

Программный комплекс «Система автоматизированного учета ключей»

Руководство администратора

Содержание

Введение	3
Программный модуль KeyManagement	5
Запуск и остановка программного модуля KeyManagement.....	5
Конфигурация программного модуля KeyManagement.....	6
Отладка программного модуля KeyManagement.....	7
Сброс программного модуля KeyManagement.....	7
Обновление программного модуля KeyManagement.....	8
Удаление программного модуля KeyManagement	8
Программный модуль KeyCore.....	9
Запуск и остановка программного модуля KeyCore.....	9
Конфигурация программного модуля KeyCore.....	9
Отладка программного модуля KeyCore	13
Сброс программного модуля KeyCore	14
Обновление программного модуля KeyCore	14
Удаление программного модуля KeyCore.....	14
Программный модуль KeyUI.....	15
Запуск и остановка программного модуля KeyUI.....	15
Конфигурация программного модуля KeyUI.....	15
Отладка программного модуля KeyUI	15
Сброс программного модуля KeyUI	15
Обновление программного модуля KeyUI	15
Удаление программного модуля KeyUI.....	15

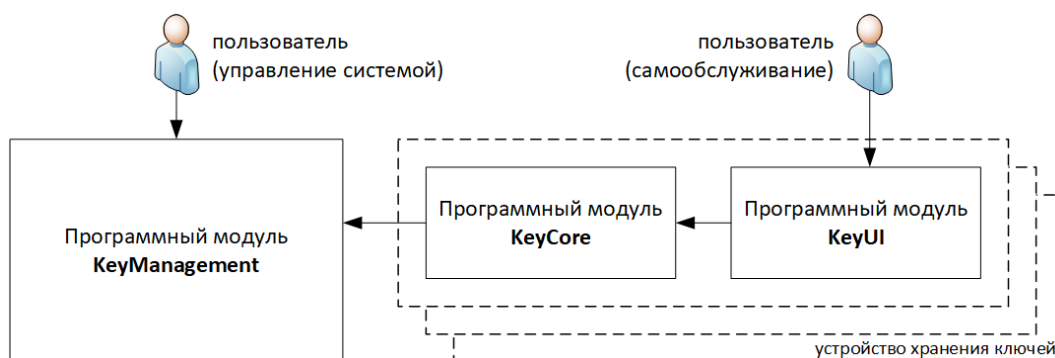
Введение

Программный комплекс «Система автоматизированного учета ключей» (далее – программный комплекс, или ПК) предназначен для организации системы контроля и управления доступом к таким активам, как ключи от помещений, ключи от автомобильной техники и т.п.

Программный комплекс состоит из набора программных модулей:

- программный модуль управления (условное название - **KeyManagement**) – управление обеспечивает выполнение административных задач: мониторинг ключей и устройств хранения, ведение справочников ключей и пользователей, формирование правил доступа, просмотр журнала событий, - а также реализует набор интеграционных возможностей;
- программный модуль ядра (условное название - **KeyCore**) – обеспечивает поддержку специализированных устройств хранения ключей и отвечает за исполнение процедуры обслуживания пользователей на них;
- программный модуль самообслуживания (условное название - **KeyUI**) – предоставляет интерактивный интерфейс взаимодействия пользователей с устройствами хранения ключей.

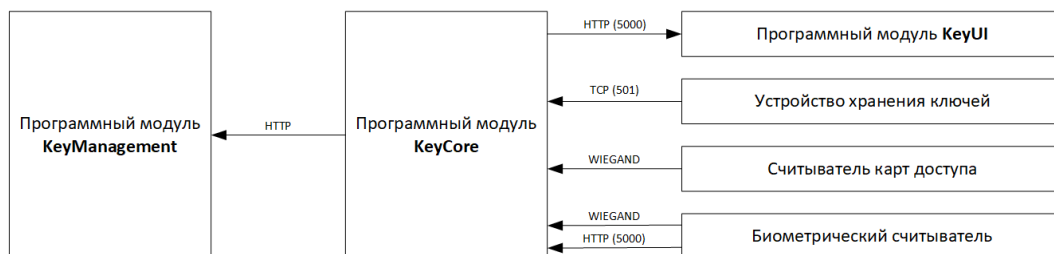
Верхнеуровневая архитектура программного комплекса представлена на рисунке ниже.



Взаимодействие программных модулей друг с другом, а также с поддерживаемыми внешними устройствами происходит на базе стандартных протоколов, представленных на рисунке ниже.

