



VisionLabs
MACHINES CAN SEE

VisionLabs LUNA Access

Инструкция по эксплуатации

ООО «ВижнЛабс»

123458, г. Москва, ул. Твардовского д. 8, стр. 1

☎ +7 (499) 399 3361

✉ info@visionlabs.ru

🌐 www.visionlabs.ru

Содержание

Глоссарий	3
1. Введение	4
2. Общие сведения о сервисе.....	Ошибка! Закладка не определена.
2.1. Функции сервиса	Ошибка! Закладка не определена.
2.2. Системные требования.....	Ошибка! Закладка не определена.
3. Процессы, обеспечивающие Жизненный цикл сервиса	Ошибка! Закладка не определена.
3.1. Общее описание Жизненного цикла сервиса	Ошибка! Закладка не определена.
3.2. Процессы в рамках этапов Жизненного цикла сервиса	Ошибка! Закладка не определена.
3.3. Роли участников процессов Жизненного цикла сервиса	Ошибка! Закладка не определена.
4. Создание сервиса.....	Ошибка! Закладка не определена.
5. Штатное функционирование сервиса	Ошибка! Закладка не определена.
6. Сопровождение сервиса	Ошибка! Закладка не определена.
7. Устранение неисправностей работы сервиса	Ошибка! Закладка не определена.
8. Совершенствование сервиса	Ошибка! Закладка не определена.
9. Техническая поддержка	Ошибка! Закладка не определена.
9.1. Техническая поддержка первого уровня ..	Ошибка! Закладка не определена.
9.2. Техническая поддержка второго уровня...	Ошибка! Закладка не определена.
10. Требования к персоналу, обеспечивающему поддержку сервиса	Ошибка! Закладка не определена.

Глоссарий

Термин	Определение
User Interface (UI)	Графический интерфейс пользователя
Программное обеспечение (ПО)	Программа или множество программ, используемых для управления компьютером
Система контроля управления доступом (СКУД)	Совокупность программно-аппаратных технических средств, направленных на контроль входа и выхода в помещение с целью обеспечения безопасности и регулирования посещения определенного объекта. Например, турникеты на входе в банки/офисные здания

1. Введение

Настоящий документ описывает процесс эксплуатации программного обеспечения «VisionLabs Luna Access».

Исключительные права на ПО «VisionLabs Luna Access» принадлежат ООО «VisionLabs» (далее — Компания).

Настоящий документ подлежит размещению на официальном сайте Компании в сети Интернет по адресу: <https://visionlabs.ai/ru/> (далее — официальный сайт).

