



Бастион-2 – ИКС. Руководство программиста

Версия 2.5.13

(05.11.2024)



Самара, 2024



Оглавление

1	Общие сведения.....	4
2	Условия применения	5
3	Установка и настройка модуля	6
3.1	Общие сведения об установке.....	6
3.2	Установка модуля «Бастион-2 – ИКС» под Windows.....	6
3.3	Установка модуля «Бастион-2 – ИКС» под Linux	7
3.4	Запуск модуля «Бастион-2 – ИКС».....	8
3.5	Лицензирование	8
3.6	Настройка модуля	10
3.6.1	Настройка подключений по HTTPS.....	10
3.6.2	Настройка портов.....	10
3.6.3	Настройка схемы интеграции	11
4	Типы данных.....	13
4.1	EntryPoint	13
4.2	Person	13
4.3	TimeInterval.....	15
4.4	Pass	15
4.5	Org.....	16
4.6	Depart	16
4.7	DictVal.....	16
4.8	Attendance.....	16
4.9	CardEvent.....	17
4.10	BastionOperator	17
4.11	PersonBrief.....	17
4.12	PassBrief.....	17
4.13	MatValue.....	18
4.14	Car	18
4.15	MatValuePass	19
4.16	CarPass.....	19
4.17	Device.....	20
4.18	AccessLevel	20
4.19	AccessPoint	20
4.20	ControlArea.....	21
5	Методы web-API	21
5.1	GetVersion	22
5.2	Login.....	22
5.3	LogOff.....	23
5.4	CheckConnection	23
5.5	GetServers.....	23



5.6	GetEntryPoints.....	23
5.7	GetDictVals.....	24
5.8	GetCardEvents	24
5.9	GetAttendance.....	25
5.10	GetPasses	25
5.11	GetPassesByChangeDate	26
5.12	GetPersons.....	27
5.13	GetPersonsByChangeDate	28
5.14	GetPerson	29
5.15	GetPassCount.....	29
5.16	GetPassesByPerson.....	30
5.17	GetPassesByCardCode	30
5.18	GetPassesByPersonId.....	31
5.19	PutPass.....	31
5.20	BlockPass.....	33
5.21	UnBlockPass	34
5.22	ReturnPass	34
5.23	IssuePass	35
5.24	GetOrgs	36
5.25	PutOrg	36
5.26	UpdateOrg.....	37
5.27	DeleteOrg.....	37
5.28	GetDeparts.....	38
5.29	PutDepart.....	38
5.30	UpdateDepart	39
5.31	DeleteDepart.....	39
5.32	PutMVPass	40
5.33	GetMVPasses	41
5.34	GetMVPassesByPersonPass.....	41
5.35	IssueMVPass	41
5.36	WithdrawMVPass	42
5.37	PutCarPass	42
5.38	GetCarPasses.....	43
5.39	GetCarPassesByPersonPass	44
5.40	IssueCarPass.....	44
5.41	WithdrawCarPass.....	44
5.42	GetDevices	45
5.43	GetAccessLevels.....	46
5.44	GetAccessPoints.....	46
5.45	GetControlAreas	46



5.46	UpdatePersonData.....	47
5.47	GetQrPass	48
6	Нештатные ситуации.....	48
6.1	Проблемы с доступом на страницу конфигурации серверов.....	48
7	Приложения.....	49
7.1	Приложение 1 – Типы драйверов.....	49

1 Общие сведения

Модуль «Бастиян-2 – ИКС» (*Интеграция Кадровых Систем*) предназначен для организации информационного взаимодействия АПК «Бастиян-2» с системами управления предприятием (ERP) в части обмена данными СКУД (персонал, пропуски, проходы). «Бастиян-2 – ИКС» предоставляет API для интеграции и не содержит готовых конфигураций для каких-либо внешних систем.

Модуль «Бастиян-2 – ИКС» позволяет интегрировать:

- Кадровые системы (HRMS);
- Автоматизированные системы заказа пропусков (АСЗП);
- Бухгалтерские системы.

Модуль решает следующие задачи:

- Передача в АПК «Бастиян-2» пропусков и заявок на пропуск из внешней системы с возможностью указания прав доступа для СКУД и номера карты доступа;
- Передача в АПК «Бастиян-2» из внешней системы транспортных пропусков, пропусков на материальные ценности, а также заявок на транспортные пропуски и заявок на пропуск на материальные ценности;
- Активация персональных, транспортных и материальных пропусков в СКУД из внешней системы;
- Управление пропусками из внешней системы (блокировка, разблокировка, возврат);
- Получение из АПК «Бастиян-2» во внешнюю систему информации о персонах, персональных пропусках, транспортных пропусках, материальных пропусках, точках прохода, подразделениях, должностях и о других справочниках, доступных в АПК «Бастиян-2»;
- Получение из АПК «Бастиян-2» во внешнюю систему информации о последнем месте предъявления пропуска;
- Получение из АПК «Бастиян-2» во внешнюю систему списка событий по заданному пропуску;
- Получение из АПК «Бастиян-2» во внешнюю систему исходных данных для расчета отработанного времени (пары событий «вход-выход»);
- Получения из АПК «Бастиян-2» во внешнюю систему информации об устройствах АПК «Бастиян-2».

Система поддерживает одновременную работу с несколькими АПК «Бастиян-2». Их количество ограничивается лицензией на модуль «Бастиян-2 – ИКС».

Общая структура системы отображена на Рис. 1:

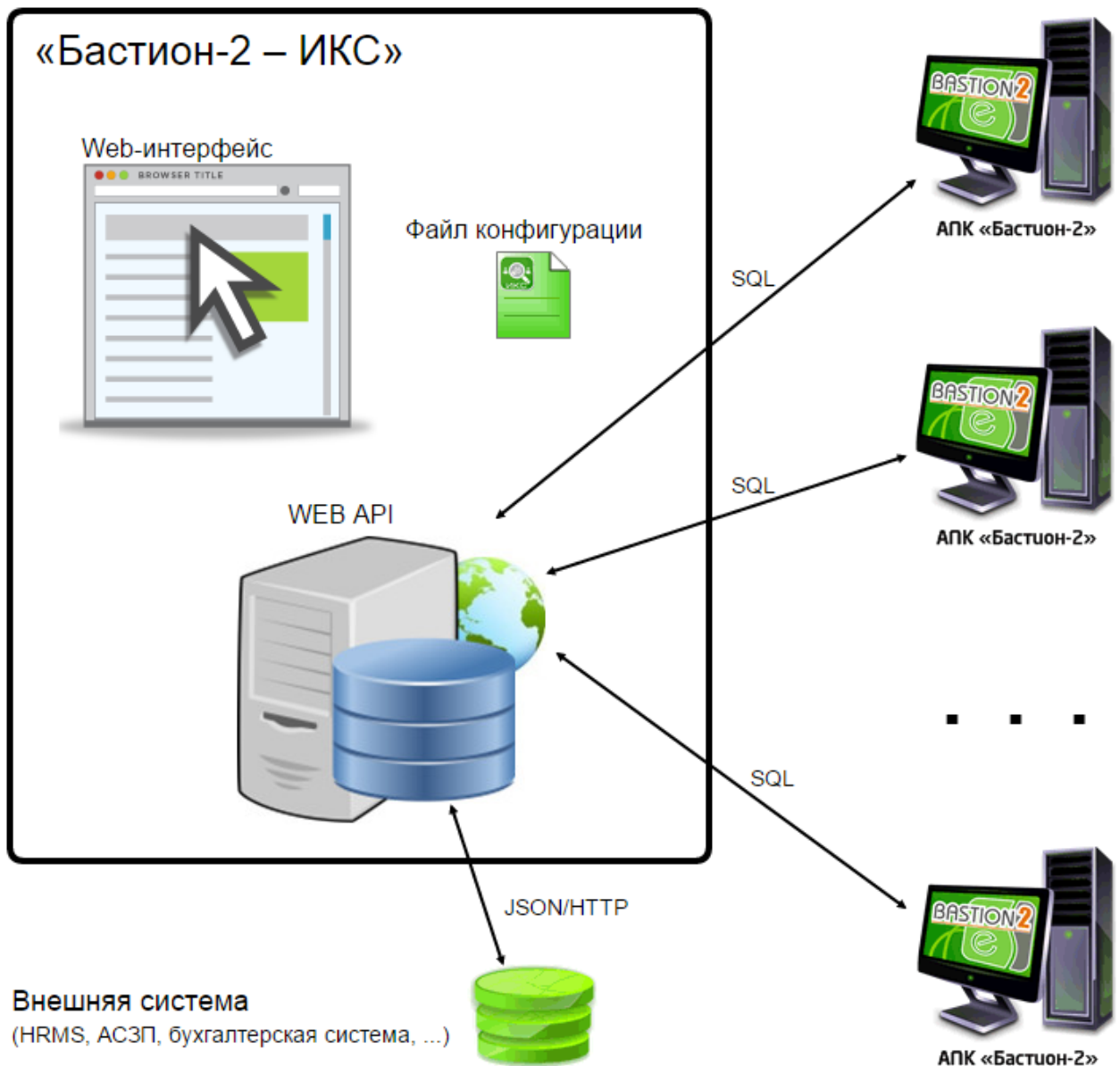


Рис. 1. Структура и схема взаимодействий модуля «Бастион-2 – ИКС»

Модуль «Бастион-2 – ИКС» представляет из себя службу Windows и предоставляет наружу web-API для взаимодействия со внешней системой и web-интерфейс, предназначенный для настройки конфигурации модуля. Конфигурация системы представляет собой список подключений к серверам АПК «Бастион-2» (схему интеграции) и сохраняется в файл servers.xml. Работа с web-интерфейсом возможна в любом браузере. Играть роль внешней системы может любой HTTP-клиент, взаимодействующий с web-API «Бастион-2 – ИКС». Обмен данными с web-API осуществляется в формате JSON. Поддерживается возможность подключений по протоколам HTTP и HTTPS.

2 Условия применения

Требования к аппаратной и программной конфигурации компьютера, на который устанавливается модуль:

- оперативная память – 1 Гб (и выше);
- тактовая частота процессора – 2 ГГц (и выше);



- жёсткий диск – 1 Гб свободного места на системном диске;
- операционная система семейства Windows – **Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016;**

Список поддерживаемых операционных систем семейства Linux:

- Red Hat Enterprise Linux x64, версии 6, 7;
- CentOS x64, версия 7;
- Oracle Linux x64, версия 7;
- Fedora x64, версии 27, 28;
- Debian x64, версии 9, 8.7 и выше;
- Ubuntu x64, версии 18.04, 17.10, 16.04, 14.04;
- Linux Mint x64, версии 18, 17;
- OpenSUSE x64, версия 42.3 и выше;
- SUSE Enterprise Linux (SLES) x64, версия 12 SP2 и выше;
- Alpine Linux x64, версия 3.7 и выше;
- ALT Linux x64, версия 8.0 и выше;
- Astra Linux x64, версия 2.11.5 и выше.

Установка дополнительных компонентов или фреймворков в Linux не требуется.

Система предназначена для совместной работы с АПК «Бастион-2» версий 2.1.8 PostgreSQL и выше.

Web-приложение «Бастион-2 – ИКС» должно быть доступно по протоколу HTTP или HTTPS со всех рабочих станций, где предполагается размещение внешних систем – клиентов модуля интеграции, а так же модуль должен иметь доступ по сети ко всем серверам БД АПК «Бастион-2», которые прописаны в конфигурации.

3 Установка и настройка модуля

3.1 Общие сведения об установке

Для установки модуля «Бастион-2 – ИКС» потребуются права администратора системы.

Если установка производится на ОС Windows 7, или Windows Server 2008 R2, то необходимо убедиться, что операционная система находится в актуальном состоянии, т.е. установлены все обязательные обновления системы.

Пакет установки для операционных систем семейства Linux поставляется в виде ZIP-архива.

3.2 Установка модуля «Бастион-2 – ИКС» под Windows

Для установки модуля «Бастион-2 – ИКС» необходимо запустить файл инсталлятора **IKS.msi** и следовать указаниям мастера установки.

