

# ОПИСАНИЕ СЕРВИСА ИНТЕГРАЦИИ PARSECNET

## версия от 18.10.2018

### ОГЛАВЛЕНИЕ

<b>1. ВВЕДЕНИЕ .....</b>	<b>6</b>
Общие положения.....	6
Принцип работы сервиса .....	6
Функционал клиентского приложения .....	6
Некоторые рекомендации разработчикам .....	7
Состав комплекта разработчика .....	7
Поддержка функций интеграционного сервиса продуктами PNSoft и PNOOffice .....	7
<b>2. ИСПОЛЬЗУЕМЫЕ ОБЪЕКТЫ .....</b>	<b>11</b>
BaseResult .....	11
GuidResult .....	11
SessionResult .....	11
StringResult .....	11
ObjectResult.....	12
EventsHistoryResult.....	12
Session .....	12
BaseObject.....	12
BaseOrgUnit .....	12
OrgUnit.....	13
BasePerson .....	13
Person .....	13
PersonWithPhoto .....	13
PersonExtraFieldTemplate .....	14
PersonScheduleFix .....	14
ExtraFieldValue .....	14
VisitorRequest .....	15
Schedule .....	15
AccessSchedule .....	15
WorktimeSchedule .....	16
ScheduleDay .....	16
ScheduleFix .....	16
TimeInterval.....	16
WorktimeInterval .....	17
Holiday .....	17
BaseIdentifier .....	17
Identifier.....	17

IdentifierTemp .....	18
StockIdentifier .....	18
BaseTerritory .....	18
Territory .....	19
TerritoryWithComponent .....	19
AccessGroup.....	20
SubAccessGroup .....	20
Event.....	20
EventsHistory .....	20
EventObject .....	21
Domain.....	21
EventHistoryQueryParams .....	21
HardwareState.....	23
<b>3. ФУНКЦИИ.....</b>	<b>24</b>
Функция GetVersion.....	24
Функция GetDomains .....	24
Функция OpenSession .....	24
Функция OpenSessionWithInLocale .....	24
Функция ContinueSession .....	25
Функция CloseSession .....	25
Функция CheckRole .....	25
Функция SendHardwareCommand .....	26
Функция GetHardwareState .....	26
Функция GetRootOrgUnit.....	28
Функция GetOrgUnitsHierarhy .....	28
Функция GetOrgUnitsHierarhyWithPersons.....	28
Функция GetOrgUnitsHierarhyWithVisitors .....	28
Функция GetOrgUnitsHierarhyWithVehicle.....	28
Функция GetOrgUnitSubItems.....	29
Функция GetOrgUnitSubItemsHierarhyWithPersons.....	29
Функция GetOrgUnitSubItemsHierarhyWithVisitors .....	29
Функция GetOrgUnitSubItemsHierarhyWithVehicle.....	29
Функция GetOrgUnit .....	30
Функция CreateOrgUnit .....	30
Функция OpenOrgUnitEditingSession.....	30
Функция CloseOrgUnitEditingSession .....	30
Функция SaveOrgUnit .....	31
Функция DeleteOrgUnit .....	31
Функция GetPersonExtraFieldTemplates .....	31

Функция GetVisitorExtraFieldTemplates .....	31
Функция GetVehicleExtraFieldTemplates .....	31
Функция FindPeople .....	32
Функция FindVisitors .....	32
Функция FindVehicle .....	32
Функция FindPersonByIdentifier .....	33
Функция GetPerson .....	33
Функция GetPersonsChangedAfter .....	33
Функция GetPersonExtraFieldValue .....	33
Функция GetPersonExtraFieldValues .....	34
Функция GetPersonExtraFieldValueString .....	34
Функция ValidateExtraFieldValue .....	34
Функция GetPersonScheduleFixes .....	34
Функция AddPersonScheduleFix .....	35
Функция SavePersonScheduleFix .....	35
Функция DeletePersonScheduleFix .....	35
Функция CreatePerson .....	35
Функция CreateVisitor .....	36
Функция CreateVehicle .....	36
Функция OpenPersonEditingSession .....	36
Функция ClosePersonEditingSession .....	36
Функция SavePerson .....	37
Функция SetPersonPhoto .....	37
Функция SetPersonOrgUnit .....	37
Функция SetPersonExtraFieldValue .....	37
Функция SetPersonExtraFieldValues .....	38
Функция DeletePerson .....	38
Функция BlockPerson .....	38
Функция UnblockPerson .....	39
Функция GetRootTerritory .....	39
Функция GetTerritoriesHierarchy .....	39
Функция GetTerritorySubItems .....	39
Функция GetTerritory .....	40
Функция GetPersonIdentifiers .....	40
Функция DeleteIdentifier .....	40
Функция AddPersonIdentifier .....	40
Функция ChangePersonIdentifier .....	41
Функция SetIdentifierPrivileges .....	41
Функция GetAccessSchedules .....	42

Функция GetWorktimeSchedules .....	42
Функция GetScheduleIntervals .....	42
Функция CreateAccessSchedule.....	43
Функция CreateWorktimeSchedule.....	43
Функция GetSchedule .....	44
Функция SaveSchedule .....	44
Функция DeleteSchedule .....	44
Функция GetScheduleDetails .....	44
Функция SetScheduleDays.....	44
Функция SetScheduleFix.....	45
Функция DeleteScheduleDays .....	45
Функция GetHolidays .....	45
Функция SetHolidays .....	46
Функция DeleteHolidays .....	46
Функция GetAccessGroups.....	46
Функция CreateTempAccessGroup .....	46
Функция CreateAccessGroup .....	47
Функция DeleteAccessGroup .....	47
Функция AddSubAccessGroup .....	47
Функция DeleteSubAccessGroup.....	48
Функция GetSubAccessGroups .....	48
Функция GetInheritedAccessGroups .....	48
Функция SetInheritedAccessGroups .....	48
Функция GetAcceptedVisitorRequests .....	49
Функция FindVisitorRequest .....	49
Функция ActivateVisitorRequest.....	49
Функция CreateVisitorRequest.....	49
Функция GetVisitorRequest.....	50
Функция SaveVisitorRequest .....	50
Функция DeleteIssuedVisitorRequest.....	50
Функция GetIssuedVisitorRequests .....	50
Функция GetVisitorRequests .....	51
Функция CloseAllActiveVisitorRequests .....	51
Функция CloseVisitorRequest .....	51
Функция GetPersonVisitorRequests .....	52
Функция GetEvents.....	53
Функция OpenEventHistorySession .....	53
Функция CloseEventHistorySession .....	53
Функция GetEventHistoryResultCount .....	54

Функция GetEventHistoryResult.....	54
Функция GetHardwareEvents .....	56
Функция GetHardwareEventsResolved.....	59
ИСТОРИЯ ИЗМЕНЕНИЙ.....	62
АЛФАВИТНЫЙ УКАЗАТЕЛЬ .....	65

# 1. ВВЕДЕНИЕ

## Общие положения

Данный документ описывает функционал интеграционного сервиса (далее просто «сервис») интегрированной системы безопасности ParsecNET.

Сервис позволяет сторонним программным комплексам получить доступ к данным и событиям системы ParsecNET для реализации специфического функционала, отсутствующего в системе. Сервис позволяет интегрировать такие внешние приложения, как управление кадрами, бюро пропусков, учет рабочего времени и многие другие, которым необходим доступ к информации о персонале (пользователях) ParsecNET и к событиям авторизованного доступа.

Сервис устанавливается автоматически на сервере системы ParsecNET и входит в состав базовой версии ПО.

## Принцип работы сервиса

Сервис работает на компьютере, являющемся сервером системы ParsecNET, как служба Windows. Он реализован как XML/SOAP WEB-сервис, что позволяет обращаться к нему с любого компьютера в сети, а не только с компьютера-сервера системы. Соответственно, внешнее приложение может быть создано в любой среде разработки, поддерживающей технологию WEB-сервисов и, в частности, SOAP. Сервис обслуживается приложением MDO.Parsec.ParsecIntegrationServiceHost.exe.

Адрес размещения сервиса указывается в конфигурационном файле MDO.Parsec.ParsecIntegrationServiceHost.exe.config (по-умолчанию – <http://localhost:10101/IntegrationService/IntegrationService.asmx>).

## Функционал клиентского приложения

Клиентское приложение по правам эквивалентно оператору одной конкретно взятой организации системы. Для получения доступа к сервису в системе необходимо создать оператора с необходимыми правами и областями видимости. Результат выполнения функций интеграционного сервиса определяется правами оператора в системе Parsec, с которыми открыта сессия.

Работа с сервисом идет на уровне сессий: клиентское приложение после логина открывает сессию и в рамках этой сессии использует функции сервиса. Сервис автоматически закроет сессию, если более пяти минут клиент не использовал обращений к Функциям сервиса.

Клиентское приложение с помощью сервиса может получить доступ к структуре персонала системы ParsecNET, включая всю иерархию подразделений в рамках конкретной организации.

Сервис предоставляет достаточно функций для того, чтобы можно было создать, отредактировать или удалить пользователя системы ParsecNET. При этом, как и при работе с приложениями системы, любое сделанное изменение без участия клиента реплицируется на все объекты системы, включая контроллеры.

Сервис позволяет создавать как временные, так и обычные (постоянные) группы доступа. Сервис может предоставить полный список групп доступа, чтобы назначить персоналу в клиентском приложении необходимые права доступа по территории объекта.

Кроме работы с персоналом сервис дает доступ к событиям системы.

Каждый получаемый из системы объект сопровождается его уникальным ключом, который необходим для идентификации этого объекта. В качестве ключей используются GUID-ы Windows.

## Некоторые рекомендации разработчикам

Данные из системы можно получать с использованием разных функций сервиса. Разработчик сам выбирает стратегию получения данных в зависимости от стоящих перед ним задач.

Однако следует иметь в виду, что некоторые функции могут в крупных системах возвращать очень большие массивы данных (в зависимости от масштаба самой системы). Соответственно, операция может выполняться достаточно долго и потреблять необоснованно много ресурсов.

Поэтому предпочтительно сначала получать общую иерархию сущности (например, дерево подразделений), а затем получать данные о персонале не для всей системы сразу, а для подразделения, с которым клиентское приложение реально в настоящее время работает.

## Состав комплекта разработчика

В комплект разработчика входят следующие компоненты:

Данное руководство

Пример клиентского приложения на языке C# с исходными текстами и работающим откомпилированным примером.

Пример клиентского приложения на языке Object Pascal с исходными текстами и работающим откомпилированным примером (среда разработки Delphi, Borland Developer Studio 2007. Более ранние версии Delphi не обеспечивают в полной мере поддержки требуемых технологий).

## Поддержка функций интеграционного сервиса продуктами PNSoft и PNOOffice

Интеграционный сервис, входящий в комплект программного обеспечения PNOOffice (ParsecNET Office) по причине отсутствия части функционала, например, бюро пропусков, поддерживает подмножество описанных в данном документе функций.

В таблице ниже приведены данные по совместимости функций в программных продуктах PNSoft и PNOOffice.

PNSoft	PNOOffice
<b>Работа с сессией</b>	
<a href="#">CloseSession</a>	
<a href="#">ContinueSession</a>	
<a href="#">OpenSession</a>	
<a href="#">OpenSessionWithInLocale</a>	
<b>Работа с группами доступа</b>	
<a href="#">AddSubAccessGroup</a>	не поддерживается
<a href="#">CreateAccessGroup</a>	не поддерживается
<a href="#">CreateTempAccessGroup</a>	
<a href="#">DeleteAccessGroup</a>	не поддерживается
<a href="#">DeleteSubAccessGroup</a>	не поддерживается
<a href="#">GetAccessGroups</a>	

<b>PNSoft</b>	<b>PNOffice</b>
<b>Наследование групп доступа</b>	
<a href="#">GetInheritedAccessGroups</a>	не поддерживается
<a href="#">SetInheritedAccessGroups</a>	не поддерживается
<b>Получение событий (архивных)</b>	
<a href="#">OpenEventHistorySession</a>	
<a href="#">GetEventHistoryResult</a>	
<a href="#">GetEventHistoryResultCount</a>	
<a href="#">CloseEventHistorySession</a>	
<a href="#">GetEvents</a> (устаревшая, не рекомендуется к использованию)	
<b>Получение оперативных событий и статусов контроллеров доступа</b>	
<a href="#">GetHardwareEvents</a>	не поддерживается
<a href="#">GetHardwareEventsResolved</a>	не поддерживается
<a href="#">GetHardwareState</a>	не поддерживается
<b>Команды прямого управления контроллерам доступа</b>	
<a href="#">SendHardwareCommand</a>	не поддерживается
<b>Работа с персоналом и идентификаторами</b>	
<a href="#">AddPersonIdentifier</a>	
<a href="#">ChangePersonIdentifier</a>	
<a href="#">CloseOrgUnitEditingSession</a>	
<a href="#">ClosePersonEditingSession</a>	
<a href="#">CreateOrgUnit</a>	
<a href="#">CreatePerson</a>	
<a href="#">DeleteIdentifier</a>	
<a href="#">DeleteOrgUnit</a>	
<a href="#">DeletePerson</a>	
<a href="#">FindPeople</a>	
<a href="#">FindPersonByIdentifier</a>	не поддерживается
<a href="#">GetOrgUnit</a>	
<a href="#">GetOrgUnitSubItems</a>	
<a href="#">GetOrgUnitSubItemsHierarchyWithPersons</a>	
<a href="#">GetOrgUnitsHierarchy</a>	
<a href="#">GetOrgUnitsHierarchyWithPersons</a>	
<a href="#">GetPerson</a>	
<a href="#">GetPersonIdentifiers</a>	
<a href="#">GetRootOrgUnit</a>	
<a href="#">OpenOrgUnitEditingSession</a>	
<a href="#">OpenPersonEditingSession</a>	
<a href="#">SaveOrgUnit</a>	
<a href="#">SavePerson</a>	
<a href="#">SetIdentifierPrivileges</a>	не поддерживается
<a href="#">SetPersonOrgUnit</a>	
<a href="#">SetPersonPhoto</a>	
<b>Дополнительные поля субъекта доступа</b>	
<a href="#">GetPersonExtraFieldTemplates</a>	
<a href="#">GetPersonExtraFieldValue</a>	
<a href="#">GetPersonExtraFieldValueString</a>	



