



**Информационная система «Система инвестиционного мониторинга и
анализа (СИМА)»
Руководство системного администратора**

**Москва
2025**

Оглавление

ВВЕДЕНИЕ	3
1. НАЗНАЧЕНИЕ И ОБЩИЕ ПРИНЦИПЫ	4
1.1. Назначение	4
1.2. Общие принципы	4
2. СОПРОВОЖДЕНИЕ И АДМИНИСТРИРОВАНИЕ СИМА	5
2.1. Инсталляция.....	5
2.2. Конфигурирование файлов для запуска контейнеров	8
2.2.1. Композиционные файлы для запуска контейнеров СИМА	9
2.2.2. Конфигурирование стартовой страницы	16
2.2.3. Конфигурирование структуры библиотеки и справочной информации	17
2.3. Запуск системы СИМА.....	19
2.4. Остановка СИМА	20
2.5. Администрирование баз данных	20
2.6. Мониторинг и сбор логов	20
2.7. Резервирование и архивирование СИМА	20
2.8. Изменение структуры	21
2.8.1. Добавление модулей	21
2.8.2. - Изменение стартовой страницы :	21
3. СТРУКТУРА API	23

ВВЕДЕНИЕ

Полное наименование продукта: Информационная система «Система инвестиционного мониторинга и анализа»

Краткое наименование: СИМА, Система.

1. НАЗНАЧЕНИЕ И ОБЩИЕ ПРИНЦИПЫ

1.1. Назначение

Документ описывает и определяет порядок действия системного администратора по установке и сопровождению информационной системы «Система инвестиционного мониторинга и анализа» (СИМА).

1.2. Общие принципы

Минимальные требования к ИТ инфраструктуре для СИМА – 1 CPU, 8 Gb RAM, 50 Gb HDD

СИМА работает в актуальной версии ОС Debian/Ubuntu/Astra linux.

Модули системы запускаются в docker контейнерах.

Для запуска и управления сразу несколькими контейнерами рекомендуется использовать оснастку docker-compose

Набор сервисов (контейнеров) в docker-compose может быть произвольным, но рекомендуется использовать отдельный docker-compose для запуска обратного прокси сервера TRAEFIK.

Также рекомендуется использовать отдельный docker-compose для запуска контейнера с базой данных MySQL/PostgreSQL для модуля авторизации (AUTH_RTT)

2. СОПРОВОЖДЕНИЕ И АДМИНИСТРИРОВАНИЕ СИМА

2.1. Инсталляция

Подготовка сервера

```
apt-get update
```

Установка пакетов с docker

```
apt -y install docker-ce docker-ce-cli containerd.io  
systemctl start docker  
systemctl enable docker
```

Установка пакета с docker-compose

```
curl -L https://github.com/docker/compose/releases/download/1.25.3/docker-  
compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

Состав файлов в дистрибутиве СИМА:

distrib_app.tar.gz - единый архив, в котором собраны все необходимые для запуска СИМЫ файлы.

Данные для загрузки в базу данных (дампы ДБ) подготовлены для первичной настройки системы инвестиционного мониторинга и анализа (СИМА), используется база данных - MySQL. Данный архив собран командой tar, затем сжат командой gzip (из стандартного набора)

app_install.sh - скрипт для первичной установки, который копирует архив в хомовую папку пользователя, разарчивает там структуру, загружает образы и запускает приложение (более подробно в описании процесса ниже)

READ.me - этот файл с описанием установки

astra_docker/ - папка с пакетами deb для оффлайн установки docker в Астра Линукс

Описание процесса установки:

Для установки платформы с приложением СИМА нужно подключить папку с дистрибутивом к целевому серверу (это может быть флэш/накопитель/NFS/SMB и тд) или просто скопировать ее целиком в доступное место на сам сервер.

Для запуска установочного скрипта нужны права root, в скрипте не используется sudo, поэтому, чтобы было проще, предлагается запускать скрипт из-под пользователя root.

Запуск скрипта:

```
#~ ./app_install.sh <аргумент_1> <аргумент_2> где,
```

<аргумент_1> - имя пользователя в операционной системе, в хомовую директорию которого будет делаться установка и из-под которого будет запускаться платформа с СИМА, это обязательный параметр;

<аргумент_2> - имя приложения, это не обязательной, но важный параметр для настройки. Если его не указать, то он будет равен аргументу_1.

Нужно учесть, что при установке данный аргумент используется в качестве имени базы данных приложения, со всеми связанными с этим ограничениями имен.

```
#~ ./app_install.sh sima
```

- проверяет наличие в системе пользователя с именем <аргумент_1>, если такого нет, то создает его и его хомовую директорию в /home/<аргумент_1>

- создает внутри хомовой директории пользователя папку для приложения /home/<аргумент_1>/<аргумент_2>

- разворачивает архив и создает следующую структуру для приложения (при <аргумент 1> = <аргумент 2>= sima):

6

		reporting_init.sql	- файл для создания первичной структуры базы данных приложения СИМА (либо другого) с демо данными
		support_init.sql	- файл для создания первичной структуры базы данных модуля техподдержки
		creating_databases.sh	- скрипт для создания баз данных, пользователей в БД и загрузки демо данных
		postgres	- папка для базы данных (пустая, так как используется другая БД)
		traefik	- корневая папка обратного прокси сервера
		data	
		config	
			traefik_dynamic.toml - файл для динамической настройки обратного прокси сервера
			traefik.toml - файл с конфигурацией обратного прокси сервера
		ssl	- папка для сертификатов ssl для обратного прокси сервера
			rtt.digital.crt - пример сертификата
			rtt.digital.key - пример приватного ключа

- проверяет наличие пакетов для контейнеризации docker, если их нет или их недостаточно, то делается установка docker со всеми необходимыми пакетами.

При установке docker скрипт определяет текущую ОС линукс, и в зависимости от типа (Debian/Ubuntu/Alt/RedOS/Centos/Astra/Suse) добавляет репозиторий и устанавливает docker (через yum, apt и тд). Для установки пакетов docker нужен доступ с сервера в интернет.

Внимание ! Для Астры скрипт не делает установку пакетов docker, поэтому рекомендуем поставить их перед установкой СИМА, для этого в папке astra_docker текущего дистрибутива есть пакеты для ручной установки в астра линукс через команду dpkg.

Указанные пакеты можно поставить и в случае отсутствия доступа в интернет, но только для ОС линукс семейства Debian/Ubuntu/Astra.

- добавляет пользователя <аргумент_1> в группу docker для управления запусками контейнеров

- переименовывает текущие служебные папки:

```

app_attachments      ---> <аргумент2>_attachments
edu_app_attachments  ---> edu_<аргумент2>_attachments
queue/app            ---> queue/<аргумент2>
queue/edu_app        ---> queue/edu_<аргумент2>

```

- по имеющимся шаблонам подготавливает композиционный файл docker-compose.yml и настроечные файлы .env и modules.json

- после этого шага скрипт предложит проверить и заполнить валидными значениями файл .env с переменными окружения для docker-compose.yml, это нужно сделать обязательно -- прописать валидные значения (имена или ip адреса) для переменных WEB_HOST_NAME и AUTH_HOST_NAME, так как приложение будет доступно именно по этим адресам. Это можно сделать из параллельной сессии, или прервать выполнение скрипта, выполнить скрипт по созданию БД creating_database.sh, убедиться, что они создались, затем запустить приложение через команду docker compose up -d в папке /home/<аргумент_1>/<аргумент_2>/app/

- запускает контейнер с mysql и создает базы данных для модуля авторизации, модуля приложения и модуля техподдержки.

При этом в приложении создается первичный логин и пользователь во всех модулях со всеми правами :

```

login/username :      admin
password       :      AdminPassword123!

```

В зависимости от мощности сервера (иногда шаги по созданию БД наступают раньше, чем готова БД), где запускается СУБД, шаг по созданию баз данных

и загрузки в них демо данных можно повторять, до тех пор, пока не получится создать все БД, а можно в параллельной сессии отдельно запустить скрипт

`creating_database.sh` (смотри структуру выше) с аргументом = имени приложения

- в случае успешного предыдущего шага, скрипт запускает все остальные контейнеры, необходимые для работы приложения, их должно быть 13 штук :

1. `auth` - модуль авторизации (обязательный)
2. `<аргумент_2>` - модуль приложения с первичной настройкой и демо данными (обязательный)
3. `<аргумент_2>-worker` - модуль обработчика сценариев приложения (не обязательно)
4. `<аргумент_2>_kreator` - модуль загрузки и обработки реестров инвестиций (обязательный)
5. `<аргумент_2>_redis` - модуль обработчика очередей (не обязательно)
6. `edu_<аргумент_2>` - модуль песочницы для приложения (не обязательно)
7. `edu_<аргумент_2>-worker` - модуль обработчика сценариев песочницы приложения (не обязательно)
8. `edu_<аргумент_2>_redis` - модуль обработчика очередей песочницы (не обязательно)
9. `rtt-static` - модуль для работы с библиотекой и статическими файлами (обязательный)
10. `traefik` - прокси сервер (обязательный)
11. `support` - модуль с тех. поддержки (обратной связи) с первичной настройкой (не обязательно)
12. `mailhog` - это ловушка для писем, рекомендуется использовать при тестировании массовых рассылок (не обязательно)
13. `mysql` - контейнер с базой данных `mysql` (обязательный)

Все контейнеры запускаются через композиционный файл:
`/home/<аргумент_1>/<аргумент_2>/app/docker-compose.yml`

Если какие-либо контейнеры не запущены, то нужно совместно с системным администратором проверить этот файл на наличие ошибок или не соответствий и после корректировки запустить приложение в ручную из-под пользователя `<аргумент_1>` (или от другого имеющего права для работы с `docker`) :

```
#~ cd /home/<аргумент_1>/<аргумент_2>/app
#~ docker compose up -d
```

После успешной инсталляции можно удалить папку `/home/<аргумент_1>/<аргумент_2>/images` , при этом освободится более 700 мб места на диске.

2.2. Конфигурирование файлов для запуска контейнеров

Ниже в примерах файлов используются следующие обозначения:

<аргумент1> - задается для установочного скрипта, означает имя пользователя, в домашнюю директорию которого будет устанавливаться приложение СИМА

<аргумент2> - это имя самого приложения СИМА, в нашем примере
<аргумент1>=<аргумент2>=sima

<WEB_HOST_NAME> - это имя хоста СИМА, т.е. имя, по которому пользователь набирает в строке браузера для подключения к СИМА

<AUTH_HOST_NAME> - имя хоста с сервисом авторизации (модуль AUTH), обычно совпадает с именем хоста СИМА

2.2.1. Композиционные файлы для запуска контейнеров СИМА

В данном разделе приведены примеры файлов docker-compose, необходимые для запуска СИМА.

Данные файлы генерятся установочным скриптом из соответствующих шаблонов.

Системный администратор всегда может внести необходимые изменения в данные конфигурационные файлы, в зависимости от возникших требований.

Основной файл для запуска системы СИМА (который запускает все контейнеры) находится :

/home/<аргумент1>/<аргумент2>/app/docker-compose.yml

Контейнеры прокси сервера и базы данных можно запускать и останавливать отдельно, из своих «локальных» композиционных файлов :

- Пример файла конфигурации прокси сервера Traefik для запуска контейнера (Полную документацию по Traefik v2/3 можно найти по адресу <https://doc.traefik.io/traefik/>) :

/home/<аргумент1>/<аргумент2>/traefik/docker-compose-traefik.yml

```
services:
#####
#   обратный прокси сервер для маршрутизации запросов между модулями
#####
  traefik:
    image: "traefik:2.8"
    container_name: "traefik"
    restart: "unless-stopped"
    ports:
      - "80:80"
      - "8090:8090"
      - "443:443"
    volumes:
      - "/var/run/docker.sock:/var/run/docker.sock:ro"
      - "../traefik/data/acme.json:/acme.json"
      - "../traefik/data/ssl:/ssl"
      - "../traefik/data/config:/etc/traefik/"
      - "../traefik/data/./configuration/"
    labels:
      - "traefik.http.routers.traefik.rule=Host(`${WEB_HOST_NAME}`) &&
(PathPrefix(`/api`) || PathPrefix(`/dashboard`))"
      - "traefik.http.routers.traefik.service=api@internal"
      - "traefik.http.routers.traefik.tls=true"
      - "traefik.http.routers.traefik.middlewares=traefik-ratelimit"
      - "traefik.http.middlewares.traefik-ratelimit.ratelimit.average=100"
```

