

eXpress

Система
коммуникаций

Outlook-плагин

Руководство по установке

Версия 2.0.0.0



© Компания «Анлимитед продакшен», 2024. Все права защищены.

Все авторские права на эксплуатационную документацию защищены.

Без специального письменного разрешения компании «Анлимитед продакшен» этот документ или его часть в печатном или электронном виде не могут быть подвергнуты копированию или передаче третьим лицам с коммерческой целью.

Информация, содержащаяся в этом документе, может быть изменена разработчиком без специального уведомления, что не является нарушением обязательств по отношению к пользователю со стороны компании «Анлимитед продакшен».

Почтовый адрес:	127030, г. Москва, ул. Новослободская, д. 24, стр. 1
Телефон:	+7 (499) 288-01-22
E-mail:	sales@express.ms
Web:	https://express.ms/

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	3
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	4
ГЛАВА 1	
ОБЩИЕ ТРЕБОВАНИЯ.....	5
ГЛАВА 2	
АРХИТЕКТУРА	7
Single CTS или разделенный CTS	7
Несколько CTS без ETS	8
Несколько CTS с ETS	9
ГЛАВА 3	
УСТАНОВКА И НАСТРОЙКА ЧАТ-БОТА CONFERENCE NOTIFIER BOT	10
Шаг 1. Включение API Conference Notifier Bot без пароля	10
Шаг 2. Защита Conference Notifier Bot	11
ГЛАВА 4	
УСТАНОВКА OUTLOOK-ПЛАГИНА.....	13
Шаг 1. Разворачивание в IIS express-core-service	13
Шаг 2. Настройка записи логов серверной части плагина и клиентских частей.....	15
Шаг 3. Разворачивание Outlook-плагина	15
Шаг 4. Разворачивание в IIS express-template-service.....	16
Шаг 5. Настройка записи логов express-template-service	19
ГЛАВА 5	
ДИАГНОСТИКА НЕИСПРАВНОСТЕЙ ПЛАГИНА OUTLOOK.....	20
Шаг 1. Проверка работоспособности Conference Notifier Bot	20
Шаг 2. Проверка доступности Conference Notifier Bot с сервера IIS	21
Шаг 3. Проверка доступности сервера IIS с клиентской рабочей станции.....	21
Шаг 4. Общая диагностика плагина Outlook	21
ПРИЛОЖЕНИЕ 1	
ДИАГНОСТИЧЕСКИЙ СКРИПТ № 1	22
ПРИЛОЖЕНИЕ 2	
ДИАГНОСТИЧЕСКИЙ СКРИПТ № 2	23

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Термин	Определение
AD	Active Directory — служба каталогов корпорации Microsoft для операционных систем семейства Windows Server
API	Application Programming interface — интерфейс для взаимодействия программ и приложений
CTS	Corporate Transport Server — корпоративный сервер
ETS	Enterprise Transport Server — сервер предприятия
JSON	Текстовый формат обмена данными, основанный на JavaScript
Single CTS	Единый корпоративный сервер
БД	База данных
ПК	Персональный компьютер
ПО	Программное обеспечение
Разделенный CTS	Разделенный корпоративный сервер: Front CTS и Back CTS
СК «Express», Express, система	Система коммуникаций «Express»
Чат-бот	Чат-бот Conference Notifier Bot
express-core-service	Серверное приложение для Outlook-плагины, обеспечивающее основную функциональность (создание и изменение конференций)
express-template-service	Серверное приложение для Outlook-плагины, обеспечивающее дополнительную функциональность (создание и изменение шаблонов письма, текст, логотип, подпись и т. п.)
Лог	Запись в журнале событий сервера и/или клиента

Глава 1

ОБЩИЕ ТРЕБОВАНИЯ

Внимание! Для выполнения операций из настоящей инструкции необходимо обладать следующими компетенциями:

- администрирование Windows Server;
- администрирование IIS;
- администрирование PostgreSQL;
- администрирование Linux;
- администрирование eXpress;
- понимание JSON.

Перед началом работ убедитесь, что программные и технические средства соответствуют следующим требованиям:

- версия клиентской части Outlook должна быть не ниже 2013;
- если на виртуальной машине не будет других сервисов, то сервер IIS должен соответствовать следующим техническим требованиям:

Элемент	Параметры
Процессор	64-bit 1.4 GHz processor 2 core
Оперативная память	2 ГБ
Операционная система	Не ниже Windows Server 2016
Жесткий диск	Не менее 40 ГБ

- версия IIS должна быть не ниже 10.0 (при установке IIS поставьте отметку web-server → security → windows authentication);
- для бесшовной аутентификации сервер IIS должен быть введен в домен, и должны быть настроены соответствующие доступы.

Если на виртуальной машине не будет других сервисов, то сервер PostgreSQL должен соответствовать следующим техническим требованиям:

Элемент	Параметры
Процессор	64-bit 1.4 GHz processor 2 core
Оперативная память	2 ГБ
Операционная система	Debian GNU. Не ниже Linux 11
Жесткий диск	Не менее 40 ГБ

Все пользователи, которые хотят использовать Outlook-плагин, должны быть зарегистрированы в СК «Express», и быть активными. Проверить статус пользователя можно в консоли администратора СК «Express» – раздел «Пользователи» (Рисунок 1).

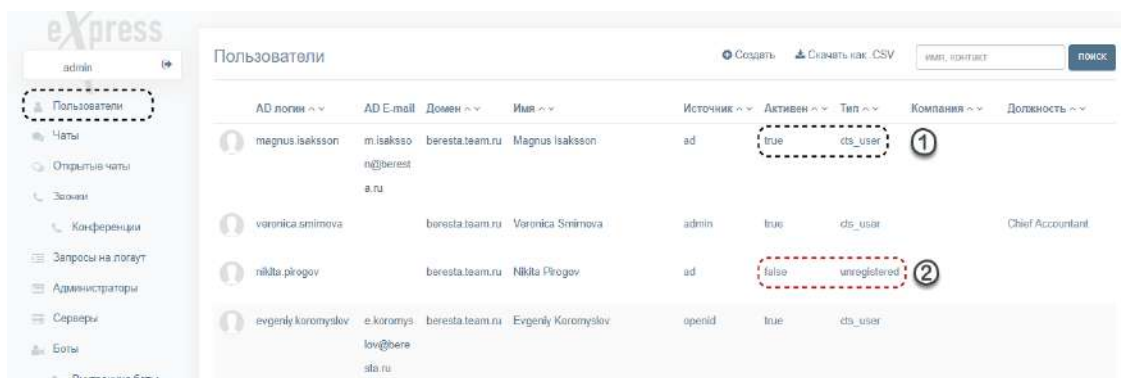


Рисунок 1. Статус пользователя: 1 – активный; 2 – неактивный

Требования к используемому ПО:

- Версия PostgreSQL должна быть не ниже 13.11 (Debian 13.11-0+deb11u1, Windows).
- Наличие супер-пользователя на сервере PostgreSQL.
- Должна быть создана отдельная БД PostgreSQL, отдельный пользователь БД PostgreSQL, данный пользователь должен быть владельцем это БД и иметь все права на нее.
- Outlook должен использоваться под доменной учетной записью, с почтовым ящиком, который соответствует адресу e-mail, прописанному в учетной записи AD.
- На IIS-сервере должен быть установлен пакет <https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-7.0.11-windows-hosting-bundle-installer>.
- Должно быть установлено и настроено решение СК «Express» не ниже версии 3.8.
- В СК «Express» должен быть настроен чат-бот Conference Notifier Bot.

Требования по настройке чат-бота и методы проверки корректности настройки описаны в Главе 3 «Установка и настройка чат-бота Conference Notifier Bot».

