc74
```

Установите пакет локалей и найденный runtime-пакет ICU:

```
sudo apt-get install -y glibc-locales <ICU_PACKAGE>
```

После установки проверьте доступные локали:

```
locale -a | grep -Ei 'ru_RU|en_US'
```



После изменения локали рекомендуется перезапустить сервер или повторно подключиться к консоли.



Отключать локализацию .NET не рекомендуется. Это может привести к некорректной обработке дат, чисел, строк и региональных форматов.

В крайнем случае, если backend нужно временно запустить для диагностики, можно включить invariant-режим .NET:

```
export DOTNET_SYSTEM_GLOBALIZATION_INVARIANT=1  
  
./BackendWebService
```

Для постоянного запуска через systemd переменная добавляется в файл службы:

```
Environment=DOTNET_SYSTEM_GLOBALIZATION_INVARIANT=1
```



После устранения проблемы с ICU и локалями эту настройку необходимо убрать и перезапустить службу.

16. Системные пакеты для работы серверной части



Серверная часть S-INFO поставляется как самодостаточная сборка .NET 8. Отдельно устанавливать .NET Runtime или .NET SDK на сервер для запуска backend обычно не требуется.

При этом на Linux рекомендуется установить системные пакеты для локализации, отчетов, шрифтов, сжатия и обслуживания PostgreSQL. Часть пакетов уже может быть установлена в системе. Названия пакетов также могут отличаться в зависимости от версии Ubuntu или другого Linux-дистрибутива.

Если какой-либо пакет отсутствует в репозитории конкретной системы, это не означает, что backend обязательно не запустится. Список ниже подобран для более широкого покрытия типовых серверных установок и снижения риска ошибок при формировании отчетов, работе с локалями, архивами и резервным копированием.

17. Основные пакеты

Установите основной набор пакетов:

```
sudo apt install -y \  
    tzdata \  
    locales \  
    libicu74 \  
    zstd \  
    fontconfig \  
    libfontconfig1 \  
    libfreetype6 \  
    libharfbuzz0b \  
    libfribidi0 \  
    fonts-dejavu-core \  
    fonts-dejavu-extra \  
    fonts-liberation \  
    fonts-noto-core \  
    fonts-noto-extra \  
    postgresql-client
```

Назначение основных пакетов:

Пакеты	Для чего нужны
tzdata	Часовые пояса, корректное время в журналах и фоновых задачах
locales, libicu74	Локализация .NET: даты, числа, строки, региональные форматы
zstd	Работа с Zstandard-сжатием
fontconfig, libfontconfig1	Поиск и настройка шрифтов в Linux
libfreetype6	Отрисовка шрифтов
libharfbuzz0b, libfribidi0	Корректная обработка текста в отчетах
fonts-dejavu-*, fonts-liberation, fonts-noto-*	Базовые шрифты для отчетов, PDF, кириллицы и Unicode
postgresql-client	Утилиты psql, pg_dump, pg_restore для проверки PostgreSQL, резервного копирования и восстановления

Таблица 6



Эти пакеты особенно важны для работы локализации .NET, формирования отчетов через FastReport, генерации документов, работы со шрифтами и встроенных задач резервного копирования базы данных.

18. Дополнительные рекомендуемые пакеты

Для большей совместимости с библиотеками отчетов, графики, изображений и архивов рекомендуется установить дополнительный набор:

```
sudo apt install -y \  
    ca-certificates \  
    libssl3 \  
    libc6 \  
    libc6-dev \  
    libstdc++6 \  
    libgcc-s1 \  
    zlib1g \  
    gzip \  
    unzip \  
    zip \  
    p7zip-full \  
    libgdiplus \  
    libcairo2 \  
    libpango-1.0-0 \  
    libpangocairo-1.0-0 \  
    libgdk-pixbuf-2.0-0 \  
    libjpeg-turbo8 \  
    libpng16-16 \  
    libtiff6 \  
    libwebp7 \  
    libgif7 \  
    libx11-6 \  
    libx11-dev \  
    libxext6 \  
    libxrender1 \
```

libxrandr2 \
libxi6 \
libxfixed3 \
libxcursor1 \
libxinerama1 \
fonts-noto-cjk \
fonts-noto-color-emoji \
curl \
wget \
rsync \
jq \
nano \
htop

Назначение дополнительных пакетов:

Пакеты	Для чего нужны
ca-certificates, libssl3	HTTPS, сертификаты и защищенные соединения
libc6, libc6-dev, libstdc++6, libgcc-s1, zlib1g	Базовые системные библиотеки для native-компонентов
gzip, zip, unzip, p7zip-full	Работа с архивами
libgdipplus, libcairo2, libpango-*, libgdk-pixbuf-*	Совместимость с графическими и отчетными библиотеками
libjpeg-turbo8, libpng16-16, libtiff6, libwebp7, libgif7	Работа с распространенными форматами изображений
libx11-*, libxext6, libxrender1, libxrandr2, libxi6, libxfixed3, libxcursor1, libxinerama1	Совместимость со старыми графическими компонентами на Linux
fonts-noto-cjk, fonts-noto-color-emoji	Дополнительные шрифты Unicode, CJK и emoji
curl, wget, rsync, jq, nano, htop	Проверка доступности, копирование файлов, работа с JSON, редактирование конфигурации и диагностика сервера

Таблица 7

19. Проверка шрифтов и отчетов

Проверьте, что Fontconfig видит установленные шрифты:

```
fc-match "DejaVu Sans"
```

```
fc-match "Liberation Sans"
```

```
fc-match "Noto Sans"
```

Обновите кэш шрифтов:

```
sudo fc-cache -fv
```



Если в отчетах используются корпоративные шрифты, установите их на сервер отдельно.

После установки дополнительных шрифтов повторно обновите кэш:

```
sudo fc-cache -fv
```



На Linux отсутствие нужного шрифта может привести к замене шрифта другим системным шрифтом. Из-за этого PDF-отчеты могут визуально отличаться от отчетов, сформированных на Windows

19.1. Проверка Zstandard

Проверьте наличие Zstandard:

```
zstd --version
```

19.2. Проверка PostgreSQL-утилит

Проверьте наличие клиентских утилит PostgreSQL:

```
psql --version
```

```
pg_dump --version
```

```
pg_restore --version
```



Эти утилиты используются для ручной диагностики, резервного копирования и восстановления базы данных. Встроенные фоновые задачи резервного копирования базы данных также используют стандартные утилиты PostgreSQL.

19.3. Если пакет не найден

Если один из пакетов недоступен в репозитории, сначала обновите список пакетов:

```
sudo apt update
```

Затем найдите доступный аналог:

```
apt-cache search libicu
```

```
apt-cache search libwebp
```

```
apt-cache search libtiff
```

```
apt-cache search postgresql-client
```



Для других Linux-дистрибутивов используйте пакетный менеджер этой системы и устанавливайте ближайшие по назначению пакеты: ICU, локали, Fontconfig, FreeType, HarfBuzz, базовые шрифты, Zstandard и PostgreSQL client tools.

20. Установка и первичная настройка PostgreSQL



Серверная часть S-INFO использует PostgreSQL как основную базу данных.



Для S-INFO рекомендуется PostgreSQL версии 14 или новее. В Ubuntu 24.04 LTS из стандартных репозиториях устанавливается PostgreSQL 16, поэтому отдельный внешний репозиторий PostgreSQL для базовой установки не требуется. Если в инфраструктуре заказчика используется другая версия PostgreSQL, она должна быть не ниже версии 14.

20.1. Установка PostgreSQL

Установите PostgreSQL и дополнительные утилиты:

```
sudo apt install -y postgresql postgresql-contrib postgresql-client
```

Проверьте установленную версию:

```
psql --version
```

Проверьте состояние службы PostgreSQL:

```
sudo systemctl status postgresql
```

Если служба не запущена, запустите ее и добавьте в автозагрузку:

```
sudo systemctl enable --now postgresql
```

Проверьте готовность PostgreSQL принимать подключения:

```
pg_isready
```

Ожидаемый результат:

```
/var/run/postgresql:5432 - accepting connections
```

20.2. Создание пользователя и базы данных

Перейдите в консоль PostgreSQL от имени системного пользователя postgres:

```
sudo -u postgres psql
```

Создайте пользователя для серверной части S-INFO:

```
CREATE USER sinfo_user  
  
WITH PASSWORD '<DB_PASSWORD>'  
  
CREATEDB;
```

Создайте базу данных и назначьте владельцем созданного пользователя:

```
CREATE DATABASE sinfo_db  
  
OWNER sinfo_user  
  
ENCODING 'UTF8';
```

Выдайте пользователю права на подключение, создание временных объектов и создание объектов в базе данных:

```
GRANT CONNECT, TEMPORARY, CREATE ON DATABASE sinfo_db TO sinfo_user;
```

Подключитесь к созданной базе данных:

```
\c sinfo_db
```

Дополнительно выдайте права на стандартную схему public:

```
GRANT USAGE, CREATE ON SCHEMA public TO sinfo_user;
```

Завершите работу с консолью PostgreSQL:

```
\q
```

В командах выше:

Значение	Что заменить
sinfo_user	Имя пользователя PostgreSQL для S-INFO
sinfo_db	Имя базы данных S-INFO
<DB_PASSWORD>	Надежный пароль пользователя PostgreSQL

Таблица 8



Пароль необходимо сохранить в защищенном месте. Он будет использоваться в строке подключения серверной части S-INFO.

20.3. Проверка подключения к базе данных

Проверьте подключение к созданной базе данных:

```
psql "host=localhost port=5432 dbname=sinfo_db user=sinfo_user password=<DB_PASSWORD>"
```

После подключения выполните проверочный запрос:

```
SELECT current_database(), current_user;
```



Ожидаемый результат должен содержать имя базы данных `sinfo_db` и пользователя `sinfo_user`.

Выйдите из консоли PostgreSQL:

```
\q
```

20.4. Настройка параметров PostgreSQL для S-INFO



Для серверной части S-INFO рекомендуется увеличить параметры `max_connections` и `max_locks_per_transaction`.



`max_connections` задает максимальное количество одновременных подключений к PostgreSQL. По умолчанию это значение часто равно 100, но для серверной части S-INFO рекомендуется закладывать больший



запас. `max_locks_per_transaction` влияет на количество блокировок объектов, которые PostgreSQL может обслуживать в транзакциях; этот параметр особенно важен при миграциях, обслуживании базы и операциях, затрагивающих большое количество таблиц. PostgreSQL хранит обычные и advisory locks в общем пуле памяти, размер которого зависит от `max_locks_per_transaction` и `max_connections`, поэтому увеличивать эти значения нужно с учетом ресурсов сервера.

Для типовой установки S-INFO можно задать следующие значения:

```
sudo -u postgres psql -c "ALTER SYSTEM SET max_connections = '500';"
```

```
sudo -u postgres psql -c "ALTER SYSTEM SET max_locks_per_transaction = '5000';"
```

Перезапустите PostgreSQL:

```
sudo systemctl restart postgresql
```

Проверьте примененные значения:

```
sudo -u postgres psql -c "SHOW max_connections;"
```

```
sudo -u postgres psql -c "SHOW max_locks_per_transaction;"
```

Если PostgreSQL не запускается после изменения параметров, необходимо проверить журнал службы:

```
sudo journalctl -u postgresql -n 100 --no-pager
```

При необходимости параметры можно вернуть к значениям по умолчанию:

```
sudo -u postgres psql -c "ALTER SYSTEM RESET max_connections;"
```

```
sudo -u postgres psql -c "ALTER SYSTEM RESET max_locks_per_transaction;"
```

```
sudo systemctl restart postgresql
```

20.5. Где находятся основные файлы PostgreSQL

Для PostgreSQL 16 на Ubuntu 24.04 основные файлы обычно находятся здесь:

`/etc/postgresql/16/main/postgresql.conf`

`/etc/postgresql/16/main/pg_hba.conf`

`/var/lib/postgresql/16/main`



Если установлена другая версия PostgreSQL, номер версии в пути будет отличаться.

Посмотреть установленные кластеры PostgreSQL можно командой:

`pg_lsclusters`

20.6. Если PostgreSQL установлен на отдельном сервере



PostgreSQL может быть установлен на том же сервере, что и backend, или вынесен на отдельный сервер базы данных.

Если база данных находится на отдельном сервере, необходимо дополнительно настроить:

- сетевую доступность PostgreSQL;
- параметр `listen_addresses` в `postgresql.conf`;
- правила доступа в `pg_hba.conf`;
- правила межсетевого экрана;
- строку подключения в `appsettings.Production.json`.

Пример строки подключения для серверной части S-INFO:

```
{
  "Database": {
    "ConnectionString":
      "Server=<DB_HOST>;Port=5432;Database=sinfo_db;Username=sinfo_user;Password=<DB_PASSWORD>",
    "DataFolder": "Data",
    "DisableAutoSettingsDb": false
  }
}
```



Значение `Server` должно указывать на адрес PostgreSQL-сервера.

20.7. Оптимизация PostgreSQL



Настройка PostgreSQL под фактическую нагрузку — отдельная задача системного администратора.

Значения

`max_connections`

и

`max_locks_per_transaction`

задают только базовый запас для работы серверной части S-INFO. Для рабочей эксплуатации также рекомендуется отдельно оценить и настроить:

- объем оперативной памяти;
- параметры кэширования;
- параметры записи на диск;
- дисковую подсистему;
- autovacuum;
- журналы PostgreSQL;
- стратегию резервного копирования и восстановления;
- мониторинг нагрузки и долгих запросов.



Точные значения зависят от размера базы, количества пользователей, интенсивности загрузки файлов, числа фоновых задач и характеристик сервера.

20.8. Резервное копирование PostgreSQL



Резервное копирование базы данных необходимо настроить сразу после установки PostgreSQL.



Рекомендуется использовать независимый механизм резервного копирования: средства PostgreSQL, средства операционной системы, средства виртуализации, корпоративную систему резервного копирования или другой принятый в организации инструмент.



В S-INFO также есть встроенная фоновая задача резервного копирования базы данных по расписанию. Она будет описана отдельно. Несмотря на это, для рабочей эксплуатации рекомендуется иметь дополнительную независимую резервную копию базы данных, чтобы восстановление не зависело только от одного механизма.

Минимальная ручная проверка резервного копирования через

`pg_dump:`

```
pg_dump \  
  --host=localhost \  
  --port=5432 \  
  --username=sinfo_user \  
  --format=custom \  
  --file=/tmp/sinfo_db_test.backup \  
sinfo_db
```

После проверки тестовый файл можно удалить:

```
rm /tmp/sinfo_db_test.backup
```

21. Размещение файлов серверной части



После подготовки PostgreSQL необходимо разместить файлы серверной части S-INFO на сервере.

В примерах ниже используется папка:

`/home/sinfo-admin/backend`



Это только пример. В рабочей установке можно использовать другой путь, принятый в инфраструктуре заказчика.

21.1. Создание папки

Создайте папку для backend:

```
mkdir -p /home/sinfo-admin/backend
```

21.2. Копирование файлов



Скопируйте в эту папку файлы поставки backend для Linux.



В папке должны находиться исполняемый файл BackendWebService, конфигурационные файлы, папка configs с базовыми настройками и остальные файлы поставки.

Пример копирования из временной папки:

```
cp -r /tmp/BackendWebService/* /home/sinfo-admin/backend/
```



Если файлы передаются архивом, сначала распакуйте архив, затем скопируйте содержимое папки сборки в /home/sinfo-admin/backend.

21.3. Настройка прав

Назначьте владельцем папки пользователя, от имени которого будет запускаться backend:

```
sudo chown -R sinfo-admin:sinfo-admin /home/sinfo-admin/backend
```

Выдайте право на выполнение исполняемого файла:

```
chmod +x /home/sinfo-admin/backend/BackendWebService
```

Выдайте право на выполнение исполняемого файла консольной утилиты:

```
chmod +x /home/sinfo-admin/backend/ConsoleTools/ConsoleTools
```



После этого пользователь `sinfo-admin` сможет запускать backend и записывать данные в рабочие папки приложения.



Папки `Data` и `logs` создавать вручную не требуется. Серверная часть создаст их при запуске, если они отсутствуют. Папка `configs` уже входит в поставку с базовыми конфигурационными файлами.

22. Конфигурационные файлы backend

В папке backend находятся основные конфигурационные файлы:

appsettings.json

appsettings.Development.json

appsettings.Production.json



Серверная часть читает базовый файл appsettings.json, а затем файл для выбранного режима запуска.

В рабочей установке backend запускается в режиме

Production,

поэтому основной файл для настройки сервера:

appsettings.Production.json



Именно в нем обычно настраиваются подключение к PostgreSQL, папка хранения файлов, публичный адрес сервера, почтовые уведомления, логирование и другие рабочие параметры.

Файл:

appsettings.Development.json



Используется только при запуске backend в режиме Development. В этом режиме дополнительно доступен Swagger по адресу backend-сервера. Для обычной установки на сервере заказчика этот режим не используется.

Файл:

appsettings.json



Используется как базовая конфигурация. В поставке S-INFO он обычно остается пустым или содержит минимальные общие настройки. Рабочие параметры принято указывать в `appsettings.Production.json` или, при разработке, в `appsettings.Development.json`.

Для установки на сервере заказчика изменяйте:

`appsettings.Production.json`

23. Настройка конфигурации backend

Основные настройки серверной части находятся в файле:

```
appsettings.Production.json
```

Перейдите в папку backend:

```
cd /home/sinfo-admin/backend
```

Откройте файл настроек:

```
nano appsettings.Production.json
```

Перед изменением конфигурации рекомендуется сохранить копию исходного файла:

```
cp appsettings.Production.json appsettings.Production.json.bak
```

23.1. Настройка подключения к PostgreSQL

В первую очередь необходимо настроить строку подключения к базе данных:

```
"Database": {  
    "ConnectionString":  
    "Server=localhost;Database=sinfo_db;Username=sinfo_user;Password=<DB_PASSWORD>",  
    "DataFolder": "Data"  
}
```

Замените:

Параметр	Значение
Server	Адрес PostgreSQL. Если база установлена на этом же сервере, оставьте localhost
Database	Имя базы данных S-INFO
Username	Пользователь PostgreSQL
Password	Пароль пользователя PostgreSQL
DataFolder	Папка хранения файлов backend

Таблица 9

Обычно

DataFolder

оставляют со значением по умолчанию:

```
"DataFolder": "Data"
```



В этом случае папка Data будет создана рядом с backend при первом запуске. Если в инфраструктуре заказчика принято хранить пользовательские файлы отдельно, можно указать другой путь, например отдельный диск или каталог.



Пользователь операционной системы, от имени которого запускается backend, должен иметь права на чтение, создание, изменение и удаление файлов в этой папке.

23.2. Настройка системных уведомлений

В блоке

ServerSettings

рекомендуется указать почту администратора:

```
"AlertEmails": [  
    "admin@example.ru"  
]
```



Если SMTP настроен, backend будет отправлять на этот адрес системные уведомления, например о нехватке свободного места или других проблемах на сервере

SMTP настраивается отдельно в файле:

configs/mailSmtp.settings.json



Если SMTP не настроен, серверная часть будет работать, но отправка писем и почтовых уведомлений будет недоступна.

23.3. Настройка публичного адреса сервера

Также рекомендуется указать публичный адрес backend-сервера:

```
"PublicServerUrl": "https://<DOMAIN>"
```



Этот адрес используется в письмах, ссылках и сценариях, где backend должен знать собственный внешний адрес.

Пример:

```
"PublicServerUrl": "https://api.example.ru"
```



Если сервер публикуется через Nginx или IIS, указывайте именно внешний адрес, по которому к backend подключаются пользователи.

23.4. Настройка адреса веб-клиента

Если приобретен и используется модуль S-INFO WEB, необходимо указать адрес веб-приложения:

```
"Clients": {  
  "WebApplication": {  
    "Url": "https://<WEB_DOMAIN>"  
  }  
}
```



Этот параметр не запускает веб-клиент автоматически и не указывает путь к его файлам.



S-INFO WEB — это отдельное веб-приложение, которое устанавливается и публикуется отдельно. В этом параметре указывается внешний адрес уже опубликованного веб-клиента.

Адрес используется:

- для разрешения запросов от веб-клиента к backend;
- для CORS;
- для ссылок в письмах и служебных сценариях.



Если веб-клиент не используется, значение можно оставить null.

23.5. Настройка порта backend



Настройка endpoint является необязательной.



По умолчанию backend может запускаться на локальном HTTP-порту 5000. Для установки через Nginx это нормальный вариант: backend работает локально, а внешний доступ настраивается через reverse proxy.

Если нужно явно указать адрес и порт, можно добавить или раскомментировать блок Kestrel:

```
"Kestrel": {  
  "Endpoints": {  
    "Http": {  
      "Url": "http://localhost:5000"  
    }  
  }  
},  
"AllowedHosts": "*"
```



Kestrel — это системный блок настроек ASP.NET Core. Через него можно настраивать адреса прослушивания, порты, HTTPS, протоколы и другие параметры веб-сервера Kestrel.

Полный список доступных параметров и вариантов настройки зависит от версии ASP.NET Core. При необходимости используйте официальную документацию Microsoft по настройке Kestrel и конфигурации ASP.NET Core.

Для стандартной установки на Linux рекомендуется оставлять backend на локальном адресе и не раскомментировать этот блок настроек (Kestrel)

Внешний HTTPS-доступ будет настроен позже через Nginx.

23.6. Минимальный набор обязательных настроек

Для первого запуска backend обычно достаточно настроить:

```
"Database": {  
    "ConnectionString":  
    "Server=localhost;Database=sinfo_db;Username=sinfo_user;Password=<DB_PASSWORD>",  
    "DataFolder": "Data"  
}
```

Рекомендуется также сразу заполнить:

```
"AlertEmails": [  
    "admin@example.ru"  
],  
"PublicServerUrl": "https://<DOMAIN>"
```

Если используется S-INFO WEB, дополнительно укажите:

```
"Clients": {  
    "WebApplication": {  
        "Url": "https://<WEB_DOMAIN>"  
    }  
}
```

24. Дополнительные настройки и папка `configs`

Часть настроек `backend` вынесена из `appsettings.Production.json` в отдельную папку:

`configs`



Папка `configs` находится рядом с исполняемым файлом `backend`. В ней хранятся дополнительные конфигурации серверной части: SMTP, очередь отправки писем, расписание фоновых задач, настройки интеграций и служебные параметры безопасности.

В текущей поставке используются следующие файлы:

Файл	Назначение
configs/mailSmtplib.settings.json	Настройки SMTP-сервера для отправки писем
configs/emailQueue.settings.json	Настройки очереди отправки email-сообщений
configs/systemScheduler.settings.json	Настройки системных фоновых задач
configs/security.auth.json	Служебный файл авторизации и подписи токенов
configs/integrations/integration.secrets.json	Секреты и токены для движка интеграций
configs/integrations/example_property_update_job.json	Пример конфигурации интеграционной задачи

Таблица 10



security.auth.json создается автоматически при первом запуске, если его еще нет

24.1. SMTP

Файл:

configs/mailSmtп.settings.json



используется для настройки отправки писем.

В нем указываются адрес SMTP-сервера, порт, логин, email отправителя, пароль, режим SSL/TLS и таймаут подключения.

Примерный состав настроек:

```
{  
  "Host": "smtp.example.ru",  
  "Port": 587,  
  "UseSSL": true,  
  "Login": "user@example.ru",  
  "Email": "user@example.ru",  
  "Password": "<SMTP_PASSWORD>",  
  "SslMode": "StartTlsWhenAvailable",  
  "Timeout": 30000  
}
```



Если SMTP не настроен, backend может запускаться и работать, но отправка почтовых уведомлений, писем восстановления пароля и других email-сообщений будет недоступна.

24.2. Очередь отправки email

Файл:

`configs/emailQueue.settings.json`



управляет ограничениями и повторными попытками при отправке писем.

В нем можно настроить количество писем в секунду и минуту, задержки между повторными попытками, максимальное число повторов и поведение при последовательных ошибках отправки.

Обычно эти настройки можно оставить по умолчанию. Их имеет смысл менять, если почтовый сервер имеет строгие ограничения на частоту отправки или если требуется более аккуратная обработка временных ошибок SMTP.

24.3. Системные фоновые задачи

Файл:

`configs/systemScheduler.settings.json`



используется для настройки системных фоновых задач.

В этом файле задаются расписания, включение или отключение задач, уведомления, параметры конкретных заданий. В конфигурации описаны ночные системные задачи: резервное копирование базы данных, архивация журналов, очистка служебных данных и обслуживание базы данных.

К основным задачам относятся:

- резервное копирование базы данных;
- резервное копирование папки Data;
- архивация журналов;
- очистка временных и устаревших данных;
- обслуживание базы данных;
- мониторинг свободного места и системных проблем;
- выполнение интеграционных задач.



Расписание выполняется по времени сервера. При необходимости администратор может изменить расписание задач или отключить отдельные задачи в конфигурации. Во время тяжелых задач, например резервного копирования или обслуживания базы данных, сервер может временно работать медленнее.

24.4. Общие правила работы с папкой configs



При базовой установке обычно достаточно настроить SMTP при необходимости и оставить остальные файлы с настройками по умолчанию.

При обновлении backend папку configs нельзя бездумно перезаписывать файлами из новой поставки. В ней могут находиться локальные настройки заказчика, секреты интеграций и служебные файлы, созданные после первого запуска.

Рекомендуется:

- сохранять резервную копию папки configs перед обновлением;
- не публиковать содержимое файлов с паролями, токенами и ключами;
- ограничить права доступа к папке configs;
- после изменения настроек перезапустить backend;
- при появлении новых конфигурационных файлов в будущих версиях использовать комментарии внутри файлов и документацию к конкретной версии поставки

25. Включение резервного копирования файловой папки



По умолчанию в S-INFO включено резервное копирование базы данных. При необходимости можно дополнительно включить резервное копирование файловой папки backend, то есть папки Data или другой папки, указанной в параметре Database:DataFolder.

Для этого откройте конфигурацию системных фоновых задач:

```
cd /home/sinfo-admin/backend/configs
```

```
nano systemScheduler.settings.json
```

В массиве

JobSchedules

найдите и раскомментируйте блок задачи:

```
{
  "DisplayName": "Резервное копирование файлов сервера",
  "JobType": "DataFolderBackupJob",
  "Enabled": true,
  "EnableHistory": true,
  "EnableAuditLog": true,
  "NotificationStatuses": [ "Failed", "Warning" ],
  "NotificationTargets": [ "AlertEmails" ],
  "CronExpression": "0 40 1 * * ?",
  "Settings": {
    // "BackupFolderPath": "data_files_backups",
    // "TrashRetentionDays": 14
  }
}
```



При такой настройке backend будет ежедневно запускать резервное копирование файловой папки в 01:40 по времени сервера.

По умолчанию путь для резервной копии будет выбран из настроек задачи. Если нужно указать свою папку, раскомментируйте и измените параметр:

`"BackupFolderPath": "data_files_backups"`



Путь может быть относительным от папки backend или абсолютным.

Параметр:

`"TrashRetentionDays": 14`



задает срок хранения измененных и удаленных файлов в папке-корзине. По умолчанию используется 14 дней.

Резервное копирование файловой папки работает как актуализируемая копия: backend поддерживает копию папки в заданном месте, а измененные или удаленные файлы переносит в отдельную папку-корзину.

После изменения `systemScheduler.settings.json` сохраните файл и перезапустите backend.

26. Первый запуск backend и получение ключа для лицензии



После настройки `appsettings.Production.json` необходимо выполнить первый ручной запуск backend.



Первый запуск нужен для проверки конфигурации, создания служебных файлов и получения ключа пользователя, по которому выпускается файл лицензии.

Перейдите в папку backend:

```
cd /home/sinfo-admin/backend
```

Запустите backend вручную:

```
./BackendWebService
```

При первом запуске backend:

- проверит конфигурацию;
- создаст служебные папки, если они отсутствуют;
- создаст файл `configs/security.auth.json`, если его еще нет;
- выведет информацию о сервере;
- сформирует ключ пользователя для выпуска лицензии.



Если файл лицензии еще не добавлен, запуск завершится ошибкой лицензии. На этом этапе это ожидаемое поведение.

В выводе необходимо найти строку вида:

Machine code:

"f3a8d21c9e6b74a0c5f8923d14e7b6a9c0d4e8f1a2b3c5d6e7f8091a2b3c4d5e6f708192a3b4c5d6e7f8091a2b3c

Также ключ будет продублирован ниже в сообщении об активации:

Отправьте ключ пользователя:

f3a8d21c9e6b74a0c5f8923d14e7b6a9c0d4e8f1a2b3c5d6e7f8091a2b3c4d5e6f708192a3b4c5d6e7f8091a2b3c4

на почту support@sinfo.tech.

Скопируйте ключ пользователя полностью и отправьте его на почту:

support@sinfo.tech

В письме укажите:

- ключ пользователя;
- название организации;
- доменное имя или адрес сервера, если уже известно;
- контактное лицо для связи.

После обработки запроса будет выдан файл лицензии:

```
server_license.json
```

Файл лицензии необходимо поместить в корневую папку backend:

```
/home/sinfo-admin/backend/server_license.json
```

27. Важные условия выпуска лицензии



Ключ пользователя необходимо копировать полностью. Частая ошибка — скопировать только часть строки из терминала.

Лицензия привязывается к серверу и пользователю операционной системы, от имени которого был запущен backend.

Если ключ был получен при запуске от имени пользователя:

`sinfo-admin`



то в дальнейшем backend-служба должна запускаться от имени этого же пользователя.

Если получить ключ от одного пользователя, а затем настроить службу на запуск от другого пользователя, лицензия может не подойти.

Также не рекомендуется получать лицензию до окончательного выбора сервера, так как лицензия привязывается к окружению конкретной машины.

28. Проверка журналов при первом запуске

Если вывод в терминале недоступен или был закрыт, backend записывает журналы в папку:

```
/home/sinfo-admin/backend/logs
```

Пример файла журнала:

```
/home/sinfo-admin/backend/logs/sinfo_log_20260429.log
```

Посмотреть последние строки журнала можно командой:

```
tail -n 100 /home/sinfo-admin/backend/logs/sinfo_log_*.log
```

Найти ключ пользователя в журналах можно так:

```
grep -R "Machine code" /home/sinfo-admin/backend/logs
```

или так:

```
grep -R "Отправьте ключ пользователя" /home/sinfo-admin/backend/logs
```



Если файл журнала появился не сразу, подождите несколько секунд и повторите проверку. Логирование может записывать данные на диск с небольшой задержкой.

Если журналы не создаются, проверьте права доступа к папке backend:

```
ls -la /home/sinfo-admin/backend
```



Пользователь, от имени которого запускается backend, должен иметь права на запись в папку backend и создаваемые внутри нее папки logs, Data и configs.

29. Настройка systemd-службы



После успешного ручного запуска backend необходимо настроить службу systemd. Это позволит запускать серверную часть автоматически после перезагрузки сервера и перезапускать ее при нештатном завершении

Перед настройкой службы остановите ручной запуск backend, если он еще выполняется:

Ctrl + C

29.1. Важное условие по пользователю



Служба должна запускаться от имени того же пользователя, под которым был получен ключ для лицензии.

Если ключ лицензии был получен при запуске от имени пользователя:

`sinfo-admin`

то и в файле службы необходимо указать:

`User=sinfo-admin`

`Group=sinfo-admin`



Если указать другого пользователя, лицензия может не подойти.

29.2. Проверка прав на папку backend

Проверьте владельца папки backend:

```
ls -la /home/sinfo-admin/backend
```

При необходимости назначьте владельца:

```
sudo chown -R sinfo-admin:sinfo-admin /home/sinfo-admin/backend
```

Проверьте право на выполнение backend:

```
chmod +x /home/sinfo-admin/backend/BackendWebService
```

29.3. Создание файла службы

Создайте файл службы:

```
sudo nano /etc/systemd/system/BackendWebService.service
```

Добавьте конфигурацию:

[Unit]

Description=S-INFO Backend

After=network-online.target postgresql.service

Wants=network-online.target

[Service]

User=sinfo-admin

Group=sinfo-admin

WorkingDirectory=/home/sinfo-admin/backend

ExecStart=/home/sinfo-admin/backend/BackendWebService

Environment=ASPNETCORE_ENVIRONMENT=Production

Environment=DOTNET_ENVIRONMENT=Production

Environment=DOTNET_EnableDiagnostics=0

Environment=DOTNET_GCServer=1

Restart=always

RestartSec=5

TimeoutStopSec=30

KillSignal=SIGINT

LimitNOFILE=200000

StandardOutput=journal

StandardError=journal

[Install]

WantedBy=multi-user.target

Если PostgreSQL установлен на отдельном сервере, зависимость от локальной службы PostgreSQL можно убрать:

After=network-online.target

Wants=network-online.target

Если папка backend находится в другом месте, замените пути:

WorkingDirectory=/home/sinfo-admin/backend

ExecStart=/home/sinfo-admin/backend/BackendWebService



На фактический путь установки.

29.4. Применение службы

Перечитайте конфигурацию

systemd:

```
sudo systemctl daemon-reload
```

Добавьте backend в автозагрузку и запустите службу:

```
sudo systemctl enable --now BackendWebService.service
```

Проверьте состояние службы:

```
sudo systemctl status BackendWebService.service
```

Если служба запустилась корректно, в статусе должно быть:

```
active (running)
```

29.5. Просмотр журналов службы

Посмотреть последние строки журнала:

```
sudo journalctl -u BackendWebService.service -n 100 --no-pager
```

Смотреть журнал в реальном времени:

```
sudo journalctl -u BackendWebService.service -f
```

В журнале нужно дождаться строки:

```
Server is listening on "http://localhost:5000"
```



Это означает, что backend успешно запущен через службу.

29.6. Проверка доступности

Проверьте локальный ответ backend:

```
curl http://localhost:5000
```



Если backend работает корректно, сервер вернет JSON со статусом.

29.7. Управление службой

Остановить backend:

```
sudo systemctl stop BackendWebService.service
```

Запустить backend:

```
sudo systemctl start BackendWebService.service
```

Перезапустить backend:

```
sudo systemctl restart BackendWebService.service
```

Проверить, включена ли автозагрузка:

```
systemctl is-enabled BackendWebService.service
```

Отключить автозагрузку:

```
sudo systemctl disable BackendWebService.service
```

29.8. Если служба не запускается

Если служба не перешла в состояние

`active (running)`,

проверьте журнал

:

```
sudo journalctl -u BackendWebService.service -n 200 --no-pager
```

Чаще всего проблемы связаны с одним из пунктов:

- указан неправильный путь в `WorkingDirectory` или `ExecStart`;
- у файла `BackendWebService` нет права на выполнение;
- служба запускается не от того пользователя, под которого выпущена лицензия;
- у пользователя службы нет прав на папки `Data`, `logs` или `configs`;
- неверная строка подключения к PostgreSQL;
- PostgreSQL недоступен;
- отсутствует или не подходит файл `server_license.json`;
- ошибка в `appsettings.Production.json`.

30. Загрузка файлов S-INFO Desktop на сервер



После установки backend необходимо загрузить на сервер файлы настольного клиента S-INFO Desktop. Эти файлы используются для автообновления клиентского приложения у пользователей.



Файлы клиента должны быть той же версии, что и backend.

Скачайте архив:

S-INFO Desktop (unpacked)

из центра загрузок:

<https://download.sinfo.tech/all>



Распакуйте архив на сервере в отдельную временную папку.

В примере используется путь:

/home/sinfo-admin/temp_client



Это только пример. Можно использовать другую временную папку.

После распаковки исполняемый файл настольного клиента должен находиться по пути:

/home/sinfo-admin/temp_client/S_Info.exe

Файл:

S_Info.exe



Является исполняемым файлом настольного клиента S-INFO Desktop.



Перед загрузкой файлов клиента рекомендуется убедиться, что backend уже запускался с лицензией, база данных создана, миграции выполнены, а папка Data доступна для записи.

30.1. Способ 1. Загрузка через команду ConsoleTools

Перейдите в папку консольной утилиты:

```
cd /home/sinfo-admin/backend/ConsoleTools
```

При необходимости посмотрите справку:

```
./ConsoleTools run --help
```

Запустите загрузку файлов клиента:

```
./ConsoleTools run upgrade /home/sinfo-admin/temp_client
```

Где:

```
/home/sinfo-admin/temp_client
```



— путь к распакованной папке клиента.



В процессе выполнения утилита использует настройки backend, подключается к базе данных и загружает файлы клиента в серверное хранилище.

Пример ключевых строк успешного запуска:

Base setup

use: /home/sinfo-admin/backend/ConsoleTools/../../appsettings.Production.json

Run core...

Backend core started...

Start upgrade

Release path: /home/sinfo-admin/temp_client

Wait...



После успешного завершения файлы клиента будут загружены на сервер.

30.1.1. Возможная ошибка libzstd

Если при загрузке появляется ошибка вида:

Unable to load shared library 'libzstd'

или:

libzstd.so: cannot open shared object file: No such file or directory



Значит в системе отсутствует библиотека Zstandard, необходимая для сжатия файлов клиента.

Для Ubuntu установите пакеты:

```
sudo apt install -y libzstd1 libzstd-dev zstd
```

После установки повторите загрузку:

```
./ConsoleTools run upgrade /home/sinfo-admin/temp_client
```



Для других Linux-дистрибутивов необходимо установить аналогичные пакеты Zstandard через пакетный менеджер используемой системы.

30.2. Способ 2. Загрузка через интерактивный режим ConsoleTools

Перейдите в папку консольной утилиты:

```
cd /home/sinfo-admin/backend/ConsoleTools
```

Запустите ConsoleTools:

```
./ConsoleTools
```



На первом шаге выберите конфигурацию, через которую утилита подключится к серверным данным.

Для рабочей установки обычно выбирается:

```
[Auto] appsettings.Production.json (direct connection)
```



Далее введите логин и пароль администратора S-INFO.

По умолчанию при первичной установке создается учетная запись:

Логин: admin

Пароль: admin



После завершения настройки сервера пароль администратора необходимо изменить.

После входа выберите:

```
Client menu (desktop client)
```

Затем выберите пункт:

Upload client update to server

Укажите путь к распакованной папке клиента:

/home/sinfo-admin/temp_client



После этого ConsoleTools выполнит загрузку файлов клиента на сервер. В выводе будет отображаться аналогичный журнал выполнения, как при запуске через команду `run upgrade`.

30.3. Проверка после загрузки

После загрузки файлов клиента проверьте health-статус backend:

```
curl http://localhost:5000
```

В ответе должна отображаться версия клиента:

```
"ClientVersion": "..."
```

Если версия клиента не определяется, проверьте:

- что была загружена именно распакованная версия клиента;
- что путь к папке клиента указан правильно;
- что в папке клиента есть файл S_Info.exe;
- что ConsoleTools завершился без ошибок;
- что у пользователя backend есть права на запись в папку Data.



После загрузки рекомендуется проверить автообновление S-INFO Desktop с другого компьютера.

30.4. Важно при обновлении backend



При обновлении серверной части S-INFO необходимо также загрузить на сервер новые файлы S-INFO Desktop.



Это отдельный шаг. Замена файлов backend сама по себе не обновляет файлы настольного клиента в серверном хранилище.

После обновления backend выполните загрузку клиента через ConsoleTools:

```
cd /home/sinfo-admin/backend/ConsoleTools
```

```
./ConsoleTools run upgrade /home/sinfo-admin/temp_client
```

31. Ручное резервное копирование через ConsoleTools

При необходимости резервную копию базы данных можно создать вручную через консольную утилиту ConsoleTools, которая входит в поставку backend.

Перейдите в папку утилиты:

```
cd /home/sinfo-admin/backend/ConsoleTools
```

Посмотрите справку по доступным командам:

```
./ConsoleTools --help
```

```
./ConsoleTools run --help
```

Создайте резервную копию базы данных:

```
./ConsoleTools run dump -y
```

По умолчанию утилита читает настройки подключения из файла:

```
/home/sinfo-admin/backend/appsettings.Production.json
```



Для обычной установки дополнительные параметры указывать не требуется.

Если нужно явно указать другой файл настроек, используйте параметр:

```
./ConsoleTools run --use-appsettings-path
```

```
/path/to/appsettings.Production.json dump -y
```



Резервная копия базы данных создается средствами PostgreSQL. Поэтому на сервере должны быть доступны клиентские утилиты PostgreSQL, включая `pg_dump`.

Проверить наличие `pg_dump` можно командой:

```
pg_dump --version
```



Резервное копирование файлов выполняется отдельно — обычным копированием папки `Data` или другой папки, указанной в параметре `Database:DataFolder`.

Пример копирования папки

`Data`

:

```
mkdir -p /home/sinfo-admin/backup/Data
```

```
rsync -a /home/sinfo-admin/backend/Data/ /home/sinfo-admin/backup/Data/
```



При обновлении backend резервную копию базы данных необходимо выполнять обязательно. Также необходимо сохранить папку хранения файлов, конфигурацию, папку `configs` и файл лицензии.

32. Текущий этап установки



К этому моменту серверная часть S-INFO уже должна быть установлена и проверена локально.

Перед настройкой Nginx убедитесь, что выполнены следующие шаги:

- PostgreSQL установлен и запущен.
- Созданы база данных и пользователь PostgreSQL для S-INFO.
- Настроены параметры PostgreSQL, включая `max_connections` и `max_locks_per_transaction`.
- Файлы backend скопированы в рабочую папку сервера.
- Настроен файл `appsettings.Production.json`.
- Настроены необходимые файлы в папке `configs`, включая параметры резервного копирования.
- Backend был запущен вручную.
- Получен и добавлен файл лицензии `server_license.json`.
- При запуске backend успешно выполнил миграции базы данных.
- Backend запускается через `systemd`-службу.
- Служба backend добавлена в автозагрузку.
- Файлы S-INFO Desktop загружены на сервер через ConsoleTools.
- Локальная проверка backend через `curl` возвращает health-статус.
- В health-статусе отображаются версия backend, версия клиента и информация о лицензии.

Проверить локальную доступность backend можно командой:

```
curl http://localhost:5000
```

33. Настройка Nginx как обратного прокси

На этом этапе **backend** уже должен запускаться через `systemd`

и отвечать локально:

```
curl http://localhost:5000
```

Nginx будет принимать внешние HTTP/HTTPS-запросы и передавать их во внутренний backend, работающий на локальном адресе:

```
http://localhost:5000
```

Рекомендуемая схема:

Пользователь / S-INFO Desktop / S-INFO WEB

↓ HTTPS

Nginx

↓ HTTP localhost:5000

S-INFO Backend

Backend будет опубликован по адресу:

```
https://<DOMAIN>/api/
```

Если используется S-INFO WEB, веб-клиент можно разместить на корне домена:

```
https://<DOMAIN>/
```

33.1. Что важно учесть



S-INFO Backend работает через REST API, принимает загрузку файлов, выполняет долгие операции и использует SignalR для быстрых серверных уведомлений.

Поэтому в Nginx необходимо:

- увеличить максимальный размер тела запроса;
- увеличить таймауты проксирования;
- не блокировать WebSocket/SignalR;
- передавать X-Forwarded-* заголовки в backend;
- не кэшировать ответы backend;
- отключить буферизацию для API-запросов и загрузки файлов.

33.2. Установка Nginx

Установите Nginx:

```
sudo apt update
```

```
sudo apt install -y nginx
```

Запустите Nginx и добавьте его в автозагрузку:

```
sudo systemctl enable --now nginx
```

Проверьте состояние службы:

```
sudo systemctl status nginx
```

33.3. Подготовка HTTPS-сертификата



Для рабочей установки рекомендуется использовать сертификат от корпоративного удостоверяющего центра или публичный сертификат для домена.

Для тестовой или закрытой установки можно временно использовать самоподписанный сертификат.

Создайте папку для сертификата:

```
sudo mkdir -p /etc/ssl/sinfo
```

Создайте самоподписанный сертификат:

```
sudo openssl req -x509 -nodes -days 3650 -newkey rsa:4096 \  
-keyout /etc/ssl/sinfo/sinfo.key \  
-out /etc/ssl/sinfo/sinfo.crt \  
-subj "/CN=<DOMAIN>" \  
-addext "subjectAltName=DNS:<DOMAIN>"
```

Если сервер будет открываться по IP-адресу, добавьте IP в subjectAltName

:

```
sudo openssl req -x509 -nodes -days 3650 -newkey rsa:4096 \  
-keyout /etc/ssl/sinfo/sinfo.key \  
-out /etc/ssl/sinfo/sinfo.crt \  
-subj "/CN=<DOMAIN>" \  
-addext "subjectAltName=DNS:<DOMAIN>,IP:<SERVER_IP>"
```

Назначьте права:

```
sudo chmod 600 /etc/ssl/sinfo/sinfo.key
```

```
sudo chmod 644 /etc/ssl/sinfo/sinfo.crt
```

Замените:

Значение	Что указать
<DOMAIN>	Доменное имя сервера
<SERVER_IP>	IP-адрес сервера, если доступ будет выполняться по IP

Таблица 11

33.4. Создание конфигурации сайта

Создайте файл конфигурации:

```
sudo nano /etc/nginx/sites-available/sinfo.conf
```

Добавьте конфигурацию:

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    "" "";
}

upstream sinfo_backend {
    server 127.0.0.1:5000;
    keepalive 64;
}

server {
    listen 80;

    server_name <DOMAIN>;

    server_tokens off;

    client_max_body_size 100M;

    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl http2;
```

```

server_name <DOMAIN>;

server_tokens off;

ssl_certificate      /etc/ssl/sinfo/sinfo.crt;
ssl_certificate_key  /etc/ssl/sinfo/sinfo.key;

ssl_protocols TLSv1.2 TLSv1.3;
ssl_prefer_server_ciphers off;
ssl_session_cache shared:SSL:50m;
ssl_session_timeout 1d;
ssl_session_tickets off;

client_max_body_size 100M;

proxy_read_timeout 43200;
proxy_connect_timeout 43200;
proxy_send_timeout 43200;

access_log /var/log/nginx/sinfo_access.log;
error_log /var/log/nginx/sinfo_error.log warn;

# ===== Backend S-INFO =====
# Весь backend публикуется под /api/.
# /api/ -> http://127.0.0.1:5000/
# /api/signalr_hub -> http://127.0.0.1:5000/signalr_hub
# /api/<method> -> http://127.0.0.1:5000/<method>
location /api/ {

```

```

proxy_pass http://sinfo_backend/;
proxy_http_version 1.1;

proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Forwarded-Host $host;
proxy_set_header X-Forwarded-Prefix /api;

# WebSocket / SignalR
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection $connection_upgrade;

# Долгие REST-запросы, отчеты, импорт, экспорт
proxy_read_timeout 43200;
proxy_send_timeout 43200;
proxy_connect_timeout 43200;

# Большие загрузки файлов
proxy_request_buffering off;
proxy_buffering off;

# API не кэшируется
proxy_cache off;
proxy_no_cache 1;
proxy_cache_bypass 1;
add_header Cache-Control "no-store" always;

```

```

}

# ===== S-INFO WEB, если веб-клиент установлен =====
root /var/www/sinfo-web;
index index.html;

location = /index.html {
    add_header Cache-Control "no-cache, must-revalidate" always;
    etag on;
    try_files $uri =404;
}

location = /config.json {
    add_header Cache-Control "no-cache, must-revalidate" always;
    etag on;
    try_files $uri =404;
}

location ~* \.(js|mjs|css|png|jpg|jpeg|gif|svg|webp|ico|woff2?|ttf|eot)$ {
    expires 2h;
    add_header Cache-Control "public, max-age=7200" always;
    etag on;
    try_files $uri =404;
}

location / {
    try_files $uri $uri/ /index.html;
}

```

}

Замените:

Значение	Что указать
<DOMAIN>	Доменное имя сервера
/etc/ssl/sinfo/sinfo.crt	Путь к SSL-сертификату
/etc/ssl/sinfo/sinfo.key	Путь к закрытому ключу
/var/www/sinfo-web	Путь к опубликованному S-INFO WEB, если используется веб-клиент

Таблица 12

33.5. Если S-INFO WEB не используется



Если веб-клиент не используется, можно не размещать файлы в `/var/www/sinfo-web`.

В этом случае для проверки backend используйте адрес:

`https://<DOMAIN>/api/`



Именно по этому адресу должен открываться health-статус backend.

При необходимости можно заменить блок

`location /`

на простую страницу-заглушку:

```
location / {  
    default_type text/plain;  
    return 200 "S-INFO Backend is available at /api/\n";  
}
```



Тогда при открытии корня домена будет выводиться короткое сообщение, а сам backend останется доступен по `/api/`.

33.6. Включение сайта

Удалите стандартный сайт Nginx, если он не используется:

```
sudo rm -f /etc/nginx/sites-enabled/default
```

Создайте ссылку на конфигурацию S-INFO:

```
sudo ln -s /etc/nginx/sites-available/sinfo.conf /etc/nginx/sites-enabled/sinfo.conf
```

Проверьте конфигурацию Nginx:

```
sudo nginx -t
```

Если проверка успешна, примените настройки:

```
sudo systemctl reload nginx
```

Проверьте состояние Nginx:

```
sudo systemctl status nginx
```

33.7. Настройка appsettings.Production.json

Если backend публикуется через Nginx по адресу

/api/,

рекомендуется проверить в

appsettings.Production.json

:

```
"ServerSettings": {  
    "UseForwardedHeaders": true,  
    "PublicServerUrl": "https://<DOMAIN>/api"  
}
```

Если используется S-INFO WEB, укажите адрес веб-клиента:

```
"Clients": {  
    "WebApplication": {  
        "Url": "https://<DOMAIN>"  
    }  
}
```

После изменения

appsettings.Production.json

перезапустите backend:

```
sudo systemctl restart BackendWebService.service
```

33.8. Проверка с сервера

Проверьте **backend** через **Nginx** локально:

```
curl -k https://<DOMAIN>/api/
```



При успешной настройке сервер вернет JSON со статусом работы backend.

