



ИНТЕГРИРОВАННАЯ СИСТЕМА БЕЗОПАСНОСТИ

**БАСТИОН. WEB-СЕРВЕР ИНТЕГРАЦИИ. РУКОВОДСТВО
СИСТЕМНОГО ПРОГРАММИСТА**



Оглавление

Оглавление	1
1 Наименование и область применения.....	2
2 Назначение разработки	2
3 Технические требования.....	2
3.1 Функциональные характеристики.....	2
3.2 Спецификация методов веб-сервиса	5
3.2.1 Получение информации о подключении к веб-сервису – GetServerVersion.....	5
3.2.2 Получение результата выполнения операции - GetResult.....	5
3.2.3 Получение данных о точках доступа (прохода) СКУД - GetEntryPoints.....	6
3.2.4 Добавление/редактирования пропуска в СКУД – MsgPutPass	6
3.2.5 Блокировка/разблокировка пропуска MsgBlockPass	8
3.2.6 Возврат пропуска MsgReturnPass	8
3.2.7 Удаление пропуска MsDeletePass	9
3.2.8 Получение списка организаций.....	9
3.2.9 Получение список организаций по фильтру - GetOrgs(filter).....	10
3.2.10 Добавление организации - AddOrg	10
3.2.11 Переименование организации - UpdateOrg	10
3.2.12 Удаление организации - DeleteOrg.....	10
3.2.13 Получение списка подразделений - GetDeparts	10
3.2.14 Получение списка подразделений, принадлежащих конкретной организации - GetDepartsByOrg	11
3.2.15 Добавление подразделения - AddDepart	11
3.2.16 Переименование подразделения - UpdateDepart	11
3.2.17 Удаление подразделения - DeleteDepart.....	11
3.3 Требования к информационной и программной совместимости	11
4 Работа с веб-сервисом	12
5 Особенности работы модуля интеграции.....	13
5.1 Особенности работы при блокировке или разблокировке.....	13
5.2 Особенности работы при создании пропуска.....	13

1 Наименование и область применения

Модуль интеграции предназначен для информационного взаимодействия с системой контроля и управления доступом (далее – СКУД) аппаратно-программного комплекса «Бастيون» (далее – АПК Бастيون).

2 Назначение разработки

Модуль интеграции является программой, обеспечивающей взаимодействие АПК Бастيون со внешней системой в целях решения задачи автоматического конфигурирования в СКУД карт доступа (КД) посетителей для оформляемых во внешней системе пропусков.

3 Технические требования

3.1 Функциональные характеристики

- 1) Модуль интеграции реализован как SOAP-совместимый веб-сервис для доступа к нему по протоколам HTTP/HTTPS;
- 2) Модуль интеграции описывается WSDL-документом для обеспечения подключения к нему клиентского приложения (далее – Клиент);
- 3) Веб-сервис при каждом его вызове возвращает числовой идентификатор, по которому запрашивается результат выполнения операции путем вызова отдельного метода;
- 4) В рамках решения задачи конфигурирования КД посетителей модуль интеграции обеспечивает автоматическое выполнение следующих функций:
 - 4.1) Передачу Клиенту данных о точках доступа (прохода) СКУД:
 - названия точки доступа;
 - код объекта, в котором используется точка доступа;
 - уникальный номер (идентификатор) точки доступа;
 - 4.2) Передачу Клиенту данных об организациях:
 1. Получение списка всех организаций с удалением повторяющихся названий;
 2. Получение списка организаций, имеющих в своем названии переданную подстроку;
 - 4.3) Передачу Клиенту данных о подразделениях (информация о подразделении включает в себя имя родительской организации):
 1. Получение списка всех подразделений с удалением повторяющихся названий;
 2. Получения списка подразделений, принадлежащих организации, название которой передано в запросе;
 - 4.4) Получение от Клиента запросов на редактирование организаций:
 1. Создание новой организации;
 2. Редактирование (переименование) существующей организации;
 3. Удаление организации;
 - 4.5) Получение от Клиента запросов на редактирование подразделений:
 1. Создание нового подразделения (принадлежащего организации, название которой передано в запросе) ;
 2. Редактирование (переименование) существующего подразделения;
 3. Удаления подразделения;
 - 4.6) Получение от Клиента запроса с данными для автоматической конфигурации КД и заявок на КД посетителю;

№ п/п	Наименование данных
1	Код (номер и серия) карты доступа (если данных нет – будет создана заявка)
2	Список разрешенных для прохода точек доступа
3	Время действия карты доступа посетителя: дата окончания, время начала и окончания разрешенного доступа, признаки разрешения прохода в выходные и праздничные дни.
4	Тип пропуска: <ul style="list-style-type: none">• разовый (РП);• временный (ВП);• постоянный (ПП).
5	Информация о посетителе (владелец КД): <ul style="list-style-type: none">• фамилия посетителя;• имя посетителя;• отчество посетителя;• дата рождения;• название должности посетителя;• название организации посетителя;• название департамента посетителя;• идентификатор посетителя (табельный номер сотрудника);• фотография посетителя (в формате JPEG, размер не более 64 килобайт).

