

# Описание API для WEB портала

версия 1.31

## Оглавление

Настройка системы для работы с API .....	3
Синтаксис вызова .....	3
Функции доступные для вызова .....	5
Модуль Server .....	5
Список доступных серверов .....	5
Добавление сервера .....	5
Редактирование сервера .....	5
Удаление сервера .....	6
Поиск сервера .....	6
Модуль User .....	7
Список пользователей .....	7
Добавить пользователя .....	7
Редактировать пользователя .....	8
Удаление пользователя .....	8
Поиск пользователя .....	8
Модуль DevSets .....	9
Список наборов устройств пользователей .....	9
Добавление набора устройств пользователей (только для администраторов) .....	9
Редактирование набора устройств пользователей (только для администраторов) .....	10
Удаление набора устройств пользователей (только для администраторов) .....	10
Получение имен устройств из набора устройств пользователя .....	10
Получение параметров устройств из набора устройств пользователя .....	10
Модуль Company .....	12
Получение информации о компании .....	12
Модуль Web (только для администраторов) .....	13
Получение списка баз данных Web сервиса .....	13
Получение списка устройств базы данных Web сервиса .....	13
Получение списка групп устройств базы данных Web сервиса .....	13
Получение списка параметров групп устройств базы данных Web сервиса .....	14
Получение данных Web сервиса .....	14
Модуль Data .....	15
Получение данных на основе набора устройств .....	15
Получение данных на основе набора устройств (мультизапрос) .....	15
Пример получения подписи на языке PHP .....	16

# Настройка системы для работы с API

Для работы с API интерфейсом портала необходимо задать секретный ключ. Именно на основе этого ключа формируется сигнатура подписи запроса.

## Синтаксис вызова

Вызов функции производится с использованием REST запросов. В качестве адреса для вызова функций (serverURL) используется путь вида:

[serverURL]/api/[название модуля]/[вызываемая функция]

Например

<http://argo37.ru/api/server/getlist>

В этом адресе:

- [serverURL] это «http://argo37.ru»;
- [api] указание, что используем API
- [название модуля] это «Server»
- [вызываемая функция] это «getlist»

Типы HTTP – методов:

- GET – получение ресурса.
- POST – создание ресурса.
- PUT – обновление ресурса.
- DELETE – удаление ресурса.

Обязательные входные параметры:

- appid – идентификатор приложения, он же логин пользователя.
- sign – генерированная подпись.

Подпись **sign** формируется в виде md5 в верхнем регистре от строки вида:

«url?параметры&appid=...&secretkey=...». В данной строке:

- url – адрес системы с учетом ее контекста;
- secretkey – API ключ системы. Ключ системы используется только для формирования подписи **sign**.

Например

sign= strtoupper(md5(http://argo37.local/api/server/getlist/?appid=login&secretkey=apipass))= 58EA7DF6453131F24BE225B1ADAD9933

В этом случае полный запрос будет выглядеть следующим образом:

<http://argo37.local/api/server/getlist/?appid=guest&sign=58EA7DF6453131F24BE225B1ADAD9933>

Внимание!

- Параметры для генерации подписи должны следовать в отсортированном порядке.
- API ключ системы (secretkey) не передается в составе входных параметров запроса к серверу. Он используется только для формирования подписи **sign**.

Результат выполнения метода возвращается в виде JSON. Код ответа зависит от результата выполнения метода. Коды ответов:

- 200 – запрос выполнен успешно.
- 201 – запись создана (только для POST).
- 304 – данные не изменились (только для PUT).
- 400 – некорректный запрос.
- 401 – неавторизованный доступ.
- 403 – доступ запрещен.
- 404 – данные не найдены.
- 500 – внутренняя ошибка сервера.

Функция может вернуть JSON объект содержащий результат выполнения функции (методы GET, POST, PUT).