<b>PNSoft</b>	<b>PNOffice</b>
<a href="#">GetPersonExtraFieldValues</a>	
<a href="#">SetPersonExtraFieldValue</a>	
<a href="#">SetPersonExtraFieldValues</a>	
<a href="#">ValidateExtraFieldValue</a>	
<b>Блокировка/разблокировка доступа</b>	
<a href="#">BlockPerson</a>	
<a href="#">UnblockPerson</a>	
<b>Работа с топологией</b>	
<a href="#">GetRootTerritory</a>	
<a href="#">GetTerritoriesHierarhy</a>	
<a href="#">GetTerritory</a>	
<a href="#">GetTerritorySubItems</a>	
<b>Расписания, праздники</b>	
<a href="#">CreateAccessSchedule</a>	не поддерживается
<a href="#">CreateWorktimeSchedule</a>	не поддерживается
<a href="#">DeleteSchedule</a>	не поддерживается
<a href="#">DeleteScheduleDays</a>	не поддерживается
<a href="#">GetAccessSchedules</a>	
<a href="#">GetSchedule</a>	не поддерживается
<a href="#">GetScheduleDetails</a>	не поддерживается
<a href="#">GetScheduleIntervals</a>	
<a href="#">SaveSchedule</a>	не поддерживается
<a href="#">SetScheduleDays</a>	не поддерживается
<a href="#">SetScheduleFix</a>	не поддерживается
<b>Праздники</b>	
<a href="#">GetHolidays</a>	не поддерживается
<a href="#">SetHolidays</a>	не поддерживается
<a href="#">DeleteHolidays</a>	не поддерживается
<b>Поправки рабочего времени</b>	
<a href="#">AddPersonScheduleFix</a>	не поддерживается
<a href="#">DeletePersonScheduleFix</a>	не поддерживается
<a href="#">GetPersonScheduleFixes</a>	не поддерживается
<a href="#">SavePersonScheduleFix</a>	не поддерживается
<b>Бюро пропусков, посетители</b>	
<a href="#">ActivateVisitorRequest</a>	не поддерживается
<a href="#">CloseAllActiveVisitorRequests</a>	не поддерживается
<a href="#">CloseVisitorRequest</a>	не поддерживается
<a href="#">CreateVisitor</a>	не поддерживается
<a href="#">CreateVisitorRequest</a>	не поддерживается
<a href="#">DeleteIssuedVisitorRequest</a>	не поддерживается
<a href="#">FindVisitors</a>	не поддерживается
<a href="#">FindVisitorRequest</a>	не поддерживается
<a href="#">GetAcceptedVisitorRequests</a>	не поддерживается
<a href="#">GetIssuedVisitorRequests</a>	не поддерживается
<a href="#">GetPersonVisitorRequests</a>	не поддерживается
<a href="#">GetVisitorRequest</a>	не поддерживается
<a href="#">SaveVisitorRequest</a>	не поддерживается

<b>PNSoft</b>	<b>PNOffice</b>
<a href="#">GetOrgUnitsHierarhyWithVisitors</a>	не поддерживается
<a href="#">GetOrgUnitSubItemsHierarhyWithVisitors</a>	не поддерживается
<a href="#">GetVisitorExtraFieldTemplates</a>	не поддерживается
<b>Security</b>	
<a href="#">CheckRole</a>	не поддерживается
<a href="#">GetDomains</a>	
<b>Получение версии SDK (версии ПО)</b>	
<a href="#">GetVersion</a>	

## 2. ИСПОЛЬЗУЕМЫЕ ОБЪЕКТЫ

### BaseResult

Базовый класс, используется в качестве результата исполнения операции.

<code>int</code> Result	Результат выполнения операции.
<code>string</code> ErrorMessage	Описание ошибки, произошедшей при выполнении операции.

Результат выполнения может быть следующим:

0 – операция выполнена успешно;

-1 – операция выполнена с ошибкой;

Значения >0 планируется использовать для кодов специфических ошибок.

### GuidResult

Базовый класс: [BaseResult](#).

Класс используется в качестве результата в функциях, возвращающих Guid.

<code>int</code> Result	Результат выполнения операции. 0 – операция выполнена успешно, -1 – операция выполнена с ошибкой. Значения >0 планируется использовать для кодов специфических ошибок.
<code>string</code> ErrorMessage	Описание ошибки, произошедшей при выполнении операции.
<code>Guid</code> Value	Результирующий ID.

### SessionResult

Базовый класс: [BaseResult](#).

Класс используется в качестве результата функции OpenSession.

<code>int</code> Result	Результат выполнения операции. 0 – операция выполнена успешно, -1 – операция выполнена с ошибкой. Значения >0 планируется использовать для кодов специфических ошибок.
<code>string</code> ErrorMessage	Описание ошибки, произошедшей при выполнении операции.
<code>Session</code> Value	Результат выполнения операции.

### StringResult

Базовый класс: [BaseResult](#).

Класс используется в качестве результата в функциях, возвращающих String.

<code>int</code> Result	Результат выполнения операции. 0 – операция выполнена успешно, -1 – операция выполнена с ошибкой. Значения >0 планируется использовать для кодов специфических ошибок.
-------------------------	--

<code>string</code> ErrorMessage	Описание ошибки, произошедшей при выполнении операции.
<code>String</code> Value	Результат выполнения операции.

## ObjectResult

Базовый класс: [BaseResult](#).

Класс используется в качестве результата в функциях, возвращающих Object.

<code>int</code> Result	Результат выполнения операции. 0 – операция выполнена успешно, -1 – операция выполнена с ошибкой. Значения >0 планируется использовать для кодов специфических ошибок.
<code>string</code> ErrorMessage	Описание ошибки, произошедшей при выполнении операции.
<code>Object</code> Value	Результат выполнения операции.

## EventsHistoryResult

Базовый класс: [BaseResult](#).

Класс используется в качестве результата функции [GetEvents](#).

<code>int</code> Result	Результат выполнения операции. 0 – операция выполнена успешно, -1 – операция выполнена с ошибкой. Значения >0 планируется использовать для кодов специфических ошибок.
<code>string</code> ErrorMessage	Описание ошибки, произошедшей при выполнении операции.
<code>EventsHistory</code> Value	Результат выполнения операции.

## Session

Класс, содержащий информацию для сессии.

<code>Guid</code> SessionID	Уникальный ключ сессии, используемый при выполнении дальнейших операций.
<code>Guid</code> RootOrgUnitID	ID корневого элемента дерева персонала.
<code>Guid</code> RootTerritoryID	ID корневого элемента дерева территорий.

## BaseObject

Класс, используемый в качестве базового для основных классов сервиса. Собственных членов не имеет.

## BaseOrgUnit

Базовый класс: [BaseObject](#).

Класс, используемый для описания подразделения.

<code>Guid</code> ID	Уникальный ключ подразделения.
----------------------	--------------------------------

<code>string</code> NAME	Название подразделения.
<code>string</code> DESC	Описание подразделения.

## OrgUnit

Базовый класс: [BaseOrgUnit](#).

Класс, используемый для описания подразделения.

<code>Guid</code> ID	Уникальный ключ подразделения.
<code>string</code> NAME	Название подразделения.
<code>string</code> DESC	Описание подразделения.
<code>Guid</code> PARENT_ID	Уникальный ключ родительского подразделения.

## BasePerson

Базовый класс: [BaseObject](#).

Класс, используемый для описания сотрудника.

<code>Guid</code> ID	Уникальный ключ сотрудника.
<code>string</code> LAST_NAME	Фамилия.
<code>string</code> FIRST_NAME	Имя.
<code>string</code> MIDDLE_NAME	Отчество.
<code>string</code> TAB_NUM	Табельный номер.

## Person

Базовый класс: [BasePerson](#).

Класс, используемый для описания сотрудника.

<code>Guid</code> ID	Уникальный ключ сотрудника.
<code>string</code> LAST_NAME	Фамилия.
<code>string</code> FIRST_NAME	Имя.
<code>string</code> MIDDLE_NAME	Отчество.
<code>string</code> TAB_NUM	Табельный номер.
<code>Guid</code> ORG_ID	Уникальный ключ подразделения для сотрудника.

## PersonWithPhoto

Базовый класс: [Person](#).

Класс, используемый для описания сотрудника.

<code>Guid</code> ID	Уникальный ключ сотрудника.
<code>string</code> LAST_NAME	Фамилия.
<code>string</code> FIRST_NAME	Имя.

<code>string</code> MIDDLE_NAME	Отчество.
<code>string</code> TAB_NUM	Табельный номер.
<code>Guid</code> ORG_ID	Уникальный ключ подразделения для сотрудника.
<code>byte[]</code> PHOTO	Фотография сотрудника.

## PersonExtraFieldTemplate

Класс, используемый для определения дополнительного поля данных персонала.

<code>Guid</code> ID	Уникальный ключ шаблона.
<code>XmlTypeCode</code> TYPE	Тип представленных данных.
<code>string</code> NAME	Наименование шаблона.

## PersonScheduleFix

Класс, используемый для определения поправки к рабочему времени сотрудника.

<code>Guid</code> FIX_ID	Уникальный ключ поправки.
<code>Guid</code> PERSON_ID	Уникальный ключ сотрудника.
<code>int</code> TYPE_ID	Тип поправки; может принимать значения: 4 – Больничный; 5 – Командировка; 6 – Отпуск; 7 – Принят на работу; 8 – Уволен; 9 – Отпуск без сохранения содержания; 10 – Полный рабочий день; 11 – Присутствие (только в этой поправке может передаваться ненулевое время, остальные поправки должны быть с временем 00:00).
<code>DateTime</code> START	Дата и время начала интервала поправки.
<code>DateTime</code> END	Дата и время завершения интервала поправки.
<code>string</code> COMMENT	Комментарий к поправке рабочего времени.

## ExtraFieldValue

Базовый класс: [BaseObject](#).

Класс, используемый для описания значения дополнительного поля сотрудника.

<code>Guid</code> TEMPLATE_ID	Уникальный ключ шаблона.
<code>Object</code> VALUE	Значение поля.

## VisitorRequest

Класс, используемый для описания заявок Бюро Пропусков

<code>Guid</code> ID	Уникальный ключ заявки.
<code>int</code> NUMBER	Уникальный номер заявки.
<code>DateTime</code> DATE	Дата создания заявки.
<code>Guid</code> ORGUNIT_ID	Уникальный ключ подразделения, для посещения которого создана заявка.
<code>Guid</code> PERSON_ID	Уникальный ключ посетителя.
<code>string</code> PERSON_INFO	Дополнительная информация о посетителе.
<code>string</code> PURPOSE	Цель визита.
<code>int</code> STATUS	Статус заявки.
<code>DateTime</code> ADMIT_START	Дата начала разрешенного посещения.
<code>DateTime</code> ADMIT_END	Дата окончания разрешенного посещения.

Статус заявки может принимать одно из значений:

- 0 – на стадии согласования;
- 1 – одобрена (можно выдавать идентификатор посетителю);
- 2 – отклонена (можно только закрыть заявку);
- 3 – активна (идентификатор на руках у посетителя);
- 4 – закрыта.

## Schedule

Базовый класс: [BaseObject](#).

Класс, используемый для представления расписания системы

<code>Guid</code> ID	Уникальный ключ расписания.
<code>string</code> NAME	Наименование.

## AccessSchedule

Базовый класс: [Schedule](#)

Класс, используемый для представления расписания доступа.

<code>Guid</code> ID	Уникальный ключ расписания.
<code>string</code> NAME	Наименование.
<code>string</code> DESC	Описание расписания.
<code>bool</code> IS_WEEK	Признак, является ли расписание недельным.
<code>int</code> HOLIDAYS_ACTION	Тип применения праздничных дней к расписанию: 0 – Применять с заменой 1 – Применять со вставкой 2 – Не применять

## WorktimeSchedule

Базовый класс: [AccessSchedule](#)

Класс, используемый для представления расписания рабочего времени.

<code>Guid</code> ID	Уникальный ключ расписания.
<code>string</code> NAME	Наименование.
<code>string</code> DESC	Описание расписания.
<code>bool</code> IS_WEEK	Признак, является ли расписание недельным.
<code>int</code> HOLIDAYS_ACTION	Способ применения праздничных дней к расписанию: 0 – Применять с заменой; 1 – Применять со вставкой; 2 – Не применять.
<code>int</code> HOURS_PER_WEEK	Норма отработки в неделю (часы).
<code>int</code> HOURS_PER_DAY	Норма отработки в день (часы).

## ScheduleDay

Базовый класс: [BaseObject](#).

Класс, используемый для описания шаблона дня в цикле расписания.

<code>DateTime</code> DATE	Дата начала цикла расписания.
<code>int</code> INDEX	Номер дня в цикле расписания. Индекс первого дня в цикле имеет значение «1».
<code>TimeInterval[]</code> INTERVALS	Массив временных интервалов в шаблоне дня.

## ScheduleFix

Базовый класс: [ScheduleDay](#)

Класс, используемый для описания дня-поправки.

<code>DateTime</code> DATE	Дата поправки.
<code>int</code> INDEX	Не используется.
<code>TimeInterval[]</code> INTERVALS	Массив временных интервалов в дне-поправке.
<code>int</code> ACTION	Способ применения дня-поправки в расписании: 0 – Применить с заменой; 1 - Применить со вставкой. Для неделных расписаний значение всегда «0».

## TimeInterval

Базовый класс: [BaseObject](#).

Класс, используемый для представления временного интервала



<code>DateTime</code> START	Начало интервала.
<code>DateTime</code> END	Окончание интервала.

## WorktimeInterval

Базовый класс: [TimeInterval](#)

Класс, используемый для представления временного интервала расписания рабочего времени.

<code>DateTime</code> START	Начало интервала.
<code>DateTime</code> END	Окончание интервала.
<code>int</code> TYPE	Тип временного интервала: 0 - Рабочее время (доступ разрешен); 1 - Ночная смена; 2 - Перерыв (обед); 3 - Обязательное рабочее время.

