

eXpress

Система
коммуникаций

Надстройка для Microsoft Outlook

Руководство по установке

Версия Keycloak 3.1.0.2



© Компания «Анлимитед продакшен», 2025. 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».

Указанные в документе адреса серверов, значения конфигурационных файлов, учетные пользовательские данные указаны для примера и носят исключительно ознакомительный характер. Пользовательские данные, в том числе биометрические, вымышленные и не содержат персональных данных.

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

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	4
ГЛАВА 1	
ОБЩИЕ ТРЕБОВАНИЯ	5
ГЛАВА 2	
АРХИТЕКТУРА	7
Single CTS или разделенный CTS	7
Несколько CTS	8
Несколько CTS и 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 НАДСТРОЙКИ ДЛЯ OUTLOOK	13
Шаг 1. Обновление серверной части (опционально)	13
Шаг 2. Подготовка БД	13
Шаг 3. Разворачивание в Docker express-core-service и template-core-service	14
Шаг 4. Разворачивание надстройки для Outlook на клиентских ПК	17
ГЛАВА 5	
ВЕБ-ИНТЕРФЕЙС АДМИНИСТРАТОРА	20
Авторизация в веб-интерфейсе администратора	20
Описание интерфейса панели администратора	20
Управление администраторами	20
Шаблоны	22
ГЛАВА 6	
ДИАГНОСТИКА НЕИСПРАВНОСТЕЙ НАДСТРОЙКИ ДЛЯ OUTLOOK	24
Шаг 1. Проверка работоспособности Conference Notifier Bot	24
Шаг 2. Проверка доступности Conference Notifier Bot с сервера Docker.....	25
Шаг 3. Проверка доступности приложения express-meeting-core-service с клиентской рабочей станции	25
Шаг 4. Общая диагностика надстройки для Outlook	25
ПРИЛОЖЕНИЕ 1	
ДИАГНОСТИЧЕСКИЙ СКРИПТ № 1	26
ПРИЛОЖЕНИЕ 2	
ДОБАВЛЕНИЕ СЕРТИФИКАТА В КОНТЕЙНЕР EXPRESS-MEETING-CORE-SERVICE-MAIN	27
ПРИЛОЖЕНИЕ 3	
СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ	29
ПРИЛОЖЕНИЕ 4	
ТИПОВЫЕ ОШИБКИ	32

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
Docker	платформа контейнеризации с открытым исходным кодом, с помощью которой можно автоматизировать создание приложений, их доставку и управление
Docker-compose	Надстройка(утилита) над Docker, приложение на Python, которое позволяет запускать множество контейнеров одновременно и маршрутизировать потоки данных между ними.
ETS	Enterprise Transport Server — сервер предприятия
express-core-service	Серверное приложение для надстройки для Outlook, обеспечивающее основную функциональность (создание и изменение конференций)
express-template-service	Серверное приложение для надстройки для Outlook, обеспечивающее дополнительную функциональность (создание и изменение шаблонов письма, текст, логотип, подпись и т. п.)
JSON	Текстовый формат обмена данными, основанный на JavaScript
KeyCloak	Приложение для реализации единой точки идентификации, аутентификации и авторизации. Это система идентификации и управления доступом с открытым исходным кодом (Open Source)
MS Exchange	Почтовый сервер Microsoft
MS Outlook	Почтовый клиент Microsoft, часть пакета приложений MS Office
Single CTS	Единый корпоративный сервер
БД	База данных
Контейнер	Стандартизированный, изолированный и портативный пакет программного обеспечения, который включает в себя все необходимое для запуска приложения, включая код, среду выполнения, системные инструменты, библиотеки и настройки.
Лог	Запись в журнале событий сервера и/или клиента
ОС	Операционная система
ПК	Персональный компьютер
ПО	Программное обеспечение
Приглашенный участник	Пользователь(и), которого(ых) пригласил создатель конференции Express с помощью надстройки Outlook
Разделенный CTS	Разделенный корпоративный сервер: Front CTS и Back CTS
СК «Express», Express, система	Система коммуникаций «Express»
Создатель конференции	Пользователь, который создает и может редактировать конференцию Express с помощью надстройки Outlook
СУБД	Система управления базами данных
Чат-бот	Чат-бот Conference Notifier Bot

Глава 1

ОБЩИЕ ТРЕБОВАНИЯ

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

- администрирование Windows Server;
- администрирование Keycloak;
- администрирование PostgreSQL;
- администрирование Linux;
- администрирование Docker;
- администрирование eXpress;
- понимание JSON.

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

- версия Docker не ниже Docker version 24.0.7;
- версия Docker-compose не ниже version 1.29.2;
- требования к БД:
 - версия PostgreSQL не ниже 13.11 (Debian 13.11-0+deb11u1, Windows);
 - наличие супер-пользователя на сервере PostgreSQL;
 - отдельная БД PostgreSQL, отдельный пользователь БД PostgreSQL: данный пользователь должен быть владельцем этой БД и иметь все права на нее;
- наличие установленного и настроенного решения Keycloak не ниже версии 22.0.1;
- версия клиентской части Outlook не ниже 2013;
- Outlook должен использоваться с почтовым ящиком, который соответствует адресу e-mail, прописанному в учетной записи СК «Express»;
- наличие установленного и настроенного решения 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».
- должно быть настроено сетевое взаимодействие между компонентами по соответствующим портам;
- на клиентских ПК должен быть установлен пакет WebView2 Runtime версии не ниже 122.0.2365.66 (<https://developer.microsoft.com/ru-ru/microsoft-edge/webview2/?form=MA13LH>).

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

Элемент	Параметры
Процессор	64-bit 2.4 GHz processor 2 core
Оперативная память	4 ГБ
Операционная система	Debian GNU. Не ниже Linux 11
Жесткий диск	Не менее 40 ГБ

СУБД PostgreSQL может располагаться на выделенном сервере на ОС Windows или ОС Linux. Если предустановленная СУБД PostgreSQL отсутствует, и архитектура развернутого решения предполагает наличие Bot-сервера СК «Express», то для создания БД можно использовать docker-контейнер PostgreSQL, который входит в состав Bot-сервера. Для создания БД PostgreSQL с помощью docker-контейнера

PostgreSQL можно использовать сервер Linux, на котором будет развернута надстройка Outlook.

Внимание! Для создания БД PostgreSQL запрещено использовать сервера СК «Express»: CTS Back, Front, Media, Rec, ETS и другие. (кроме Bot-сервера).

Если на виртуальной машине не будет других сервисов, то сервер СУБД PostgreSQL под ОС Linux 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

Если на сервере ОС Linux для надстройки Outlook дополнительно разворачивается СУБД PostgreSQL, то добавьте 2 ГБ RAM и 20 ГБ HDD.

Если на виртуальной машине не будет других сервисов, то сервер СУБД PostgreSQL под ОС Windows 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

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

Важно! При использовании MS Exchange создатель конференции должен быть зарегистрирован в СК «Express» с почтовым адресом, который указан в его Primary SMTP в MS Exchange.

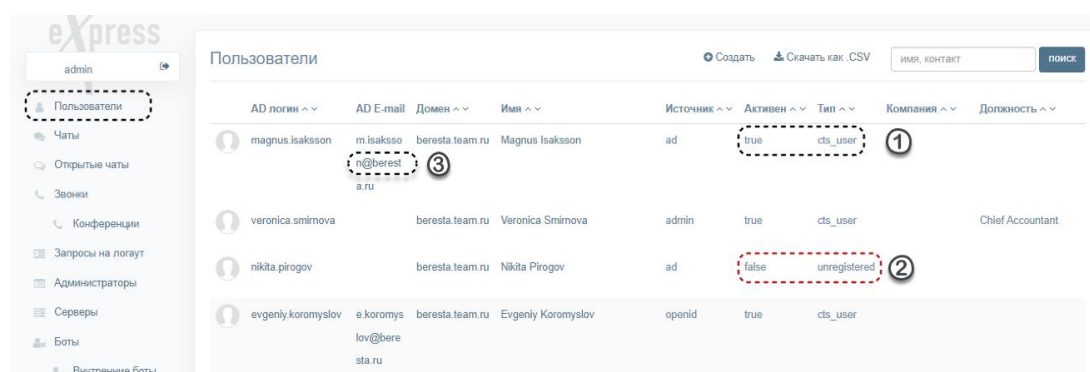


Рисунок 1. Статус пользователя: 1 – активный; 2 – неактивный

Чтобы предстоящая конференция автоматически добавлялась в приложении СК «Express» приглашенного пользователя, данный пользователь должен быть зарегистрирован в приложении с тем же почтовым адресом, который создатель конференции использовал для отправки приглашения в надстройке (Рисунок 1 указатель 3).