Глава 2

АРХИТЕКТУРА

В зависимости от используемого архитектурного решения СК «Express» применяются различные схемы развертывания серверной части плагина Outlook. Ниже описаны наиболее распространенные (типовые) архитектурные решения.

Конкретные схемы развертывания определяются требованиями заказчика после консультации с компанией-разработчиком.

Если планируется использовать единый шаблон письма, то все express-core-service могут использовать одну БД, если разные – для разных express-core-service, – то следует развернуть для каждой из них свою БД и свой express-template-service.

При использовании более одного экземпляра express-core-service или express-template-service на одном IIS-сервере пути записи логов для каждого серверного приложения лучше сделать разными (для каждого серверного приложения должна быть своя папка логов).

SINGLE CTS ИЛИ РАЗДЕЛЕННЫЙ CTS

Если СК «Express» содержит один CTS-сервер, то серверную часть плагина следует развернуть в одном экземпляре ([Рисунок 2](#)).

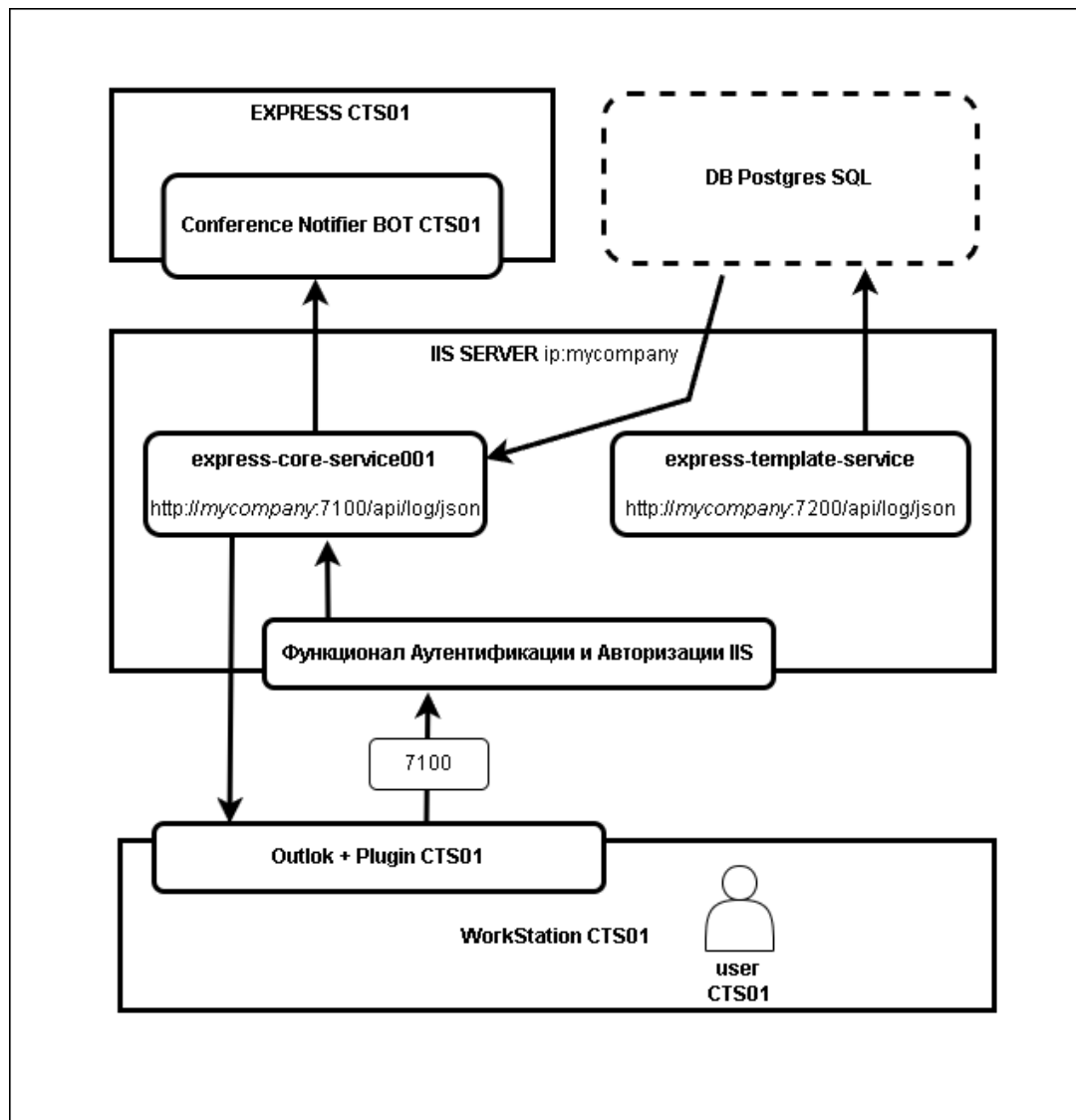


Рисунок 2. Типовая схема: 1 CTS, 1 express-core-service, 1 express-template-service

НЕСКОЛЬКО CTS БЕЗ ETS

Если СК «Express» содержит несколько CTS-серверов, без ETS-сервера, то серверную часть плагина следует развернуть для каждого CTS-сервера отдельно (можно на одном IIS, но с разными портами, можно на нескольких IIS) (Рисунок 3).

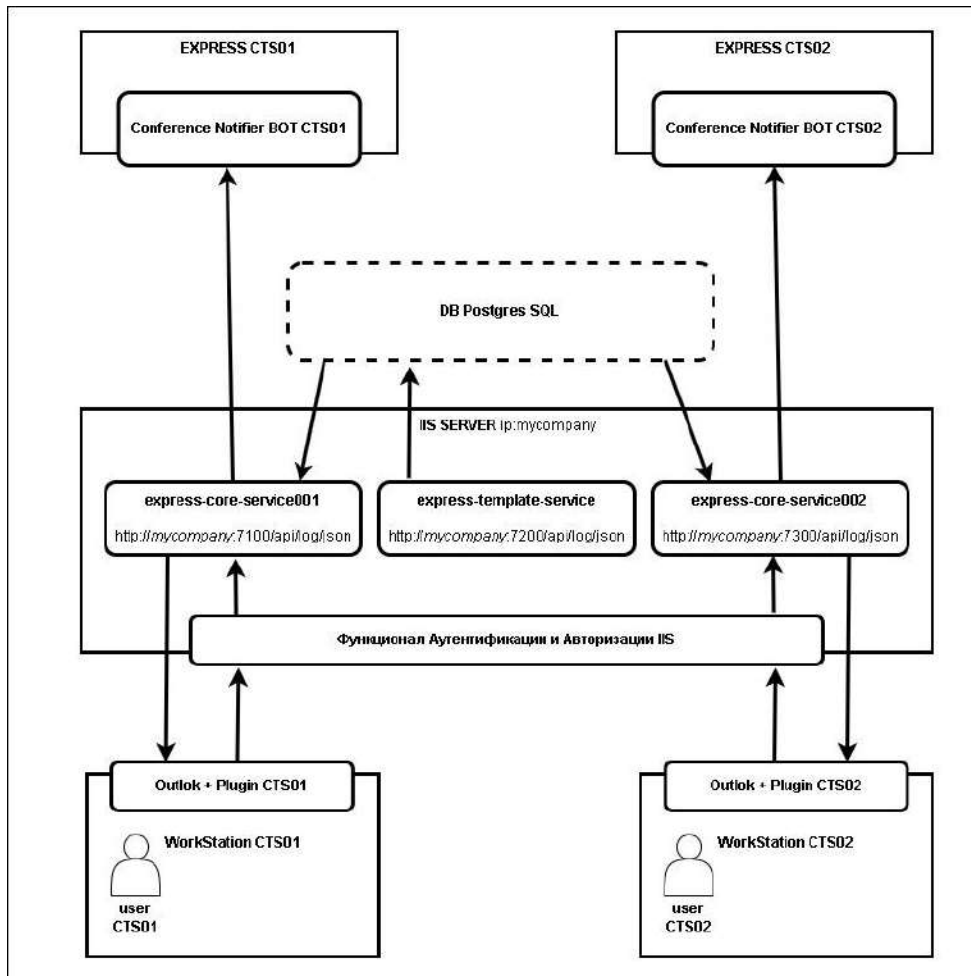


Рисунок 3. Типовая схема: 2 CTS, 2 `express-core-service`, 1 `express-template-service`

НЕСКОЛЬКО CTS С ETS

Если СК «Express» содержит несколько CTS-серверов, объединенных ETS-сервером (Рисунок 4), то серверную часть плагина следует развернуть только для ETS-сервера.