## Holiday

Базовый класс: [BaseObject](#).

Класс, используемый для описания праздничных дней.

<code>string</code> NAME	Название праздника.
<code>byte</code> MONTH	Месяц (даты праздника).
<code>byte</code> DAY	Число (даты праздника).

## BaseIdentifier

Базовый класс: [BaseObject](#).

Класс, используемый для описания идентификатора.

<code>string</code> CODE	Код идентификатора (строка, содержащая число в шестнадцатеричном формате, длиной ровно 8 символов).
<code>Guid</code> PERSON_ID	Уникальный ключ сотрудника.
<code>bool</code> IS_PRIMARY	Признак, является ли идентификатор первичным.

## Identifier

Базовый класс: [BaseIdentifier](#).

Класс, используемый для описания идентификатора.

<code>string</code> CODE	Код идентификатора (строка, содержащая число в шестнадцатеричном формате, длиной ровно 8 символов).
<code>Guid</code> PERSON_ID	Уникальный ключ сотрудника.
<code>bool</code> IS_PRIMARY	Признак, является ли идентификатор первичным.

<code>Guid</code> ACCGROUP_ID	Уникальный ключ группы доступа идентификатора.
<code>long</code> PRIVILEGE_MASK	Маска привилегий.
<code>int</code> IDENTIFYTYPE	Тип идентификатора. (Значения: 0 - Подсистема доступа «Parsec»; 1 – Автомобильный номер).

## IdentifierTemp

Базовый класс: [Identifier](#).

Класс, используемый для описания временного идентификатора.

<code>string</code> CODE	Код идентификатора (строка, содержащая число в шестнадцатичном формате, длиной ровно 8 символов).
<code>Guid</code> PERSON_ID	Уникальный ключ сотрудника.
<code>bool</code> IS_PRIMARY	Признак, является ли идентификатор первичным.
<code>Guid</code> ACCGROUP_ID	Уникальный ключ группы доступа идентификатора.
<code>DateTime</code> VALID_FROM	Дата начала действия идентификатора.
<code>DateTime</code> VALID_TO	Дата окончания действия идентификатора.

## StockIdentifier

Базовый класс: [IdentifierTemp](#).

Класс, используемый для описания идентификатора из пула.

<code>string</code> CODE	Код идентификатора (строка, содержащая число в шестнадцатичном формате, длиной ровно 8 символов).
<code>Guid</code> PERSON_ID	Уникальный ключ сотрудника.
<code>bool</code> IS_PRIMARY	Признак является ли идентификатор первичным.
<code>Guid</code> ACCGROUP_ID	Уникальный ключ группы доступа идентификатора.
<code>DateTime</code> VALID_FROM	Дата начала действия идентификатора.
<code>DateTime</code> VALID_TO	Дата окончания действия идентификатора.

## BaseTerritory

Базовый класс, используемый для описания территории.

<code>Guid</code> ID	Уникальный ключ территории.
<code>byte</code> TYPE	Тип объекта территории.
<code>string</code> NAME	Название территории.
<code>string</code> DESC	Описание территории.

Тип объекта территории может принимать значение:

0 – папка;

1 – дверь;

2 – деталь (пока не используется);

3 – остальные компоненты.

## Territory

Базовый класс: [BaseTerritory](#).

Класс, используемый для описания территории.

<code>Guid</code> ID	Уникальный ключ территории.
<code>byte</code> TYPE	Тип объекта территории.
<code>string</code> NAME	Название территории.
<code>string</code> DESC	Описание территории.
<code>Guid</code> PARENT_ID	Уникальный ключ родительской территории.

## TerritoryWithComponent

Базовый класс: [Territory](#).

Класс, используемый для описания территории с информацией о связанном с ней компоненте.

<code>Guid</code> ID	Уникальный ключ территории.
<code>byte</code> TYPE	Тип объекта территории.
<code>string</code> NAME	Название территории.
<code>string</code> DESC	Описание территории.
<code>Guid</code> PARENT_ID	Уникальный ключ родительской территории.
<code>Guid</code> COMPONENT_ID	Уникальный ключ компонента.
<code>long</code> FEATURE_MASK	Битовая маска свойств компонента. Значение битов описано в таблице ниже.

Маска свойств компонента:

Номер бита	Значение
0	Дверь
1	2 считывателя
2	АПБ
3	поддерживает сложные расписания
4	используется как настольный считыватель
5	охранная область
6	источник видео потока
7	турникет
8	-- Зарезервирован --
9	не поддерживает расписания
10	программный контроллер
11	-- Зарезервирован --

12	не ограничивается лицензией
13	лифтовой контроллер
14	-- Зарезервирован --
62	отключен оператором
63	не обслуживается по ограничениям лицензии

## AccessGroup

Класс, используемый для описания группы доступа.

<code>Guid</code> ID	Уникальный ключ группы доступа.
<code>string</code> NAME	Название группы доступа.
<code>int</code> IDENTIFYTYPE	Тип группы доступа. (Значения: 0 - Подсистема доступа «Parsec»; 1 – Автомобильный номер).

## SubAccessGroup

Базовый класс: [BaseObject](#).

Класс, используемый для описания подгруппы доступа.

<code>Guid</code> SubGroupID	Уникальный ключ группы компонент.
<code>Guid</code> ScheduleID	Уникальный ключ расписания.
<code>Guid[]</code> Territories	Массив ключей территорий.

## Event

Базовый класс: [BaseObject](#).

Класс, используемый для описания события системы.

<code>DateTime</code> EventDate	Дата события.
<code>int</code> EventType	Тип события (0-Вход; 1-Выход).
<code>int</code> EventPersonIndex	Индекс сотрудника в EventsHistory.
<code>string</code> CODE	Код идентификатора.
<code>int</code> EventTerritoryIndex	Индекс территории в EventsHistory.

Тип события может принимать значение:

0 – Вход;

1 – Выход.

## EventsHistory

Класс, используемый для описания событий системы.

<code>Event[]</code> Events	Массив событий.
<code>Guid[]</code> Persons	Массив уникальных ключей сотрудников.
<code>string[]</code> PersonFullNames	Массив Ф.И.О. сотрудников.

<code>Guid[] Territories</code>	Массив уникальных ключей территорий.
<code>string[] TerritoryNames</code>	Массив названий территорий.

## EventObject

Базовый класс: [BaseObject](#).

Класс, используемый для описания данных события.

<code>Object[] Values</code>	Массив значений полей, описывающих событие.
------------------------------	---

## Domain

Базовый класс: [BaseObject](#).

Класс, используемый для описания организации.

<code>string NAME</code>	Наименование организации.
<code>string DESCRIPTION</code>	Описание организации.
<code>bool VISITOR_CONTROL</code>	Признак Бюро Пропусков (не используется начиная с версии 3.2).
<code>bool IS_SYSTEM</code>	Признак организации SYSTEM.

## EventHistoryQueryParams

Класс, используемый для описания критериев формируемого отчета по событиям. В качестве любого параметра может быть передан null. В таком случае данный критерий не будет использован при отборе событий в отчет.

Параметры даты и времени всегда передаются в UTC.

<code>Guid[] IDs</code>	Массив ключей событий, которые будут отображены в отчет. Если задан, все остальные параметры игнорируются.
<code>Guid[] ParentEventID</code>	Массив ключей «первичных» событий. В отчет будут отображены все события, связанные с первичными событиями. Если задан, все параметры (кроме <code>IDs</code> ) игнорируются.  Значение <code>ParentEventId={Guid.Empty}</code> интерпретируется так же, как значение <code>{null}</code> (для совместимости с Delphi).
<code>DateTime StartDate</code>	Дата начала временного периода, за который создается отчет.
<code>DateTime EndDate</code>	Дата окончания временного периода, за который создается отчет.
<code>DateTime StartTime0</code>	Время начала часового диапазона внутри временного периода. В отчет будут отображены только события, произошедшие в течение этого диапазона.
<code>DateTime EndTime0</code>	Время окончания часового диапазона внутри временного периода. В отчет будут отображены

	только события, произошедшие в течение этого диапазона.
<code>DateTime StartTime1</code>	Время начала второго часового диапазона внутри временного периода. Задействуется, если в диапазон попадает окончание суток.*
<code>DateTime EndTime1</code>	Время окончания второго часового диапазона внутри временного периода. Задействуется, если в диапазон попадает окончание суток.*
<code>Guid[] Territories</code>	Массив ключей территорий, по которым будет сформирован отчет.
<code>Guid[] Operators</code>	Параметр предназначен для внутреннего использования.
<code>UInt[] TransactionTypes</code>	Массив ключей типов транзакций. В качестве параметра можно использовать один или несколько ключей транзакций из приведенной ниже таблицы.
<code>Guid[] Organizations</code>	Массив ключей организаций, по которым будет сформирован отчет.
<code>Guid[] Users</code>	Массив ключей пользователей, по событиям которых будет сформирован отчет.
<code>bool EventsWithoutUser</code>	Параметр предназначен для внутреннего использования.
<code>int MaxResultSize</code>	Максимальное количество событий, отображенных в отчете. При передаче null отображается количество событий по умолчанию (5000 шт.).

\* Если в часовой диапазон попадает момент окончания суток (24.00), то такой диапазон нужно разбить на два: с XX часов по 24.00 и с 00.00 часов до YY. При этом для первой части диапазона используются параметры `StartTime0` и `EndTime0`, а для второй части - `StartTime1` и `EndTime1`.

Ключи транзакций:

Ключ	Описание транзакции
590144	Нормальный вход по ключу
590152	Фактический вход
590145	Нормальный выход по ключу
590153	Фактический выход
590146	Выход вне временного профиля
590244	Нормальный выход посетителя
590245	Фактический выход посетителя

## HardwareState

Базовый класс: [BaseObject](#).

Класс, используемый для описания состояния территории.

<code>Guid</code> TerritoryID	Уникальный ключ территории.
<code>ulong</code> State	Набор состояний территории – битовая маска размером 4 байта.

### 3. ФУНКЦИИ

#### ВЕРСИЯ

##### Функция **GetVersion**

`string` GetVersion()

Параметры: без параметров.

Результат: Возвращает версию сервиса интеграции.

Описание: Функция возвращает версию сервиса интеграции.

#### СЕССИЯ

##### Функция **GetDomains**

`Domain[]` GetDomains()

Параметры: Нет

Результат: Возвращает массив [Domain](#).

Описание: Возвращает массив организаций.

##### Функция **OpenSession**

`SessionResult` OpenSession( `string` domain, `string` userName, `string` password )

Параметры:

<code>string</code> domain	Название организации для входа. Для входа в системную организацию можно использовать пустую строку.
<code>string</code> userName	Имя оператора.
<code>string</code> password	Пароль оператора.

Результат: Возвращает класс.

Описание: Данная функция используется для аутентификации оператора в сервисе интеграции. Полученный ключ сессии используется в дальнейшем для выполнения всех операций. Каждая сессия открывается на 5 минут, при выполнении любой операции на сервере интеграции время сессии продлевается.

##### Функция **OpenSessionWithInLocale**

`SessionResult` OpenSessionWithInLocale( `string` domain, `string` userName, `string` password, `string` locale )

Параметры:

<code>string</code> domain	Название организации для входа. Для входа в системную организацию можно использовать пустую строку.
<code>string</code> userName	Имя оператора.
<code>string</code> password	Пароль оператора.
<code>string</code> locale	Требуемый язык. Возможны значения «ru-RU», «en-US» или «es-ES».

Результат: Возвращает класс.



Описание: Данная функция используется для аутентификации оператора в сервисе интеграции с указанием языка интерфейса. Полученный ключ сессии используется в дальнейшем для выполнения всех операций. Каждая сессия открывается на 5 минут, при выполнении любой операции на сервере интеграции время сессии продлевается.

### Функция ContinueSession

```
int ContinueSession( Guid sessionId )
```

Параметры:

`Guid sessionId`      Уникальный ключ сессии.

Результат: В качестве результата возвращается значение: 0 – если операция прошла успешно; 1 – если операция выполнена с ошибками.

Описание: Используется для продления сессии.

### Функция CloseSession

```
void CloseSession( Guid sessionId )
```

Параметры:

`Guid sessionId`      Уникальный ключ сессии.

Результат:

Описание: Используется для закрытия сессии.

### Функция CheckRole

```
BaseResult CheckRole( Guid sessionId, string roleName )
```

Параметры:

`Guid sessionId`      Уникальный ключ сессии.

`string roleName`      Наименование права на совершение операций.

Результаты: Возвращает [BaseResult](#).

Описание: Проверяет доступность действий для текущего оператора. В интеграционном сервисе используются следующие права:

"EmployeeReader"	Право получения сведений о персонале и структуре подразделений.
"EmployeeWriter"	Право на удаление/изменение сведений о персонале и структуре подразделений.
"AccessGroupReader"	Право получения сведений о группах доступа.
"AccessGroupWriter"	Право на изменение групп доступа.
"GuestReader"	Право получения сведений о посетителях.
"VisitorRequestCreator"	Право на создание, редактирование и удаление заявок и посетителей.
"VisitorPassDistributor"	Право на одобрение и закрытие заявок.
"TimesheetReader"	Право просмотра сведений о расписании.

## ОБОРУДОВАНИЕ

### Функция SendHardwareCommand

`BaseResult SendHardwareCommand( Guid sessionID, Guid territoryID, int command )`

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid territoryID</code>	Уникальный ключ территории.
<code>int command</code>	Код посылаемой команды.

Результат: Возвращает [BaseResult](#).

Описание: посылает команду устройству, представленному территорией. Коды команд и устройства описаны в таблице ниже.

Команды		Контрол. доступа Parsec	Охранная область AC-08	Охранный раздел Болид
Код	Описание			
1	Включить реле на вход* \ Открыть дверь	✓		
2	Включить реле на выход*	✓		
4	Закрыть дверь	✓		
8	Установить относительную блокировку	✓		
16	Снять относительную блокировку	✓		
32	Установить абсолютную блокировку	✓		
64	Снять абсолютную блокировку	✓		
128	Установить на охрану	✓	✓	✓
256	Снять с охраны	✓	✓	✓
512	Включить доп. реле	✓		
1024	Выключить доп. реле	✓		
2048	Сброс антипассбека	✓		
* Команды работают только для контроллеров в режиме турникета.				