Пользователю придет письмо со ссылкой для самостоятельного подключения, если:

- пользователь отсутствует в СК «Express»;
- у пользователя статус «не активен» в СК «Express»;
- пользователь зарегистрирован в СК «Express» с другим почтовым адресом.

Глава 2

АРХИТЕКТУРА

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

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

При использовании единого шаблона письма все express-core-service могут использовать одну БД. При использовании разных шаблонов письма для разных express-core-service разверните для каждого из них свою БД и свой express-template-service.

SINGLE CTS ИЛИ РАЗДЕЛЕННЫЙ CTS

Если СК «Express» содержит один CTS-сервер, то разверните серверную часть надстройки в одном экземпляре. Ниже представлена типовая схема развертывания данного типа:

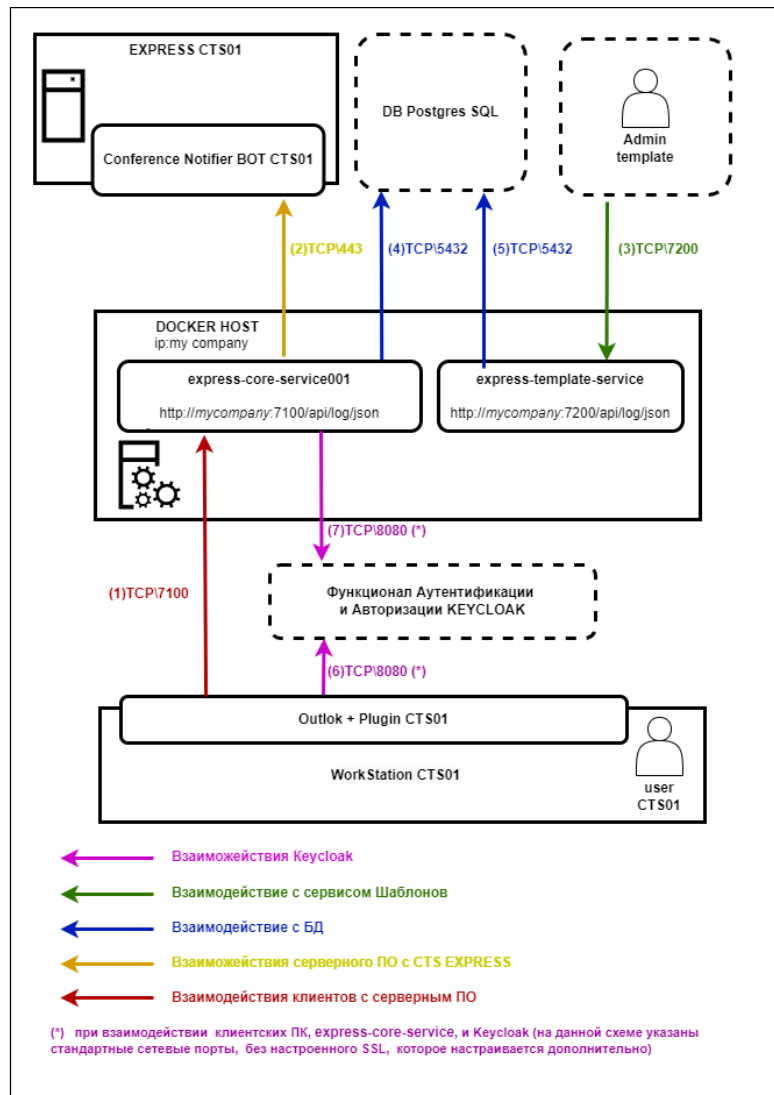


Рисунок 2. Типовая схема: 1 CTS, 1 express-core-service, 1 express-template-service

Номера сетевых взаимодействий соответствуют номеру строки в Таблице 1 Приложения 3.

НЕСКОЛЬКО CTS

Если СК «Express» содержит несколько CTS-серверов, то разверните серверную часть надстройки для каждого CTS-сервера отдельно (допускается развертывание на одном сервере Docker, но с разными портами и именами контейнеров, или на нескольких серверах Docker). Ниже представлена типовая схема развертывания данного типа:

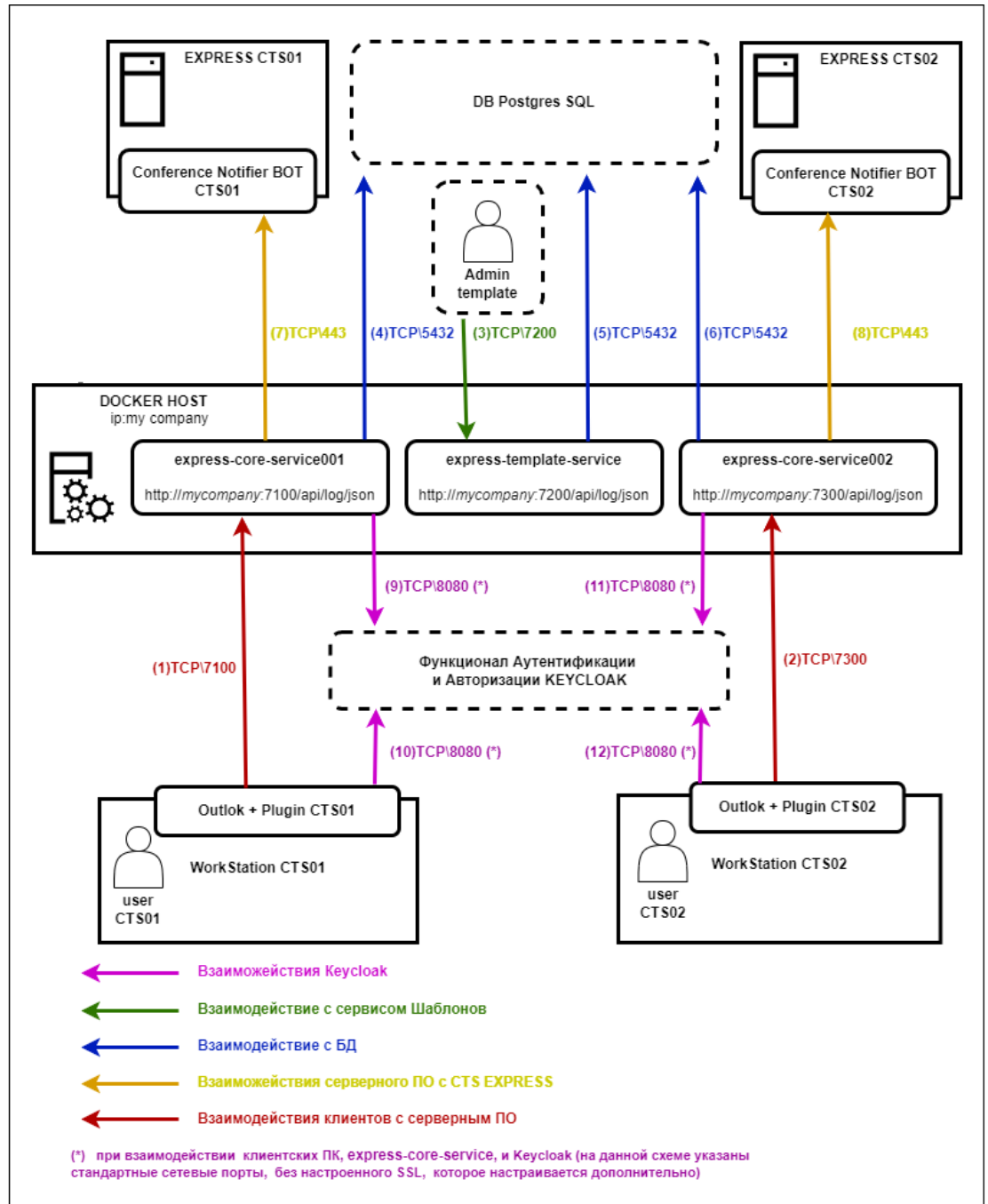


Рисунок 3. Типовая схема: 2 CTS, 2 express-core-service, 1 express-template-service

Номера сетевых взаимодействий соответствуют номеру строки в Таблице 2 Приложения 3.