2. Схема работы LUNA Access

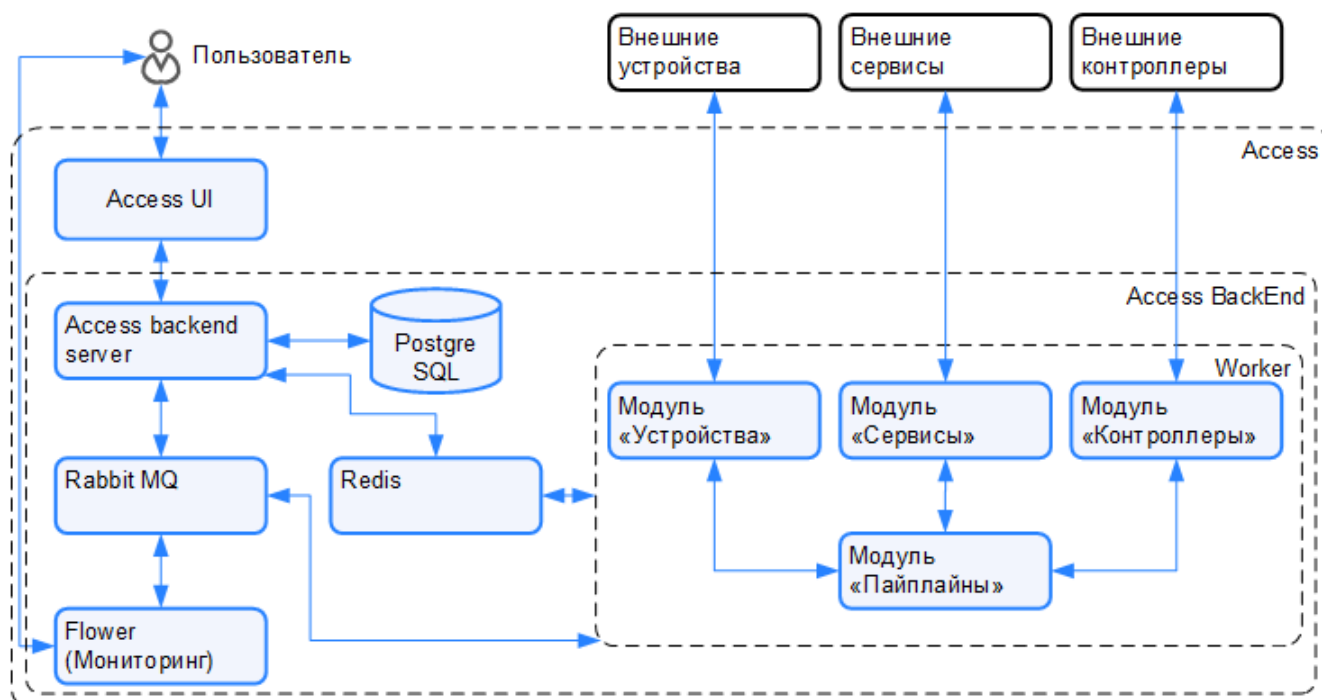


Таблица 1. Описание схемы

Компонент	Описание
Пользователи	Администратору системы, обеспечивающие настройку и поддержание работоспособность Access,
Access	Совокупность программные средства управления, позволяющие реализовать совместную работу продуктов VisionLabs и различных систем контроля и управления доступом (СКУД).
Access UI	Графический интерфейс Access.
Access BackEnd	BackEnd Access, отвечающий за работу с внешними компонентами, взаимодействием с UI и БД.
Access backend server	Набор библиотек, объединяющий модули Access, БД и UI.
Rabbit MQ	Брокер очередей сообщений.
Redis	Система хранения данных в виде структур для обеспечения работоспособности Access.
PostgreSQL	База данных для хранения настроек интеграций и данных пользователей.
Worker	Набор модулей для взаимодействия с внешними компонентами.
Модуль «Пайплайн»	Модуль Access, содержащий библиотеки для настройки работы внешних устройств и внешних сервисов.

Компонент	Описание
Модуль «Устройства»	Модуль Access, содержащий библиотеки для подключения внешних устройств к Access.
Модуль «Сервисы»	Модуль Access, содержащий библиотеки для подключения внешних сервисов к Access.
Модуль «Контроллеры»	Модуль Access, содержащий библиотеки для работы с внешними контроллерами.
Внешние устройства	Подключаемые камеры, терминалы и тепловизоры, передающие видеопоток для дальнейшей обработки. Полный список доступных устройств см. в Руководстве пользователя.
Внешние сервисы	СКУД или продукты VisionLabs, которые могут быть использованы в интеграции. Полный список доступных СКУД и продуктов VisionLabs см. в Руководстве пользователя.
Внешние контроллеры	Внешние контроллеры (например, считыватели карт), используемые в интеграциях.

2.1. Диаграмма последовательности работы

Взаимодействие компонентов Access на примере типовой интеграции источник видеосигнала + LP5 + Sigur (Рисунок 2).

