

Инструкция по установке экземпляра программного обеспечения
программа для ЭВМ «Регистрация пользователей»

Оглавление

1. О данном документе.....	3
2. Требование к ресурсам ЭВМ	3
3. Требование к ПО.....	3
4. Подготовка к работе.....	4
4.1 Настройка БД PostgreSQL.....	4
4.2 Установка и настройка серверной части.....	4
4.3 Настройка клиентской части	6
4.4 Запуск	6
5. Сборка и запуск программного обеспечения в Docker	6

1. О данном документе

Данный документ регламентирует процесс установки экземпляра программного обеспечения программа для ЭВМ «Регистрация пользователей»

2. Требование к ресурсам ЭВМ

Для корректной работы программы для ЭВМ «Регистрация пользователей» применимы следующие минимальные требования:

- Операционная система: Windows, Linux, MacOS
- Установленное ПО: Node.js, .net 8
- Процессор: не слабее Intel Core i3
- Оперативная память: не менее 8 gb
- Жесткий диск: не менее 30 gb

3. Требование к ПО

Требования к ПО представлены в таблице (Таблица 1).

Таблица 1. Технические и эксплуатационные характеристики Программы

№ п/п	Программные средства	Наименование ПО
1	Способ распространения ПО	Размещение в инфраструктуре Заказчика
2	Базовая операционная система	Linux
		CentOS
		Astra Linux
3	Система виртуализации	VMWare
4	Балансировщик нагрузки	Nginx
		Ingress/Egress
5	Контейнеризация, оркестрация	Docker
		Kubernetes
		Helm
6	Облачное хранилище	minIO
	Веб-сервер/ балансировщик/	Nginx

№ п/п	Программные средства	Наименование ПО
7	прокси-сервер	Ingress/Egress
8	Программная платформа	Платформа с открытой лицензией .NET
9	СУБД	PostgreSQL
		Postgres Pro
		Redis версии 7.2
		OpenSearch
10	JavaScript	React
11	Microsoft.NET	C#

4. Подготовка к работе

4.1 Настройка БД PostgreSQL

Для запуска приложения требуется БД Postgresql с включенными подготовленными транзакциями. (Значение `max_prepared_transactions` в конфиге сервера БД больше нуля)

Команда изменения:

```
ALTER SYSTEM SET max_prepared_transactions = 100;
```

Команда просмотра значения:

```
SHOW max_prepared_transactions;
```

Redis версии 7.2, Opensearch (в случае выбора хранилища логов), Kafka.

4.2 Установка и настройка серверной части

Для начала работы приложения необходимо настроить авторизацию. Минимально необходимо иметь одну активную учетную запись с ролью "administrator". При необходимости ее можно переименовать - константа `CustomSecurityConstants.AdministratorRoleString`. Из коробки авторизацию можно настроить двумя способами:

- внутренний механизм авторизации, использующий Identity ASP.NET Core
- механизм авторизации, использующий IdentityServer.

Для использования Identity ASP.NET Core необходимо убедиться что в конфиге значение по ключу "Authentication" → "DisableAuthentication", "Authentication" → "ExternalIdentityServerEnabled" имеют значения false.

Для создания пользователя в БД можно воспользоваться скриптом (login: umdadmin, password: myadminpassword):

```
INSERT INTO "identity"."AspNetUsers" ("Id","UserName","NormalizedUserName","EmailConfirmed","AccessFailedCount") VALUES
```

```
('244223b0-cbd1-4cec-8bb9-7fe06a79a934','umdadmin','UMDADMIN','admin@example.com','ADMIN@EXAMPLE.COM','9c7ca5f0-6a12-4426-a3e4-14b74ce8fdd4','123','false,false,NULL,true,0);
```

```
INSERT INTO "identity"."AspNetRoles" ("Id","Name","NormalizedName","ConcurrencyStamp") VALUES
```

```
('38cfa248-188c-4a5c-87df-dbbe71d6a286','administrator','ADMINISTRATOR','c15c92');
```

```
INSERT INTO "identity"."AspNetUserRoles" ("UserId","RoleId") VALUES
```

```
('244223b0-cbd1-4cec-8bb9-7fe06a79a934','38cfa248-188c-4a5c-87df-dbbe71d6a286');
```

```
INSERT INTO "identity"."AspNetUserClaims" ("UserId","ClaimType","ClaimValue")
```

```
('244223b0-cbd1-4cec-8bb9-7fe06a79a934','given_name','Админ Петров'),
```

```
('244223b0-cbd1-4cec-8bb9-7fe06a79a934','family_name','Петров'),
```

```
('244223b0-cbd1-4cec-8bb9-7fe06a79a934','nickname','admin'),
```

```
('244223b0-cbd1-4cec-8bb9-7fe06a79a934','birthdate','2020-12-12'),
```

```
('244223b0-cbd1-4cec-8bb9-7fe06a79a934','profile','1'),
```

```
('244223b0-cbd1-4cec-8bb9-7fe06a79a934','name','Админ');
```