### Функция GetHardwareState

`HardwareState[] GetHardwareState( Guid sessionID, Guid[] territoryIDs )`

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid[] territoryIDs</code>	Массив ключей территорий.

Результат: Возвращает массив HardwareState – битовых масок (размером 4 байта).

Описание: Выдает набор состояний для выбранных территорий. Коды состояний описаны в таблицах ниже. Функция применима только к территориям, порождаемым контроллерами доступа серии NC, охранным контроллером AC-08 и охранной системой Болид.

### Биты состояния территории для контроллеров серии NC

№ бита	Значение	Состояние при значении бита	
		0	1
0	Аккумулятор	Разряжен	Норма
1	Сетевое питание	Отключено	Норма
2	Батарея	Неисправна	Норма
3	Корпус	Открыт	Закрит
4	Реле на вход* \ Замок	Выключено* \ Закрит	Включено* \ Открыт
5	Реле на выход*	Выключено*	Включено*
6	Доп. реле	Выключено	Включено
7	Реле картоприемника	Выключено	Включено
8	Абсолютная блокировка	Выключена	Включена
9	Относительная блокировка	Выключена	Включена
10	Экстренное открывание двери	Выключено	Включено
11	Охрана	Снята	На охране
12	Охранный датчик	Норма	Активирован
13	Датчик входа** \ Дверной контакт	Норма	Активирован
14	Датчик выхода**	Норма	Активирован
28	Выключен	Нет	Да
31	Недоступен	Нет	Да
* Значения указаны для контроллеров в режиме турникета. **Значения указаны только для контроллера NC-100K.			

### Биты состояния территории-охранной области контроллера AC-08 и охранной системы Bolid

№ бита	Значение	Состояние при значении бита	
		0	1
0	Аккумулятор*	Разряжен	Норма
1	Сетевое питание*	Отключено	Норма
3	Корпус*	Открыт	Закрит
11	Охрана	Снята	На охране
12	Охранный датчик	Норма	Активирован
24	Требуется внимания	Нет	Да

28	Выключен	Нет	Да
31	Недоступен	Нет	Да
* Значения указаны для контроллера AC-08.			

## ПОДРАЗДЕЛЕНИЯ

### Функция GetRootOrgUnit

`OrgUnit GetRootOrgUnit( Guid sessionId )`

Параметры:

`Guid sessionId`      Уникальный ключ сессии.

Результат: Возвращает корневую OrgUnit.

Описание: данная функция предназначена для получения корневого объекта дерева подразделений.

### Функция GetOrgUnitsHierarchy

`OrgUnit[] GetOrgUnitsHierarchy( Guid sessionId )`

Параметры:

`Guid sessionId`      Уникальный ключ сессии.

Результат: Массив OrgUnit.

Описание: Возвращает полную иерархию подразделений.

### Функция GetOrgUnitsHierarchyWithPersons

`BaseObject[] GetOrgUnitsHierarchyWithPersons( Guid sessionId )`

Параметры:

`Guid sessionId`      Уникальный ключ сессии.

Результат: Массив BaseObject, элементы массива могут быть OrgUnit или Person.

Описание: Возвращает полную иерархию подразделений вместе с сотрудниками.

### Функция GetOrgUnitsHierarchyWithVisitors

`BaseObject[] GetOrgUnitsHierarchyWithVisitors( Guid sessionId )`

Параметры:

`Guid sessionId`      Уникальный ключ сессии.

Результат: Массив BaseObject, элементы массива могут быть OrgUnit или Person.

Описание: Возвращает полную иерархию подразделений вместе с посетителями.

### Функция GetOrgUnitsHierarchyWithVehicle

`BaseObject[] GetOrgUnitsHierarchyWithVehicle( Guid sessionId )`

Параметры:

`Guid sessionID`      Уникальный ключ сессии.

Результат: Массив `BaseObject`, элементы массива могут быть `OrgUnit` или `Person`.

Описание: Возвращает полную иерархию подразделений вместе с автомобилями.

### Функция `GetOrgUnitSubItems`

```
BaseObject[] GetOrgUnitSubItems( Guid sessionID, Guid orgUnitID )
```

Параметры:

`Guid sessionID`      Уникальный ключ сессии.

`Guid orgUnitID`      Уникальный ключ подразделения.

Результат: Массив `BaseObject`, элементы массива могут быть `BaseOrgUnit` или `BasePerson`.

Описание: Возвращает массив подразделений и сотрудников, принадлежащих подразделению с указанным ключом.

### Функция `GetOrgUnitSubItemsHierarchyWithPersons`

```
BaseObject[] GetOrgUnitSubItemsHierarchyWithPersons( Guid sessionID, Guid orgUnitID )
```

Параметры:

`Guid sessionID`      Уникальный ключ сессии.

`Guid orgUnitID`      Уникальный ключ подразделения.

Результат: Массив `BaseObject`, элементы массива могут быть `OrgUnit` или `Person`.

Описание: Возвращает полную иерархию подразделений и их сотрудников, начиная с подразделения с указанным ключом.

### Функция `GetOrgUnitSubItemsHierarchyWithVisitors`

```
BaseObject[] GetOrgUnitSubItemsHierarchyWithVisitors( Guid sessionID, Guid orgUnitID )
```

Параметры:

`Guid sessionID`      Уникальный ключ сессии.

`Guid orgUnitID`      Уникальный ключ подразделения.

Результат: Массив `BaseObject`, элементы массива могут быть `OrgUnit` или `Person`.

Описание: Возвращает полную иерархию подразделений, начиная с подразделения с указанным ключом, и посетителей этих подразделений.

### Функция `GetOrgUnitSubItemsHierarchyWithVehicle`

```
BaseObject[] GetOrgUnitSubItemsHierarchyWithVehicle( Guid sessionID, Guid orgUnitID )
```

Параметры:

`Guid sessionID`      Уникальный ключ сессии.

`Guid orgUnitID`      Уникальный ключ подразделения.

Результат: Массив BaseObject, элементы массива могут быть OrgUnit или Person.

Описание: Возвращает полную иерархию подразделений, начиная с подразделения с указанным ключом, и автомобилей этих подразделений.

### Функция GetOrgUnit

```
OrgUnit GetOrgUnit( Guid sessionId, Guid orgUnitID )
```

Параметры:

`Guid sessionId`            Уникальный ключ сессии.

`Guid orgUnitID`            Уникальный ключ подразделения.

Результат: Возвращает OrgUnit для подразделения с указанным ключом или null, если указанное подразделение не найдено.

Описание: Возвращает информацию о подразделении с указанным ключом.

## РЕДАКТИРОВАНИЕ ПОДРАЗДЕЛЕНИЯ

### Функция CreateOrgUnit

```
GuidResult CreateOrgUnit( Guid sessionId, OrgUnit orgUnit )
```

Параметры:

`Guid sessionId`            Уникальный ключ сессии.

`OrgUnit orgUnit`            Параметры подразделения.

Результат: Возвращает GuidResult.

Описание: Создает подразделение с указанными данными. Возвращает ключ вновь созданного подразделения.

### Функция OpenOrgUnitEditingSession

```
GuidResult OpenOrgUnitEditingSession( Guid sessionId, Guid orgUnitID )
```

Параметры:

`Guid sessionId`            Уникальный ключ сессии.

`Guid orgUnitID`            Уникальный ключ подразделения.

Результат: Возвращает GuidResult.

Описание: Открывает сессию редактирования подразделения. Возвращает ключ вновь созданной сессии редактирования подразделения.

### Функция CloseOrgUnitEditingSession

```
void CloseOrgUnitEditingSession( Guid orgUnitEditSessionID )
```

Параметры:

`Guid orgUnitEditSessionID`    Уникальный ключ сессии редактирования подразделения.

Результат: Пустой.

Описание: Закрывает сессию редактирования подразделения.

## Функция SaveOrgUnit

`BaseResult SaveOrgUnit( Guid orgUnitEditSessionID, BaseOrgUnit orgUnit )`

Параметры:

<code>Guid orgUnitEditSessionID</code>	Уникальный ключ сессии редактирования подразделения.
<code>BaseOrgUnit orgUnit</code>	Параметры подразделения.

Результат: Возвращает BaseResult.

Описание: Изменяет параметры подразделения. Параметр orgUnit может быть BaseOrgUnit или OrgUnit. После удачного сохранения подразделения сессия редактирования закрывается.

## Функция DeleteOrgUnit

`BaseResult DeleteOrgUnit( Guid sessionID, Guid orgUnitID )`

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid orgUnitID</code>	Уникальный ключ подразделения.

Результат: Возвращает BaseResult.

Описание: Удаляет подразделение с указанным ключом.

# ПЕРСОНАЛ

## Функция GetPersonExtraFieldTemplates

`PersonExtraFieldTemplate[] GetPersonExtraFieldTemplates( Guid sessionID )`

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

Результат: Возвращает массив PersonExtraFieldTemplate.

Описание: Возвращает набор шаблонов дополнительных полей у сотрудников.

## Функция GetVisitorExtraFieldTemplates

`PersonExtraFieldTemplate [] GetVisitorExtraFieldTemplates( Guid sessionID )`

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

Результат: Возвращает массив PersonExtraFieldTemplate.

Описание: Возвращает набор шаблонов дополнительных полей у посетителей.

## Функция GetVehicleExtraFieldTemplates

`PersonExtraFieldTemplate [] GetVehicleExtraFieldTemplates( Guid sessionID )`

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

Результат: Возвращает массив PersonExtraFieldTemplate.

Описание: Возвращает набор шаблонов дополнительных полей у автомобилей.

## Функция FindPeople

```
BasePerson[] FindPeople( Guid sessionId, string lastname, string  
firstname, string middlename )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>string lastname</code>	Значение фамилии для поиска.
<code>string firstname</code>	Значение имени для поиска.
<code>string middlename</code>	Значение отчества для поиска.

Результат: Возвращает массив Person.

Описание: Выдает набор сотрудников, подходящих под переданные критерии. Поиск может производиться как по любому параметру отдельно, так и по всем параметрам. Должен быть указан хотя бы один параметр.

## Функция FindVisitors

```
BasePerson[] FindVisitors( Guid sessionId, string lastname, string  
firstname, string middlename )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>string lastname</code>	Значение фамилии для поиска.
<code>string firstname</code>	Значение имени для поиска.
<code>string middlename</code>	Значение отчества для поиска.

Результат: Возвращает массив Person.

Описание: Выдает набор посетителей, подходящих под переданные критерии. Поиск может производиться как по любому параметру отдельно, так и по всем параметрам. Должен быть указан хотя бы один параметр.

## Функция FindVehicle

```
BasePerson[] FindVehicle( Guid sessionId, string number, string model,  
string color )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>string number</code>	Значение номера автомобиля для поиска.
<code>string model</code>	Значение модели автомобиля для поиска.
<code>string color</code>	Значение цвета для поиска.

Результат: Возвращает массив Person.

Описание: Выдает набор автомобилей, подходящих под переданные критерии. Поиск может производиться как по любому параметру отдельно, так и по всем параметрам. Должен быть указан хотя бы один параметр.



## Функция FindPersonByIdentifier

`Person` FindPersonByIdentifier( `Guid` sessionId, `string` cardCode )

Параметры:

<code>Guid</code> sessionId	Уникальный ключ сессии.
<code>string</code> cardCode	Код идентификатора карты в 16-ричном формате в верхнем регистре (Например: A12345BCF).

Результат: Возвращает `Person`.

Описание: Возвращает информацию о субъекте доступа с указанным кодом идентификатора.

## Функция GetPerson

`PersonWithPhoto` GetPerson( `Guid` sessionId, `Guid` personID )

Параметры:

<code>Guid</code> sessionId	Уникальный ключ сессии.
<code>Guid</code> personID	Уникальный ключ субъекта доступа.

Результат: Возвращает `PersonWithPhoto`.

Описание: Возвращает информацию о субъекте доступа с указанным ключом.

## Функция GetPersonsChangedAfter

`Person[]` GetPersonsChangedAfter( `Guid` sessionId, `Guid` orgID, `DateTime` dateFrom, `bool` includeSubOrg )

Параметры:

<code>Guid</code> sessionId	Уникальный ключ сессии.
<code>Guid</code> orgID	Уникальный ключ подразделения.
<code>DateTime</code> dateFrom	Начальная дата для анализа.
<code>bool</code> includeSubOrg	Признак поиска по вложенным подразделениям.

Результат: Возвращает массив `Person`.

Описание: Возвращает набор субъектов доступа (сотрудник, посетитель, автомобиль), чьи данные были изменены, начиная с указанной даты. Поиск производится либо в указанном подразделении, либо в указанном и во всех вложенных.

## Функция GetPersonExtraFieldValue

`ObjectResult` GetPersonExtraFieldValue( `Guid` sessionId, `Guid` personID, `Guid` templateID )

Параметры:

<code>Guid</code> sessionId	Уникальный ключ сессии.
<code>Guid</code> personID	Уникальный ключ субъекта доступа.

`Guid` templateID                      Уникальный ключ шаблона.

Результат: Возвращает ObjectResult.

Описание: Возвращает значение указанного дополнительного поля субъекта доступа. Тип значения определяется типом поля.

### Функция GetPersonExtraFieldValues

```
ExtraFieldValue[] GetPersonExtraFieldValues( Guid sessionID, Guid  
personID )
```

Параметры:

`Guid` sessionID                      Уникальный ключ сессии.  
`Guid` personID                      Уникальный ключ субъекта доступа.

Результат: Возвращает массив ExtraFieldValue.

Описание: Возвращает массив значений дополнительных полей субъекта доступа.

### Функция GetPersonExtraFieldValueString

```
StringResult GetPersonExtraFieldValueString( Guid sessionID, Guid  
personID, Guid templateID )
```

Параметры:

`Guid` sessionID                      Уникальный ключ сессии.  
`Guid` personID                      Уникальный ключ субъекта доступа.  
`Guid` templateID                      Уникальный ключ шаблона.

Результат: Возвращает StringResult.

Описание: Возвращает значение дополнительного поля субъекта доступа в виде форматированной строки.

### Функция ValidateExtraFieldValue

```
BaseResult ValidateExtraFieldValue( Guid sessionID, ExtraFieldValue  
value )
```

Параметры:

`Guid` sessionID                      Уникальный ключ сессии.  
`ExtraFieldValue` value              Значение дополнительного поля.

Результат: Возвращает BaseResult.

Описание: Проверяет валидность значения дополнительного поля.

### Функция GetPersonScheduleFixes

```
PersonScheduleFix[] GetPersonScheduleFixes( Guid sessionID, Guid  
personID )
```

Параметры:

`Guid` sessionID                      Уникальный ключ сессии.  
`Guid` personID                      Уникальный ключ сотрудника.

Результат: Возвращает массив PersonScheduleFix.

Описание: Возвращает список существующих поправок к рабочему времени сотрудника.

### Функция AddPersonScheduleFix

```
GuidResult AddPersonScheduleFix( Guid sessionId, PersonScheduleFix fix )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>PersonScheduleFix fix</code>	Параметры поправки рабочего времени.