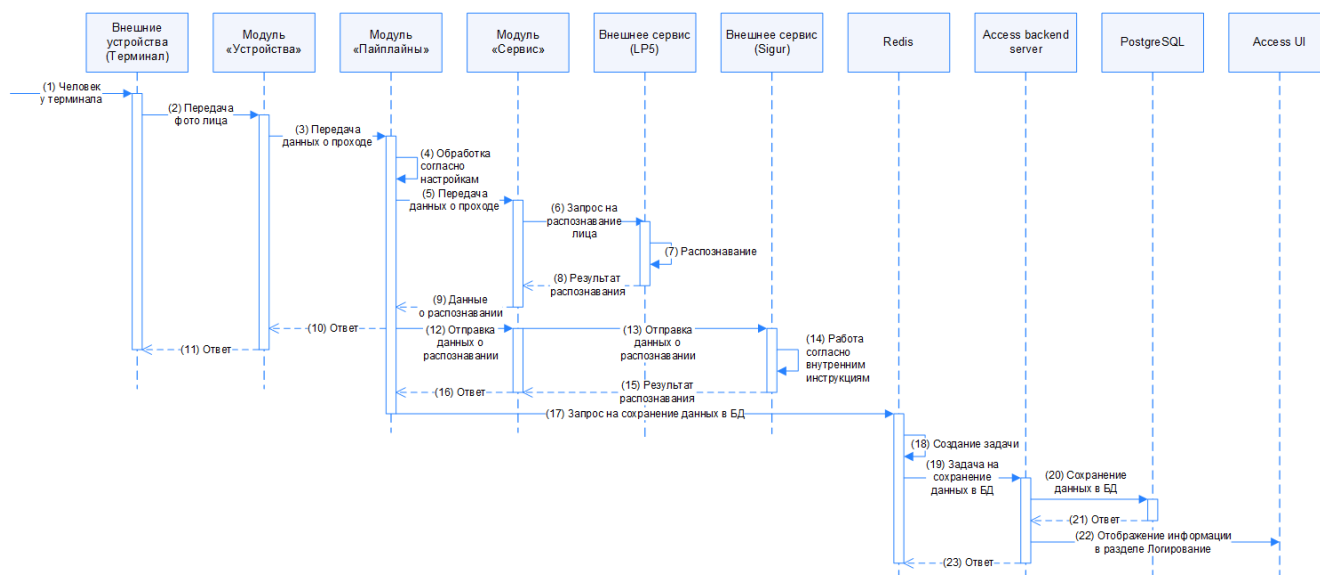


Рисунок 2. Взаимодействие компонентов

Таблица 2. Описание диаграммы последовательности

Шаг	Описание
(1)	Человек подошел к турникету с целью прохода, лицо человека зафиксировал терминал.
(2)	Терминал передал фото человека в модуль «Устройства»

Шаг	Описание
(3)	Модуль устройства передает в модуль Пайплайны данные о лице для дальнейшей обработки
(4)	Модуль Пайплайн обрабатывает фото согласно внутренним настройкам.
(5)	Модуль Пайплайн передает данные о проходе в модуль Сервис для дальнейшей обработки.
(6)	Модуль Сервис отправляет запрос во внешний сервис (LP5) а распознавание лица.
(7)	Внешние сервис (LP5) распознает лицо и производит идентификацию.
(8)	Внешние сервис (LP5) возвращает ответ о результата распознавания и идентификации лица.
(9)	Модуль Сервисы передает в модуль Пайплайн данные о распознавании.
(10)	Модуль Пайплайны возвращает ответ в Модуль устройство о результате распознавания.
(11)	Модуль Устройства возвращает данные о имени человека у терминала для отображения имени и сообщения об успешной идентификации на экране терминала
(12)	Модуль Пайплайны передает в модуль Пайплайн данные о распознавании в модуль Сервисы.
(13)	Модуль Сервисы передает данные о распознавании во Внешние сервис (Sigur)
(14)	Внешний сервис (Sigur) выполняет обработку результата распознавания согласно внутренним инструкциям. В случае успешного распознавания отправляет сигнал на открытие турникета.
(15)	Внешний сервис (Sigur) возвращает результат обработки в Модуль Сервисы.
(16)	Модуль Сервисы передает в модуль Пайплайн данные об обработке.
(17)	Модуль Пайплайны отправляет запрос в Redis на сохранение данных о проходе в БД.
(18)	Redis создает задачу на сохранение информации в БД.
(19)	Redis передает задачу в Access backend server для реализации
(20)	Access backend server отправляет данные в БД для сохранения информации о проходе.
(21)	БД возвращает ответ о результате сохранения.
(22)	Access backend server отправляет в Access UI данные для отображения результата прохода в раздел логирование.
(23)	Access backend server возвращает в Redis информацию о выполнении задачи.