ООО «Платформикс»
ОГРН 1037739996392
ИНН/КПП 7707507077/771301001

127018, г. Москва,
ул. Складочная, д. 3, стр. 1



Настоящий документ охватывает вопросы, связанные выполнением ключевых задач по администрированию программного комплекса.

Программный модуль KeyManagement

Запуск и остановка программного модуля KeyManagement

Для **инициализации** (и одновременного запуска) программного модуля в командной строке, перейдя в рабочую папку с программным модулем KeyMGMT, выполните команду:

docker compose up (Windows) или **sudo docker compose up** (Linux)

Программный модуль KeyMGMT будет готов к обслуживанию пользователей, когда в консоли появится информационное сообщение:

```
keymgmt-key_management-1 | info: Microsoft.Hosting.Lifetime[0]
keymgmt-key_management-1 | Now listening on: http://0.0.0.0:80
keymgmt-key_management-1 | info: Microsoft.Hosting.Lifetime[0]
keymgmt-key_management-1 | Application started. Press Ctrl+C to shut
down.
keymgmt-key_management-1 | info: Microsoft.Hosting.Lifetime[0]
keymgmt-key_management-1 | Hosting environment: Production
keymgmt-key_management-1 | info: Microsoft.Hosting.Lifetime[0]
keymgmt-key_management-1 | Content root path: /key-management
```

После выполненной инициализации программного модуля при перезапуске ОС по умолчанию вместе с запуском среды контейнеризации будет автоматически осуществляться также и запуск программного модуля.

Для **де-инициализации** (и одновременной остановки) программного модуля в командной строке, перейдя в рабочую папку с программным модулем KeyMGMT, выполните команду:

docker compose down (Windows) или **sudo docker compose down** (Linux)

После выполнения де-инициализации программного модуля он может быть снова запущен только путем выполнения повторной инициализации.

Для **ручной остановки, запуска и перезапуска** ранее инициализированного программного модуля в командной строке, перейдя в рабочую папку с программным модулем KeyMGMT, выполните необходимую команду:

docker compose stop (Windows) или **sudo docker compose stop** (Linux)

docker compose start (Windows) или **sudo docker compose start** (Linux)

docker compose restart (Windows) или **sudo docker compose restart** (Linux)

Примечание: в среде Windows вы также можете управлять запуском / остановкой программного модуля из графического интерфейса среды контейнеризации Docker Desktop.

Конфигурация программного модуля KeyManagement

Конфигурация программного модуля находится в файле **appsettings.json**. После изменения файла конфигурации необходимо выполнить перезапуск программного модуля, чтобы изменения вступили в силу.

По умолчанию файл конфигурации имеет следующий вид:

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Debug",
      "Microsoft": "Warning",
      "Microsoft.EntityFrameworkCore.Query": "Error",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "Kestrel": {
    "Endpoints": {
      "Http": {"Url": "http://0.0.0.0:80"},
      // "Https": {"Url": "https://0.0.0.0:443"}
    },
    // "Certificates": {
    //   "Default": {
    //     "Path": "Certs/localhost.pfx",
    //     "Password": "1"
    //   }
    // }
  },
  "ConnectionStrings": {
    "DbConnectionString":
"server=key management db;port=3306;database=KeyManagement;
  user
id=root;password=example;persistsecurityinfo=True"
  },
  "AppSettings": {
    "TimeDifferenceForKeyBoxStatusNoConnection": "00:00:05",
    "UploadingFolderPath": "uploads",
    "KeysNumberParameter": 6,
    "TempFilesFolder": "TempFiles",
    "EnableSwagger": true,
    "SessionTTL": 240
  }
}
```

Описание конфигурационных параметров:

- Kestrel.Endpoints* - определяет, на каких портах сервер будет принимать входящие запросы. Номера портов должны соответствовать открытым портам в файле docker-compose.yml
- Certificates* - параметры сертификата, используемого для безопасного соединения по HTTPS. Сертификат localhost.pfx является демонстрационным и к использованию не рекомендуется
- DbConnectionString - строка подключения к БД. Параметры подключения должны соответствовать параметрам, определенным в файле docker-compose.yml

- TimeDifferenceForKeyBoxStatusNoConnection - длительность, при превышении которой, устройство считается недоступным. Каждый экземпляр модуля KeyCore поддерживает связь с модулем KeyManagement, в случае отсутствия связи дольше заданного значения, устройство (модуль KeyCore) переходит в состояние “Нет связи”, пока связь не возобновится
- UploadingFolderPath - путь к папке с загруженными пользователями файлами, такими как фото пользователей
- KeysNumberParameter - максимальное количество ключей, которым пользователь может назначить категории
- TempFilesFolder - служебная папка для временного хранения файлов при работе программного модуля
- EnableSwagger - (необязательный) - включение/отключение доступа к документации API. Допустимые значения true/false, по умолчанию - false. Если включено, документация API доступна по URL:
<key_management_host>/swagger, например http://localhost/swagger
- SessionTTL - Время жизни пользовательской сессии (минут)

* Для работы только по протоколу HTTP, необходимо удалить или закомментировать из раздела конфигурации **Kestrel** параметры **Endpoints.Https** и **Certificates**.

Отладка программного модуля KeyManagement

Для просмотра текущего состояния контейнеров программного модуля в командной строке, перейдя в рабочую папку с программным модулем, выполните команду:

docker compose ps (Windows) или **sudo docker compose ps** (Linux)

Для просмотра логов контейнеров программного модуля в командной строке, перейдя в рабочую папку с программным модулем, выполните команду:

docker compose logs (Windows) или **sudo docker compose logs** (Linux)

Примечание: в среде Windows вы также можете выполнять отладку программного модуля и поиск неисправностей из графического интерфейса среды контейнеризации Docker Desktop.

Сброс программного модуля KeyManagement

Выполните [де-инициализацию](#) программного модуля. Удалите папки mysql, TempFiles и uploads из рабочей папки программного модуля. Выполните [инициализацию](#) программного модуля. Внимание! Все пользовательские данные программного модуля при сбросе к его исходному состоянию будут потеряны.

Обновление программного модуля KeyManagement

Выполните [де-инициализацию](#) программного модуля. Замените содержимое рабочей папки программного модуля на файлы новой версии (не удаляйте папку mysql для обеспечения переноса пользовательских данных). Выполните [инициализацию](#) программного модуля.

Удаление программного модуля KeyManagement

Выполните [де-инициализацию](#) программного модуля. Полностью удалите содержимое рабочей папки программного модуля. Внимание! Все пользовательские данные программного модуля при его удалении будут потеряны.

Программный модуль KeyCore

Запуск и остановка программного модуля KeyCore

При перезапуске ОС по умолчанию будет автоматически осуществляться также и запуск установленного в качестве сервиса программного модуля.

Для **ручной остановки, запуска и перезапуска** программного модуля в командной строке, перейдя в рабочую папку с программным модулем, выполните необходимую команду:

```
sudo systemctl stop KeyCore
```

```
sudo systemctl start KeyCore
```

```
sudo systemctl restart KeyCore
```

Конфигурация программного модуля KeyCore

Конфигурация программного модуля находится в файле **appsettings.json**. После изменения файла конфигурации необходимо выполнить перезапуск программного модуля, чтобы изменения вступили в силу.

По умолчанию файл конфигурации имеет следующий вид:

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Debug",
      "Microsoft": "Warning",
      "Microsoft.EntityFrameworkCore.Query": "Error",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "Kestrel": {
    "Endpoints": {
      "Http": { "Url": "http://0.0.0.0:5000" }
    }
  },
  "ConnectionStrings": {
    "SqliteConnection": "Data Source=KeyCore.db"
  },
  "AppSettings": {
    "DataSyncInterval": "00:00:05",
    "HeartBitInterval": "00:00:01",
    "BiometricSynchronizationInterval": "00:05:00",
    "DetectedUserLifeSpan": "00:00:05",
    "AuthorizationScreenTimeout": "00:00:10",
    "UserSessionTimeout": "00:00:20",
    "AlarmActivationTimeout": "00:00:10",
    "UserSessionCompletionDelay": "00:00:03",
```

ООО «Платформикс»
ОГРН 1037739996392
ИНН/КПП 7707507077/771301001

127018, г. Москва,
ул. Складочная, д. 3, стр. 1

```
"KeyBoxSheduleCheckInterval": "00:00:05",
"UsingOnlyCardReader": true,
"CardFilterBytes": 2,
"RequireControlIdentification": false,
"BackendUrl": "http://192.168.144.100/api",
"KeyBoxSerialNumber": 1,
"FilesFolderPath": "Photos",
"KeyBoxPingInterval": "00:00:00",
"KeysNumberParameter": 6,
"TerminalEnabled": false,
"LimitToCategoriesOnKeysOverflow": true,
"KeyBoxImplementation": {
  "AssemblyName": "KeyCore.KeyBox.Valberg",
  "ClassName": "KeyCore.KeyBox.Valberg.KeyBoxValberg",
  "Settings": {
    "KeyBoxHost": "192.168.144.2",
    "KeyBoxPort": 501,
    "RfidLen": 4,
    "MaxUnlockedCells": 6,
    "CellTimeout": 0
  }
}
}
```