НЕСКОЛЬКО CTS И ETS

Если СК «Express» содержит несколько CTS-серверов, объединенных ETS-сервером, разверните серверную часть надстройки только для ETS-сервера. Ниже представлена типовая схема развертывания данного типа:

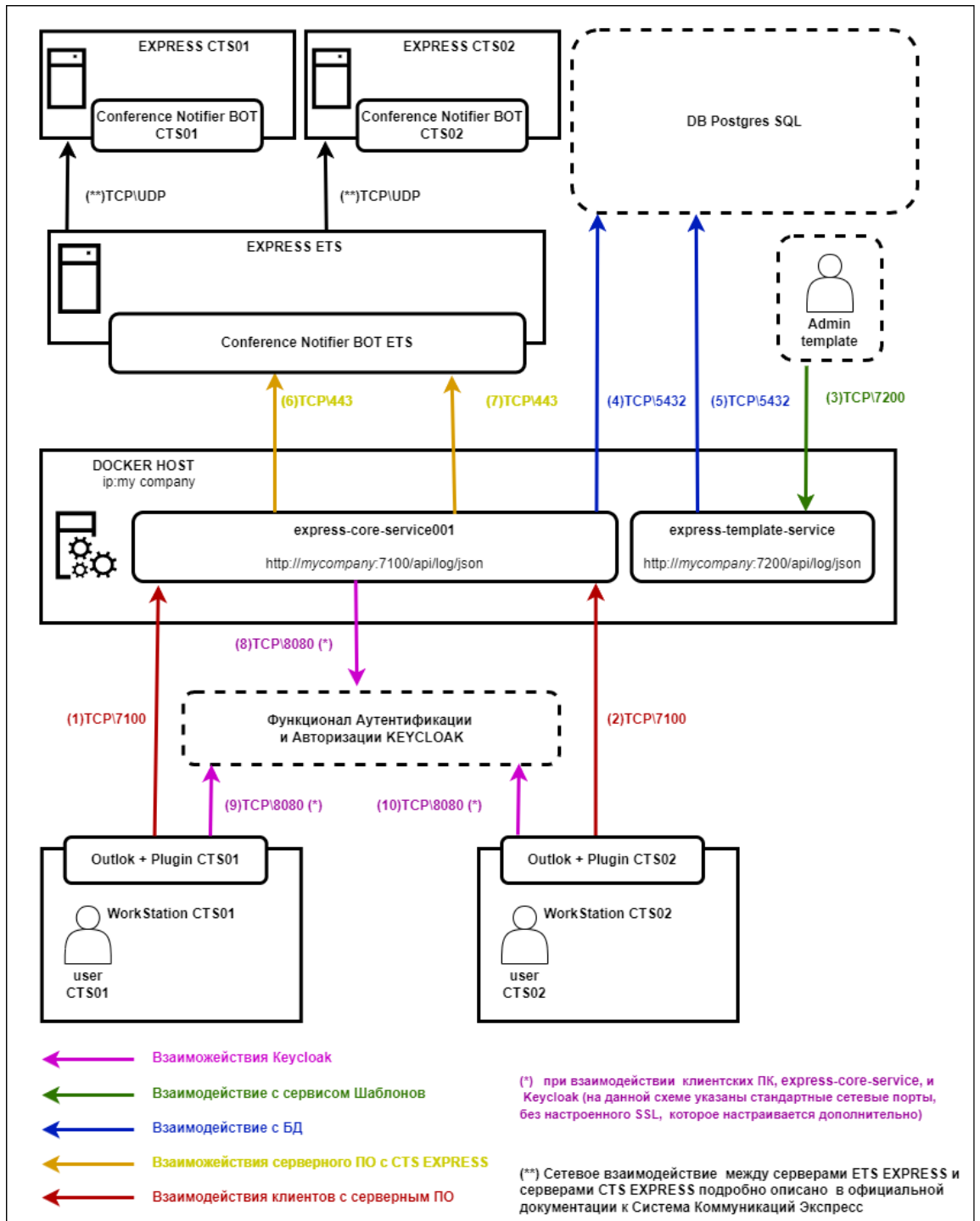


Рисунок 4. Типовая схема: 1 ETS, 2 CTS, 1 express-core-service, 1 express-template-service

Номера сетевых взаимодействий соответствуют номеру строки в Таблице 3 Приложения 3.

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 docker-контейнера после установки бота на сервере СК «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 СК «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 СК «Express» до последней версии.
2. Убедитесь, что версия дистрибутива «Надстройка для Microsoft Outlook» (серверная и клиентская) не ниже 1.2.0.0.

Внимание! Введенные учетные данные будут использоваться при настройке надстройки для Outlook.

3. Сгенерируйте BEARER TOKEN размером от 30 до 40 символов с помощью средства генерации паролей или с помощью команд:

```
openssl rand -hex 40
```

или

```
head /dev/urandom | tr -dc A-Za-z0-9 | head -c 40 ; echo
```

4. 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).

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

```
dp1 -d conference_bot
```

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

6. Проверьте недоступность чат-бота с помощью запроса без bearer token 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 без bearer token недоступен (так и должно быть), CURL выведет пустую строку.

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

```
curl -X POST -H "Content-Type: application/json" -H "Authorization:  
Bearer 3bfef52b32685srdsrherhFGd1f73301dd3c0431" -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"
```

8. В веб-интерфейсе администратора CTS-сервера перейдите в раздел «Звонки → Конференции».
9. В поле поиска введите `test_plugin_001` (был задан в п. 6) и нажмите кнопку «Поиск».

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

Глава 4

УСТАНОВКА НАДСТРОЙКИ ДЛЯ OUTLOOK

Для установки надстройки для Outlook выполните шаги, описанные ниже. Дистрибутив (серверная и клиентская часть) для скачивания доступен по ссылке <https://nc.express.ms/s/f3iGiJieaGyPM7j?path=%2FKeycloak> (выберите текущую версию продукта).

ШАГ 1. ОБНОВЛЕНИЕ СЕРВЕРНОЙ ЧАСТИ (ОПЦИОНАЛЬНО)

Шаг выполняется, если сервис уже был развернут в рамках предыдущей версии надстройки.

Для обновления серверной части:

1. Загрузите новые версии image-образов сервисов `express-meeting-core-service` и `express-meeting-template-service` в `docker`.
2. В папках, где лежат файлы форматов ENV и YML, запустите команду:

```
docker-compose --env-file ./env -f ./docker-compose.yml down
```
3. Из новой версии дистрибутива возьмите новые файлы форматов `.env` и `.yml` и укажите в них ваши параметры и новые image-образы.
4. Замените старые файлы форматов ENV и YML на новые в соответствующей папке на сервере Docker .
5. Выполните [Шаг 2](#) текущей главы для обновления структуры БД.
6. Запустите из папки `docker` контейнеры:

```
docker-compose --env-file ./env -f ./docker-compose.yml up -d
```

ШАГ 2. ПОДГОТОВКА БД

Для разворачивания развернуть БД, в которой хранятся шаблоны блока приглашения в конференцию при создании приглашения в письме Outlook:

1. Перейдите на Windows-машину, на которой есть доступ к серверу PostgreSQL. На машине должен быть установлен .NET Desktop Runtime 7.0 (<https://dotnet.microsoft.com/en-us/download/dotnet/7.0>).
2. Перенесите на машину папку `tools` из дистрибутива.
3. В файле `tools\appsettings.json` установите строку подключения к БД PostgreSQL в ключе:

```
"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`» и «`Password=Pass1234`» должны соответствовать учетным данным, которые были созданы при настройке PostgreSQL.

Сохраните изменения.

4. Запустите из папки `tools` в консоли:

```
ExpressMeeting.Tools.TemplatesDB.exe --ef-migrate
```

5. Дождитесь завершения работы утилиты.

БД готова к использованию.

ШАГ 3. РАЗВОРАЧИВАНИЕ В DOCKER EXPRESS-CORE-SERVICE И TEMPLATE-CORE-SERVICE

Сервис разворачивается в Docker.

Для разворачивания сервиса:

1. Загрузите из папки images-образов сервисов в docker:

```
docker load --input express-meeting-core-service-main.tar
docker load --input express-meeting-template-service-main.tar
```

2. Подготовьте настройки запуска сервисов в файле docker\env.

Файл содержит следующие настройки:

- наименование образа, из которого стартует контейнер с core-service (не нужно менять):

```
EXPRESS_MEETING_CORE_SERVICE_MAIN_IMAGE_NAME=express-meeting-  
core-service-main
```

- порт, по которому будет доступен core-service:

```
EXPRESS_MEETING_CORE_SERVICE_PORT=7100
```

- наименование образа, из которого стартует контейнер с template-service (не нужно менять):

```
EXPRESS_MEETING_TEMPLATE_SERVICE_MAIN_IMAGE_NAME=express-meeting-  
template-service-main
```

- порт, по которому будет доступен template-service:

```
EXPRESS_MEETING_TEMPLATE_SERVICE_PORT=7200
```

- IP-адрес, который соответствует DNS-имени сервера, на котором развернуто API чат-бота Conference Notifier Bot (указать реальные данные):

```
EXPRESS_MEETING_EXTRA_HOST=api.bot.express:127.0.0.1
```

- IP-адрес, который соответствует DNS-имени сервера, на котором развернут Keycloak (указать реальные данные):

```
EXPRESS_MEETING_EXTRA_HOST2=keycloak.mycompany:127.0.0.1
```

- имя файла с настройками core-service (не нужно менять):

```
EXPRESS_MEETING_CORE_ENV_FILE=core.env
```

- имя файла с настройками template-service (не нужно менять):

```
EXPRESS_MEETING_TEMPLATE_ENV_FILE=template.env
```

- папка контейнера с логами сервиса (не нужно менять):

```
EXPRESS_MEETING_DOCKER_LOGS_PATH=/public/express-meeting-logs
```

- папка docker-хоста, в которую будут добавляться логи из контейнеров сервисов (при необходимости изменить на нужную):

```
EXPRESS_MEETING_LOGS_PATH=/public/express-outlook/logs
```

3. Подготовьте настройки запуска core-service в файле docker\core.env.

Файл содержит следующие настройки:

- адрес API чат-бота Conference Notifier Bot (указать реальный адрес, по которому развернуто API):

```
ExpressOptions__Uri=https://api.bot.express/api/v1/conference_bot  
/conferences/
```

- схема авторизации для API чат-бота Conference Notifier Bot (Принимает значения: Basic – авторизация по логину/паролю; StaticToken – авторизация по статическому токenu):

```
ExpressOptions__AuthenticationScheme=StaticToken
```

- токен, который будет использован при обращении к API чат-бота Conference Notifier Bot (используется если в параметре ExpressOptions__AuthenticationScheme стоит значение StaticToken):

```
ExpressOptions__Token=Bearer 3bb326abc5609fed301dd3c0431
```

- логин учетной записи, от имени которой происходит обращение к API чат-бота Conference Notifier Bot (используется если в параметре ExpressOptions__AuthenticationScheme стоит значение Basic):

```
ExpressOptions__UserLogin=admin
```

- пароль учетной записи, от имени которой происходит обращение к API чат-бота Conference Notifier Bot (используется если в параметре ExpressOptions__AuthenticationScheme стоит значение Basic):

```
ExpressOptions__UserPassword=admin
```

- тип авторизации для core-service (не нужно менять):

```
AuthorizationOptions__AuthType=keycloak
```

- адрес Keycloak (указать реальный):

```
AuthorizationOptions__Configuration__Url=https://keycloak.mycompany
```

- realm, который используется для авторизации в Keycloak (указать реальный):

```
AuthorizationOptions__Configuration__Realm=realm
```

- значение Client ID, который используется для авторизации в Keycloak (указать реальный):

```
AuthorizationOptions__Configuration__ServiceClientId=client
```

- значение Client Secret, который используется для авторизации в Keycloak (указать реальный):

```
AuthorizationOptions__Configuration__ServiceClientSecret=secret
```

- строка подключения к БД PostgreSQL (это БД, в которой хранятся шаблоны блока приглашения в конференцию при создании приглашения в письме Outlook):

```
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» и «Password=Pass1234» должны соответствовать учетным данным, которые были созданы при настройке PostgreSQL.

- папка в контейнере core-service, в которую будут сохраняться логи (не нужно менять):

```
Serilog__WriteTo__1__Args__configureLogger__WriteTo__0__Args__path=/public/express-meeting-logs/coreservice.log
```

- папка в контейнере core-service, в которую будут сохраняться логи, полученные с клиентов (не нужно менять):

```
Serilog__WriteTo__2__Args__configureLogger__WriteTo__0__Args__path=/public/express-meeting-logs/client/plugin.log
```

- среда выполнения сервиса (не нужно менять):

```
ASPNETCORE_ENVIRONMENT=Production
```

4. Подготовка настроек запуска template-service в файле docker\template.env.

Файл содержит следующие настройки:

- строку подключения к БД PostgreSQL (это БД, в которой хранятся шаблоны блока приглашения в конференцию при создании приглашения в письме Outlook):

```
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» и «Password=Pass1234» должны соответствовать учетным данным, которые были созданы при настройке PostgreSQL.

- произвольный набор символов, который используется для генерации ключа, которым будет подписан JWT-токен доступа к сервису:
`JWTOptions__Key=JFRINXV0LGihLXL1EVz1YdXlae316UU0zJzR9WVUtIWQ`
- значение поля «Issuer» при генерации и проверке JWT-токена:
`JWTOptions__Issuer=http://127.0.0.1`
- значение поля «Audience» при генерации и проверке JWT-токена:
`JWTOptions__Audience=http://127.0.0.1`
- папку в контейнере template-service, в которую будут сохраняться логи (не нужно менять):
`Serilog__WriteTo__1__Args__path=/public/express-meeting-logs/templateservice.log`
- среда выполнения сервиса (не нужно менять):
`ASPNETCORE_ENVIRONMENT=Production`

5. Запустите из папки docker-контейнеры:

```
docker-compose --env-file ./env -f ./docker-compose.yml up -d
```

После запуска контейнеров, если в файле docker\core.env в ключах ExpressOptions__Uri, AuthorizationOptions__Configuration__Url были указаны HTTPS-адреса, в случае необходимости (например, не доступен удостоверяющий центр для используемых компанией сертификатов) добавьте в контейнер express-meeting-core-service-main полную цепочку сертификатов для соответствующих HTTPS-соединений. Пример добавления сертификатов в контейнер описан в [Приложении 2](#).

6. Зайдите в веб-интерфейс администратора Keycloak, перейдите в соответствующий realm и client, которые используются для аутентификации пользователей в надстройке для Outlook (данный клиент был указан в параметре [Client ID](#)).
7. В раздел Valid redirect URIs добавьте адрес(а) развернутого core-service, который(е) будут указаны в [клиентских конфигурационных файлах](#).
8. При необходимости добавьте в исключения антивируса файлы из следующих расположений:

```
/var/lib/docker
```

После запуска template-service контейнера при переходе на обслуживаемый этим контейнером адрес по пути /front (например: http://localhost:7200/front) в браузере должна открыться страница логина в систему администрирования шаблонов надстройки ([Рисунок 5](#)).

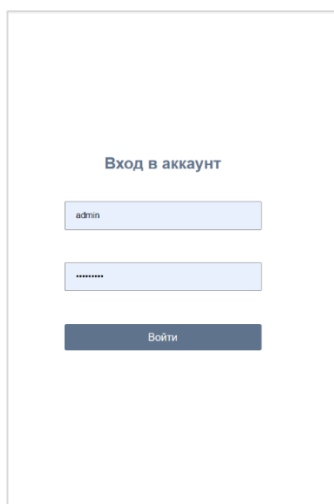


Рисунок 5

Примечание. Первый вход осуществляется под учетной записью admin с пустым паролем.

ШАГ 4. РАЗВОРАЧИВАНИЕ НАДСТРОЙКИ ДЛЯ OUTLOOK НА КЛИЕНТСКИХ ПК

Для разворачивания надстройки для Outlook на клиентских ПК:

1. Запустите инсталлятор ExpressMeeting_v_3.1.0.2.msi на машине, на которой установлен Outlook.
2. После инсталляции в файле ExpressMeeting.dll.config³ выполните настройки для работы надстройки:

- адрес core-service (серверной части надстройки) в ключе (замените mycompany на ip/dns-имя машины docker-хоста, где развернут core-service):

```
<add key="ExpressMeetingUrl" value="http://mycompany:7100/api/" />
```

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

Пример: ExpressMeeting_v_3.1.0.2.msi
ExpressMeetingUrl="http://mycompany:7100/api/"

- таймаут в секундах на установление соединения при обращении к Keycloak:

```
<add key="Timeout" value="120"/>
```

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

Пример: ExpressMeeting_v_3.1.0.2.msi Timeout="120"

- при необходимости установите режим запрета выбора типа конференции. Выпадающий список с выбором типа (Общее/Корпоративное/Доверенное)⁴ 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.

³ Предполагаемый путь c:\Program Files\Express\ExpressMeeting\ExpressMeeting.dll.config

⁴ Подробнее о типах создаваемых конференций можно узнать в руководствах пользователя СК «Express»

Пример: ExpressMeeting_v_3.1.0.2.msi DenyLinkType=true

- при необходимости установите тип создаваемой по умолчанию конференции (возможные значения: Public, Trusts, Corporate). По кнопке создать конференцию будут создаваться встречи указанного типа. Для этого добавьте в секцию configuration/appSettings параметр:

```
<add key="DefaultLinkType" value="Corporate"/>
```

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

Пример: ExpressMeeting_v_3.1.0.2.msi DefaultLinkType=Corporate

- при необходимости в файле ExpressMeeting.dll.config установите минимальный уровень требований к паролю (возможные значения: Public, Trusts, Corporate). По кнопке создать конференцию переключатель пароля будет недоступен для конференций указанного типа, а также более публичных. Для этого добавьте в секцию configuration/appSettings параметр:

```
<add key="LinkMinimumLevelRequirePassword" value="Corporate"/>
```

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

Пример: ExpressMeeting_v_3.1.0.2.msi LinkMinimumLevelRequirePassword=Corporate

- при необходимости включите фоновое обновление токена аутентификации, чтобы пользователь не вводил повторно учётные данные при длительном бездействии. Для этого добавьте в секцию configuration/appSettings параметр (По умолчанию значение — false (фоновое обновление отключено)):

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

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

Пример: ExpressMeeting_v_3.1.0.2.msi BackgroundRefreshTokenProcess=true.

- при необходимости задайте интервал фонового обновления токена аутентификации. Период указывается в формате «часы:минуты:секунды». Например, для 15 минут укажите 00:15:00. Для этого добавьте в секцию configuration/appSettings параметр:

```
<add key="BackgroundRefreshTokenProcessPeriod" value="00:15:00"/>
```

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

Пример: ExpressMeeting_v_3.1.0.2.msi
BackgroundRefreshTokenProcessPeriod=00:15:00

- при необходимости включить запись внутреннего лога Serilog (например, для диагностики случаев, когда основные логи не записываются), укажите полный путь к файлу. Для этого добавьте в секцию configuration/appSettings параметр:

```
<add key="selfLogPath" value="C:\logs\serilogInternal.log"/>
```

3. В файле `serilogSettings.json`⁵ если нужно использовать телеметрию, то установите адрес `core-service` (серверной части надстройки) в ключе (замените `mycompany` на IP/DNS-имя машины `docker`-хоста, где развернут `core-service`):

```
Serilog.WriteTo [Name=Telemetry] .Args .telemetryUrl="http://mycompany:7100/api/log/json"
```

Примечание. Параметр можно автоматически проинициализировать при установке `msi`-пакета. Для этого в командной строке необходимо передать параметр `TelemetryUrl`. По умолчанию секция настройки телеметрии в `serilogSettings.json` отсутствует

Пример: `ExpressMeeting_v_3.1.0.2.msi`