Например

В случае выполнения функции «Server/getlist» возвращается результат:

```
[
  {
    "id": "67",
    "name": "111",
    "srv_ip": "176.110.166.82",
    "srv_login": "guest"
  },
  {
    "id": "57",
    "name": "Test Server 1",
    "srv_ip": "109.60.135.135",
    "srv_login": "guest"
  }
]
```

В случае ошибки возвращается JSON объект с кодом и текстом ошибки.

Например

```
{
  "errorCode": 500,
  "errorMsg": "Ошибочный запрос"
}
```

### 3. Формат параметров

Дата – в формате dd-ММ-уууу

Строка – строки передаются в кодировке UTF-8 в стандартном представлении для соответствующего метода (urlencode для GET). Максимально возможная длина каждого параметра типа «Строка» указана в модели данных в скобках. Строки большей длины, чем максимально разрешенная, при помещении в базу данных обрезаются до максимально возможной длины для соответствующего поля.

Внимание!

В начале и конце параметров не допускаются использование пробелов.

Подпись рассчитывается от строкового представления параметров, а передавать нужно значение, кодированное с помощью urlencode.

# Функции доступные для вызова

## Модуль Server

Во всех функциях информация о сервере предоставляется следующей моделью данных:

Название поля	Описание	Тип
id	Идентификатор сервера	Число
name	Название сервера	Строка (255)
srv_ip	IP адрес сервера (если «:», то и порт)	Строка (21)
srv_login	Имя пользователя сервера	Строка (32)
srv_passw	Пароль сервера	Строка (32)

## Список доступных серверов

Тип запроса: GET

**/api/server/getlist**

---

Входные параметры:

- нет

Выходные параметры:

- Массив данных из параметров модели сервера

Внимание

В данном запросе параметр «Пароль сервера» не возвращается по соображениям безопасности

## Добавление сервера

Тип запроса: POST

**/api/server/add**

---

Входные параметры (обязательно):

- name – Название сервера
- srv\_ip – IP адрес сервера (или IP адрес:порт)
- srv\_login – Имя пользователя сервера
- srv\_passw – Пароль сервера

Выходные параметры:

- Возвращается число – ID добавленного сервера

## Редактирование сервера

Тип запроса: PUT

**/api/server/edit**

---

Входные параметры (обязательно):

- id – ID изменяемого сервера
- name – Название сервера
- srv\_ip – IP адрес сервера (или IP адрес:порт)
- srv\_login – Имя пользователя сервера
- srv\_passw – Пароль сервера

Выходные параметры:

- Возвращается строку «ОК»

Внимание!

ID сервера должен принадлежать той же компании, пользователь которой редактирует запись, в противном случае будет выдана ошибка.

## Удаление сервера

Тип запроса: DELETE

**/api/server/delete**

---

Входные параметры (обязательно):

- id – ID удаляемого сервера

Выходные параметры:

- Возвращается строку «ОК»

## Поиск сервера

Тип запроса: GET

**/api/server/find**

---

Входные параметры (обязательно):

- id – ID искомого сервера

Выходные параметры:

- Возвращается массив с параметрами найденного сервера на основе модели данных

## Модуль User

Во всех функциях информация о пользователе предоставляется следующей моделью данных:

Название поля	Описание	Тип
id	Идентификатор пользователя	Число
login	Логин пользователя	Строка (50)
password	Пароль пользователя	Строка (50)
email	E-mail пользователя	Строка (255)
name	Имя пользователя	Строка (50)
last_name	Фамилия пользователя	Строка (50)
web_admin	Является ли администратором группы	Число (1) – 0 или 1

Внимание!

ID пользователя должен принадлежать той же компании, пользователь которой редактирует запись, в противном случае будет выдана ошибка.

### Список пользователей

Тип запроса: GET

**/api/user/getlist**

---

Входные параметры:

- нет

Выходные параметры:

- Массив данных из параметров модели пользователя

Внимание!

В данном запросе параметр «Пароль пользователя» не возвращается по соображениям безопасности.