Описание конфигурационных параметров:

- **Kestrel.Endpoints** - определяет, на каких портах сервер будет принимать входящие запросы, например от биометрического терминала и приложения KeyUI .
- **SQLiteConnection** - имя базы данных SQLite.
- **DataSyncInterval** - интервал двусторонней синхронизации данных (таких, как события, пользователи, политики и т.д.) между модулями KeyCore и KeyManagement.
- **HeartBitInterval** - интервал уведомлений модуля KeyManagement модулем KeyCore о наличии связи (см. параметр конфигурации TimeDifferenceForKeyBoxStatusNoConnection модуля KeyManagement).
- **BiometricSynchronizationInterval** - интервал загрузки данных пользователей в биометрический терминал.
- **DetectedUserLifeSpan** - длительность, в течении которой пользователь, распознанный биометрическим терминалом “в фоне” считается действительным и имеет возможность

пройти автоматическую идентификацию без необходимости прохождения явного распознавания лица. Пример: на экране планшета отображается надпись “Нажмите для обслуживания”, пользователь попал в поле видео камеры, был автоматически распознан и в течении 5 сек. коснулся экрана планшета, в этом случае приложение сразу перейдет в режим обслуживания, без доп. запроса идентификации.

- **AuthorizationScreenTimeOut** - время, в течении которого пользователь должен пройти контрольную идентификацию на биометрическом терминале или приложить бесконтактную карту, прежде чем пользовательская сессия будет завершена.
- **UserSessionTimeOut** - допустимая длительность бездействия пользователя. По истечении указанного значения с момента последней активности пользователя (операции с ключами или взаимодействие с приложением), будет предложено пройти контрольную идентификацию.
- **AlarmActivationTimeout** - время, отведенное на закрытие дверцы после завершения пользовательской сессии. Если после завершения пользовательской сессии, дверца не будет закрыта в течении заданного времени, будет активирован режим тревоги. Для отключения режима тревоги, необходимо закрыть дверцу.
- **UserSessionCompletionDelay** – (необязательный) - задает задержку завершения пользовательской сессии. Необходимо для обновления статусов ключей на случай, когда от системы хранения ключей по какой-либо причине не были получены события изъятия/возврата ключей. (рекомендуется 3 секунды для конфигураций со считывателем карт, и 5 секунд для конфигураций с биометрическими считывателями).
- **KeyBoxSheduleCheckInterval** - интервал проверки расписания на предмет включения/выключения устройства.

- **UsingOnlyCardReader** - определяет, используется ли в качестве идентификатора пользователя, ID карты, назначенной пользователю.
- **CardFilterBytes** – количество младших байт идентификатора, получаемого по интерфейсу Wiegand, интерпретируемых как идентификатор пользователя (при значении 0 – используется полное количество байт)
- **RequireControllIdentification** - определяет, требовать ли от пользователя дополнительной идентификации. Имеет смысл, только если параметр UsingOnlyCardReader - false. Если задан, то в случае автоматической идентификации (описан в параметре DetectedUserLifeSpan) от пользователя потребуется пройти доп. идентификацию. Целесообразно использовать для предотвращения несанкционированного доступа после идентификации пользователя, случайно попавшего в поле видео камеры.
- **BackendUrl** - URL модуля KeyManagement.
- **KeyBoxSerialNumber** - серийный номер данного устройства. Должен быть уникален.
- **FilesFolderPath** - путь к файлам фото пользователей, загруженным в рамках обмена данными с модулем KeyManagement.
- **KeysNumberParameter** - максимальное количество ключей, которым пользователь может назначить категории.
- **TerminalEnabled** - признак наличия планшета. От данного параметра зависит алгоритм выдачи ключей и подсветки пустых ячеек для сдачи ключей. Если включен, после идентификации пользователя, автоматически разблокируются (подлежат выдаче) только ключи, находящиеся в категориях. Если выключен, после идентификации, автоматически разблокируются все ключи, доступные пользователю, независимо от наличия категорий, а так же подсвечиваются пустые ячейки для сдачи ключей.