Результат: Возвращает GuidResult.

Описание: Добавляет поправку рабочего времени сотрудника.

### Функция SavePersonScheduleFix

```
BaseResult SavePersonScheduleFix( Guid sessionId, PersonScheduleFix fix )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>PersonScheduleFix fix</code>	Параметры поправки рабочего времени.

Результат: Возвращает BaseResult.

Описание: Изменяет параметры существующей поправки рабочего времени сотрудника.

### Функция DeletePersonScheduleFix

```
BaseResult DeletePersonScheduleFix( Guid sessionId, Guid personID, Guid fixID )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>Guid personID</code>	Уникальный ключ сотрудника.
<code>Guid fixID</code>	Уникальный ключ поправки.

Результат: Возвращает BaseResult.

Описание: Удаляет поправку к рабочему времени сотрудника.

## РЕДАКТИРОВАНИЕ СУБЪЕКТА ДОСТУПА

### Функция CreatePerson

```
GuidResult CreatePerson( Guid sessionId, Person person )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>Person person</code>	Данные сотрудника.

Результат: Возвращает GuidResult.

Описание: Создает сотрудника с указанными данными. В качестве параметра person может быть Person или PersonWithPhoto. Возвращает ключ вновь созданного сотрудника. Если поле ID в структуре person равен Guid.Empty (00000000-0000-0000-0000-000000000000), то ID генерируется автоматически и возвращается. Иначе используется заданное значение ID.

### Функция CreateVisitor

```
GuidResult CreateVisitor( Guid sessionId, Person person )
```

Параметры:

Guid sessionId	Уникальный ключ сессии.
Person person	Данные посетителя.

Результат: Возвращает GuidResult.

Описание: Создает посетителя с указанными данными. В качестве параметра person может быть Person или PersonWithPhoto. Возвращает ключ вновь созданного посетителя. Если поле ID в структуре person равен Guid.Empty (00000000-0000-0000-0000-000000000000), то ID генерируется автоматически и возвращается. Иначе используется заданное значение ID.

### Функция CreateVehicle

```
GuidResult CreateVehicle( Guid sessionId, Person person )
```

Параметры:

Guid sessionId	Уникальный ключ сессии.
Person person	Данные автомобиля.

Результат: Возвращает GuidResult.

Описание: Создает автомобиль с указанными данными. В качестве параметра person может быть Person или PersonWithPhoto. Возвращает ключ вновь созданного автомобиля. Если поле ID в структуре person равен Guid.Empty (00000000-0000-0000-0000-000000000000), то ID генерируется автоматически и возвращается. Иначе используется заданное значение ID.

### Функция OpenPersonEditingSession

```
GuidResult OpenPersonEditingSession( Guid sessionId, Guid personID )
```

Параметры:

Guid sessionId	Уникальный ключ сессии.
Guid personID	Уникальный ключ субъекта доступа.

Результат: Возвращает GuidResult.

Описание: Открывает сессию редактирования субъекта доступа. Возвращает ключ вновь созданной сессии.

### Функция ClosePersonEditingSession

```
void ClosePersonEditingSession( Guid personEditSessionID )
```

Параметры:

<code>Guid</code> <code>personEditSessionID</code>	Уникальный ключ сессии редактирования субъекта доступа.
---	---

Результат: Пусто.

Описание: Закрывает сессию редактирования субъекта доступа.

## Функция **SavePerson**

```
BaseResult SavePerson( Guid personEditSessionID, BasePerson person )
```

Параметры:

<code>Guid</code> <code>personEditSessionID</code>	Уникальный ключ сессии редактирования субъекта доступа.
<code>BasePerson</code> <code>person</code>	Данные субъекта доступа.

Результат: Возвращает `BaseResult`.

Описание: Изменяет данные субъекта доступа. В качестве параметра `person` может служить `BasePerson`, `Person`, `PersonWithPhoto`. В случае удачного сохранения сессия редактирования субъекта доступа закрывается.

## Функция **SetPersonPhoto**

```
BaseResult SetPersonPhoto( Guid personEditSessionID, byte[] photoByteArray )
```

Параметры:

<code>Guid</code> <code>personEditSessionID</code>	Уникальный ключ сессии редактирования субъекта доступа.
<code>byte[]</code> <code>photoByteArray</code>	Фото субъекта доступа.

Результат: Возвращает `BaseResult`.

Описание: Сохраняет фото субъекта доступа. В случае удачного сохранения сессия редактирования субъекта доступа закрывается.

## Функция **SetPersonOrgUnit**

```
BaseResult SetPersonOrgUnit( Guid personEditSessionID, Guid orgUnitID )
```

Параметры:

<code>Guid</code> <code>personEditSessionID</code>	Уникальный ключ сессии редактирования субъекта доступа.
<code>Guid</code> <code>orgUnitID</code>	Уникальный ключ подразделения.

Результат: Возвращает `BaseResult`.

Описание: Устанавливает подразделение субъекта доступа. В случае удачного сохранения сессия редактирования субъекта доступа закрывается.

## Функция **SetPersonExtraFieldValue**

```
BaseResult SetPersonExtraFieldValue( Guid personEditSessionID, Guid templateID, object value )
```

Параметры:

<code>Guid</code> <code>personEditSessionID</code>	Уникальный ключ сессии редактирования субъекта доступа.
<code>Guid</code> <code>templateID</code>	Уникальный ключ шаблона дополнительного поля.
<code>object</code> <code>value</code>	Значение дополнительного поля.

Результат: Возвращает `BaseResult`.

Описание: Устанавливает новое значение указанного дополнительного поля субъекта доступа.

## Функция `SetPersonExtraFieldValues`

```
BaseResult SetPersonExtraFieldValues( Guid personEditSessionID,  
ExtraFieldValue[] values )
```

Параметры:

<code>Guid</code> <code>personEditSessionID</code>	Уникальный ключ сессии редактирования субъекта доступа.
<code>ExtraFieldValue</code> [] <code>values</code>	Массив значений дополнительных полей.

Результат: Возвращает `BaseResult`.

Описание: Устанавливает значения для дополнительных полей.

## Функция `DeletePerson`

```
BaseResult DeletePerson( Guid sessionID, Guid personID )
```

Параметры:

<code>Guid</code> <code>sessionID</code>	Уникальный ключ сессии.
<code>Guid</code> <code>personID</code>	Уникальный ключ субъекта доступа.

Результат: Возвращает `BaseResult`.

Описание: Удаляет субъект доступа с указанным ключом.

## Функция `BlockPerson`

```
BaseResult BlockPerson( Guid personEditSessionID, int typeBlock )
```

Параметры:

<code>Guid</code> <code>personEditSessionID</code>	Уникальный ключ сессии редактирования данных субъекта доступа.
<code>int</code> <code>typeBlock</code>	Тип блокировки.

Результат: Возвращает `BaseResult`.

Описание: Блокирует субъект доступа в соответствии с указанным типом блокировки.

Значения типа блокировки описаны в таблице:

Значение <code>typeBlock</code>	Описание
1	Установка привилегии «Выход запрещен».

2	Установка привилегии «Вход запрещен».
3	Установка привилегии «Вход запрещен» и «Выход запрещен».

В случае удачного выполнения функции сессия редактирования субъекта доступа закрывается.

## Функция UnblockPerson

`BaseResult UnblockPerson( Guid personEditSessionID )`

Параметры:

<code>Guid</code>	Уникальный ключ сессии редактирования
<code>personEditSessionID</code>	данных субъекта доступа.

Результат: Возвращает `BaseResult`.

Описание: Снимает все блокировки с субъекта доступа, заблокированного ранее.

В случае удачного выполнения функции сессия редактирования субъекта доступа закрывается.

## ТОПОЛОГИЯ

### Функция GetRootTerritory

`Territory GetRootTerritory( Guid sessionID )`

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

Результат: Возвращает корневую `Territory`.

Описание: Данная функция предназначена для получения корневого объекта дерева территорий.

### Функция GetTerritoriesHierarchy

`Territory[] GetTerritoriesHierarchy( Guid sessionID )`

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

Результат: Возвращает массив `Territory`, значения могут быть `TerritoryWithComponent`.

Описание: Возвращает полную иерархию территорий.

### Функция GetTerritorySubItems

`BaseTerritory[] GetTerritorySubItems( Guid sessionID, Guid TerraID )`

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid TerraID</code>	Уникальный ключ территории.

Результат: Возвращает массив `BaseTerritory`, значения могут быть `TerritoryWithComponent`.

Описание: Возвращает список территорий, принадлежащих территории с указанным ключом.

## Функция GetTerritory

`Territory` GetTerritory( `Guid` sessionId, `Guid` territoryID )

Параметры:

<code>Guid</code> sessionId	Уникальный ключ открытой сессии.
<code>Guid</code> territoryID	Уникальный ключ территории.

Результат: Возвращает Territory, значение может быть TerritoryWithComponent.

Описание: Данная функция предназначена для получения описания территории по ее ключу.

## ИДЕНТИФИКАТОРЫ И ДОСТУП

### Функция GetPersonIdentifiers

`Identifier[]` GetPersonIdentifiers( `Guid` sessionId, `Guid` personID )

Параметры:

<code>Guid</code> sessionId	Уникальный ключ сессии.
<code>Guid</code> personID	Уникальный ключ субъекта доступа.

Результат: Массив Identifier, элементы массива могут быть Identifier или IdentifierTemp.

Описание: Возвращает массив идентификаторов для указанного субъекта доступа.

### Функция DeleteIdentifier

`BaseResult` DeleteIdentifier( `Guid` sessionId, `string` Code )

Параметры:

<code>Guid</code> sessionId	Уникальный ключ сессии.
<code>string</code> Code	Код идентификатора.

Результат: Возвращает BaseResult.

Описание: Удаляет идентификатор с указанным кодом.

### Функция AddPersonIdentifier

`BaseResult` AddPersonIdentifier( `Guid` personEditSessionID, `BaseIdentifier` identifier )

Параметры:

<code>Guid</code> personEditSessionID	Уникальный ключ сессии редактирования субъекта доступа.
<code>BaseIdentifier</code> identifier	Параметры идентификатора.

Результат: Возвращает BaseResult.

Описание: Добавляет идентификатор субъекту доступа или изменяет данные уже имеющегося идентификатора. В качестве параметра identifier могут служить BaseIdentifier, Identifier, IdentifierTemp. Элемент IS\_PRIMARY используется только в позитивном смысле, т.е. при установке этого признака,



соответствующий идентификатор становится первичным, а при снятии этого признака у первичного идентификатора при сохранении в смысле первичности идентификатора ничего не меняется. Элемент PRIVILEGE\_MASK игнорируется.

## Функция ChangePersonIdentifier

```
BaseResult ChangePersonIdentifier( Guid personEditSessionID,  
BaseIdentifier identifier )
```

Параметры:

Guid personEditSessionID	Уникальный ключ сессии редактирования субъекта доступа.
BaseIdentifier identifier	Параметры идентификатора.

Результат: Возвращает BaseResult.

Описание: Изменяет параметры существующего идентификатора. В качестве параметра identifier могут служить BaseIdentifier, Identifier, IdentifierTemp. Элемент IS\_PRIMARY используется только в позитивном смысле, т.е. при установке этого признака, соответствующий идентификатор становится первичным, а при снятии этого признака у первичного идентификатора при сохранении в смысле первичности идентификатора ничего не меняется. Элемент PRIVILEGE\_MASK игнорируется.

## Функция SetIdentifierPrivileges

```
BaseResult SetIdentifierPrivileges( Guid sessionID, string cardCode,  
long privilegesMask )
```

Параметры:

Guid sessionID	Уникальный ключ сессии.
string cardCode	Код идентификатора.
long privilegesMask	Битовая маска привилегий.

Результат: Возвращает BaseResult.

Описание: Функция устанавливает набор привилегий с помощью битовой маски. Значения битов маски описаны в таблице:

№ бита	Описание
0	Выключение звука двери.
1	Управление охраной.
2	Проход при блокировке.
3	Прием тревоги.
4	Постановка на охрану.
5	Снятие с охраны.
6	Проход при антипассбеке.
7	Гостевая карта.
8	Карта с привилегиями.
9	Выход запрещен.

10	Выход вне временного профиля разрешен.
11	Управление доступом.
12	-
13	Карта владельца.
14	Не использовать счетчик проходов.
15	Вход запрещен.
16	Проход без сопровождения запрещен.

*Пример:* для установки идентификатору привилегий «Проход при блокировке» и «Проход при антипассбеке» значение параметра *privilegesMask* должно быть равно  $2^2 + 2^6 = 4 + 64 = 68$   $_{10} = 1000100_2$

## РАСПИСАНИЯ И ГРУППЫ ДОСТУПА

### Функция **GetAccessSchedules**

```
Schedule[] GetAccessSchedules( Guid sessionId )
```

Параметры:

`Guid sessionId`                      Уникальный ключ сессии.

Результат: Возвращает массив `Schedule`, значение может быть `AccessSchedule`.

Описание: Выдает набор расписаний доступа системы

### Функция **GetWorktimeSchedules**

```
Schedule[] GetWorktimeSchedules( Guid sessionId )
```

Параметры:

`Guid sessionId`                      Уникальный ключ сессии.

Результат: Возвращает массив `Schedule`, значение может быть `WorktimeSchedule`.

Описание: Выдает набор расписаний рабочего времени.

### Функция **GetScheduleIntervals**

```
TimeInterval[] GetScheduleIntervals( Guid sessionId, Guid scheduleID,
DateTime from, DateTime to )
```

Параметры:

`Guid sessionId`                      Уникальный ключ сессии.

`Guid scheduleID`                    Уникальный ключ расписания.

`DateTime from`                      Начальная дата для анализа.

`DateTime to`                          Конечная дата для анализа.

Результат: Возвращает массив `TimeInterval`, значения могут быть `WorktimeInterval`.

Описание: Выдает набор интервалов указанного расписания, ограниченный начальной и конечной датой. Функцией принимается во внимание только DATE часть параметров.

## Функция CreateAccessSchedule

```
GuidResult CreateAccessSchedule( Guid sessionID, AccessSchedule  
schedule, ScheduleDay[] days )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>AccessSchedule schedule</code>	Параметры расписания.
<code>ScheduleDay[] days</code>	Массив шаблонов дней цикла расписания.