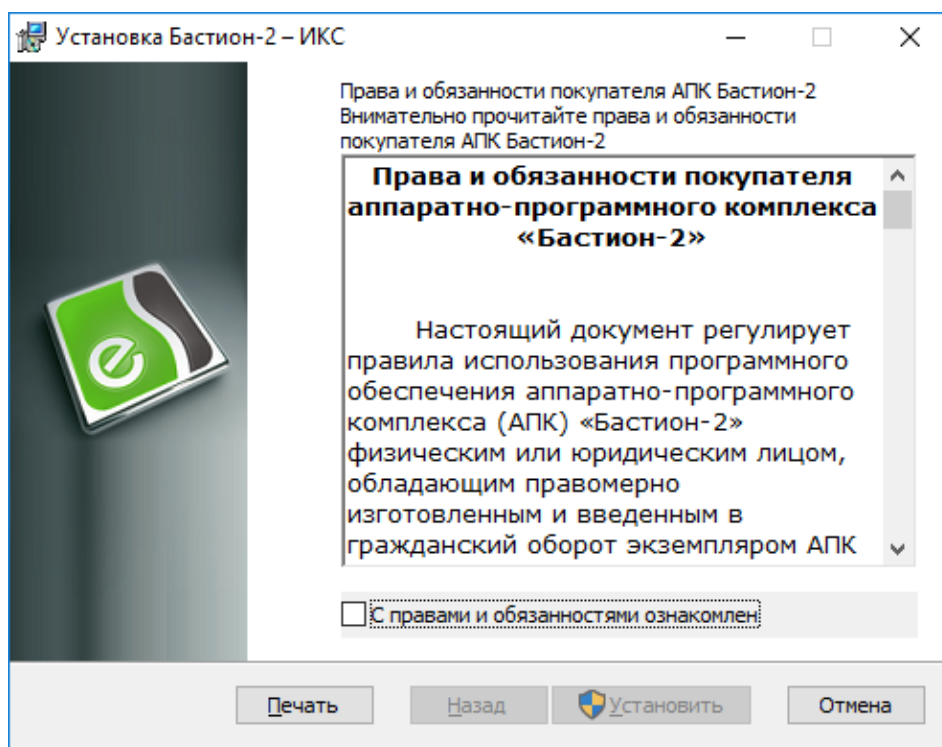


Рис. 2. Установка модуля «Бастион-2 – ИКС»

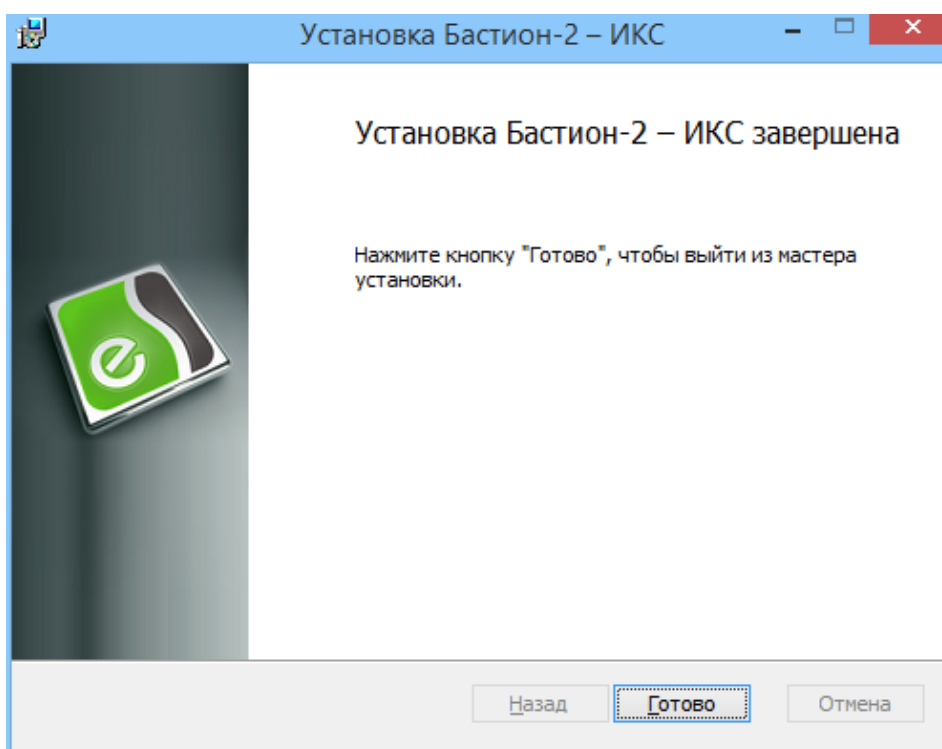


Рис. 3. Завершение установки

Внимание! Если служба «Бастион-2 – ИКС» не запускается, убедитесь, что установлены все обновления ОС Windows.

3.3 Установка модуля «Бастион-2 – ИКС» под Linux

Для установки модуля «Бастион-2 – ИКС» необходимо выполнить следующую последовательность шагов:

1. Поместить папку `iks` вместе с её содержимым в директорию `/var`;
2. Разрешить выполнение файла `IKS`, выполнив команду «`chmod u+x /var/iks/IKS`»;
3. Поместить файл `iks.service` в папку `/etc/systemd/system/`;
4. Выполнить в терминале команду «`systemctl enable iks`»;
5. Выполнить команду «`systemctl start iks`»;
6. Убедиться, что сервис был успешно установлен и запущен, выполнив команду `systemctl status iks`. Статус сервиса должен быть «Active» (Рис. 4).

```
eejkee9@ubuntu:~$ systemctl status iks
● iks.service - IKS 2.3
   Loaded: loaded (/etc/systemd/system/iks.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2018-06-19 03:52:20 PDT; 2 days ago
     Main PID: 778 (IKS)
    CGroup: /system.slice/iks.service
            └─778 /var/iks/IKS
```

Рис. 4. Статус активного демона «iks»

После выполнения приведенной выше последовательности действий сервис `iks` будет установлен и запускаться автоматически при старте операционной системы.

Все остальные действия по настройке модуля «Бастион-2 – ИКС» аналогичны действиям в ОС Windows.

3.4 Запуск модуля «Бастион-2 – ИКС»

Модуль запускается автоматически в виде службы после установки, и, в дальнейшем, при каждом запуске операционной системы.

Доступ к web-интерфейсу модуля доступен по адресу http://*:5005 и/или https://*:5006 (в зависимости от того, с какой конфигурацией запущен модуль), где * - сетевое имя рабочей станции (localhost, 127.0.0.1, сетевой адрес или доменное имя компьютера).

По умолчанию web-API и web-интерфейс модуля «Бастион-2 – ИКС» доступен по порту 5005 в режиме http. Для настройки подключений по протоколу https обратитесь к пункту 3.6.1 данного руководства.

3.5 Лицензирование

Перед началом работы необходимо получить лицензию на использование модуля «Бастион-2 – ИКС». Для этого необходимо зайти в web-интерфейс по его сетевому адресу, в результате чего откроется страница следующего вида:

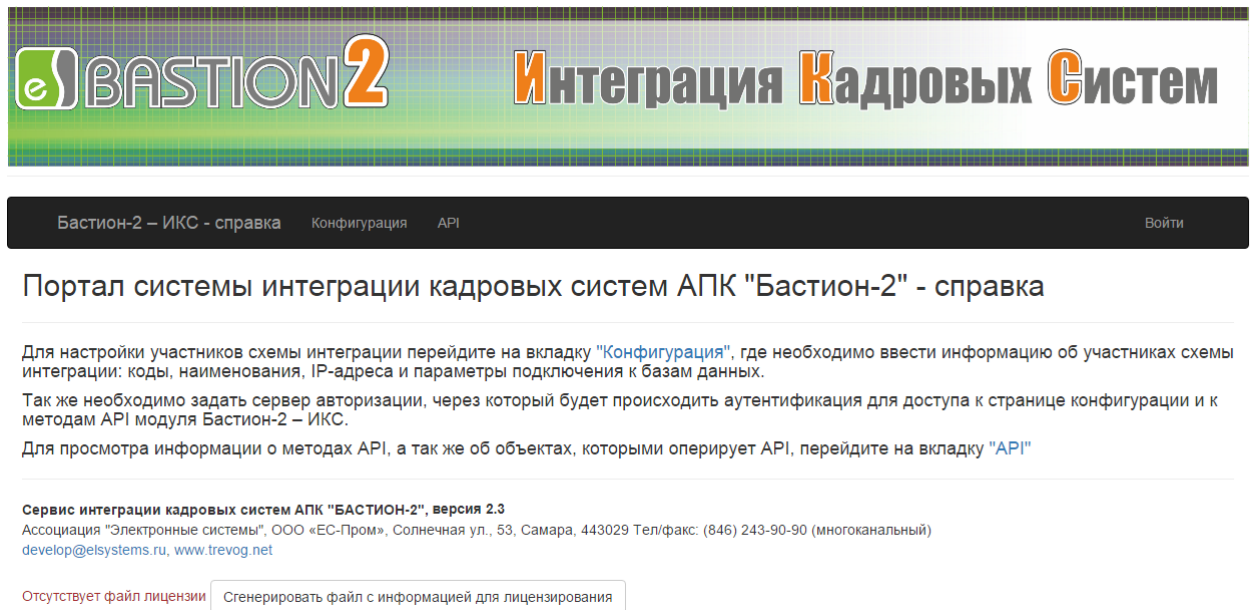


Рис. 5. Web-интерфейс

Необходимо нажать на кнопку «Сгенерировать файл с информацией для лицензирования», в результате чего будет создан файл **IKSDataForLicense.dat**, который нужно выслать в службу технической поддержки АПК «Бастион-2».

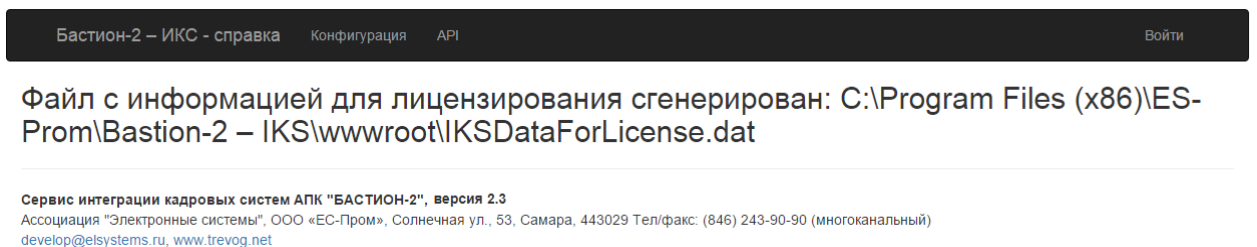


Рис. 6. Результат нажатия на кнопку «Сгенерировать файл с информацией для лицензирования»

После того как служба технической поддержки пришлёт лицензионный файл **ikslicense.dat**, его необходимо положить в папку **wwwroot** приложения (%ProgramFiles%\ES-Prom\Bastion-2 – IKS\wwwroot). Количество серверов БД АПК «Бастион-2», сконфигурированных в файле **servers.xml**, не должно превышать число, прописанное в лицензионном файле.

После того, как файл **ikslicense.dat** будет размещен в папке **wwwroot**, нижняя часть страниц web-интерфейса примет вид, подобный изображенному на рисунке:

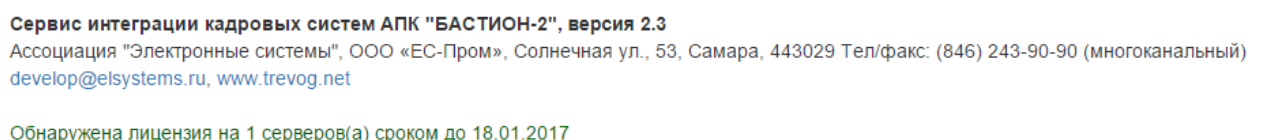


Рис. 7. Обнаружена лицензия

Строка зеленого цвета «Обнаружена лицензия на N серверов сроком до D» сигнализирует о том, что модуль опознал корректный файл лицензии, N – количество лицензированных серверов, а D – дата, до которой будет действовать лицензия.

3.6 Настройка модуля

3.6.1 Настройка подключений по HTTPS

Внимание! Для настройки доступа по протоколу HTTPS необходим корректный сертификат SSL в виде файла формата *.pfx.

