

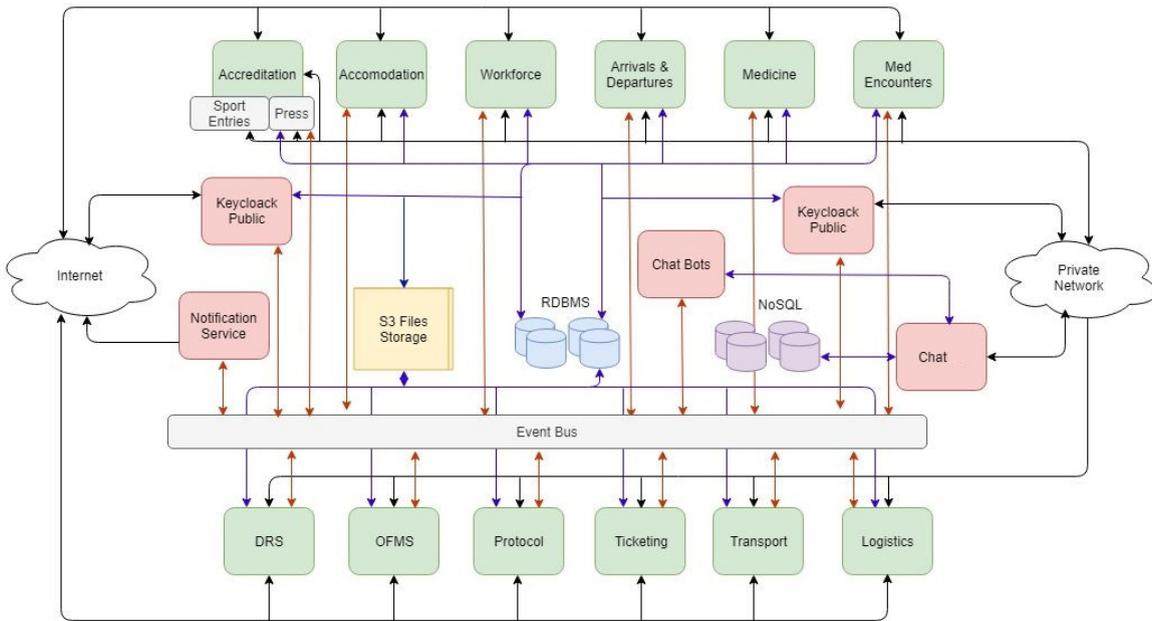
**Руководство по
установке платформы
LOOM**

Архитектура LOOM	4
Архитектура модуля LOOM	4
Описание модулей LOOM	8
Модули приложений (API):	8
Модули web клиентов:	9
Модули стриминга событий приложений в шину данных:	10
Схема информационных потоков	12
Языки программирования и фреймворки	12
Развертывание платформы	13
Прerequisites	13
Kubernetes Ingress Controller	13
Настройка параметров окружения	13
Сборка и развертывание модулей	14
Модуль Keycloak	14
Модуль Keycloak User Management	15
Модуль нотификаций	17
Модуль загрузки файлов	19
Модули отправки сообщений Kafka	20
Модуль администрирования словарей	20
Модуль Daily Run Sheet (DRS)	21
Модуль Логистика	22
Модуль Accommodation (ACM)	26
Модуль Прибытия и отъезды	27
Модуль Ticketing (Управление билетными квотами)	30
Модуль Аккредитация (ACR)	33
Модуль accreditation-search	37
Модуль интеграции с пограничными службами	38
Модуль Протокол (PRT)	39
Модуль Chromeheadless (HDS)	42
Модуль Transport-public (TRN-public)	43
Модуль Transport-private (TRN-private)	44
Модуль Workforce (WKF)	46
Модуль Overlay and Facility Management System (OFMS)	50
Модуль Equipment (EQP)	52
Модуль Medical Information System (MIS)	54
Модуль Medical Encounters (MES)	55
Модуль CHATBOT	58
Модуль TSRD (прием сообщений от сторонних систем)	59
Модуль TSRD (клиент для сайта)	62
Модуль Printable-Info (прием сообщений от сторонних систем)	62

Модуль Printable-Info (API для клиента)	64
Модули WEB-клиентов(<имя_модуля_клиента>-static)	64
Модуль S3Cache	65
Модуль S3Auth	66
Модуль Monitoring	67
Сборка клиентской части приложения	68
Модуль Аккредитация (ACR)	68
Модуль Билетная программа (TIC)	70
Модуль Логистика (MDS)	72
Модуль Медицинская инфо-система (MIS)	74
Модуль Медицинские обращения (MES)	76
Модуль Оснащение объектов (OFMS)	77
Модуль Персонал и волонтеры (WKF)	78
Модуль Приезды и отъезды (A&D)	80
Модуль Протокол (PRT)	83
Модуль Размещение (ACM)	85
Модуль Расписание работы объектов (DRS)	87
Модуль Спортивные заявки (SEQ)	88
Модуль Транспорт (TRN)	90
Модуль Экипировка (EQP)	93
Модуль Admin	94
Модуль Dashboard	95
Модуль Dictionaries	97
Загрузка клиентской части на GCS/S3	98
Требования к облачной платформе для LOOM	100

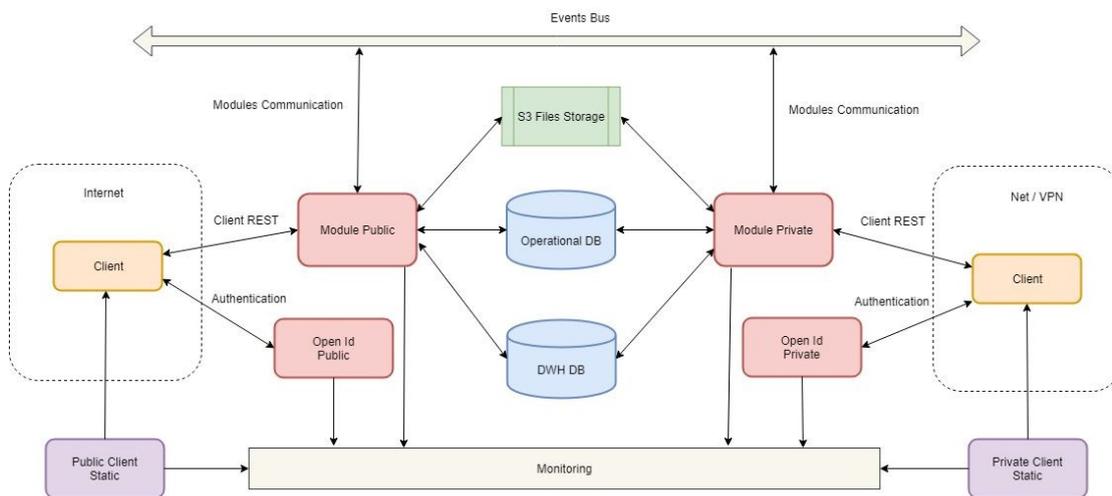
Архитектура LOOM

Общая схема:



Архитектура модуля LOOM

Общая схема:



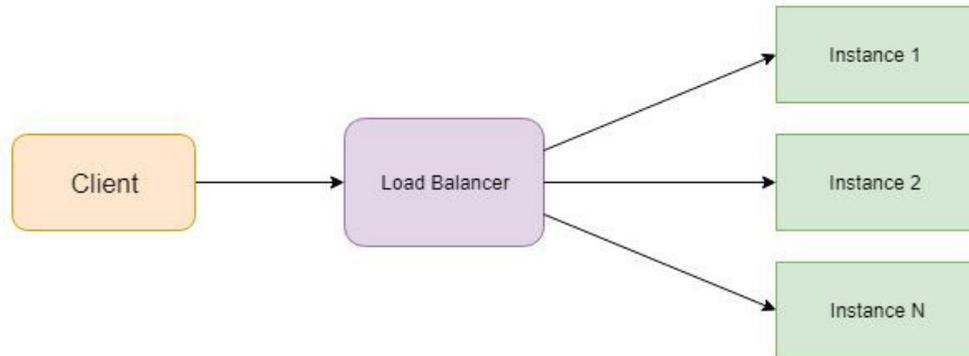
- 1) Module Public - публичная часть модуля доступная внешним пользователям и сторонним организациям через сеть интернет. Модуль предоставляет для клиента REST API и требует OpenID авторизацию для доступа к данным.
- 2) Module Private - приватная часть модуля доступная только для внутренних пользователей на территории РБ или через закрытый канал со стороны ответственной организации. Модуль предоставляет для клиента REST API и требует OpenID авторизацию для доступа к данным.
- 3) Open ID Public - публичный авторизационный Single sign-on (Open ID Connect) сервис хранящий данные о пользователях и осуществляющий аутентификацию пользователей работающих с публичной частью модуля.
- 4) Open ID Private - приватный авторизационный Single sign-on (Open ID Connect) сервис хранящий данные о пользователях и осуществляющий аутентификацию пользователей работающих с приватной частью модуля.
- 5) Public Client Static - компонент системы содержащий публичную часть клиента (статические файлы).
- 6) Private Client Static - компонент системы содержащий приватную часть клиента (статические файлы).
- 7) S3 Files Storage - распределенная система хранения файлов загружаемых и используемых в модуле.
- 8) Operational DB - база данных которая хранит текущее состояние данных в модуле.
- 9) DWH DB - база данных модуля которая хранит исторические данные и данные аудита.
- 10) Events Bus - шина данных которая позволяет осуществлять коммуникацию и обмен данными между модулями.
- 11) Monitoring - система мониторинга позволяющая осуществлять контроль за состоянием и работоспособностью модуля с помощью метрик собираемых с компонентов модуля.

OpenID — открытый стандарт децентрализованной системы аутентификации, предоставляющей пользователю возможность создать единую учетную запись для аутентификации на множестве связанных или не связанных друг с другом интернет ресурсов. В качестве реализации в системе LOOM используется Keycloak (<http://keycloak.jboss.org/>)— это open-source сервер аутентификации и управления учетными записями (IDM), построенный на базе спецификаций OAuth 2.0, Open ID Connect, JSON Web Token (JWT) и SAML 2.0.

В качестве реляционной базы данных в модуле используется свободно распространяемая объектно-реляционная система управления базами с открытым кодом PostgreSQL.

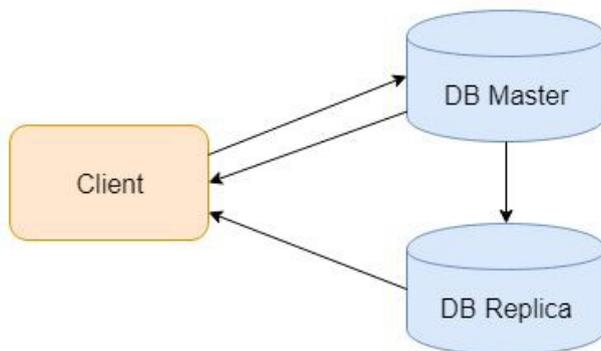
Module Private и Module Public представляет собой кластер из 2-х и более экземпляров серверов осуществляющих обработку запросов (Instance 1 ... Instance N), а так же балансировщика нагрузки HTTP(S) Load Balancer, который последовательно

отправляет запросы на разные экземпляры компонента системы, а также осуществляет контроль работоспособности каждого экземпляра (periodic health-check) и при обнаружении нерабочего сервера временно прекращает отправлять на него

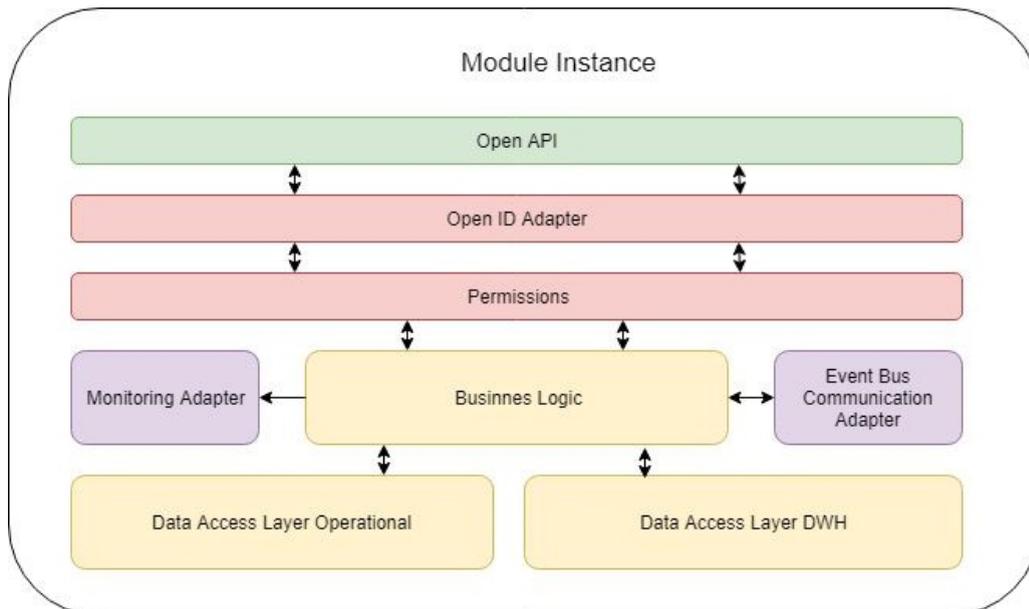


запросы.

Каждая СУБД имеет основной экземпляр сервера базы данных (DB Master), а также одну или несколько копий (DB Replica). В случае отказа основного экземпляра приложение продолжит работу с копией:



Внутренняя структура экземпляра Module Private и Module Public выглядит следующим образом:



- 1) Open API - предоставляет открытый интерфейс взаимодействия с клиентом через REST API (OpenAPI Specification).
- 2) Open ID Adapter - адаптер позволяющий проверить валидность авторизационных данных содержащихся в запросе клиента с помощью механизмов авторизации Open ID Connect.
- 3) Permissions - компонент проверяющий доступ авторизованного пользователя к соответствующему функционалу на уровне бизнес логики.
- 4) Business Logic - бизнес логика приложения
- 5) Monitoring Adapter - компонент модуля отвечающий за сбор и отправку данных в систему мониторинга
- 6) Event Bus Communication Adapter - компонент модуля отвечающий за взаимодействие с другими модулями LOOM через шину данных.
- 7) Data Access Layer Operational - компонент который отвечает за взаимодействие с базой данных которая хранит текущее состояние данных в модуле.
- 8) Data Access Layer DWH - компонент который отвечает за взаимодействие с базой данных модуля которая хранит исторический данные и данные аудита.