4.7) Автоматическая конфигурация в СКУД КД по запросам, полученным согласно п.5.6:

1. Информация о посетителе записывается в базу данных (БД) СКУД.

2. Если код (номер и серия) карты доступа отсутствует в БД СКУД, то создаётся новая запись КД, если код карты не передан в запросе, то создаётся новая заявка на КД.

3. Для КД назначается пропуск с уровнем (режимом) доступа в соответствие с полученными в запросе требованиями по точкам и времени доступа для посетителя и устанавливается связь с информацией о ее текущем владельце. После конфигурации КД СКУД обеспечивает управление доступом по КД и отображает информацию о проходе пользователем собственными аппаратно-программными средствами.

4.8) Получение от Клиента запроса с данными для автоматического приостановления/возобновления действия пропуска по ранее сконфигурированной в СКУД КД:

- код (номер и серия) карты доступа;
- признак необходимости блокировки (1) / разблокировки (0) карты доступа.

4.9) Автоматический запрет допуска посетителя по запросу на блокировку КД без удаления в СКУД информации о пропуске (о владельце КД и разрешенном ему режиме доступа). Автоматическое снятие запрета допуска посетителя по КД по запросу на разблокирование КД (возобновление действия ранее заблокированного пропуска).

4.10) Получение от клиента запроса с данными для автоматического возврата сконфигурированного в СКУД пропуска:

- код (номер и серия) карты доступа.

4.11) Получение от клиента запроса с данными для автоматического удаления сконфигурированного в СКУД пропуска:

- код (номер и серия) карты доступа.

4.12) Режим получения и применения в АПК Бастион команд конфигурирования КД, поступающих из внешней системы синхронный, время обработки команд находится в пределах 1 минуты (в зависимости от количества серверов и их доступности).

5) Алгоритм работы модуля интеграции разработан с учетом следующих функциональных требований и ограничений по управлению пропусками:

5.1) Оформление и утверждение заявки на пропуск реализовано средствами внешней системы;

5.2) При оформлении пропуска во внешней системе модулем интеграции в АПК Бастион передаётся запрос на конфигурирование КД с данными о пропуске.

5.3) При переоформлении пропуска от внешней системы в АПК Бастион передаётся запрос на конфигурирование КД с измененными значениями параметров выданного ранее типа пропуска (персональные данные владельца КД, список точек доступа, время и срок действия пропуска).

5.4) Печать временного пропуска на карте доступа производится как во внешней системе, так и в СКУД.

5.5) В БД АПК Бастион идентификация и поиск владельца КД выполняются по значению составного атрибута **<ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ДАТА РОЖДЕНИЯ>** (далее – основной идентификатор). Поля **<ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ДАТА РОЖДЕНИЯ>** в запросах на конфигурирование КД обязательны к заполнению, - иначе возвращается ошибка. При совпадении значений основного идентификатора у посетителя (в запросе от внешней системы) и владельца КД (в БД АПК Бастион) принимается решение об их идентичности.

5.6) Наличие в АЗСП полных однофамильцев по основному идентификатору может привести к ошибочной (неправильной) идентификации владельца КД (Персоны) на стороне АПК Бастион.

5.7) Выданная КД может быть временно заблокирована по запросу от внешней системы. При этом соответствующий пропуск остаётся «активным» в АПК Бастион.

5.8) Возврат КД осуществляется в АПК Бастион автоматически при сдаче через картоприёмник или вручную администратором АПК Бастион. При возврате КД в АПК Бастион никакая информация не передаётся во внешнюю систему. В АПК Бастион существует возможность возврата заблокированного пропуска.

- б) Схема алгоритма обработки и применения данных, поступивших в АПК Бастион из внешней системы, представлена на Рисунке 1:

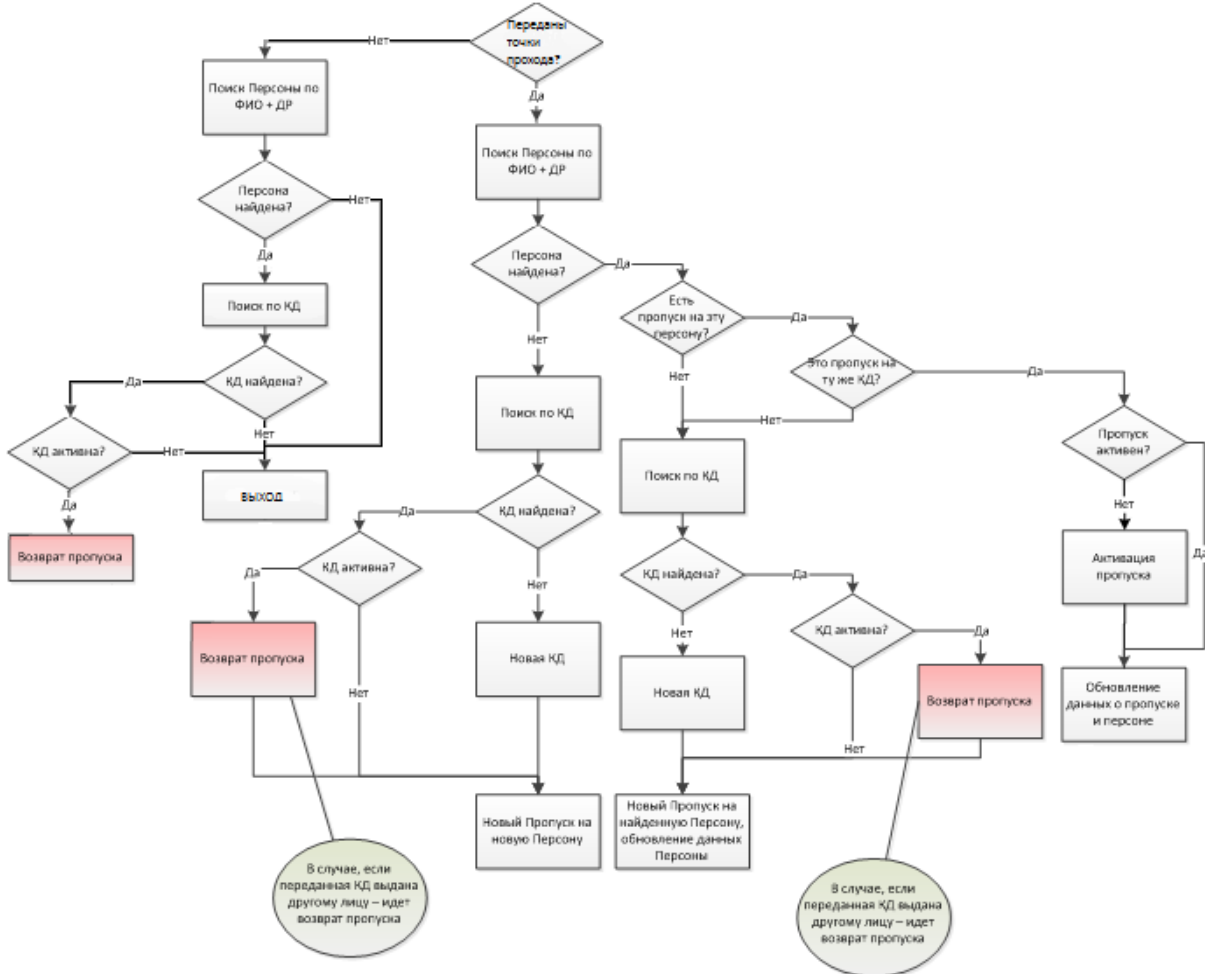


Рисунок 1

3.2 Спецификация методов веб-сервиса

3.2.1 Получение информации о подключении к веб-сервису – GetServerVersion

Формат вызова: **GetServerVersion: result**, где:

- **result** – возвращаемая в результате выполнения метода строка, содержащая данные о версии Веб-Сервиса (версия ninteg.dll), например = «1.3».

3.2.2 Получение результата выполнения операции - GetResult

Формат вызова: **GetResult(opcode): result**, где:

- **opcode** - целочисленный (integer) идентификатор результата операции (возвращается в результате вызова любого из методов веб-сервиса, кроме GetServerVersion, который возвращает строку с версией сервиса);

- **result** - объект класса TResult, который представляет собой результат выполнения операции, и включает в себя данные:

№ п/п	Название поля	Тип	Описание
1	Code	integer	Целочисленный код результата операции
2	Description	string	Строка с описанием результата операции

Возможные результаты операций:

Код	Описание результата операции
-1	Операция еще не завершена
0	Операция выполнена успешно
1	Операция выполнена успешно не на всех серверах
3	Не сконфигурированы доступные сервера для интеграции
4	Операция не выполнена ни на одном сервере
5	Нет связи с сервером
51	Нет связи с сервером (указаны неверные логин и/или пароль)
52	Нет связи с сервером (таймаут подключения: указан неверный ip адрес, сервер не запущен, либо подключение блокируется фаерволом)
53	Нет связи с сервером (TNS: no listener. Указаны неверные порт и/или имя сервиса, на сервере не запущен прослушивающий процесс (listener), либо сервер неверно сконфигурирован)
6	Сервер отсутствует в конфигурации или не лицензирован
7	Сервер не сконфигурирован (IP-адрес не указан, либо вместо IP-адреса введена некорректная строка)
8	Неизвестная ошибка при выполнении команды
9	Ошибка применения данных на сервере
10	Неверные входные параметры
11	Ошибка лицензирования

Результат выполнения операции с кодом -1 возможен в случае, если операция асинхронная и в момент вызова метода GetResult она находится в процессе выполнения. Для получения результата выполнения данной операции необходимо вызвать метод GetResult через некоторое время.