### Добавить пользователя

Тип запроса: POST

**/api/user/add**

---

Входные параметры (обязательно):

- login – логин пользователя
- password – пароль пользователя
- email – E-mail пользователя

Входные параметры (необязательно):

- web\_admin – Это администратор
- name – Имя пользователя
- last\_name – Фамилия пользователя

Выходные параметры:

- Возвращается число – ID добавленного пользователя

Внимание!

При добавлении пользователя вначале логина будет добавлен префикс компании, администратор которой добавляет пользователя.

## Редактировать пользователя

Тип запроса: PUT

**/api/user/edit**

---

Входные параметры (обязательно):

- id – ID редактируемого пользователя
- login – логин пользователя
- password – пароль пользователя
- email – E-mail пользователя

Входные параметры (необязательно):

- web\_admin – Это администратор
- name – Имя пользователя
- last\_name – Фамилия пользователя

Выходные параметры:

- Возвращается строку «ОК»

Внимание!

При редактировании логина необходимо учитывать, что логин вводится без префикса.

## Удаление пользователя

Тип запроса: DELETE

**/api/user/delete**

---

Входные параметры (обязательно):

- id – ID удаляемого пользователя

Выходные параметры:

- Возвращается строку «ОК»

## Поиск пользователя

Тип запроса: GET

**/api/user/find**

---

Входные параметры (обязательно):

- id – ID искомого пользователя

Выходные параметры:

- Возвращается массив с параметрами найденного пользователя на основе модели данных

## Модуль DevSets

Во всех функциях информация о наборах устройств предоставляется следующей моделью данных:

Название поля	Описание	Тип
id	Идентификатор набора устройств	Число
name	Название набора устройств	Строка (50)
srv_id	ID сервера	Число
db_guid	GUID базы данных	Строка (38)
db_name	Название базы данных	Строка (255)
data	Comma-separated данные	Строка
user_id	ID пользователя	Число

Внимание!

ID пользователя должен принадлежать той же компании, в противном случае будет выдана ошибка.

Comma-separated данные – данные, разделенные запятой. Это число или два числа через дефис. Два числа через дефис – это означает, что устройства находятся под регистратором.

### Список наборов устройств пользователей

Тип запроса: GET

**/api/devsets/getlist**

---

Входные параметры (необязательно):

- user\_id – ID пользователя набора устройств

Выходные параметры:

- Массив данных из параметров модели набора устройств

Внимание!

В данном запросе параметр «ID пользователя» будет принят, если запрашивающий пользователь является администратором

### Добавление набора устройств пользователей (только для администраторов)

Тип запроса: POST

**/api/devsets/add**

---

Входные параметры (обязательно):

- name – Название набора устройств
- srv\_id – ID сервера
- db\_guid – GUID базы данных
- data – Comma-separated данные

Входные параметры (необязательно):

- user\_id – ID пользователя набора устройств

Выходные параметры:

- Возвращается число – ID добавленного набора устройств

## Редактирование набора устройств пользователей (только для администраторов)

Тип запроса: PUT

**/api/devsets/edit**

---

Входные параметры (обязательно):

- id – ID редактируемого набора
- name – Название набора устройств
- srv\_id – ID сервера
- db\_guid – GUID базы данных
- data – Comma-separated данные

Входные параметры (необязательно):

- user\_id – ID пользователя набора устройств

Выходные параметры:

- Возвращается строку «OK»

## Удаление набора устройств пользователей (только для администраторов)

Тип запроса: DELETE

**/api/devsets/delete**

---

Входные параметры (обязательно):

- id – ID удаляемого набора

Входные параметры (необязательно):

- user\_id – ID пользователя набора устройств

Выходные параметры:

- Возвращается строку «OK»

## Получение имен устройств из набора устройств пользователя

Тип запроса: GET

**/api/devsets/getnames**

---

Входные параметры (обязательно):

- id – ID набора устройств

Входные параметры (необязательно):

- user\_id – ID пользователя набора устройств

Выходные параметры:

- Массив данных, где в качестве ключей – число или два числа через дефис, а значений – массив, где
  - name – Название устройства (если число) / Название регистратора (если два числа через дефис)
  - name2 – Название устройства (присутствует, если только два числа через дефис)
  - user\_type – Тип устройства
  - serial\_num – Серийный номер устройства

Внимание!