Описание модулей LOOM

Модули приложений (API):

- arrivals-and-departures-private - модуль для управления заявками на прибытие/отправление (создание/редактирование/подтверждение/отклонение). Модуль в закрытом контуре
- arrivals-and-departures-public - модуль для управление заявками клиентами. Модуль в открытом контуре.
- daily-run-schedule - расписание объектов
- ems-private-fileupload - загрузка файлов на private часть
- ems-private-keycloak - Keycloak на private части (авторизация пользователей)
- ems-private-notifications - нотификации на private части
- ems-public-fileupload - загрузка файлов на public часть
- ems-public-keycloak - Keycloak на private части (авторизация пользователей)
- ems-public-notifications - нотификации на public части
- ems-public-s3auth - модуль проверки доступа к скачиванию файлов, отдаваемых через модули s3cache-*
- keycloak-user-management - модуль создания и менеджмента пользователей в системе
- s3cache-private - модуль для скачивания файлов, загруженных через ems-private-fileupload сервис
- s3cache-public - модуль для скачивания файлов, загруженных через ems-public-fileupload сервис
- monitoring - внутренний мониторинг потребления событий с шины данных Kafka и отправка метрик в Telegraf
- accreditation-private - приватная часть модулей Accreditation, SEQ, PRESS
- accreditation-public - публичная часть модулей Accreditation, SEQ, PRESS
- logistics-private - приватная часть модуля Логистика (MDS). Создание, редактирование и согласование заявок на доставку. Регистрация пользователей внутри системы. Формирование, скачивание и рассылка сводных графиков поставки.
- logistics-public - публичная часть модуля Логистика (MDS). Создание и редактирование заявок на доставку. Публичная регистрация. Скачивание сводных графиков поставки.
- ticketing-private - модуль для распределения билетов организациям. Закрытый контур
- ticketing-public - модуль для создания заявок на билеты клиентами. Открытый модуль
- workforce-private - приватная часть модуля Workforce (WKF сервис управления трудовыми ресурсами).
- workforce-public - публичная часть модуля Workforce (WKF сервис управления трудовыми ресурсами).
- accomodation-private - приватная часть модуля Accommodation (ACM сервис размещения).

- accommodation-public - публичная часть модуля Accommodation (АСМ сервис размещения).
- ofms-private - модуль учета помещений, сервисов и оборудования на объектах
- Frisbee - приложение Frisbee (мессенджер).
- Frisbee-fileupload - модуль загрузки файлов для приложения Frisbee.
- s3cache-Frisbee - модуль чтения и кеширования файлов загруженных в приложении Frisbee.
- Frisbee-static - сервис для кеширования и отдачи web-клиента и desktop-клиента Frisbee.
- transport-private - печать и учет РВСТ, управление парком автомобилей, заказ gametaxi (индивидуального, массового)
- transport-public - открытый эндпоинт для получение координат от стороннего провайдера
- chromeheadless - конвертация html в pdf
- dictionary-private - централизованное управление общими справочниками всех модулей
- tsrd-private - модуль результатов соревнований (Timing Scoring Result Distribution)
- printable-info-private - модуль печатной информации
- medical-encounters - сервис для учета и управления медицинскими обращениями и услугами
- protocol-private - сервис для мониторинга и координации услуг, предназначенных для высокопоставленных гостей
- protocol-public - сервис для управления публичными приглашениями в рамках модуля protocol-private
- ems-private-chatbot - сервис чат-ботов для закрытого контура
- ems-public-chatbot - сервис чат-ботов публичный
- equipment-private - сервис экипировки

Модули web клиентов:

- accommodation-private-static - модуль кеширования и отдачи статических файлов модуля "Accommodation" на private части
- accommodation-public-static - модуль кеширования и отдачи статических файлов модуля "Accommodation" на public части
- accreditation-private-static - Модуль кеширования и отдачи статических файлов модуля "Accreditation" на private части
- accreditation-public-static - Модуль кеширования и отдачи статических файлов модуля "Accreditation" на public части
- arrivals-departures-private-static - модуль кеширования и отдачи статических файлов web-клиента arrivals and departures private
- arrivals-departures-public-static - модуль кеширования и отдачи статических файлов web-клиента arrivals and departures public
- dashboard-private-static - модуль кеширования и отдачи статических файлов модуля "Dashboard" на private части (только front-end часть)

- dashboard-public-static - модуль кэширования и отдачи статических файлов модуля "Dashboard" на public части (только front-end часть)
- dictionary-private-static - drs-private-static - модуль кэширования и отдачи статических файлов модуля "Dictionary" (редактирование списков и словарей)
- printable-info-static - модуль кэширования и отдачи статических файлов модуля печатной информации

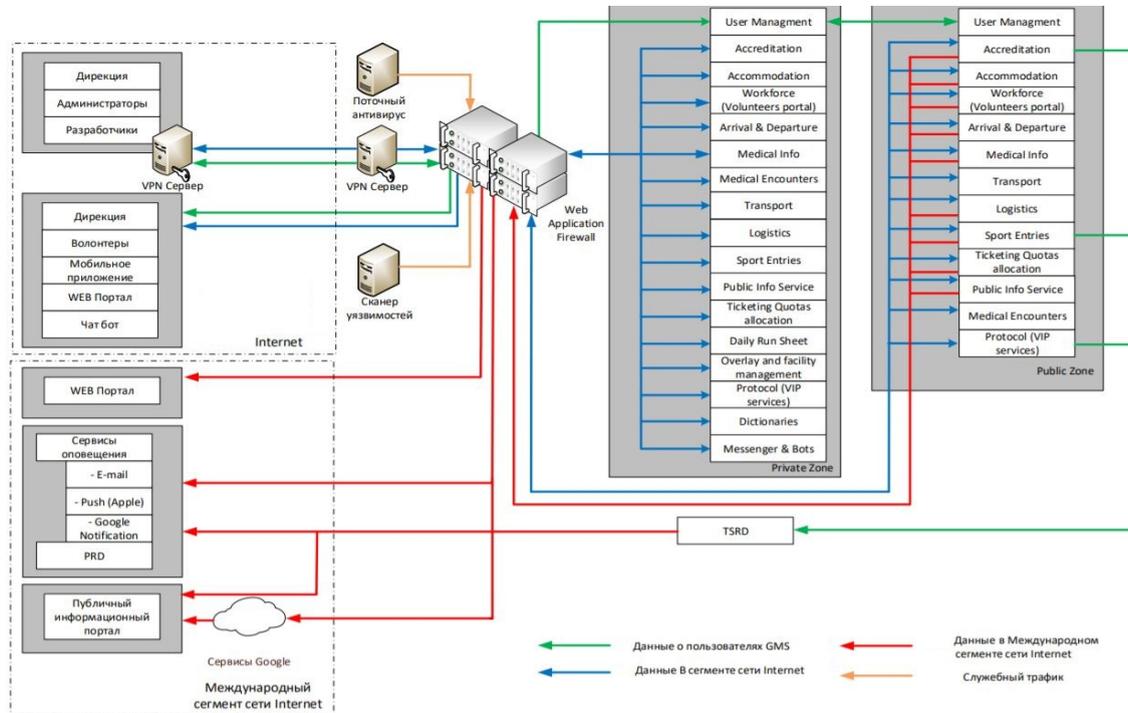
- logistics-private-static - модуль кэширования и отдачи статических файлов web-клиента модуля Logistics приватной части
- logistics-public-static - модуль кэширования и отдачи статических файлов web-клиента модуля Logistics публичной части
- mes-private-static - модуль кэширования и отдачи статических файлов web-клиента модуля Medical Encounters приватной части
- ofms-private-static - модуль кэширования и отдачи статических файлов модуля "OFMS"
- protocol-private-static - модуль кэширования и отдачи статических файлов web-клиента модуля Protocol приватной части
- protocol-public-static - модуль кэширования и отдачи статических файлов web-клиента модуля Protocol публичной части
- sportentries-private-static
- sportentries-public-static
- ticketing-private-static - модуль кэширования и отдачи статических файлов web-клиента ticketing private
- ticketing-public-static - модуль кэширования и отдачи статических файлов web-клиента ticketing public
- transport-private-static - модуль кэширования и отдачи статических файлов web-клиента модуля Transport приватной части
- um-private-static - модуль кэширования и отдачи статических файлов модуля "Keycloak User Management"
- workforce-private-static - модуль кэширования и отдачи статических файлов web-клиента workforce-private
- workforce-public-static - модуль кэширования и отдачи статических файлов web-клиента workforce-public
- equipment-private-static - модуль кэширования и отдачи статических файлов web-клиента для модуля экипировки

Модули стриминга событий приложений в шину данных:

- accommodation-sync-event-streamer - стример в шину данных событий из модуля accommodation
- accreditation-sync-event-streamer - стример в шину данных событий из модуля accreditation
- arrivals-departure-sync-event-streamer - стример в шину данных событий из модуля arrivals and departures

- dictionaries-sync-event-streamer - стример в шину данных событий из модуля dictionary
- drs-sync-event-streamer - стример в шину данных эвентов с модуля “Daily run schedule”
- logistics-sync-event-streamer стример в шину данных событий из модуля logistics
- medicine-sync-event-streamer - стример в шину данных событий из модуля medical-encounters
- ofms-sync-event-streamer - стример в шину данных событий из модуля overlay and facility
- protocol-sync-event-streamer - стример в шину данных событий из модуля protocol
- ticketing-sync-event-streamer - стример в шину данных событий из модуля ticketing
- transport-sync-event-streamer - стример в шину данных событий из модуля transport
- workforce-sync-event-streamer - стример в шину данных событий из модуля workforce

Схема информационных потоков



Языки программирования и фреймворки

Server:

Java, Spring Framework, Spring Boot, Spring Security, Spring Data (JPA, Hibernate) + QueryDSL, Swagger, Gradle

Web:

Javascript, TypeScript, Angular, Angular Universal, CSS, HTML

Mobile:

iOs: Swift 4.2, Alamofire, ReactiveSwift, ReactiveCocoa

Android: Kotlin, rxJava, Retrofit, Room

Развертывание платформы

Пререквизиты

- Kubernetes (далее k8s) кластер;
- Хранилище файлов совместимое с AWS S3 или с Google Cloud Storage;
- Кластер Kafka, доступный из k8s кластера;
- 22 кластера СУБД Postgres, доступные из k8s кластера;
- Для отправки писем потребуется аккаунт на mailgun.com;
- Опционально: сервис Telegraf для сбора метрик приложений;

Kubernetes Ingress Controller

В проекте используются k8s ingresses (правила маршрутизации трафика по сервисам kubernetes), в частности HAProxy Ingress Controller.

Настройка параметров окружения

Общие параметры для развертывания описываются в файле `env.yaml` в корне проекта. В документе описаны объекты-конфигурации для каждого окружения отдельно. Рассмотрим конфигурации на примере “preprod” кластера, развернутого в Google Cloud:

```
preprod: # название окружения
registry: # настройки Docker registry
  host: u.gr.io
  project: preprod
  username: _json_key
  password: file://preprod.gcr-pusher-key.json # может быть указан явно или
считываться из файла file://somefile.json
clusters: # указание kubernetes кластеров
- name: site # настройки кластера под сайт
  kubernetes:
    definitionsDir: k8s/site # не требует изменения
    context: gke_preprod_site # имя локальносуществующего k8s контекста
- name: vp # настройки кластера под волонтерский портал
  kubernetes:
    definitionsDir: k8s/site
    context: gke_preprod_ems-public # имя локальносуществующего k8s контекста
- name: ems-public # настройки публичного кластера
```

```
kubernetes:
  context: gke_sport-games-preprod_europe-west1-b_ems # имя локально
существующего k8s контекста
- name: ems-private # настройки приватного кластера

модулей (используется для построения URL модулей)
kubernetes:
  context: gke_sport-games-preprod_europe-west1-b_ems # имя локально
существующего k8s контекста
```

Сборка и развертывание модулей

В общем случае сборка и развертывание производится следующей командой:

```
./gradlew -p path/to/project clean build dockerPushImage-envName
deployToK8s-envName -PclusterName=clusterName
```

Где

- *path/to/project* - путь к собираемому модулю (пример: ems-common/notifications);
- *envName* - название окружения, сконфигурированного в *env.yaml* (пример: preprod);
- *clusterName* - название кластера (например: ems-private).

В случае использования Windows нужно запускать вместо *gradlew.sh* скрипт *gradlew.bat*.

Эта команда очистит папку с собранными артефактами (*clean*), соберет и протестирует проект (*build*), соберет и загрузит образ в указанный в *env.yaml* Docker registry (*dockerPushImage-envName*), применит сгенерированные k8s спецификации на заданном кластере (*deployToK8s-envName*).

В случае разворачивания системы без использования *kubernetes* можно собрать проект следующей командой:

```
./gradlew -p path/to/project clean build
```

Собранный *jar* файл можно найти в папке *build/libs*. Данный файл можно запускать как самостоятельное приложение с помощью команды "java -jar application.jar". Также необходимо будет задать параметры приложения через переменные окружения). Приложение будет обслуживать *http* запросы на порту 8080 (если не указано иное).

Модуль Keycloak

Ветка: *master* (здесь и далее будет указываться ветка *GIT* в которой находится рабочая версия модуля; в выгруженном исходном коде ветки соответствуют папкам в корне архива).

Модуль отвечает за управление пользователями системы и их авторизацию.

Развертывается отдельно как на приватную, так и на публичную часть.

Пример развертывания публичной части:

```
./gradlew -p keycloak clean build dockerPushImage-preprod deployToK8s-preprod  
-PclusterName=ems-public
```

Пример развертывания приватной части:

```
./gradlew -p keycloak clean build dockerPushImage-preprod deployToK8s-preprod  
-PclusterName=ems-private
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- `ems-private-keycloak` - основные настройки плагинов, в общем случае изменения не требуются;
- `ems-private-keycloak-service-account` - указывается `client id` и `client secret` сервисной учетной записи `keycloak`;
- `ems-private-keycloak-client` - настройки клиентов `keycloak` (указывается хост, по которому происходит авторизация, а также `client secret`);
- `postgres-ems-private-keycloak-secret` - параметры подключения к СУБД для приватной части.

Аналогичные `configmap/secrets` с префиксом `"ems-public"` существуют для настройки публичной части `Keycloak`.

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- `DB_VENDOR` - `"postgres"`;
- `DB_ADDR` - адрес хоста `Postgresql`;
- `DB_PORT` - порт `Postgresql`;
- `DB_USER` - пользователь `Postgresql`;
- `DB_PASSWORD` - пароль `Postgresql`;
- `DB_DATABASE` - имя базы `Postgresql`;
- `KEYCLOAK_USER` - начальное имя администратора `Keycloak`;
- `KEYCLOAK_PASSWORD` - начальный пароль администратора `Keycloak` (может быть изменен позже);
- `KAFKA_HOST` - хост для подключения к `Kafka`;
- `KAFKA_USERNAME` - имя пользователя `Kafka` (используется `SASL_PLAINTEXT` аутентификация), если необходимо;
- `KAFKA_PASSWORD` - пароль `Kafka`, если необходимо;
- `KAFKA_TOPIC` - название топика `Kafka` куда будут посылаться сообщения от `Keycloak`;
- `REALM_FILTER` - `"private"`;
- `DASHBOARD_URL` - URL приложения `"Dashboard"`;
- `PROXY_ADDRESS_FORWARDING` - `"true"`.

Модуль `Keycloak User Management`

Ветка: `master`.

Модуль отвечает за создание пользователей в `Keycloak` с заданными атрибутами и ролями. Разворачивается на приватной части, может получать доступ к как к приватной, так и к публичной версии `Keycloak`.

Пример развертывания:

```
./gradlew -p keycloak-user-management clean build dockerPushImage-preprod  
deployToK8s-preprod
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- keycloak-admin-client-private - хост, логин/пароль от административного аккаунта приватной части Keycloak;
- keycloak-admin-client-public - хост, логин/пароль от административного аккаунта публичной части Keycloak;

