Оглавление

[API сервера 2](#_Toc418152498)

[Словари 2](#_Toc418152499)

[Структуры и классы 2](#_Toc418152500)

[Методы 2](#_Toc418152501)

[Безопасность 2](#_Toc418152502)

[Структуры и классы 2](#_Toc418152503)

[Методы 3](#_Toc418152504)

[События 4](#_Toc418152505)

[Тарифы 5](#_Toc418152506)

[Структуры и классы 5](#_Toc418152507)

[Методы 5](#_Toc418152508)

[События 6](#_Toc418152509)

[Оборудование и мониторинг 6](#_Toc418152510)

[Структуры и классы 6](#_Toc418152511)

[Методы 7](#_Toc418152512)

[События 9](#_Toc418152513)

[Отчеты 9](#_Toc418152514)

[Структуры и классы 9](#_Toc418152515)

[Методы 10](#_Toc418152516)

[События 11](#_Toc418152517)

[Сессии 12](#_Toc418152518)

[Структуры и классы 12](#_Toc418152519)

[Методы 12](#_Toc418152520)

[События 14](#_Toc418152521)

[Платежи 14](#_Toc418152522)

[Структуры и классы 14](#_Toc418152523)

[Методы 15](#_Toc418152524)

[События 15](#_Toc418152525)

# API сервера

## Словари

### Структуры и классы

#### ServerDictionary

public class ServerDictionary

{

public int Id;

public string Name;

public IDictionary<int, string> Items;

}

### Методы

#### GetAllServerDictionaries

Возвращает список всех словарных элементов сервера, сгрупированных по группам

##### Вход

отсутствует

##### Выход

* **Dictionaries**: сериализованный List<ServerDictionary>

## Безопасность

### Структуры и классы

#### ServerRequest

public class ServerRequest

{

public int Id;

public int GroupId;

public string Name;

public string Description;

}

#### User

public class User

{

public int Id;

public int TypeId;

public string Login;

public string Password;

public bool IsActive;

public List<int> Roles;

}

#### UserRole

public class UserRole

{

public int Id;

public string Name;

public List<string> Requests;

}

### Методы

#### GetAllRequests

Возвращает список всех запросов, зарегистрированных на сервере

##### Вход

отсутствует

##### Выход

* **Requests**: сериализованный List<ServerRequest>

#### GetAvailableRequests

Возвращает список всех запросов, доступных текущему пользователю

##### Вход

отсутствует

##### Выход

* **Requests**: сериализованный List<ServerRequest>

#### GetAllUserRoles

Возвращает список всех ролей пользователей, зарегистрированных на сервере

##### Вход

отсутствует

##### Выход

* **UserRoles**: сериализованный List<UserRole>

#### CreateUserRole

Создает новую роль пользователя

##### Вход

* **UserRole**: сериализованный UserRole

##### Выход

* **UserRole**: сериализованный UserRole с присвоенным сервером ID

#### UpdateUserRole

Обновляет роль пользователя (название, а также список доступных роли запросов) по ID роли

##### Вход

* **UserRole**: сериализованный UserRole

##### Выход

* **UserRole**: обновленный сериализованный UserRole

#### DeleteUserRole

Удаляет роль пользователя

##### Вход

* **UserRole**: сериализованный UserRole

##### Выход

отсутствует

#### GetAllUsers

Возвращает список всех пользователей, зарегистрированных на сервере

##### Вход

отсутствует

##### Выход

* **Users**: сериализованный List<User>

#### CreateUser

Создает нового пользователя

##### Вход

* **User**: сериализованный User

##### Выход

* **User**: сериализованный User с присвоенным сервером ID

#### UpdateUser

Обновляет пользователя (название, а также список ролей пользователя) по ID пользователя