Результаты выполнения операции с кодами 1 (операция выполнена успешно не на всех серверах) и 4 (операция не выполнена ни на одном сервере) содержат так же в конце описания (поле Description) список серверов, на которых операция не была выполнена в результате ошибки применения или запроса данных на сервере в формате:

«;CLI1;CLI2»,

где CLI1 и CLI2 - коды серверов, на которых операция не выполнена.

3.2.3 Получение данных о точках доступа (прохода) СКУД - GetEntryPoints

Формат вызова: **GetEntryPoints(out TEntryPoint[]): result**; где **TEntryPoint[]** – массив точек прохода **TEntryPoint** в виде выходного параметра - данные в виде списка записей, описывающих точки прохода (доступа) СКУД, а **result** - идентификатор результата выполнения операции.

Метод синхронный, т.е. возвращает управление после сбора данных. Результат выполнения операции доступен сразу после вызова метода (путем вызова метода **getResult** с передачей в него идентификатора выполнения операции **result**).

Описание класса TEntryPoint:

№ п/п	Название поля	Тип	Описание
1	ServID	string	Клиент, на котором стоит точка прохода
2	SubDeviceNo	string	Уникальный номер устройства в АПК Бастион
3	SubDeviceName	string	Название точки прохода

В случае ошибки, возвращается пустой указатель. Наличие достоверной информации о точках прохода является критически важной для работы системы. Для успешного выполнения процедуры сбора информации о точках прохода, необходимо обеспечить связь со всеми серверами «АПК «Бастион». Полученные точки прохода необходимо сохранить на стороне клиента и пользоваться ими. По этому, при вызове функции **GetEntryPoints** – результат возвращается только при гарантированности получения полноценной и достоверной информации.

3.2.4 Добавление/редактирования пропуска в СКУД – MsgPutPass

Формат вызова: **MsgPutPass(CardNo, SiteCode, PersonData, EntryPointsArray, TimeInterval, dateFrom, dateTo, PassType): result**, где **result** - идентификатор результата выполнения операции.

Описание входных параметров:

№ п/п	Название	Тип	Описание
1	cardNo	string	Номер карты доступа, представляет собой остаток от деления полного номера карты (CardFullNo) на 65536: $ОСТАТ(CardFullNo;65536)$; Может быть пустым: в этом случае будет создана заявка на КД
2	siteCode	integer	Серия карты доступа, представляет собой остаток от деления на 256 значения целочисленного деления полного номера карты: $ОСТАТ(ЦЕЛОЕ(CardFullNo /65536); 256)$;
3	personData	TPersonData	Информация о посетителе
4	entryPointsArray	TEntryPoints[]	Массив территорий (список точек доступа)
5	timeInterval	TTimeInterval	Время действия пропуска
6	dateFrom	DateTime	Дата начала действия
7	dateTo	DateTime	Дата завершения действия
8	passType	integer	Тип пропуска: 1 – постоянный 2 – временный 4 – разовый

Метод асинхронный, т.е. возвращает управление сразу после вызова. Результат выполнения операции доступен после завершения применения данных на всех серверах (путем вызова метода **getResult** с передачей в него идентификатора выполнения операции **result**).



Описание атрибутов класса TPersonData (данные посетителя):

№ п/п	Имя	Тип	Описание	Обязательность заполнения
1	NAME	string	Фамилия	Да
2	FIRSTNAME	string	Имя	Да
3	SECONDNAME	string	Отчество	Да
4	TABLENO	string	Табельный номер	Нет
5	PERSONCAT	string	Категория	Нет
6	ORG	string	Организация	Да
7	DEP	string	Департамент	Да
8	POST	string	Должность	Да
9	COMMENTS	string	Комментарии	Нет
10	DOCISSUEORGAN	string	Орган, выдавший документ, удостоверяющий личность	Нет
11	DOCSE	string	Серия документа	Нет
12	DOCNO	string	Номер документа	Нет
13	DOCISSUEDATE	DateTime	Дата выдачи документа	Нет
14	BIRTHDATE	DateTime	Дата рождения	Да
15	BIRTHPLACE	string	Место рождения	Нет
16	ADDRESS	string	Адрес прописки	Нет
17	PHONE	string	Телефон	Нет
18	FOTO	string (Base64 закодированная строка)	Фото	Нет
19	ADDFIELDS	string	Унифицированный список дополнительных полей	Нет

Описание унифицированного списка дополнительных полей:

Все данные, которые необходимо занести в АПК «Бастيون», но для которых нет соответствующих полей, - заносятся в дополнительные поля от ADDFLD1 до ADDFLD20.