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST - URL keycloak приватной части;
- KC_ADMIN_USERNAME - имя пользователя-администратора приватной части;
- KC_ADMIN_PASSWORD - пароль пользователя-администратора приватной части;
- KC_AUTH_HOST_PUBLIC - URL keycloak публичной части;
- KC_ADMIN_PUBLIC_USERNAME - имя пользователя-администратора публичной части;
- KC_ADMIN_PUBLIC_PASSWORD - пароль пользователя-администратора публичной части;
- KC_SA_CLIENT_ID_PUBLIC - - "client id" сервисной учетной записи на публичной части;
- KC_SA_CLIENT_SECRET_PUBLIC - "client secret" сервисной учетной записи на публичной части;
- KC_SA_CLIENT_ID_PRIVATE - - "client id" сервисной учетной записи на приватной части;
- KC_SA_CLIENT_SECRET_PRIVATE - "client secret" сервисной учетной записи на приватной части;
- ISSUE_REPORT_EMAIL - адрес по которому происходит доставка сообщений о проблемах;
- SWAGGER_ENABLED - "true", если необходимо включить Swagger (автоматическая документация HTTP API);
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125");

Далее описаны переменные окружения, в которые необходимо передать внешний URL соответствующих приложений:

- ACM_REDIRECT_URL_PRIVATE
- ACM_REDIRECT_URL_PUBLIC
- ACR_REDIRECT_URL_PUBLIC
- ACR_REDIRECT_URL_PRIVATE
- AD_REDIRECT_URL_PUBLIC
- AD_REDIRECT_URL_PRIVATE
- DSH_REDIRECT_URL_PUBLIC
- DSH_REDIRECT_URL_PRIVATE
- DRS_REDIRECT_URL_PRIVATE
- MDS_REDIRECT_URL_PUBLIC
- MDS_REDIRECT_URL_PRIVATE
- MES_REDIRECT_URL_PRIVATE
- OFM_REDIRECT_URL_PRIVATE
- PRS_REDIRECT_URL_PUBLIC
- PRT_REDIRECT_URL_PUBLIC
- PRT_REDIRECT_URL_PRIVATE
- SEQ_REDIRECT_URL_PUBLIC
- SEQ_REDIRECT_URL_PRIVATE
- TIC_REDIRECT_URL_PUBLIC
- TIC_REDIRECT_URL_PRIVATE
- TRN_REDIRECT_URL_PUBLIC
- TRN_REDIRECT_URL_PRIVATE
- WKF_REDIRECT_URL_PUBLIC
- WKF_REDIRECT_URL_PRIVATE

Модуль уведомлений

Ветка: master.

Модуль необходим для отправки уведомлений, писем. Разворачивается отдельно для публичной и приватной части.

Команда для сборки приватной части:

```
./gradlew -p ems-common/notifications clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-private
```

Команда для сборки публичной части:

```
./gradlew -p ems-common/notifications clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-public
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets:

- mailgun - указывается domain-name, api-key от учетной записи mailgun, а также задается "sender-domain (см. описание одноименной переменной окружения)"
- postgres-ems-private-notifications-secret - параметры подключения к СУБД для приватной части;

- postgres-ems-public-notifications-secret - параметры подключения к СУБД для публичной части.

В случае запуска приложения без k8s кластера необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST - URL keycloak;
- MAILGUN_DOMAIN_NAME - доменное имя mailgun выдаваемое при регистрации;
- MAILGUN_API_KEY - ключ mailgun, выдаваемый при регистрации;
- EMAIL_SENDER_DOMAIN - доменное имя, которое модуль нотификаций будет подставлять в поле "from", если приложение его явно не специфицирует;
- CHATBOT_URL - URL приложения чатботов, куда следует слать уведомления;
- KC_SA_CLIENT_ID - "client id" сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в приложение чатботов;
- KC_SA_CLIENT_SECRET - "client secret" сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в приложение чатботов;
- KAFKA_HOST - адрес Kafka брокера для чтения очереди нотификаций, писем;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо;
- KAFKA_PASSWORD - пароль Kafka;
- KAFKA_TOPIC_SUFFIX - суффикс, который будет использоваться для формирования имени топика откуда будут читаться сообщения (для уточнения названий топиков см. Файл `ems-common/notifications/src/resources/application.yaml`, секция `app.kafka-listeners`);
- SWAGGER_ENABLED - "true", если необходимо включить Swagger (автоматическая документация HTTP API);
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: `"telegraf.kube-system.svc.cluster.local:8125"`);
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр;
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр.

Модуль загрузки файлов

Ветка: master.

Модуль необходим для загрузки файлов. Разворачивается отдельно для публичной и приватной части.

Команда для сборки приватной части:

```
./gradlew -p fileupload clean build dockerPushImage-preprod deployToK8s-preprod  
-PclusterName=ems-private
```

Команда для сборки публичной части:

```
./gradlew -p fileupload clean build dockerPushImage-preprod deployToK8s-preprod  
-PclusterName=ems-public
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets:

- `ems-private-fileupload` - configmap с настройками сервиса, указывается имя бакета, S3 endpoint и другие настройки.

В случае запуска приложения без k8s кластера необходимо задать следующие переменные окружения:

- `JAVA_OPTS` - опции jvm, опционально;
- `SPRING_PROFILES_ACTIVE` - активный профиль спринг который включает выбранный тип хранилища: "s3" или "gcloud";
- `GCS_SERVICE_ACCOUNT_JSON` - json-токен сервисной учетной записи для доступа к Google Cloud Storage (GCS), если последний используется в качестве хранилища;
- `GCS_BUCKET` - имя бакета в GCS, если последний используется в качестве хранилища;
- `S3_BUCKET` - имя бакета в S3, если последний используется в качестве хранилища;
- `S3_ENDPOINT` - хост S3 API сервера, если последний используется в качестве хранилища;
- `S3_ACCESS_KEY` - ключ доступа к S3, если последний используется в качестве хранилища;
- `S3_SECRET_KEY` - секретный ключ доступа к S3, если последний используется в качестве хранилища;
- `S3_PUBLIC_HOST` - публичный хост, который будет подставлен в формируемые модулем ссылки для скачивания;
- `KC_AUTH_HOST` - URL keycloak;
- `SWAGGER_ENABLED` - "true", если необходимо включить Swagger (автоматическая документация HTTP API);
- `METRICS_TELEGRAF_ENABLED` - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125");
- `DB_MAX_POOL_SIZE` - максимальное количество соединений в пуле базы данных, опциональный параметр;

- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр.

Модули отправки сообщений Kafka

Ветка: master

Модуль служит для отправки сообщений в Kafka, разворачивается по одному на базу приложения.

Команда для сборки:

```
./gradlew -p ems-common/sync-event-streamer clean build dockerPushImage-preprod  
deployToK8s-preprod -Dapp.name=$APP
```

В качестве параметра app указывается одно из приложений:

- accommodation
- accreditation
- arrivals-departure
- dictionaries
- drs
- logistics
- medicine
- notifications
- ofms
- protocol
- ticketing
- transport
- workforce

Таким образом, в результате развертывания должны быть запущены следующие сервисы:

- accommodation-sync-event-streamer
- accreditation-sync-event-streamer
- arrivals-and-departures-sync-event-streamer
- arrivals-departure-sync-event-streamer
- daily-run-schedule-sync-event-streamer
- dictionaries-sync-event-streamer
- drs-sync-event-streamer
- equipment-private-sync-event-streamer
- equipment-sync-event-streamer
- logistics-sync-event-streamer
- medicine-sync-event-streamer
- ofms-sync-event-streamer
- protocol-sync-event-streamer
- ticketing-sync-event-streamer
- transport-sync-event-streamer
- workforce-sync-event-streamer

Модуль администрирования словарей

Ветка: dictionary

Модуль отвечает за управления словарями - добавление/изменение/удаление элементов в словарях, экспортирование и импортирование элементов словарей в/из xlsx файлов.

Команда для сборки:

```
./gradlew -p ems-private/dictionary-private clean build dockerPushImage-preprod  
deployToK8s-preprod
```

Для запуска модуля необходимо сконфигурировать следующие k8s configmap/secrets:

- postgres-dictionaries-secret - параметры подключения к СУБД;
- ems-private-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- kafka - настройки соединения с кафкой

В случае запуска приложения без k8s кластера необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST - URL keycloak;
- SWAGGER_ENABLED - "true", если необходимо включить Swagger (автоматическая документация HTTP API);
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125");
- KAFKA_HOST - адрес Kafka брокера для чтения очереди нотификаций, писем;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо;
- KAFKA_PASSWORD - пароль Kafka;
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр;
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр.

Модуль Daily Run Sheet (DRS)

Ветка: daily-run-schedule.

Модуль управления расписанием. Разворачивается в приватной части.

Команда для сборки:

```
./gradlew -p ems-private/daily-run-schedule clean build dockerPushImage-preprod  
deployToK8s-preprod
```

Для запуска модуля необходимо сконфигурировать следующие k8s configmap/secrets:

- drs - настройки специфичные для DRS;
- postgres-drs-secret - параметры подключения к СУБД;

В случае запуска приложения без k8s кластера необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST - URL keycloak;
- KC_SA_CLIENT_ID - "client id" сервисной учетной записи с помощью которой происходит аутентификация для взаимодействия с другими сервисами (в данном случае notification service);
- KC_SA_CLIENT_SECRET - "client secret" сервисной учетной записи с помощью которой происходит аутентификация для взаимодействия с другими сервисами;
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125");
- EMAIL_LOGO - URL логотипа вставляемого в рассылаемые письма-нотификации;
- ISSUE_REPORT_EMAIL - email на который будут отсылаться сообщения о проблемах;
- KAFKA_HOST - адрес Kafka брокера для чтения очереди нотификаций, писем;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо;
- KAFKA_PASSWORD - пароль Kafka;
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр;
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр.

Модуль Логистика

Ветка: logistics

Модуль MDS - система для организации движения логистических грузов на объекты во время проведения мероприятий.

Развертывается отдельно как на приватную, так и на публичную часть.

Пример развертывания публичной части:

```
./gradlew :ems-public:logistics-public:clean
:ems-public:logistics-public:dockerPushImage-stage-1
:ems-public:logistics-public:deployToK8s-stage-1 -PclusterName=ems-public
```

Пример развертывания приватной части:

```
./gradlew :ems-private:logistics-private:clean
:ems-private:logistics-private:dockerPushImage-stage-1
:ems-private:logistics-private:deployToK8s-stage-1 -PclusterName=ems-private
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- postgres-logistics-secret – настройки подключения к бд;
- database-common – настройки бд
- ems-public-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- ems-private-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- ems-private-keycloak-service-account - указывается client id и client secret сервисной учетной записи keycloak;
- ems-public-keycloak-service-account - указывается client id и client secret сервисной учетной записи keycloak;
- frontend-info – содержит приватные/публичные ссылки модулей
- logistics – настройки модуля Логистика
- metrics – настройки метрик

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST - URL keycloak

- KC_CLIENT_SECRET – secret key для private
- KC_SA_CLIENT_ID_PUBLIC - “client id” публичной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений;

- KC_SA_CLIENT_SECRET_PUBLIC - “client secret” публичной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений
- KC_SA_CLIENT_ID_PRIVATE - “client id” приватной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений;
- KC_SA_CLIENT_SECRET_PRIVATE - “client secret” приватной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений
- KC_SA_CLIENT_AUTH_HOST_PRIVATE – auth host для приватной сервисной учетной записи
- KC_SA_CLIENT_AUTH_HOST_PUBLIC - auth host для публичной сервисной учетной записи
- NOTIFICATIONS_HOST_PUBLIC – публичный url модуля Уведомлений
- NOTIFICATIONS_HOST_PRIVATE – приватный url модуля Уведомлений
- EMAIL_LOGO – ссылка на логотип письма
- FRONTEND_PRIVATE_LOGISTICS_URL – приватный url модуля Логистики
- FRONTEND_PUBLIC_LOGISTICS_URL – публичный url модуля Логистики
- MDS_ENABLE_REQUESTS_CANCELLING – автоматическое аннулирование заявок в заданное время
- MDS_ENABLE_NOTIFICATION – флажок автоматической отправки уведомления менеджеру MDS о необходимости согласовать заявки
- KC_UM_CLIENT_ID - “client id” сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль KUM
- KC_UM_CLIENT_SECRET - “client secret” сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль KUM
- ISSUE_REPORT_EMAIL – email на который будут присылаться issue репорты
- MDS_CURRENT_TIME_ZONE – timezone модуля Логистики
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- METRICS_TELEGRAF_ENABLED - “true”, если необходимо включить отсылку метрик приложения (адрес задан явно: “telegraf.kube-system.svc.cluster.local:8125”);
- RECAPTCHA_PRIVATE_KEY – приватный ключ RECAPTCHA

Зависимости:

- KUM
- Notification
- RECAPTCHA

Модуль Accommodation (ACM)

Ветка: accommodation

Accommodation - система для обработки заявок на расселение гостей в отелях.

Развертывается отдельно как на приватную, так и на публичную часть.

Получить артефакт приватной части:

```
./gradlew :ems-private:accommodation-private:clean build
```

Получить артефакт публичной части:

```
./gradlew :ems-public:accommodation-public:clean build
```

Зависимости на внутренние модули:

- keycloak-user-management
- notification-service
- fileupload

Зависимости на сторонние сервисы:

- Google Distance Matrix Service

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST_PUBLIC - URL публичного keycloak
- KC_AUTH_HOST_PRIVATE - URL приватного keycloak;
- KC_SA_CLIENT_ID_PUBLIC - "client id" публичной учетной записи с помощью которой происходит аутентификация;
- KC_SA_CLIENT_ID_PRIVATE - "client id" приватной учетной записи с помощью которой происходит аутентификация;
- KC_SA_CLIENT_SECRET_PUBLIC - "client secret" публичной учетной записи с помощью которой происходит аутентификация;
- KC_SA_CLIENT_SECRET_PRIVATE - "client secret" приватной учетной записи с помощью которой происходит аутентификация;
- SITE_PUBLIC_URL - url по которому доступен UI для публичной части;
- SITE_PRIVATE_URL - url по которому доступен UI для приватной части;
- S3_PUBLIC_HOST - хранилище для статического контента для публичной части;
- S3_PRIVATE_HOST - хранилище для статического контента для приватной части;

- ISSUE_REPORT_EMAIL - email на который будут отсылааться сообщения о проблемах;
- SWAGGER_ENABLED - "true", если необходимо включить Swagger (автоматическая документация HTTP API);
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125");
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр;
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- GOOGLE_API_KEY - ключ для доступа к расчету расстояний с помощью google maps;
- GOOGLE_DISTANCE_SERVICE_ENABLED - "true", если необходимо включить расчет расстояний с помощью google maps, иначе расчет будет произведен геометрическим путем;
- NOTIFICATIONS_PUBLIC_URL - url для доступа к публичной части notification-service;
- DWH_POSTGRES_ADDR - адрес Postgresql для записи аудита изменений в системе;
- DWH_POSTGRES_PORT - порт Postgresql для записи аудита изменений в системе;
- DWH_POSTGRES_USER - пользователь Postgresql для записи аудита изменений в системе;
- DWH_POSTGRES_PASSWORD - пароль Postgresql для записи аудита изменений в системе;
- DWH_POSTGRES_DATABASE - имя базы данных Postgresql для записи аудита изменений в системе.

Модуль Прибытия и отъезды

Ветка: arrivals-and-departures

Модуль A&D - система для обеспечения координации процесса прибытия и отъезда представителей всех клиентских групп, управление соответствующими мероприятиями в аэропорту; мониторинг процессов прибытия и отъездов.

Развертывается отдельно как на приватную, так и на публичную часть.

Пример развертывания публичной части:

```
./gradlew :ems-public:arrivals-and-departures-public:clean
```

```
:ems-public:arrivals-and-departures-public:dockerPushImage-stage-1
```

```
:ems-public:arrivals-and-departures-public:deployToK8s-stage-1 -PclusterName=ems-public
```

Пример развертывания приватной части:

```
./gradlew :ems-private:arrivals-and-departures-private:clean  
:ems-private:arrivals-and-departures-private:dockerPushImage-stage-1  
:ems-private:arrivals-and-departures-private:deployToK8s-stage-1  
-PclusterName=ems-private
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- postgres-arrivals-diparture-secret – настройки подключения к бд;
- database-common – настройки пула соединений с базой в приложении
- ems-public-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- ems-private-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- ems-private-keycloak-service-account - указывается client id и client secret сервисной учетной записи keycloak;
- ems-public-keycloak-service-account - указывается client id и client secret сервисной учетной записи keycloak;
- frontend-info – содержит приватные/публичные ссылки модулей
- arrivals-departures – специфичные настройки модуля Прибытия и отъезды
- metrics – настройки метрик
- swagger – настройки для отключения/включения swagger ui
- \${clusterName}-fileupload – специфичные для кластера настройки fileupload'a. clusterName указывается при gradle сборке -PclusterName
- kafka - параметры подключения к Kafka

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- KC_CLIENT_SECRET – secret key для private
- KC_SA_CLIENT_ID_PUBLIC - “client id” публичной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль нотификаций;
- KC_SA_CLIENT_SECRET_PUBLIC - “client secret” публичной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для отправки сообщений в модуль нотификаций
- KC_SA_CLIENT_ID_PRIVATE - “client id” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для отправки сообщений в модуль нотификаций;
- KC_SA_CLIENT_SECRET_PRIVATE - “client secret” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для отправки сообщений в модуль нотификаций
- KC_SA_CLIENT_AUTH_HOST_PRIVATE – auth host для приватной сервисной учетной записи в keycloak
- KC_SA_CLIENT_AUTH_HOST_PUBLIC - auth host для публичной сервисной учетной записи в keycloak

- KAFKA_HOST - адрес Kafka брокера для чтения очереди нотификаций, писем;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо;
- KAFKA_PASSWORD - пароль Kafka;
- KAFKA_TOPIC_SUFFIX - суффикс, который будет использоваться для формирования имени топика откуда будут читаться сообщения (для уточнения названий топиков см. Файл `ems-common/notifications/src/resources/application.yaml`, секция `app.kafka-listeners`);
- SITE_PUBLIC_URL - url public части приложения. Подставляется в `redirect_uri` параметр `keycloak` при регистрации пользователя в public части
- SITE_PRIVATE_URL - url private части приложения. Подставляется в `redirect_uri` параметр `keycloak` при регистрации пользователя в private части
- FILE_UPLOAD_HOST - url для api `fileupload`'a
- DAILY_REPORT_EMAILS - включает/выключает отправку ежедневных отчетов на email
- DAILY_REPORT_CRON_EXPRESSION - крон выражение, используемое `quartz`, для отправки ежедневных отчетов по расписанию
- GLOBAL_SETTINGS_START - параметр, по которому юай открывает нужное число в календаре. Задается в ms. Опциональный параметр.
- GLOBAL_SETTINGS_REQUEST_BULK_CREATE_LIMIT - максимальное количество людей, на которых можно создавать заявку. Опциональный параметр. По умолчанию - 200
- ISSUE_REPORT_EMAIL - email на который будут отсылаться сообщения о проблемах;
- KEYCLOAK_PRIVATE_KAFKA_TOPIC - топик `kafka`, в который записываются сообщения о событиях в `keycloak-private`. Приложение слушает этот топик и автоматический создает, обновляет, удаляет пользователей в приложении
- KEYCLOAK_PUBLIC_KAFKA_TOPIC - топик `kafka`, в который записываются сообщения о событиях в `keycloak-public`. Приложение слушает этот топик и автоматический создает, обновляет, удаляет пользователей в приложении
- EMAIL_SENDER_DOMAIN - домен от которого отправляются email'ы
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125"); Рекомендуется всегда выставлять true из-за особенностей работы `io.micrometer` во избежание OOM

Зависимости:

- KUM
- Notification
- Kafka
- Fileupload

Модуль Ticketing (Управление билетными квотами)

Ветка: ticketing

Модуль TIC – позволяет управлять билетными квотами и распределять их между клиентскими группами, с целью дальнейшей реализации билетов, а также достижения заполненности трибун на соревновательных объектах. Имеет приватную и публичную части.

Пример развертывания публичной части:

```
./gradlew -p ems-public/ticketing-public clean dockerPushImage-stage-1  
deployToK8s-stage-1
```

Пример развертывания приватной части:

```
./gradlew -p ems-private/ticketing-private clean dockerPushImage-stage-1  
deployToK8s-stage-1
```

Конфигурация приложения описана в следующих k8s configmaps / secrets:

- postgres-ticketing-secret – настройки подключения к бд;
- database-common – настройки бд;
- ems-public-keycloak-client – настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- ems-private-keycloak-client – настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- ems-private-keycloak-service-account - указывается client id и client secret сервисной учетной записи keycloak;
- ems-public-keycloak-service-account - указывается client id и client secret сервисной учетной записи keycloak;
- frontend-info – содержит приватные/публичные ссылки модулей;
- swagger – позволяет включить или отключить Swagger API;
- ems-private-fileupload – хранит ссылку на хост fileupload сервиса в private;
- ems-public-fileupload – хранит ссылку на хост fileupload сервиса в public;
- Metrics – настройки метрик;
- kafka - настройки Kafka;
- ticketing – настройки модуля Ticketing;

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_CLIENT_SECRET – secret key для private;

- KC_AUTH_HOST_PRIVATE - URL приватного keycloak ;
- KC_SA_CLIENT_ID_PRIVATE - "client id" приватной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений;
- KC_SA_CLIENT_SECRET_PRIVATE - "client secret" приватной учетной записи с помощью которой происходит аутентификация;
- KC_SA_CLIENT_ID_PUBLIC - "client id" публичной учетной записи с помощью которой происходит аутентификация;
- KC_SA_CLIENT_SECRET_PUBLIC - "client secret" публичной учетной записи с помощью которой происходит аутентификация;
- KC_AUTH_HOST_PUBLIC - URL keycloak public части;
- SITE_PUBLIC_URL - url public части приложения. Подставляется в redirect_uri параметр keycloak при регистрации пользователя в public части;
- SWAGGER_ENABLED - "true", если необходимо включить Swagger (автоматическая документация HTTP API);
- FILE_UPLOAD_HOST - Fileupload API URL;
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125");
- DAILY_REPORT_ENABLED - "true" включает ежедневную рассылку отчетов о заявках;
- DAILY_REPORT_EMAILS – список email-адресов для получения ежедневной рассылки о заявках;
- DAILY_REPORT_CRON_EXPRESSION – CRON-выражение для задания расписания рассылки;
- KAFKA_HOST - адрес Kafka-брокера для чтения/записи сообщений;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо;
- KAFKA_PASSWORD - пароль Kafka;
- ISSUE_REPORT_EMAIL – email службы поддержки, на который будут отсылаться сообщения о проблемах от пользователей;
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр;
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- GLOBAL_SETTINGS_START – параметр (TIMESTAMP), по которому задается день начала мероприятия;
- GLOBAL_SETTINGS_START_LIMIT_MS – Минимальное время до начала спортивного события, которое позволяет клиенту создать заявку на билеты;
- EMAIL_SENDER_DOMAIN - домен с которого отправляются email'ы;
- PRIVATE_NOTIFICATION_EMAIL – email на который приходят уведомления о создании/обновлении заявок на билеты для мероприятий;

Зависимости:

- KUM
- Notification
- Dictionary Service
- Kafka
- Fileupload

Модуль Аккредитация (ACR)

Ветка: accreditation

Модуль Accreditation отвечает за процесс регистрации и доступ к объектам клиентских групп, которые участвуют в мероприятии, за исключением зрителей. Также в этом же модуле сосуществуют модуль SEQ (Система, позволяющая принимать заявки (entry form) от спортивных делегаций (НОКов) из любой точки мира в реальном времени с интеграцией данной платформы и базой данных по аккредитации (ACR), размещению и транспорту.

Система также предназначена для ведения квалификационных требований к отбору участников мероприятия), аккредитация прессы с публичной регистрацией и подмодуль рейткарт - тарифный справочник.

Развертывается отдельно на приватную и публичную части.

Пример развертывания публичной части:

```
./gradlew -p ems-public/accreditation-public clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-public
```

Пример развертывания приватной части:

```
./gradlew -p ems-private/accreditation-private clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-private
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- postgres-accreditation-secret – настройки подключения к бд;
- database-common – настройки бд
- ems-public-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- ems-private-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- ems-private-keycloak-service-account - указывается client id и client secret сервисной учетной записи keycloak;
- ems-public-keycloak-service-account - указывается client id и client secret сервисной учетной записи keycloak;
- frontend-info – содержит приватные/публичные ссылки модулей
- accreditation – настройки модуля Аккредитация
- metrics – настройки метрик

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

Приватная часть:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST - URL keycloak
- KC_AUTH_HOST_PRIVATE - ...
- KC_CLIENT_SECRET - secret key для private
- FILE_UPLOAD_HOST - URL сервиса File Upload
- FRONTEND_PUBLIC_ACCREDITATION_URL - публичный url модуля Аккредитации
- KC_SA_CLIENT_ID_PUBLIC - "client id" публичной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль;
- KC_SA_CLIENT_SECRET_PUBLIC - "client secret" публичной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль
- KC_SA_CLIENT_AUTH_HOST_PUBLIC - auth host для публичной сервисной учетной записи
- KC_SA_CLIENT_ID_PRIVATE - "client id" приватной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль;
- KC_SA_CLIENT_SECRET_PRIVATE - "client secret" приватной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль
- KC_SA_CLIENT_AUTH_HOST_PRIVATE - auth host для приватной сервисной учетной записи
- JOB_SCHEDULING_ENABLED - "true", если необходимо включить запуск запланированных задач (отправка уведомлений, автоматическое включение/отключение функциональности, пересчет квот, выгрузка фотографий и тд)
- NOTIFICATION_JOB_CRON - cron расписание отправки ACR уведомлений (об изменениях форм)
- GUEST_PASS_PRIVATE_NOTIFICATION_JOB_CRON - cron расписание отправки уведомлений по гест пассам приватным пользователям
- GUEST_PASS_PUBLIC_NOTIFICATION_JOB_CRON - cron расписание отправки уведомлений по гест пассам публичным пользователям

- FINAL_SUBMISSION_SYNC_JOB_CRON - cron расписание синхронизации информации по final submission с SEQ
- SHORT_LIST_ENTRY_QUOTA_RECALCULATING_JOB_CRON - cron расписание пересчета квот для NOC Short List
- PHOTOS_DUMP_UPDATE_JOB_CRON - cron расписание для обновления архива фотографий аккредитуемых лиц, который используется в мобильном клиенте.
- ISSUE_REPORT_EMAIL – email на который будут присылаться issue репорты
- PRESS_EMAIL_LOGO – url логотип, который отображается в отчетах;
- MEDIA_EMAIL - адрес для уведомлений пресс службы;
- RATE_CARD_EMAIL - адрес для уведомлений о сформированном заказе в тарифном справочнике;
- STORE_EXTERNAL_EMAILS - внешние адреса для уведомлений о сформированном заказе в тарифном справочнике;
- METRICS_TELEGRAF_ENABLED - “true”, если необходимо включить отсылку метрик приложения
- KC_UM_CLIENT_ID - “client id” сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль KUM
- KC_UM_CLIENT_SECRET - “client secret” сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль KUM
- FRONTEND_PUBLIC_PRESS_URL - публичный url для пресс части аккредитации
- PRS_NOTIFICATION_JOB_CRON - крон, определяющий частоту отправки писем службе безопасности Президента о количестве непросмотренных заявок
- PRS_PRESIDENT_SECURITY_NOTIFICATION_START_DATE - дата, с которой начинается рассылка уведомлений службе безопасности Президента
- ACR_NOTIFICATION_FROM_EMAIL - from адрес ACR уведомлений (email)
- AUTHORIZED_DOWNLOAD - Параметр включает авторизацию при скачивании файлов с S3. Если используется хранилище Google или AWS, параметр должен быть выставлен в **false**. При использовании приватного хранилища, которое требует авторизацию, параметр должен быть выставлен в true. В этом случае при скачивании файлов (бэкендом) будет передаваться Authorization заголовок с токеном системного юзера.
- KAFKA_HOST - адрес Kafka брокера;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо;
- KAFKA_PASSWORD - пароль Kafka;
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- guest.pass.timeZone - тайм зона, значение по-умолчанию +3. Задается в двойных кавычках (например “+5”)

- `guest.pass.request.createUntilTime` - время с которого создание гест пассов на следующий день запрещено, по-умолчанию 15:00. Формат "HH:mm:ss".
- `pass-card.restrictions.startDate` - дата с которой доступно выделение квот для гест пассов. Формат уууу-ММ-dd
- `pass-card.restrictions.endDate` дата до которой доступно выделение квот для гест пассов, по-умолчанию. Формат уууу-ММ-dd

Публичная часть:

параметр должен быть выставлен в true. В этом случае при скачивании файлов (бэкендом) будет передаваться Authorization заголовок с токеном системного юзера.

- `DB_MAX_POOL_SIZE`
- `DB_MIN_IDLE`

Для изменения начального диапазона действия аккредитационных карт необходимо выполнить скрипт на базе аккредитации:

```
UPDATE acr_date_range SET start_date = 'START_DATE', end_date = 'END_DATE'
WHERE date_range_set_id = 1;
```

Где START_DATE и END_DATE даты в формате:

`"YYYY-MM-DD HH24:MI:SS+TIMEZONE_OFFSET"`

Зависимости приватной части:

- KUM
- Notification
- Fileupload
- Chromeheadless (Генерация PDF образов карт)
- Dictionaries

Зависимости публичной части:

- KUM
- Notification
- Fileupload
- Dictionaries

Загрузка начальных данных

Для начальной загрузки данных (Объекты, роли, категории ролей) можно использовать скрипты, расположенные в модуле `accreditation-common/src/main/resources/demodata`

Модуль accreditation-search

Ветка: accreditation-search.

Модуль предоставляет API для поиска аккредитационных форм.

Разворачивается в приватной части.

Команда для сборки приватной части:

```
./gradlew -p ems-private/accreditation-search-private clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:Secrets:

- postgres-accreditation-secret - параметры подключения к СУБД;
- ems-private-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);

Configmaps:

- metrics - настройки метрик;

В случае запуска приложения без k8s кластера необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST - URL keycloak
- KC_CLIENT_SECRET - secret key для private
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125");

Модуль интеграции с пограничными службами

Ветка: accreditation-search.