Создание объекта в Access UI (Рисунок 3)

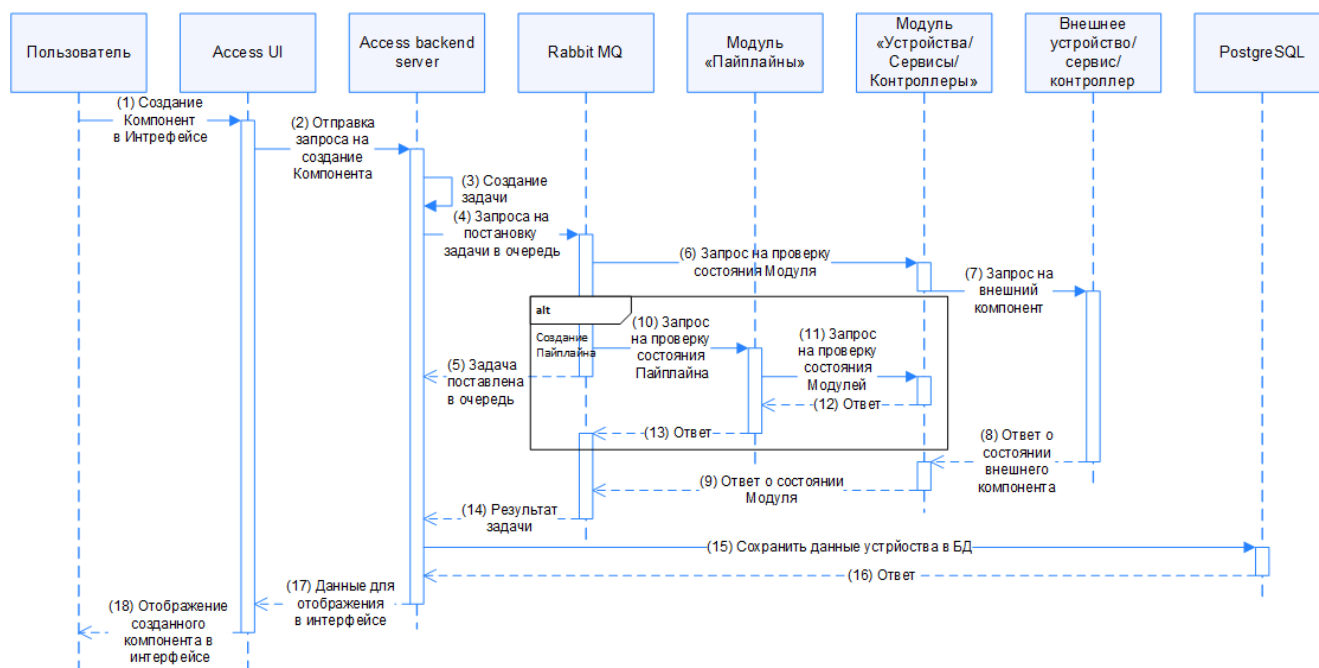


Рисунок 3. Взаимодействие компонентов

Таблица 3. Описание диаграммы последовательности

Шаг	Описание
(1)	Пользователь заполняет данные нового объекта (устройства, сервиса, пайплайна или контроллера) в Access UI.
(2)	Access UI отправляет запрос на Access backend server на создание компонента.
(3)	Access backend server создает задачу на добавление нового компонента.
(4)	Access backend server отправляет задачу в Rabbit MQ для постановки задачи в очередь.
(5)	Rabbit MQ возвращает ответ о постановки задачи в очередь.
(6)	Rabbit MQ отправляет запрос на проверку состояния компонента в соответствующий модуль (Устройства/Сервисы/Контроллеры), указанного в настройках компонента.
(7)	Модуль Устройства/Сервисы/Контроллеры перенаправляет запрос на проверку состояния во внешние устройство/сервис/контроллер.
(8)	Внешние устройство/сервис/контроллер возвращает ответ о состоянии активности.
(9)	Модуль Устройства/Сервисы/Контроллеры возвращает ответ о состоянии внешнего устройство/сервис/контроллер.
(10)	Alt создание пайплайна. Rabbit MQ отправляет запрос в Модуль Пайплайн на проверку состояния компонентов, указанных в настройках пайплайна.
(11)	Alt создание пайплайна. Модуль Пайплайн перенаправляет запрос на проверку состояния в модули Устройства/Сервисы/Контроллеры, которые указаны в настройках пайплайна.

Шаг	Описание
(12)	Alt создание пайплайна. Модули Устройства/Сервисы/Контроллеры возвращают ответ о доступности компонентов.
(13)	Alt создание пайплайна. Модуль Пайплайны возвращает ответ о состоянии.
(14)	Rabbit MQ возвращает результат задачи в Access backend server.
(15)	Access backend server отправляет данные в БД для сохранения информации о проходе.
(16)	БД возвращает ответ о результате сохранения.
(17)	Access backend server отправляет в Access UI данные для отображения результата создания компонента.
(18)	Access UI отображает созданные компонент с активным статусом для пользователя.

Автоматический мониторинг состояния компонентов (Рисунок 4).