```

- "traefik.http.middlewares.traefik-ratelimit.ratelimit.burst=50"
- "traefik.http.routers.traefik.middlewares=simpleAuth"
-
"traefik.http.middlewares.simpleAuth.basicauth.users=admin:$apr1$$ybjN1m4n$$hxtHZ
xkNYkJkd1ENUumnU."
- "traefik.http.routers.traefik.middlewares=traefik-compress"
- "traefik.http.middlewares.traefik-compress.compress=true"
networks:
- sima

```

- Пример файла конфигурации traefik:
/home/<аргумент1>/<аргумент2>/traefik/data/traefik.toml

```

[log]
  level = "DEBUG" # для обычной работы рекомендуем заменить на INFO

[accessLog]

[entryPoints]
  [entryPoints.dashboard]
    address = ":8090"

  [entryPoints.web]
    address = ":80"
    [entryPoints.web.http.redirects.entryPoint]
      to = "websecure"
      scheme = "https"
  [entryPoints.websecure]
    address = ":443"
    [[tls.certificates]]
      certFile = "/ssl/rtt.digital.crt"
      keyFile = "/ssl/rtt.digital.key"

[api]
  dashboard = true
  insecure = true

[providers.docker]
  watch = true
  network = "rtt"

[providers.file]
  filename = "traefik_dynamic.toml"
  watch = true
  directory = "/configuration"

```

Пример файла конфигурации traefik_dynamic.toml :

/home/rtt/traefik/data/traefik_dynamic.toml

```

[tls.stores]
  [tls.stores.default]
    [tls.stores.default.defaultCertificate]
      certFile = "/ssl/rtt.digital.crt"
      keyFile = "/ssl/rtt.digital.key"

```

- Пример композиционного файла для запуска приложений СИМА
/home/<аргумент1>/<аргумент2>/app/docker-compose-sima.yml

```

#####
# - environments :
# APP_NETWORK=secured - not send e-mail, but generate url to auth module
# TWO_FACTOR=yes - ON/OFF sent autorisated code to user
# SECRET= - include secure code between modules
# HELP_URL= - url to help resources, tied to "?"
# DASHBOARD=quality/metro - on dashboards for ASUK or for Metrology
# BODY_LIMIT=50Mb - max size of the clients requests
# SHOW_PRODUCTS=0 - on/off products list ( if set this attribute to
1 - on)

```

```

# MAIL_SERVICE=
# MAIL_HOST=
# MAIL_PORT=
# MAIL_USER=
# MAIL_PASSWORD=
# MAIL_FROM=
# ENABLE_CONSIERGE=0/1          - switch to dashboards after logging
# TITLE=                        - name in information line
# WEB_TITLE=                    - name of header in WEB browser and in starting
page
# ENABLE_REPORTING=             - on/off approval process and position point after
connection (Y/N)
# ATTACHMENTS_DIR=              - directory for attached files
# REDIS=redis://redis_metrolog:6379
# DB=                            - database connection string
# BASE_URL=                     - address for redirect in proxy traefik
# AUTH_HOST=                    - authorisation host
# DEFAULT_STATE=                - reports status ( from 1 to 4), default is 1 -
state created
# EXPORT_TEMPLATE_PATH          - path to export in excel reports
# ENABLE_PARENT_EDITORS         - if 0 (default) as ASUK else 1 - as role privileges
#
# - example connections string to databases :
# DB=sqlite:///usr/data/dev_auth.db
# DB=mysql://root:mysql/auth
# DB=mysql://root:mysql/reporting?decimalNumbers=true
# DB=mysql://asukdemo:asukdemo_123456@172.16.1.240/reporting?decimalNumbers=true
# DB=postgresql://rtt:Nhbfy0y%2301@172.16.1.242:5432/reporting
#
services:
#####
###      это модуль авторизации
#####

auth:
  image: rtt-auth:latest
  container_name: auth
  restart: "unless-stopped"
  depends_on:
    - mysql
    - traefik
  environment:
    - BASE_URL=https://${AUTH_HOST_NAME}/auth/
    - DB=mysql://root:mysql/auth?decimalNumbers=true
    - APP_NETWORK=secured # если есть почтовый сервис, то эту строку можно
закомментировать
    - MAIL_SERVICE=${MAIL_SERVICE}
    - MAIL_USER=${MAIL_USER}
    - MAIL_PASSWORD=${MAIL_PASSWORD}
    - MAIL_FROM=${MAIL_FROM}
    - SECRET=${SECRET_KEY}
  volumes:
    - ./data:/usr/data
  labels:
    - "traefik.enable=true"
    - "traefik.http.routers.auth.rule=PathPrefix(`/auth`) || Path(`/`)" # ТАК
ЗАДАЕТСЯ ПРЕФИКС. ДОЛЖЕН СОВПАДАТЬ С ПРЕФИКСОМ В BASE_URL
    - "traefik.http.middlewares.auth-
redirect.redirectRegex.regex=^(https?:\\|\\|/(\\|\\|\\|w:.|+\\|)|\\|w\\|._-
|+)(:\\|d+)?\\|/$$"
    - "traefik.http.middlewares.auth-
redirect.redirectRegex.replacement=${1}/auth/"
    - "traefik.http.routers.auth.middlewares=auth-redirect"
    - "traefik.http.routers.auth.tls=true"
  networks:
    - sima

#####
###      это модуль приложения
#####
sima:
  image: "rtt-reporting:latest"
  container_name: sima
  restart: on-failure
  environment:

```