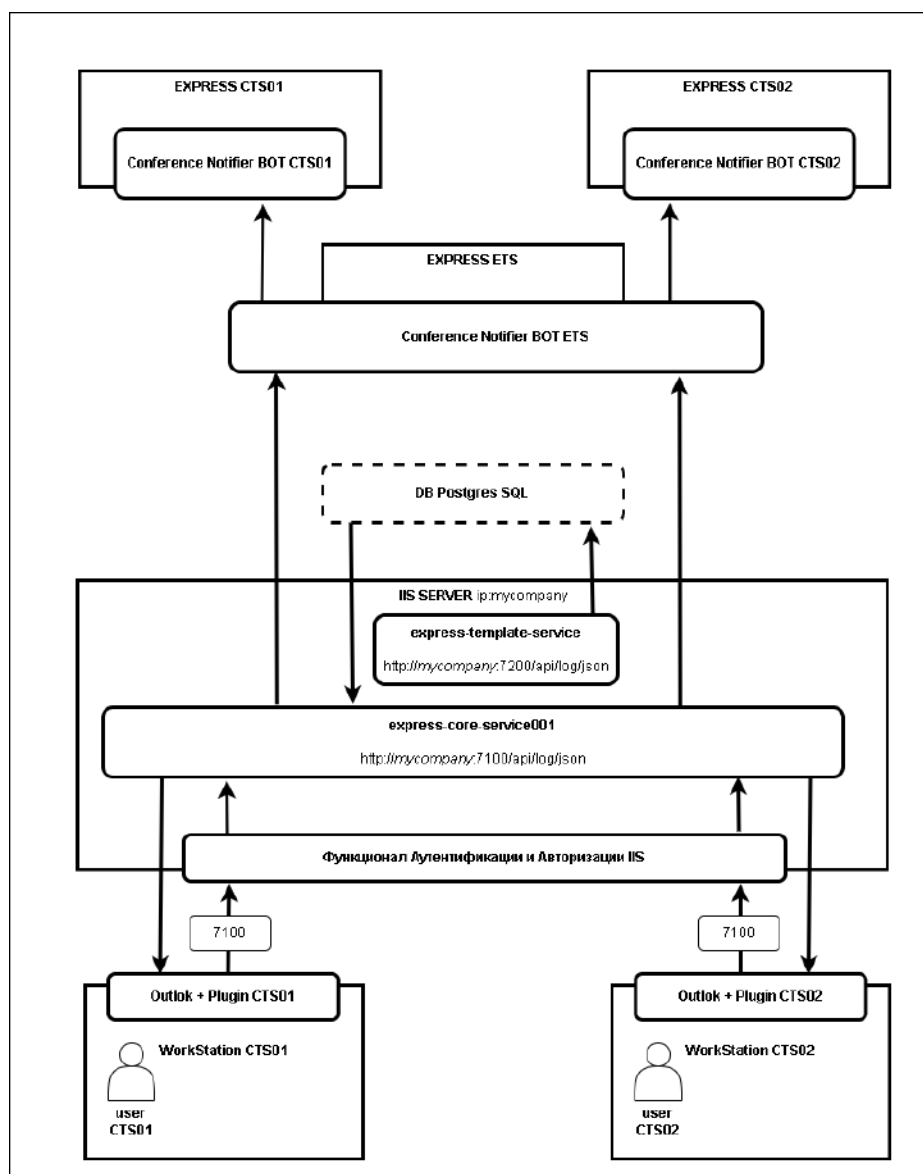


Рисунок 4. Типовая схема: 1 ETS, 2 CTS, 1 express-core-service, 1 express-template-service

Глава 3

УСТАНОВКА И НАСТРОЙКА ЧАТ-БОТА CONFERENCE NOTIFIER BOT

В разделе описывается порядок подключения API чат-бота Conference Notifier Bot на CTS-сервере программного решения СК «Express».

Примечание. Процедура описана для ОС Debian 11.4 и Ubuntu 2X. При возникновении проблем во время настройки чат-бота на других ОС рекомендуем обратиться к компании-разработчику.

ШАГ 1. ВКЛЮЧЕНИЕ API CONFERENCE NOTIFIER BOT БЕЗ ПАРОЛЯ

Внимание! При настройке чат-бота без защиты его может использовать любой пользователь из внешнего контура если он открыт.

Важно:

- Если СК «Express» содержит несколько CTS, то данную процедуру следует выполнить на каждом CTS, который будет взаимодействовать с express-core-service.
- Если СК «Express» содержит несколько один или несколько ETS, то данную процедуру следует выполнить на каждом ETS и CTS, который будет взаимодействовать с express-core-service.

Для включения API Conference Notifier Bot без пароля:

1. На CTS-сервере добавьте в файл settings.yaml¹ следующий код:

```
conference_bot_env_override:
  API_ENABLED: true
```

В случае использования разделенной серверной архитектуры изменения вносятся только в файл на сервер Back CTS.

2. Сохраните файл и запустите из /OPT/EXPRESS:

```
dpl -d conference_bot
```

3. Откройте веб-консоль администратора и перейдите в раздел «Боты → Внутренние боты → Conference Notifier Bot».

Откроется окно «Редактирование бота».

4. В открывшемся окне проставьте отметку в поле «Включено» и нажмите кнопку «Сохранить», при необходимости добавьте порт 4000.

5. Проверьте доступность чат-бота с помощью запроса через CURL, например:

```
curl -X POST -H "Content-Type: application/json" -d
'{"name":"test_plugin_001","members":["user001@mydomain.com",
"user002@mydomain.com"],"admins":["user001@mydomain.com
"],"creator":"user001@mydomain.com","start_at":"2023-12-
21T10:10:00.0Z","end_at":"2023-12-
21T11:10:00.0Z","link":{"link_type":"public","access_code":null}}'
https://my-CTS-domain.com/api/v1/conference_bot/conferences/
```

Примечание. Красным цветом отмечены параметры, которые требуется актуализировать под конкретный запрос.

Если чат-бот доступен, CURL выведет:

```
"status":"ok"
```

6. В веб-интерфейсе администратора CTS-сервера перейдите в раздел «Звонки → Конференции».

¹ Предполагаемый путь /opt/express/settings.yaml.

7. В поле поиска введите `test_plugin_001` (был задан в п. 5) и нажмите кнопку «Поиск».

Если в таблице будет выведена конференция с именем `test_plugin_001`, то чат-бот настроен правильно.

Для проверки статуса докер-контейнера после установки бота на сервере Express введите команду:

```
docker ps
```

Контейнер `conference_bot` должен иметь статусы «Up» и «healthy».

Внимание! Версии контейнеров `conference_bot` и `messaging` должны совпадать, если не совпадают, то следует обновить Express полностью.