В рамках интеграции:

№ п/п	Имя	Тип	Поле	Описание
1	PRINTCARDNO	string	AddFld1	Номер, напечатанный на КД
2	CAR	string	AddFld2	Марка машины.
3	CARNUMBER	string	AddFld3	Номер машины

Таким образом, если необходимо передать данные в АПК «Бастيون», то:
ADDFIELDS := «<ADDFLD1>PRINTCARDNO</ADDFLD1><ADDFLD2>CAR</ADDFLD2><ADDFLD3>CARNUMBER</ADDFLD3>».

Назначение дополнительных полей настраивается штатными средствами АПК «Бастيون».

Описание атрибутов класса TEntryPoint:

№ п/п	Имя	Тип	Описание
1	SERV_ID	string	Клиент, на котором стоит точка прохода
2	SUBDEVICENO	integer	Уникальный номер устройства в АПК Бастيون
3	SUBDEVICENAME	string	Название точки прохода

Допускается не более 1200 точек прохода на одном клиенте АПК «Бастيون».

Описание атрибутов класса TTimeInterval:

№	Имя	Тип	Описание
---	-----	-----	----------

п/п			
1	TimeStart	string	Время начала действия пропуска в течение дня, например, '02:03' или '15:30'
2	TimeEnd	string	Время окончания действия пропуска в течение дня
3	InSaturday	integer	1/0 – Разрешение/Запрет прохода в субботу (по умолчанию 0)
4	InSanday	integer	1/0 – Разрешение/Запрет прохода в воскресенье (по умолчанию 0)
5	InHollidays	integer	1/0 – Разрешение/Запрет прохода в праздники (по умолчанию 0)

При выполнении метода должен создаваться временный уровень доступа, который автоматически удаляется вместе с пропуском. Создаваемый уровень доступа должен включать в себя блок доступа с недельным расписанием (понедельник, вторник, среда, четверг, пятница, суббота, воскресенье) с одинаковыми интервалом времени доступа в каждый день с TimeStart по TimeEnd.

3.2.5 Блокировка/разблокировка пропуска MsgBlockPass

Формат вызова: **MsgBlockPass(CardNo, SiteCode, Blocked): result**, где **result** - идентификатор результата выполнения операции.

Описание входных параметров:

№ п/п	Название поля	Тип	Описание входных данных
1	cardNo	string	Номер карты доступа
2	siteCode	integer	Серия карты доступа
3	blocked	integer	Признак необходимости блокировки/разблокировки карты 1 – карту заблокировать 0 – карту разблокировать

Метод асинхронный, т.е. возвращает управление сразу после вызова. Результат выполнения операции доступен после завершения применения данных на всех серверах (путем вызова метода GetResult с передачей в него идентификатора выполнения операции **result**).

Обработка ситуаций при работе с пропусками приведена в таблице:

№ п/п	Ситуация	Реализация
1	Команда Блокировка/Разблокировка пропуска во внешней системе действующего активного пропуска с указанной в команде картой доступа.	Блокировка в АПК Бастион за счет изменения поля BLOCKEDID.
2	Команда блокировки / разблокировки карты при отсутствии такой карты в базе	Формируется сообщение об ошибке выполнения команды (просмотр и обработка согласно пп.4 п.4.1)

3.2.6 Возврат пропуска MsgReturnPass

Формат вызова: **MsgReturnPass(srvld, CardNo, SiteCode): result**, где **result** - идентификатор результата выполнения операции.

Описание входных параметров:

№ п/п	Название поля	Тип	Описание входных данных
1	srvld	string	Код сервера, на котором требуется выполнить возврат пропуска. При передаче пустой строки возврат будет выполнен на всех серверах

2	cardNo	string	Номер карты доступа
3	siteCode	integer	Серия карты доступа

Метод асинхронный, т.е. возвращает управление сразу после вызова. Результат выполнения операции доступен после завершения применения данных на всех серверах (путем вызова метода `GetResult` с передачей в него идентификатора выполнения операции **result**).

Обработка ситуаций при работе с пропусками приведена в таблице:

№ п/п	Ситуация	Реализация
1	Команда возврата во внешней системе действующего активного пропуска с указанной в команде картой доступа.	Возврат пропуска в АПК Бастион (cardStatus = 3 (возвращен))
2	Команда возврата карты при отсутствии такой карты в базе	Формируется сообщение об ошибке выполнения команды (просмотр и обработка согласно пп.4 п.4.1)

3.2.7 Удаление пропуска `MsDeletePass`

Формат вызова: **MsgDeletePass(CardNo, SiteCode): result**, где **result** - идентификатор результата выполнения операции.