В случае, если требуется работа модуля «Бастион-2 – ИКС» с обеспечением подключений по протоколу https, необходимо отредактировать конфигурацию модуля. Для этого нужно открыть файл **host.json**, располагающийся в директории %ProgramFiles%\ES-Prom\Bastion-2 – IKS\wwwroot.

Для активации режима https необходимо изменить значение переменной **UseHttps** со значения «0» на значение «1» (активировано), а также задать путь к файлу сертификата (значение переменной **CertificatePath**) и пароль сертификата (значение переменной **CertificatePassword**, которое следует оставить пустым в случае, если сертификат не защищен паролем). Путь к файлу сертификата следует указывать, используя двойные обратные слеша «\».

После выполнения этих действий содержимое файла **host.json** должно иметь вид, подобный изображенному на рисунке:

```
{
  "SSL": {
    "CertificatePath": "C:\\sertificat.pfx",
    "CertificatePassword": "12345"
  },
  "HOST": {
    "UseHttp": "0",
    "UseHttps": "1",
    "HttpPort": "5005",
    "HttpsPort": "5006"
  }
}
```

Рис. 8. Файл **host.json**

На рисунке изображена конфигурация модуля, при котором он будет доступен только по порту 5006 в режиме https. Если требуется, чтобы модуль был доступен по обоим протоколам одновременно, следует значения обеих переменных **UseHttp** и **UseHttps** установить в значение «1».

После изменения и сохранения изменений файла конфигурации следует перезапустить службу **iks**, либо перезагрузить компьютер целиком.

3.6.2 Настройка портов

По умолчанию web-API и web-интерфейс модуля «Бастион-2 – ИКС» доступны по порту 5005 в режиме http и 5006 в режиме https. Если требуется изменить эти порты, то это возможно сделать посредством внесения изменений в файл конфигурации **host.json**, располагающийся в директории %ProgramFiles%\ES-Prom\Bastion-2 – IKS\wwwroot.

Для того чтобы изменить порт подключения по **http**, требуется изменить параметр **HttpPort**, а если нужно изменить порт подключения в режиме **https**, то необходимо скорректировать значение параметра **PortHttps** (см. Рис. 8).

3.6.3 Настройка схемы интеграции

На этапе подготовки к работе должны быть развёрнуты сервера АПК «Бастион-2» в требуемом количестве. Каждый сервер должен быть запущен и сконфигурирован. На каждом сервере АПК «Бастион-2» должен быть установлен драйвер СКУД и сконфигурировано хотя бы одно устройство.

Настройка схемы интеграции заключается в определении и создании описания для каждого из участников.

Для включения одного сервера в систему интеграции необходимо:

1. Войти на web-интерфейс по адресу http://*:5005 (или https://*:5006, в зависимости от того, в каком режиме запущен модуль).
2. Перейти на страницу «Конфигурация». Страница имеет следующий вид:

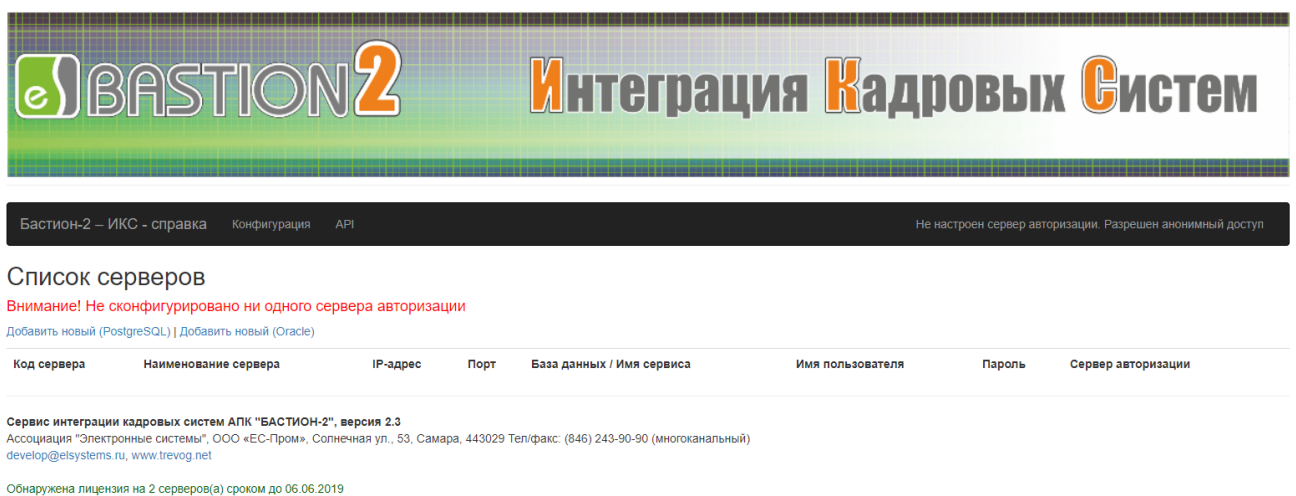


Рис. 9. Список участников интеграции

Внимание! Доступ к странице конфигурации не требует авторизации в том случае, если в схеме интеграции не определен ни один сервер авторизации. В дальнейшем, после настройки схемы для доступа к конфигурированию необходима аутентификация через сервер авторизации, роль которого могут выполнять один или несколько участников схемы.

3. Перейти по ссылке «Добавить новый»). Откроется страница, содержащая форму добавления нового сервера:

Бастион-2 – ИКС - справка Конфигурация API

Добавление нового сервера

Код сервера

Наименование сервера

IP-адрес

Имя сервиса

Порт

Имя пользователя

Пароль

Является сервером авторизации

|

Рис. 10. Добавление нового участника схемы интеграции

4. Ввести в соответствующие столбцы описание участника (для примера):

- Код сервера – текстовый идентификатор сервера, используемый в запросах и в некоторых типах данных для идентификации сервера-владельца записи;
- Наименование сервера – описание сервера, комментарий к нему;
- IP-адрес – IP адрес или имя хоста сервера БД;
- Порт – порт сервиса БД;
- База данных;
- Имя пользователя БД;
- Пароль пользователя БД;
- Указать, является ли участник сервером авторизации.

5. По окончании ввода нажать кнопку «Добавить».

Подобным образом необходимо выполнить создание описаний для всех клиентов сервиса интеграции.

Бастион-2 – ИКС - справка Конфигурация API

Вход.

Введите имя пользователя и пароль оператора Бастиона сервера, помеченного как сервер авторизации

Логин оператора

Пароль оператора

Рис. 11. Авторизация пользователя для доступа к странице конфигурации

Внимание! В конфигурации должен присутствовать по меньшей мере один сервер авторизации.

Через сервер авторизации будет происходить аутентификация всех запросов к веб-сервису, а также доступ к странице конфигурирования схемы интеграции. Если в схеме не указан сервер авторизации, то web-интерфейс предоставляет возможность зайти на страницу конфигурации без авторизации, а также на странице будет отображаться предупреждающее сообщение «Внимание! Не сконфигурировано ни одного сервера авторизации», как на Рис. 9.

Внимание! В случае потери связи с серверами авторизации пропадает доступ к странице конфигурации. В таком случае необходимо восстановить доступность по сети сервера авторизации, либо отредактировать файл конфигурации *servers.xml* (распологающийся в директории %ProgramFiles%\ES-Prom\Bastion-2 – IKS\wwwroot) вручную: убрать из него все пометки серверов авторизации, заменив для всех серверов значение тега *IsAuthenticationServer* с *true* на *false*, либо удалить файл *servers.xml* целиком (в этом случае файл будет создан заново автоматически при создании новой конфигурации). Если в схеме не определен ни один сервер авторизации, то к странице конфигурации возможен анонимный доступ (до тех пор, пока не будет добавлен сервер авторизации в схему).

Заключительным этапом настройки схемы интеграции является проверка связи с каждым объектом (сервером), участвующим в схеме. Для проверки необходимо в каждой строке таблицы серверов нажать кнопку «Проверить соединение» . Результатом выполнения этого действия будет открытие новой страницы с JSON-описанием состояния связи с сервером вида SRV1:OK, где SRV1 – код сервера, а строка “OK” сигнализирует об успешной проверке связи с сервером (в случае ошибки вместо “OK” будет строка с описанием проблемы).

4 Типы данных

4.1 EntryPoint

EntryPoint – объект, представляющий точку (направление) прохода - считыватель в СКУД АПК «Бастион-2». Используется в качестве выходного параметра в методе *GetEntryPoints*, возвращающем коллекцию точек прохода. Кроме того, объекты типа *EntryPoint* являются полями в типах *Pass* и *CardEvent*.

№ п/п	Имя	Тип	Описание
1	ServID	String	Идентификатор сервера – владельца точки прохода
2	SubDeviceNo	Int32?	Идентификатор устройства на сервере
3	SubDeviceName	String	Имя точки прохода

4.2 Person

Объект типа *Person* представляет собой набор данных о персоне. Объект типа *Person* является полем объекта *Pass*, использующегося в методе создания редактирования пропуска/заявки на пропуск *PutPass* и метода получения коллекции пропусков *GetPasses*.



№ п/п	Имя	Тип	Описание	Обязательность заполнения
1	PersonId	int?	Идентификатор персоны	Нет
2	Name	string	Фамилия	Да
3	FirstName	string	Имя	Да
4	SecondName	string	Отчество	Нет
5	TableNo	string	Табельный номер	Нет
6	PersonCat	string	Категория	Нет
7	Org	string	Организация	Да
8	Dep	string	Департамент	Да
9	Post	string	Должность	Да
10	Comments	String	Комментарии	Нет
11	DocIssueOrgan	string	Орган, выдавший документ, удостоверяющий личность	Нет
12	DocSer	string	Серия документа	Нет
13	DocNo	string	Номер документа	Нет
14	DocIssueDate	DateTime?	Дата выдачи документа	Нет
15	BirthDate	DateTime?	Дата рождения	Нет
16	BirthPlace	string	Место рождения	Нет
17	Address	string	Адрес прописки	Нет
18	Phone	string	Телефон	Нет
19	Foto	string	Фотография в виде Base64-строки	Нет
20	AddField1	string	Дополнительное поле 1	Нет
21	AddField2	string	Дополнительное поле 2	Нет
22	AddField3	string	Дополнительное поле 3	Нет
23	AddField4	string	Дополнительное поле 4	Нет
24	AddField5	string	Дополнительное поле 5	Нет
25	AddField6	string	Дополнительное поле 6	Нет
26	AddField7	string	Дополнительное поле 7	Нет
27	AddField8	string	Дополнительное поле 8	Нет
28	AddField9	string	Дополнительное поле 9	Нет
29	AddField10	string	Дополнительное поле 10	Нет
30	AddField11	string	Дополнительное поле 11	Нет
31	AddField12	string	Дополнительное поле 12	Нет
32	AddField13	string	Дополнительное поле 13	Нет
33	AddField14	string	Дополнительное поле 14	Нет
34	AddField15	string	Дополнительное поле 15	Нет
35	AddField16	string	Дополнительное поле 16	Нет
36	AddField17	string	Дополнительное поле 17	Нет
37	AddField18	string	Дополнительное поле 18	Нет
38	AddField19	string	Дополнительное поле 19	Нет
39	AddField20	string	Дополнительное поле 20	Нет
40	CreateDate	DateTime?	Дата создания	Нет
41	DocType	string	Тип документа, удостоверяющий личность	Нет
42	Citizenship	string	Гражданство	Нет
43	Email	string	Электронная почта	Нет
44	Gender	int?	Пол	Нет

4.3 TimeInterval

Объект типа TimeInterval представляет собой временной промежуток, фигурирующий в уровне доступа, привязанному к пропуску в СКУД АПК «Бастион-2». Объект типа TimeInterval является полем в типе данных Pass, используемом в методе создания/редактирования пропуска/заявки на пропуск PutPass и в методе получения коллекции пропусков GetPasses.