Выполнять необходимо в БД, которой будут храниться учетные записи пользователей (та что прописана в конфиге по ключу "Identity" → "DatabaseOptions" → "ConnectionString")

При необходимости пароль можно изменить с помощью вызова метода /identity/changepassword

```
curl -X 'POST' \  
http://localhost:5203/identity/changepassword \  
-H 'accept: */*' \  
-H 'Content-Type: application/json' \  
-d '{  
  "currentPassword": "",  
  "newPassword": ""
```

}'

Для использования IdentityServer необходимо убедиться что в конфиге значение по ключу "Authentication" → "DisableAuthentication" имеет значение false, "Authentication" → "ExternalIdentityServerEnabled" - значение true.

4.3 Настройка клиентской части

Необходимо задать корректное значение для следующих переменных окружения:

APOLLO_SERVER_URL - адрес backend graphql

REST_SERVER_URL - адрес backend сервера

IDENTITY_LOGIN - признак использования авторизации через identity (1 - да, 0 - нет)

4.4 Запуск

DEVELOPMENT режим

Выполнить в консоли из директории с исходным кодом frontend приложения:

- yarn
- yarn dev

Перейти в браузере на <http://localhost:3000>

PRODUCTION режим

Выполнить в консоли из директории с исходным кодом frontend приложения:

- yarn
- yarn build
- yarn start

Перейти в браузере на <http://localhost:3000>

5. Сборка и запуск программного обеспечения в Docker

Для сборки и разворачивания приложения в Docker, требуется установленный Docker в системе.

docker-compose файл отправляется на электронную почту по заявке системного администратора.

Для установки и обновления в среде оркестрации необходимо используя переданный helm-chart, подготовить конфигурационный файл values.yml с необходимыми параметрами, описанными ниже и выполнить команду:

```
helm upgrade --install isbox-prod charts/box-identity-server \
--namespace isbox-prod \
--create-namespace \
```

```
--values environments/prod/values.yaml \  
--set universal-chart.defaultImageTag=1.4.1
```