##### Вход

* **User**: сериализованный User

##### Выход

* **User**: обновленный сериализованный User

#### DeleteUser

Удаляет пользователя

##### Вход

* **User**: сериализованный User

##### Выход

отсутствует

#### ChangePassword

Обновляет пароль у текущего подключенного пользователя

##### Вход

* **Password**: новый пароль

##### Выход

отсутствует

### События

#### UserRoleCreated

Событие создания роли пользователя

##### Выход

* **UserRole**: сериализованный UserRole

#### UserRoleChanged

Событие изменения роли пользователя

##### Выход

* **UserRole**: сериализованный UserRole

#### UserRoleDeleted

Событие удаления роли пользователя

##### Выход

* **UserRoleId**: идентификатор удаленной роли пользователя

#### UserCreated

Событие создания пользователя

##### Выход

* **User**: сериализованный User

#### UserChanged

Событие изменения пользователя

##### Выход

* **User**: сериализованный User

#### UserDeleted

Событие удаления пользователя

##### Выход

* **UserId**: идентификатор удаленного пользователя

## Тарифы

### Структуры и классы

#### Rate

public class Rate

{

public int Id;

public int TypeId;

public string Name;

public string Config;

}

### Методы

#### GetAllRates

Возвращает список всех тарифов

##### Вход

отсутствует

##### Выход

* **Rates**: сериализованный List<Rate>

#### CreateRate

Создает новый тариф

##### Вход

* **Rate**: сериализованный Rate

##### Выход

* **Rate**: сериализованный Rate с присвоенным сервером ID

#### UpdateRate

Обновляет тариф

##### Вход

* **Rate**: сериализованный Rate

##### Выход

* **Rate**: сериализованный Rate

#### DeleteRate

Удаляет тариф

##### Вход

* **Rate**: сериализованный Rate

##### Выход

отсутствует

### События

#### RateCreated

Событие создания нового тарифа

##### Выход

* **Rate**: сериализованный Rate

#### RateChanged

Событие изменения тарифа

##### Выход

* **Rate**: сериализованный Rate

#### RateDeleted

Событие удаления тарифа

##### Выход

* **RateId**: идентификатор тарифа

## Оборудование и мониторинг

### Структуры и классы

#### Device

public class Device

{

public Guid Id;

public Guid? ParentId;

public int TypeId;

public string Name;

public string Config;

}

#### DeviceState

public class DeviceState

{

public Guid DeviceID;

public string State;

public DateTime TimeStamp;

}

#### DeviceEvent

public class DeviceEvent

{

public Guid DeviceID;

public int TypeId;

public string Value;

public DateTime TimeStamp;

}

#### ZoneConfig

public class ZoneConfig

{

public int PlacesNumber;

public int? RateId;

}

#### GateConfig

public class GateConfig

{

public int TypeId;

public Guid ZoneInId;

public Guid ZoneOutId;

}

### Методы

#### GetAllDevices

Возвращает список всех устройств, зарегистрированных на сервере

##### Вход

отсутствует

##### Выход

* Devices: сериализованный List<Device>

#### RegisterDevices

Регистрирует новые устройства на сервере

##### Вход

* Devices: сериализованный List<Device>

##### Выход

отсутствует

#### UpdateDevices

Обновляет устройства на сервере

##### Вход

* **Devices**: сериализованный List<Device>

##### Выход

отсутствует

#### DeleteDevices

Обновляет устройства на сервере

##### Вход

* **Devices**: сериализованный List<Device>

##### Выход

отсутствует

#### GetLastDevicesStates

Возвращает список последних (по времени) зарегистрированных состояний для всех устройств