Результат: Возвращает GuidResult.

Описание: Создает новое расписание доступа в системе с указанными параметрами и заданным циклом. Возвращает ключ вновь созданного расписания доступа.

*При создании расписания необходимо строго соблюдать следующие условия:*

*Для недельного расписания доступа обязателен тип применения праздников «Применять с заменой».*

*Для недельного расписания рабочего времени не может быть установлен тип применения праздников «Применять со вставкой».*

*Для недельных расписаний массив шаблонов дней days должен содержать 7 элементов.*

*Элементы массива days должны иметь индексы дня в цикле расписания. Индекс первого дня в цикле имеет значение «1». Все элементы цикла должны иметь одинаковую дату.*

*Для недельных расписаний дата начала цикла должна приходиться на понедельник.*

*В цикле недельного расписания доступа не должно быть более трех уникальных шаблонов дней, один из которых – выходной (не содержит временных интервалов).*

## Функция CreateWorktimeSchedule

```
GuidResult CreateWorktimeSchedule( Guid sessionID, WorktimeSchedule  
schedule, ScheduleDay[] days )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>WorktimeSchedule schedule</code>	Параметры расписания.
<code>ScheduleDay[] days</code>	Массив шаблонов дней цикла расписания.

Результат: Возвращает GuidResult.

Описание: Создает новое расписание рабочего времени в системе с указанными параметрами и заданным циклом. Возвращает ключ вновь созданного расписания рабочего времени. При создании необходимо соблюдать правила, описанные в разделе функции [CreateAccessSchedule](#).

## Функция GetSchedule

```
Shedule GetSchedule ( Guid sessionID, Guid scheduleID )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid scheduleID</code>	Уникальный ключ расписания.

Результат: Возвращает Schedule, результат может быть AccessSchedule или WorktimeSchedule.

Описание: Возвращает информацию о расписании с указанным ключом.

## Функция SaveSchedule

```
BaseResult SaveSchedule( Guid sessionID, Schedule schedule )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Schedule schedule</code>	Параметры расписания.

Результат: Возвращает BaseResult.

Описание: Изменяет параметры расписания. Параметр schedule может быть AccessSchedule или WorktimeSchedule.

## Функция DeleteSchedule

```
BaseResult DeleteSchedule( Guid sessionID, Guid scheduleID )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid scheduleID</code>	Уникальный ключ расписания, которое нужно удалить.

Результат: Возвращает BaseResult.

Описание: Удаляет расписание с указанным ключом.

## Функция GetScheduleDetails

```
ScheduleDay[] GetScheduleDetails ( Guid sessionID, Guid scheduleID )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid scheduleID</code>	Уникальный ключ расписания.

Результат: Возвращает массив ScheduleDay. Массив может содержать ScheduleFix.

Описание: Возвращает информацию о всех шаблонах дней и днях-поправках, содержащихся в расписании с указанным ключом.

## Функция SetScheduleDays

```
BaseResult SetScheduleDays ( Guid sessionID, Guid scheduleID, ScheduleDay[] days )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid scheduleID</code>	Уникальный ключ расписания.
<code>ScheduleDay[] days</code>	Массив шаблонов дней цикла расписания.

Результат: Возвращает `BaseResult`.

Описание: Добавляет новый или изменяет существующий цикл (при совпадении даты начала цикла) расписания с указанным ключом. Параметр `days` должен соответствовать правилам, описанным в разделе функции `CreateAccessSchedule`.

## Функция `SetScheduleFix`

```
BaseResult SetScheduleFix ( Guid sessionID, Guid scheduleID, ScheduleFix[] fixes )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid scheduleID</code>	Уникальный ключ расписания.
<code>ScheduleFix[] fixes</code>	Массив дней-поправок расписания.

Результат: Возвращает `BaseResult`.

Описание: Добавляет или изменяет существующие дни-поправки (при совпадении даты) в расписании с указанным ключом. Не может быть применена для недельного расписания доступа.

## Функция `DeleteScheduleDays`

```
BaseResult DeleteScheduleDays ( Guid sessionID, Guid scheduleID, DateTime[] days )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid scheduleID</code>	Уникальный ключ расписания.
<code>DateTime[] days</code>	Массив дат.

Результат: Возвращает `BaseResult`.

Описание: Удаляет цикл расписания и дни-поправки, если дата в массиве `days` совпадает с датой начала цикла или датой поправки. Недопустимо удаление последнего цикла в расписании.

## Функция `GetHolidays`

```
Holiday[] GetHolidays ( Guid sessionID )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

Результат: Возвращает массив `holiday`.

Описание: Выдает список праздников, существующих в системе.

## Функция SetHolidays

`BaseResult SetHolidays( Guid sessionId, Holiday[] holidays )`

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>Holiday[] holidays</code>	Массив праздничных дней.

Результат: Возвращает `BaseResult`.

Описание: Добавляет или изменяет существующие (при совпадении дат) праздничные дни.

## Функция DeleteHolidays

`BaseResult DeleteHolidays ( Guid sessionId, Holiday[] holidays )`

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>Holiday[] holidays</code>	Массив праздничных дней для удаления.

Результат: Возвращает `BaseResult`.

Описание: Удаляет праздничные дни с указанными параметрами.

## Функция GetAccessGroups

`AccessGroup[] GetAccessGroups( Guid sessionId )`

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

Результат: Возвращает массив `AccessGroup`.

Описание: Возвращает массив доступных групп доступа.

## Функция CreateTempAccessGroup

`GuidResult CreateTempAccessGroup( Guid sessionId, Guid scheduleID, Guid[] territories )`

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>Guid scheduleID</code>	Уникальный ключ расписания.
<code>Guid[] territories</code>	Массив уникальных ключей территорий.

Результат: Возвращает `GuidResult`.

Описание: Создает временную группу доступа по заданному расписанию для доступа на заданные территории. Группа является временной и имеет ограниченное «время жизни». Недействительна без привязки к идентификатору сотрудника.

Для создания доступна только группа доступа типа «Подсистема доступа «Parsec».

## Функция CreateAccessGroup

```
GuidResult CreateAccessGroup( Guid sessionID, string groupName, Guid  
scheduleID, Guid[] territories )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>string groupName</code>	Наименование группы доступа. Должно быть уникальным.
<code>Guid scheduleID</code>	Уникальный ключ расписания.
<code>Guid[] territories</code>	Массив ключей территорий.

Результат: Возвращает GuidResult.

Описание: Создает группу доступа по указанному расписанию для доступа на выбранные территории. Группа доступа будет иметь указанное в функции имя.

Для создания доступна только группа доступа типа «Подсистема доступа «Parsec».

## Функция DeleteAccessGroup

```
BaseResult DeleteAccessGroup( Guid sessionID, Guid accessGroupID )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid accessGroupID</code>	Уникальный ключ группы доступа.

Результат: Возвращает BaseResult.

Описание: Удаляет группу доступа с указанным ключом.

Для удаления доступна только группа доступа типа «Подсистема доступа «Parsec».

## Функция AddSubAccessGroup

```
GuidResult AddSubAccessGroup( Guid sessionID, Guid accessGroupID, Guid  
scheduleID, Guid[] territories )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid accessGroupID</code>	Уникальный ключ группы доступа.
<code>Guid scheduleID</code>	Уникальный ключ расписания.
<code>Guid[] territories</code>	Массив ключей территорий.

Результат: Возвращает GuidResult.

Описание: Добавляет к существующей группе доступа группу компонент с указанным расписанием для доступа на выбранные территории.

Если указанная территория уже используется в какой-либо группе компонент (в пределах одной группы доступа), то она будет удалена из этой группы компонент.

Для изменения доступна только группа доступа типа «Подсистема доступа «Parsec».

## Функция DeleteSubAccessGroup

```
BaseResult DeleteSubAccessGroup( Guid sessionID, Guid accessGroupID, Guid subGroupID )
```

Параметры:

Guid sessionID	Уникальный ключ сессии.
Guid accessGroupID	Уникальный ключ группы доступа.
Guid subGroupID	Уникальный ключ группы компонент.

Результат: Возвращает BaseResult.

Описание: Удаляет группу компонент с указанным ключом.

Для изменения доступна только группа доступа типа «Подсистема доступа «Parsec».

## Функция GetSubAccessGroups

```
SubAccessGroup[] GetSubAccessGroups( Guid sessionID, Guid accessGroupID )
```

Параметры:

Guid sessionID	Уникальный ключ сессии.
Guid accessGroupID	Уникальный ключ группы доступа.

Результат: Возвращает массив SubAccessGroup.

Описание: Выдает список групп компонент (подгрупп) для указанной группы доступа.

## Функция GetInheritedAccessGroups

```
Guid[] GetInheritedAccessGroups( Guid sessionID, Guid accessGroupID )
```

Параметры:

Guid sessionID	Уникальный ключ сессии.
Guid accessGroupID	Уникальный ключ группы доступа.

Результат: Возвращает массив ключей групп доступа.

Описание: Выдает список ключей групп доступа, вложенных для указанной группы доступа. Ключи вложенных групп возвращаются в порядке убывания приоритета наследования.

## Функция SetInheritedAccessGroups

```
BaseResult SetInheritedAccessGroup( Guid sessionID, Guid accessGroupID, Guid[] inheritedGroups )
```

Параметры:

Guid sessionID	Уникальный ключ сессии.
Guid accessGroupID	Уникальный ключ группы доступа.
Guid[] inheritedGroups	Массив ключей вложенных групп доступа



Результат: Возвращает BaseResult.

Описание: Задаёт вложенные группы доступа для указанной группы. Приоритет наследования определяется порядком элементов в массиве.

Для отмены наследования необходимо передать пустой массив.

Для изменения доступна только группа доступа типа «Подсистема доступа «Parsec»».

## РАБОТА С ЗАЯВКАМИ БЮРО ПРОПУСКОВ

### Функция GetAcceptedVisitorRequests

```
VisitorRequest[] GetAcceptedVisitorRequests( Guid sessionId )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

Результат: Возвращает массив VisitorRequest.

Описание: Выдаёт набор заявок бюро пропусков со статусом «Одобрены».

### Функция FindVisitorRequest

```
VisitorRequest FindVisitorRequest( Guid sessionId, int requestNumber )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

<code>int requestNumber</code>	Номер заявки.
--------------------------------	---------------

Результат: Возвращает VisitorRequest, либо null, если заявка с указанным номером не найдена.

Описание: Выполняет поиск заявки с указанным номером среди заявок со статусом «Одобрена».

### Функция ActivateVisitorRequest

```
BaseResult ActivateVisitorRequest( Guid sessionId, Guid requestID, string cardCode )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

<code>Guid requestID</code>	Уникальный ключ заявки.
-----------------------------	-------------------------

<code>String cardCode</code>	Код идентификатора.
------------------------------	---------------------

Результат: Возвращает BaseResult.

Описание: Выдаёт посетителю идентификатор с указанным кодом из пула. Данная функция применима только к заявкам со статусом «Одобрена». Переводит заявку в состояние «Активная».

### Функция CreateVisitorRequest

```
VisitorRequest CreateVisitorRequest( Guid sessionId, VisitorRequest request )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

<code>VisitorRequest</code>	Параметры заявки.
<code>request</code>	

Результат: Возвращает `VisitorRequest`.

Описание: Создает заявку на посетителя и возвращает информацию о ней. В параметре обязательно должны быть заполнены `PERSON_ID`, `ORGUNIT_ID`, `ADMIT_START`, `ADMIT_END`. Все остальные параметры необязательны. Заявка создается со статусом «Ожидание согласования».

### Функция `GetVisitorRequest`

```
VisitorRequest GetVisitorRequest( Guid sessionId, Guid requestId )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>Guid requestId</code>	Уникальный ключ заявки.

Результат: Возвращает `VisitorRequest`.

Описание: Возвращает информацию о затребованной заявке

### Функция `SaveVisitorRequest`

```
BaseResult SaveVisitorRequest( Guid sessionId, VisitorRequest request )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>VisitorRequest request</code>	Параметры заявки.

Результат: Возвращает `BaseResult`.

Описание: Изменяет параметры в существующей заявке.

### Функция `DeleteIssuedVisitorRequest`

```
BaseResult DeleteIssuedVisitorRequest( Guid sessionId, Guid requestId )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>Guid requestId</code>	Уникальный ключ заявки.

Результат: Возвращает `BaseResult`.

Описание: Удаляет заявку на посетителя. Может быть удалена только заявка со статусом «Ожидание согласования».

### Функция `GetIssuedVisitorRequests`

```
VisitorRequest[] GetIssuedVisitorRequests( Guid sessionId )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
-----------------------------	-------------------------

Результат: Возвращает массив `VisitorRequest`.