```

- BASE_URL=https://${WEB_HOST_NAME}/sima/ # ВОТ ТУТ НАДО ВПИСАТЬ ПРАВИЛЬНЫЙ
ДОМЕН
- DB=mysql://root@mysql/${DB_NAME}?decimalNumbers=true # for internal
connecting (local), for other connect string to DB - see above
- ATTACHMENTS_DIR=/usr/data/sima_attachments
- ENABLE_CONSIERGE=0
- PERIODICITY=year
- ENABLE_REPORTING_APPROVAL=0
- ENABLE_PARENT_EDITORS=1
- DEFAULT_STATE=2
- DASHBOARD=investments
- HELP_URL=https://${AUTH_HOST_NAME}/static/documents/faqs/sima_faqs.html
- REDIS=redis://sima_redis:6389
- BODY_LIMIT=10mb
- WEB_TITLE=sima
- TITLE=${TITLE}
- SECRET=${SECRET_KEY}
- PREFILL_PARSING_URL=https://${WEB_HOST_NAME}/kreator/jsonmaker/
volumes:
- ./data:/usr/data
labels:
- "traefik.enable=true"
- "traefik.http.routers.sima.rule=PathPrefix(`/sima`)" # ТАК ЗАДАЕТСЯ
ПРЕФИКС. ДОЛЖЕН СОВПАДАТЬ С ПРЕФИКСОМ В BASE URL
- "traefik.http.routers.sima.tls=true"
#
- "traefik.http.routers.edu.tls.certresolver=lets-encrypt"
- "traefik.http.middlewares.sima-compress.compress=true"
networks:
- sima
#####
# этот блок пока не используется в СИМА, но его можно настроить, если потребуются#
# сценарии или какое-либо workflow #
#####
# sima_redis:
#   container_name: sima_redis
#   image: redis:latest
#   restart: always
#   command: redis-server --port 6389 --save 20 1 --loglevel warning
#   ports:
#     - 6389
#   volumes:
#     - ./data/queue/sima:/data
#   networks:
#     - sima
#
# sima-worker:
#   image: "rtt-reporting:latest"
#   container_name: sima-worker
#   restart: on-failure
#   depends_on:
#     - sima_redis
#     - sima
#   command: ['node', 'worker.js']
#   environment:
#     - BASE_URL=https://${AUTH_HOST_NAME}/sima/ # ВОТ ТУТ НАДО ВПИСАТЬ ПРАВИЛЬНЫЙ
ДОМЕН
#     - DB=mysql://root@mysql/${DB_NAME}?decimalNumbers=true
#     - REDIS=redis://sima_redis:6389
#     - ATTACHMENTS_DIR=/usr/data/sima_attachments
#     - PERIODICITY=year
#     - BODY_LIMIT=10mb
#     - WEB_TITLE=sima
#     - ENABLE_PARENT_EDITORS=1
#     - SECRET=${SECRET_KEY}
## use this cestion only for production mode and after testing mailing processing
###   - MAIL_HOST=${MAIL_HOST}
###   - MAIL_PORT=${MAIL_PORT}
###   - MAIL_SERVICE=${MAIL_SERVICE}
###   - MAIL_USER=${MAIL_USER}
###   - MAIL_PASSWORD=${MAIL_PASSWORD}
###   - MAIL_FROM=${MAIL_FROM}
# Use this section for testing group mail sending or mailing processing, if not -
comment it
- MAIL_HOST=mailhog
- MAIL_PORT=1025

```

```

volumes:
  - ./data:/usr/data
networks:
  - sima

#####
###      это модуль песочницы приложения
#####
edu_sima:
  image: "rtt-reporting:latest"
  container_name: edu_sima
  restart: on-failure
  environment:
    - BASE_URL=https://${WEB_HOST_NAME}/edu_sima/      # БОТ ТУТ НАДО ВПИСАТЬ
ПРАВИЛЬНЫЙ ДОМЕН
    - DB=mysql://root@mysql/edu_${DB_NAME}?decimalNumbers=true # for internal
connecting (local), for other connect string to DB - see above
    - ATTACHMENTS_DIR=/usr/data/edu_sima_attachments
    - ENABLE_CONSIERGE=0
    - PERIODICITY=year
    - ENABLE_REPORTING_APPROVAL=0
    - ENABLE_PARENT_EDITORS=1
    - DEFAULT_STATE=2
    - DASHBOARD=investments
    - HELP_URL=https://${AUTH_HOST_NAME}/static/documents/faqs/sima_faqs.html
    - REDIS=redis://edu_sima_redis:6388
    - BODY_LIMIT=10mb
    - TITLE=EDU ${TITLE}
    - WEB_TITLE=edu_sima
    - SECRET=${SECRET_KEY}
    - PREFILL_PARSING_URL=https://${WEB_HOST_NAME}/kreator/jsonmaker/
  volumes:
    - ./data:/usr/data
  labels:
    - "traefik.enable=true"
    - "traefik.http.routers.edu_sima.rule=PathPrefix(`/edu_sima`)" # ТАК ЗАДАЕТСЯ
ПРЕФИКС. ДОЛЖЕН СОВПАДАТЬ С ПРЕФИКСОМ В BASE_URL
    - "traefik.http.routers.edu_sima.tls=true"
#    - "traefik.http.routers.edu.tls.certresolver=lets-encrypt"
    - "traefik.http.middlewares.edu_sima-compress.compress=true"
  networks:
    - sima

#####
# этот блок пока не используется в СИМА, но его можно настроить, если потребуются#
# сценарии или какое-либо workflow #
#####
# edu_sima_redis:
#   container_name: edu_sima_redis
#   image: redis:latest
#   restart: always
#   command: redis-server --port 6388 --save 20 1 --loglevel warning
#   ports:
#     - 6388
#   volumes:
#     - ./data/queue/edu_sima:/data
#   networks:
#     - sima
#
# edu_sima-worker:
#   image: "rtt-reporting:latest"
#   container_name: edu_sima-worker
#   restart: on-failure
#   depends_on:
#     - edu_sima_redis
#     - edu_sima
#   command: ['node', 'worker.js']
#   environment:
#     - BASE_URL=https://edu_${AUTH_HOST_NAME}/sima/ # БОТ ТУТ НАДО ВПИСАТЬ
ПРАВИЛЬНЫЙ ДОМЕН
#     - DB=mysql://root@mysql/edu_${DB_NAME}?decimalNumbers=true
#     - REDIS=redis://edu_sima_redis:6388
#     - ATTACHMENTS_DIR=/usr/data/edu_sima_attachments
#     - PERIODICITY=year
#     - BODY_LIMIT=10mb
#     - WEB_TITLE=sima

```