№ п/п	Имя	Тип	Описание
1	TimeStart	string	Время начала действия пропуска в течение дня, указывается в форматах «HH:MM:SS» либо «HH:MM», другие форматы не допускаются. Примеры правильных данных: "02:03", "15:30", "12:35:30", "01:22:15"
2	TimeEnd	string	Время окончания действия пропуска в течение дня. Формат передачи времени аналогичен полю TimeStart
3	InSaturday	Int32	1/0 – Разрешение/запрет прохода в субботу (по умолчанию 0)
4	InSunday	Int32	1/0 – Разрешение/запрет прохода в воскресенье (по умолчанию 0)
5	InHolidays	Int32	1/0 – Разрешение/запрет прохода в праздники и короткие дни (по умолчанию 0)

4.4 Pass

Тип Pass представляет набор данных пропуска или заявки на пропуск. Используется в методе создания/редактирования пропуска/заявки на пропуск и в методе получения списка пропусков.

№ п/п	Название поля	Тип	Описание
1	CardCode	string	Код карты доступа (может быть пустым в том случае если объект представляет набор данных заявки на пропуск)
2	PersonData	Person	Набор персональных данных пропуска (класс Person описан в пп. 4.2)
3	PassType	Int32	Тип пропуска (Постоянный - 1, Временный - 2, Разовый - 4)
4	DateFrom	DateTime?	Дата начала действия пропуска (может быть пустой)
5	DateTo	DateTime?	Дата окончания действия пропуска (может быть пустой)
6	CardStatus	Int32	Статус пропуска
7	TimeInterval	TimeInterval	Временной интервал (класс TimeInterval описан в пп. 4.3)
8	EntryPoints	EntryPoint[]	Массив точек прохода (класс EntryPoint описан в пп. 4.1)
9	AccessLevels	AccessLevel[]	Массив уровней доступа (класс AccessLevel описан в пп 4.18)
10	PassCat	string	Категория пропуска
11	CreateDate	DateTime?	Дата создания
12	IssueDate	DateTime?	Дата выдачи
13	Pincode	Int64	Pin-код пропуска
14	IsBlocked	Boolean	Заблокирован ли пропуск
15	BlockedReason	string	Причина блокировки

4.5 Org

Тип Org представляет собой данные организации. Используется в методах получения, создания, редактирования и удаления организаций.

№ п/п	Название поля	Тип	Описание
1	OrgName	string	Полное имя организации (включая путь и все дочерние организации, разделенные символом "\ " (отделённым от названий пробелами с обеих сторон)

4.6 Depart

Тип Depart представляет собой данные подразделения и используется в методах получения, создания, редактирования и удаления подразделений.

№ п/п	Название поля	Тип	Описание
1	DepName	string	Полное имя подразделения (включая путь и все дочерние подразделения, разделенные символом "\ " (отделённым от названий пробелами с обеих сторон)
2	OrgName	string	Полное имя родительской организации (включая путь и все дочерние организации, разделенные символом "\ " (отделённым от названий пробелами с обеих сторон)

4.7 DictVal

Тип DictVal представляет данные словарного значения. Используется в методе запроса словарных значений GetDictVals.

№ п/п	Название поля	Тип	Описание
1	Category	Int32	Категория словарного значения
2	Value	string	Текст словарного значения

4.8 Attendance

Тип Attendance представляет собой данные о посещении с картой доступа. Используется в методе запроса посещений GetAttendance.

№ п/п	Название поля	Тип	Описание
1	CardCode	string	Код карты, с которой произошло посещение
2	IsEntrance	bool	Флаг, определяющий, является ли посещение входом (true), либо выходом (false)
3	DateTime	DateTime?	Дата и время возникновения посещения
4	Comments	string	Комментарий
5	CtrlArea	string	Зона контроля
6	Tableno	string	Табельный номер персоны
7	PersonData	PersonBrief	Краткий набор персональных данных пропуска (класс PersonBrief описан в пп. 4.11)

4.9 CardEvent

Тип CardEvent представляет собой набор данных о событии, произошедшем с картой доступа. Используется в методе получения события с картой GetCardEvents.

№ п/п	Название поля	Тип	Описание
1	CardCode	string	Код карты, с которой произошло событие
2	EntryPoint	AccessPoint	Точка доступа, на которой произошло событие с картой (описание класса AccessPoint приведено в пп 4.19)
3	DateTime	DateTime?	Дата и время возникновения события
4	MsgText	string	Текст события
5	MsgCode	Int32	Код события
6	MsgType	Int32	Тип события
7	Comments	String	Комментарий
8	Photo	String	Фотография, прикрепленная к событию

4.10 BastionOperator

Тип BastionOperator представляет собой набор данных об операторе «Бастион-2 – ИКС», учетные данные которого используются для авторизации. Объект типа BastionOperator используется в качестве входного параметра в методе авторизации Login.

№ п/п	Название поля	Тип	Описание
1	Opername	string	Логин оператора
2	Password	String	Пароль оператора

4.11 PersonBrief

Тип PersonBrief представляет собой краткий набор данных персоны. Объект типа PersonBrief является полем объекта PassBrief, который является полем объектов типа MatValuePass и CarPass, используемых в методах создания и редактирования транспортных и материальных пропусков и заявок PutMVPass и PutCarPass, а также в методах получения списков материальных и транспортных пропусков GetMVPasses, GetMvPassesByPersonPass, GetCarPasses, GetCarPassesByPersonPass.

№ п/п	Имя	Тип	Описание	Обязательность заполнения
1	Name	string	Фамилия	Да
2	FirstName	string	Имя	Нет
3	SecondName	string	Отчество	Нет
4	BirthDate	DateTime?	Дата рождения	Нет

4.12 PassBrief

Тип Pass представляет краткий набор данных персонального пропуска или заявки на пропуск. Используется в методах получения материальных и транспортных пропусков GetMVPassesByPersonPass, GetCarPassesByPersonPass, а также является полем типов MatValuePass и CarPass, используемых в методах создания и редактирования транспортных и материальных пропусков и заявок PutMVPass и PutCarPass, а также в методах получения списков материальных и

транспортных пропусков GetMVPasses, GetMVPassesByPersonPass, GetCarPasses, GetCarPassesByPersonPass.

№ п/п	Название поля	Тип	Описание
1	CardCode	String	Код карты, выданной на пропуск
2	PersonData	PersonBrief	Краткий набор персональных данных пропуска (класс PersonBrief описан в пп. 4.11)
3	PassType	Int32	Тип пропуска (Постоянный - 1, Временный - 2, Разовый - 4)
4	CardStatus	Int32	Статус пропуска

4.13 MatValue

Тип MatValue представляет собой набор данных материальной ценности. Объект типа MatValue является полем объекта MatValuePass, использующегося в методах создания и редактирования, а также получения списков материальных пропусков и заявок GetMVPasses, GetMVPassesByPersonPass и PutMVPass.

№ п/п	Имя	Тип	Описание	Обязательность заполнения
1	Type	string	Тип материальной ценности	Да
2	ValFld1	String	Количество	Нет
3	ValFld2	String	Вес	Нет
4	ValFld3	String	Объем	Нет
5	ValFld4	String	Номер доверенности	Нет
6	ValFld5	String	Кем выдана доверенность	Нет
7	ValFld6	String	Номер накладной	Нет
8	ValFld7	String	Кем выдана накладная	Нет
9	ValFld8	String	№ вет. Свидетельства	Нет
10	ValFld9	String	Номер разрядки	Нет
11	ValFld10	String	Куда предназначается	Нет
12	ValFld11	String	Дополнительное поле	Нет
13	ValFld12	String	Дополнительное поле	Нет
14	ValFld13	String	Дополнительное поле	Нет
15	ValFld14	String	Дополнительное поле	Нет
16	ValFld15	String	Дополнительное поле	Нет
17	ValFld16	String	Дополнительное поле	Нет
18	ValFld17	String	Дополнительное поле	Нет
19	ValFld18	String	Дополнительное поле	Нет
20	ValFld19	String	Дополнительное поле	Нет
21	ValFld20	String	Дополнительное поле	Нет
22	OrderNum	Int32?	Порядковый номер	Нет

4.14 Car

Тип Car представляет собой набор данных транспортного средства. Объект типа Car является полем объекта CarPass, использующегося в методах создания и редактирования, а так же получения списков транспортных пропусков и заявок GetCarPasses, GetCarPassesByPersonPass и PutCarPass.

№ п/п	Имя	Тип	Описание	Обязательность заполнения
1	RegNo	String	Регистрационный номер	Да
2	CarModel	String	Модель	Нет
3	CarDescription	String	Описание	Нет
4	CarPhoto	String	Фотография	Нет
5	CarYear	Int32?	Год выпуска	Нет
6	CarColorStr	String	Цвет	Нет
7	CarWeight	String	Вес	Нет
8	CarVolume	String	Версия	Нет
9	CarOwnerStr	String	Владелец	Нет
10	CarTypeStr	String	Тип	Нет

4.15 MatValuePass

Тип `MatValuePass` представляет собой набор данных материального пропуска или заявки на пропуск. Объект типа `MatValuePass` используется в методах создания и редактирования, а также получения списков материальных пропусков и заявок `GetMVPasses`, `GetMVPassesByPersonPass` и `PutMVPass`.

№ п/п	Имя	Тип	Описание	Обязательность заполнения
1	Id	Int32?	Идентификатор пропуска	Нет
2	PassNum	String	Номер пропуска	нет
3	CreateDate	DateTime?	Дата создания	Нет
4	MatPerson	PersonBrief	Материально-ответственное лицо	Нет
5	ToExport	Boolean	На вынос	Да
6	ToImport	Boolean	На внос	Да
7	Status	Int32?	Статус пропуска	Нет
8	MatValues	MatValue[]	Коллекция материальных ценностей	Да
9	Pass	PassBrief	Персональный пропуск	Да
10	StartDate	DateTime?	Дата начала действия	Да
11	EndDate	DateTime?	Дата окончания действия	Да
12	GoalOrganization	String	Организация назначения	Нет
13	GoalDepartment	String	Подразделение назначения	Нет

4.16 CarPass

Тип `CarPass` представляет собой набор данных транспортного пропуска или заявки на пропуск. Объект типа `CarPass` используется в методах создания и редактирования, а также получения списков транспортных пропусков и заявок `GetCarPasses`, `GetCarPassesByPersonPass` и `PutCarPass`.

№ п/п	Имя	Тип	Описание	Обязательность заполнения
1	Id	Int32?	Идентификатор пропуска	Нет
2	PassNum	String	Номер пропуска	Нет
3	DateCreate	DateTime?	Дата создания	Нет
4	Status	Int32?	Статус пропуска	Нет
5	Cars	Car[]	Коллекция транспорта	Да
6	Pass	PassBrief	Персональный пропуск	Да

№ п/п	Имя	Тип	Описание	Обязательность заполнения
7	StartDate	DateTime?	Дата начала действия	Да
8	EndDate	DateTime?	Дата окончания действия	Да

4.17 Device

Тип Device представляет собой набор данных устройства. Объект типа Device используется в методе получения набора устройств GetDevices.

№ п/п	Имя	Тип	Описание
1	Sdn	Int32?	Идентификатор устройства
2	ParentSdn	Int32?	Идентификатор родительского устройства
3	DriverId	Int32?	Идентификатор типа драйвера, которому принадлежит устройство
4	Name	String	Наименование устройства
5	DeviceType	Int32?	Код типа устройства
6	DeviceTypeName	String	Текстовое имя типа устройства
7	Childs	Device[]	Коллекция дочерних устройств

4.18 AccessLevel

Тип AccessLevel представляет собой набор данных уровня доступа. Объект типа AccessLevel используется в методе создания/редактирования пропуска/заявки на пропуск PutPass, в методе получения списка пропусков GetPasses, а также в методе получения списка уровней доступа GetAccessLevels (пп. 5.38).

№ п/п	Имя	Тип	Описание
1	ServID	String	Идентификатор сервера – владельца точки прохода
2	Id	Int64?	Идентификатор уровня доступа
3	Name	String	Имя уровня доступа

4.19 AccessPoint

AccessPoint – объект, представляющий точку доступа (дверь, ворота, турникет) в СКУД АПК «Бастион-2». Используется в качестве выходного параметра в методе GetAccessPoints, возвращающем коллекцию точек доступа.

№ п/п	Имя	Тип	Описание
1	ServID	String	Идентификатор сервера – владельца точки прохода
2	SubDeviceNo	Int32?	Идентификатор устройства на сервере
3	SubDeviceName	String	Имя точки прохода

4.20 ControlArea

Тип ControlArea представляет собой набор данных областей контроля. Используется в качестве выходного параметра метода GetControlAreas, возвращающего коллекцию областей контроля, а также является полем типа данных Attendance.