Описание: Возвращает массив заявок, имеющих статус «Ожидание согласования». Если заявок не найдено, то возвращается пустой массив.

## Функция GetVisitorRequests

```
VisitorRequest[] GetVisitorRequests ( Guid sessionId, Guid orgUnitID,
DateTime from, bool issued, bool accepted, bool declined, bool active,
bool completed )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>Guid orgUnitID</code>	Уникальный ключ подразделения.
<code>DateTime from</code>	Начальная дата для отбора заявок.
<code>bool issued</code>	Заявки со статусом «Ожидание согласования».
<code>bool accepted</code>	Заявки со статусом «Одобрена».
<code>bool declined</code>	Заявки со статусом «Отклонена».
<code>bool active</code>	Заявки со статусом «Выдан пропуск».
<code>bool completed</code>	Заявки со статусом «Закрота».

Результат: Массив VisitorRequest.

Описание: Выдает все заявки в указанном подразделении, отобранные по дате и статусам. Функцией принимается во внимание только DATE часть параметра from.

## Функция CloseAllActiveVisitorRequests

```
BaseResult CloseAllActiveVisitorRequests( Guid sessionId, Guid
visitorID )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>Guid visitorID</code>	Уникальный ключ посетителя.

Результат: Возвращает BaseResult.

Описание: Ищет и закрывает все активные заявки, связанные с указанным посетителем, идентификатор после закрытия заявки отвязывается от посетителя и возвращается в пул карт бюро пропусков.

## Функция CloseVisitorRequest

```
BaseResult CloseVisitorRequest( Guid sessionId, Guid requestID )
```

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>Guid requestID</code>	Уникальный ключ заявки.

Результат: Возвращает BaseResult.

Описание: Закрывает заявку с указанным кодом. Закрытие выполняется только для активных заявок (статус «Пропуск выдан»). Связанный

идентификатор после закрытия заявки отвязывается от посетителя и возвращается в пул карт бюро пропусков.

## Функция **GetPersonVisitorRequests**

```
VisitorRequest[] GetPersonVisitorRequests ( Guid sessionID,  
Guid visitorID, bool issued, bool accepted, bool declined,  
bool active, bool completed )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid visitorID</code>	Уникальный ключ посетителя.
<code>bool issued</code>	Заявки со статусом «Ожидание согласования»
<code>bool accepted</code>	Заявки со статусом «Одобрена»
<code>bool declined</code>	Заявки со статусом «Отклонена»
<code>bool active</code>	Заявки со статусом «Выдан пропуск»
<code>bool completed</code>	Заявки со статусом «Закрыта»

Результат: Возвращает массив `VisitorRequest`.

Описание: Возвращает массив всех заявок, доступных по области видимости, оформлявшихся на указанного посетителя, с возможностью отбора/фильтрации по всем типам статусов заявок. Если заявок не найдено, то возвращается пустой массив.

## СОБЫТИЯ СИСТЕМЫ

### Функция GetEvents

```
EventsHistoryResult GetEvents( Guid sessionID, Guid TerritoryID, Guid  
PersNodeID, DateTime dtFrom, DateTime dtTo )
```

Параметры:

Guid sessionID	Уникальный ключ сессии.
Guid TerritoryID	Уникальный ключ территории.
Guid PersNodeID	Уникальный ключ персонала (подразделения или сотрудника).
DateTime dtFrom	Дата с...
DateTime dtTo	Дата по...

Результат: Возвращает EventsHistoryResult.

Описание: Возвращает историю событий для указанных территорий и персонала за указанный период времени. Для ключа территории и ключа персонала можно использовать специальное значение – Guid.Empty (00000000-0000-0000-0000-000000000000), при указании которого в качестве критерия будет взят корень соответствующей иерархии.

**Функция GetEvents устарела. Вместо неё рекомендуется использовать пакет функций [OpenEventHistorySession](#) (см. ниже).**

### Функция OpenEventHistorySession

```
GuidResult OpenEventHistorySession( Guid sessionID,  
EventHistoryQueryParams parameters )
```

Параметры:

Guid sessionID	Уникальный ключ сессии.
EventHistoryQueryParams parameters	Параметры формирования\фильтрации событий.

Результат: Возвращает GuidResult.

Описание: Открывает сессию отчета по событиям. Возвращает ключ для дальнейшего получения событий.

### Функция CloseEventHistorySession

```
void CloseEventHistorySession( Guid sessionID, Guid  
eventHistorySessionID)
```

Параметры:

Guid sessionID	Уникальный ключ сессии.
Guid eventHistorySessionID	Уникальный ключ сессии отчета по событиям.

Результат: Пустой.

Описание: Закрывает сессию отчета по событиям.

## Функция `GetEventHistoryResultCount`

```
int GetEventHistoryResultCount( Guid sessionID, Guid  
eventHistorySessionID )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid eventHistorySessionID</code>	Уникальный ключ сессии отчета по событиям.

Результат: Возвращает `int`.

Описание: Возвращает количество событий в открытой сессии.

## Функция `GetEventHistoryResult`

```
EventObject[] GetEventHistoryResult( Guid sessionID, Guid  
eventHistorySessionID, Guid[] fields, int offset, int count )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>Guid eventHistorySessionID</code>	Уникальный ключ сессии отчета по событиям.
<code>Guid[] fields</code>	Массив уникальных ключей полей, значения которых должны быть в результате.
<code>int offset</code>	Количество событий с начала, которое должно быть проигнорировано в этом результате.
<code>int count</code>	Количество событий, которое должно быть в результате.

Результат: Возвращает массив [EventObject](#).

Описание: Возвращает массив событий ([EventObject](#)) со значениями затребованных полей. Список затребованных полей (`fields`) состоит из одного или нескольких ключей из приведенной ниже таблицы. В списке могут присутствовать ключи дополнительных полей.

Guid	Описание
71b03d7b-2e11-47cd-bf47-adaf320aeb10	Дата события
c7ad4f51-d8af-4944-bf92-23714715147e	Время события
2c5ee108-28e3-4dcc-8c95-7f3222d8e67f	Дата/время события
633904b5-971b-4751-96a0-92dc03d5f616	Источник события (наименование территории или оператора)
42DAB9C6-5D30-4030-8CCD-2CAD6FCBC5F2	Источник события (массив идентификаторов территорий)
d1847aff-11aa-4ef2-aaaa-795ceefe5f9f	Тип события (наименование)
9F7A30E6-C9ED-4E62-83E3-59032A0F8D27	Идентификатор события (Guid)

Guid	Описание
C4AE9465-8375-4169-BA61-EB7E365A7352	Тип события (код в 16-ричной сист. счисления)
57CA38E4-ED6F-4D12-ADCB-2FAA16F950D7	Тип события (код в 10-чной сист. счисления)
68ef9fd3-a72d-4520-9c63-5c37b0ae8539	Субъект (ФИО) – из dictionary
7C6D82A0-C8C8-495B-9728-357807193D23	Идентификатор субъекта (PERS_ID - Guid)
66C5B505-C3A7-4227-AAD3-6B7BA3D8E612	Идентификатор субъекта (PERS_ID - Guid) <b>в ParsecNET Office</b>
4c5807cb-2c06-4725-9243-747e40c41d6c	Область (название)
2ab38696-1e30-4e04-a956-b951cb7c2033	Деталь (название)
89c9d5ac-6e13-4715-a524-7c3b34931385	Рабочая станция
fea92e1c-e07d-4932-a6a1-e8c53e3087d9	Оператор
03ceb65f-dcad-4b56-94b8-be9fdb463988	Подробности
e5ac823f-c4f6-48e7-bebe-e6d44c57c7ad	Сводка
66aa3a39-c866-4f34-9e99-e75f9918eae7	Комментарии оператора
99914915-c882-4d11-80ff-57acdc6cc015	Заголовок
2f4a647e-4d9e-48ad-bf11-b1e49ffeac7f	Сообщение
1bf8a893-7d21-4c0c-9a2d-2e333a2d769d	Полное имя субъекта
0de358e0-c91b-4333-b902-000000000003	Фамилия / Номер авто
0de358e0-c91b-4333-b902-000000000001	Имя / Модель авто
0de358e0-c91b-4333-b902-000000000002	Отчество / Цвет авто
0de358e0-c91b-4333-b902-000000000006	Табель
0de358e0-c91b-4333-b902-000000000009	Организация
0de358e0-c91b-4333-b902-000000000004	Подразделение
0a679144-d5ce-476d-a56e-0a696f079b71	Описание подразделения
0de358e0-c91b-4333-b902-00000000000a	Группа доступа
0de358e0-c91b-4333-b902-000000000005	Код карты
0de358e0-c91b-4333-b902-000000000007	Время действия с
0de358e0-c91b-4333-b902-000000000008	Время действия по
3ad06d24-43f6-45e0-8164-a98b4da955dc	Фото
6FCFA1BB-9624-4248-A2D5-AA84901C53C8	Субъект в чёрном списке
68D13785-C708-4418-8683-678A3F74957B	Изображения, прикрепленные к событию

## Функция GetHardwareEvents

```
string[] GetHardwareEvents( Guid sessionID, int fromIndex )
```

Параметры:

<code>Guid sessionID</code>	Уникальный ключ сессии.
<code>int fromIndex</code>	События с заданного значения индекса включительно.

Результат: Возвращает набор событий.

Описание: Возвращает набор из заданного количества оперативных событий. Данные возвращаются в виде JSON-документа.

*Пример объекта Event, полученного в результате выполнения функции*

```
{
  "ID": "e299479e-596d-4be0-9688-950918621783",
  "Index": 1,
  "ParentID": "00000000-0000-0000-0000-000000000000",
  "Code": 590193,
  "Component": "770ca0ee-4ecb-4648-bd3f-02b47140a670",
  "Territories": [
    "ee7b6630-bcee-4137-b759-61c90889bf39"
  ],
  "Date": "2015-06-10T11:43:30Z",
  "Items": [
    {
      "Instance": 0,
      "Key": 83886592,
      "Type": 5,
      "Value": "0084C471"
    },
    {
      "Instance": 0,
      "Key": 16778496,
      "Type": 1,
      "Value": 83886081
    },
    {
      "Instance": 0,
      "Key": 67110912,
      "Type": 4,
      "Value": "eec6409a-c092-46b7-9d8f-d311fc45c66d"
    },
    {
      "Instance": 0,
```



```

    "Key": 67110144,
    "Type": 4,
    "Value": "770ca0ee-4ecb-4648-bd3f-02b47140a670"
  },
  {
    "Instance": 0,
    "Key": 50334464,
    "Type": 3,
    "Value": 65600
  },
  {
    "Instance": 0,
    "Key": 67109376,
    "Type": 4,
    "Value": "039b2bf8-5a66-4b22-be4d-fb3d42af14d5"
  },
  {
    "Instance": 0,
    "Key": 201329152,
    "Type": 4,
    "Value": "a16486f7-8356-4fa9-8a4a-79e5659de052"
  }
]

```

*Структура объекта Event*

*Таблица 1.*

Tag	Физический/ логический тип	Описание
ID	string/Guid	Идентификатор события
Index	int/int	Индекс события. Отсчитывается с момента логина оператора в интеграционный сервис.
ParentID	string/Guid	Идентификатор родительского события, которое выступало «инициирующим»
Code	int/int	Код типа события
Component	string/Guid	Идентификатор компонента-источника события
Territories	string[]/Guid[]	Массив идентификаторов территорий, ссылающихся на компонент-источник события
Date	string/DateTime	Дата и время события
Items	ItemData[]	Массив служебных данных типа ItemData (см. Таблица 2.)

*Структура объекта ItemData*

*Таблица 2.*

Tag	Физический/ логический тип	Описание
-----	-------------------------------	----------

Instance	int/int	Индекс структуры данных в массиве данных события, сгруппированных по идентичности значений тэгов «Key» и «Type»
Key	int/int	Служебные данные. Содержат класс данных тэга «Value» в терминах системы ParsecNet. Класс данных вычленяется выполнением операции ( ( Key >> 8 ) & 0xffff ). (см. Таблица 3.)
Type	int/int	Логический тип данных тэга «Value» в терминах системы ParsecNet (см. Таблица 4.).

*Возможные значения класса данных в тэге «Key» структуры ItemData*

*Таблица 3.*

<b>Значение</b>	<b>Описание</b>
0	Неопределенное значение
2	Пользователь
4	Команда
5	Компонент
6	Состояние
7	Деталь
8	Устройство
9	Получатель быстрого транспорта
10	Отправитель быстрого транспорта
11	Маска событий
12	Рабочая станция
13	Оператор
14	Объект системы ParsecNet
15	Область видимости
16	Задание автоматизации
17	Интерпретатор события
18	Сервер
19	Дата/Время
20	Изображение
21	Автомобильный номер
22	Область изображения
23	Комментарии оператора
24	Канал
25	Звук
26	Связанное событие
27	Текстовое сообщение
28	Адрес получателя текстового сообщения

*Возможные значения тэга «Type» структуры ItemData*

*Таблица 4.*

<b>Значение</b>	<b>Описание</b>
0	Неопределенное значение
1	DWORD. 4-байтовое целое число
2	double. 8-байтовое число с плавающей точкой
3	long. 8-байтовое целое число
4	guid. 16-байтовый глобальный уникальный идентификатор
5	char[]. строковое значение с максимальной длиной в 16 символов
6	datetime. дата-время

7	byte[]. массив данных с максимальной длиной в 16 байт
8	guid. ссылка на бинарные данные большого размера
9	struct { int, int, int, int }. структура-описатель прямоугольной области

## Функция GetHardwareEventsResolved

`string[] GetHardwareEventsResolved( Guid sessionId, int fromIndex )`

Параметры:

<code>Guid sessionId</code>	Уникальный ключ сессии.
<code>int fromIndex</code>	События с заданного значения индекса включительно.

Результат: Возвращает набор событий.

Описание: Возвращает набор из заданного количества оперативных событий. Данные представляются в удобном для пользователя виде.

*Пример объекта EventResolved, полученного в результате выполнения функции (для того же события, что и в примере к предыдущей функции):*