```

# - ENABLE_PARENT_EDITORS=1
# - SECRET=${SECRET_KEY}
# use this cestion only for production mode and after testing mailing processing
### - MAIL_HOST=${MAIL_HOST}
### - MAIL_PORT=${MAIL_PORT}
### - MAIL_SERVICE=${MAIL_SERVICE}
### - MAIL_USER=${MAIL_USER}
### - MAIL_PASSWORD=${MAIL_PASSWORD}
### - MAIL_FROM=${MAIL_FROM}
# Use this section for testing group mail sending or mailing processing, if not -
comment it
- MAIL_HOST=mailhog
- MAIL_PORT=1025
volumes:
- ./data:/usr/data
networks:
- sima

#####
### CREATOR
#####
sima_creator:
  image: sima-reestr-creator:latest
  container_name: sima_creator
  ports:
    - 8019:8000
  environment:
    - DB_HOST=${DB_HOST} # Service name for Database or DB host name
    - DB_USER=${DB_USER}
    - DB_PASS=${DB_PASSWORD}
    - DB_NAME_CREATE=${DB_NAME}
    - DB_NAME=${DB_NAME}
    - DB_PORT=${DB_PORT}
    - DBMS=mysql # DB type - (mysql ore sqlite3 only)
    - AUTH_PATH=/mnt/auth.db
  labels:
    - "traefik.enable=true"
    - "traefik.http.routers.sima_creator.rule=PathPrefix(`/sima_creator`)" # TAK
ЗАДАЕТСЯ ПРЕФИКС. ДОЛЖЕН СОВПАДАТЬ С ПРЕФИКСОМ>
    - "traefik.http.routers.sima_creator.tls=true"
    - "traefik.http.middlewares.sima_creator-compress.compress=true"
  volumes:
    - ./data:/mnt/
  networks:
    - sima

#####
### это модуль обратной связи
#####
support:
  container_name: support
  image: "rtt-reporting:latest"
  restart: on-failure
  depends_on:
    - auth
  environment:
    - BASE_URL=https://${AUTH_HOST_NAME}/support/ # БОТ ТУТ НАДО ВПИСАТЬ
ПРАВИЛЬНЫЙ ДОМЕН
    - DB=mysql://root@${DB_HOST}/support?decimalNumbers=true
    - ATTACHMENTS_DIR=/usr/data/support_attachments
    - ENABLE_CONSIERGE=0
    - TITLE=ПОДДЕРЖКА
    - HELP_URL=https://${AUTH_HOST_NAME}/auth/library/#section-0
    - ENABLE_REPORTING=N
    - ENABLE_PARENT_EDITORS=1
    - SECRET=${SECRET_KEY}
  volumes:
    - ./data:/usr/data
  labels:
    - "traefik.enable=true"
    - "traefik.http.routers.support.rule=PathPrefix(`/support`)" # TAK ЗАДАЕТСЯ
ПРЕФИКС. ДОЛЖЕН СОВПАДАТЬ С ПРЕФИКСОМ В BASE_URL
    - "traefik.http.routers.support.tls=true"
    - "traefik.http.middlewares.support-compress.compress=true"

```

```

networks:
  - sima

#####
###   эта ловушка для писем (используется для проверки почтовой рассылки)
#####
#   mailhog:
#     container_name: mailhog
#     image: mailhog/mailhog
#     logging:
#       driver: 'none' # отключить запись логов
#     ports:
#       - 1025 # smtp server
#       - 8025:8025 # web ui
#     networks:
#       - sima
#####
###   библиотечка, файлы и прочая статика
#####
static:
  container_name: static
  image: "rtt-static:latest"
  restart: on-failure
  depends_on:
    - auth
  environment:
    - AUTH_HOST=auth
  volumes:
    - ./data/static:/var/www/static
    - /opt/ssl:/etc/ssl
  labels:
    - "traefik.enable=true"
    - "traefik.http.routers.static.rule=PathPrefix(`/static`)"
    - "traefik.http.middlewares.static-compress.compress=true"
    - "traefik.http.routers.static.tls=true"
  ports:
    - "80"
    - "443"
  networks:
    - sima

#####   СЕТЕВОЕ ПРОСТРАНСТВО   #####
networks:
  sima:
    external: true

```

- Пример композиционного файла для запуска СУБД MySQL:
/home/<аргумент1>/<аргумент2>/mysql/docker-compose-
mysql.yml

```

services:
#####
###   База данных , если запускается в контейнере
#####
mysql:
  container_name: mysql
  image: "mysql:latest"
  restart: on-failure
  deploy:
    replicas: 1
    placement:
      constraints: [node.role == manager]
  environment:
    - MYSQL_ALLOW_EMPTY_PASSWORD=YES
  volumes:
    - ../mysql/data/mysql:/var/lib/mysql
    - ../mysql/data/config/my.cnf:/etc/my.cnf
  ports:
    - '3306:3306'
  networks:
    - sima
  healthcheck:
    test: ["CMD", "mysqladmin" ,"ping", "-h", "localhost"]

```

```

        timeout: 5s
        retries: 10
##### СЕТЕВОЕ ПРОСТРАНСТВО #####
networks:
  sima:
    external: true

```

2.2.2. Конфигурирование стартовой страницы

Пример конфигурации стартовой страницы, т.е. той страницы СИМА, которая открывается сразу после авторизации.

Здесь могут быть добавлены или скорректированы любые секции, которые отрисовывают на стартовой странице ссылки для доступа к в эти приложения, структура секции (это обычный формат json) понятна из приведенного ниже примера.

Редактировать стартовую страницу можно в любое время, при этом перестартовывать приложения не требуется.

Формат секции в файле конфигурации :