№ п/п	Имя	Тип	Описание
1	ServID	String	Идентификатор сервера – владельца области контроля
2	SubDeviceNo	Int32?	Идентификатор области контроля на сервере
3	SubDeviceName	String	Имя области контроля.

5 Методы web-API

Web-интерфейс модуля «Бастион-2 – ИКС» содержит страницу, доступную по адресу http://*:5005/api (или https://*:5006, в зависимости от того, в каком режиме запущен модуль), которая содержит краткое описание всех методов web-API: пути, входные и выходные параметры.

Все методы web-API «Бастион-2 – ИКС» доступны по HTTP (или HTTPS) в следующем формате: **http://<сервер>:5005/api/<метод>?<параметр1>=<значение1>&<параметрN>=<значениеN>**, где <параметр1> - <параметрN> – параметры, которые передаются в строке запроса (request-параметры). Так же некоторые методы web-API (Login, PutPass, PutOrg, UpdateOrg, DeleteOrg, PutDepart, UpdateDepart и DeleteDepart) имеют один входной параметр, передаваемый в теле запроса в формате JSON (body-параметр), который представляет собой объект какого-либо типа из описанных в пункте 4 данного руководства.

Методы web-API вызываются запросами GET, POST и PUT.

Методы web-API возвращают данные в формате JSON, либо строку с результатом выполнения операции.

Все методы web-API модуля «Бастион-2 – ИКС», за исключением методов GetVersion и CheckConnection требуют авторизацию. Авторизация выполняется при помощи методов Login (войти) и LogOff (выйти). При вызове какого-либо метода web-API неавторизованным пользователем будет возвращена строка "unauthorized". Метод Login принимает в качестве входного параметра объект типа BastionOperator (пп. 4.10), который представляет собой описание оператора сервера АПК «Бастион-2». Авторизация возможна с учетными данными оператора любого из серверов АПК «Бастион-2», включенных в схему интеграции и отмеченных как сервера авторизации. Оператор должен быть активным и иметь профиль с правом использования модуля "Бастион-2 – ИКС" (Рис. 12). В случае успешной авторизации, метод Login возвращает HTTP cookie, которые должны использоваться в дальнейшем при вызове методов, требующих авторизацию.

Подробные описания, включая форматы вызовов, а также входные и выходные параметры каждого из методов приведены ниже в подпунктах данного пункта руководства.

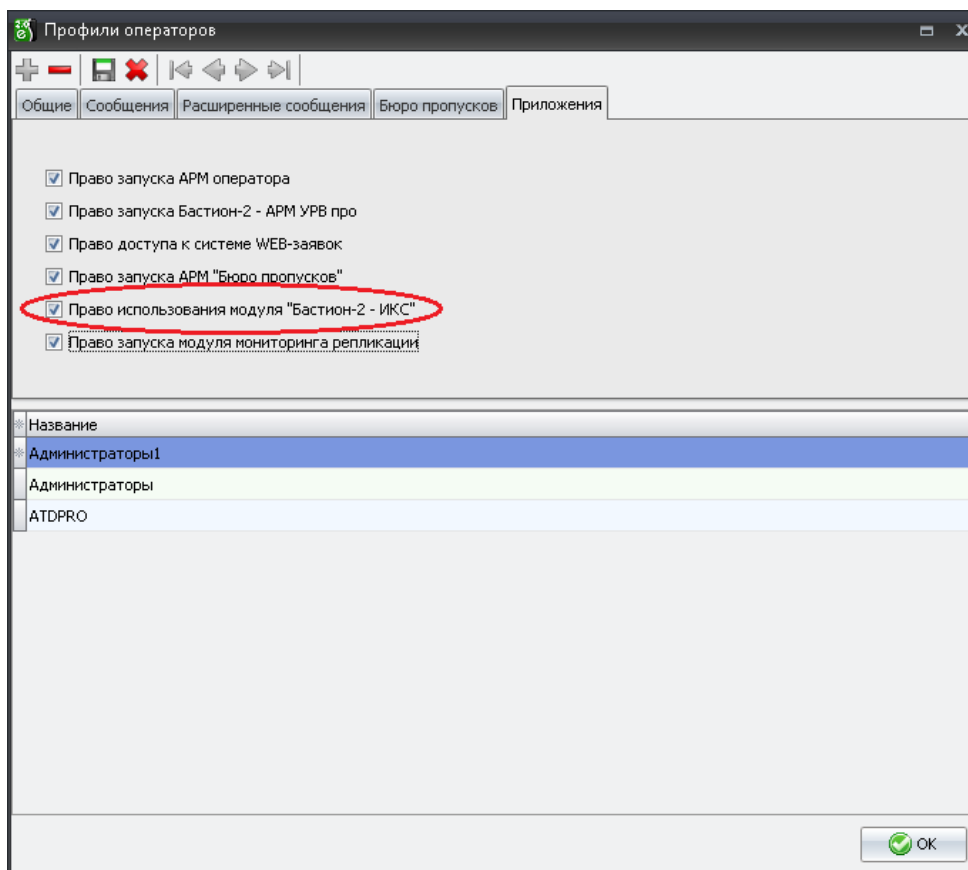


Рис. 12. Профиль оператора с правом использования модуля «Бастион-2 – ИКС»

5.1 GetVersion

Метод, возвращающий строку с версией модуля, например, "2.1".

HTTP-метод: GET, путь: /api/GetVersion.

Входные параметры: нет.

5.2 Login

Метод, предназначенный для авторизации пользователя.

HTTP-метод: POST, путь: /api/Login.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	oper	Body	BastionOperator	Учетные данные оператора сервера авторизации. Объект типа BastionOperator (пп 4.10)

В случае успешной авторизации метод возвращает строку "success", в противном случае – "fail".

Также в случае успешной авторизации метод возвращает HTTP cookie, которые в дальнейшем необходимо передавать в запросе при вызове методов web-API, которые требуют авторизацию.

5.3 LogOff

Метод для выхода пользователя.

HTTP-метод: POST, путь: /api/LogOff.

Входные параметры: нет.

5.4 CheckConnection

Метод предназначен для проверки связи со всеми серверами, добавленными в схему интеграции, либо с одним, код которого передан в качестве входного параметра. Не требует авторизацию.

HTTP-метод: GET, путь: /api/CheckConnection.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которым требуется выполнить проверку связи. Значение параметра может быть пустым, в этом случае будет выполнена проверка связи со всеми серверами схемы интеграции

Метод возвращает строку в следующем формате:

```
<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;
```

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

5.5 GetServers

Метод, предназначенный для получения списка серверов, сконфигурированных в схеме интеграции. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetServers.

Входные параметры: нет.

Метод возвращает коды серверов, добавленных в схему интеграции в виде коллекции строк.

5.6 GetEntryPoints

Метод, предназначенный для получения информации о точках прохода со всех серверов, добавленных в схему, либо с одного сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetEntryPoints.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
---	----------	-----------------	------------	----------

1	srvCode	Request	string	Код сервера, с которого требуется получить точки прохода. Значение параметра может быть пустым, в этом случае будут возвращены точки прохода со всех серверов
---	---------	---------	--------	---

Метод возвращает коллекцию объектов EntryPoint (пп. 4.1).

5.7 GetDictVals

Метод предназначен для запроса списка словарных значений со всех серверов, добавленных в схему (либо с одного сервера, код которого передан в качестве входного параметра) с фильтрацией по категории. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetDictVals.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить точки прохода. Значение параметра может быть пустым, в этом случае будут возвращены точки прохода со всех серверов
2	category	Request	Int32?	Категория словарных значений, информацию о которых требуется получить. Значение параметра может быть пустым, в этом случае будут возвращены словарные значения всех категорий

Метод возвращает коллекцию объектов DictVal (пп. 4.7).

5.8 GetCardEvents

Метод предназначен для получения списка событий, произошедших с конкретной картой доступа (код которой передан в качестве входного параметра), либо со всеми картами доступа с сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetCardEvents.

Выходные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	String	Код сервера, с которого требуется получить события, произошедшие с картой доступа
2	cardCode	Request	String	Код карты доступа, с которой произошли возвращаемые события. Значение параметра может быть пустым, в этом случае будут возвращены события со всеми картами доступа
3	dateFrom	Request	DateTime?	Минимальная дата, которую должны иметь возвращаемые события. Значение параметра может быть пустым

4	dateTo	Request	DateTime?	Максимальная дата, которую должны иметь возвращаемые события. Значение параметра может быть пустым.
5	withPhotos	Request	Bool	Флаг, определяющий, нужно ли возвращать фотографию, прикрепленную к событию (true если фотографию возвращать нужно), если имеется соответствующая фотография

Метод возвращает коллекцию объектов CardEvent (пп. 4.9).

5.9 GetAttendance

Метод предназначен для получения списка посещений с конкретной картой доступа (код которой передан в качестве входного параметра), либо со всеми картами доступа с сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetAttendance.

Выходные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить посещения, выполненные с картой доступа
2	cardCode	Request	string	Код карты доступа, с которой произошли посещения. Значение параметра может быть пустым, в этом случае будут возвращены посещения со всеми картами доступа
3	dateFrom	Request	DateTime?	Минимальная дата, которую должны иметь возвращаемые посещения. Значение параметра может быть пустым
4	dateTo	Request	DateTime?	Максимальная дата, которую должны иметь возвращаемые посещения. Значение параметра может быть пустым

Метод возвращает коллекцию объектов типа Attendance (пп. 4.8).

5.10 GetPasses

Метод предназначен для получения списка пропусков с фильтрацией по их статусу и типу с сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetPasses.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить список пропусков

2	cardStatus	Request	Int32?	Статус возвращаемых пропусков. Значение параметра может быть пустым, в этом случае будут возвращены пропуска с любым статусом
3	passType	Request	Int32?	Тип возвращаемых пропусков. Значение параметра может быть пустым, в этом случае будут возвращены пропуска всех типов (Постоянный - 1, Временный - 2, Разовый - 4)
4	withoutPhoto	Request	Bool?	Флаг, определяющий, нужно ли возвращать фотографии владельцев пропусков (true если фотографии возвращать не нужно). Значение параметра может быть пустым, в этом случае будут возвращены пропуска с фотографиями
5	startFrom	Request	Int32?	Порядковый номер, начиная с которого будут возвращены пропуска (постраничный вывод). Значение параметра может быть пустым, в этом случае будут возвращаться пропуска начиная с первого
6	maxCount	Request	Int32?	Максимальное количество пропусков, которое будет возвращено методом (постраничный вывод). Значение параметра может быть пустым, в этом случае количество возвращаемых пропусков ограничиваться не будет
7	startDateFrom	Request	DateTime?	Дата начала действия пропуска, от (включительно). Значение параметра может быть пустым
8	startDateTo	Request	DateTime?	Дата начала действия пропуска, до. Значение параметра может быть пустым

Метод возвращает коллекцию объектов типа Pass (п. 4.4).

Метод предоставляет возможность одним запросом получить пропуска только с одного конкретного сервера и требует обязательного указания параметра **srvCode**.

5.11 GetPassesByChangeDate

Метод предназначен для получения списка пропусков по дате их последнего изменения с сервера, код которого передан в качестве входного параметра. Дата изменения высчитывается «каскадно», то есть датой изменения пропуска считается самая поздняя дата изменения среди дат изменения самого пропуска, персоны, а также дат изменения связанных с ними словарных значений. Метод требует авторизацию.

HTTP-метод: GET, путь: /api/GetPasses.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить список пропусков
2	cardStatus	Request	Int32?	Статус возвращаемых пропусков. Значение параметра может быть пустым, в этом случае будут возвращены пропуска с любым статусом

3	passType	Request	In32?	Тип возвращаемых пропусков. Значение параметра может быть пустым, в этом случае будут возвращены пропуска всех типов (Постоянный - 1, Временный - 2, Разовый - 4)
4	withoutPhoto	Request	Bool?	Флаг, определяющий, нужно ли возвращать фотографии владельцев пропусков (true если фотографии возвращать не нужно). Значение параметра может быть пустым, в этом случае будут возвращены пропуска с фотографиями
5	changeDateFrom	Request	DateTime?	Дата последнего изменения, нижняя граница. Будут возвращены все пропуска, которые были изменены в эти или более поздние даты и время. Значение параметра может быть пустым, в этом случае будут возвращены пропуска без ограничения даты изменения по нижней границе.
6	changeDateTo	Request	DateTime?	Дата последнего изменения, верхняя граница. Будут возвращены все пропуска, которые были изменены не позднее переданных в данном параметре даты и времени. Значение параметра может быть пустым, в этом случае будут возвращены пропуска без ограничения даты изменения по верхней границе.

