

**Программный комплекс
«Система автоматизированного учета ключей»**

Руководство по интеграции

Содержание

Введение	3
Интеграционный интерфейс	3
Пример интеграции	4
Аутентификация и авторизация	4
Использование токена API	5
Создание пользователя.....	6
Получение данных пользователя.....	7
Изменение данных пользователя.....	7
Блокирование пользователя	7
Удаление пользователя	8

Введение

Программный комплекс «Система автоматизированного учета ключей» поддерживает интеграцию с внешними системами на базе возможностей программного модуля управления KeyManagement.

Интеграционные возможности могут использоваться для следующих сценариев:

- синхронизация пользователей с внешними системами (например, СКУД) – наиболее частый сценарий интеграции;
- получение данных для кастомизируемых отчетов и оповещений о событиях;
- создание сложных логик обслуживания пользователей в связке с внешними системами.

Интеграционный интерфейс

Для интеграции системы предусмотрен интерфейс HTTP REST API. Техническая спецификация интерфейса доступна в виде веб-страницы по адресу: **http://keymanagement_host_ip/swagger** (где keymanagement_host_ip - IP адрес сетевого интерфейса в ОС, на которой установлен программный модуль KeyManagement).

Техническая спецификация интерфейса доступна в виде веб-страницы по указанному выше адресу только в случае установленной соответствующей конфигурационной опции.

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

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

Пример интеграции

Настоящий пример интеграции содержит описание функций, необходимых для обеспечения синхронизации справочника пользователей с внешней системой.

Аутентификация и авторизация

Начало сеанса работы – вход под учетной записью (в данном примере – admin/admin):

Тип запроса	POST
URL	/api/UserLoginAuth/authenticate
HTTP headers	Content-Type: application/json-patch+json
Тело запроса	{ "Login": "admin", "Password": "admin" }
Ответ - успешно	Status code 200 + данные авторизованного пользователя

Завершение сеанса работы:

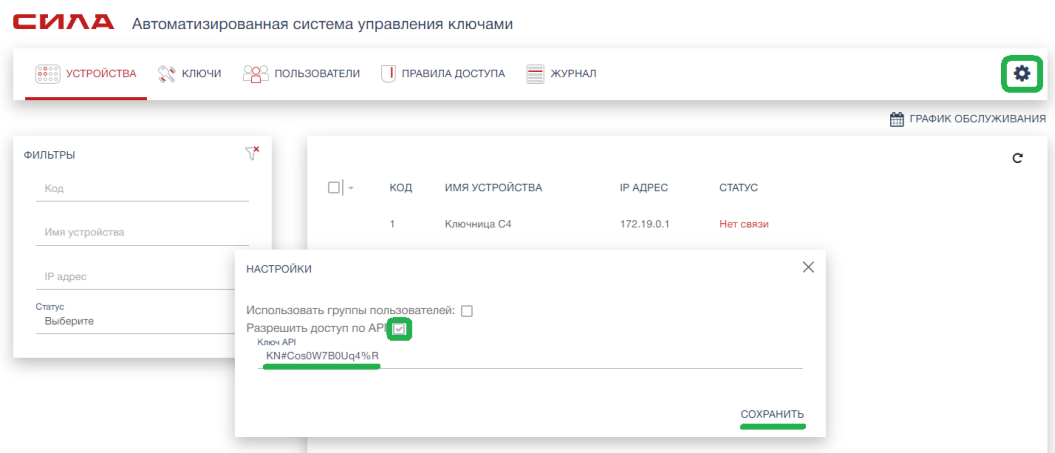
Тип запроса	POST
URL	/api/UserLoginAuth/logout
HTTP headers	нет
Тело запроса	нет
Ответ - успешно	Status code 200