```

{
  "name": "короткое описание или название приложения",
  "href": "ссылка/имя сервиса контейнера/путь к файлу",
  "description": "Подробное описание, комментарии, любая разъяснительная информация"
},

```

Каждая такая секция генерит на стартовой странице «плитку», при нажатии на которую курсором «мышки», происходит переход по ссылке, указанной в поле "href".

Стартовая страница конфигурируется через описание в файле :

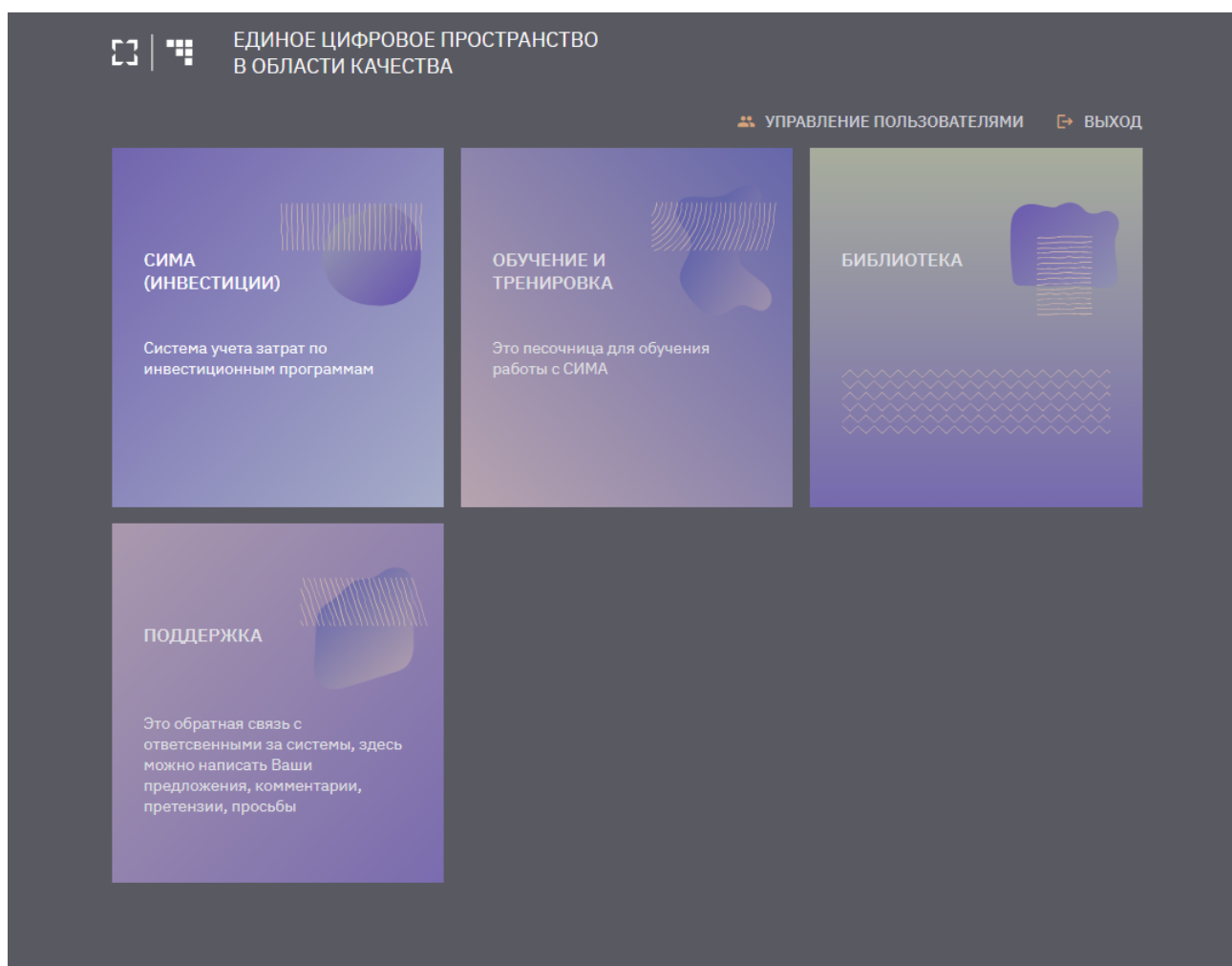
/home/<аргумент1>/<аргумент2>/app/data/static/modules.json

```

[
  {
    "name": "СИМА (Инвестиции)",
    "href": "/sima",
    "description": "Система учета затрат по инвестиционным программам"
  },
  {
    "name": "Обучение и тренировка",
    "href": "/edu_sima/",
    "description": "Это песочница для обучения работы с СИМА"
  },
  {
    "name": "Библиотека",
    "href": "/auth/library/"
  },
  {
    "name": "Поддержка",
    "href": "/support/",
    "description": "Это обратная связь с ответственными за системы, здесь можно написать Ваши предложения, комментарии, претензии, просьбы"
  }
]

```

В соответствии с приведенном выше файлом, стартовая страница должны выглядеть следующим образом :



2.2.3. Конфигурирование структуры библиотеки и справочной информации

В этом файле описана структура реальной библиотеки в формате json, указанные в секциях файлы копируются в соответствующие папки (см. структуру выше) из дистрибутива СИМА, структура библиотеки может быть скорректирована в любое время, при этом рестарта приложений не требуется:

/home/<аргумент1>/<аргумент2>/app/data/static/library.json