Метод возвращает коллекцию объектов типа Pass (пп. 4.4).

Метод предоставляет возможность одним запросом получить пропуска только с одного конкретного сервера и требует обязательного указания параметра **srvCode**.

5.12 GetPersons

Метод предназначен для получения списка персон (отдельно от пропусков) с сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetPersons.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить список персон
2	withoutPhoto	Request	Bool?	Флаг, определяющий, нужно ли возвращать фотографии персон (true если фотографии возвращать не нужно). Значение параметра может быть пустым, в этом случае будут возвращены персоны с фотографиями
3	startFrom	Request	Int32?	Порядковый номер, начиная с которого будут возвращены персоны (постраничный вывод). Значение параметра может быть пустым, в этом случае будут возвращаться персоны начиная с первой

4	maxCount	Request	Int32?	Максимальное количество персон, которое будет возвращено методом (постраничный вывод). Значение параметра может быть пустым, в этом случае количество возвращаемых персон ограничиваться не будет
5	createDateFrom	Request	DateTime?	Дата создания персоны, от (включительно). Значение параметра может быть пустым
6	createDateTo	Request	DateTime?	Дата создания персоны, до. Значение параметра может быть пустым

Метод возвращает коллекцию объектов типа Person (пп. 4.2).

Метод предоставляет возможность одним запросом получить список персон только с одного конкретного сервера и требует обязательного указания параметра **srvCode**.

5.13 GetPersonsByChangeDate

Метод предназначен для получения списка персон (отдельно от пропусков) по дате их последнего изменения с сервера, код которого передан в качестве входного параметра. Дата изменения высчитывается «каскадно», то есть датой изменения персоны считается самая поздняя дата изменения самой персоны, либо связанных с ней словарных значений. Метод требует авторизацию.

HTTP-метод: GET, путь: /api/GetPersons.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить список персон
2	withoutPhoto	Request	Bool?	Флаг, определяющий, нужно ли возвращать фотографии персон (true если фотографии возвращать не нужно). Значение параметра может быть пустым, в этом случае будут возвращены персоны с фотографиями
	changeDateFrom	Request	DateTime?	Дата последнего изменения, нижняя граница. Будут возвращены все персоны, которые были изменены в эти или более поздние даты и время. Значение параметра может быть пустым, в этом случае будут возвращены персоны без ограничения даты изменения по нижней границе.
4	changeDateTo	Request	DateTime?	Дата последнего изменения, верхняя граница. Будут возвращены все персоны, которые были изменены не позднее переданных в данном параметре даты и времени. Значение параметра может быть пустым, в этом случае будут возвращены персоны без ограничения даты изменения по верхней границе.

Метод возвращает коллекцию объектов типа Person (пп. 4.2).

Метод предоставляет возможность одним запросом получить список персон только с одного конкретного сервера и требует обязательного указания параметра **srvCode**.

5.14 GetPerson

Метод предназначен для поиска персоны по её идентификатору, переданному в качестве входного параметра, (отдельно от пропусков) с сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetPersons.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить список персон
2	personId	Request	Int32	Идентификатор персоны
3	withoutPhoto	Request	Bool?	Флаг, определяющий, нужно ли возвращать фотографии персон (true если фотографии возвращать не нужно). Значение параметра может быть пустым, в этом случае будут возвращены персоны с фотографиями

Метод возвращает коллекцию объектов типа Person (пп. 4.2).

Метод предоставляет возможность одним запросом получить список персон только с одного конкретного сервера и требует обязательного указания параметра **srvCode**.

5.15 GetPassCount

Метод предназначен для получения общего количества пропусков с фильтрацией по их статусу и типу с сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetPassCount.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить список пропусков
2	cardStatus	Request	Int32?	Статус подсчитываемых пропусков. Значение параметра может быть пустым, в этом случае будут подсчитаны пропуска, имеющие любой статус
3	passType	Request	Int32?	Тип подсчитываемых пропусков. Значение параметра может быть пустым, в этом случае будут подсчитаны пропуска всех типов (Постоянный - 1, Временный - 2, Разовый - 4)

Метод возвращает количество пропусков (Int64).

5.16 GetPassesByPerson

Метод предназначен для получения списка пропусков любых статусов (в том числе заявок на пропуск), которые выдавались человеку, данные которого переданы во входных параметрах. Метод позволяет одновременно запросить пропуска только с одного сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetPassesByPerson.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	String	Код сервера, с которого требуется получить список пропусков
2	name	Request	String	Фамилия человека, на которого оформлялся пропуск
3	firstname	Request	String	Имя
4	secondname	Request	String	Отчество
5	birthDate	Request	DateTime?	Дата рождения. Значение параметра может быть пустым (в этом случае будет выполняться поиск персоны с незаполненной датой рождения)
6	tableno	Request	String	Табельный номер. Значение параметра может быть пустым, в этом случае значение табельного номера не будет учитываться при поиске персоны
7	withoutPhoto	Request	Bool?	Флаг, определяющий, нужно ли возвращать фотографии владельцев пропусков (true если фотографии возвращать не нужно). Значение параметра может быть пустым, в этом случае будут возвращены пропуска с фотографиями

Метод возвращает коллекцию объектов типа Pass (пп. 4.4).

Метод предоставляет возможность одним запросом получить пропуска только с одного конкретного сервера и требует обязательного указания параметра **srvCode**.

5.17 GetPassesByCardCode

Метод предназначен для получения списка пропусков любых статусов, по которым когда-либо выдавалась карта, код которой передан во входных параметрах. Метод позволяет одновременно запросить пропуска только с одного сервера, код которого передан во входных параметрах. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetPassesByCardCode.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
-------	----------	-----------------	------------	----------

1	srvCode	Request	String	Код сервера, с которого требуется получить список пропусков
2	cardCode	Request	String	Код карты
3	withoutPhoto	Request	Bool?	Флаг, определяющий, нужно ли возвращать фотографии владельцев пропусков (true если фотографии возвращать не нужно). Значение параметра может быть пустым, в этом случае будут возвращены пропуска с фотографиями

Метод возвращает коллекцию объектов типа Pass (пп. 4.4).

Метод предоставляет возможность одним запросом получить пропуска только с одного конкретного сервера и требует обязательного указания параметра **srvCode**.

5.18 GetPassesByPersonId

Метод предназначен для получения списка пропусков персоны, идентификатор которой передан в параметре. Список пропусков будет составлен из любых статусов, по которым когда-либо выдавалась карта. Метод позволяет одновременно запросить пропуска только с одного сервера, код которого передан во входных параметрах. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetPassesByPersonId.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	String	Код сервера, с которого требуется получить список пропусков
2	personId	Request	Int32	Идентификатор персоны
3	withoutPhoto	Request	Bool?	Флаг, определяющий, нужно ли возвращать фотографии владельцев пропусков (true если фотографии возвращать не нужно). Значение параметра может быть пустым, в этом случае будут возвращены пропуска с фотографиями

Метод возвращает коллекцию объектов типа Pass (пп. 4.4).

Метод предоставляет возможность одним запросом получить пропуска только с одного конкретного сервера и требует обязательного указания параметра **srvCode**.

5.19 PutPass

Метод предназначен для создания или редактирования КД, а также для создания/редактирования заявки на пропуск на всех серверах, добавленных в схему интеграции, либо на одном, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: PUT, путь: /api/PutPass.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	String	Код сервера, на котором требуется выполнить создание/обновление пропуска/заявки на пропуск. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	pass	Body	Pass	Набор данных пропуска/заявки на пропуск. Объект типа Pass (пп. 4.4)
3	useAccessLevelsInsteadOfEntryPoints	Request	Boolean	Флаг, при выставлении которого в значение true при создании пропуска учитываются данные поля AccessLevels. По умолчанию значение флага – false, в этом случае используются данные поля EntryPoints.

Метод возвращает строку в следующем формате:

```
<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;
```

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

Внимание! Особенностью АПК «Бастион-2» является многоуровневая структура организаций и подразделений. Каждая организация может иметь дочерние организации и подразделения, каждое подразделение может иметь дочерние подразделения. Это значит, что при передаче в метод *PutPass* персональных данных пропуска нужно указывать полные наименования организации и подразделения, которые представляют собой полный путь к организации или подразделению, включая имена всех родительских организаций, разделённые символом " \ " (отделённым от названий пробелами с обеих сторон).

Например, если требуется создать пропуск на человека, работающего в подразделении "Подразделение 2", которое является дочерним к подразделению "Подразделение 1", которое, в свою очередь, принадлежит организации "Организация 2", подчинённой организации "Организация 1", то значения полей **Org** и **Dep** поля **PersonData** объекта **Pass** должны быть следующими: "Организация 1\Организация 2" и "Подразделение 1\Подразделение 2". Если требуется создать пропуск на человека, например, в "Организацию 2", то поле **Org** должно содержать значение "Организация 1\Организация 2", а поле **Dep** необходимо оставить пустым. Назначение дополнительных полей настраивается штатными средствами АПК «Бастион-2».

Алгоритм применения данных на сервере при вызове метода **PutPass** при условии передачи непустого кода карты (поле **CardCode** объекта **Pass**) приведен на рисунке 12. В случае передачи пустого значения в качестве кода карты выполняется создание новой заявки на пропуск без выдачи КД на новую или существующую персону (при условии передачи непустого списка точек прохода (поле **EntryPoints** объекта **Pass**) или списка уровней доступа (поле **AccessLevels** объекта **Pass**) при включенном флаге **useAccessLevelsInsteadOfEntryPoints**). Выдать карту доступа по заявке возможно при помощи метода **IssuePass**.

В поле **EntryPoints** объекта **Pass** поместить необходимые точки прохода (из списка полученных методом **GetEntryPoints**) всех серверов, на которых требуется создать пропуск или заявку на

пропуск, или, в случае передачи значения true в параметре **useAccessLevelsInsteadOfEntryPoints**, необходимо поместить в поле **AccessLevels** объекта **Pass** по одному уровню доступа (из списка полученных методом **GetAccessLevels**) на каждый сервер, на котором требуется создать пропуск или заявку на пропуск.

Если выполняется создание пропуска на одном конкретном сервере (указан код сервера в параметре **srvCode**), то необходимо заполнить поле **EntryPoints** точками прохода только одного сервера, код которого указан в параметре **srvCode**, либо заполнить поле **AccessLevels** одним уровнем доступа с сервера, код которого указан в параметре **srvCode**, если передается значение true в параметре **useAccessLevelsInsteadOfEntryPoints**.

Если в объекте **PersonData** указано поле **PersonId**, то поиск персоны будет производиться по указанному идентификатору персоны. Если **PersonId** не указан, то поиск производится по ФИО персоны.