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

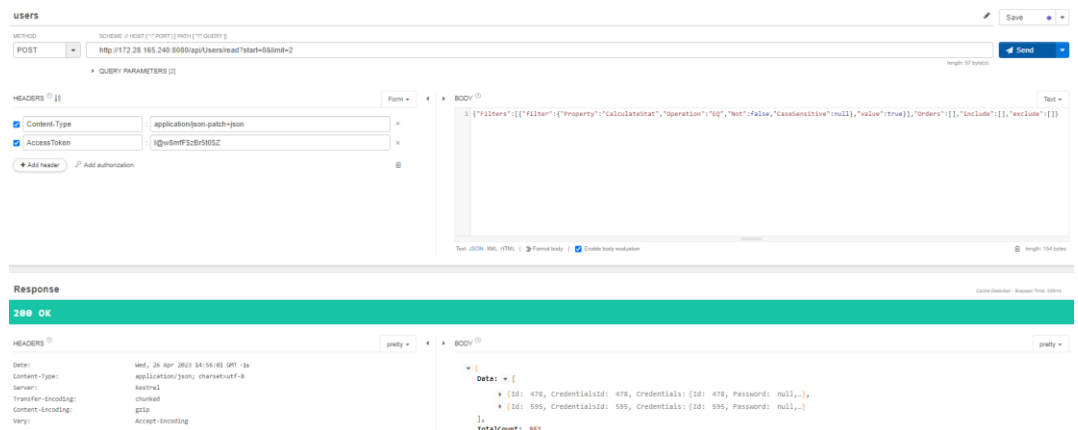
127018, г. Москва,
ул. Складочная, д. 3, стр. 1

Использование токена API

Также доступен вариант осуществления операций с API через токен, вставляемый в каждый запрос (в таком случае процедура аутентификации и авторизации, описанная в разделе «Аутентификация и авторизация», не потребуется):



После этого можно отправлять запросы к HTTP API с заголовком с именем **AccessToken** и значением, указанным в форме, см. пример ниже:



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

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

Создание пользователя

Тип запроса	POST
URL	/api/Users/create
HTTP headers	Content-Type: application/json-patch+json AccessToken: xxx
Тело запроса	<pre>{ "FullName": "Фамилия Имя Отчество", "Credentials": { "Password": null, "DevicesPassword": null, }, "Role": "User", "IdentificationCode": 12345678, "CardId": 16777215, "Group": { "Id": 1, "Name": "По умолчанию", }, }</pre> <p>Примечание 1. Представлен самый минимальный набор полей, необходимых для создания пользователя в логической группе «По умолчанию».</p> <p>Примечание 2. Для связи с внешней базой пользователей предлагается использовать поле IdentificationCode (уникальный идентификатор пользователя) – числовое значение.</p>
Ответ успешно	- Status code 200 + внутренний Id созданного пользователя

Получение данных пользователя

Тип запроса	POST
URL	/api/Users/readOne
HTTP headers	Content-Type: application/json-patch+json AccessToken: xxx
Тело запроса	<pre>{ "Filters": [{ "filter": { "Property": "IdentificationCode", "Operation": "EQ", "Not": false, "CaseSensitive": null }, "value": 12345678 }], "Orders": [], "include": [], "exclude": [] }</pre> <p>Примечание 1. Показан пример запроса данных пользователя по значению поля IdentificationCode, используемому для связи с внешней базой данных.</p>
Ответ успешно	- Status code 200 + данные пользователя

Изменение данных пользователя

Тип запроса	PATCH
URL	/api/Users/update
HTTP headers	Content-Type: application/json-patch+json AccessToken: xxx
Тело запроса	Полная структура с данными пользователя (см. «ПОЛУЧЕНИЕ ДАННЫХ ПОЛЬЗОВАТЕЛЯ») с необходимыми изменениями
Ответ - успешно	Status code 200

Блокирование пользователя

Блокирование пользователя производится через установку IsBlocked: true (см. раздел «Изменение данных пользователя»)

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

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

Удаление пользователя

Вариант 1:

Тип запроса	PATCH
URL	/api/Users/delete
HTTP headers	Content-Type: application/json-patch+json AccessToken: xxx
Тело запроса	[Одна или несколько полных структур с данными пользователя (см. «ПОЛУЧЕНИЕ ДАННЫХ ПОЛЬЗОВАТЕЛЯ»)] Примечание 1. При попытке удаления пользователя, которому выданы на руки ключи, возникает ошибка.
Ответ - успешно	Status code 200

Вариант 2:

Тип запроса	PATCH
URL	/api/Users/deleteBy
HTTP headers	Content-Type: application/json-patch+json
Тело запроса	{ "Filters": [{"filter": {"Property": "IdentificationCode", "Operation": "EQ", "Not": false, "CaseSensitive": null }, "value": 12345678 },], "Orders": [], "include": [], "exclude": [] } Примечание 1. Показан пример удаления пользователя по его IdentificationCode.
Ответ - успешно	Status code 200