Если используется самоподписанный сертификат, параметр `-k` отключает проверку доверия к сертификату только для этой команды.

33.9. Проверка с другого компьютера

С другого компьютера откройте в браузере:

`https://<DOMAIN>/api/`



Должен открыться health-статус backend.

Если используется S-INFO WEB, откройте:

`https://<DOMAIN>/`



Должно открыться веб-приложение.

33.10. Проверка SignalR



SignalR также проходит через общий backend-прокси `/api/`.

Если уведомления в клиентах не приходят или SignalR-соединение нестабильно, проверьте:

- что клиент использует адрес backend с префиксом `/api/`;
- что Nginx применил актуальную конфигурацию;
- что не удалены заголовки `Upgrade` и `Connection`;
- что таймауты `proxy_read_timeout` и `proxy_send_timeout` увеличены;
- что между клиентом и сервером нет другого прокси, который блокирует WebSocket.

33.11. Проверка журналов Nginx

Журнал ошибок:

```
sudo tail -n 100 /var/log/nginx/sinfo_error.log
```

Журнал запросов:

```
sudo tail -n 100 /var/log/nginx/sinfo_access.log
```

Журнал ошибок в реальном времени:

```
sudo tail -f /var/log/nginx/sinfo_error.log
```

33.12. Если используется межсетевой экран

Если включен

ufw,

разрешите HTTP и HTTPS:

```
sudo ufw allow 80/tcp
```

```
sudo ufw allow 443/tcp
```

```
sudo ufw status
```



Внешний порт 5000 открывать не требуется, если backend работает только через Nginx.

33.13. Типовые проблемы

Проблема	Что проверить
502 Bad Gateway	Запущена ли служба backend, отвечает ли curl http://localhost:5000
413 Request Entity Too Large	Увеличить client_max_body_size
Долгий запрос обрывается	Проверить proxy_read_timeout, proxy_sendtimeout, proxy_connect_timeout
Не работает SignalR	Проверить WebSocket-заголовки Upgrade и Connection
Backend видит HTTP вместо HTTPS	Проверить X-Forwarded-Proto и UseForwardedHeaders
Не открывается S-INFO WEB	Проверить путь root /var/www/sinfo-web и наличие index.html
Не работает API по /api/	Проверить proxy_pass http://sinfo_backend/; и что backend локально отвечает на 5000
По /api/ нет health-статуса	Проверить, что в location /api/ указан proxy_pass http://sinfo_backend/; именно со слэшем на конце

Таблица 13

34. Установка S-INFO WEB



S-INFO WEB — это отдельное веб-приложение. Backend сам его не запускает и не публикует.

Веб-клиент представляет собой статическое SPA-приложение на Vue 3. Его файлы размещаются в отдельной папке, а Nginx отдает их как обычную статику. Адрес backend указывается в конфигурации веб-клиента.

34.1. Загрузка файлов S-INFO WEB

Скачайте архив:

S-INFO WEB

из центра загрузок:

<https://download.sinfo.tech/all>



Версия S-INFO WEB должна соответствовать версии backend.

В примерах ниже веб-клиент размещается в папке:

`/var/www/sinfo-web`

Это рекомендуемый пример. При необходимости можно использовать другой путь, но тогда этот же путь нужно указать в конфигурации Nginx:

root `/var/www/sinfo-web;`

34.2. Создание папки

Создайте папку для веб-клиента:

```
sudo mkdir -p /var/www/sinfo-web
```

34.3. Копирование файлов веб-клиента



Распакуйте архив S-INFO WEB во временную папку, затем скопируйте файлы в /var/www/sinfo-web.

Пример:

```
sudo rsync -a --delete /home/sinfo-admin/temp_web/ /var/www/sinfo-web/
```

После копирования в папке

/var/www/sinfo-web

должен находиться файл:

index.html

Проверить содержимое папки:

```
ls -la /var/www/sinfo-web
```

34.4. Настройка прав для Nginx



Nginx должен иметь права на чтение файлов веб-клиента и проход по каталогам.

Базовый вариант:

```
sudo chown -R root:www-data /var/www/sinfo-web  
sudo find /var/www/sinfo-web -type d -exec chmod 750 {} \;  
sudo find /var/www/sinfo-web -type f -exec chmod 640 {} \;
```

При такой настройке:

- владелец файлов — root;
- группа — www-data;
- Nginx может читать файлы;
- обычные пользователи не имеют доступа к папке.

34.5. Вариант для загрузки файлов через SFTP



Если требуется подключаться к серверу через SFTP от имени пользователя `sinfo-admin` и вручную заменять файлы S-INFO WEB, можно сделать владельцем папки пользователя `sinfo-admin`, а группе `www-data` оставить доступ на чтение.

Назначьте владельца и группу:

```
sudo chown -R sinfo-admin:www-data /var/www/sinfo-web
```

Выставьте права:

```
sudo find /var/www/sinfo-web -type d -exec chmod 750 {} \;
```

```
sudo find /var/www/sinfo-web -type f -exec chmod 640 {} \;
```

Включите наследование группы `www-data` для новых файлов и папок:

```
sudo find /var/www/sinfo-web -type d -exec chmod g+s {} \;
```



После этого пользователь `sinfo-admin` сможет загружать и заменять файлы через SFTP, а Nginx сможет читать их через группу `www-data`.

Если после загрузки новых файлов через SFTP веб-клиент не открывается, повторно примените права:

```
sudo chown -R sinfo-admin:www-data /var/www/sinfo-web
```

```
sudo find /var/www/sinfo-web -type d -exec chmod 750 {} \;
```

```
sudo find /var/www/sinfo-web -type f -exec chmod 640 {} \;
```

```
sudo find /var/www/sinfo-web -type d -exec chmod g+s {} \;
```

34.6. Проверка доступа Nginx к файлам

Проверьте, что пользователь Nginx может прочитать
index.html

:

```
sudo -u www-data test -r /var/www/sinfo-web/index.html && echo "OK"
```

Если команда вернула:

OK



значит Nginx имеет доступ к файлам веб-клиента.

34.7. Настройка config.json

Файл конфигурации веб-клиента находится рядом с index.html

:

/var/www/sinfo-web/config.json

Откройте файл:

```
nano /var/www/sinfo-web/config.json
```

Если backend опубликован через Nginx по адресу:

https://<DOMAIN>/api/

то в

config.json

можно указать относительный адрес:

```
{  
  "BackendUrl": "/api/",  
  "AxiosRateLimit": null,  
  "UseAxiosCache": false,  
  "HideWarningLoginPageNotServerSet": false,  
  "DisablePrefixMergeProperties": false  
}
```

При такой настройке S-INFO WEB будет обращаться к backend на том же домене, с которого открыт веб-клиент:

https://<DOMAIN>/api/



Это рекомендуемый вариант, если S-INFO WEB и backend опубликованы через один Nginx.

Если backend находится на другом домене, в

BackendUrl

нужно указать полный адрес backend:

```
{  
  "BackendUrl": "https://api.example.ru/",  
  "AxiosRateLimit": null,  
  "UseAxiosCache": false,  
  "HideWarningLoginPageNotServerSet": false,  
  "DisablePrefixMergeProperties": false  
}
```



После изменения config.json перезапуск backend не требуется. Достаточно обновить страницу S-INFO WEB в браузере.

Если браузер продолжает использовать старую конфигурацию, очистите кэш страницы или откройте веб-клиент в приватном окне.

34.8. Проверка через браузер

После настройки Nginx откройте:

`https://<DOMAIN>/`



Если S-INFO WEB установлен корректно, должна открыться стартовая страница веб-клиента.

Backend при рекомендуемой схеме доступен отдельно:

`https://<DOMAIN>/api/`



По этому адресу должен открываться health-статус backend.

35. Финальный чек-лист установки

После завершения установки проверьте, что серверная часть S-INFO, база данных, Nginx, клиентские файлы и резервное копирование работают корректно.

35.1. Проверка службы backend

Проверьте состояние службы:

```
sudo systemctl status BackendWebService.service
```

Служба должна быть в состоянии:

```
active (running)
```

Проверьте, что служба включена в автозагрузку:

```
systemctl is-enabled BackendWebService.service
```

Ожидаемый результат:

```
enabled
```

35.2. Проверка локальной доступности backend

На сервере проверьте, что backend отвечает локально:

```
curl http://localhost:5000
```



В ответ должен вернуться health-статус backend в формате JSON.

В ответе проверьте:

- "Status": "Healthy";
- "Mode": "Production";
- версию backend;
- версию клиента;
- информацию о лицензии.

35.3. Проверка доступности через Nginx

С сервера или другого компьютера откройте адрес backend через Nginx:

`https://<DOMAIN>/api/`



Должен открыться health-статус backend.

Проверьте, что в ответе отображаются:

- статус Healthy;
- режим Production;
- версия backend;
- версия клиента;
- данные лицензии;
- количество активных подключений.



Если страница не открывается, проверьте Nginx, DNS, сертификат, правила межсетевого экрана и работу службы backend.

35.4. Проверка S-INFO WEB

Если установлен модуль S-INFO WEB, откройте в браузере:

`https://<DOMAIN>/`

Проверьте:

- веб-клиент открывается без ошибок;
- config.json указывает на правильный адрес backend;
- авторизация проходит успешно;
- после входа нет ошибок подключения к серверу;
- в браузере нет критических ошибок CORS;
- основные разделы системы загружаются корректно.



Если веб-клиент не используется, этот пункт можно пропустить.

35.5. Проверка S-INFO Desktop



Откройте настольное приложение S-INFO Desktop на клиентском компьютере.

Укажите адрес сервера:

`https://<DOMAIN>/api/`

Проверьте:

- клиент подключается к серверу;
- авторизация проходит успешно;
- отображается корректная версия сервера;
- нет ошибок соединения;
- работает получение данных из базы;
- при необходимости проверяется автообновление клиента.



Если клиент сообщает о несовпадении версий или не предлагает обновление, проверьте, что файлы S-INFO Desktop были загружены на сервер через ConsoleTools.

35.6. Проверка файлов клиента на сервере

Проверьте health-статус backend:

```
curl http://localhost:5000
```

В ответе должна отображаться версия клиента:

```
"ClientVersion": "..."
```



Если версия клиента не определяется, повторно загрузите файлы клиента через ConsoleTools.

35.7. Проверка SMTP



Если на сервере настроена отправка почты, проверьте SMTP через административные инструменты S-INFO или через ConsoleTools, если используется соответствующий пункт меню.

Проверьте:

- SMTP-сервер доступен;
- логин и пароль указаны корректно;
- письма отправляются;
- письма восстановления пароля доходят до пользователя;
- системные уведомления могут отправляться на адреса из AlertEmails.



Если SMTP не настроен, почтовые уведомления и восстановление пароля через email работать не будут.

35.8. Проверка резервного копирования

Убедитесь, что настроено резервное копирование:

- базы данных PostgreSQL;
- папки хранения файлов Data или другой папки из Database:DataFolder;
- конфигурации backend;
- папки configs;
- файла server_license.json.

Если используется встроенное расписание S-INFO, проверьте файл:

/home/sinfo-admin/backend/configs/systemScheduler.settings.json

Убедитесь, что включены нужные задачи:

- резервное копирование базы данных;
- резервное копирование файловой папки, если оно требуется;
- архивация журналов;
- очистка временных данных;
- мониторинг проблем.

Проверить выполнение фоновых задач можно двумя способами:

1. Дождаться планового запуска по расписанию.
2. Запустить нужную задачу вручную из веб-клиента через панель администратора.



После выполнения проверьте, что резервная копия действительно создана и доступна для восстановления.

35.9. Проверка журналов

Проверьте журналы backend:

```
tail -n 100 /home/sinfo-admin/backend/logs/sinfo_log_*.log
```

Проверьте журнал службы:

```
sudo journalctl -u BackendWebService.service -n 100 --no-pager
```

Проверьте журналы Nginx:

```
sudo tail -n 100 /var/log/nginx/sinfo_error.log
```

```
sudo tail -n 100 /var/log/nginx/sinfo_access.log
```



В журналах не должно быть повторяющихся критических ошибок, ошибок лицензии, ошибок подключения к PostgreSQL или ошибок доступа к папке Data.