- **KeyBoxPingInterval** (необязательный) - задает интервал проверки связи с контроллером ключницы. В случае потери связи, модуль KeyCore переходит в режим блокировки с соответствующим сообщением в приложении планшета. Для восстановления работы, требуется перезапуск модуля KeyCore. Значение по умолчанию - 5 сек.
- **LimitToCategoriesOnKeysOverflow** – включить/выключить оптимизированный режим выдачи ключей для пользователей с большим количеством прав доступа.
- **KeyBoxImplementation.AssemblyName** - имя файла сборки драйвера ключницы
- **KeyBoxImplementation.ClassName** - имя класса драйвера ключницы
- **KeyBoxImplementation.Settings.KeyBoxHost** - IP адрес ключницы
- **KeyBoxImplementation.Settings.KeyBoxPort** - порт ключницы
- **KeyBoxImplementation.Settings.RfidLen** - длина RFID метки в байтах
- **KeyBoxImplementation.Settings.MaxUnlockedCells** - максимальное количество одновременно разблокированных ячеек.

Отладка программного модуля KeyCore

Для просмотра текущего состояния сервиса программного модуля в командной строке выполните команду:

```
sudo systemctl status KeyCore
```

Для просмотра логов сервиса программного модуля в командной строке выполните команду:

```
journalctl -fu KeyCore
```

Сброс программного модуля KeyCore

[Остановите](#) сервис программного модуля.

Удалите из рабочей папки **/opt/KeyCore** программного модуля следующие объекты:

- файл KeyCore.db – SQLite база данных программного модуля;
- файл key-box-token.txt – токен идентификации программного модуля KeyCORE в программном модуле KeyMGMT;
- файл data.txt – дополнительные данные, используемые программным модулем;
- папка photos – фотографии пользователей.

[Запустите](#) сервис программного модуля.

Внимание! Все пользовательские данные программного модуля будут потеряны.

Обновление программного модуля KeyCore

[Остановите](#) сервис программного модуля.

Скопируйте во временное расположение из рабочей папки **/opt/KeyCore** следующие объекты:

- файл KeyCore.db – SQLite база данных программного модуля;
- файл key-box-token.txt – токен идентификации программного модуля KeyCORE в программном модуле KeyMGMT;
- файл data.txt – дополнительные данные, используемые программным модулем;
- папка photos – фотографии пользователей.

[Удалите](#) программный модуль KeyCore.

Установите обновленный программный модуль KeyCore (воспользуйтесь руководством по установке). Верните ранее скопированные объекты в рабочую папку **/opt/KeyCore**.

[Запустите](#) сервис программного модуля.

Удаление программного модуля KeyCore

[Остановите](#) сервис программного модуля. В командной строке, перейдя в рабочую папку с программным модулем, выполните:

```
cd /opt/KeyCore/install  
sudo ./KeyCore.uninstall
```

Внимание! Все пользовательские данные программного модуля будут потеряны.

Программный модуль KeyUI

Запуск и остановка программного модуля KeyUI

Запускайте / останавливайте приложение средствами лаунчера вашего Android-устройства. В случае наличия в вашем Android-устройстве лаунчера с поддержкой режима «киоск», вы можете организовать автозапуск программного модуля и полностью ограничить доступ пользователей к иным элементам вашего Android-устройства.

Конфигурация программного модуля KeyUI

При первом запуске программного модуля KeyUI необходимо ввести IP-адрес того устройства, на котором запущен программный модуль KeyCore (по умолчанию 192.168.144.1) и номер порта (по умолчанию 5000).

Отладка программного модуля KeyUI

У программного модуля KeyUI отсутствуют встроенные средства отладки и поиска неисправностей. Для выполнения этих задач необходимо воспользоваться средствами отладки и поиска неисправностей, имеющимися на стороне программного модуля KeyCore.

Сброс программного модуля KeyUI

Выполните удаление данных (кэш, данные приложения) программного модуля через менеджер приложений вашего Android-устройства.

Обновление программного модуля KeyUI

Выполните удаление установленного программного модуля через менеджер приложений вашего Android-устройства. Выполните установку новой версии программного модуля.

Удаление программного модуля KeyUI

Выполните удаление установленного программного модуля через менеджер приложений вашего Android-устройства.