ШАГ 2. ЗАЩИТА CONFERENCE NOTIFIER BOT

Важно:

- Если СК «Express» содержит несколько CTS, то данную процедуру следует выполнить на каждом CTS, который будет взаимодействовать с `express-core-service`.
- Если СК «Express» содержит один объединенный ETS, то данную процедуру следует выполнить только на нем.

Для защиты Conference Notifier Bot выполните установку bearer token:

1. Убедитесь, что установлена СК «Express» не ниже версии 3.8. Если ниже, то выполните обновление Express до последней версии.
2. Убедитесь, что версия дистрибутива «Плагин Outlook» (серверная и клиентская) не ниже 1.2.0.0.

Важно! Введенные учетные данные будут использоваться при настройке Outlook-плагина.

3. На CTS-сервере добавьте в файл `settings.yaml`² в секцию `conference_bot_env_override`: следующий код:

```
API_AUTH_METHOD: BEARER_TOKEN
BEARER_TOKEN: 3bDewf52b3268sdg59f1f7fff33w01dd3c0431
```

После чего секция `conference_bot_env_override` должна принять вид:

```
conference_bot_env_override:
  API_ENABLED: true
  API_AUTH_METHOD: BEARER_TOKEN
  BEARER_TOKEN: 3bDewf52b3268sdg59f1f7fff33w01dd3c0431
```

В случае использования разделенной серверной архитектуры изменения вносятся только в файл на сервер Back CTS.

Значение токена придумайте самостоятельно или сгенерируйте любым сторонним ПО (он должен быть не короче 40 символов, содержать большие и маленькие латинские буквы и цифры).

4. Сохраните файл и запустите из `/OPT/EXPRESS`:

```
dp1 -d conference_bot
```

5. Проверьте недоступность чат-бота с помощью запроса без `bearer token` через CURL, например:

² Предполагаемый путь `/opt/express/settings.yaml`

```
curl -X POST -H "Content-Type: application/json" -d
'{"name":"test_plugin_001","members":["user001@mydomain.com",
"user002@mydomain.com"],"admins":["user001@mydomain.com
"],"creator":"user001@mydomain.com","start_at":"2023-12-
21T10:10:00.0Z","end_at":"2023-12-
21T11:10:00.0Z","link":{"link_type":"public","access_code":null}}'
https://my-CTS-domain.com/api/v1/conference_bot/conferences/
```

Примечание. Красным цветом отмечены параметры, которые требуется актуализировать под конкретный запрос.

Если чат-бот без bearer token недоступен (это так и должно быть), CURL выведет:

```
Пустую строку
```

6. Проверьте доступность чат-бота с помощью запроса с bearer token через CURL, например:

```
curl -X POST -H "Content-Type: application/json" -H "Authorization:
Bearer 3bfef52b32685srdsrherhFGdlf73301dd3c0431" -d
'{"name":"test_plugin_001","members":["user001@mydomain.com",
"user002@mydomain.com"],"admins":["user001@mydomain.com
"],"creator":"user001@mydomain.com","start_at":"2023-12-
21T10:10:00.0Z","end_at":"2023-12-
21T11:10:00.0Z","link":{"link_type":"public","access_code":null}}'
https://my-CTS-domain.com/api/v1/conference_bot/conferences/
```

Примечание. Красным цветом отмечены параметры, которые требуется актуализировать под конкретный запрос.

Если чат-бот доступен, CURL выведет:

```
"status":"ok"
```

7. В веб-интерфейсе администратора CTS-сервера перейдите в раздел «Звонки → Конференции».
8. В поле поиска введите `test_plugin_001` (был задан в п. 5) и нажмите кнопку «Поиск».

Если в таблице будет выведена конференция с именем `test_plugin_001`, то чат-бот настроен правильно.

Глава 4

УСТАНОВКА OUTLOOK-ПЛАГИНА

Для установки Outlook-плагина выполните шаги, описанные ниже.

ШАГ 1. РАЗВОРАЧИВАНИЕ В IIS EXPRESS-CORE-SERVICE

Если сервис уже был развернут в рамках предыдущей версии плагина, то:

- убедитесь, что учетной записи, под которой будет производиться обновление, выданы все необходимые разрешения на файловую систему;
- замените файлы сервиса файлами из дистрибутива из папки IIS\core-service (кроме конфигурационного файла appsettings.Test.json. Этот файл заменять не нужно! Иначе придется вновь выполнить настройку конфигурации, описанную в п.7 текущего шага);
- переименуйте сохраненный файл appsettings.Test.json в appsettings.Production.json;
- на IIS-сервере замените значение параметра ASPNETCORE_ENVIRONMENT на Production (установка значения описана в п.6 текущего шага);
- выполните п.9 текущего шага для обновления структуры БД.

Для разворачивания в IIS express-core-service:

1. Создайте в IIS AppPool (Рисунок 5):

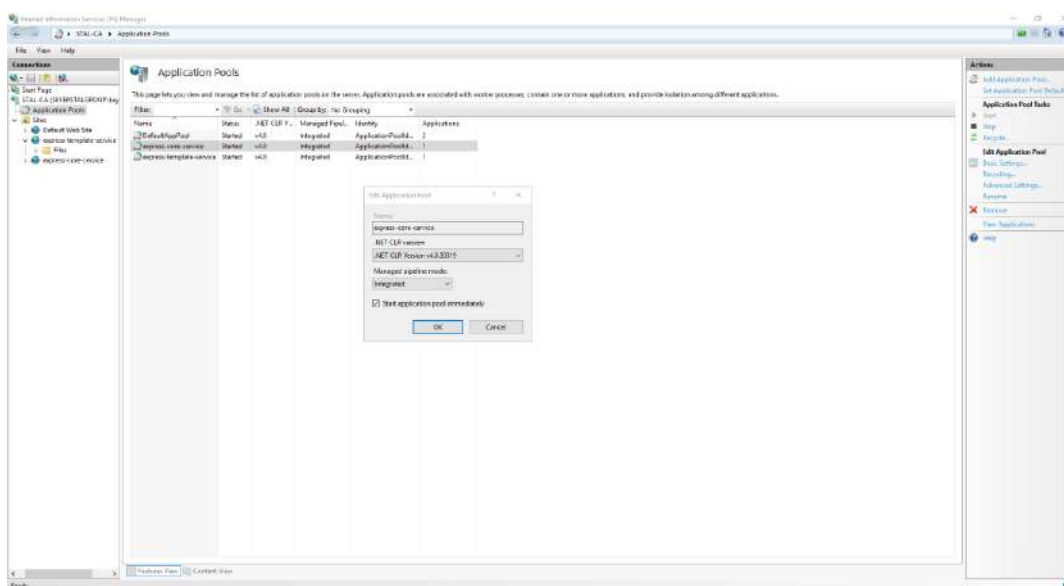


Рисунок 5

2. Создайте сайт express-core-service.

При создании в поле PhysicalPath должен быть указан путь к папке IIS\core-service.

3. В поле Application pool выберите ранее созданный пул.
4. В поле Port укажите порт, по которому будет осуществляться связь клиента с сервером IIS.
5. Перейдите в «Главный узел» → Feature Delegation, Custom Site Delegation». В открывшемся окне для сайта express-core-service выдайте права на чтение и запись (read/write) для параметров Authentication – Windows и Authentication – Anonymous (Рисунок 6).

