

- [Схема](#)
- [Отправка запроса на сервер](#)
- [Структура тела запроса](#)
 - [Структура ответа. Успешное выполнение](#)
 - [Ошибка при выполнении](#)

Схема

При работе с API запросы необходимо отправлять по адресу: <https://your-domain/api>

Все запросы принимаются по https в кодировке UTF-8. Ответ формируется в формате JSON-RPC 2.2. Данные типа «Дата» указываются в формате «Y-m-d».

Отправка запроса на сервер

При формировании запросов **токен** отправляются на сервер в **заголовке(header)** http запроса в атрибуте **Authorization** со значением Bearer {Ваш токен}

Пример запроса:

```
curl -k 'https://test.evateam.ru/api/' -X POST -H "Content-Type: application/json" --data-raw '{
    "jsonrpc": "2.2",
    "method": "CmfTask.list",
    "kwargs": {"filter": ["code", "==", "DEV-000003"]}
}' -H 'Authorization: Bearer {Ваш токен}'
```

Структура тела запроса

Все передаваемые данные — простые записи, сериализованные в JSON.

Запрос — это вызов определенного метода, предоставляемого удаленной системой.

```
{
  "jsonrpc": "2.2",
  "method": "",
  "args": [],
  "kwargs": {
    "filter": [],
    "fields": [],
    ...
  },
  "callid": "Необязательный идентификатор запроса"
}
```

Описание полей запроса:

- *jsonrpc* — строка, указывающая версию протокола JSON-RPC;
- method* — строка, содержащая имя вызываемого метода;
- *args* — позиционные аргументы;
- *kwargs* — словарь с именованными аргументами
 - *fields* — список мета-правил для указания какие поля необходимо загрузить (можно не указывать);
 - *filter* — массив в формате BQL, для фильтрации результата;

Структура ответа. Успешное выполнение

```
{
  "jsonrpc": "2.2",
  "result": "",
```

```
"meta": "",  
"callid": "",  
"profiler_data": "",  
"alert": "",  
"abort": ""  
}  
}
```

Описание полей ответа:

jsonrpc — строка, указывающая версию протокола JSON-RPC;
result — словарь с результатом ответа в формате json;
meta — описание полей, для удобства использования на frontend;
callid — значение, которое используется для установки соответствия между запросом и ответом(необязательно);
profiler_data — информация о выполнении запроса;

alert — уведомление для frontend;
abort — уведомление об ошибке;

Ошибка при выполнении

```
{
  "jsonrpc": "2.2",
  "error": {"code": code, "message": message},
  "alert": "",
  "callid": "Идентификатор запроса, который вызвал ошибку"
}
```

Описание полей ошибки:

jsonrpc – строка, указывающая версию протокола JSON-RPC;
error – code – код ошибки и message – сообщение;
alert – уведомление для frontend;

```

    "version": "0.0.1",
    "name": "node-serialport",
    "description": "Read and write serial port using Node.js v0.10.20+",
    "author": "J. M. Smith <jm.smith@jmsmith.com>",
    "repository": "https://github.com/jm-smith/node-serialport",
    "dependencies": {
      "util": "0.0.20"
    },
    "devDependencies": {
      "jshint": "2.0.0",
      "nodeunit": "0.10.0",
      "serialport": "0.0.1"
    },
    "scripts": {
      "test": "nodeunit test"
    },
    "main": "index.js",
    "bin": {
      "serialport": "index.js"
    }
  }
}

```

oas_rpc_generated_v1_9_
2 июл. 2025 г., 13:30