ИНФОРМАЦИОННАЯ СИСТЕМА «EXON.ИСР»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРЕДОСТАВЛЕННОГО ДЛЯ ПРОВЕДЕНИЯ ЭКСПЕРТНОЙ ПРОВЕРКИ

Содержание

Термины и определения

В настоящем документе используются следующие термины:

Термин	**Определение**
Дамп	Содержимое рабочей памяти одного процесса, ядра или всей операционной системы. Также может включать дополнительную информацию о состоянии программы или системы, например значения регистров процессора и содержимое стека
Доменное имя	Символьное имя, служащее для идентификации областей, которые являются единицами административной автономии в сети Интернет, в составе вышестоящей по иерархии такой области. Каждая из таких областей называется доменом
Система	Информационная система «Exon.ИСР»

В настоящем документе используются следующие сокращения:

Сокращение	**Определение**
ИСР	Иерархическая структура работ
ОС	Операционная система
ПО	Программное обеспечение
DNS	Domain name server — приложение, предназначенное для ответов на DNS-запросы по соответствующему протоколу
HDD	Hard (magnetic) disk drive — запоминающее устройство (устройство хранения информации, накопитель) произвольного доступа, основанное на принципе магнитной записи
IP	Internet Protocol — маршрутизируемый протокол сетевого уровня стека TCP/IP
RAM	Random Access Memory — один из видов памяти компьютера, позволяющий единовременно получить доступ к любой ячейке (всегда за одно и то же время, вне зависимости от расположения) по ее адресу на чтение или запись

1. Общие сведения

1.1. Назначение

Exon.ИСР — единое пространство для планирования работ по проекту всеми участниками строительного Проекта. Диаграмма Ганта иллюстрирует последовательность, взаимосвязи и прогресс выполнения работ, которые строятся на основании временных, объемных и стоимостных показателей. График работ по проекту поддерживается в актуальном состоянии с возможностью отслеживания внесенных изменений и утверждения базового плана.

Система предоставляет единое интегрированное пространство, позволяющее объединить всех участников процесса календарного планирования:

- Заказчик;
- Технический заказчик;
- Генеральный проектировщик;
- Проектировщик;
- Генподрядчик;
- Субподрядчик;

.

1.2. Автоматизируемые функции

Система позволяет автоматизировать выполнение следующих функций в части планирования:

— создание иерархический структуры работ и вех;

— создание связей;

— отображение в виде настраиваемой Диаграммы Ганта;

— управление сроками, продолжительностью работ, связями на диаграмме Ганта;

— управление текущими и базовыми датами работ;

— задание стоимостных и объёмных параметров работ;

— контроль прогресса и просрочек.

В рамках иерархический структуры работ и вех обеспечивается создание следующих видов работ:

— работа;

— суммарная работа;

— веха.

В рамках соединения работ и/или вех должны обеспечиваются следующие виды связей:

— начало – начало;

— начало – окончание;

— окончание – начало;

— окончание – окончание.

Модуль обеспечивает возможность получения аналитики по работе относительно текущих и базовых дат, отклонений/опережений по объемам и стоимости, по прогнозной дате завершения на основании введенных данных по прогрессу.

Модуль обеспечивает возможность получения аналитики по Проекту: количество записей, количественный и стоимостной прогресс по графику: фактический и относительно текущих и базовых дат.

Модуль обеспечивает возможность создания ролей с разными уровнями доступов:

— «Исполнитель»;

— «Администратор Исполнителя»;

— «Редактор»;

— «Ответственный за объем»;

— «Наблюдатель» (по умолчанию).

Администратор и редакторы корневой работы должен иметь возможность назначать ответственных на любую работу, передавая права на управление по вложенным работам, а также самостоятельно редактировать эти работы.

Администратор корневой работы должен иметь возможность формировать договорной и базовый план для созданных работ.

Модуль обеспечивает возможность просмотра истории как общей, так и для конкретных суммарных работ/работ/вех.

2. Установка Системы

2.1. Системные требования

Для установки требуется ОС Linux, kernel 4.15 или выше. Рекомендуется использовать дистрибутив Debian 10 или выше.