```
[
  {
    "section": "Часто задаваемые вопросы. (FAQ)",
    "content": [
      {
        "id": "СИМА",
        "name": "Часто задаваемые вопросы по СИМА",
        "href": "/static/documents/faqs/sima_faqs.html"
      }
    ]
  },
  {
    "section": "Методические материалы",
    "content": [
      {
        "id": "Схема по инвестированию (пример)",
        "name": "Логическая схема процесса согласования инвестиционного реестра (пример)",
        "downloadHref": "/static/documents/standards/Investments_scheme.pdf"
      }
    ]
  }
]
```

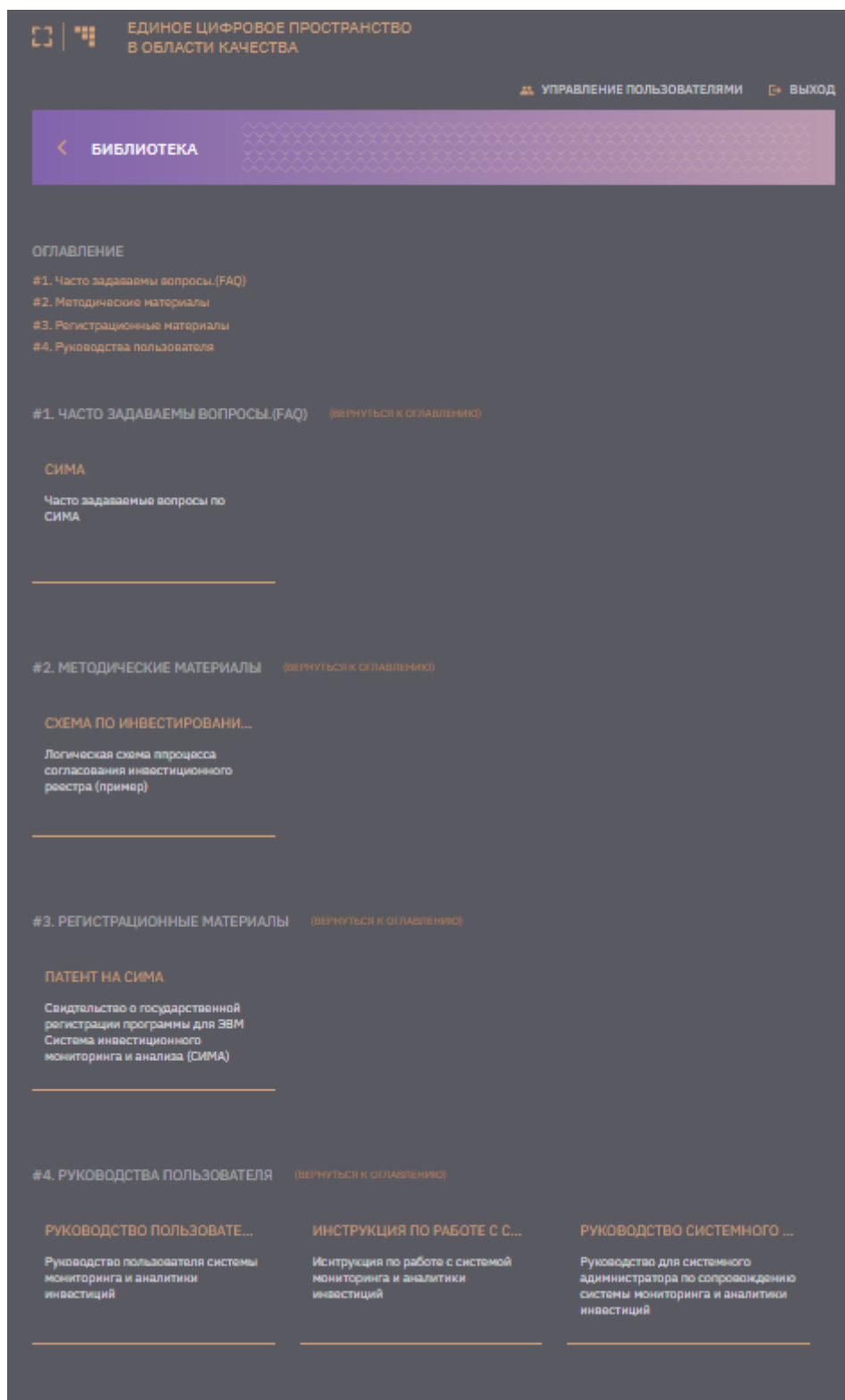
```

    }
    ],
    {
      "section": "Регистрационные материалы",
      "content": [
        {
          "id": "Патент на СИМА",
          "name": "Свидетельство о государственной регистрации программы для ЭВМ Система инвестиционного мониторинга и анализа (СИМА)",
          "downloadHref": "/static/documents/regulations/Sertificate_SIMA.pdf"
        }
      ]
    },
    {
      "section": "Руководства пользователя",
      "content": [
        {
          "id": "Руководство пользователя СИМА",
          "name": "Руководство пользователя системы мониторинга и аналитики инвестиций",
          "downloadHref": "/static/documents/manuals/UserGuide_SIMA.pdf"
        },
        {
          "id": "Инструкция по работе с СИМА",
          "name": "Инструкция по работе с системой мониторинга и аналитики инвестиций",
          "downloadHref": "/static/documents/manuals/Instruction_SIMA.pdf"
        },
        {
          "id": "Руководство системного администратора СИМА",
          "name": "Руководство для системного администратора по сопровождению системы мониторинга и аналитики инвестиций",
          "downloadHref": "/static/documents/manuals/SysAdminGuide_SIMA.pdf"
        }
      ]
    }
  ]
}
]

```

Система делает сборку библиотеки в соответствии с этим конфигурационным файлом, страницы отображаются через веб браузер, при этом автоматически делается оглавление.

Пример страницы с библиотекой по этому файлу приведен ниже :



2.3. Запуск системы СИМА

- Система СИМА может запускаться через единый композиционный файл :

Перейти в папку с приложением :

```
cd /home/<аргумент1>/<аргумент2>/app/
```

запустить приложение

```
docker-compose up -d
```

- Компоненты (модули) СИМА также могут запускаться отдельно :

Обратный прокси сервер

Перейти в директорию откуда будет стартован traefik

```
cd /home/<аргумент1>/<аргумент2>/traefik
```

Запустить контейнер с traefik в фоновом режиме

```
docker-compose -f docker-compose-traefik.yml up -d
```

Запуск СУБД

Перейти в директорию откуда будет стартован СУБД

```
cd /home/<аргумент1>/<аргумент2>/mysql
```

Запустить контейнер с СУБД в фоновом режиме

```
docker-compose -f docker-compose-mysql.yml up -d
```

Запуск модулей СИМА

Перейти в директорию, откуда будут запускаться приложения

```
cd /home/<аргумент1>/<аргумент2>/app
```

Запустить контейнеры с приложениями в фоновом режиме

```
docker-compose -f docker-compose-sima.yml up -d
```

2.4. Остановка СИМА

Остановка системы (может быть в любой последовательности) делается командами управления контейнерами `docker kill`, `docker-compose stop`.

Для рестарта СИМА в случае внесения изменений, можно использовать `docker kill/ docker-compose up` или `docker-compose restart`

Логи работы приложений можно отслеживать через стандартные команды `docker`.

2.5. Администрирование баз данных

В системе используются общеизвестные СУБД: MySQL, PostgreSQL, SQLite, каждая из которых имеет полноценную документацию/

Специальных рекомендаций по использованию вышеуказанных СУБД в системе СИМА нет.

2.6. Мониторинг и сбор логов

СИМА не имеет собственных средств сбора и обработки логов работы приложений.

Мониторинг системы, обработка и анализ логов делается на уровне системного администрирования программно-аппаратного комплекса.

При этом система сбора и обработки логов может быть выбрана любая, не противоречащая требованиям выбора ПО в организациях.

2.7. Резервирование и архивирование СИМА

В СИМА предусмотрены два основных типа хранения данных:

- на файловой системе, где размещены прикрепляемые документы, инструкции, стандарты и тп, т.е. все статические файлы библиотеки, расположенные в /home/rtt/app/data/static/;

- информация хранимая в базах данных.

Резервирование и архивирование данных производится средствами имеющегося ПО или действующих процедур в организации.

2.8. Изменение структуры

2.8.1. Добавление модулей

Добавление нового модуля с новой функциональностью производится через добавление контейнера с образом reporting_rtt.

Если новая функциональность не требует изменение программного кода, то для подключения дополнительного модуля используется существующий образ (пример, новый модуль называется newmodule):

- добавление новой секции с сервисом в файл docker-compose.yml:

```
newmodule:
  container_name: newmodule
  image: "<имя образа>"
  restart: on-failure
  environment:
    - BASE_URL=https://<имя хоста>/newmodule/ # ВОТ ТУТ НАДО ВПИСАТЬ ПРАВИЛЬНЫЙ
ДОМЕН
    - DB=sqlite:///usr/data/<новое имя базы данных>.db
    - ATTACHMENTS_DIR=/usr/data/<новое имя директории>
  volumes:
    - ./data:/usr/data
  labels:
    - "traefik.enable=true"
    - "traefik.http.routers.newmodule.rule=PathPrefix(`/newmodule`)" # ТАК
ЗАДАЕТСЯ ПРЕФИКС. ДОЛЖЕН СОВПАДАТЬ С ПРЕФИКСОМ В BASE_URL
    - "traefik.http.routers.newmodule.tls=true"
#    - "traefik.http.routers.newmodule.tls.certresolver=lets-encrypt"
    - "traefik.http.middlewares.newmodule.compress=true"
  networks:
    - rtt
```

Здесь <имя образа> совпадает со старым – rtt_reporting

- создание новой директории для прикрепленных документов в новом модуле :

```
mkdir /home/<аргумент1>/<аргумент2>/app/<new attachments name>
```

2.8.2. - Изменение стартовой страницы :

Стартовая страница конфигурируется через описание в файле :

```
/home/<аргумент1>/<аргумент2>/app/data/static/modules
.json
```

добавить секцию :

```
{
  "name": "СИМА NEW Module",
  "href": "/newmodule/",
  "description": "СИМА с новой отчетностью"
},
```

В стартовой страницы в качестве параметра секции могут использоваться web ссылки на любые объекты (сайты, файлы, сервисы)

.

3. СТРУКТУРА API

Обмен данными между модулями системы СИМА осуществляется в виде обмена текстовыми сообщениями в формате JSON. В общем виде формат сообщения содержит:

- METHOD – метод доступа;
- URI – адрес запрашиваемого ресурса;
- Заголовки
- Тело запроса
- Тело ответа

Текстовые сообщения (JSON) могут содержать следующие типы данных:

- Строка
- Число
- Объект (объект JSON)
- Массив
- boolean
- null

Строки в JSON должны записываться в двойных кавычках: { "name": "Иванов Иван Иванович" }.

Числа в JSON должны быть целочисленными или с плавающей точкой: { "type": 5 }.

Интеграционные методы взаимодействия перечислены в Таблица 1 – Методы взаимодействия.

Таблица 1 – Методы взаимодействия

METHOD	URI	Описание
GET	/reporting/api/companies ? limit = x ? offset = y ? term = z	Показать список компаний СИМА из x количества с отступом y содержащие строку z
GET	/reporting/api/companies/{id}	Показать информацию по компании в СИМА с id
POST	/reporting/api/companies	Позволяет создать компанию в СИМА
PUT	/reporting/api/companies/{id}	Позволяет обновить запись компании с id
DELETE	/reporting/api/companies/{id}	Позволяет удалить запись компании с id
GET	/reporting/api/division ? limit = x ? offset = y ? term = z	Показать список дивизионов СИМА из x количества с отступом y содержащие строку z
GET	/reporting/api/divisions/{id}	Показать информацию по дивизиону в СИМА с id
POST	/reporting/api/divisions	Позволяет создать дивизион в СИМА
PUT	/reporting/api/divisions/{id}	Позволяет обновить запись дивизиона с id

DELETE	/reporting/api/divisions/{id}	Позволяет удалить запись дивизиона с id
GET	/reporting/api/holdings/{id}	Показать информацию по холдингу в СИМА с id
POST	/reporting/api/holdings	Позволяет создать холдинг в СИМА
PUT	/reporting/api/holdings/{id}	Позволяет обновить запись холдинга с id
DELETE	/reporting/api/holdings/{id}	Позволяет удалить запись холдинга с id
GET	/reporting/api/products/{id}	Показать информацию по изделию в СИМА с id
POST	/reporting/api/products	Позволяет создать изделие в СИМА
PUT	/reporting/api/products/{id}	Позволяет обновить запись изделия с id
DELETE	/reporting/api/products/{id}	Позволяет удалить запись изделия с id
GET	/reporting/api/reports/{id периода отчётности}/companies/{id компании}/forms/{код формы}/entries	Показать записи в отчетности за период {id периода отчётности} в компании {id компании} в форме {код формы}
GET	/reporting/api/reports/{id периода отчётности}/companies/{id компании}/forms/{код формы}/entries/{id entry}	Показать информацию по записи {id entry} в отчетности за период {id периода отчётности} в компании {id компании} в форме {код формы}
POST	/reporting/api/reports/{id периода отчётности}/companies/{id компании}/forms/{код формы}/entries/	Создание записи в отчетности за период {id периода отчётности} в компании {id компании} в форме {код формы}
PUT	/reporting/api/reports/{id периода отчётности}/companies/{id компании}/forms/{код формы}/entries/{id entry}	Обновить информацию по записи {id entry} в отчетности за период {id периода отчётности} в компании {id компании} в форме {код формы}
DELETE	/reporting/api/reports/{id периода отчётности}/companies/{id компании}/forms/{код формы}/entries/{id entry}	Удалить запись {id entry} в отчетности за период {id периода отчётности} в компании {id компании} в форме {код формы}