В данном запросе параметр «ID пользователя» будет принят, если запрашивающий пользователь является администратором

## Получение параметров устройств из набора устройств пользователя

Тип запроса: GET

**/api/devsets/getparams**

---

Входные параметры (обязательно):

- id – ID набора устройств

Входные параметры (необязательно):

- user\_id – ID пользователя набора устройств

Выходные параметры:

- Массив данных, где в качестве ключей – число или два числа через дефис, а значений – массив, где
  - если два числа через дефис:  
ключ – ID параметра, значение – Название параметра
  - число:  
ключ – ID группы, значение – массив, где  
ключ – ID параметра, значение – Название параметра

## Модуль Company

Во всех функциях информация о компании предоставляется следующей моделью данных:

Название поля	Описание	Тип
pref	Префикс компании	Строка (20)
srv_cnt	Ограничение серверов	Число

### Получение информации о компании

Тип запроса: GET

**/api/company/getinfo**

---

Входные параметры:

- нет

Выходные параметры:

- Массив данных из параметров модели компании

Внимание!

Ограничение серверов означает, сколько может компания добавить серверов в список для работы. При попытке добавить еще один сервер, будет возвращена ошибка.

## Модуль Web (только для администраторов)

Во всех функциях информация о web сервисе предоставляется следующей моделью данных:

Название поля	Описание	Тип
db_name	Название базы данных	Строка (255)
db_guid	GUID базы данных	Строка (38)

### Получение списка баз данных Web сервиса

Тип запроса: POST

/api/web/get\_dbs

---

Входные параметры (обязательно):

- srv\_id – ID сервера

Выходные параметры:

- Массив данных из параметров модели компании

### Получение списка устройств базы данных Web сервиса

Тип запроса: POST

/api/web/get\_devices

---

Входные параметры (обязательно):

- srv\_id – ID сервера
- db\_guid – GUID базы данных

Выходные параметры:

- Массив данных, где в качестве ключей – число или два числа через дефис, а значений – массив, где
  - name – Название устройства (если число) / Название регистратора (если два числа через дефис)
  - name2 – Название устройства (присутствует, если только два числа через дефис)
  - user\_type – Тип устройства
  - serial\_num – Серийный номер устройства

### Получение списка групп устройств базы данных Web сервиса

Тип запроса: POST

/api/web/get\_groups

---

Входные параметры (обязательно):

- srv\_id – ID сервера
- db\_guid – GUID базы данных
- dev\_id – Comma-separated ID запрашиваемых устройств

Выходные параметры:

- Массив данных, где в качестве ключей – число или два числа через дефис, а значений – массив, где
  - id – Название устройства (если число) / Название регистратора (если два числа через дефис)
  - name – Название группы (если число) / «Все параметры» (если два числа через дефис)
  - user\_type – Тип устройства (присутствует, если только число)
  - serial\_num – Серийный номер устройства (присутствует, если только число)

Внимание!

Если в параметр «dev\_id» попадут ID, состоящие из двух чисел через дефис (устройства, которые находятся под регистратором), то для них будут выведены ID групп, равные «0» и «name=Все параметры».

## Получение списка параметров групп устройств базы данных Web сервиса

Тип запроса: POST

**/api/web/get\_params**

---

Входные параметры (обязательно):

- `srv_id` – ID сервера
- `db_guid` – GUID базы данных
- `dev_group` – Comma-separated данные, состоящие из «`dev_id-group_id`», где
  - `dev_id` – ID запрашиваемого устройства
  - `group_id` – ID группы

Выходные параметры:

- Массив данных, где в качестве ключей – два числа через дефис, а значений – массив, где
  - `first_record_date` – Дата начала интервала данных
  - `last_record_date` – Дата окончания интервала данных
  - `params` – Массив параметров, где в качестве ключей – ID параметра, а значений – массив, где
    - `name` – Название параметра
    - `value_type` – Тип значения
    - `param_type` – Тип параметра
    - `measure` – Единица измерения
    - `mul_koeff` – Коэффициент

**Внимание!**

Если Дата начала интервала данных и/или Дата окончания интервала данных вернулись пустыми, то можно считать, что данных в базе нет.