Для минимальной установки необходимо 16 ГБ RAM и 50 ГБ HDD. Рекомендуется 32 ГБ RAM, объем диска зависит от количества загружаемых в Систему пользователями файлов.

Для запуска необходимо установить ПО docker и docker-compose Рекомендуется также установка: **curl, git, iotop,less, mlocate, tcpdump, telnet, traceroute, vim, make.**

Для получения пакетов Системы из репозитория понадобится доступ в Интернет.

Данная инструкция описывает установку Системы для целей разработки или тестирования. При установке в продуктивной среде необходимо предпринять меры по защите и предотвращению доступа к базам данных и другой инфраструктуре проекта.

2.2. Начальная конфигурация

Для работы Системы необходим DNS-сервер и доменное имя.

Для примера в данной инструкции будет использоваться домен первого уровня exon. Если необходима работа с другим именем домена, понадобятся дополнительные настройки.

1) В файле .**env** необходимо установить значение переменной HOST "exon" (для работы локально на одном компьютере, либо любое другое значение вашего домена): `HOST=exon` .

Для корректной работы необходимо, чтобы в DNS существовала wildcard запись (*.exon либо отдельные записи для поддоменов: **pdftron**, **traefik**, **rmq**, **kibana**) для доменов нижнего уровня на тот же IP-адрес. Если нет возможности использовать DNS-сервер, можно добавить в файл **/etc/hosts** строку: `127.0.0.1 exon pdftron.exon kibana.exon traefik.exon rmq.exon`

2) В файле **config/application.yml** заменить домен **exon** на актуальный.

3) В файле **config/frontSettings.json** заменить домен **exon** на актуальный в переменных: `"api": "http://exon/api", "pdfTronServer": "http://pdftron.exon", "urlIdentity": "http://exon/auth", "urlReactApp": "http://exon"`.

2.3. Установка окружения

Для работы Системы необходимы базы данных: Mongodb и Postgresql. Также необходимы сервисы:

- Rabbitmq;
- Redis;
- Elasticsearch;
- Kibana;
- Traefik.

Конфигурация данных сервисов описывается в файле **infrastructure.yml** для docker-compose. Запуск всех необходимых сервисов выполняется командой: `sh docker-compose -f infrastructure.yml up -d`

После запуска сервисов необходимо сконфигурировать доступ к ним согласно инструкциям на сайтах разработчиков. Конфигурирование баз данных выходит за рамки данного документа.

2.4. Установка зависимостей

Для корректной работы Системы необходимы сервисы:

- Keycloak (аутентификация и авторизация пользователей);
- pdftron (отображение и редактирование pdf документов через браузеры);
- config (сервер для хранения настроек приложения).

Перед запуском сервисов необходимо создать базу для keycloak и импортировать в нее дамп **keycloak.sql** с начальными настройками keycloak для проекта.

Конфигурация данных сервисов описывается в файле **dependencies.yml** для docker-compose. Запуск всех необходимых сервисов выполняется командой: `sh docker-compose -f dependencies.yml up -d`

Если доменное имя отличается от **exon**, после запуска необходимо прописать настройки клиента аутентификации. Для этого:

1) С помощью любого веб-браузера перейти по адресу http://exon/auth (заменить exon на используемое имя домена). Имя и пароль по умолчанию: admin/admin.
2) Выбрать **realm SpringBoot**, перейти в раздел **Clients**, для клиента ExonReactApp прописать корректные url в параметры **Root URL, Valid Redirect URIs, Admin URL, Web Origins**.
3) Указать пароль для пользователя доступа к API keycloak.
4) Выбрать **realm Master**, перейти в раздел **Users**, найти пользователя **admin**, перейти на вкладку **Credentials** и установить пароль.
5) Перед сохранением убрать флаг **temporary**. Данный пароль вписать в конфигурационный файл **config/user_service.yml** в секцию **keycloak-config.admin-password**.

2.5. Запуск Системы

Конфигурация сервисов описывается в файле **services.yml** для docker-compose. Запуск всех необходимых сервисов выполняется командой: sh docker-compose -f services.yml up -d

Для работоспособности Системы используются основные микросервисы (Таблица 1) и служебные микросервисы (Таблица 2).