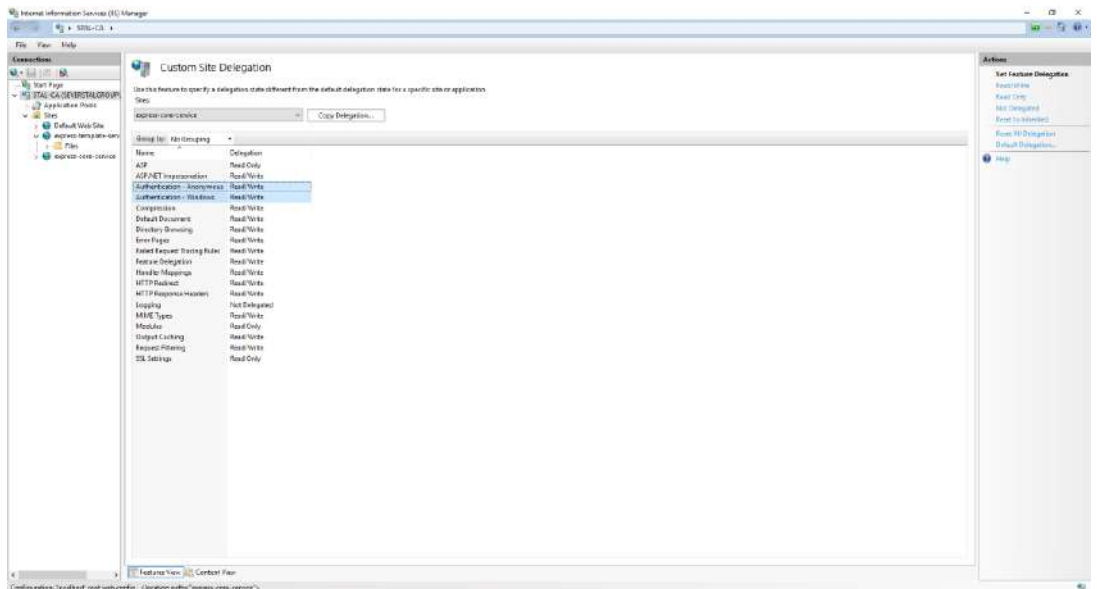


Рисунок 6

- Для узла сайта express-core-service через Configuration Editor выставьте среду выполнения system.webServer/aspNetCore → environmentVariables From ApplicationHost.config (Рисунок 7). Задайте соответствующие параметры:

```
name: ASPNETCORE_ENVIRONMENT
value: Production
```

затем закройте окно environmentVariables, и в Configuration Editor нажмите apply:

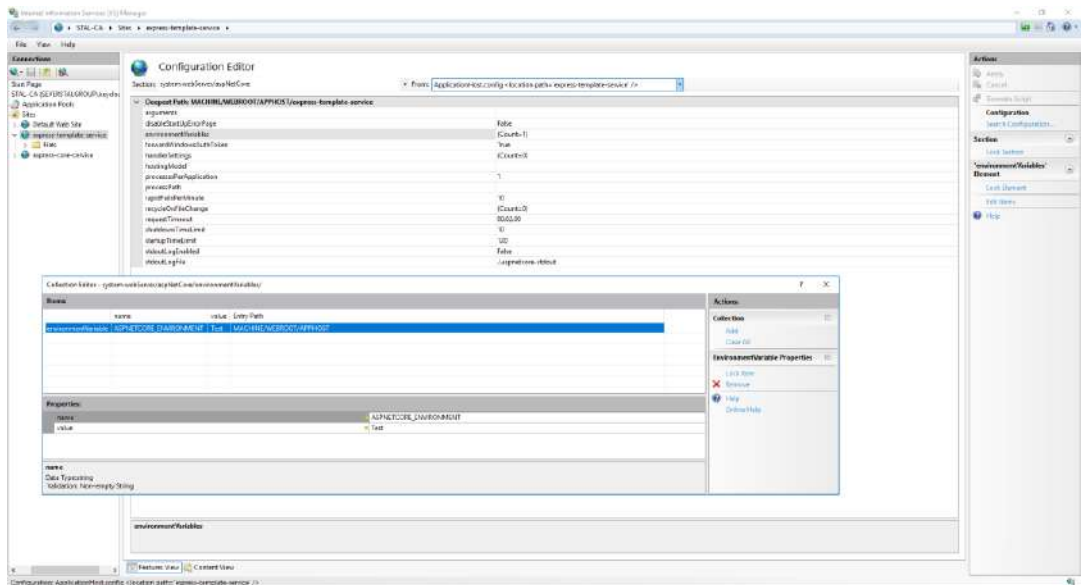


Рисунок 7

- Отредактируйте в файле appsettings.Production.json³ параметры среды:
 - установите адрес Express Conferences Bot API в ключе:


```
ExpressOptions.Uri="https://my-CTS-domain.com/api/v1/Conference_bot/conferences/"
```
 - тип аутентификации, который будет использоваться при обращении к сервису Express Conferences Bot API в ключе:


```
ExpressOptions.AuthenticationScheme="StaticToken"
```

³ Предполагаемый путь IIS\core-service\appsettings.Production.json.

Примечание. StaticToken – штатный и рекомендуемый метод аутентификации по статическому токenu.

Если данный параметр принимает значения: Basic – аутентификация производится по логину/паролю (данный метод является нештатным использовался для версий ниже 1.2.0, в версиях выше к использованию не рекомендуется);

Если Заказчику по каким-то причинам требуется именно Basic – аутентификация рекомендуем обратиться к компании-разработчику для консультаций.

В ключе AuthenticationScheme .StaticToken, установите значение токена, который будет использован при обращении к сервису ExpressConferencesBotAPI в ключе:

```
ExpressOptions.Token="Bearer dfgdf509usds0f9n0ud"
```

Примечание. Данный параметр должен соответствовать значению bearer token, который был задан при настройке чат-бота. Если при настройке чат-бота аутентификация не настраивалась, оставьте параметры конфигурационного файла как есть.

- установите строку подключения к БД PostgreSQL в ключе:

```
DatabaseOptions.Connection="Host=127.0.0.1;Port=5432;Database=express_meeting_db;Username=db_express;Password=Pass1234;Pooling=true;Minimum Pool Size=50;Maximum Pool Size=100;Include Error Detail=True;"
```

Примечание. Параметры Username=db_express;Password=Pass1234 должны соответствовать учетным данным, которые были созданы при настройке PostgreSQL.

8. Выполните

```
iisreset
```

9. Выполните инициализацию БД PostgreSQL с шаблонами конференций. В командной строке перейдите в папку, в которой находится express-core-service, и выполните:

```
dotnet ExpressMeeting.CoreService.dll --environment Production --ef-migrate
```

ШАГ 2. НАСТРОЙКА ЗАПИСИ ЛОГОВ СЕРВЕРНОЙ ЧАСТИ ПЛАГИНА И КЛИЕНТСКИХ ЧАСТЕЙ

Для настройки записи логов серверной части плагина и клиентских частей:

1. Выдайте права FullControl для Everyone на сервере, где установлен cores-service, для папки c:/logs/core-service/⁴ и ее содержимого.
2. При необходимости поменяйте путь к папке логов (см. сноску к п.1 раздела «Шаг 2. Настройка записи логов серверной части плагина и клиентских частей»).

ШАГ 3. РАЗВОРАЧИВАНИЕ OUTLOOK-ПЛАГИНА

Если сервис уже был развернут в рамках предыдущей версии плагина, то удалите предыдущую версию клиентского ПО на клиентских машинах и выполните установку новой версии.

⁴ Папка предназначена для хранения логов серверной и клиентской частей. Путь к папке прописан в конфигурационном файле appsettings.Production.json, в ключе Serilog.WriteTo[Name = "Logger"].Args.configureLogger.WriteTo[Name="File"].Args.path.