```
TelemetryUrl=" http://mycompany:7100/api/log/json "
```

4. Надстройка поддерживает разные уровни логирования. Для изменения уровня логирования в файле `serilogSettings.json`⁶ установите значение ключа `Serilog.MinimumLevel.Default` на требуемый уровень. Возможные значения уровня логирования: `Verbose`, `Debug`, `Information`, `Warning`, `Error`, `Fatal`. Где `Verbose` – логировать все, `Debug` – логировать `Debug` и выше, `Information` – логировать `Information` и выше и т.п.

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

Пример: `ExpressMeeting_v_3.1.0.2.msi LogLevel=Information`

5. При необходимости добавьте в исключения антивируса клиентских ПК следующие расположения файлов:
- `C:\Program Files\Express\ExpressMeeting` (для брендированных версий надстройки данный путь может отличаться);
 - `C:\logs` (если в [Главе 4, Шаг 4, п. 2](#) был задан другой путь, то следует указать его).

⁵ Предполагаемый путь `c:\Program Files\Express\ExpressMeeting\serilogSettings.json`.

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

Глава 5

ВЕБ-ИНТЕРФЕЙС АДМИНИСТРАТОРА

В данном разделе описаны веб-интерфейс администратора и работа в нем.

АВТОРИЗАЦИЯ В ВЕБ-ИНТЕРФЕЙСЕ АДМИНИСТРАТОРА

После установки серверной части надстройки Outlook в браузере должна открыться страница входа в систему администрирования шаблонов надстройки (Рисунок 6).

Для авторизации в консоли администратора:

1. В адресной строке браузера укажите адрес веб-интерфейса администратора (например, <http://localhost:7200/front>).

Откроется окно авторизации (Рисунок 7).

2. Введите имя учетной записи и пароль в соответствующие поля.

Примечание. Первый вход осуществляется под учетной записью admin с пустым паролем.

3. Нажмите кнопку «Войти».

Авторизация будет осуществлена.

ОПИСАНИЕ ИНТЕРФЕЙСА ПАНЕЛИ АДМИНИСТРАТОРА

В данном разделе описывается интерфейс панели администратора на примере интерфейса раздела «Администраторы» (Рисунок 6).

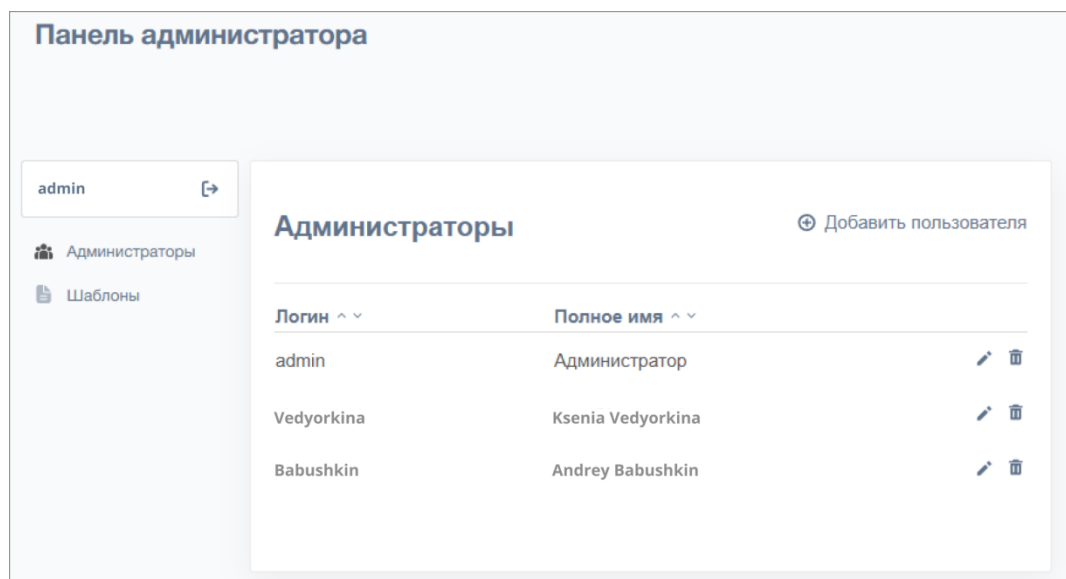



Рисунок 6. Интерфейс раздела «Администраторы»

УПРАВЛЕНИЕ АДМИНИСТРАТОРАМИ

Администратору доступны следующие операции:

- добавление администратора надстройки;
- редактирование данных администратора надстройки;
- удаление администратора надстройки.

Для добавления администратора надстройки:

1. Нажмите на кнопку « Добавить пользователя».
Откроется форма добавления администратора ([Рисунок 7](#)).
2. Внесите всю необходимую информацию в поля формы и нажмите кнопку «Добавить».

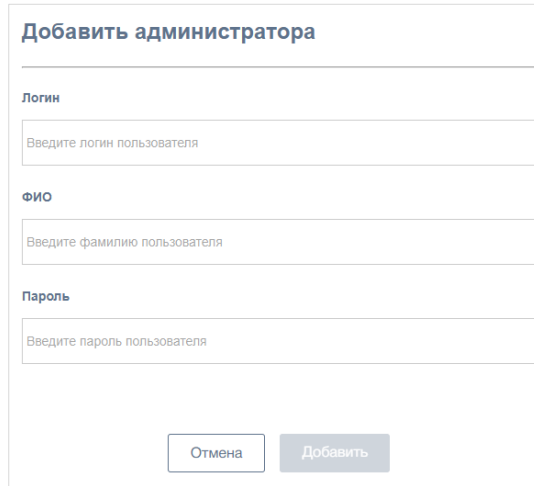


Рисунок 7. Добавление администратора

Для редактирования данных администратора:

1. Выберите администратора из списка ([Рисунок 6](#)) и нажмите на кнопку  «Редактировать».

Откроется форма редактирования данных администратора ([Рисунок 8](#)):

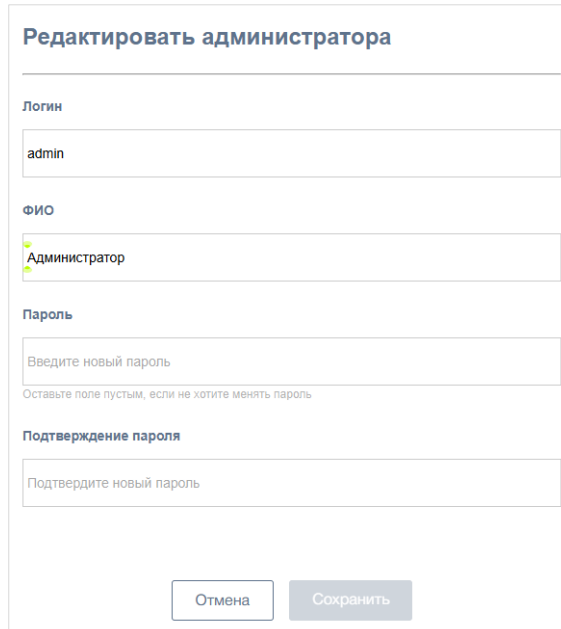



Рисунок 8. Внесение изменений в данные администратора

2. Внесите необходимые изменения и нажмите кнопку «Сохранить».

Для удаления администратора выберите администратора из списка ([Рисунок 6](#)) и нажмите на кнопку  .

В разделе «Шаблоны» ([Рисунок 9](#)) администратор может создавать, редактировать (в том числе в режиме HTML), и удалять шаблоны пригласительных писем.

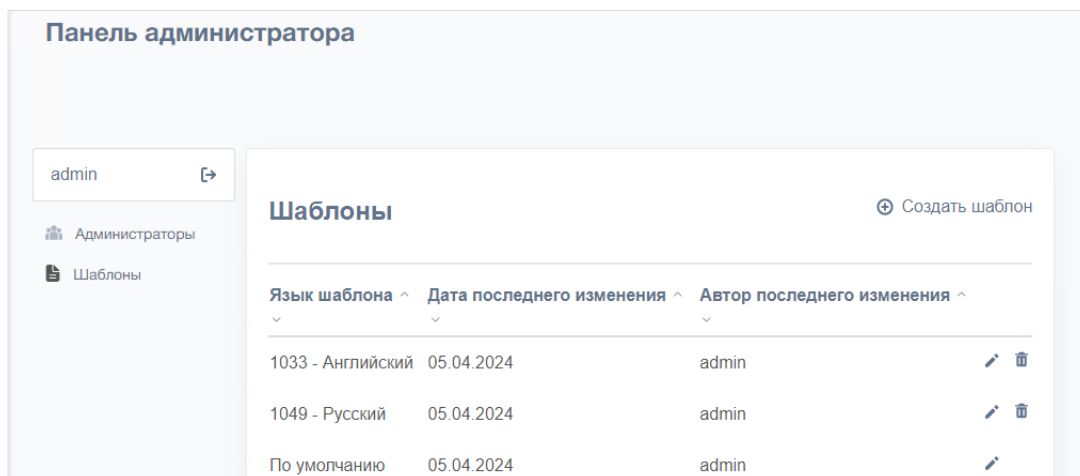


Рисунок 9. Раздел «Шаблоны»

Для создания шаблона:

1. Нажмите кнопку «Создать шаблон».

Откроется специальный визуальный редактор шаблона. Он включает в себя множество функций: редактирование содержания, коррекция шрифта и размера текста, добавление верхнего и нижнего индексов, HTML-редактор, а также кнопки для вставки шаблонов-автоподстановки (ссылки на конференцию, SIP-номера текущего пользователя) ([Рисунок 10](#)).

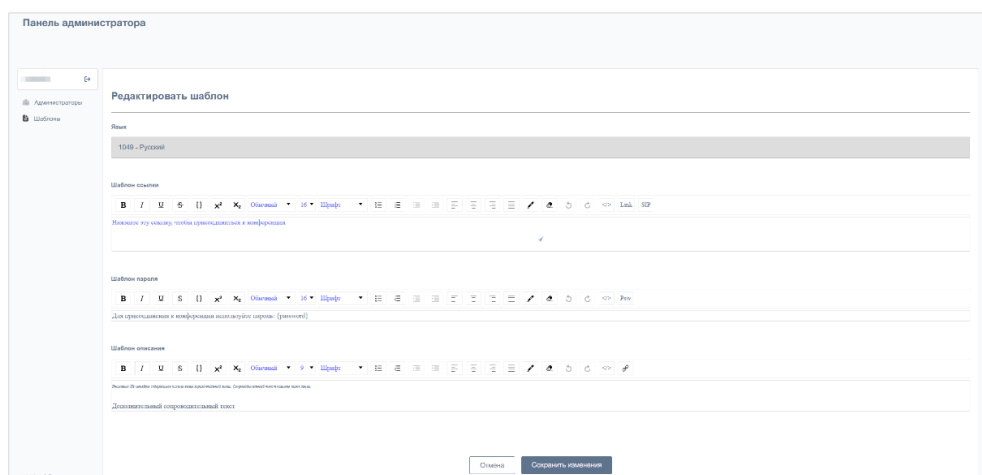


Рисунок 10. Визуальный редактор шаблона


2. Заполните поля шаблона и нажмите кнопку «Сохранить изменения».

Для редактирования шаблона:

1. Выберите шаблон из списка ([Рисунок 9](#)) и нажмите на кнопку «Редактировать».

Откроется визуальный редактор шаблона ([Рисунок 10](#)).

2. Отредактируйте поля шаблона и нажмите кнопку «Сохранить изменения».

Для удаления шаблона выберите его из списка и нажмите на кнопку  .

HTML-РЕДАКТОР

Для перехода в HTML-редактор нажмите на кнопку «</>» в меню визуального редактора шаблонов. Окно редактора примет следующий вид (Рисунок 11):



Рисунок 11. HTML-редактор

Редактор в полной мере поддерживает язык разметки HTML. В нем можно добавлять изображения, используя тег ``, и кодировать их в формате base64. Например:

```

```

В приглашении письмо это отобразится следующим образом: (Рисунок 12):

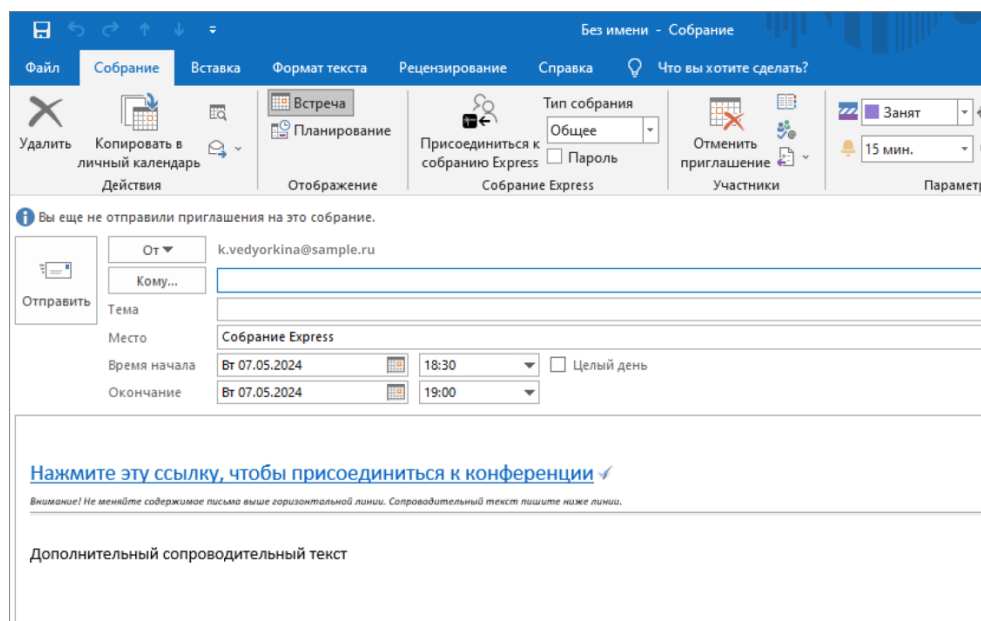


Рисунок 12. Результат работы в HTML-редакторе

Шаблонов на данный момент два – на русском и на английском языках. Шаблон выбирается клиентской частью надстройки и зависит от языковой локали MS Outlook. В русском MS Outlook будет автоматически применен русский шаблон, в английском – английский шаблон. Ручной выбор шаблонов не предусмотрен.


Глава 6

ДИАГНОСТИКА НЕИСПРАВНОСТЕЙ НАДСТРОЙКИ ДЛЯ OUTLOOK

Данный раздел содержит рекомендации для проведения корректной диагностики.

Внимание! Раздел не содержит указаний по исправлению выявленных неисправностей.

Принцип работы надстройки для Outlook:

1. При нажатии кнопки  в Outlook формируется и отправляется запрос на сервер Keycloak.
2. На Keycloak-сервере производится аутентификация и авторизация.
3. В случае успешного прохождения аутентификации и авторизации, Keycloak передает Token в express-meeting-core-service.
4. Сервис express-meeting-core-service передает в надстройку для Outlook шаблон письма из БД и отправляет запрос на создание шаблона будущей конференции в чат-бот.
5. Conference Notifier Bot создает шаблон будущей конференции в СК «Express».
6. Клиентская часть надстройки для Outlook создает письмо в Outlook из шаблона, вставив в него ссылку на будущую конференцию.
7. Пользователь указывает участников, дату, время и название будущей конференции, заполняя соответствующие поля в письме Outlook.
8. Когда пользователь нажимает кнопку «Отправить», надстройка для Outlook формирует и отправляет запрос на изменение созданной ранее конференции в express-meeting-core-service с учетом параметров, заполненных пользователем.
9. В случае успешного прохождения аутентификации и авторизации express-meeting-core-service отправляет запрос на изменение шаблона будущей конференции в чат-бот с учетом параметров, заполненных пользователем.
10. Чат-бот меняет параметры конференции в СК «Express» на актуальные.

Если описанный выше принцип работы надстройки не выполняется, то используйте следующие шаги для диагностики проблемы:

ШАГ 1. ПРОВЕРКА РАБОТОСПОСОБНОСТИ CONFERENCE NOTIFIER BOT

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

Для проверки работоспособности Conference Notifier Bot:

1. Зайдите по SSH на CTS-сервер СК «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 СК «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 надстройки для Outlook» 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. 7—9 раздела «Шаг 2. Защита Conference Notifier Bot».

ШАГ 2. ПРОВЕРКА ДОСТУПНОСТИ CONFERENCE NOTIFIER BOT С СЕРВЕРА DOCKER

Далее рекомендуется проверить доступность Conference Notifier Bot с сервера Docker.

Для проверки доступности Conference Notifier Bot с сервера Docker:

1. Зайдите по SSH или другим способом на сервер Docker, где установлено приложение express-core-service.
2. Проведите диагностику аналогично Шагу 1 «Проверка работоспособности Conference Notifier Bot» – выше.

ШАГ 3. ПРОВЕРКА ДОСТУПНОСТИ ПРИЛОЖЕНИЯ EXPRESS-MEETING-CORE-SERVICE С КЛИЕНТСКОЙ РАБОЧЕЙ СТАНЦИИ

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

Для проверки доступности сервера Docker с клиентской рабочей станции:

1. Подключитесь через RDP (или другим способом) к клиентской рабочей станции, на которой установлены приложение Microsoft Outlook и надстройка для Outlook.

В PowerShell ISE запустите диагностический скрипт из [Приложения 1 «Диагностический скрипт № 1»](#) и отредактируйте значения переменных: учетную запись пользователя Outlook, ссылку на Docker и токен, полученный из Keycloak. Токен Keycloak можно получить, запросив его у Администратора KeyCloak Заказчика.

2. Если скрипт отработал с ошибками, проанализируйте и устраните их причины. Вероятные причины проблемы в настройках Docker и приложения express-core-service: отсутствует сетевой доступ, проблемы аутентификации и авторизации. Перезапустите скрипт, пока он не отработает без ошибок.
3. Если скрипт отработал без ошибок и выдал ID конференции, которую можно найти в консоли администрирования CTS-сервера СК «Express», тогда считается, что Conference Notifier Bot доступен с сервера Docker.

Переходите к следующему шагу.

ШАГ 4. ОБЩАЯ ДИАГНОСТИКА НАДСТРОЙКИ ДЛЯ OUTLOOK

Если предыдущие 3 шага пройдены успешно, но надстройка для Outlook все равно не работает, проверьте корректность настроек в конфигурационном файле клиентского ПО (см. раздел «Шаг 3. Разворачивание в Docker express-core-service и template-core-service» настоящей инструкции).

Если конфигурационный файл клиентского ПО корректен, изучите файлы логов из клиентских приложений, описанные в разделе «Шаг 3. Разворачивание в Docker express-core-service и template-core-service» настоящей инструкции, и серверные файлы логов, описанные в разделе «Шаг 2. Подготовка БД». Устраните причины ошибок.

Если все перечисленные выше шаги не помогли, 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.

```
$headers = New-Object
[System.Collections.Generic.Dictionary[[String],[String]]
$headers.Add("Content-Type", "application/json")
$headers.Add("Authorization", токен)

# ниже заполнить создателя
$body = @"
{"Creator`":`"Пользователь_Экспесс@Ваш_домен.ru`",`"LanguageId`":1049,`
"isRecurrent`":false,`"$type`":`"MeetingCreateRequest`"}
"@

# ниже ввести адрес Docker express-core-service такаяже ссылка должна
# быть в настройках клиентской части настройки.

$response = Invoke-RestMethod 'http://Ваш_Docker:Ваш_port/api/meetings'
-Method 'POST' -Headers $headers -Body $body
$response | ConvertTo-Json
```

Приложение 2

ДОБАВЛЕНИЕ СЕРТИФИКАТА В КОНТЕЙНЕР EXPRESS-MEETING-CORE-SERVICE-MAIN

Данный раздел описывает типовое решение по установке сертификатов в контейнер Docker. Процедура может меняться в зависимости от инфраструктуры и архитектурного решения.

Важно! Для корректной работы SSL в контейнер `express-meeting-core-service-main` следует поместить все сертификаты в цепочки сертификатов (корневой сертификат, промежуточные сертификаты, конечный сертификат).

Для добавления сертификата в контейнер:

1. Соберите все сертификаты в цепочки в виде файлов формата CRT:

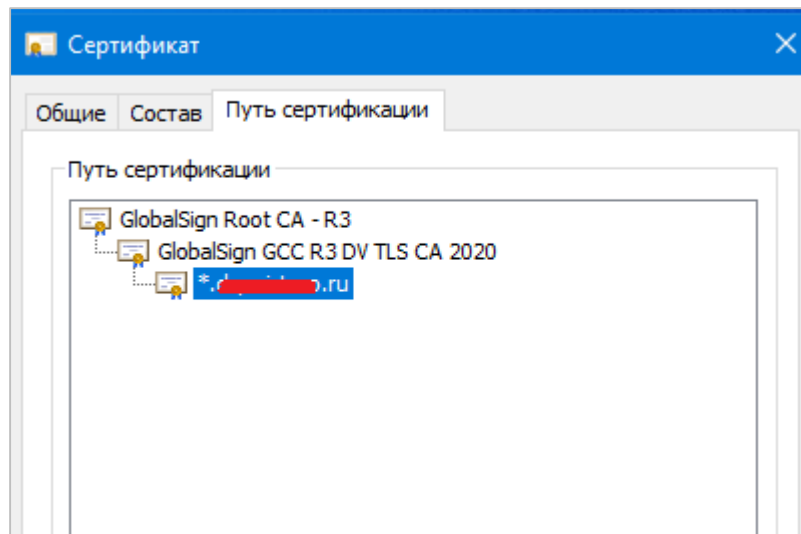


Рисунок 13

2. Создайте на сервере Docker, где развернут контейнер `express-meeting-core-service-main`, папку. Пример названия папки: `/opt/express/bots/cert/`.
3. Переместите файлы сертификатов, собранные в п. 1, в папку, созданную в п. 2.
4. Выдайте папке `/opt/express/bots/cert/` соответствующие права доступа.
5. В файле `docker-compose.yml` в параметрах контейнера `express-meeting-core-service-main` (Рисунок 14) добавьте в разделе `volumes` следующий параметр:

```
- /opt/express/bots/cert/:/usr/local/share/ca-certificates/
```

Важно! Первая часть строки `/opt/express/bots/cert/` может быть любой и должна соответствовать пути из п.2. Вторая часть данной строки `:/usr/local/share/ca-certificates/` постоянная, менять ее запрещено. Из указанной папки `express-meeting-core-service-main` берет все необходимые сертификаты.