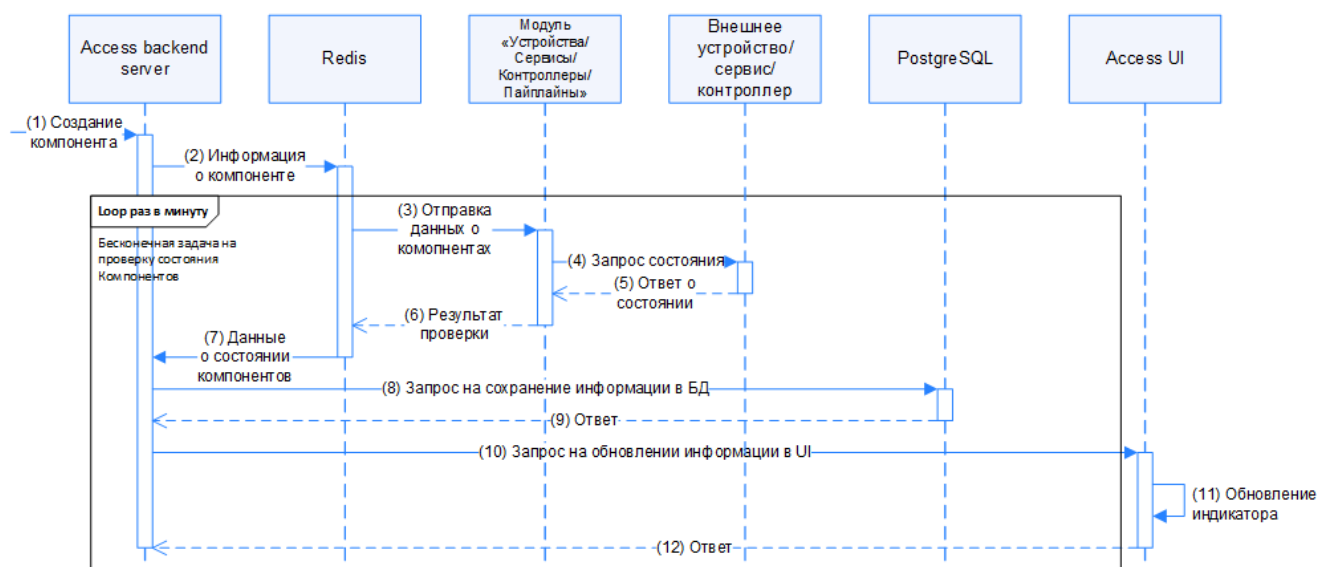


Рисунок 4. Взаимодействие компонентов

Таблица 4. Описание диаграммы последовательности

Шаг	Описание
(1)	Входной точкой для этого процесса является создание любого компонента.
(2)	Access backend server передает данные о созданном компоненте в Redis
(3)	Redis каждую минуту отправляет запрос в подключенные Модули для проверки состояния внешних устройств.
(4)	Модули перенаправляют запрос на проверку подключения к внешним компонентам.
(5)	Внешние компоненты возвращают ответ о состоянии подключения, если ответ на запрос не был получен, то модуль считает подключенное устройство отключенным.
(6)	Модуль возвращает ответ о состоянии компонента.

Шаг	Описание
(7)	Redis передает данные о состоянии компонентов в Access backend server.
(8)	Access backend server отправляет данные в БД для сохранения информации о компонентах.
(9)	БД возвращает ответ о результате сохранения.
(10)	Access backend server отправляет в Access UI данные для отображения статуса активности компонента.
(11)	Access UI обновляет индикатор статуса активности компонента.