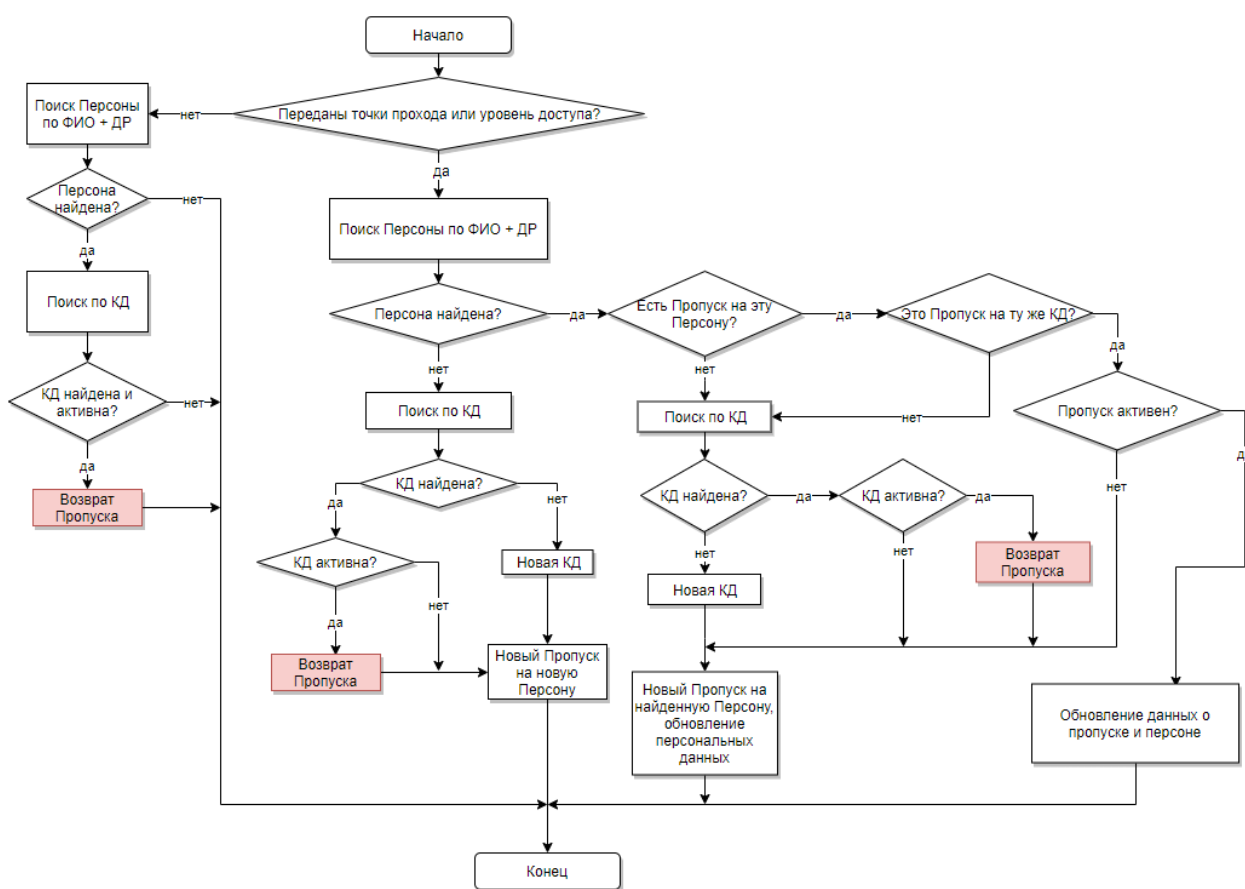


Рис. 13. Алгоритм обработки данных в методе PutPass

5.20 BlockPass

Метод предназначен для блокировки КД на всех серверах, добавленных в схему интеграции, либо на одном, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/BlockPass.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется заблокировать пропуск. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	cardCode	Request	string	Код КД, которую требуется заблокировать
3	blockReason	Request	string	Причина блокировки

Метод возвращает строку в следующем формате:

```
<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;
```

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

5.21 UnblockPass

Метод предназначен для разблокировки КД на всех серверах, добавленных в схему интеграции, либо на одном, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/UnblockPass.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется разблокировать пропуск. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	cardCode	Request	string	Код КД, которую требуется разблокировать

Метод возвращает строку в следующем формате:

```
<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;
```

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

5.22 ReturnPass

Метод предназначен для возврата (переноса в архив) карты доступа на всех серверах, добавленных в схему интеграции, либо на одном, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/ReturnPass.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется вернуть пропуск. Значение параметра может быть

				пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	cardCode	Request	string	Код КД, которую требуется вернуть

Метод возвращает строку в следующем формате:

<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

5.23 IssuePass

Метод предназначен для выдачи КД по ранее созданной заявке на пропуск на всех серверах, добавленных в схему интеграции, либо на одном, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/IssuePass.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется выдать пропуск. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	name	Request	string	Фамилия человека, на которого оформлена заявка на пропуск
3	firstname	Request	string	Имя человека, на которого оформлена заявка на пропуск
4	secondname	Request	string	Отчество человека, на которого оформлена заявка на пропуск
5	birthDate	Request	DateTime?	Дата рождения человека, на которого оформлена заявка на пропуск
6	passType	Request	Int32?	Тип пропуска (Постоянный - 1, Временный - 2, Разовый - 4)
7	cardCode	Request	string	Код КД, который будет иметь выданный пропуск
8	personId	Request	Int32?	Идентификатор персоны, на которого оформлена заявка на пропуск

Метод возвращает строку в следующем формате:

<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

5.24 GetOrgs

Метод предназначен для получения списка организаций со всех серверов, добавленных в схему, с удалением повторяющихся значений, либо списка организаций с одного сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetOrgs.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить информацию об организациях. Значение параметра может быть пустым, в этом случае организации будут возвращены со всех серверов, добавленных в схему интеграции (с удалением повторяющихся)
2	filter	Request	string	Строка, которую должны содержать в своем названии возвращаемые организации. Значение параметра может быть пустым, в этом случае будут возвращены все организации без фильтрации

Метод возвращает коллекцию объектов типа Org (пп. 4.5).

5.25 PutOrg

Метод предназначен для добавления организации на всех серверах, добавленных в схему, либо на одном сервере, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: PUT, путь: /api/PutOrg.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется добавить организацию. Значение параметра может быть пустым, в этом случае организация будет добавлена на всех серверах, добавленных в схему интеграции
2	org	Body	Org	Организация, которую требуется добавить. Объект типа Org (пп. 4.5)

Метод возвращает строку в следующем формате:

`<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;`

где `<srvN>` – код сервера, а `<resultN>` – результат выполнения операции.

5.26 UpdateOrg

Метод предназначен для переименования организации на всех серверах, добавленных в схему интеграции, либо на одном сервере, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: POST, путь: /api/UpdateOrg.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется переименовать организацию. Значение параметра может быть пустым, в этом случае организация будет переименована на всех серверах, добавленных в схему интеграции
2	orgNewName	Request	string	Новое имя организации
3	org	Body	Org	Организация, которую требуется переименовать. Объект типа Org (пп. 4.5)

Метод возвращает строку в следующем формате:

```
<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;
```

где <srvN> – код сервера, а <resultN> – результат выполнения операции.

5.27 DeleteOrg

Метод предназначен для удаления организации (вместе со всеми дочерними организациями и подразделениями) на всех серверах, добавленных в схему интеграции, либо на одном сервере, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: POST, путь: /api/DeleteOrg.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется удалить организацию. Значение параметра может быть пустым, в этом случае организация будет удалена на всех серверах, добавленных в схему интеграции
2	org	Body	Org	Организация, которую требуется удалить. Объект типа Org (пп. 4.5)

Метод возвращает строку в следующем формате:

```
<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;
```

где <srvN> – код сервера, а <resultN> – результат выполнения операции.

5.28 GetDeparts

Метод предназначен для получения списка подразделений (всех, либо принадлежащих организации, имя которой передано в качестве входного параметра) со всех серверов, добавленных в схему, либо списка организаций (всех, либо принадлежащих организации, имя которой передано в качестве входного параметра) с одного сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetDeparts.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить информацию о подразделениях. Значение параметра может быть пустым, в этом случае подразделения будут возвращены со всех серверов, добавленных в схему интеграции (с удалением повторяющихся)
2	parentOrgName	Request	string	Полное имя организации (включая путь и все дочерние организации, разделенные символом " \ " (отделённым от названий пробелами с обеих сторон), которая должна быть родительской для возвращаемых подразделений. Значение параметра может быть пустым, в этом случае будут возвращены подразделения всех организаций

Метод возвращает коллекцию объектов типа Depart (пп. 4.6).

5.29 PutDepart

Метод предназначен для добавления подразделения на всех серверах, добавленных в схему интеграции, либо на одном сервере, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: PUT, путь: /api/PutDepart.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется добавить подразделение. Значение параметра может быть пустым, в этом случае подразделение будет добавлено на всех серверах, включенных в схему интеграции
2	depart	Body	Depart	Подразделение, которое требуется добавить. Объект типа Depart (пп. 4.6)

Метод возвращает строку в следующем формате:

```
<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;
```

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

5.30 UpdateDepart

Метод предназначен для переименования подразделения на всех серверах, добавленных в схему интеграции, либо на одном сервере, код которого передан в качестве входного параметра. Требует авторизацию.

HTTP-метод: POST, путь: /api/UpdateDepart.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется переименовать подразделение. Значение параметра может быть пустым, в этом случае подразделение будет переименовано на всех серверах, добавленных в схему интеграции
2	departNewName	Request	string	Новое имя подразделения
3	depart	Body	Depart	Подразделение, которое требуется переименовать. Объект типа Depart (пп. 4.6)

Метод возвращает строку в следующем формате:

<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

5.31 DeleteDepart

Метод предназначен для удаления подразделения (вместе со всеми дочерними) на всех серверах, добавленных в схему интеграции, либо на одном сервере, код которого передан в качестве входного параметра. Требует авторизацию.

HTTP-метод: POST, путь: /api/DeleteDepart.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется удалить подразделение. Значение параметра может быть пустым, в этом случае подразделение будет удалено на всех серверах, добавленных в схему интеграции
2	depart	Body	Depart	Подразделение, которое требуется удалить. Объект типа Depart (пп. 4.6)

Метод возвращает строку в следующем формате:

<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

5.32 PutMVPass

Метод предназначен для создания или редактирования материального пропуска, а также для создания/редактирования заявки на материальный пропуск на всех серверах, добавленных в схему интеграции, либо на одном, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: PUT, путь: /api/PutMVPass.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется выполнить создание/обновление материального пропуска/заявки на пропуск. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	issuePass	Request	Boolean	Флаг, определяющий, будет создан выданный материальный пропуск, либо же будет создана заявка на пропуск
3	matValuePass	Body	MatValuePass	Набор данных материального пропуска/заявки на пропуск. Объект типа MatValuePass (пп. 4.15)

Метод возвращает строку в следующем формате:

<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

Внимание! Особенностью АПК «Бастион-2» является многоуровневая структура организаций и подразделений. Каждая организация может иметь дочерние организации и подразделения, каждое подразделение может иметь дочерние подразделения. Это значит, что при передаче в метод PutMVPass данных пропуска нужно указывать полные наименования организации и подразделения, которые представляют собой полный путь к организации или подразделению, включая имена всех родительских организаций, разделённых символом " \ " (отделённым от названий пробелами с обеих сторон).

Например, если требуется указать в качестве подразделения назначения "Подразделение 2", которое является дочерним к подразделению "Подразделение 1", которое, в свою очередь, принадлежит организации "Организация 2", подчинённой организации "Организация 1", то значения полей **GoalOrganization** и **GoalDepartment** объекта **MatValuePass** должны быть следующими: "Организация 1\Организация 2" и "Подразделение 1\Подразделение 2". Если требуется указать в качестве подразделения назначения "Организацию 2", то поле **GoalOrganization**

должно содержать значение "Организация 1\Организация 2", а поле **GoalDepartment** необходимо оставить пустым.

При передаче идентификатора пропуска, равного null или 0, будет выполняться создание нового материального пропуска. При передаче ненулевого идентификатора пропуска будет выполняться редактирование существующего материального пропуска.

В случае передачи в параметре **IssuePass** значения False методом будет создана заявка на материальный пропуск (статус -1). При значении параметра **IssuePass** True метод создаст выданный материальный пропуск (статус 1).

5.33 GetMVPasses

Метод предназначен для получения списка материальных пропусков с фильтрацией по их статусу с сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetMVPasses.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить список материальных пропусков
2	status	Request	Int32?	Статус возвращаемых материальных пропусков. Значение параметра может быть пустым, в этом случае будут возвращены пропуска с любым статусом

Метод возвращает коллекцию объектов типа MatValuePass (пп. 4.15).

5.34 GetMVPassesByPersonPass

Метод предназначен для получения списка материальных пропусков с фильтрацией по персональному пропуску с сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: POST, путь: /api/GetMVPassesByPersonPass.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить список материальных пропусков
2	pass	Body	PassBrief	Персональный пропуск, к которому прикреплены возвращаемые материальные пропуска

Метод возвращает коллекцию объектов типа MatValuePass (пп. 4.15).