##### Вход

отсутствует

##### Выход

* **DevicesStates**: сериализованный List<DeviceState>

#### AddDevicesStates

Обновляет состояния устройств на сервере

##### Вход

* **DevicesStates**: сериализованный List<Device>

##### Выход

отсутствует

#### GetDevicesEvents

Возвращает список событий для устройств по фильтру

##### Вход

* **TimeStampFrom\*** — фильтр событий по времени

##### Выход

* **DevicesStates**: сериализованный List<DeviceEvent>

#### AddDevicesEvents

Регистрирует события устройств на сервере

##### Вход

* **DevicesEvents**: сериализованный List<DeviceEvent>

##### Выход

отсутствует

#### CorrectPlaces

Корректирует количество занятых мест в зоне

##### Вход

* **PlacesLogs**: сериализованный List<PlacesLog>

##### Выход

отсутствует

### События

#### DevicesCreated

Событие регистрации устройств на сервере

##### Выход

* **Devices**: сериализованный List<Device>

***DevicesChanged***

Событие изменения устройств

##### Выход

* **Devices**: сериализованный List<Device>

#### DevicesDeleted

Событие удаления устройств

##### Выход

* **Devices**: сериализованный List<Guid> — список идентификаторов удаленных устройств

#### DevicesStatesCreated

Событие появления обновленных статусов устройств

##### Выход

* **DevicesStates**: сериализованный List<DeviceState>

#### DevicesEventsCreated

Событие появления новых событий устройств

##### Выход

* **DevicesEvents**: сериализованный List<DeviceEvent>

## Отчеты

### Структуры и классы

#### Report

public class Report

{

public int Id;

public string Name;

public string Description;

public string ReportXML;

}

#### PlacesLog

public class PlacesLog

{

public long Id;

public int ZoneId;

public int PlacesCount;

public int PlacesOccupied;

public DateTime InsertStamp;

}

#### PassageLog

public class PassageLog

{

public long Id;

public int GateId;

public DateTime InsertStamp;

}

#### RequestLog

public class RequestLog

{

public long Id;

public int RequestId;

public int UserId;

public DateTime TimeStamp;

public string RequestData;

public bool IsSuccess;

public string ResponseData;

public string ResponseMessage;

}

### Методы

#### GetAllReports

Возвращает список всех отчетов

##### Вход

отсутствует

##### Выход

* **Reports**: сериализованный List<Report>

#### CreateReport

Создает новый отчет

##### Вход

* **Report**: сериализованный Report

##### Выход

* **Report**: сериализованный Report с присвоенным сервером ID

#### UpdateReport

Обновляет отчет

##### Вход

* **Report**: сериализованный Report

##### Выход

* **Report**: сериализованный Report

#### DeleteReport

Удаляет отчет

##### Вход

* **Report**: сериализованный Report

##### Выход

отсутствует

#### GetPlacesLogReport

Возвращает лог занятости мест по фильтру

##### Вход

* ZoneId\* — идентификатор зоны
* InsertStampFrom\* — время от
* InsertStampTo\* — время до

##### Выход

* **PlacesLogReport**: сериализованный List<PlacesLog>

#### GetPassageLogReport

Возвращает лог проезда по фильтру

##### Вход

* GateId\* — идентификатор стойки
* InsertStampFrom\* — время от
* InsertStampTo\* — время до

##### Выход

* **PassageLogReport**: сериализованный List<PassageLog>

#### GetRequestLogs

Возвращает лог запросов по фильтру

##### Вход

* RequestId\* — идентификатор запроса
* UserId\* — идентификатор пользователя
* IsSuccess\* — успешный/неуспешный запрос
* TimeStampFrom\* — время от
* TimeStampTo\* — время до

##### Выход

* **RequestLogs**: сериализованный List<RequestLog>

### События

#### ReportCreated

Событие создания нового отчета

##### Выход

* **Report**: сериализованный Report

#### ReportChanged

Событие изменения отчета

##### Выход

* **Report**: сериализованный Report

#### ReportDeleted

Событие удаления отчета

##### Выход

* **ReportId**: идентификатор тарифа

## Сессии

### Структуры и классы

#### Session

public class Session

{

public long Id;

public int StatusId;

public int TicketTypeId;

public long TicketNumber;

public int RateId;

public DateTime? ArriveTime;

public DateTime? DepartTime;

public DateTime? CancelTime;

public Guid? GateInId;

public Guid? GateOutId;

public DateTime GraceTime;

public DateTime CreateTime;

public DateTime LastChangeTime;