(12)	Access UI возвращает ответ в Access backend server об обновлении статуса.

Ручной мониторинг списка задач с помощью Flower (Рисунок 5).

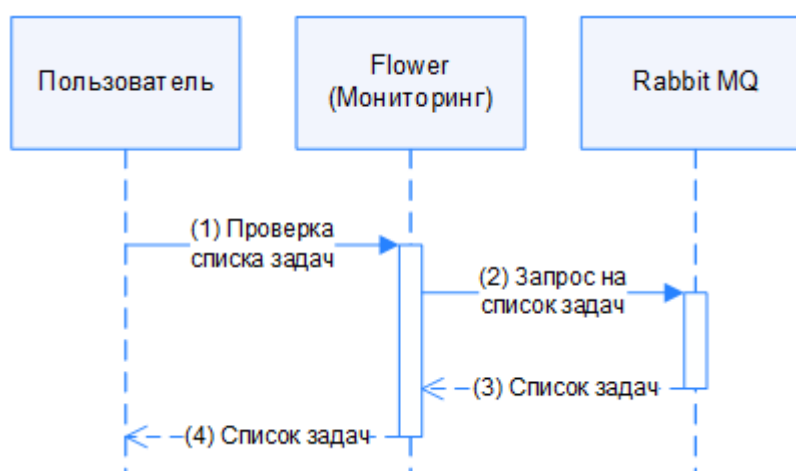


Рисунок 5. Взаимодействие компонентов

Таблица 5. Описание диаграммы последовательности

Шаг	Описание
(1)	Пользователь отправляет запрос в компонент мониторинга Flower на получение списка задач.
(2)	Flower отправляет запрос на получение списка задач в менеджер очередей Rabbit MQ.
(3)	Rabbit MQ возвращает в ответе список задач.
(4)	Flower отображает для пользователя список задач.

3. Коды ошибок

В случае отправки GET-запроса `/api/v1-access/service/active/` на получение данных существующих сервисов или GET-запроса `/api/v1-access/schema/services/{component_name}/` на получение схемы параметров сервиса, или POST-запроса `/api/v1-access/service/` на создание сервиса определенного типа и возникновения ошибок обработки, возвращается ошибка со статус-кодом HTTP 404 или HTTP 422.

В случае отправки GET-запроса `/api/v1-access/service/` на получение существующих типов сервиса и возникновения ошибок обработки, возвращается ошибка со статус-кодом HTTP 404.