Описание входных параметров:

№ п/п	Название поля	Тип	Описание входных данных
1	cardNo	string	Номер карты доступа
2	siteCode	integer	Серия карты доступа

Метод асинхронный, т.е. возвращает управление сразу после вызова. Результат выполнения операции доступен после завершения применения данных на всех серверах (путем вызова метода `GetResult` с передачей в него идентификатора выполнения операции **result**).

Обработка ситуаций при работе с пропусками приведена в таблице:

№ п/п	Ситуация	Реализация
1	Команда удаления во внешней системе действующего активного пропуска с указанной в команде картой доступа.	Удаление пропуска в АПК Бастион
2	Команда удаления пропуска при отсутствии указанной карты в базе	Формируется сообщение об ошибке выполнения команды (просмотр и обработка согласно пп.4 п.4.1)

3.2.8 Получение списка организаций

Формат вызова: **GetOrgs(out TOrg[]): result**, где **TOrg[]** - массив организаций **TOrg** в виде выходного параметра - данные в виде списка записей-организации, а **result** - идентификатор результата выполнения операции.

Метод синхронный, т.е. возвращает управление после сбора данных. Результат выполнения операции доступен сразу после вызова метода (путем вызова метода `GetResult` с передачей в него идентификатора выполнения операции **result**).

Описание класса `TOrg`:

№ п/п	Название поля	Тип	Описание
1	OrgName	string	Название организации

3.2.9 Получение список организаций по фильтру - GetOrgs(filter)

Формат вызова: **GetOrgs(filter, out TOrg[])**: **result**, где **filter** - строка, которую должны содержать в своем названии организации, возвращаемые методом, **TOrg[]** - массив организаций **TOrg** в виде выходного параметра - данные в виде списка записей-организаций, а **result** - идентификатор результата выполнения операции.

Метод синхронный, т.е. возвращает управление после сбора данных. Результат выполнения операции доступен сразу после вызова метода (путем вызова метода GetResult с передачей в него идентификатора выполнения операции **result**).

Описание класса TOrg:

№ п/п	Название поля	Тип	Описание
1	OrgName	string	Название организации

3.2.10 Добавление организации - AddOrg

Формат вызова: **AddOrg(TOrg)**: **result**, где **TOrg** - организация, а **result** - идентификатор результата выполнения операции.

Метод асинхронный, т.е. возвращает управление сразу после вызова. Результат выполнения операции доступен после завершения применения данных на всех серверах (путем вызова метода GetResult с передачей в него идентификатора выполнения операции **result**).

3.2.11 Переименование организации - UpdateOrg

Формат вызова: **UpdateOrg(TOrg, newName)**: **result**, где **TOrg** - организация, **newName** - строка с новым именем организации, а **result** - идентификатор результата выполнения операции.

Метод асинхронный, т.е. возвращает управление сразу после вызова. Результат выполнения операции доступен после завершения применения данных на всех серверах (путем вызова метода GetResult с передачей в него идентификатора выполнения операции **result**).

3.2.12 Удаление организации - DeleteOrg

Формат вызова: **DeleteOrg(TOrg)**: **result**, где **TOrg** - организация, а **result** - идентификатор результата выполнения операции.

Метод удаляет организацию и все дочерние подразделения на всех серверах.

Метод асинхронный, т.е. возвращает управление сразу после вызова. Результат выполнения операции доступен после завершения применения данных на всех серверах (путем вызова метода GetResult с передачей в него идентификатора выполнения операции **result**).

3.2.13 Получение списка подразделений - GetDeparts

Формат вызова: **GetDeparts(out TDepart[])**: **result**, где **TDepart[]** - массив организаций **TDepart** в виде выходного параметра - данные в виде списка записей-подразделений, а **result** - идентификатор результата выполнения операции.

Метод возвращает список всех подразделений со всех организаций с удалением повторяющихся названий.

Метод синхронный, т.е. возвращает управление после сбора данных. Результат выполнения операции доступен сразу после вызова метода (путем вызова метода GetResult с передачей в него идентификатора выполнения операции **result**).

Описание класса TDepart:

№ п/п	Название поля	Тип	Описание
-------	---------------	-----	----------

1	DepName	string	Название подразделения
2	OrgName	string	Название родительской организации

3.2.14 Получение списка подразделений, принадлежащих конкретной организации - **GetDepartsByOrg**

Формат вызова: **GetDepartsByOrg(TOrg, out TDepart[]): result**, где **TOrg** - родительская организация, **TDepart[]** - массив организаций **TDepart** в виде выходного параметра - данные в виде списка записей-подразделений, а **result** - идентификатор результата выполнения операции.

Метод возвращает список всех подразделений, принадлежащих конкретно заданной организации.

Метод синхронный, т.е. возвращает управление после сбора данных. Результат выполнения операции доступен сразу после вызова метода (путем вызова метода **getResult** с передачей в него идентификатора выполнения операции **result**).