35.10. Проверка прав доступа

Убедитесь, что пользователь, от имени которого запущен backend, имеет права на запись в рабочие папки:

```
ls -la /home/sinfo-admin/backend
```

Особенно важно проверить доступ к:

- Data;
- logs;
- configs;
- server_license.json.



Backend должен иметь возможность создавать, изменять и удалять файлы в папке хранения данных.

35.11. Проверка режима работы

В health-статусе backend должно быть:

```
"Mode": "Production"
```

Если отображается

Development,

проверьте переменные окружения службы

systemd

:

```
sudo systemctl cat BackendWebService.service
```

Для рабочей установки должны быть указаны:

```
Environment=ASPNETCORE_ENVIRONMENT=Production
```

```
Environment=DOTNET_ENVIRONMENT=Production
```

После изменения службы выполните:

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart BackendWebService.service
```

35.12. Итог

Установка считается завершенной, если:

- PostgreSQL установлен, настроен и доступен backend;
- backend запускается через systemd;
- backend работает в режиме Production;
- лицензия применена успешно;
- health-статус открывается через браузер;
- Nginx публикует backend по внешнему адресу;
- S-INFO Desktop подключается к серверу;
- S-INFO WEB открывается и авторизация работает, если модуль используется;
- файлы S-INFO Desktop загружены на сервер;
- резервное копирование базы данных и файлов настроено;
- в журналах нет критических ошибок.



После этого сервер готов к эксплуатации.

36. Обновление S-INFO



Этот раздел описывает общий порядок обновления серверной части S-INFO, файлов настольного клиента и S-INFO WEB.



Перед обновлением рекомендуется изучить материалы к новой версии в центре загрузок и на сайте



Если к версии приложены заметки к выпуску, их необходимо прочитать до обновления. В отдельных версиях могут быть дополнительные требования: перевыпуск лицензии, изменение конфигурации, обновление зависимостей или специальные действия перед запуском.

36.1. Важное условие по лицензии



Возможность обновления зависит от условий лицензии и срока технической поддержки.

Если срок технической поддержки или действия лицензии истек, сервер может не принять новую версию, собранную после даты окончания поддержки. В этом случае можно использовать только те версии, которые разрешены действующей лицензией.

Перед обновлением на новую версию убедитесь, что лицензия позволяет запуск этой версии backend.

36.2. Что скачать

Из центра загрузок необходимо скачать компоненты нужной версии:

- S-INFO Backend для нужной платформы;
- S-INFO Desktop unpacked для загрузки файлов клиента на сервер;
- S-INFO WEB, если используется веб-клиент.



Версии backend, настольного клиента и веб-клиента должны соответствовать друг другу, если в заметках к выпуску не указано иное.

36.3. Что обязательно сохранить

Перед заменой файлов необходимо сохранить:

- базу данных PostgreSQL;
- папку хранения файлов Data или другую папку, указанную в Database:DataFolder;
- appsettings.Production.json;
- папку configs;
- server_license.json;
- web.config, если используется Windows/IIS;
- журналы работы, если они нужны для диагностики.



Backend при запуске автоматически выполняет миграции базы данных. Поэтому резервная копия базы данных перед обновлением обязательна.

36.4. Общий порядок обновления backend

1. Изучить заметки к новой версии.
2. Скачать новую версию backend из центра загрузок.
3. Скачать распакованный S-INFO Desktop той же версии.
4. Если используется S-INFO WEB, скачать его новую версию.

5.

Остановить службу backend:

```
sudo systemctl stop BackendWebService.service
```

6. Сделать резервную копию базы данных.
7. Сделать резервную копию папки Data или другой папки хранения файлов.
8. Сохранить конфигурацию, папку configs и файл server_license.json.
9. Скопировать файлы новой версии backend в рабочую папку с заменой исполняемых файлов и библиотек.
10. Проверить, что не были перезаписаны локальные конфигурации и лицензия.

11.

Запустить backend вручную:

```
cd /home/sinfo-admin/backend
```

```
./BackendWebService
```

12. Дождаться завершения миграций базы данных.

13.

Дождаться успешного запуска backend и строки вида:

```
Server is listening on "http://localhost:5000"
```

14.

Проверить health-статус:

```
curl http://localhost:5000
```

15.

Остановить ручной запуск:

```
Ctrl + C
```

16.

Запустить службу backend:

```
sudo systemctl start BackendWebService.service
```

17.

Проверить состояние службы:

```
sudo systemctl status BackendWebService.service
```

18.

Проверить внешний health-статус через Nginx:

```
https://<DOMAIN>/api/
```

36.5. Обновление файлов S-INFO Desktop на сервере



После обновления backend необходимо загрузить на сервер файлы S-INFO Desktop новой версии.

Это отдельный обязательный шаг. Замена файлов backend не обновляет файлы настольного клиента в серверном хранилище.

Распакуйте архив

S-INFO Desktop (unpacked)

во временную папку, например:

```
/home/sinfo-admin/temp_client
```

Загрузите файлы клиента через

ConsoleTools

:

```
cd /home/sinfo-admin/backend/ConsoleTools
```

```
./ConsoleTools run upgrade /home/sinfo-admin/temp_client
```

После загрузки проверьте health-статус backend. В нем должны корректно отображаться:

- версия backend;
- версия клиента.



Если версия клиента не изменилась, проверьте, что была загружена распакованная версия клиента нужной версии и что команда `ConsoleTools run upgrade` завершилась без ошибок.

36.6. Обновление S-INFO WEB



Если используется S-INFO WEB, обновите файлы веб-клиента отдельно.

Обычно порядок такой:

1. Скачать новую версию S-INFO WEB.
2. Распаковать архив во временную папку.
3. Скопировать файлы в папку веб-клиента с заменой.
4. Не перезаписывать локальные файлы конфигурации.

Для установки через Nginx обычно важно сохранить:

`config.json`

Для установки через IIS также может потребоваться сохранить:

`web.config`

Пример обновления через

`rsync`

с исключением

`config.json`

:

```
sudo rsync -a --delete \  
  --exclude config.json \  
  /home/sinfo-admin/temp_web/ \  
  /var/www/sinfo-web/
```

После копирования проверьте права:

```
sudo chown -R sinfo-admin:www-data /var/www/sinfo-web
```

```
sudo find /var/www/sinfo-web -type d -exec chmod 750 {} \;  
sudo find /var/www/sinfo-web -type f -exec chmod 640 {} \;  
sudo find /var/www/sinfo-web -type d -exec chmod g+s {} \;
```

Затем откройте веб-клиент в браузере:

`https://<DOMAIN>/`



Если браузер продолжает использовать старые файлы, очистите кэш страницы или откройте веб-клиент в приватном окне.

36.7. Что проверить после обновления

После обновления проверьте:

- backend запускается через службу;
- health-статус открывается через браузер;
- в health-статусе указан режим Production;
- версия backend отображается корректно;
- версия клиента отображается корректно;
- версия backend и версия клиента соответствуют обновляемой версии;
- лицензия применена успешно;
- S-INFO Desktop подключается к серверу;
- автообновление S-INFO Desktop работает;
- S-INFO WEB открывается и проходит авторизация, если модуль используется;
- SMTP работает, если настроена отправка почты;
- фоновые задачи запущены;
- резервное копирование продолжает работать;
- в журналах нет критических ошибок.

36.8. Если обновление не прошло

Если после обновления backend не запускается или миграция базы данных завершилась ошибкой:

1. Не продолжайте эксплуатацию сервера в неисправном состоянии.
2. Сохраните журналы backend и journalctl.
3. Проверьте ошибку в логах.
4. При необходимости верните предыдущую версию файлов backend.
5. Восстановите базу данных из резервной копии.
6. Восстановите папку Data, если она была повреждена или перезаписана.
7. Повторите обновление после устранения причины ошибки.



Именно поэтому резервная копия базы данных и папки хранения файлов перед обновлением обязательна.

37. Частые проблемы и способы диагностики



В этом разделе собраны типовые проблемы, которые могут возникнуть при установке, обновлении и эксплуатации S-INFO Backend, S-INFO Desktop и S-INFO WEB.

В большинстве случаев причина ошибки видна в журналах backend, Nginx, PostgreSQL или клиентского приложения. Перед изменением настроек сначала найдите ошибку в логах и определите, где именно она возникает: на backend, в Nginx, в PostgreSQL, в веб-клиенте, в настольном клиенте или на сетевом уровне.

37.1. Основные журналы

Журналы backend



Backend пишет журналы в папку logs, которая находится рядом с исполняемым файлом backend.

Для стандартного примера установки:

```
/home/sinfo-admin/backend/logs
```

Посмотреть последние строки логов:

```
tail -n 200 /home/sinfo-admin/backend/logs/sinfo_log_*.log
```

Отсортировать файлы логов по дате изменения:

```
ls -lt /home/sinfo-admin/backend/logs
```

Журнал службы

systemd

:

```
sudo journalctl -u BackendWebService.service -n 200 --no-pager
```

Журнал службы в реальном времени:

```
sudo journalctl -u BackendWebService.service -f
```



Если журнал systemd не содержит полной информации, проверяйте физические файлы в папке logs.

37.2. Журналы S-INFO Desktop

У настольного клиента также есть папка:

logs



Она находится в каталоге приложения S-INFO Desktop.

Если ошибка возникает на клиентском компьютере, проверьте эту папку. Для передачи в техническую поддержку рекомендуется архивировать и отправлять всю папку logs, а не один файл. Внутри может быть несколько типов журналов.

Чтобы найти последние записи, отсортируйте файлы по дате изменения.

37.3. Журналы Nginx

Для конфигурации из инструкции:

```
sudo tail -n 200 /var/log/nginx/sinfo_error.log  
sudo tail -n 200 /var/log/nginx/sinfo_access.log
```

Если используются стандартные пути Nginx:

```
sudo tail -n 200 /var/log/nginx/error.log  
sudo tail -n 200 /var/log/nginx/access.log
```

Проверить конфигурацию Nginx:

```
sudo nginx -t
```

37.4. Журналы PostgreSQL

Проверить службу PostgreSQL:

```
sudo systemctl status postgresql
```

Посмотреть журнал PostgreSQL:

```
sudo journalctl -u postgresql -n 200 --no-pager
```

38. Что отправлять в техническую поддержку

При обращении в техническую поддержку рекомендуется подготовить:

- архив папки `logs backend`;
- журнал службы `backend` из `journalctl`;
- архив папки `logs` настольного клиента, если ошибка на клиентском ПК;
- журналы `Nginx`, если проблема связана с доступом через браузер, веб-клиентом или внешним адресом;
- описание действия, при котором возникает ошибка;
- точное время возникновения ошибки;
- адрес сервера, по которому выполнялось подключение;
- версию `backend` и клиента из `health`-статуса.



Не отправляйте пароли, токены,



содержимое `server_license.json`, `security.auth.json` и реальные секреты интеграций.

39. Backend не запускается

Возможные причины:

- ошибка в `appsettings.Production.json`;
- неверная строка подключения к PostgreSQL;
- PostgreSQL не запущен или недоступен;
- нет прав на папки `Data`, `logs` или `configs`;
- отсутствует или не подходит `server_license.json`;
- служба запускается не от того пользователя, под которого выпущена лицензия;
- не хватает системных библиотек Linux;
- порт backend уже занят другим процессом.

Проверьте службу:

```
sudo systemctl status BackendWebService.service
```

```
sudo journalctl -u BackendWebService.service -n 200 --no-pager
```

Проверьте ручной запуск:

```
cd /home/sinfo-admin/backend
```

```
./BackendWebService
```

Проверьте, занят ли порт

5000

:

```
sudo ss -tulnp | grep 5000
```

Проверьте права на папку backend:

```
ls -la /home/sinfo-admin/backend
```

```
ls -la /home/sinfo-admin/backend/Data
```

```
ls -la /home/sinfo-admin/backend/logs
```

```
ls -la /home/sinfo-admin/backend/configs
```

40. Ошибка лицензии или backend выводит новый ключ пользователя

Если backend раньше запускался, но после обновления или изменения сервера начал падать с ошибкой лицензии и выводит новый ключ пользователя, вероятные причины:

- изменился пользователь операционной системы, от имени которого запускается backend;
- backend перенесли на другой сервер или виртуальную машину;
- изменилась часть системной конфигурации сервера;
- изменился механизм лицензирования в новой версии;
- устанавливаемая версия backend не разрешена текущей лицензией или сроком технической поддержки.

Проверьте пользователя службы:

```
sudo systemctl cat BackendWebService.service
```



В файле службы параметр User= должен указывать на того же пользователя, под которым был получен ключ для лицензии.

Если ключ изменился, необходимо выпустить новую лицензию и заменить файл:

```
server_license.json
```

Файл лицензии должен лежать рядом с исполняемым файлом:

```
/home/sinfo-admin/backend/server_license.json
```

41. Backend локально работает, но недоступен с других компьютеров

Если на сервере backend отвечает:

```
curl http://localhost:5000
```

но с другого компьютера не открывается:

```
https://<DOMAIN>/api/
```



значит backend как приложение работает, а проблема находится на уровне сети, Nginx, DNS, сертификата или межсетевого экрана.

Проверьте:

```
sudo nginx -t
```

```
sudo systemctl status nginx
```

```
sudo ss -tulnp | grep nginx
```

```
sudo ufw status
```

Также проверьте:

- домен указывает на правильный сервер;
- открыт порт 443;
- Nginx применил актуальную конфигурацию;
- сертификат доверен клиентскому компьютеру;
- корпоративный прокси или межсетевой экран не блокирует соединение;
- клиент использует правильный адрес сервера: `https://<DOMAIN>/api/`.



Внешний порт 5000 открывать не требуется, если backend опубликован через Nginx.

42. 502 Bad Gateway



Ошибка 502 Bad Gateway обычно означает, что Nginx не может подключиться к backend.

Проверьте:

```
sudo systemctl status BackendWebService.service
```

```
curl http://localhost:5000
```

```
sudo tail -n 100 /var/log/nginx/sinfo_error.log
```

Типовые причины:

- backend не запущен;
- backend упал при запуске;
- backend слушает не localhost:5000;
- в Nginx указан неправильный upstream;
- ошибка лицензии;
- ошибка подключения к PostgreSQL;
- служба запускается от неправильного пользователя.

43. 413 Request Entity Too Large



Ошибка возникает при загрузке больших файлов, если в Nginx слишком маленький лимит тела запроса.

Проверьте в конфигурации Nginx:

```
client_max_body_size 100M;
```



Если пользователи загружают файлы больше 100 МБ, увеличьте значение.

После изменения:

```
sudo nginx -t
```

```
sudo systemctl reload nginx
```

44. Долгие запросы обрываются



Если обрываются длительные операции, например генерация отчетов, импорт, экспорт или обработка больших данных, проверьте таймауты Nginx.

В конфигурации должны быть увеличены параметры:

```
proxy_read_timeout 43200;
```

```
proxy_connect_timeout 43200;
```

```
proxy_send_timeout 43200;
```

После изменения:

```
sudo nginx -t
```

```
sudo systemctl reload nginx
```

45. Не работают обновления данных в реальном времени



Если при многопользовательской работе изменения не приходят другим пользователям без ручного обновления, вероятно, не проходит SignalR/WebSocket.

Проверьте, что в Nginx внутри блока

/api/

есть:

```
proxy_http_version 1.1;
```

```
proxy_set_header Upgrade $http_upgrade;
```

```
proxy_set_header Connection $connection_upgrade;
```

Также проверьте:

- клиент подключается к backend по адресу `https://<DOMAIN>/api/`;
- Nginx применил актуальную конфигурацию;
- между клиентом и сервером нет прокси, который блокирует WebSocket;
- в логах клиента или backend нет ошибок SignalR/WebSocket;
- таймауты `proxy_read_timeout` и `proxy_send_timeout` увеличены.

46. По /api/ не открывается health-статус



При рекомендуемой настройке backend опубликован под /api/.

Health-статус должен открываться по адресу:

https://<DOMAIN>/api/

В Nginx важно, чтобы в блоке

`location /api/`

был указан

`proxy_pass`

со слэшем на конце:

```
location /api/ {  
    proxy_pass http://sinfo_backend/  
}
```

Тогда Nginx передает запросы так:

/api/	-> /
/api/signalr_hub	-> /signalr_hub
/api/<method>	-> /<method>



Если слэш в `proxy_pass` указан неправильно, backend может получать неверные пути.

47. Backend видит HTTP вместо HTTPS

Если backend формирует ссылки с

`http://` вместо `https://`

, проверьте передачу заголовков от Nginx:

```
proxy_set_header X-Forwarded-Proto $scheme;
```

```
proxy_set_header X-Forwarded-Host $host;
```

```
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```

В

`appsettings.Production.json`

должно быть включено:

```
"ServerSettings": {  
  "UseForwardedHeaders": true,  
  "PublicServerUrl": "https://<DOMAIN>/api"  
}
```

После изменения настроек перезапустите backend:

```
sudo systemctl restart BackendWebService.service
```

48. Backend работает не в режиме Production

В health-статусе должно быть:

```
"Mode": "Production"
```

Если отображается Development, проверьте службу:

```
sudo systemctl cat BackendWebService.service
```

В файле службы должны быть переменные:

```
Environment=ASPNETCORE_ENVIRONMENT=Production
```

```
Environment=DOTNET_ENVIRONMENT=Production
```

После изменения:

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart BackendWebService.service
```



Swagger доступен только при запуске backend в режиме Development. Для рабочей установки режим Development использовать не рекомендуется.

49. Ошибка подключения к PostgreSQL

Проверьте состояние PostgreSQL:

```
sudo systemctl status postgresql
```

```
pg_isready
```

Проверьте подключение пользователем S-INFO:

```
psql "host=localhost port=5432 dbname=sinfo_db user=sinfo_user password=<DB_PASSWORD>"
```

Проверьте строку подключения в файле:

```
/home/sinfo-admin/backend/appsettings.Production.json
```

Типовые причины:

- неверное имя базы данных;
- неверный пользователь или пароль;
- PostgreSQL не запущен;
- PostgreSQL находится на другом сервере и недоступен по сети;
- не настроены listen_addresses и pg_hba.conf, если база вынесена на отдельную машину;
- межсетевой экран блокирует порт 5432.

50. Ошибки прав PostgreSQL при миграциях



Backend самостоятельно создает схемы, таблицы, индексы и служебные объекты базы данных.

Пользователь PostgreSQL должен быть владельцем своей базы и иметь право создавать объекты внутри нее.

Проверьте владельца базы:

```
sudo -u postgres psql -c "\l sinfo_db"
```

Проверьте пользователя:

```
sudo -u postgres psql -c "\du sinfo_user"
```

Если нужно пересоздать правильные права:

```
ALTER DATABASE sinfo_db OWNER TO sinfo_user;
```

```
GRANT CONNECT, TEMPORARY, CREATE ON DATABASE sinfo_db TO sinfo_user;
```

```
GRANT USAGE, CREATE ON SCHEMA public TO sinfo_user;
```



Если схемы уже созданы другим пользователем, администратор PostgreSQL должен передать права владельцу базы или выдать необходимые права на эти схемы.

51. Ошибки на больших базах или при миграциях

Если при миграциях, обслуживании базы, резервном копировании или работе с большим количеством объектов возникают ошибки PostgreSQL, проверьте параметры:

```
sudo -u postgres psql -c "SHOW max_connections;"
```

```
sudo -u postgres psql -c "SHOW max_locks_per_transaction;"
```

Рекомендуемые значения для S-INFO:

```
sudo -u postgres psql -c "ALTER SYSTEM SET max_connections = '500';"
```

```
sudo -u postgres psql -c "ALTER SYSTEM SET max_locks_per_transaction = '5000';"
```

```
sudo systemctl restart postgresql
```

Проверьте свободное место на диске:

```
df -h
```

Проверьте журнал PostgreSQL:

```
sudo journalctl -u postgresql -n 200 --no-pager
```



Оптимизация PostgreSQL под фактическую нагрузку является отдельной задачей системного администратора.

52. При первом запуске видны ошибки __EFMigrationsHistory



На пустой базе данных в начале миграции могут появляться ошибки чтения таблицы истории миграций __EFMigrationsHistory.

Если после этого миграции продолжают выполняться и завершаются строкой:

Database migration successfully



то это не считается ошибкой установки.



Если миграция остановилась и backend завершился с ошибкой, нужно смотреть первую фактическую ошибку миграции в журнале.

53. Нет прав на папку Data



Если backend не может сохранять файлы, документы, модели или временные данные, проверьте права на папку Data.



Пользователь службы backend должен иметь права на чтение, создание, изменение и удаление файлов.

Проверка пользователя службы:

```
sudo systemctl cat BackendWebService.service
```

Проверка прав:

```
ls -la /home/sinfo-admin/backend/Data
```

Исправление для стандартного примера:

```
sudo chown -R sinfo-admin:sinfo-admin /home/sinfo-admin/backend/Data
```

```
chmod -R u=rwX,g=rX,o= /home/sinfo-admin/backend/Data
```

54. Не работает восстановление пароля или почтовые уведомления



Восстановление пароля использует email. Если SMTP не настроен, пользователь не получит код восстановления.

Проверьте файл:

`/home/sinfo-admin/backend/configs/mailSmtplib.settings.json`

Проверьте:

- Host;
- Port;
- Login;
- Email;
- Password;
- SslMode;
- доступность SMTP-сервера с backend-сервера;
- что почтовый сервер не блокирует отправку.



Если SMTP не настроен, backend может работать, но восстановление пароля, почтовые уведомления и системные письма будут недоступны.

Предупреждения вида:

SMTP Login is not configured properly

SMTP Password is not configured properly



не блокируют запуск backend, но означают, что отправка email не будет работать.

55. Не отображается версия клиента или не работает автообновление S-INFO Desktop

Если в health-статусе не отображается версия клиента, автообновление не работает или клиент не получает новую версию после обновления backend, вероятные причины:

- файлы S-INFO Desktop не загружены на сервер;
- загружена не та версия клиента;
- загружена не распакованная версия клиента;
- в папке клиента отсутствует S_Info.exe;
- команда `ConsoleTools run upgrade` завершилась с ошибкой;
- у backend нет прав на запись в Data;
- клиент подключается не к тому адресу сервера;
- версия backend и версия клиента не совпадают.

Загрузите распакованный клиент через

`ConsoleTools`

:

```
cd /home/sinfo-admin/backend/ConsoleTools
```

```
./ConsoleTools run upgrade /home/sinfo-admin/temp_client
```

После этого проверьте health-статус:

<https://<DOMAIN>/api/>



В ответе должны корректно отображаться версия backend и версия клиента.

56. Кнопка «Войти» в клиенте не активна



Обычно это означает, что клиент не видит сервер.

Проверьте:

- backend запущен;
- health-статус открывается в браузере;
- адрес сервера введен правильно;
- используется правильная схема http или https;
- порт и домен доступны с клиентского компьютера;
- Nginx настроен корректно;
- сертификат доверен клиентскому компьютеру;
- межсетевой экран не блокирует соединение.

Проверочный адрес:

`https://<DOMAIN>/api/`



Он должен открывать health-статус backend.

57. Ошибка Unable to load shared library 'libzstd'

Ошибка вида:

Unable to load shared library 'libzstd'

libzstd.so: cannot open shared object file



означает, что в системе отсутствует библиотека Zstandard.

Для Ubuntu:

```
sudo apt install -y libzstd1 libzstd-dev zstd
```



После установки повторите операцию.



Для ALT Linux, Astra Linux и других дистрибутивов пакет может называться иначе. Установите системный пакет Zstandard через пакетный менеджер ОС.

Проверка:

```
ldconfig -p | grep zstd
```

58. Ошибки Globalization, CultureInfo, ICU или Invariant mode



Такие ошибки связаны с локализацией .NET и системными локалями Linux.

Для Ubuntu 24.04:

```
sudo apt install -y libicu74 locales
```

```
sudo locale-gen ru_RU.UTF-8
```

```
sudo locale-gen en_US.UTF-8
```

```
sudo update-locale LANG=ru_RU.UTF-8
```

Проверки:

```
locale
```

```
locale -a | grep -Ei 'ru_RU|en_US'
```

```
ldconfig -p | grep icu
```



На других Linux-дистрибутивах имя ICU-пакета может отличаться. Нужно установить доступный runtime-пакет ICU и пакет локалей.



Отключать локализацию .NET через invariant-режим не рекомендуется. Это допустимо только как временная диагностическая мера.

59. Ошибки No usable version of libssl was found



Такая ошибка означает, что приложению или одной из native-библиотек не хватает совместимой версии OpenSSL.

Для Ubuntu 24.04 обычно должен использоваться актуальный пакет:

```
sudo apt install -y libssl3 ca-certificates
```



Для других версий Linux установите штатный OpenSSL-пакет из репозитория вашей ОС.



Не рекомендуется без необходимости устанавливать старые пакеты libssl1.0 или libssl1.1 из архивных репозиториев. Это может создать проблемы безопасности и совместимости. Такой вариант допустим только как временная мера по согласованию с администратором инфраструктуры и при понимании рисков.

60. Ошибки отчетов, PDF или шрифтов



Если отчеты не создаются, PDF формируется некорректно, пропадает кириллица или используются неправильные шрифты, проверьте системные библиотеки для отчетов и шрифтов.

Для Ubuntu:

```
sudo apt install -y \  
    fontconfig \  
    libfontconfig1 \  
    libfreetype6 \  
    libharfbuzz0b \  
    libfribidi0 \  
    libgdiplus \  
    libcairo2 \  
    libpango-1.0-0 \  
    libpangocairo-1.0-0 \  
    fonts-dejavu-core \  
    fonts-dejavu-extra \  
    fonts-liberation \  
    fonts-noto-core \  
    fonts-noto-extra \  
    fonts-noto-cjk
```

Проверьте шрифты:

```
fc-match "DejaVu Sans"  
fc-match "Liberation Sans"  
fc-match "Noto Sans"
```

```
sudo fc-cache -fv
```



Если в шаблонах отчетов используются корпоративные шрифты, их нужно установить на сервер отдельно.



На Linux отсутствие нужного шрифта может привести к автоматической замене шрифта. Из-за этого отчет может визуально отличаться от отчета, сформированного на Windows.

61. Не открывается S-INFO WEB

Проверьте:

- файлы веб-клиента скопированы в правильную папку;
- в папке есть index.html;
- Nginx имеет права на чтение файлов;
- в Nginx указан правильный root;
- config.json содержит правильный адрес backend;
- backend доступен по адресу из config.json;
- в браузере нет ошибок CORS;
- браузер не использует старую версию файлов из кэша.

Проверка прав:

```
sudo -u www-data test -r /var/www/sinfo-web/index.html && echo "OK"
```

Пример

config.json,

если

backend

опубликован на том же домене по

/api/

:

```
{  
  "BackendUrl": "/api/",  
  "AxiosRateLimit": null,  
  "UseAxiosCache": false,  
  "HideWarningLoginPageNotServerSet": false,  
  "DisablePrefixMergeProperties": false  
}
```



Если после обновления веб-клиента браузер продолжает показывать старую версию, очистите кэш или откройте страницу в приватном окне.

62. Ошибки CORS в S-INFO WEB



Если S-INFO WEB открывается, но не может обращаться к backend, а в браузере видны ошибки CORS, проверьте настройку адреса веб-клиента в `appsettings.Production.json`.

Если S-INFO WEB открыт по адресу:

`https://<DOMAIN>/`

то в backend должно быть указано:

```
"Clients": {  
  "WebApplication": {  
    "Url": "https://<DOMAIN>"  
  }  
}
```

После изменения

`appsettings.Production.json`

перезапустите backend:

```
sudo systemctl restart BackendWebService.service
```

63. Старый index.html, config.json или service worker после обновления S-INFO WEB

Для SPA-приложения важно не кэшировать критические файлы:

- /;
- /index.html;
- /config.json;
- /sw.js, если используется service worker;
- /manifest.webmanifest, если используется.

В

Nginx для index.html

и

config.json

рекомендуется отключить долговременное кэширование:

```
location = /index.html {  
    add_header Cache-Control "no-cache, must-revalidate" always;  
    etag on;  
    try_files $uri =404;  
}
```

```
location = /config.json {  
    add_header Cache-Control "no-cache, must-revalidate" always;  
    etag on;  
    try_files $uri =404;  
}
```



Для файлов с хэшами в имени, например js, css, woff2, можно использовать ограниченное кэширование.

После обновления веб-клиента проверьте страницу в приватном окне или очистите кэш браузера.

64. После обновления S-INFO WEB потерялись настройки



При обновлении веб-клиента нельзя бездумно перезаписывать локальные конфигурационные файлы.

Обычно нужно сохранять:

`config.json`

Если веб-клиент размещается через IIS, также может потребоваться сохранить:

`web.config`

Для Nginx пример копирования с сохранением config.json:

```
sudo rsync -a --delete \  
  --exclude config.json \  
  /home/sinfo-admin/temp_web/ \  
  /var/www/sinfo-web/
```

65. При обновлении backend потерялись настройки



При обновлении backend нельзя перезаписывать локальные данные и конфигурации.

Перед обновлением необходимо сохранить:

- базу данных PostgreSQL;
- папку Data или другую папку из Database:DataFolder;
- appsettings.Production.json;
- папку configs;
- server_license.json;
- web.config, если используется IIS;
- журналы, если они нужны для диагностики.



Если эти файлы были перезаписаны, потребуется повторно настраивать подключение к базе, почту, фоновые задачи, лицензию и другие параметры.

66. Ошибки при резервном копировании базы данных



Резервное копирование базы данных использует стандартные утилиты PostgreSQL.

Проверьте наличие:

```
pg_dump --version
```

```
pg_restore --version
```

Если утилит нет:

```
sudo apt install -y postgresql-client
```



Если резервное копирование требует пароль и не может выполняться автоматически, настройте `.pgpass` для пользователя, от имени которого выполняется резервное копирование.

Права на `.pgpass` должны быть строгими:

```
chmod 0600 ~/.pgpass
```



Если PostgreSQL вынесен на отдельный сервер, проверьте доступность базы по сети и совместимость версии `pg_dump` с версией PostgreSQL-сервера.

67. Не выполняются фоновые задачи

Проверьте файл:

```
/home/sinfo-admin/backend/configs/systemScheduler.settings.json
```

Проверьте, что системные задачи включены:

```
"EnabledSystemJobs": true
```

Проверьте журнал backend:

```
tail -n 200 /home/sinfo-admin/backend/logs/sinfo_log_*.log
```

Фоновые задачи выполняются по времени сервера. Проверьте часовой пояс:

```
timedatectl
```



Если нужно проверить задачу без ожидания расписания, запустите ее вручную из панели администратора S-INFO, если такой пункт доступен в используемой версии.

68. Недостаточно места на диске



Нехватка места может привести к ошибкам загрузки файлов, резервного копирования, журналов, PostgreSQL и временных операций.

Проверьте:

```
df -h
```

```
du -sh /home/sinfo-admin/backend/Data
```

```
du -sh /home/sinfo-admin/backend/logs
```



Также проверьте место, куда сохраняются резервные копии.

Если настроено резервное копирование файловой папки, учитывайте, что измененные и удаленные файлы могут временно храниться в папке-корзине.

69. Предупреждение HTTP/2 is not enabled for 127.0.0.1:5000



Если backend работает локально по HTTP за Nginx, такое предупреждение не является критической ошибкой.

Обычно схема такая:

Пользователь -> HTTPS Nginx -> HTTP localhost:5000 Backend



TLS завершается на Nginx, а backend внутри сервера работает по локальному HTTP. Это нормальная схема установки.

70. Предупреждение No XML encryptor configured



Такое предупреждение связано с хранением ключей ASP.NET Core Data Protection.

Для базовой установки оно обычно не блокирует запуск backend. При этом доступ к папке backend и configs должен быть ограничен, так как там находятся служебные и конфиденциальные файлы.

Проверьте права:

```
ls -la /home/sinfo-admin/backend/configs
```

71. Особенности ALT Linux, Astra Linux и других Linux-дистрибутивов

Основная инструкция дана на примере Ubuntu. На ALT Linux, Astra Linux и других системах общий порядок установки сохраняется, но могут отличаться:

названия пакетов;

- версия ICU;
- пакет локалей;
- пакет Zstandard;
- пакет OpenSSL;
- имя службы PostgreSQL или Postgres Pro;
- путь к конфигурации PostgreSQL;
- пакетный менеджер;
- политики безопасности и межсетевой экран;
- состав минимальной установки ОС.

Если пакет из инструкции не найден, это не означает, что backend не будет работать. Нужно установить ближайший аналог по назначению:

- ICU и локали;
- Fontconfig, FreeType, HarfBuzz и базовые шрифты;
- Zstandard;
- PostgreSQL client tools;
- Nginx;
- OpenSSL и системные сертификаты.

Примеры поиска пакетов в Debian/Ubuntu/Astra-подобных системах:

```
apt-cache search libicu
```

```
apt-cache search zstd
```

```
apt-cache search postgresql-client
```

```
apt-cache search fontconfig
```



В ALT Linux могут использоваться другие названия пакетов и команды установки. Ориентируйтесь на документацию конкретной версии ОС и устанавливайте пакеты по назначению, а не только по точному имени из Ubuntu.



Для текущей самодостаточной сборки S-INFO Backend на .NET 8 не требуется устанавливать .NET 3.1 или запускать backend через dotnet BackendWebService.dll.

Штатный запуск на Linux:

```
./BackendWebService
```

72. Общий порядок диагностики

Если проблема не описана выше:

1. Определите, где возникает ошибка: backend, Nginx, PostgreSQL, S-INFO WEB, S-INFO Desktop или сеть.
2. Откройте соответствующие журналы.
3. Найдите первую ошибку по времени, а не только последнюю строку.
4. Проверьте права доступа, свободное место, доступность PostgreSQL и корректность конфигурации.
5. Если ошибка относится к системной библиотеке Linux, найдите ее название в пакетах используемой ОС.
6. Если ошибка относится к Nginx, проверьте `nginx -t` и журналы Nginx.
7. Если ошибка относится к клиенту, проверьте адрес сервера, сертификат, версию клиента и клиентские логи.
8. После исправления перезапустите соответствующую службу и повторите проверку.

Основные команды:

```
sudo systemctl status BackendWebService.service
```

```
sudo journalctl -u BackendWebService.service -n 200 --no-pager
```

```
sudo systemctl status nginx
```

```
sudo nginx -t
```

```
sudo tail -n 200 /var/log/nginx/sinfo_error.log
```

```
sudo systemctl status postgresql
```

```
sudo journalctl -u postgresql -n 200 --no-pager
```

```
df -h
```

```
free -h
```



Если ошибка системная и не относится напрямую к S-INFO, полезно искать решение по точному тексту ошибки из журнала. Обычно сообщения о нехватке библиотек, проблемах доступа, сетевых блокировках или ошибках PostgreSQL достаточно информативны для первичной диагностики.

73. Тезаурус

Термин	Значение
Backend	Серверная часть S-INFO. Обрабатывает запросы клиентских приложений, работает с базой данных, файлами, лицензией и фоновыми задачами.
Серверная часть	То же, что Backend. В инструкции эти термины используются как равнозначные.
Серверное приложение	Исполняемое приложение S-INFO Backend, которое запускается на сервере Linux или Windows.
Сервер	Компьютер или виртуальная машина, на которой размещается серверная часть S-INFO.
Клиентское приложение	Приложение, которое подключается к серверной части. К серверу подключаются S-INFO Desktop и S-INFO WEB.
S-INFO Desktop	Настольное клиентское приложение S-INFO.
S-INFO WEB	Веб-клиент S-INFO, работающий через браузер.
PostgreSQL	Система управления базой данных, в которой S-INFO хранит основные данные.
База данных	Хранилище основных данных S-INFO: пользователей, проектов, настроек, связей, записей и другой прикладной информации.
Папка хранения файлов	Папка на диске, в которой S-INFO хранит загруженные документы, модели, файлы проектов и другие пользовательские данные. По умолчанию используется папка Data.
Конфигурация	Настройки серверной части. Основной файл настроек — appsettings.Production.json. Дополнительные настройки хранятся в папке configs.
Лицензия	Файл server_license.json, необходимый для запуска серверной части.

Служба	Способ запуска серверной части в операционной системе. На Linux используется systemd, на Windows запуск обычно выполняется через IIS.
Nginx	Веб-сервер, который на Linux обычно принимает внешние HTTP/HTTPS-запросы и передает их в S-INFO Backend.
IIS	Веб-сервер Windows, через который размещается серверная часть S-INFO на Windows Server.
Reverse proxy	Схема, при которой Nginx или IIS принимает внешние запросы и передает их во внутренний backend-сервис.
Health-статус	Служебный ответ backend-сервера, по которому можно проверить, что серверная часть запущена и доступна.
SMTP	Настройки почтового сервера, через который S-INFO отправляет уведомления и письма для восстановления пароля.
Файлы клиента	Файлы S-INFO Desktop, которые загружаются на сервер через консольную утилиту и используются для автообновления клиентского приложения.

Таблица 14