Для разворачивания Outlook-плагина:

1. Запустите инсталлятор ExpressMeeting.X.Y.Z.N.msi на машине, на которой установлен Outlook (X.Y.Z.N – версия релиза).
2. После инсталляции в файле ExpressMeeting.dll.config⁵ установите адрес сервиса серверной части плагина в ключе (замените my-IIS-domain.com на ip/dns-имя машины, где развернут express-core-service):

```
<add key="ExpressMeetingUrl" value="http://my-IIS-domain.com:7100/api/" />
```

Примечание. Параметр можно автоматически проинициализировать при установке msi-пакета. Для этого в командной строке необходимо передать параметр ExpressMeetingUrl.

Пример: ExpressMeeting_v_2.0.6.3.msi ExpressMeetingUrl="http://IIS-name.ru:7100/api/"

3. При необходимости в том же файле ExpressMeeting.dll.config установите режим запрета выбора типа конференции. Выпадающий список с выбором типа (Общее/Корпоративное/Доверенное) будет скрыт в настройках встречи. Всегда будут создаваться общие встречи. Для этого добавьте в секцию configuration/appSettings параметр:

```
<add key="DenyLinkType" value="true" />
```

Примечание. Параметр можно автоматически проинициализировать при установке msi-пакета. Для этого в командной строке необходимо передать параметр DenyLinkType.

Пример: ExpressMeeting_v_2.0.6.3.msi DenyLinkType=true

4. В файле serilogSettings.json⁶ установите адрес сервиса серверной части плагина в ключе (замените my-IIS-domain.com на ip/dns-имя машины, где развернут express-core-service):

```
Serilog.WriteTo[Name=Telemetry].Args.telemetryUrl="http://my-IIS-domain.com:7100/api/log/json"
```

Примечание. Если при установке msi-пакета в командной строке был передан параметр ExpressMeetingUrl, то данный параметр установится автоматически.

ШАГ 4. РАЗВОРАЧИВАНИЕ В IIS EXPRESS-TEMPLATE-SERVICE

Если сервис уже был развернут в рамках предыдущей версии плагина, и было выполнено обновление согласно «Шаг 1. Разворачивание в IIS express-core-service», то:

- убедитесь, что учетной записи, под которой будет производиться обновление, выданы все необходимые разрешения на файловую систему;
- замените файлы сервиса файлами из дистрибутива из папки IIS\template-service (кроме конфигурационного файла appsettings.Test.json. Этот файл заменять не нужно! Иначе придется вновь выполнить настройку конфигурации, описанную в п.7 текущего шага);
- переименуйте сохраненный файл appsettings.Test.json в appsettings.Production.json;
- на IIS-сервере замените значение параметра ASPNETCORE_ENVIRONMENT на Production (установка значения описана в п.6 текущего шага);

⁵ Предполагаемый путь c:\Program Files\Express\ExpressMeeting\ExpressMeeting.dll.config.

⁶ Предполагаемый путь c:\Program Files\Express\ExpressMeeting\serilogSettings.json.

Для разворачивания в IIS express-template-service:

1. Создайте в IIS AppPool (Рисунок 8):

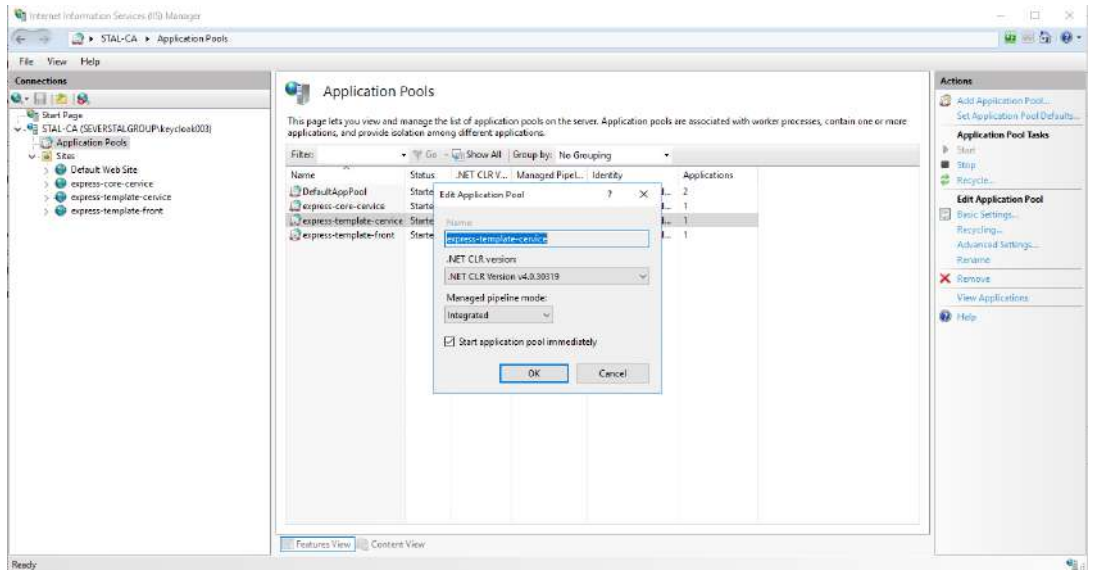


Рисунок 8

2. Создайте сайт express-template-service.

При создании в поле PhysicalPath должен быть указан путь к папке IIS\template-service.

3. В поле Application pool выберите ранее созданный пул.
4. В поле Port укажите порт, по которому будет осуществляться связь клиента с сервером IIS.
5. Перейдите в «Главный узел» → «Feature Delegation, Custom Site Delegation» (Рисунок 9). В открывшемся окне для сайта express-template-service выдайте права на чтение и запись (read/write) для параметра Authentication – Anonymous.

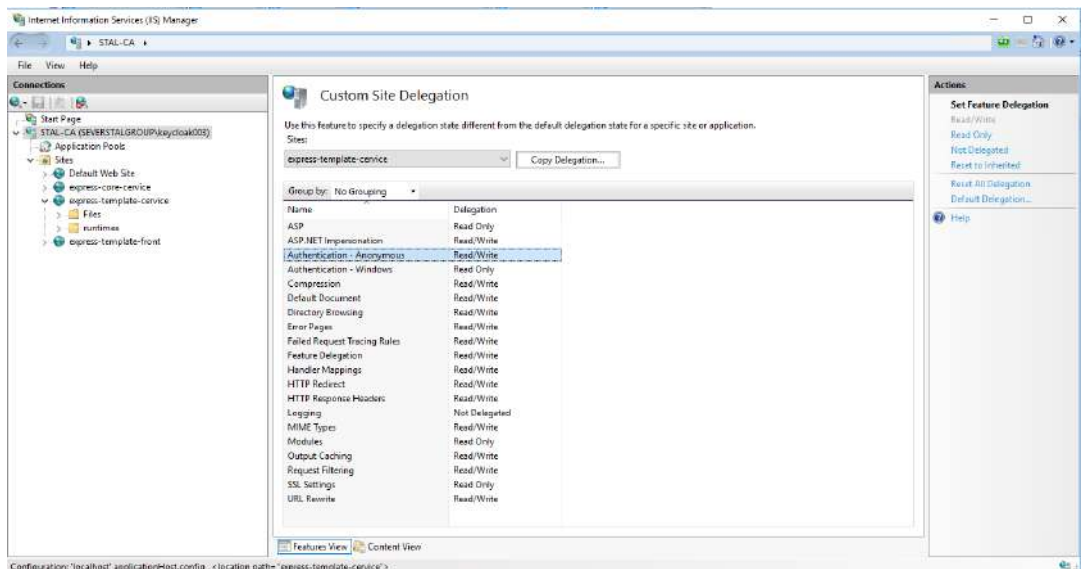


Рисунок 9

6. Для узла сайта express-template-service через Configuration Editor выставьте среду выполнения system.webServer/aspNetCore → environmentVariables From ApplicationHost.config (Рисунок 10). Заполните параметры:

```
name: ASPNETCORE_ENVIRONMENT
value: Production
```