Модуль предоставляет API для поиска и выгрузки данных по аккредитационным формам и билетам. Информация по аккредитационным формам предоставляется модулем Аккредитация. Информация по билетам предоставляется сторонним сервисом (модуль rf-integration выступает в качестве прокси). Разворачивается в приватной части.

Команда для сборки приватной части:

```
./gradlew -p ems-private/rf-integration-private clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:
Secrets:

- postgres-accreditation-secret - параметры подключения к СУБД;
- ems-private-rf-integration-secret - специфические настройки модуля
- ems-private-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);

Configmaps:

- ems-private-rf-integration - специфические настройки модуля
- metrics - настройки метрик;

В случае запуска приложения без k8s кластера необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST - URL keycloak
- KC_CLIENT_SECRET - secret key для private
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125");
- HASH_SALT - crc-code, добавляемый при вычислении контрольной суммы сообщения
- TICKETING_API_BASE - базовый URL сервиса, предоставляющего информацию по билетам

Модуль Протокол (PRT)

Ветка: protocol

Модуль Protocol - система для управления гостями, персоналом и мероприятиями. Позволяет создавать и редактировать мероприятия, назначать на мероприятие гостей и персонал, отправлять приглашения и сообщения. Гостей, персонал и мероприятия можно создавать самостоятельно, либо использовать синхронизацию с модулями DRS, ACM, A&D.

Развертывается отдельно как на приватную, так и на публичную часть.

Пример развертывания публичной части:

```
./gradlew :ems-public:protocol-public:clean  
:ems-public:protocol-public:dockerPushImage-stage-1  
:ems-public:protocol-public:deployToK8s-stage-1 -PclusterName=ems-public
```

Пример развертывания приватной части:

```
./gradlew :ems-private:protocol-private:clean  
:ems-private:protocol-private:dockerPushImage-stage-1  
:ems-private:protocol-private:deployToK8s-stage-1 -PclusterName=ems-private
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- postgres-protocol-secret – настройки подключения к бд;
- database-common – настройки пула соединений с базой в приложении
- ems-private-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- ems-private-keycloak-service-account - указывается client id и client secret сервисной учетной записи keycloak;
- frontend-info – содержит приватные/публичные ссылки модулей
- metrics – настройки метрик
- swagger – настройки для отключения/включения swagger ui
- kafka - параметры подключения к Kafka

Модуль

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- [KC_AUTH_HOST](#) - URL keycloak private части;
- KC_CLIENT_SECRET – secret key для private
- KC_SA_CLIENT_ID_PRIVATE - “client id” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений;

- KC_SA_CLIENT_SECRET_PRIVATE - “client secret” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений
- NOTIFICATION_PUBLIC_URL – публичный url модуля Уведомлений
- NOTIFICATION_PRIVATE_URL – приватный url модуля Уведомлений
- KAFKA_HOST - адрес Kafka брокера для чтения очереди уведомлений, писем;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо;
- KAFKA_PASSWORD - пароль Kafka;
- KAFKA_TOPIC_SUFFIX - суффикс, который будет использоваться для формирования имени топика откуда будут читаться сообщения (для уточнения названий топиков см. Файл `ems-common/notifications/src/resources/application.yaml`, секция `app.kafka-listeners`);
- ISSUE_REPORT_EMAIL - email на который будут отправляться сообщения о проблемах;
- KEYCLOAK_PRIVATE_KAFKA_TOPIC - топик kafka, в который записываются сообщения о событиях в keycloak-private. Приложение слушает этот топик и автоматически создает, обновляет, удаляет пользователей в приложении
- KEYCLOAK_PUBLIC_KAFKA_TOPIC - топик kafka, в который записываются сообщения о событиях в keycloak-public. Приложение слушает этот топик и автоматически создает, обновляет, удаляет пользователей в приложении
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- METRICS_TELEGRAF_ENABLED - “true”, если необходимо включить отсылку метрик
Рекомендуется всегда выставлять true из-за особенностей работы `io.micrometer` во избежание OOM
- PROTOCOL_PUBLIC_URL для формирования ссылки в приглашении гостя.
- PROTOCOL_PRIVATE_URL для формирования ссылки в уведомлении.

Зависимости:

- KUM
- Notification
- Kafka
- Dictionary Service

Модуль Chromeheadless (HDS)

Ветка: chrome-headless

Модуль предоставляет API для генерации pdf файлов. Может быть развернут как в приватной части, так и в публичной. Должен быть доступен только внутри приватной сети, т.к. не имеет механизма аутентификации.

Команда для сборки приватной части:

```
./gradlew -p chromeheadless clean build dockerPushImage-preprod deployToK8s-preprod  
-PclusterName=ems-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:
Configmaps:

- metrics - настройки метрик;

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125"); Рекомендуется всегда выставлять true из-за особенностей работы io.micrometer во избежание OOM

Зависимости: нет

Модуль Transport-public (TRN-public)

Ветка: transport

Модуль предоставляет API для обновления координат у машин из приватной части. Может быть развернут в публичной части.

Команда для сборки приватной части:

```
./gradlew -p ems-public/transport clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-public
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:

Secrets:

- postgres-transport-secret - параметры подключения к СУБД;
- ems-public-transport-secret - специфические настройки модуля;

Configmaps:

- metrics - настройки метрик;
- database-common - общие настройки базы

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125"); Рекомендуется всегда выставлять true из-за особенностей работы io.micrometer во избежание OOM

Зависимости: нет

Модуль Transport-private (TRN-private)

Ветка: transport

Модуль предоставляет API для выдачи ВАПов, планирования водительских смен и обслуживания коллцентра. Может быть развернут в приватной части.

Команда для сборки приватной части:

```
./gradlew -p ems-private/transport clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:

Secrets:

- postgres-transport-secret - параметры подключения к СУБД;
- postgres-transport-dwh-secret - параметры подключения к dwh СУБД;
- ems-private-transport-secret - специфические настройки модуля
- ems-private-keycloak-client - настройки клиентов keycloak (указывается хост, по которому происходит авторизация, а также client secret);
- ems-private-keycloak-service-account - настройка системного клиента, для межмодульного взаимодействия;
- kafka - настройки кафки;

Configmaps:

- metrics - настройки метрик;
- frontend-info - урл приватной части
- database-common - общие настройки СУБД
- transport - настройки модуля

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
 - POSTGRES_ADDR - адрес Postgresql;
 - POSTGRES_PORT - порт Postgresql;
 - POSTGRES_USER - пользователь Postgresql;
 - POSTGRES_PASSWORD - пароль Postgresql;
 - POSTGRES_DATABASE - имя базы данных Postgresql;
 - KC_AUTH_HOST - URL keycloak private части
-
- KC_CLIENT_SECRET – secret key для private
 - KC_SA_CLIENT_ID_PRIVATE - “client id” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для взаимодействия с другими модулями;
 - KC_SA_CLIENT_SECRET_PRIVATE - “client secret” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для взаимодействия с другими модулями;

- KAFKA_HOST - адрес Kafka брокера для чтения очереди нотификаций, писем;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо;
- KAFKA_PASSWORD - пароль Kafka;
- KAFKA_TOPIC_SUFFIX - суффикс, который будет использоваться для формирования имени топика откуда будут читаться сообщения (для уточнения названий топиков см. Файл `ems-common/notifications/src/resources/application.yaml`, секция `app.kafka-listeners`);
- ISSUE_REPORT_EMAIL - email на который будут отсылаться сообщения о проблемах;
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125"); Рекомендуется всегда выставлять true из-за особенностей работы `io.micrometer` во избежание OOM
- SITE_PRIVATE_URL - урл сайта
- `integration.keycloak_management.url` - сервис KUM
- `integration.chromeheadless.url` - сервис chromeheadless

Зависимости:

- `keycloak-user-management`
- `chromeheadless`
- `ems-private-notifications` (через кафку)
- `accreditation-private` (через кафку)
- `dictionaries` (через кафку)

Модуль Workforce (WKF)

Ветка: workforce

Модуль WKF - система для управления трудовыми ресурсами.

Развертывается отдельно как на приватную, так и на публичную часть.

Пример развертывания публичной части:

```
./gradlew -p ems-public/workforce-public clean dockerPushImage-stage-1  
deployToK8s-stage-1
```

Пример развертывания приватной части:

```
./gradlew -p ems-private/workforce-private clean dockerPushImage-stage-1  
deployToK8s-stage-1
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- postgres-workforce-secret – настройки подключения к бд. Должен содержать ключи - host, port, user, password, db;
- database-common – настройки пула соединений с базой в приложении, должен содержать ключи - max-pool-size, min-idle;
- ems-public-keycloak-client - настройки клиентов keycloak. Указывается хост, по которому происходит авторизация(ключ auth-host), а также secret key клиента(ключ client-secret);
- ems-private-keycloak-client - настройки клиентов keycloak. Указывается хост, по которому происходит авторизация(ключ auth-host), а также secret key клиента(ключ client-secret);
- ems-private-keycloak-service-account - указывается client id (ключ client-id и client secret (ключ client-secret) сервисной учетной записи keycloak;
- ems-public-keycloak-service-account - указывается client id (ключ client-id) и client secret (ключ client-secret) сервисной учетной записи keycloak;
- frontend-info – содержит приватные/публичные ссылки модулей. Для модуля WKF необходимы быть добавлены ключи workforce-public-url, notification-public-url, notification-private-url
- workforce – специфичные настройки модуля Workforce. Необходимы следующие ключи - issue-email (email адрес техподдержки, на который должны отправляться созданные пользователями заявки о проблемах; change-request-email (email адрес, на который должны отправляться созданные пользователями запросы на изменение данных в их анкетах); swagger-enabled (включение/отключение swagger-ui интерфейса);
- workforce-accreditation - указывается client id (ключ client-id и client secret (ключ client-secret) сервисной учетной записи keycloak, которая позволяет пользоваться API сервиса аккредитации(необходим для отправки пользователей на аккредитацию);
- metrics – настройки метрик. Должен содержать ключ telegraf-enabled, включающий/выключающий сбор и пересылку метрик приложения в telegraf.
- swagger - настройки для отключения/включения swagger ui

- `ems-public-fileupload` - хост публичного сервиса `fileupload`. Должен содержать ключ `host`
- `ems-private-fileupload` - хост приватного сервиса `fileupload`. Должен содержать ключ `host`
- `kafka` - параметры подключения к Kafka. Должен содержать ключи `bootstrap-servers`, `user`, `password`;
- `ems-public-keycloak` - содержит название `kafka`-топика, из которого читаются события о создании/изменении пользователей в `keycloak-public`. Ключ `kafka-topic`;
- `ems-private-keycloak` - содержит название `kafka`-топика, из которого читаются события о создании/изменении пользователей в `keycloak-private`. Ключ `kafka-topic`;
- `mailgun` - ключ `sender-domain`. Домен от которого отправляются email'ы;

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- `JAVA_OPTS` - опции `jvm`, опционально;
- `KC_CLIENT_SECRET` – `secret key` для `private`
- `KC_SA_CLIENT_ID_PUBLIC` - “`client id`” публичной сервисной учетной записи с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений;
- `KC_SA_CLIENT_SECRET_PUBLIC` - “`client secret`” публичной сервисной учетной записи в `keycloak` с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений
- `KC_SA_CLIENT_ID_PRIVATE` - “`client id`” приватной сервисной учетной записи в `keycloak` с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений;
- `KC_SA_CLIENT_SECRET_PRIVATE` - “`client secret`” приватной сервисной учетной записи в `keycloak` с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений
- `KC_SA_CLIENT_AUTH_HOST_PRIVATE` – `auth host` для приватной сервисной учетной записи в `keycloak`
- `KC_SA_CLIENT_AUTH_HOST_PUBLIC` - `auth host` для публичной сервисной учетной записи в `keycloak`
- `NOTIFICATION_PUBLIC_URL` – публичный url модуля Уведомлений
- `NOTIFICATION_PRIVATE_URL` – приватный url модуля Уведомлений
- `SITE_PUBLIC_URL` - публичный url модуля `workforce-public`
- `KAFKA_HOST` - адрес Kafka брокера для чтения очереди уведомлений, писем;
- `KAFKA_USERNAME` - имя пользователя Kafka (используется `SASL_PLAINTEXT` аутентификация), если необходимо;
- `KAFKA_PASSWORD` - пароль Kafka;
- `KAFKA_TOPIC_SUFFIX` - суффикс, который будет использоваться для формирования имени топика откуда будут читаться сообщения (для уточнения названий топиков см. Файл

ems-common/notifications/src/resources/application.yaml, секция app.kafka-listeners);

- SITE_PUBLIC_URL - url public части приложения. Подставляется в redirect_uri параметр keycloak при регистрации пользователя в public части
- FILE_UPLOAD_HOST - url для api fileupload'a
- ISSUE_REPORT_EMAIL - email, на который будут отсылаться сообщения о проблемах;
- USER_CHANGE_REQUEST - email, на который будут отправляться заявки пользователей на изменение данных в их анкетах;
- KEYCLOAK_PRIVATE_KAFKA_TOPIC - топик kafka, в который записываются сообщения о событиях в keycloak-private. Приложение слушает этот топик и автоматический создает, обновляет, удаляет пользователей в приложении
- KEYCLOAK_PUBLIC_KAFKA_TOPIC - топик kafka, в который записываются сообщения о событиях в keycloak-public. Приложение слушает этот топик и автоматический создает, обновляет, удаляет пользователей в приложении
- EMAIL_SENDER_DOMAIN - домен от которого отправляются email'ы
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125"); Рекомендуется всегда выставлять true из-за особенностей работы io.micrometer во избежание OOM
- SWAGGER_ENABLED - включение/выключение swagger-ui;
- KC_SA_CLIENT_ID_ACR - client-id учетной записи в keycloak, необходимый для использования API сервиса accreditation-private;
- KC_SA_CLIENT_SECRET_ACR - client-secret учетной записи в keycloak, необходимый для использования API сервиса accreditation-private;

Зависимости:

- keycloak-user-management
- ems-public-notifications
- ems-private-notifications
- ems-public-fileupload
- ems-private-fileupload
- accreditation-private
- kafka

Модуль Overlay and Facility Management System (OFMS)

Ветка: overlay-and-facility

Основная цель системы: упорядочить информацию в одном пространстве (системе) о временной и постоянной инфраструктуре, которая будет использоваться для проведения мероприятий.