## Получение данных Web сервиса

Тип запроса: POST

**/api/web/get\_data**

---

Входные параметры (обязательно):

- `srv_id` – ID сервера
- `db_guid` – GUID базы данных
- `dev_id` – ID запрашиваемого устройства
- `group_id` – ID группы
- `params` – Comma-separated данные из ID параметров
- `first_record_date` – Дата начала интервала данных
- `last_record_date` – Дата окончания интервала данных

Выходные параметры:

- Массив данных, где значения – массив, где
  - `date` – Дата
  - `time` – Время
  - `params` – Массив параметров, где в качестве ключей – ID параметра, а значений – массив, где
    - `value` – Значение
    - `Mvalue` – Приведенное значение

## Модуль Data

Во всех функциях информация о данных предоставляется следующей моделью данных:

Название поля	Описание	Тип
date	Дата	Строка (10)
time	Время	Строка (5)
params	Массив параметров	Массив

### Получение данных на основе набора устройств

Тип запроса: POST

#### /api/data/getdata

Входные параметры (обязательно):

- devset\_id – ID набора устройств
- dev\_id – ID запрашиваемого устройства
- params – Comma-separated данные из ID параметров
- first\_record\_date – Дата начала интервала данных
- last\_record\_date – Дата окончания интервала данных

Входные параметры (если dev\_id не два числа через дефис):

- group\_id – ID группы

Выходные параметры:

- Массив данных, где значения – массив, где
  - date – Дата
  - time – Время
  - params – Массив параметров, где в качестве ключей – ID параметра, а значений – массив, где
    - value – Значение
    - Mvalue – Приведенное значение

### Получение данных на основе набора устройств (мультизапрос)

Тип запроса: POST

#### /api/data/multigetdata

Входные параметры (обязательно):

- devset\_id – ID набора устройств
- dev\_id – массив устройств (форматы:  
*1267-29269/522195,522196,522197* ,  
где *1267-29269* - составной ID устройства, а *522195,522196,522197* – ID параметров  
или  
*1267,29269/522195,522196,522197*  
где *1267* – ID устройства, *29269* – ID группы, а *522195,522196,522197* – ID параметров)
- first\_record\_date – Дата начала интервала данных (формат ДД.ММ.ГГГГ или ДД.ММ.ГГГГ ЧЧ:ММ:СС)
- last\_record\_date – Дата окончания интервала данных (формат ДД.ММ.ГГГГ или ДД.ММ.ГГГГ ЧЧ:ММ:СС)

Выходные параметры:

- Массив данных, где
  - если ID устройства составной, то ключ – составной ID устройства, значение – массив записей, где
    - date – Дата
    - time – Время
    - value – Значение
    - Mvalue – Приведенное значение
  - если ID простое число, то ключ – ID устройства, значение – массив, где
    - key – ID группы, значение – массив записей, где
      - date – Дата
      - time – Время
      - value – Значение
      - Mvalue – Приведенное значение

## Пример получения подписи на языке PHP

```
function signit($url, $params,$appid,$secretkey)
{
    $ret_params = $params;
    ksort($ret_params);
    $ret_params['appid'] = $appid;
    $signstring="$url?";
    foreach ($ret_params as $key => $val) {
        if ($val != "") {
            $signstring .= "$key=$val&";
        }
    }
    $signstring .= "secretkey=".$secretkey;
    $ret_params['sign'] = strtoupper(md5($signstring));
    return $ret_params;
}
```