```
{
  "ID": "e299479e-596d-4be0-9688-950918621783",
  "Index": 1,
  "ParentID": "00000000-0000-0000-0000-000000000000",
  "Code": {
    "Raw": 590193,
    "Resolved": "Нет входа - нет карты в БД"
  },
  "Classes": [
    {
      "Raw": 64,
      "Resolved": "Отказ в доступе"
    },
    {
      "Raw": 65536,
      "Resolved": "Отказ в доступе на вход"
    }
  ],
  "Component": {
    "Raw": "770ca0ee-4ecb-4648-bd3f-02b47140a670",
    "Resolved": "Шлагбаум въезд всех и выезд посетителей (NC 5K)"
  },
  "Territories": [
    {
      "Raw": "ee7b6630-bcee-4137-b759-61c90889bf39",
      "Resolved": "Шлагбаум въезд всех и выезд посетителей (NC 5K)"
    }
  ]
}
```

```

    }
  ],
  "Date": "2015-06-10T11:43:30Z",
  "CardCode": "0084C471",
  "User": {
    "Raw": "039b2bf8-5a66-4b22-be4d-fb3d42af14d5",
    "Resolved": "Иванов Иван Иванович"
  },
  "Operator": null,
  "Workstation": null,
  "Part": null,
  "Details": null
}

```

#### Структура объекта *EventResolved*

Tag	Физический/ логический тип	Описание
ID	string/Guid	Идентификатор события
Index	int/int	Индекс события. Отсчитывается с момента логина оператора в интеграционный сервис.
ParentID	string/Guid	Идентификатор родительского события, которое выступало «инициирующим»
Code	ResolvedValue	Экземпляр объекта типа ResolvedValue, описывающий тип события
Classes	ResolvedValue[]	Массив объектов типа ResolvedValue, описывающих категории, к которым относится событие
Component	ResolvedValue	Экземпляр объекта типа ResolvedValue, описывающий компонент-источник события
Territories	ResolvedValue[]	Массив объектов типа ResolvedValue, описывающих территории, которые ссылаются на компонент-источник события
Date	string/DateTime	Дата и время события
CardCode	string/string	Код карты пользователя системы ParsecNet (если применимо к конкретному типу события)
User	ResolvedValue	Экземпляр объекта типа ResolvedValue, описывающий пользователя системы ParsecNet (если применимо к конкретному типу события)
Operator	ResolvedValue	Экземпляр объекта типа ResolvedValue, описывающий оператора системы ParsecNet (если применимо к конкретному типу события)
Workstation	ResolvedValue	Экземпляр объекта типа ResolvedValue, описывающий рабочую станцию системы ParsecNet (если применимо к конкретному типу события)
Part	ResolvedValue	Экземпляр объекта типа ResolvedValue, описывающий деталь системы ParsecNet (если применимо к конкретному типу события)
Details	ResolvedValue	Экземпляр объекта типа ResolvedValue, описывающий данные объекта системы ParsecNet, связанного с конкретным событием (если применимо к конкретному типу события)

*Структура объекта ResolvedValue*

<b>Tag</b>	<b>Физический/ логический тип</b>	<b>Описание</b>
Raw	Any/Any	Данные, подходящие для компьютерной обработки
Resolved	string/string	Данные, подходящие для прочтения человеком

## ИСТОРИЯ ИЗМЕНЕНИЙ

### Версия 3.8 (октябрь 2018)

Добавлена функция GetPersonsChangedAfter  
Исправления в целях упрощения понимания.

### Версии, начиная с 3.7.1107.65 (август 2018)

Функция GetEventHistoryResult

Добавлена строка

42DAB9C6-5D30-4030-8CCD-2CAD6FCBC5F2	Источник события (массив идентификаторов территорий)
--------------------------------------	--

### Версия 3.7.1107 (ноябрь 2017)

Добавлена строка «Элемент PRIVILEGE\_MASK игнорируется» в описание функций:

AddPersonIdentifier

ChangePersonIdentifier

Строка «Если поле ID в структуре person равен null, то ID генерируется автоматически и возвращается.» заменена строкой «Если поле ID в структуре person равен Guid.Empty (00000000-0000-0000-0000-000000000000), то ID генерируется автоматически и возвращается.» в описании функций:

CreatePerson

CreateVisitor

### Версия 3.7.407 (март 2017)

Добавлены функции работы с вложенными группами доступа через веб-сервис:

GetInheritedAccessGroups

SetInheritedAccessGroups

Реализована возможность иметь несколько идентификаторов с одинаковым кодом у субъекта доступа.

### Версия 3.6.1018.18 (от 21/11/2016)

Добавлена функция GetVisitorExtraFieldTemplates, которая возвращает шаблоны дополнительных полей Посетителей.

Внесены изменения в связи с добавлением отдельного права "Удаление персонала".

Теперь посредством интеграционного сервиса можно получить прикрепленное к событию изображение с камеры.

Исправлена ошибка в интеграционном сервисе и в Web-консоли заявок Бюро пропусков: нельзя было изменить заявку со статусом "Согласована".

Менеджер сессий интеграционного сервиса оптимизирован.

### Версия 3.5.818

Расширен функционал API ParsecNET 3. Теперь разработчикам доступно создание произвольных расписаний и групп доступа! Также расширены возможности получения списка заявок бюро пропусков с фильтрацией на

стороне сервера. Полная свобода действий и неограниченные возможности интеграции!

Удалена привилегия canBlocked.

Добавлены аудитные события на открытие и закрытие сессии интеграционного сервиса.

Добавлена новая функция GetVisitorRequests

Добавлены новые функции для работы с расписаниями:

CreateAccessSchedule

CreateWorktimeSchedule

GetSchedule

SaveSchedule

DeleteSchedule

GetScheduleDetails

SetScheduleDays

SetScheduleFix

DeleteScheduleDays

GetHolidays

SetHolidays

DeleteHolidays

Добавлена новая функция для получения списка подгрупп компонент группы доступа GetSubAccessGroups , а также новый класс SubAccessGroup .

### **Версия 3.4.755 (от 17/07/2015)**

Добавлены функции для работы с группами доступа:

CreateAccessGroup

AddSubAccessGroup

DeleteSubAccessGroup

DeleteAccessGroup

Добавлена возможность получения статусов оборудования и отправки команд прямого управления посредством интеграционного сервиса:

GetHardwareState

SendHardwareCommand

Добавлены оперативные события оборудования в интеграционном сервисе

GetHardwareEvents

GetHardwareEventsResolved

Добавлен новый метод: SetIdentifierPrivileges

### **Версия 3.3.742 (от 02/03/2015)**

Добавлена функция CheckRole для проверки ролей оператора.

Добавлено поле EVENT\_ID, чтобы посредством интеграционного сервиса можно было получать связанные транзакции.

Исправлен список ключей дополнительных полей событий для GetEventHistoryResult, необходимых для работы со связанными запросами.

Исправлена ошибка в функции CreateTempAccessGroup.

Исправлены ошибки в функциях GetRootTerritory и GetRootOrgUnit.

Исправлены ошибки в функции удаления заявки DeleteIssuedVisitorRequest.

### **Версия 3.3.702 (от 18/06/2014)**

Для совместимости интеграционного сервиса (SDK) с Delphi в функцию OpenEventHistorySession внесено изменение, интерпретирующее EventHistoryQueryParams.ParentEventId=Guid.Empty как аналог значения null.

### **Версия 3.3.682 (от 02/12/2013)**

Исправлена ошибка переподключения интеграционного сервиса к серверу приложений.

Устранена ошибка, приводящая к зависанию консоли и интеграционного сервиса при большой нагрузке на систему.

В интеграционный сервис добавлено 3 новых функции для работы с заявками бюро пропусков:

CloseAllActiveVisitorRequests

CloseVisitorRequest

GetPersonVisitorRequests

### **Версия 3.3.673 (от 12/09/2013)**

Исправлена ошибка в интеграционном сервисе в функции GetPersonIdentifiers (в клиентских приложениях, работающих через интеграционный сервис и использующих данную функцию, необходима регенерация проху-классов).

### **Версия 3.3.670 (от 19/08/2013)**

Устранена проблема с падением интеграционного сервиса при работе с web-консолью заявок бюро пропусков под нагрузкой.

Исправлена ошибка с падением службы "ParsecNET 3 Server" при интенсивном создании сессий EventReport'ов через интеграционный сервис.

### **Версия 3.3.659 (от 01/06/2013)**

Добавлены две новые функции для предоставления/отмены привилегий "Запрет выхода" (NC-100K-IP, NC-8000, NC-8000-D), "Запрет входа" (NC-8000, NC-8000-D):

BlockPerson

UnblockPerson

### **Версия 3.3.644 (от 25/02/2013)**

Добавлены новые функции для работы с посетителями:

CreateVisitor

FindVisitors

GetOrgUnitSubItemsHierarchyWithVisitors

GetOrgUnitsHierarchyWithVisitors



## АЛФАВИТНЫЙ УКАЗАТЕЛЬ

AccessGroup .....	20	DeletePersonScheduleFix .....	35
AccessSchedule .....	15	DeleteSchedule .....	44
ActivateVisitorRequest.....	49	DeleteScheduleDays .....	45
AddPersonIdentifier.....	40	DeleteSubAccessGroup .....	48
AddPersonScheduleFix.....	35	Domain.....	21
AddSubAccessGroup .....	47	Event.....	20
BaseIdentifier.....	17	EventHistoryQueryParams.....	21
BaseObject .....	12	EventObject .....	21, 23
BaseOrgUnit.....	12	EventsHistory .....	20
BasePerson .....	13	EventsHistoryResult.....	12
BaseResult.....	11	ExtraFieldValue .....	14
BaseTerritory .....	18	FindPeople .....	32
BlockPerson .....	38	FindPersonByIdentifier .....	33
ChangePersonIdentifier .....	41	FindVehicle.....	32
CheckRole.....	25	FindVisitorRequest .....	49
CloseAllActiveVisitorRequests .....	51	FindVisitors .....	32
CloseEventHistorySession .....	53	GetAcceptedVisitorRequests .....	49
CloseOrgUnitEditingSession.....	30	GetAccessGroups .....	46
ClosePersonEditingSession .....	36	GetAccessSchedules .....	42
CloseSession .....	25	GetDomains .....	24
CloseVisitorRequest .....	51	GetEventHistoryResult .....	54, 62
ContinueSession .....	25	GetEventHistoryResultCount.....	54
CreateAccessGroup .....	47	GetEvents .....	53
CreateAccessSchedule.....	43	GetHardwareEvents.....	56
CreateOrgUnit .....	30	GetHardwareEventsResolved .....	59
CreatePerson .....	35	GetHardwareState.....	26
CreateTempAccessGroup .....	46	GetHolidays.....	45
CreateVehicle .....	36	GetInheritedAccessGroups .....	48
CreateVisitor .....	36	GetIssuedVisitorRequests.....	50
CreateVisitorRequest.....	49	GetOrgUnit.....	30
CreateWorktimeSchedule.....	43	GetOrgUnitsHierarhy .....	28
DeleteAccessGroup .....	47	GetOrgUnitsHierarhyWithPersons ...	28
DeleteHolidays .....	46	GetOrgUnitsHierarhyWithVehicle ....	28
DeleteIdentifier .....	40	GetOrgUnitsHierarhyWithVisitors....	28
DeleteIssuedVisitorRequest.....	50	GetOrgUnitSubItems .....	29
DeleteOrgUnit .....	31	GetOrgUnitSubItemsHierarhyWithPers ons.....	29
DeletePerson.....	38		

GetOrgUnitSubItemsHierarhyWithVehicle .....	29	OrgUnit .....	13
GetOrgUnitSubItemsHierarhyWithVisitors .....	29	Person .....	13
GetPerson .....	33	PersonExtraFieldTemplate .....	14
GetPersonExtraFieldTemplates .....	31	PersonScheduleFix .....	14
GetPersonExtraFieldValue .....	33	PersonWithPhoto .....	13
GetPersonExtraFieldValues .....	34	SaveOrgUnit .....	31
GetPersonExtraFieldValueString .....	34	SavePerson .....	37
GetPersonIdentifiers .....	40	SavePersonScheduleFix .....	35
GetPersonsChangedAfter .....	33	SaveSchedule .....	44
GetPersonScheduleFixes .....	34	SaveVisitorRequest .....	50
GetPersonVisitorRequests .....	52	Schedule .....	15
GetRootOrgUnit .....	28	ScheduleDay .....	16
GetRootTerritory .....	39	ScheduleFix .....	16
GetSchedule .....	44	SendHardwareCommand .....	26
GetScheduleDetails .....	44	Session .....	12
GetScheduleIntervals .....	42	SessionResult .....	11
GetSubAccessGroups .....	48	SetHolidays .....	46
GetTerritoriesHierarhy .....	39	SetIdentifierPrivileges .....	41
GetTerritory .....	40	SetInheritedAccessGroups .....	48
GetTerritorySubItems .....	39	SetPersonExtraFieldValue .....	37
GetVehicleExtraFieldTemplates .....	31	SetPersonExtraFieldValues .....	38
GetVersion .....	24	SetPersonOrgUnit .....	37
GetVisitorExtraFieldTemplates .....	31	SetPersonPhoto .....	37
GetVisitorRequest .....	50	SetScheduleDays .....	44
GetVisitorRequests .....	51	SetScheduleFix .....	45
GetWorktimeSchedules .....	42	StockIdentifier .....	18
GuidResult .....	11	StringResult .....	11
Holiday .....	17	SubAccessGroup .....	20
Identifier .....	17	Territory .....	19
IdentifierTemp .....	18	TerritoryWithComponent .....	19
ObjectResult .....	12	TimeInterval .....	16
OpenEventHistorySession .....	53	UnblockPerson .....	39
OpenOrgUnitEditingSession .....	30	ValidateExtraFieldValue .....	34
OpenPersonEditingSession .....	36	VisitorRequest .....	15
OpenSession .....	24	WorktimeInterval .....	17
OpenSessionWithInLocale .....	24	WorktimeSchedule .....	16