Модуль разворачивается только в приватной части. Команда для сборки:

```
./gradlew -p oms-private/overlay-and-facility-private clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=oms-private
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- postgres-oms-secret – настройки подключения к бд. Должен содержать ключи - host, port, user, password, db;
- database-common – настройки пула соединений с базой в приложении, должен содержать ключи - max-pool-size, min-idle;
- oms-private-keycloak-client - настройки клиентов keycloak. Указывается хост, по которому происходит авторизация(ключ auth-host), а также secret key клиента(ключ client-secret);
- oms-private-keycloak-service-account - указывается client id (ключ client-id и client secret (ключ client-secret) сервисной учетной записи keycloak;
- frontend-info – содержит приватные/публичные ссылки модулей. Для модуля OFMS необходимы быть добавлены ключи oms-private-url;
- metrics – настройки метрик. Должен содержать ключ telegraf-enabled, включающий/выключающий сбор и пересылку метрик приложения в telegraf;
- swagger - настройки для отключения/включения swagger ui;
- oms-public-fileupload - хост публичного сервиса fileupload. Должен содержать ключ host;
- oms-private-fileupload - хост приватного сервиса fileupload. Должен содержать ключ host;
- kafka - параметры подключения к Kafka. Должен содержать ключи bootstrap-servers, user, password;
- oms – специфичные настройки модуля OFMS. Необходим следующий ключ - issue-email (email адрес техподдержки, на который должны отправляться созданные пользователями заявки о проблемах).

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;

- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST - URL keycloak private части;
- KC_CLIENT_SECRET – secret key для private
- KC_SA_CLIENT_ID - “client id” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений;
- KC_SA_CLIENT_SECRET - “client secret” приватной сервисной учетной записи в keycloak с помощью которой происходит;
- KAFKA_HOST - адрес Kafka брокера для чтения очереди уведомлений, писем;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо;
- KAFKA_PASSWORD - пароль Kafka;
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- METRICS_TELEGRAF_ENABLED - “true”, если необходимо включить отсылку метрик приложения (адрес задан явно: “telegraf.kube-system.svc.cluster.local:8125”); Рекомендуется всегда выставлять true из-за особенностей работы io.micrometer во избежание OOM
- ISSUE_REPORT_EMAIL - email на который будут отсылаться сообщения о проблемах;
- SWAGGER_ENABLED - для отключения/включения swagger ui;

Зависимости:

- keycloak-user-management
- kafka
- ems-private-fileupload
- ems-private-notifications

Модуль Equipment (EQP)

Ветка: equipment

Модуль EQP- система для учета и выдачи экипировки аккредитованным лицам.

Модуль разворачивается только в приватной части. Команда для развертывания:

```
./gradlew -p ems-private/equipment-private clean dockerPushImage-stage-1  
deployToK8s-stage-1
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- postgres-equipment-secret - параметры подключения к СУБД;
- postgres-equipment-dwh-secret - параметры подключения к dwh СУБД;
- database-common – настройки пула соединений с базой в приложении, должен содержать ключи - max-pool-size, min-idle;
- ems-private-keycloak-client - настройки клиентов keycloak. Указывается хост, по которому происходит авторизация(ключ auth-host), а также secret key клиента(ключ client-secret);
- ems-private-keycloak-service-account - указывается client id (ключ client-id и client secret (ключ client-secret) сервисной учетной записи keycloak;
- frontend-info – содержит ссылку на модуль. Ключ equipment-private-url;
- metrics – настройки метрик. Должен содержать ключ telegraf-enabled, включающий/выключающий сбор и пересылку метрик приложения в telegraf;
- kafka - параметры подключения к Kafka. Должен содержать ключи bootstrap-servers, user, password;
- equipment - ключ issue-email (email адрес техподдержки, на который должны отправляться созданные пользователями заявки о проблемах).

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- KC_AUTH_HOST - URL keycloak private части;
- [KC_CLIENT_SECRET](#) – secret key для private
- KC_SA_CLIENT_ID_PRIVATE - “client id” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для взаимодействия с другими модулями;
- KC_SA_CLIENT_SECRET_PRIVATE - “client secret” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для взаимодействия с другими модулями;

- KAFKA_HOST - адрес Kafka брокера для чтения очереди нотификаций, писем;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо;
- KAFKA_PASSWORD - пароль Kafka;
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- METRICS_TELEGRAF_ENABLED - "true", если необходимо включить отсылку метрик приложения (адрес задан явно: "telegraf.kube-system.svc.cluster.local:8125"); Рекомендуется всегда выставлять true из-за особенностей работы io.micrometer во избежание OOM
- ISSUE_REPORT_EMAIL - email на который будут отсылаться сообщения о проблемах;
- integration.chromeheadless.url - сервис chromeheadless
- integration.notification.url - приватный url модуля Нотификации

Зависимости:

- keycloak-user-management
- chromeheadless
- ems-private-notifications (через кафку)
- accreditation-private (через кафку)
- dictionaries (через кафку)

Модуль Medical Information System (MIS)

Ветка: medical-encounter

Модуль MIS - система для отображения справочной информации по медицинским учреждениям.

Развертывается отдельно на публичную часть, имеет публичный доступ.

Пример развертывания:

```
./gradlew -p ems-public/medical-info-public clean build dockerPushImage-stage-1  
deployToK8s-stage-1
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- postgres-mis-secret – настройки подключения к бд. Должен содержать ключи - host, port, user, password, db;

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;

Зависимости:

- модуль Dictionary

Модуль Medical Encounters (MES)

Ветка: `medical-encounter`

Модуль MES - система для управления медицинскими процессами. Содержит блоки Аптека, Регистратура, Статистика. Также система содержит функционал по загрузке данных в БД модуля MIS (потому что модуль MIS расположен в публичной части). Разворачивается на приватную часть

Команда для сборки приватной части:

```
./gradlew -p ems-private/medical-encounters clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:

Secrets:

- `postgres-medicine-secret` - параметры подключения к СУБД;
- `ems-private-keycloak-client` – настройки приватного клиентов keycloak (указывается хост, по которому происходит авторизация, а также `client secret`);
- `ems-private-keycloak-service-account` - настройка приватного системного клиента, для межмодульного взаимодействия;
- `ems-public-keycloak-client` – настройки публичных клиентов keycloak (указывается хост, по которому происходит авторизация, а также `client secret`);
- `ems-public-keycloak-service-account` – настройка публичного системного клиента, для межмодульного взаимодействия;
- `kafka` - настройки кафки;

Configmaps:

- `metrics` - настройки метрик. Должен содержать ключ `telegraf-enabled`, включающий/выключающий сбор и пересылку метрик приложения в `telegraf`.
- `frontend-info` - содержит приватные/публичные ссылки модулей. Для модуля необходимы быть добавлены ключи `notification-private-url`
- `database-common` – настройки пула соединений с базой в приложении, должен содержать ключи - `max-pool-size`, `min-idle`;
- `ems-private-accreditation` - настройки модуля аккредитации;
- `medicine` – параметры для модуля MES

В случае развертывания без k8s кластера приложению необходимо

- 1) Установить шрифты MS Core Fonts
- 2) Задать следующие переменные окружения:
 - `JAVA_OPTS` - опции `jvm`, опционально;
 - `POSTGRES_ADDR` - адрес `Postgresql`
 - `POSTGRES_PORT` - порт `Postgresql`
 - `POSTGRES_USER` - пользователь `Postgresql`

- POSTGRES_PASSWORD - пароль Postgresql
- POSTGRES_DATABASE - имя базы данных Postgresql
- KC_AUTH_HOST - URL keycloak private части
- KC_CLIENT_SECRET – secret key для private
- KC_SA_CLIENT_ID_PRIVATE - “client id” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для взаимодействия с другими модулями
- KC_SA_CLIENT_SECRET_PRIVATE - “client secret” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для взаимодействия с другими модулями
- KC_AUTH_HOST_PUBLIC - URL keycloak public части;
- KC_SA_CLIENT_ID_PUBLIC - “client id” публичной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для взаимодействия с другими модулями
- KC_SA_CLIENT_SECRET_PUBLIC - “client secret” публичной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для взаимодействия с другими модулями
- KAFKA_HOST - адрес Kafka брокера для чтения очереди нотификаций, писем;
- KAFKA_USERNAME - имя пользователя Kafka (используется SASL_PLAINTEXT аутентификация), если необходимо
- KAFKA_PASSWORD - пароль Kafka
- KAFKA_TOPIC_SUFFIX - суффикс, который будет использоваться для формирования имени топика откуда будут читаться сообщения (для уточнения названий топиков см. Файл `ems-common/notifications/src/resources/application.yaml`, секция `app.kafka-listeners`)
- ISSUE_REPORT_EMAIL - email на который будут отправляться сообщения о проблемах
- ISSUE_REPORT_EMAIL_SUBJECT – тема письма для сообщения о проблемах
- NOTIFICATION_EMAILS – список почтовых ящиков, на которые будут отправляться медицинские отчеты
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр
- METRICS_TELEGRAF_ENABLED - “true”, если необходимо включить отсылку метрик приложения (адрес задан явно: “`telegraf.kube-system.svc.cluster.local:8125`”) Рекомендуется всегда выставлять true из-за особенностей работы `io.micrometer` во избежание OOM
- EXTERNAL_POSTGRES_ADDR - адрес Postgresql
- EXTERNAL_POSTGRES_PORT - порт Postgresql
- EXTERNAL_POSTGRES_USER - пользователь Postgresql
- EXTERNAL_POSTGRES_PASSWORD - пароль Postgresql
- EXTERNAL_POSTGRES_DATABASE - имя базы данных Postgresql
- EXTERNAL_FILEUPLOAD_URL – URL сервиса Fileupload на кластере, где размещается MIS

Зависимости:

- keycloak-user-management
- ems-private-notifications
- ems-public-fileupload
- ems-private-fileupload
- accreditation-private
- Kafka

Модуль CHATBOT

Ветка: chatbot

Модуль CHATBOT отвечает за работу GameTaxi, DRS, IMS и WKF ботов. GameTaxi бот разработан для заказа такси VIP-клиентами. DRS бот оказывает помощь в эффективном планировании и отслеживании деятельности функциональных подразделений на спортивных и других объектах. IMS бот фиксирует инциденты, которые могут произойти в ходе подготовки или проведения Мероприятий. С помощью WKF бота волонтеры всегда смогут быть в курсе всех последних событий, смогут узнать о новых встречах и сменах, а также подтвердить или отказаться от предложенной должности.

Развертывается отдельно как на приватную, так и на публичную часть.

Пример развертывания приватной части:

```
./gradlew -p ems-private/chatbot-private -PSWAGGER_ENV=stage-1  
dockerPushImage-stage-1 deployToK8s-stage-1
```

Конфигурация приложения описана в следующих k8s configmap/secrets:

- postgres-Frisbee-secret – настройки подключения к бд. Должен содержать ключи -host, port, user, password, db;
- database-common – настройки пула соединений с базой в приложении, должен содержать ключи - max-pool-size, min-idle;
- ems-public-keycloak-client - настройки клиентов keycloak. Указывается хост, по которому происходит авторизация(ключ auth-host), а также secret key клиента(ключ client-secret);
- ems-private-keycloak-client - настройки клиентов keycloak. Указывается хост, по которому происходит авторизация(ключ auth-host), а также secret key клиента(ключ client-secret);
- ems-private-keycloak-service-account - указывается client id (ключ client-id и client secret (ключ client-secret) сервисной учетной записи keycloak;
- ems-private-chatbot – специфичные настройки модуля private-chatbot.
- ems-public-chatbot – специфичные настройки модуля public-chatbot.
- metrics – настройки метрик. Должен содержать ключ telegraf-enabled, включающий/выключающий сбор и пересылку метрик приложения в telegraf.
- swagger - настройки для отключения/включения swagger ui

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- KC_CLIENT_SECRET – secret key для private
- KC_SA_CLIENT_ID_PRIVATE - “client id” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений;
- KC_SA_CLIENT_SECRET_PRIVATE - “client secret” приватной сервисной учетной записи в keycloak с помощью которой происходит аутентификация для отправки сообщений в модуль уведомлений
- KC_SA_CLIENT_AUTH_HOST_PRIVATE – auth host для приватной сервисной учетной записи в keycloak
- ACCREDITATION_URL - приватный url модуля аккредитации
- SERVICE_DESK_URL - url модуля service desk
- SERVICE_DESK_TECHNICIAN_KEY - технический ключ админа для IMSBot
- FRISBEE_URL - url мессенджера chate EG
- FRISBEE_DRS_BOT_REGISTRATION_KEY - регистрационный ключ для DRSBot
- FRISBEE_IMS_BOT_REGISTRATION_KEY - регистрационный ключ для IMSBot
- FRISBEE_TRN_BOT_REGISTRATION_KEY - регистрационный ключ для GameTaxBot
- FRISBEE_WKF_BOT_REGISTRATION_KEY - регистрационный ключ для WKFBot
- DB_MAX_POOL_SIZE - максимальное количество соединений в пуле базы данных, опциональный параметр
- DB_MIN_IDLE - минимальное количество соединений в пуле базы данных, опциональный параметр;
- METRICS_TELEGRAF_ENABLED - “true”, если необходимо включить отсылку метрик приложения (адрес задан явно: “telegraf.kube-system.svc.cluster.local:8125”); Рекомендуется всегда выставлять true из-за особенностей работы io.micrometer во избежание OOM
- SWAGGER_ENABLED - включение/выключение swagger-ui;

Модуль TSRD (прием сообщений от сторонних систем)

Ветка: odf

Модуль предоставляет API на который TSRD операторы (по типу microplus) присылают результаты в реальном времени. Модуль обрабатывает эти результаты и подготавливает для последующего отображения на сайте.

Команда для сборки приватной части:

```
./gradlew -p odf/odf-engine clean build dockerPushImage-preprod deployToK8s-preprod  
-PclusterName=ems-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:

Secrets:

- db - параметры подключения к СУБД;
- security - конфигурация авторизации к сервису;
- site-keycloak-service-account - конфигурация авторизации к админке сайта
- mailgun - конфигурация почтового клиента

Configmaps:

- fileupload - configmap с настройками сервиса, указывается имя бакета, S3 endpoint и другие настройки.
- record-report - куда слать сообщения о новых рекордах если такие будут получены

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- GCS_SERVICE_ACCOUNT_JSON - json-токен сервисной учетной записи для доступа к Google Cloud Storage (GCS), в GCS будут помещаться присланные репорты;
- GCS_BUCKET - имя бакета в GCS;
- SECURITY_ENABLED - необходима ли авторизация для подключения к модулю;
- SECURITY_USER - username в случае если авторизация нужна;
- SECURITY_PASSWORD - пароль в случае если авторизация нужна;
- KC_SA_CLIENT_ID - "client id" сервисной учетной записи кейклока сайта;
- KC_SA_CLIENT_SECRET - "client secret" сервисной учетной записи кейклока сайта;
- KC_AUTH_HOST - URL keycloak сайта;
- MOBILE_SITE_CLIENT_API_URL - урл на сайте которые необходимо уведомить при появлении новых медалей. Необходимо для нотификации на мобильном устройстве о появлении медалей если пользователь подписался на соответствующую страну либо вид спорта
- NOTIFICATION_ENABLED - включена либо нет отправка писем

- MAILGUN_DOMAIN_NAME - доменное имя mailgun выдаваемое при регистрации;
- MAILGUN_API_KEY - ключ mailgun, выдаваемый при регистрации;
- EMAIL_SENDER_DOMAIN - доменное имя, которое модуль уведомлений будет подставлять в поле "from", если приложение его явно не специфицирует;
- RECORD_REPORT_EMAIL_* - адреса на которые необходимо послать email в случае установления нового рекорда по виду спорта. К примеру RECORD_REPORT_EMAIL_ARC задает адрес для рекордов по стрельбе из лука.

Зависимости: нет

Модуль TSRD (клиент для сайта)

Ветка: odf

Модуль предоставляет API используемое сайтом для отображения результатов соревнований.

Команда для сборки приватной части:

```
./gradlew -p odf/odf-web clean build dockerPushImage-preprod deployToK8s-preprod  
-PclusterName=ems-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:
Secrets:

- db - параметры подключения к СУБД;

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;

Зависимости: нет

Модуль Printable-Info (прием сообщений от сторонних систем)

Ветка: odf

Модуль предоставляет API на который TSRD операторы (по типу microplus) присылают результаты в реальном времени. Модуль обрабатывает эти результаты и генерирует PDF отчеты для последующего отображения на сайте.