Описание и пример конфигурационного файла values.yml:

```
imagePullSecrets: ""  
nodeSelector: {}  
ingress: {}  
  
components:  
  server:  
    replicaCount: 1  
    image:  
      repository: "some.private.registry.ru/box/identity-server"  
      tag: "1.0.0"  
      pullPolicy: "Always"  
    resources:  
      requests:  
        cpu: "100m"  
        memory: "128Mi"  
      limits:  
        cpu: "500m"  
        memory: "512Mi"  
    cert:  
      fileName: cert.pfx  
      mountPath: "/app/certs"  
    env:  
      KESTREL_PORT: "8080" # Порт Kestrel  
      PATHBASE: "/" # Базовый путь приложения ("/path")  
      SERILOG_CONSOLE_WRITE: "Console" # // None|Console  
      SERILOG_FILE_WRITE: "None" # // None|File  
      SERILOG_ELASTIC_WRITE: "None" # None|Elasticsearch (для отключения None)  
      SERILOG_ELASTIC_NODE_URI: "" # Адреса узлов Elasticsearch  
      SERILOG_ELASTIC_NS_PREFIX: "" # Префикс имен индексов Elasticsearch  
      DB_IDENTITY_HOST: "" # Хост БД для IdentityServer  
      DB_IDENTITY_DBNAME: "" # Имя БД для IdentityServer  
      DB_IDENTITY_USER: "" # Пользователь БД для IdentityServer  
      DB_IDENTITY_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-  
standby, read-only, read-write, any)  
      DB_CFG_HOST: "" # Хост БД для конфигурации  
      DB_CFG_DBNAME: "" # Имя БД для конфигурации  
      DB_CFG_USER: "" # Пользователь БД для конфигурации  
      DB_CFG_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-  
standby, read-only, read-write, any)  
      DB_PERSISTEDGRANT_HOST: "" # Хост БД для PersistedGrant  
      DB_PERSISTEDGRANT_DBNAME: "" # Имя БД для PersistedGrant  
      DB_PERSISTEDGRANT_USER: "" # Пользователь БД для PersistedGrant  
      DB_PERSISTEDGRANT_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary,  
standby, prefer-standby, read-only, read-write, any)  
      DB_DATAPROTECTION_HOST: "" # Хост БД для DataProtection  
      DB_DATAPROTECTION_DBNAME: "" # Имя БД для DataProtection  
      DB_DATAPROTECTION_USER: "" # Пользователь БД для DataProtection  
      DB_DATAPROTECTION_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary,  
standby, prefer-standby, read-only, read-write, any)  
      IDENTITY_PASSWORD_MAX_AGE: "" # Максимальный срок действия пароля. 0 - для отключения  
контроля срока действия.
```

```
SESSION_IDLE_MAX_TIMEOUT: "" # Время простоя пользователя перед разлогиниванием
LOCKOUT_INACTIVE_USERS_TIMEOUT: "" # Время бездействия перед блокировкой
LOCKOUT_USERS_NEVER_LOGGED_IN: "false" # Если включено, блокирует пользователей, которые
никогда не входили
SIGNINGCREDENTIAL_SOURCETYPE: "" # Источник ключей: Development|Certificate|Jwk
SIGNINGCREDENTIAL_CERT_SOURCETYPE: "" # Store|File
SIGNINGCREDENTIAL_CERT_STORE_THUMBPRINT: "" # Отпечаток сертификата
SIGNINGCREDENTIAL_CERT_FILE_RELPATH: "" # Относительный путь к файлу сертификата (относительно
BaseDirectory).
SIGNINGCREDENTIAL_JWK_RELPATH: "" # Относительный путь к файлу JWK (если используется JWK)
DATAPROTECTION_ENABLE_WITH_CERTIFICATE: "" # Использовать сертификат для защиты ключей
DataProtection (true/false)
DATAPROTECTION_SOURCETYPE: "File" # Источник сертификата для DataProtection: Store|File
DATAPROTECTION_CERT_THUMBPRINT: "" # Отпечаток сертификата DataProtection (если Store)
DATAPROTECTION_FILE_RELPATH: "" # Относительный путь к файлу сертификата DataProtection (если
File)
EMAIL_CONFIRM_REQUIRED: "" # Требуется ли подтверждение email при входе
SEND_MAIL_SMTP_SERVER: "" # SMTP сервер
SEND_MAIL_PORT: "" # SMTP порт
SEND_MAIL_FROM_MAIL: "" # Почтовый адрес отправителя
SEND_MAIL_USERNAME: "" # Логин почтового сервера
SEND_MAIL_ENABLE_SSL: "" # Использовать SSL
SEND_MAIL_DISABLE_SSL_VALIDATION: "" # Отключение валидации SSL сертификатов при отправке
почты
SEND_MAIL_REALLY_SEND_EMAIL: "" # Если true отправлять email всегда на Debug почту
SEND_MAIL_DEBUG_EMAIL: "" # Debug почта для отладки
ESIA_SIGN_PROVIDER_URI: "" # URL сервиса для подписания запросов к ЕСИА
admin:
  replicaCount: 1
  image:
    repository: "harbor.ost.digital/obp/release/box/identity-server-admin"
    tag: "1.0.0"
    pullPolicy: "Always"
  resources:
    requests:
      cpu: "100m"
      memory: "128Mi"
    limits:
      cpu: "500m"
      memory: "512Mi"
  env:
    # 🍀 Основные настройки сервера
    KESTREL_PORT: "8080" # Порт Kestrel
    PATHBASE: "/admin" # Базовый путь приложения ("/path")

    # 🍀 Настройки логирования Serilog
    SERILOG_CONSOLE_WRITE: "Console" # // None|Console
    SERILOG_FILE_WRITE: "None" # // None|File
    SERILOG_EMAIL_WRITE: "" # None|OverriddenEmail (для отключения None)
    SERILOG_FROM_EMAIL: "" # Email-адрес отправителя логов