5.35 IssueMVPass

Метод предназначен для выдачи материального пропуска по ранее созданной заявке на сервере, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/IssueMVPass.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	String	Код сервера, на котором требуется выполнить выдачу материального пропуска. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	passid	Request	Int32	Идентификатор пропуска

Метод возвращает строку в следующем формате:

<srv>:<result>;

где **<srv>** – код сервера, а **<result>** – результат выполнения операции.

5.36 WithdrawMVPass

Метод предназначен для изъятия материального пропуска на сервере, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/WithdrawMVPass.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	String	Код сервера, на котором требуется выполнить изъятие материального пропуска. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	passid	Request	Int32	Идентификатор пропуска

Метод возвращает строку в следующем формате:

<srv>:<result>;

где **<srv>** – код сервера, а **<result>** – результат выполнения операции.

5.37 PutCarPass

Метод предназначен для создания или редактирования транспортного пропуска, а также для создания/редактирования заявки на транспортный пропуск на всех серверах, добавленных в схему интеграции, либо на одном, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: PUT, путь: /api/PutCarPass.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, на котором требуется выполнить создание/обновление транспортного пропуска/заявки на пропуск. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	issuePass	Request	Boolean	Флаг, определяющий, будет создан выданный транспортный пропуск, либо же будет создана заявка на пропуск
3	carPass	Body	CarPass	Набор данных транспортного пропуска/заявки на пропуск. Объект типа CarPass (пп. 4.16)

Метод возвращает строку в следующем формате:

```
<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;
```

где **<srvN>** – код сервера, а **<resultN>** – результат выполнения операции.

Массив **Cars** в объекте **carPass** должен содержать не более 3 элементов. Первый элемент массива – основное транспортное средство, второй – полуприцеп, третий – прицеп. Если требуется добавить только основное ТС, или основное ТС с прицепом, то массив может содержать только 1 или только 2 элемента. Если требуется добавить только прицеп или только полуприцеп, то первые два или один элемента должны быть пустыми (null), чтобы сохранилась идентификация типа транспортного средства по порядковому номеру элемента в наборе.

При передаче идентификатора пропуска, равного null или 0, будет выполняться создание нового материального пропуска. При передаче ненулевого идентификатора пропуска будет выполняться редактирование существующего материального пропуска.

В случае передачи в параметре **IssuePass** значения False методом будет создана заявка на транспортный пропуск (статус -1). При значении параметра **IssuePass** True метод создаст выданный транспортный пропуск (статус 1).

5.38 GetCarPasses

Метод предназначен для получения списка транспортных пропусков с фильтрацией по их статусу с сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetCarPasses.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить список транспортных пропусков

2	status	Request	Int32?	Статус возвращаемых транспортных пропусков. Значение параметра может быть пустым, в этом случае будут возвращены пропуска с любым статусом
---	--------	---------	--------	--

Метод возвращает коллекцию объектов типа CarPass (пп. 4.16).

5.39 GetCarPassesByPersonPass

Метод предназначен для получения списка транспортных пропусков с фильтрацией по персональному пропуску с сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: POST, путь: /api/GetCarPassesByPersonPass.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	String	Код сервера, с которого требуется получить список транспортных пропусков
2	pass	Body	PassBrief	Персональный пропуск, к которому прикреплены возвращаемые транспортные пропуска

Метод возвращает коллекцию объектов типа CarPass (пп. 4.16).

5.40 IssueCarPass

Метод предназначен для выдачи транспортного пропуска по ранее созданной заявке на сервере, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/IssueCarPass.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	String	Код сервера, на котором требуется выполнить выдачу транспортного пропуска. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	passid	Request	Int32	Идентификатор пропуска

Метод возвращает строку в следующем формате:

<srv>:<result>;

где <srv> – код сервера, а <result> – результат выполнения операции.

5.41 WithdrawCarPass

Метод предназначен для изъятия транспортного пропуска на сервере, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/WithdrawCarPass.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	s String	Код сервера, на котором требуется выполнить изъятие транспортного пропуска. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	passid	Request	Int32	Идентификатор пропуска

Метод возвращает строку в следующем формате:

<srv>:<result>;

где <srv> – код сервера, а <result> – результат выполнения операции.

5.42 GetDevices

Метод предназначен для получения набора устройств, добавленных в систему на сервере, код которого передан в качестве входного параметра. Требует авторизацию. Метод возвращает дерево устройств, верхний уровень которого представлен экземплярами драйверов, добавленных в систему. Дочерние устройства каждого драйвера представлены в поле **Childs** каждого объекта типа **Device**.

В случае передачи не пустого значения параметра **deviceType** метод возвращает линейный массив устройств указанного типа. Метод требует авторизацию.

HTTP-метод: GET, путь: /api/GetDevices.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить уровни доступа. Значение параметра может быть пустым, в этом случае будут возвращены точки прохода со всех серверов
2	driverId	Request	Int?	Код типа драйвера, устройства которого необходимо получить. Значение параметра может быть пустым, в таком случае будут возвращены устройства всех драйверов.
3	deviceType	Request	Int?	Код типа устройства. В случае передачи непустого значения параметра будут возвращены устройства только данного типа. Значение параметра может быть пустым.

Метод возвращает коллекцию объектов типа Device (пп. 4.17).

5.43 GetAccessLevels

Метод, предназначенный для получения информации об уровнях доступа со всех серверов, добавленных в схему, либо с одного сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetAccessLevels.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить уровни доступа. Значение параметра может быть пустым, в этом случае будут возвращены точки прохода со всех серверов

Метод возвращает коллекцию объектов типа AccessLevel (пп. 4.18).

5.44 GetAccessPoints

Метод, предназначенный для получения информации об точках доступа со всех серверов, добавленных в схему, либо с одного сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetAccessPoints.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить точки доступа. Значение параметра может быть пустым, в этом случае будут возвращены точки прохода со всех серверов

Метод возвращает коллекцию объектов типа AccessPoint (пп. 4.19).

5.45 GetControlAreas

Метод, предназначенный для получения информации об областях контроля со всех серверов, добавленных в схему, либо с одного сервера, код которого передан в качестве входного параметра. Требуется авторизация.

HTTP-метод: GET, путь: /api/GetControlAreas.

Входные параметры:

№ п/п	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, с которого требуется получить области контроля. Значение параметра может быть пустым, в

				этом случае будут возвращены области контроля со всех серверов
--	--	--	--	--

Метод возвращает коллекцию объектов типа ControlArea (пп. 4.20).

5.46 UpdatePersonData

Метод предназначен для обновления данных персоны на всех серверах, добавленных в схему интеграции, либо на одном, код которого передан в качестве входного параметра. Требует авторизацию.

HTTP-метод: PUT, путь: /api/UpdatePersonData.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	String	Код сервера, на котором требуется выполнить обновление данных персоны. Значение параметра может быть пустым, в этом случае операция будет выполнена на всех серверах, сконфигурированных в схеме интеграции
2	name	Request	String	Фамилия человека, данные которого требуется изменить
3	firstname	Request	String	Имя
4	secondname	Request	String	Отчество
5	birthDate	Request	DateTime?	Дата рождения. Значение параметра может быть пустым (в этом случае будет выполняться поиск персоны с незаполненной датой рождения)
6	tableno	Request	String	Табельный номер. Значение параметра может быть пустым, в этом случае значение табельного номера не будет учитываться при поиске персоны
7	newPersonData	Body	Person	Набор новых данных персоны. Объект типа Person (пп. 4.2)

Метод возвращает строку в следующем формате:

`<srv1>:<result1>;<srv2>:<result2>;...;<srvN>:<resultN>;`

где `<srvN>` – код сервера, а `<resultN>` – результат выполнения операции.

Внимание! Особенностью АПК «Бастион-2» является многоуровневая структура организаций и подразделений. Каждая организация может иметь дочерние организации и подразделения, каждое подразделение может иметь дочерние подразделения. Это значит, что при передаче в метод PutPass персональных данных пропуска нужно указывать полные наименования организации и подразделения, которые представляют собой полный путь к организации или подразделению, включая имена всех родительских организаций, разделённые символом " \ " (отделённым от названий пробелами с обеих сторон).

Например, если требуется поместить человека в "Подразделение 2", которое является дочерним к подразделению "Подразделение 1", которое, в свою очередь, принадлежит организации "Организация 2", подчинённой организации "Организация 1", то значения полей **Org** и **Dep** поля **PersonData** должны быть следующими: "Организация 1\Организация 2" и "Подразделение 1\Подразделение 2". Если требуется создать пропуск на человека, например, в "Организацию 2", то поле **Org** должно содержать значение "Организация 1\Организация 2", а поле **Dep** необходимо оставить пустым.

Назначение дополнительных полей настраивается штатными средствами АПК «Бастион-2».

5.47 GetQrPass

Метод предназначен для получения QR-пропуска в соответствии с настройками сервера, код которого передан в качестве входного параметра. Требуется авторизацию.

HTTP-метод: GET, путь: /api/GetQrPass.

Входные параметры:

№	Название	Способ передачи	Тип данных	Описание
1	srvCode	Request	string	Код сервера, обязательно должен быть указан
2	cardCode	Request	string	Код КД, для которой требуется сгенерировать QR-пропуск

Метод возвращает строку, содержащую закодированное в base64 изображение, содержащее QR-код с пропуском в формате ELSM или DECD (в зависимости от настроек сервера), либо строку вида «Ошибка: <описание ошибки>», если QR-пропуск сгенерировать не удалось по той или иной причине.

6 Нештатные ситуации

6.1 Проблемы с доступом на страницу конфигурации серверов

По причине того, что страница конфигурации web-интерфейса требует авторизацию через сервер авторизации, возможна ситуация, при которой доступ к странице конфигурирования списка серверов-участников схемы интеграции ограничен по причине недоступности серверов авторизации.

В случае, когда авторизация через заданный/е сервер(а) более невозможна, необходимо отредактировать файл конфигурации **servers.xml** (располагающийся в директории %ProgramFiles%\ES-Prom\Bastion-2 – IKS\wwwroot) вручную: убрать из него все пометки серверов авторизации, заменив для всех серверов значение тега **IsAuthenticationServer** с true на false, либо удалить файл **servers.xml** целиком (в этом случае файл будет создан заново автоматически при создании новой конфигурации). Если в схеме не определен ни один сервер авторизации, то к странице конфигурации возможен анонимный доступ (до тех пор, пока не будет добавлен сервер авторизации в схему).

7 Приложения

7.1 Приложение 1 – Типы драйверов

Тип драйвера	Название
256	Бастион-2 - Elsys
257	Бастион-2 - Стрелец
258	Бастион-2 - Заря
259	Бастион-2 - Intrepid II
260	Бастион-2 - Рубеж
261	Бастион-2 - Виброн
262	Бастион-2 - Intrepid
263	Бастион-2 - Schrack
264	Бастион-2 - Маршрут
265	Бастион-2 - Пунктир-С
267	Бастион-2 - KeyGuard
268	Бастион-2 – Рубеж Прот. R3
270	Бастион-2 – ЛКД КО-15
271	Бастион-2 – ЛКД
272	Бастион-2 – EnterFace
273	Бастион-2 – Elsys-PVR
274	Бастион-2 – Suprema
275	Бастион-2 – ZK Visible Light
277	Бастион-2 – GST
256	Бастион-2 – Elsys v2
268	Бастион-2 – Интеллект
370	Бастион-2 – Domination
372	Бастион-2 – Avigilon
376	Бастион-2 – SecurOS
723833	Бастион-2 – SecurOS FaceX
889222	Бастион-2 – Face
987111	Бастион-2 – Визирь
987112	Бастион-2 – Elsys Mobile
987113	Бастион-2 – Водитель
1154379	Бастион-2 - SNMP агент
1223432	Бастион-2 - OPC сервер
3473408	Бастион-2 - ПЦН клиент
18546688	Бастион-2 - ПЦН сервер
385875968	Бастион-2 - Esmi FX Net
402653184	Бастион-2 - C2000
1392508928	Бастион-2 – Peridect
1879048192	Бастион-2 – Видео
1962934272	Бастион-2 – Мурена
1979711488	Бастион-2 – СТ
2046820352	Бастион-2 - Forteza
2063597568	Бастион-2 - Гроза