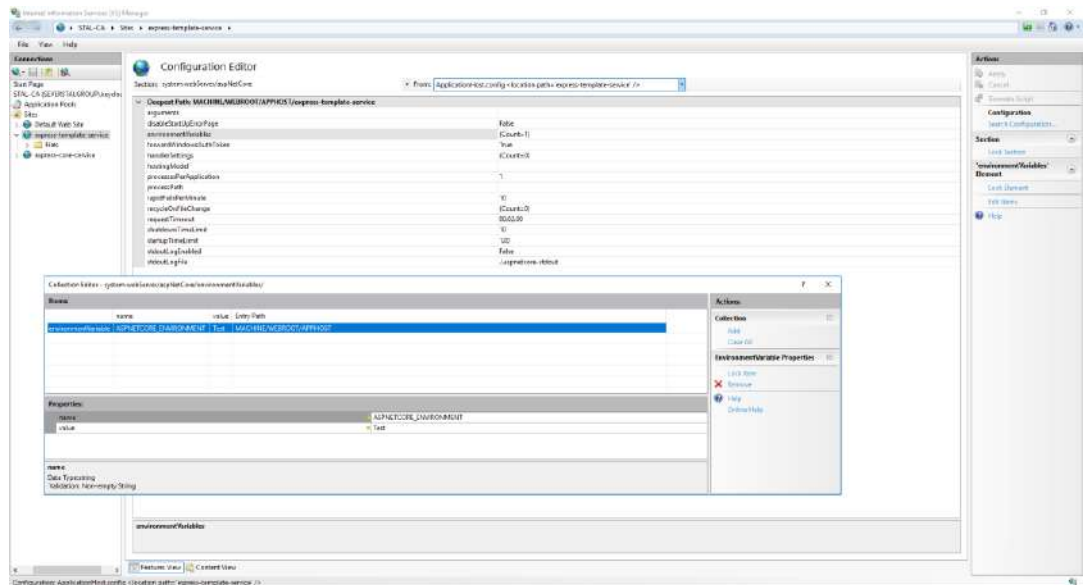


Рисунок 10

7. Отредактируйте в файле `appsettings.Production.json`⁷ параметры среды:

- установите строку подключения к БД PostgreSQL в ключе:

```
DatabaseOptions.Connection="Host=127.0.0.1;Port=5432;Database=express_meeting_db;Username=db_express;Password=Pass1234;Pooling=true;Minimum Pool Size=50;Maximum Pool Size=100;Include Error Detail=True;"
```

Примечание. Параметры `Username=db_express;Password=Pass1234` должны соответствовать учетным данным, которые были созданы при настройке PostgreSQL. База данных та же самая, что используется `express-core-service`.

- установите настройки для генерации JWT-токена аутентификации (используется при обращении к методам сервиса) в ключах:

```
JWTOptions.Key="JFRINXV0LG1hLX1EVz1YdX1ae316UU0zJzR9WVUtIXA"
JWTOptions.Issuer="Issuer"
JWTOptions.Audience="Audience"
JWTOptions.ExpiredMinutes=120
```

Примечание. Параметр `JWTOptions.Key` содержит произвольный набор символов, который используется для генерации ключа, которым будет подписан токен. Параметр `JWTOptions.ExpiredMinutes` задает время жизни токена в минутах. Параметры допустимо не менять, если это не регламентировано требованиями информационной безопасности компании.

8. Выполните

```
iisreset
```

9. Если сайт успешно развернут, то при переходе на него по пути `/front` (пример: `http://localhost:7200/front`) в браузере должна открыться страница логина в систему администрирования шаблонов плагина (Рисунок 11).

⁷ Предполагаемый путь `IIS\template-service\appsettings.Production.json`



Рисунок 11

Первый вход осуществляется под учетной записью admin с пустым паролем.

ШАГ 5. НАСТРОЙКА ЗАПИСИ ЛОГОВ EXPRESS-TEMPLATE-SERVICE

Для настройки записи логов express-template-service:


1. Выдайте права FullControl для Everyone на сервере, где установлен express-template-service, для папки `c:/logs/template-service/` и ее содержимого.
2. При необходимости поменяйте путь к папке логов (см. сноску к [п.1](#) раздела «Шаг 2. Настройка записи логов серверной части плагина и клиентских частей»).

Глава 5

ДИАГНОСТИКА НЕИСПРАВНОСТЕЙ ПЛАГИНА OUTLOOK

Данный раздел не содержит указаний по исправлению выявленных неисправностей, а помогает провести диагностику.

Принцип работы плагина Outlook:

1. При нажатии кнопки  в Outlook формируется и отправляется запрос на сервер IIS.
2. На IIS-сервере производится аутентификация и авторизация.
3. В случае успешного прохождения аутентификации и авторизации, IIS передает в плагин Outlook шаблон письма из БД и отправляет запрос на создание шаблона будущей конференции в чат-бот.
4. Conference Notifier Bot создает шаблон будущей конференции в eXpress.
5. Клиентская часть плагина Outlook создает письмо в Outlook из шаблона, вставив в него ссылку на будущую конференцию.
6. Пользователь указывает, участников, дату, время и название будущей конференции, заполняя соответствующие поля в письме Outlook.
7. После того как пользователь нажимает кнопку «Отправить», Outlook-плагин формирует и отправляет запрос на изменение созданной ранее конференции на сервер IIS с учетом параметров, заполненных пользователем.
8. В случае успешного прохождения аутентификации и авторизации, IIS-сервер отправляет запрос на изменение шаблона будущей конференции в чат-бот с учетом параметров, заполненных пользователем.
9. Чат-бот меняет параметры конференции в eXpress на актуальные.

Порядок диагностики неисправного плагина Outlook описан ниже.

ШАГ 1. ПРОВЕРКА РАБОТОСПОСОБНОСТИ CONFERENCE NOTIFIER BOT

В первую очередь проверьте работоспособность Conference Notifier Bot в серверной части eXpress, так как если чат-бот не принимает или не создает конференции в eXpress, то дальнейшие действия по настройке и диагностике других компонентов системы будут бесполезны.

Для проверки работоспособности Conference Notifier Bot:

1. Зайдите по SSH на корпоративный сервер (CTS) eXpress.
2. Выполните запрос к Conference Notifier Bot через программу Curl.
3. Выполните запрос к Conference Notifier Bot через чат мобильного приложения или веб/десктоп-версию приложения eXpress:
 - если запрос к Conference Notifier Bot через программу Curl и чат в eXpress выполнен успешно, переходите к следующему шагу;
 - если запрос к Conference Notifier Bot через программу Curl или чат в eXpress не был выполнен, тогда нужно восстановить работоспособность (проверить настройки) Conference Notifier Bot и перейти к следующему шагу.

Действия по проверке работоспособности Conference Notifier Bot через программу Curl описаны в Главе 4 «Установка Outlook-плагина» настоящей инструкции:

- если безопасность чат-бота не настраивалась см. [пп. 5—7](#) раздела «Шаг 1. Включение API Conference Notifier Bot без пароля»;
- если безопасность чат-бота настраивалась см. [пп. 6—8](#) раздела «Шаг 2. Защита Conference Notifier Bot».

ШАГ 2. ПРОВЕРКА ДОСТУПНОСТИ CONFERENCE NOTIFIER BOT С СЕРВЕРА IIS

Далее рекомендуется проверить доступность Conference Notifier Bot с сервера IIS.