Команда для сборки приватной части:

```
./gradlew -p odf/odf-engine clean build dockerPushImage-preprod deployToK8s-preprod  
-PclusterName=odf-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:
Secrets:

- db - параметры подключения к СУБД;
- kafka - настройки подключения к kafka;
- odf - настройки модуля, необходимыми при использовании дальнейшей пересылки сообщений в TSRD модуль, такие как имя пользователя в TSRD модуле и пароль пользователя в TSRD модуле;

Configmaps:

- fileupload-odf - configmap с настройками сервиса, указывается имя бакета, S3 endpoint и другие настройки;
- odf - configmap с настройками, необходимыми при использовании дальнейшей пересылки сообщений в TSRD модуль, такими как URL TSRD модуля;

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- GCS_SERVICE_ACCOUNT_JSON - json-токен сервисной учетной записи для доступа к Google Cloud Storage (GCS), в GCS будут помещаться присланные репорты;
- GCS_BUCKET - имя бакета в GCS;
- SECURITY_ENABLED - необходима ли авторизация для подключения к модулю;
- SECURITY_USER - username в случае если авторизация нужна;
- SECURITY_PASSWORD - пароль в случае если авторизация нужна;
- KC_SA_CLIENT_ID - "client id" сервисной учетной записи кейклока сайта;
- KC_SA_CLIENT_SECRET - "client secret" сервисной учетной записи кейклока сайта;
- KC_AUTH_HOST - URL keycloak сайта;
- PRIVATE_MODE_ENABLED - выставить в true
- ODF_PUBLIC_URL - url основного модуля TSRD для приема сообщений, если необходимо проксирование сообщений в него
- KAFKA_HOST - адрес kafka
- KAFKA_USERNAME - имя пользователя в kafka
- KAFKA_PASSWORD - пароль в kafka

Зависимости: kafka

Модуль Printable-Info (API для клиента)

Ветка: odf

Модуль предоставляет API используемое используемое для получения списка сгенерированных по результатам соревнований PDF отчетов.

Команда для сборки приватной части:

```
./gradlew -p odf/odf-web clean build dockerPushImage-preprod deployToK8s-preprod  
-PclusterName=odf-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:
Secrets:

- db - параметры подключения к СУБД;

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- POSTGRES_ADDR - адрес Postgresql;
- POSTGRES_PORT - порт Postgresql;
- POSTGRES_USER - пользователь Postgresql;
- POSTGRES_PASSWORD - пароль Postgresql;
- POSTGRES_DATABASE - имя базы данных Postgresql;
- CONNECTION_PARAMS - значение odf-web

Зависимости: нет

Модули WEB-клиентов(<имя_модуля_клиента>-static)

Ветка: master

Модули предназначены для кэширования и раздачи статических файлов соответствующих клиентских приложений, которые предварительно были загружены в s3-совместимое файловое хранилище.

Шаблон команды для сборки:

```
./gradlew -p nginx-cache/nginx-static-content clean build dockerPushImage-<env>  
deployToK8s-<env> -PclusterName=<cluster> -Papplication=<application-name>  
-Pbucket=<bucket-name> где
```

<env> - среда на которую происходит деплой модуля(dev, stage, prod и тд.)

<cluster> - имя кластера в kubernetes - ems-public или ems-private

<application-name> - имя приложения без указания публичная или приватная часть(accommodation, workforce, accreditation, transport и тд)

<bucket-name> - имя бакета в s3-совместимом файловом хранилище, в которое были загружены файлы web-клиента(workforce-public, workforce-private, accommodation-public, accommodation-private и тд.)

Пример команды для сборки модуля раздачи статических файлов WEB-клиента для публичной части модуля workforce, который был предварительно загружен в бакет workforce-public

```
./gradlew -p nginx-cache/nginx-static-content clean build dockerPushImage-prod  
deployToK8s-prod -PclusterName=ems-public -Papplication=workforce  
-Pbucket=workforce-public
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:

Configmaps:

- nginx-static - хранит настройку content-security-policy

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- S3_HOST - URL для доступа к s3-совместимому файловому хранилищу;
- BUCKET - имя бакета в s3-совместимом файловом хранилище, в которое предварительно были залиты статические файлы WEB-клиента;
- CSP - настройки content-security-policy;

Зависимости: S3

Модуль S3Cache

Ветка: master

Модуль предназначен для кэширования и раздачи файлов загруженных модулем загрузки файлов s3-совместимое файловое хранилище.

Пример развертывания приватной части:

```
./gradlew -p nginx-cache/s3cache-private clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-private
```

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- HOST - URL для доступа к s3-совместимому файловому хранилищу;

Зависимости: S3

Модуль S3Auth

Ветка: master

Модуль предназначен для проверки доступа к скачиванию файлов, отдаваемых модулем s3cache-*

Пример развертывания приватной части:

```
./gradlew -p ems-common/s3auth clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:
Secrets:

- ems-private-keycloak-client - настройка клиента к авторизационному серверу keycloak в приватной части;
- ems-public-keycloak-client - настройка клиента к авторизационному серверу keycloak в публичной части;
- ems-private-keycloak-service-account - настройки сервис-аккаунта в авторизационном сервере keycloak в приватной части;

Configmaps:

- metrics - включение/выключение сбора метрик для сервиса

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- KC_PUBLIC_URL - адрес keycloak в публичной части;
- KC_PRIVATE_URL - адрес keycloak в приватной части;
- KC_SA_CLIENT_ID - client-id сервис-аккаунта в keycloak в приватной части;
- KC_SA_CLIENT_SECRET - client-secret сервис-аккаунта в keycloak в приватной части;
- METRICS_TELEGRAF_ENABLED - включение/выключение сбора метрик

Зависимости: ems-public-keycloak, ems-private-keycloak

Модуль Monitoring

Ветка: master

Модуль предназначен для внутреннего мониторинга потребления сообщений с шины данных kafka и отправки этой метрики в telegraf.

Пример развертывания приватной части:

```
./gradlew -p ems-common/monitoring clean build dockerPushImage-preprod  
deployToK8s-preprod -PclusterName=ems-private
```

Для запуска модуля необходимо сконфигурировать следующие k8s secrets/configmaps:

Secrets:

- kafka - настройка клиента для доступа к kafka;

Configmaps:

- metrics - включение/выключение сбора метрик для сервиса

В случае развертывания без k8s кластера приложению необходимо задать следующие переменные окружения:

- JAVA_OPTS - опции jvm, опционально;
- KAFKA_HOST - адрес kafka;
- KAFKA_USERNAME - имя пользователя в kafka;
- KAFKA_PASSWORD - пароль пользователя в kafka;
- KC_SA_CLIENT_SECRET - client-secret сервис-аккаунта в keycloak в приватной части;
- METRICS_TELEGRAF_ENABLED - включение/выключение сбора метрик

Зависимости: kafka

Сборка клиентской части приложения

Для развертывания клиентской части приложения необходимы:

- 1) Node.js <https://nodejs.org/en/> версия 10.15.x
- 2) npm <https://www.npmjs.com> версия 6.x.x

Модуль Аккредитация (ACR)

Папка: accreditation

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

accreditationApi - урл на API модуля Аккредитация

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля уведомлений

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

- 2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)
acr - модуль Аккредитация (ACR)
seq - модуль Спортивные заявки (SEQ)
wkf - модуль Персонал и волонтеры (WKF)
mis - модуль Медицинская инфо-система (MIS)
mes - модуль Медицинские обращения (MES)
drs - модуль Расписание работы объектов (DRS)
trn - модуль Транспорт (TRN)
ad - модуль Приезды и отъезды (A&D)
ofm - модуль Оснащение объектов (OFMS)
prt - модуль Протокол (PRT)
tic - модуль Билетная программа (TIC)
mds - модуль Логистика (MDS)
eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

2.2. публичная часть модуля:

2.2.1. **projects/public/src/environments/environment.prod.ts**

accreditationApi - урл на API модуля Аккредитация

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -
настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на
файлы, которые загружены при помощи модуля

загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля уведомлений

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные
авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут
передаваться данные авторизации, чтобы сервер нам их отдал

2.2.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)
mes - модуль Медицинские обращения (MES)
drs - модуль Расписание работы объектов (DRS)
trn - модуль Транспорт (TRN)
ad - модуль Приезды и отъезды (A&D)
ofm - модуль Оснащение объектов (OFMS)
prt - модуль Протокол (PRT)
tic - модуль Билетная программа (TIC)
mds - модуль Логистика (MDS)
eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**
 - 3.2. публичная часть модуля: **npm run build:public**
4. После окончания сборки собранное приложение будет находится в папке **dist**

Модуль Билетная программа (TIC)

Папка: ticketing

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

ticketingApi - урл на API модуля Билетная программа

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

2.2. публичная часть модуля:

2.2.1. **projects/public/src/environments/environment.prod.ts**

ticketingApi - урл на API модуля Билетная программа

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля

загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.2.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:

3.1. приватная часть модуля: **npm run build:private**

3.2. публичная часть модуля: **npm run build:public**

4. После окончания сборки собранное приложение будет находиться в папке **dist**

Модуль Логистика (MDS)

Папка: logistics

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

logisticsApi - урл на API модуля Логистика

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* - настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля уведомлений

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

2.2. публичная часть модуля:

2.2.1. **projects/public/src/environments/environment.prod.ts**

logisticsApi - урл на API модуля Логистика
keycloakUrl - урл модуля Keycloak
keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -
настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на
файлы, которые загружены при помощи модуля
загрузки файлов
sharedStorageUrl - урл на шаблоны документов
uploaderUrl - урл на API модуля загрузки файлов
notificationsApi - урл на API модуля уведомлений
feedbackApi - урл на API модуля Службы поддержки
googleCaptchaApiKey - API-ключ Google reCAPTCHA
useCredential - boolean, определяет нужно ли передавать данные
авторизации при запросе файлов в хранилища
dictionariesApi - урл на API модуля Dictionaries
bugsnagApiKey - ключ <https://www.bugsnag.com/>
yandexAPIKey - API-ключ Яндекс.Карт
staticStorageUrls - список урлов, при запросе на которое будут
передаваться данные авторизации, чтобы сервер нам их отдал

2.2.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)
acr - модуль Аккредитация (ACR)
seq - модуль Спортивные заявки (SEQ)
wkf - модуль Персонал и волонтеры (WKF)
mis - модуль Медицинская инфо-система (MIS)
mes - модуль Медицинские обращения (MES)
drs - модуль Расписание работы объектов (DRS)
trn - модуль Транспорт (TRN)
ad - модуль Приезды и отъезды (A&D)
ofm - модуль Оснащение объектов (OFMS)
prt - модуль Протокол (PRT)
tic - модуль Билетная программа (TIC)
mds - модуль Логистика (MDS)
eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**
 - 3.2. публичная часть модуля: **npm run build:public**
4. После окончания сборки собранное приложение будет находиться в папке **dist**

Модуль Медицинская инфо-система (MIS)

Папка: med-info

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. публичная часть модуля:
 - 2.1.1. **projects/public/src/environments/environment.prod.ts**

medInfoApi - урл на API модуля Персонал и волонтеры

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля

загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные

авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут

передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:
 - 3.1. публичная часть модуля: **npm run build:public**
4. После окончания сборки собранное приложение будет находится в папке **dist**

Модуль Медицинские обращения (MES)

Папка: med-encounters

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

medEncountersApi - урл на API модуля Медицинские обращения
keycloakUrl - урл модуля Keycloak
keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* - настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на файлы, которые загружены при помощи модуля загрузки файлов
sharedStorageUrl - урл на шаблоны документов
uploaderUrl - урл на API модуля загрузки файлов
notificationsApi - урл на API модуля уведомлений
feedbackApi - урл на API модуля Службы поддержки
googleCaptchaApiKey - API-ключ Google reCAPTCHA
useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища
dictionariesApi - урл на API модуля Dictionaries
bugsnagApiKey - ключ <https://www.bugsnag.com/>
yandexAPIKey - API-ключ Яндекс.Карт
staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы
acm - модуль Размещение (ACM)
acr - модуль Аккредитация (ACR)
seq - модуль Спортивные заявки (SEQ)
wkf - модуль Персонал и волонтеры (WKF)
mis - модуль Медицинская инфо-система (MIS)
mes - модуль Медицинские обращения (MES)
drs - модуль Расписание работы объектов (DRS)
trn - модуль Транспорт (TRN)
ad - модуль Приезды и отъезды (A&D)
ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)
tic - модуль Билетная программа (TIC)
mds - модуль Логистика (MDS)
eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**
4. После окончания сборки собранное приложение будет находится в папке **dist**

Модуль Оснащение объектов (OFMS)

Папка: ofms

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

ofmsApi - урл на API модуля Оснащение объектов

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные

авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут

передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**

4. После окончания сборки собранное приложение будет находиться в папке **dist**

Модуль Персонал и волонтеры (WKF)

Папка: workforce

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

workforceApi - урл на API модуля Персонал и волонтеры

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля уведомлений

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные

авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут

передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

2.2. публичная часть модуля:

2.2.1. **projects/public/src/environments/environment.prod.ts**

workforceApi - урл на API модуля Персонал и волонтеры
workforceRegistrationApi: - открытый эндпоинт для регистрации
keycloakUrl - урл модуля Keycloak
keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -
настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на
файлы, которые загружены при помощи модуля

загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные
авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут
передаваться данные авторизации, чтобы сервер нам их отдал

2.2.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:

3.1. приватная часть модуля: **npm run build:private**

3.2. публичная часть модуля: **npm run build:public**

4. После окончания сборки собранное приложение будет находиться в папке **dist**

Модуль Приезды и отъезды (A&D)

Папка: arrivals-departures

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

arrivalsDeparturesApi - урл на API модуля Приезды и отъезды

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

asm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

2.2. публичная часть модуля:

2.2.1. **projects/public/src/environments/environment.prod.ts**

arrivalsDeparturesApi - урл на API модуля Приезды и отъезды

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля

загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля уведомлений

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.2.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:

3.1. приватная часть модуля: **npm run build:private**

3.2. публичная часть модуля: **npm run build:public**

4. После окончания сборки собранное приложение будет находиться в папке **dist**

Модуль Протокол (PRT)

Папка: protocol

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

protocolApi - урл на API модуля Протокол

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля уведомлений

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

2.2. публичная часть модуля:

2.2.1. **projects/public/src/environments/environment.prod.ts**

protocolApi - урл на API модуля Протокол

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля

загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.2.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**
 - 3.2. публичная часть модуля: **npm run build:public**
4. После окончания сборки собранное приложение будет находиться в папке **dist**