Таблица 1. Основные микросервисы Системы

№	Сервис	Путь к сервису	Команда для запуска	Описание
1	core-isr-front-1	/opt/service/target/*.jar	docker exec core-isr-front-1 /bin/sh	Необходим для отображения пользовательского интерфейса Системы
2	core-isr-new-service-1	/opt/service/target/*.jar	docker exec core-isr-new-service-1 /bin/sh	Необходим для реализации бизнес-логики и доступов к данным Системы
3	core-isr-new-history-service-1	/opt/service/target/*.jar	docker exec core-isr-new-history-service-1 /bin/sh	Необходим для реализации отображения

				исторических данных Системы
4	core-isr-analytics-service-1	/opt/service/target/*.jar	docker exec core-isr-analytics-service-1/bin/sh	Необходим для реализации бизнес-логики аналитики Системы
5	core-checkpoints-service-1	/opt/service/target/*.jar	docker exec core-checkpoints-service-1 /bin/sh	Необходим для реализации бизнес-логики контрольных точек Системы

Таблица 2. Служебные микросервисы

№	Сервис	Путь к сервису	Команда для запуска	Описание
1	core_core-front_1	/opt/service/target/*.jar	docker exec core-core-front-1 /bin/sh	Необходим для отображения пользовательского интерфейса Системы
2	core_org-service_1	/opt/service/target/*.jar	docker exec core-org-service-1 /bin/sh	Необходим для получения информации об организациях, участвующих в проекте
3	core_role-service_1	/opt/service/target/*.jar	docker exec core-role-service-1 /bin/sh	Необходим для регулирования доступов участникам проекта к функциям Системы
4	core_integrations-service_1	/opt/service/target/*.jar	docker exec core-integrations-service-1 /bin/sh	Необходим для интеграции с системой, предоставляющей информацию из ЕГРЮЛ
5	core_users-service_1	/opt/service/target/*.jar	docker exec core-users-service-1 /bin/sh	Необходим для получения информации о пользователях
6	core_project-service_1	/opt/service/target/*.jar	docker exec core-project-	Необходим для получения

			service-1 /bin/sh	информации о проектах
7	core_file-service_1	/opt/service/target/*.jar	docker exec core-file-service-1 /bin/sh	Необходим для хранения и получения доступов к файлам
8	core-license-service-1	/opt/service/target/*.jar	docker exec core-license-service-1	Необходим для получения информации о подключенных Модулях Системы
9	core-payment-service-1	/opt/service/target/*.jar	docker exec core-payment-service-1	Необходим для реализации бизнес-логики связи Модуля ИСР и Моудля Актирование
10	core-export-service-1	/opt/service/target/*.jar	docker exec core-export-service-1	Необходим для реализации экспорта документов
11	core-catalog-service-1	/opt/service/target/*.jar	docker exec core-catalog-service-1	Необходим для получения справочника единиц измерений
12	core_keycloak_1	/opt/service/target/*.jar	docker exec core-keycloak-1 /bin/sh	Необходим для осуществления регистрации и авторизации пользователей в Системе
13	core_config_1	/opt/service/target/*.jar	docker exec core-config-1 /bin/sh	Необходим для хранения настроек и параметров Системы
14	traefik:v2.10	/usr/local/bin/traefik	docker exec traefik /bin/sh	Необходим для распределения трафика между сервисами
15	rabbitmq:master	/etc/rabbitmq	docker exec core-rabbit-1 /bin/sh	Брокер очередей
16	redis:6.2	/usr/local/bin/redis-server	docker exec redis /bin/sh	Необходим для хранения-передачи ключ-значения

17	elasticsearch:7.17.10	/usr/share/elasticsearch	docker exec core-elasticsearch-1 /bin/sh	Агрегатор метрик для сервисов
18	postgres:12.13	/var/run/postgresql	docker exec core-postgres-1 /bin/sh	SQL база данных
19	mongo:4.4	/data/db	docker exec core-mongo-1 /bin/sh	NoSQL база данных
20	core-bpmn-service-1	/usr/local/bin/redis-server	docker exec core-bpmn-service-1	Обеспечение бизнес-логики согласование документов по маршрутам