Для проверки доступности Conference Notifier Bot с сервера IIS:

1. Зайдите по RDP или другим способом на сервер IIS, где установлено приложение express-core-service.
2. В PowerShell ISE запустите диагностический скрипт из [Приложение 1](#)
3. [Диагностический скрипт № 1](#)», отредактировав значения переменных (имена пользователей, ссылку, дату и время конференции).
4. Если скрипт отработал с ошибками, проанализируйте и устраните их причины (скорее всего проблема в сетевом доступе или аутентификации), затем запустите скрипт снова, пока он не отработает без ошибок.
5. Если скрипт отработал без ошибок и выдал ID конференции, которую можно найти в консоли администрирования корпоративного сервера CTS eXpress, тогда считается, что Conference Notifier Bot доступен с сервера IIS, переходите к следующему шагу.

ШАГ 3. ПРОВЕРКА ДОСТУПНОСТИ СЕРВЕРА IIS С КЛИЕНТСКОЙ РАБОЧЕЙ СТАНЦИИ

Далее рекомендуется проверить доступность приложения express-core-service на сервере IIS с клиентской рабочей станции.

Для проверки доступности сервера IIS с клиентской рабочей станции:

1. Подключитесь через RDP к клиентской рабочей станции, на которой установлены приложение Microsoft Outlook и плагин для Outlook.
2. В PowerShell ISE запустите диагностический скрипт из [Приложение 2](#)
3. [Диагностический скрипт № 2](#)», отредактировав значения переменных (учетную запись пользователя Outlook, ссылку на IIS).
4. Если скрипт отработал с ошибками, проанализируйте и устраните их причины (скорее всего проблема в настройках IIS и приложения express-core-service, отсутствует сетевой доступ или проблемы аутентификации и авторизации), затем запустите скрипт снова, пока он не отработает без ошибок.
5. Если скрипт отработал без ошибок и выдал ID конференции, которую можно найти в консоли администрирования корпоративного сервера CTS eXpress, тогда считается, что Conference Notifier Bot доступен с сервера IIS, переходите к следующему шагу.

ШАГ 4. ОБЩАЯ ДИАГНОСТИКА ПЛАГИНА OUTLOOK

Если предыдущие 3 шага пройдены успешно, но плагин Outlook все равно не работает, проверьте корректность настроек в конфигурационном файле клиентского ПО (см. раздел [«Шаг 3. Разворачивание Outlook-плагина»](#) настоящей инструкции).

Если конфигурационный файл клиентского ПО корректен, изучите файлы логов из клиентских приложений, описанные в разделе [«Шаг 3. Разворачивание Outlook-плагина»](#) настоящей инструкции, и серверные файлы логов, описанные в разделе [«Шаг 2. Настройка записи логов серверной части плагина и клиентских частей»](#), устраните причины ошибок, при необходимости обратитесь в техническую поддержку производителя за помощью.

Приложение 1

ДИАГНОСТИЧЕСКИЙ СКРИПТ № 1

Приложение не содержит указаний по исправлению выявленных неисправностей, а разъясняет принцип работы скрипта диагностики.

```
# Данный скрипт предназначен для проверки доступности Conference Notifier Bot с
сервера IIS express-core-service
# Запускать данный скрипт имеет смысл, только после того как доступность
Conference Notifier Bot успешно проверяется CURL с сервера Express см.
инструкции по установке Outlook-плагина Express посл.версии.
# запускать с сервера IIS который обеспечивает функционирование Outlook-плагина
Express (лучше в PowerShell ISE, так как на скриншоте будет видно и тело
скрипта, и результат)
# в результате успешного выполнения данного скрипта в Express создастся
конференция, с CallID (данный скрипт выведет CallID)
# по CallID эту конференцию можно будет найти в консоли администратора Express
# Успешное выполнение данного скрипта, говорит о том, что Conference Notifier
Bot с сервера IIS доступен.
$headers = New-Object
[System.Collections.Generic.Dictionary[[String],[String]]
$headers.Add("Content-Type", "application/json")
$headers.Add("Authorization", "Bearer 3bfef52b32685598d1f73301dd3c0431") # тут
введите свой bearer token если задан, а если не задан, то оставьте так или
закомментируйте

# ниже заполнить имя конференции, участников, создателя и дату с временем (дата
должна быть не ниже текущей) при повторном запуски лучше поменять имя
конференции
$body = @"
{"name":null,"members":["Пользователь001@mydomain.com"],"admins":["Пол
ьзователь001@mydomain.com"],"creator":"Пользователь001@mydomain.com","sta
rt_at":"0001-01-01T00:00:00","end_at":"0001-01-
01T00:00:00","link":{"link_type":"public","access_code":null}}
"@
$response = Invoke-RestMethod 'https://my-CTS-
domain.com/api/v1/conference_Bot/conferences/' -Method 'POST' -Headers $headers
-Body $body
$response | ConvertTo-Json
$callId = $response.result.call_id

$headers = New-Object
[System.Collections.Generic.Dictionary[[String],[String]]
$headers.Add("Content-Type", "application/json")
$headers.Add("Authorization", "Bearer 3bfef52b32685598d1f73301dd3c0431")# тут
введите свой bearer token если задан, а если не задан, то оставьте так или
закомментируйте

# ниже заполнить имя конференции, участников, создателя и дату с временем (дата
должна быть не ниже текущей) при повторном запуске лучше поменять имя
конференции
$body = @"
{"name":"Change
name","members":["Пользователь001@mydomain.com"],"Пользователь002@mydomain
.com"],"admins":["Пользователь001@mydomain.com"],"actor":"Пользователь0
01@mydomain.com","start_at":"2023-12-31T12:30:00+03:00","end_at":"2023-
12-31T13:00:00+03:00","link":{"link_type":null,"access_code":null}}
"@
$response = Invoke-RestMethod "https://my-CTS-
domain.com/api/v1/conference_Bot/conferences/$callId" -Method 'POST' -Headers
$headers -Body $body
$response | ConvertTo-Json
```

Приложение 2

ДИАГНОСТИЧЕСКИЙ СКРИПТ № 2

Приложение не содержит указаний по исправлению выявленных неисправностей, а разъясняет принцип работы скрипта диагностики.

Проверьте доступность чат-бота с помощью запроса с bearer token через CURL, например:

```
# Данный скрипт предназначен для проверки доступности связки Conference
Notifier Bot+IIS express-core-service с клиента (ПК, где установлен Outlook)
# Запускать данный скрипт имеет смысл, только после того как доступность
Conference Notifier Bot успешно проверяется CURL с сервера Express см.
инструкции по установке Outlook-плагина Express посл.версии. и скриптом
test_from_IIS_PC_to_Express_PC_x_x.ps1 с IIS express-core-service
# запускать с клиента (ПК, где установлен Outlook) (лучше в PowerShell ISE, так
как на скриншоте будет видно и тело скрипта, и результат)
# в результате успешного выполнения данного скрипта - в Express создается
конференция, с CallID (данный скрипт выведет CallID)
# по CallID эту конференцию можно будет найти в консоли администратора Express
# Успешное выполнение данного скрипта, говорит о том, что Conference Notifier
Bot с сервера IIS доступен, и IIS express-core-service настроен правильно.
$creds = Get-Credential # УЗ AD OUTLOOK
$headers = New-Object
[System.Collections.Generic.Dictionary[[String],[String]]
$headers.Add("Content-Type", "application/json")

# ниже заполнить создателя
$body = @"
{"Creator`":`"Пользователь_Экспесс@mydomain.com`",`"LanguageId`":1049,`"isRecu
rrent`":false,`"`$type`":`"MeetingCreateRequest`"}
"@

# ниже ввести адрес IIS express-core-service такая же ссылка должна быть в
настройках клиентской части плагина.

$response = Invoke-RestMethod 'http://my-iis-domain.com:7100/api/meetings' -
Method 'POST' -Headers $headers -Body $body -Credential $creds
$response | ConvertTo-Json
```