Модуль Размещение (АСМ)

Папка: accommodation

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

accommodationApi - урл на API модуля Размещение

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные

авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут

передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

asm - модуль Размещение (АСМ)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

2.2. публичная часть модуля:

2.2.1. **projects/public/src/environments/environment.prod.ts**

accommodationApi - урл на API модуля Размещение
keycloakUrl - урл модуля Keycloak
keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -
настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на
файлы, которые загружены при помощи модуля
загрузки файлов
sharedStorageUrl - урл на шаблоны документов
uploaderUrl - урл на API модуля загрузки файлов
notificationsApi - урл на API модуля нотификаций
feedbackApi - урл на API модуля Службы поддержки
googleCaptchaApiKey - API-ключ Google reCAPTCHA
useCredential - boolean, определяет нужно ли передавать данные
авторизации при запросе файлов в хранилища
dictionariesApi - урл на API модуля Dictionaries
bugsnagApiKey - ключ <https://www.bugsnag.com/>
yandexAPIKey - API-ключ Яндекс.Карт
staticStorageUrls - список урлов, при запросе на которое будут
передаваться данные авторизации, чтобы сервер нам их отдал

2.2.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы
acm - модуль Размещение (ACM)
acr - модуль Аккредитация (ACR)
seq - модуль Спортивные заявки (SEQ)
wkf - модуль Персонал и волонтеры (WKF)
mis - модуль Медицинская инфо-система (MIS)
mes - модуль Медицинские обращения (MES)
drs - модуль Расписание работы объектов (DRS)
trn - модуль Транспорт (TRN)
ad - модуль Приезды и отъезды (A&D)
ofm - модуль Оснащение объектов (OFMS)
prt - модуль Протокол (PRT)
tic - модуль Билетная программа (TIC)
mds - модуль Логистика (MDS)
eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**
 - 3.2. публичная часть модуля: **npm run build:public**
4. После окончания сборки собранное приложение будет находиться в папке **dist**

Модуль Расписание работы объектов (DRS)

Папка: drs

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

drsApi - урл на API модуля Расписание работы объектов

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные

авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут

передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**

4. После окончания сборки собранное приложение будет находится в папке **dist**
Модуль Спортивные заявки (SEQ)

Папка: sport-entries

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

sportEntriesApi - урл на API модуля Спортивные заявки

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля уведомлений

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

2.2. публичная часть модуля:

2.2.1. **projects/public/src/environments/environment.prod.ts**

sportEntriesApi - урл на API модуля Спортивные заявки
keycloakUrl - урл модуля Keycloak
keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -
настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на
файлы, которые загружены при помощи модуля
загрузки файлов
sharedStorageUrl - урл на шаблоны документов
uploaderUrl - урл на API модуля загрузки файлов
notificationsApi - урл на API модуля уведомлений
feedbackApi - урл на API модуля Службы поддержки
googleCaptchaApiKey - API-ключ Google reCAPTCHA
useCredential - boolean, определяет нужно ли передавать данные
авторизации при запросе файлов в хранилища
dictionariesApi - урл на API модуля Dictionaries
bugsnagApiKey - ключ <https://www.bugsnag.com/>
yandexAPIKey - API-ключ Яндекс.Карт
staticStorageUrls - список урлов, при запросе на которое будут
передаваться данные авторизации, чтобы сервер нам их отдал

2.2.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы
acm - модуль Размещение (ACM)
acr - модуль Аккредитация (ACR)
seq - модуль Спортивные заявки (SEQ)
wkf - модуль Персонал и волонтеры (WKF)
mis - модуль Медицинская инфо-система (MIS)
mes - модуль Медицинские обращения (MES)
drs - модуль Расписание работы объектов (DRS)
trn - модуль Транспорт (TRN)
ad - модуль Приезды и отъезды (A&D)
ofm - модуль Оснащение объектов (OFMS)
prt - модуль Протокол (PRT)
tic - модуль Билетная программа (TIC)
mds - модуль Логистика (MDS)
eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**
 - 3.2. публичная часть модуля: **npm run build:public**
4. После окончания сборки собранное приложение будет находиться в папке **dist**

Модуль Транспорт (TRN)

Папка: transport

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

transportApi - урл на API модуля Транспорт

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные

авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут

передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

- 2.2. публичная часть модуля:

2.2.1. **projects/public/src/environments/environment.prod.ts**

transportApi - урл на API модуля Транспорт
keycloakUrl - урл модуля Keycloak
keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -
настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на
файлы, которые загружены при помощи модуля
загрузки файлов
sharedStorageUrl - урл на шаблоны документов
uploaderUrl - урл на API модуля загрузки файлов
notificationsApi - урл на API модуля нотификаций
feedbackApi - урл на API модуля Службы поддержки
googleCaptchaApiKey - API-ключ Google reCAPTCHA
useCredential - boolean, определяет нужно ли передавать данные
авторизации при запросе файлов в хранилища
dictionariesApi - урл на API модуля Dictionaries
bugsnagApiKey - ключ <https://www.bugsnag.com/>
yandexAPIKey - API-ключ Яндекс.Карт
staticStorageUrls - список урлов, при запросе на которое будут
передаваться данные авторизации, чтобы сервер нам их отдал

2.2.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы
acm - модуль Размещение (ACM)
acr - модуль Аккредитация (ACR)
seq - модуль Спортивные заявки (SEQ)
wkf - модуль Персонал и волонтеры (WKF)
mis - модуль Медицинская инфо-система (MIS)
mes - модуль Медицинские обращения (MES)
drs - модуль Расписание работы объектов (DRS)
trn - модуль Транспорт (TRN)
ad - модуль Приезды и отъезды (A&D)
ofm - модуль Оснащение объектов (OFMS)
prt - модуль Протокол (PRT)
tic - модуль Билетная программа (TIC)
mds - модуль Логистика (MDS)
eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**
 - 3.2. публичная часть модуля: **npm run build:public**
4. После окончания сборки собранное приложение будет находится в папке **dist**

Модуль Экипировка (EQP)

Папка: equipment

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

equipmentApi - урл на API модуля Экипировка

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля уведомлений

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)
eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**
4. После окончания сборки собранное приложение будет находится в папке **dist**

Модуль Admin

Папка: admin

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

adminApi - урл на API модуля Admin (User Management)

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на

файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные

авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут

передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:

- 3.1. приватная часть модуля: **npm run build:private**
4. После окончания сборки собранное приложение будет находится в папке **dist**

Модуль Dashboard

Папка: dashboard

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**
2. Изменить необходимые конфигурационные файлы
 - 2.1. приватная часть модуля:
 - 2.1.1. **projects/private/src/environments/environment.prod.ts**

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля нотификаций

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

2.2. публичная часть модуля:

2.2.1. **projects/public/src/environments/environment.prod.ts**

keycloakUrl - урл модуля Keycloak
keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -
настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на
файлы, которые загружены при помощи модуля
загрузки файлов
sharedStorageUrl - урл на шаблоны документов
uploaderUrl - урл на API модуля загрузки файлов
notificationsApi - урл на API модуля нотификаций
feedbackApi - урл на API модуля Службы поддержки
googleCaptchaApiKey - API-ключ Google reCAPTCHA
useCredential - boolean, определяет нужно ли передавать данные
авторизации при запросе файлов в хранилища
dictionariesApi - урл на API модуля Dictionaries
bugsnagApiKey - ключ <https://www.bugsnag.com/>
yandexAPIKey - API-ключ Яндекс.Карт
staticStorageUrls - список урлов, при запросе на которое будут
передаваться данные авторизации, чтобы сервер нам их отдал

2.2.2. **src/modules-links/modules-links.public.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы
asm - модуль Размещение (ACM)
acr - модуль Аккредитация (ACR)
seq - модуль Спортивные заявки (SEQ)
wkf - модуль Персонал и волонтеры (WKF)
mis - модуль Медицинская инфо-система (MIS)
mes - модуль Медицинские обращения (MES)
drs - модуль Расписание работы объектов (DRS)
trn - модуль Транспорт (TRN)
ad - модуль Приезды и отъезды (A&D)
ofm - модуль Оснащение объектов (OFMS)
prt - модуль Протокол (PRT)
tic - модуль Билетная программа (TIC)
mds - модуль Логистика (MDS)
eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:
 - 3.1. приватная часть модуля: **npm run build:private**
 - 3.2. публичная часть модуля: **npm run build:public**
4. После окончания сборки собранное приложение будет находится в папке **dist**

Модуль Dictionaries

Папка: dictionaries

1. В папке с исходным кодом необходимого модуля нужно установить все зависимости используя команду: **npm i**

2. Изменить необходимые конфигурационные файлы

2.1. приватная часть модуля:

2.1.1. **projects/private/src/environments/environment.prod.ts**

keycloakUrl - урл модуля Keycloak

keycloakRealm - настройки realm в Keycloak ("private") *keycloakClient* -

настройки клиента в Keycloak ("frontend") *fileStorageUrl* - урл на файлы, которые загружены при помощи модуля загрузки файлов

sharedStorageUrl - урл на шаблоны документов

uploaderUrl - урл на API модуля загрузки файлов

notificationsApi - урл на API модуля уведомлений

feedbackApi - урл на API модуля Службы поддержки

googleCaptchaApiKey - API-ключ Google reCAPTCHA

useCredential - boolean, определяет нужно ли передавать данные авторизации при запросе файлов в хранилища

dictionariesApi - урл на API модуля Dictionaries

bugsnagApiKey - ключ <https://www.bugsnag.com/>

yandexAPIKey - API-ключ Яндекс.Карт

staticStorageUrls - список урлов, при запросе на которое будут передаваться данные авторизации, чтобы сервер нам их отдал

2.1.2. **src/modules-links/modules-links.private.prod.ts**

modulesLinks - объект с URL на все приватные модули платформы

acm - модуль Размещение (ACM)

acr - модуль Аккредитация (ACR)

seq - модуль Спортивные заявки (SEQ)

wkf - модуль Персонал и волонтеры (WKF)

mis - модуль Медицинская инфо-система (MIS)

mes - модуль Медицинские обращения (MES)

drs - модуль Расписание работы объектов (DRS)

trn - модуль Транспорт (TRN)

ad - модуль Приезды и отъезды (A&D)

ofm - модуль Оснащение объектов (OFMS)

prt - модуль Протокол (PRT)

tic - модуль Билетная программа (TIC)

mds - модуль Логистика (MDS)

eqp - модуль Экипировка (EQP)

dashboardLink - dashboard URL

3. Для сборки модуля используются следующие команды:

3.1. приватная часть модуля: **npm run build:private**

4. После окончания сборки собранное приложение будет находиться в папке **dist**

Загрузка клиентской части на GCS/S3

Собранный клиент можно загрузить на Google Cloud Storage (GCS) или S3-совместимое хранилище для последующей раздачи.

Пример загрузки на S3:

```
aws s3 cp dist/public s3://accommodation-public/ --recursive --endpoint-url  
http://s3.preprod.LOOM:8088 --acl public-read --cache-control 'public,max-age=60'
```

Пример загрузки на GCS:

```
gsutil -m -h "Cache-Control:public,max-age=60" cp -z html,js,dart,json,css,xml,ts,svg -a  
public-read -r "dist/public/*" gs://accommodation-emsprivate-preprod/
```

Для раздачи клиентской части можно использовать кеширующий прокси на базе NGiNX (пример ниже предполагает использование S3-совместимого хранилища). Пример сборки:

```
./gradlew clean build -x test -x check dockerPushImage-$ENV_NAME  
templateK8sManifests-$ENV_NAME -PclusterName=$CLUSTER_NAME  
-Papplication=$APPLICATION -Pbucket=$BUCKET
```

Где

- \$BUCKET - имя бакета на который необходимо проксировать трафик;
- \$ENV_NAME - имя окружения;
- \$APPLICATION - имя приложения (см. Конфигурацию NGiNX).

Также контейнеру необходимо задать следующие переменные окружения:

- S3_HOST - хост S3 API;
- BUCKET - имя бакета на который необходимо проксировать трафик;
- CSP - правила Content Security Policy (см. configmap "nginx-static.content-security-policy").

Требования к облачной платформе для LOOM

Приведены требования рассчитанные из профиля максимальной нагрузки для следующих предполагаемых значений:

- Количество одновременно работающих пользователей: до 28 000
- Из них до 25 000 внешних пользователей
- До 3000 внутренних пользователей

1. Требования к программному обеспечению:

Среда оркестрации контейнеров Kubernetes;
СУБД PostgreSQL;
СУБД Cassandra;
Шина обмена данными Apache Kafka;
ПО Централизованного сбора, хранения и обработки логов;
Система мониторинга;
HTTP(S) Load Balancer-ы.

2. Требования к вычислительным ресурсам:

Виртуальные вычислительные ядра должны быть аналогичными или более производительными, чем ядра процессоров модельного ряда Intel Skylake с частотой равной или более 2.6Ghz.

3. Требования к сети:

Обеспечение канала связи между всеми вычислительными узлами и системой хранения данных внутри кластера со скоростью не менее 10 Гбит/с.

4. Требования к системе хранения данных:

Распределенное хранилище должно обеспечивать не менее 100,000/300,000+ Write/Read IOPS. Хранилище должно быть отказоустойчивым с возможностью масштабирования, а также обеспечивать полную сохранность данных с предельной потерей производительности не более 20% при отказе какого-либо оборудования.

5. Требования к среде контейнерной оркестрации.

Требуется масштабируемая, отказоустойчивая среда выполнения для контейнеризированных приложений на базе Docker. Среда должна обеспечивать контроль работоспособности контейнеризированных приложений, а также обеспечивать восстановление работоспособности любого сервиса в случае отказа какого-либо оборудования.

6. Требования к системам управления базами данных (PostgreSQL, Cassandra) и шине данных Apache Kafka.

Должна быть обеспечена бесперебойная работа систем хранения и обработки данных без потери производительности при отказе какого-либо оборудования, а также автоматического восстановления данных при всевозможных сбоях в работе. Для

реляционных СУБД необходимо обеспечить периодическое резервное копирование данных на случай полного отказа операционного кластера.

7. Требования к ПО для централизованного сбора, хранения и обработки логов. Программное обеспечение должно позволять собирать и централизованно хранить логи со всех компонентов системы, а также позволять осуществлять их просмотр и поиск по содержимому с различными критериями. Система должна быть отказоустойчивой и обеспечивать сохранность данных в случаях отказа какого-либо оборудования.

8. Требования к системе мониторинга.

Система должна быть централизованной, масштабируемой и отказоустойчивой, должна позволять следить за всеми компонентами системы (включая разрабатываемое программное обеспечение), а также позволять определять критерии работоспособности и автоматически информировать операторов в случае их нарушения.

9. Объем вычислительных ресурсов и систем хранения (включает в себя ресурсы для 17 подсистем LOOM, 22 СУБД PostgreSQL, СУБД Cassandra, шины обмена данными Kafka, хранилище логов):

540 вычислительных ядер;

1682 Гб (гигабайт) оперативной памяти;

32.4 Тб (терабайт) хранилища для баз данных;

17.5 Тб файлового хранилища;

1 Гбит/сек интернет канал.