3.2.15 Добавление подразделения - **AddDepart**

Формат вызова: **AddDepart(TDepart): result**, где **TDepart** - подразделение, а **result** - идентификатор результата выполнения операции.

Метод добавляет новое подразделение в организацию, имя которой указано в параметре **OrgName** объекта **TDepart**.

Метод асинхронный, т.е. возвращает управление сразу после вызова. Результат выполнения операции доступен после завершения применения данных на всех серверах (путем вызова метода **getResult** с передачей в него идентификатора выполнения операции **result**).

3.2.16 Переименование подразделения - **UpdateDepart**

Формат вызова: **UpdateDepart(TDepart, newName): result**, где **TDepart** - Подразделение, **newName** - строка с новым именем подразделения, а **result** - идентификатор результата выполнения операции.

Метод переименовывает подразделение, имя которого указано в параметре **DepName**, а имя родительской организации в параметре **OrgName** объекта **TDepart**.

Метод асинхронный, т.е. возвращает управление сразу после вызова. Результат выполнения операции доступен после завершения применения данных на всех серверах (путем вызова метода **getResult** с передачей в него идентификатора выполнения операции **result**).

3.2.17 Удаление подразделения - **DeleteDepart**

Формат вызова: **DeleteDepart(TDepart): result**, где **TDepart** - организация, а **result** - идентификатор результата выполнения операции.

Метод удаляет подразделение, имя которого указано в параметре **DepName**, а имя родительской организации в параметре **OrgName** объекта **TDepart**.

Метод асинхронный, т.е. возвращает управление сразу после вызова. Результат выполнения операции доступен после завершения применения данных на всех серверах (путем вызова метода **getResult** с передачей в него идентификатора выполнения операции **result**).

3.3 Требования к информационной и программной совместимости

Допускается использование любой операционной системы на платформе Win32, не имеющей ограничений по обработке запросов к веб-сервису.

4 Работа с веб-сервисом

Ядром веб-сервера интеграции является сервис IntegrationService.svc, развернутый в составе веб-портала интеграции как компонент Internet Information Server (IIS). Описание веб-сервиса на Web Services Description Language (WSDL) можно получить через страницу веб-сервиса, набрав адрес:

<http://localhost/BInteg/IntegrationService.svc?wsdl>

На этапе подготовки должны быть развернуты сервера АПК «Бастион» в требуемом количестве (Бастион версии не ниже 1.7.4.10). Каждый сервер должен быть запущен и сконфигурирован. На каждом сервере АПК «Бастион» должен быть установлен драйвер ELSYS и сконфигурировано хотя бы одно устройство. Так же, при необходимости использовать дополнительные поля, - они должны быть сконфигурированы.

WEB-сервис должен быть развернут в соответствии с документом «Бастион-Интеграция. Web-сервер. Руководство инсталлятора.doc» и должен иметь доступ по сети к компьютерам, на которых развернуты сервера АПК «Бастион».

Веб-сервис настраивается с помощью веб-страницы сервера: <http://localhost/BInteg/servers.aspx>.

Код сервера	Наименование сервера	Имя компьютера в сети	IP-адрес	Путь к базе данных	Добавить
CLI1	Клиент 1	cli1.domain.ru	192.168.1.2	C:\Bastion\Data\Bastion.gdb	✎ ✕ ✕
CLI2	Клиент 2	cli2.domain.ru	192.168.1.3	C:\Bastion\Data\Bastion.gdb	✎ ✕ ✕
LOCAL	Локальный сервер	localhost	127.0.0.1	C:\Bastion\Data\Bastion.gdb	✎ ✕ ✕

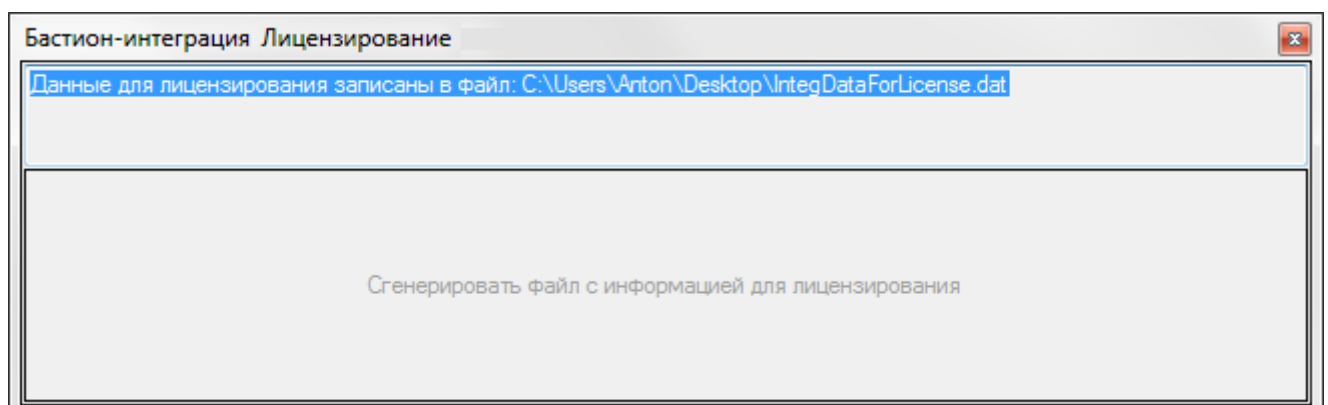
Система интеграции АПК "БАСТИОН", версия 1.4
Ассоциация "Электронные системы", ООО «ЕС-Пром», Солнечная ул., 53, Самара, 443029 Тел/факс: (846) 243-90-90 (многоканальный)
develop@elsystems.ru www.trevog.net