```

GNU nano 6.2 /opt/express/bots/outlook3000/docker-compose.yml *
version: "3.3"

services:
  core-service:
    image: ${EXPRESS_MEETING_CORE_SERVICE_MAIN_IMAGE_NAME:?error}:3.0.0.0
    container_name: ${EXPRESS_MEETING_CORE_SERVICE_MAIN_IMAGE_NAME:?error}
    env_file: ${EXPRESS_MEETING_CORE_ENV_FILE:?error}
    ports:
      - "${EXPRESS_MEETING_CORE_SERVICE_PORT:?error}:7100" # При необходимости отредактируйте порт хоста (первый)
    restart: always
    extra_hosts:
      - "${EXPRESS_MEETING_EXTRA_HOST}"
      - "${EXPRESS_MEETING_EXTRA_HOST2}"
    volumes:
      - type: bind
        source: "${EXPRESS_MEETING_LOGS_PATH:?error}/express-meeting-core-service"
        target: "${EXPRESS_MEETING_DOCKER_LOGS_PATH:?error}"
      - /opt/express/bots/keycloak/cert/crt-cert/:usr/local/share/ca-certificates/
#       bind:
#         create_host_path: true

  template-service:
    image: ${EXPRESS_MEETING_TEMPLATE_SERVICE_MAIN_IMAGE_NAME:?error}:3.0.0.0
    container_name: ${EXPRESS_MEETING_TEMPLATE_SERVICE_MAIN_IMAGE_NAME:?error}
    env_file: ${EXPRESS_MEETING_TEMPLATE_ENV_FILE:?error}
    ports:
      - "${EXPRESS_MEETING_TEMPLATE_SERVICE_PORT:?error}:80" # При необходимости отредактируйте порт хоста (первый)
    restart: always
    volumes:
      - type: bind
        source: "${EXPRESS_MEETING_LOGS_PATH:?error}/express-meeting-template-service"
        target: "${EXPRESS_MEETING_DOCKER_LOGS_PATH:?error}"
#       bind:
#         create_host_path: true

```

Рисунок 14

- Остановите контейнеры надстройки Outlook, запустив из папки с конфигурационными файлами командой:

```
docker-compose down
```

- Запустите из папки docker-контейнеры:

```
docker-compose --env-file ./env -f ./docker-compose.yml up -d
```

- Зайдите в контейнер express-meeting-core-service-main с помощью команды:

```
docker exec -it express-meeting-core-service-main bash
```

- Обновите список сертификатов командой:

```
update-ca-certificates
```

- Выйдите из контейнера express-meeting-core-service-main с помощью команды:

```
exit
```

- Попробуйте создать конференцию с помощью клиентского ПО.

- Проверьте систему на отсутствие ошибок, связанных с SSL, с помощью команды:

```
docker logs --tail=200 -f express-meeting-core-service-main
```

Приложение 3

СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ

Таблица 1. Single CTS или разделенный CTS

№ п/п	Источник	Получатель	Порт и протокол	Описание взаимодействия
1	Внутренний пользователь (клиентский ПК)	express-core-service (docker Host)	7100\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с express-core-service001 на сервере docker Host (указан стандартный сетевой порт)
2	express-core-service (docker Host)	CTS Front (CTS Single)	443\TCP	Взаимодействие express-core-service на docker Host с API Conference Notifier Bot на сервере CTS Front (CTS Single)
3	Администратор template (клиентский ПК)	express-template-service (docker Host)	7200\TCP	Взаимодействие администратора Шаблонов с express-template-service001 на docker Host через веб-интерфейс (указан стандартный сетевой порт)
4	express-core-service (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-core-service001 на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
5	express-template-service (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-template-service на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
6	Внутренний пользователь (клиентский ПК)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
7	express-core-service (docker Host)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие express-core-service001 на docker Host с Keycloak на сервере Keycloak (указан стандартный сетевой порт)

Примечание. В таблице указаны стандартные сетевые порты для взаимодействия клиентских ПК, express-core-service и Keycloak (SSL настраивается дополнительно и не учитывается в данной таблице).

Таблица 2. Несколько CTS без ETS

№ п/п	Источник	Получатель	Порт и Протокол	Описание взаимодействия
1	Внутренний пользователь на CTS01 (клиентский ПК)	express-core-service001 (docker Host)	7100\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с express-core-service001 на docker Host (указан стандартный сетевой порт)
2	Внутренний пользователь на CTS02 (клиентский ПК)	express-core-service002 (docker Host)	7300\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS02 с express-core-service002 на docker Host (указан стандартный сетевой порт)
3	Администратор template (клиентский ПК)	express-template-service (docker Host)	7200\TCP	Взаимодействие администратора Шаблонов с express-template-service на docker Host через веб-интерфейс (указан стандартный сетевой порт)
4	express-core-service001 (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-core-service001 на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
5	express-template-service (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-template-service на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
6	express-core-service002 (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-core-service002 на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
7	express-core-service001 (docker Host)	CTS Front (CTS Single)01	443\TCP	Взаимодействие express-core-service001 на docker Host с API Conference Notifier Bot на сервере CTS Front (CTS Single)01
8	express-core-service002 (docker Host)	CTS Front (CTS Single)02	443\TCP	Взаимодействие express-core-service002 на docker Host с API Conference Notifier Bot на сервере CTS02 Front (CTS02 Single)02
9	express-core-service001 (docker Host)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие express-core-service001 на docker Host с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
10	Внутренний пользователь (клиентский ПК)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
11	express-core-service0021 (docker Host)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие express-core-service002 на docker Host с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
12	Внутренний пользователь (клиентский ПК)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS02 с Keycloak на сервере Keycloak (указан стандартный сетевой порт)

Примечание. В таблице указаны стандартные сетевые порты для взаимодействия клиентских ПК, express-core-service и Keycloak (SSL настраивается дополнительно и не учитывается в данной таблице).

Таблица 3. Несколько CTS с ETS

№ п/п	Источник	Получатель	Порт и Протокол	Описание взаимодействия
1	Внутренний пользователь на CTS1(клиентский ПК)	express-core-service (docker Host)	7100\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с express-core-service001 на docker Host (указан стандартный сетевой порт)
2	Внутренний пользователь на CTS2 (клиентский ПК)	express-core-service (docker Host)	7100\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS02 с express-core-service001 на docker Host (указан стандартный сетевой порт)
3	Администратор template (клиентский ПК)	express-template-service (docker Host)	7200\TCP	Взаимодействие администратора Шаблонов с express-template-service на docker Host через веб-интерфейс (указан стандартный сетевой порт)
4	express-core-service001 (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-core-service001 на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
5	express-template-service (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-template-service на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
6	express-core-service (docker Host)	ETS EXPRESS Server	443\TCP	Взаимодействие express-core-service на docker Host с API Conference Notifier Bot на сервере ETS EXPRESS Server
7	express-core-service (Docker Host)	ETS EXPRESS Server	443\TCP	Взаимодействие express-core-service001 на docker Host с API Conference Notifier Bot на сервере ETS EXPRESS Server
8	express-core-service001 (docker Host)	Функциональность Keycloak (docker Host)	8080\TCP	Взаимодействие express-core-service001 на docker Host с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
9	Внутренний пользователь (клиентский ПК)	Функциональность Keycloak (docker Host)	8080\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
10	Внутренний пользователь (клиентский ПК)	Функциональность Keycloak (docker Host)	8080\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS02 с Keycloak на сервере Keycloak (указан стандартный сетевой порт)

Примечания:

- В таблице указаны стандартные сетевые порты для взаимодействия клиентских ПК, express-core-service и Keycloak (SSL настраивается дополнительно и не учитывается в данной таблице).
- Сетевое взаимодействие между серверами EXPRESS ETS и серверами EXPRESS CTS подробно описано в «Express. Система коммуникаций. Руководство администратора. Том 1. Установка» https://express.ms/admin_guide_install.pdf.

Приложение 4

ТИПОВЫЕ ОШИБКИ

При возникновении проблем с конфигурацией подключения к БД PostgreSQL в логах появляется сообщение:

```
"An error occurred while processing your request.", "status": 500, "detail": "System.InvalidOperationException: An exception has been raised that is likely due to a transient failure.\n --> Npgsql.NpgsqlException (0x80004005)"
```

Возможное решение: отредактируйте в файле `appsettings.Production.json`⁷ параметры среды:

- пропишите `Minimum Pool Size=0`.
- если предыдущий пункт не помогает решить проблему, укажите `Pooling=False`.

Если ошибка сохраняется, обратитесь в техническую поддержку СК «Express».

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