    SERILOG_TO_EMAIL: "" # Email-адрес получателя логов
    SERILOG_EMAIL_SERVER: "" # SMTP сервер для логов
    SERILOG_EMAIL_PORT: "" # SMTP порт для логов
    SERILOG_FROM_EMAIL_USERNAME: "" # Логин для SMTP сервера

    SERILOG_TELEGRAM_WRITE: "" # None|Telegram (для отключения None)
    SERILOG_TELEGRAM_CHAT_ID: "" # ID чата Telegram для логов

    # 🍀 Подключение к базам данных
```

```
DB_IDENTITY_HOST: "" # Хост БД для IdentityServer
DB_IDENTITY_DBNAME: "" # Имя БД для IdentityServer
DB_IDENTITY_USER: "" # Пользователь БД для IdentityServer
DB_IDENTITY_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-standby, read-only, read-write, any)

DB_CFG_HOST: "" # Хост БД для конфигурации
DB_CFG_DBNAME: "" # Имя БД для конфигурации
DB_CFG_USER: "" # Пользователь БД для конфигурации
DB_CFG_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-standby, read-only, read-write, any)

DB_PERSISTEDGRANT_HOST: "" # Хост БД для PersistedGrant
DB_PERSISTEDGRANT_DBNAME: "" # Имя БД для PersistedGrant
DB_PERSISTEDGRANT_USER: "" # Пользователь БД для PersistedGrant
DB_PERSISTEDGRANT_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-standby, read-only, read-write, any)

DB_DATAPROTECTION_HOST: "" # Хост БД для DataProtection
DB_DATAPROTECTION_DBNAME: "" # Имя БД для DataProtection
DB_DATAPROTECTION_USER: "" # Пользователь БД для DataProtection
DB_DATAPROTECTION_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-standby, read-only, read-write, any)

DB_ADMINLOG_HOST: "" # Хост БД для Admin Log
DB_ADMINLOG_DBNAME: "" # Имя БД для Admin Log
DB_ADMINLOG_USER: "" # Пользователь БД для Admin Log
DB_ADMINLOG_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-standby, read-only, read-write, any)

DB_ADMINAUDITLOG_HOST: "" # Хост БД для Admin Audit Log
DB_ADMINAUDITLOG_DBNAME: "" # Имя БД для Admin Audit Log
DB_ADMINAUDITLOG_USER: "" # Пользователь БД для Admin Audit Log
DB_ADMINAUDITLOG_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-standby, read-only, read-write, any)

# ♠ Настройки IdentityServer Admin
IDENTITY_SERVER_URI: "" # Базовый URL сервиса IdentityServer

# Env для identityserverdata.json
IDP_ADMIN_BASE_URL: ""
IDP_GRAPHQL_BASE_URL: ""
SG_BASE_URL: ""
graphql:
  replicaCount: 1
  image:
    repository: " harbor.ost.digital/obp/release/box/identity-server-graphql"
    tag: "1.0.0"
    pullPolicy: "Always"
  resources:
    requests:
      cpu: "100m"
      memory: "128Mi"
    limits:
      cpu: "500m"
      memory: "512Mi"
  env:
    KESTREL_PORT: "8080" # Порт Kestrel
    PATHBASE: "/graphql" # Базовый путь приложения ("/path")
```

```
SERIOLOG_CONSOLE_WRITE: "Console" # // None|Console
SERIOLOG_FILE_WRITE: "None" # // None|File
DB_IDENTITY_HOST: "" # Хост БД IdentityServer
DB_IDENTITY_DBNAME: "" # Имя БД IdentityServer
DB_IDENTITY_USER: "" # Пользователь БД IdentityServer
DB_IDENTITY_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-
standby, read-only, read-write, any)
OAUTH2_INTROSPECTION_AUTHORITY: "" # URL IdentityServer
OAUTH2_INTROSPECTION_CLIENT_ID: "" # ID клиента для OAuth2
integration:
  replicaCount: 1
  image:
    repository: "harbor.ost.digital/obp/release/box/identity-server-integration"
    tag: "1.0.0"
    pullPolicy: "Always"
  resources:
    requests:
      cpu: "100m"
      memory: "128Mi"
    limits:
      cpu: "500m"
      memory: "512Mi"
  env:
    KESTREL_PORT: "8080" # Порт Kestrel
    PATHBASE: "/integration" # Базовый путь приложения ("/path")
    SERIOLOG_CONSOLE_WRITE: "Console" # // None|Console
    SERIOLOG_FILE_WRITE: "None" # // None|File
    DB_IDENTITY_HOST: "" # Хост БД IdentityServer
    DB_IDENTITY_DBNAME: "" # Имя БД IdentityServer
    DB_IDENTITY_USER: "" # Пользователь БД IdentityServer
    DB_IDENTITY_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-
standby, read-only, read-write, any)
    DB_REBUS_HOST: "" # Хост БД Rebus
    DB_REBUS_DBNAME: "" # Имя БД Rebus
    DB_REBUS_USER: "" # Пользователь БД Rebus
    DB_REBUS_TARGET_SESSION: "any" # Тип подключения к серверу PostgreSQL (primary, standby, prefer-
standby, read-only, read-write, any)
    FS_REBUS_DIRECTORY: "" # Директория для файлового транспорта Rebus
    SEND_MAIL_SMTP_SERVER: "" # SMTP сервер
    SEND_MAIL_PORT: "" # SMTP порт
    SEND_MAIL_FROM_MAIL: "" # Почтовый адрес отправителя
    SEND_MAIL_USERNAME: "" # SMTP логин
    SEND_MAIL_ENABLE_SSL: "" # Использовать SSL
    SEND_MAIL_DISABLE_SSL_VALIDATION: "" # Отключение валидации SSL сертификатов при отправке
почты
    SEND_MAIL_REALLY_SEND_EMAIL: "" # Если true отправлять email всегда на Debug почту
    SEND_MAIL_DEBUG_EMAIL: "" # Debug почта для отладки
```