Полями, необходимыми для заполнения являются:


- SERV_ID (1 столбец) – краткое наименование клиента латиницей
- SERV_IP (4 столбец) – IP адрес клиента
- SERV_DBPATH (5 столбец) – путь к БД Bastion.gdb на клиенте.

Данные настройки хранятся в файле servers.xml.

Перед началом работы необходимо получить лицензию на использование модуля интеграции с АПК «Бастион». Для этого необходимо запустить утилиту LicenseInfo.exe. При помощи которой необходимо сформировать файл запроса получения лицензии integDataForLicense.dat, который надо выслать в службу тех. поддержки АПК «Бастион».



После того как служба тех. поддержки пришлёт лицензионный файл License.dat, его необходимо положить в папку рядом веб-портала интеграции (c:\inetpub\wwwroot\bininteg\). Количество серверов БД АПК «Бастион», сконфигурированных в файле Servers.xml, не должно превышать количества, прописанного в лицензионном файле.

На WEB-сервисе необходимо зайти на страничку сервиса по пути вида «http://localhost/BInteg/servers.aspx». Нажать на каждой строке с настройками клиентов кнопку кнопку , которая проверяет соединение с указанной БД. При прошедшей проверке и синхронизации на страничке формируется результат вида: **Операция выполнена успешно**, либо, при неудачной синхронизации строка с описанием результата выполнения операции.

5 Особенности работы модуля интеграции

5.1 Особенности работы при блокировке или разблокировке

Если вызов функции **msgBlockPass(CardNo, SiteCode, 1)** возвращает «1», то пропуск считается заблокированным. В бюро пропусков (БП) заблокированный пропуск меняет цвет шрифта на красный и стиль на «bold». Тем не менее, следует учитывать, что пропуск будет заблокирован только в момент автоматической инициализации контроллеров (каждые 3 минуты, но не более 100 пропусков за каждый раз).

Заявки							
Выданные		Просроченные		Архив			
Фамилия	Имя	Отчество	Таб. №	№ пропуска	Серия пропуска	Уровень доступа	Должность
▶ Задов	Василий	Петрович	0000000001	888		88 (авто) №2 08:00-17:00	нет

При предъявлении заблокированного пропуска в АПК «Бастион» поступает событие вида «Блокированная карта на вход Задов 888», и, если настроена фотоидентификация – будет выведено окно с пропуском и фотографией сотрудника.

Если вызов функции **msgBlockPass(CardNo, SiteCode, 0)** возвращает «1», то пропуск считается разблокированным. В бюро пропусков (БП) при обновлении данных о пропусках разблокированный пропуск поменяет цвет шрифта на цвет «по умолчанию» и стиль на «обычный». Реально пропуск будет разблокирован в момент автоматической инициализации контроллеров (каждые 3 минуты, но не более 100 пропусков за каждый раз).

Заявки							
Выданные		Просроченные		Архив			
Фамилия	Имя	Отчество	Таб. №	№ пропуска	Серия пропуска	Ур	
▶ Задов	Василий	Петрович	0000000001	888		88 (ав	

Блокировка привязана к сущности «пропуск», и если пропуск не будет удалён или принудительно разблокирован – сохраняется до конца существования записи (даже в архиве). При создании или изменении уже выданного активного пропуска на ту же персону с той же картой, – блокировка переносится на новую сущность.

5.2 Особенности работы при создании пропуска

Алгоритм работы приведён на Рисунке 1. Из особенностей работы следует заметить, что для упрощения функционирования и сохранения истории изменений – реализовано:

- 1) При изменении пропуска, происходит возврат его в архив с причиной «Возвращён web репликацией»;
- 2) Если по какой-то причине карта была активна и выдана какому-либо сотруднику, то происходит возврат пропуска этого сотрудника в архив с пометкой «Возвращён web репликацией»;



- 3) При вызове функции создания пропуска с разными КД для одной персоны несколько раз – будет создано несколько пропусков на одну и ту же персону в АПК "Бастион".