}

### Методы

#### GetSessions

Возвращает список сессий по фильтру (все фильтры являются необязательными)

##### Вход

* **Id**: сериализованный Report
* **StatusId**: статус сессии
* **TicketTypeId**: тип билета
* **TicketNumber**: номер билета
* **ArriveTimeFrom**: время приезда с
* **ArriveTimeTo**: премя приезда по
* **DepartTimeFrom**: время выезда с
* **DepartTimeTo**: время выезда по
* **CancelTimeFrom**: время отмены с
* **CancelTimeTo**: время отмены по
* **GateInId**: стойка въезда
* **GateOutId**: стойка выезда
* **CreateTimeFrom**: время создания сессии с
* **CreateTimeTo**: время создания сессии по
* **LastChangeTimeFrom**: время последнего изменения сессии с
* **LastChangeTimeTo**: время последнего изменения сессии по

##### Выход

* **Sessions**: сериализованный List<Session>

#### GetNewOneUseTicket

Создает новую сессию по одноразовому билету

##### Вход

* **RateId**: идентификатор тарифа для сессии
* **GateId**: стойка

Один из этих параметров обязателен — на его основе выбирается тариф для сессии

##### Выход

* **SessionId**: идентификатор сесси
* **TicketNumber**: номер билета
* **CreateTime**: время создания сессии

#### Arrive

Сигнал въезда на территорию парковки

##### Вход

* **SessionId**: идентификатор сессии
* **GateId**: идентификатор стойки въезда

##### Выход

отсутствует

#### Leave

Сигнал уезда клиента без заезда на территорию парковки

##### Вход

* **SessionId**: идентификатор сессии

##### Выход

отсутствует

#### Depart

Сигнал выезда с территории парковки

##### Вход

* **SessionId**: идентификатор сессии
* **GateId**: идентификатор стойки выезда

##### Выход

отсутствует

#### SetSessionGraceTime

Устанавливает грейс-период для сессии

##### Вход

* **SessionId**: идентификатор сессии
* **GraceTime**: DateTime, до какого момента длится грейс-период

##### Выход

* **Session**: сериализованный Session

#### RefreshSessionGracePeriod

Устанавливает новый грейс-период для сессии по-умолчанию начиная с текущего момента

##### Вход

* **SessionId**: идентификатор сессии

##### Выход

* **Session**: сериализованный Session

#### AskForDepart

Запрос возможности выезда по сессии на основе данных задолженности и платежей

##### Вход

* **SessionId**: идентификатор сессии

##### Выход

* **IsDepartAvailable**: bool, возможность выезда

#### CloseSession

Закрывает сессию

##### Вход

* **SessionId**: идентификатор сессии

##### Выход

отсутствует

### События

#### SessionCreated

Событие создания сессии

##### Выход

* **Session**: сериализованный Session

#### SessionChanged

Событие изменения сессии

##### Выход

* **Session**: сериализованный Session — новое состояние сессии
* **OriginalSession**: сериализованный Session — исходное состояние сессии

## Платежи

### Структуры и классы

#### Payment

public class Payment

{

public long Id;

public int UserId;

public long? SessionId;

public int Summ;

public DateTime Date;

}

### Методы

#### GetPayments

Возвращает список платежей по фильтру (все фильтры являются необязательными)

##### Вход

* **SessionId**: идентификатор сессии
* **DateFrom**: время платежа с
* **DateTo**: время платежа по

##### Выход

* **Payments**: сериализованный List<Payment>

#### CreatePayment

Создание платежа

##### Вход

* **Payment**: сериализованный Payment

##### Выход

* **Payment**: сериализованный Payment с присвоенным сервером идентификатором

### События

#### PaymentAdded

Событие создания платежа

##### Выход

* **Payment**: сериализованный Payment