

# О реализации взаимодействия программы SiteSputnik со сторонними API

В данном обзоре представлен один из вариантов взаимодействия программы SiteSputnik с API сайтов.

Прежде, чем описать работу серверной части, необходимо знать, что делается программой SiteSputnik на стороне клиента, а именно:

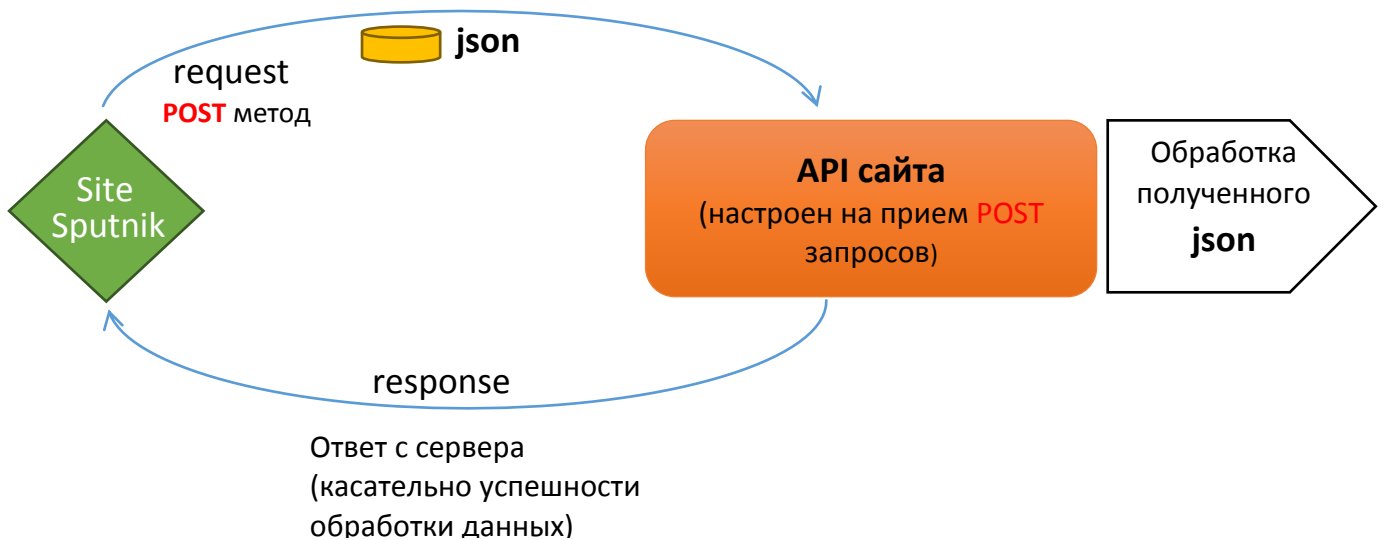
- 1) какая информация подготавливается к отправке на сервер;
- 2) метод и протокол отправки этой информации;
- 3) её детальную структуру и формат.

На сервер программой SiteSputnik может быть отправлено содержание Рубрик или Чистого потока, отправка осуществляется по протоколу http/https методом POST в формате JSON, детальная структура которого такова:

```
{
  "route-api": "адрес отправки(собственно API)",
  "heading": "название проекта / рубрики",
  "date": "дата проведения мониторинга",
  "time": "время проведения мониторинга",
  "data": [
    {
      "title": "заголовок новости",
      "author": "автор",
      "description": "описание новости",
      "date": "дата новости",
      "time": "время новости",
      "rubrics": [
        {
          "head": "название рубрики, в которые попала новость"
        },
        ...
      ],
      "objects": [
        {
          "type": "тип объекта, размещенного в новости",
          "name": "его имя",
          "rank": "количество упоминаний"
        },
        ...
      ],
      "domain": "домен второго уровня",
      "url": "ссылка на новость"
    },
    ...
  ]
}
```

Подробнее о работе на стороне клиента на ссылке: [SiteSputnik: Конвертация выдачи в JSON-формат и её использование](#).

## Общая схема взаимодействия выглядит в общем так:



Как один из примеров возможной реализации этой схемы, представляется реализация сервера на платформе NodeJs / Express с записью получаемого json в базу данных MongoDB, вот ссылка на репозиторий с кодом - <https://github.com/NickRbk/Simple-API>.

В папке **'server'** находятся все файлы для запуска сервера. Папки **'app'** и **'config'** определяют настройки для запуска/работы БД и являются несущественными для описания общей схемы работы с SiteSputnik.

Основной файл - **server.js**, в котором определен API, отвечающий за прием запросов методом POST.

Это частный случай реализации серверной части сайта для приема данных с SiteSputnik. Все зависит от конкретной технологии, на которой работает конкретный сайт. В данном случае в параметр "route-api": СайтСпутник должен поместить следующее API: <http://localhost:7777/api/news> (здесь приведен адрес API для локально запущенного сервера).

## Информация о создании API размещена в табличке:

```
// Создание API

const apiRouter = express.Router();
app.use('/api', apiRouter);

// Определяем адрес /api/news и назначаем ему метод POST
// То есть если Ваш сайт условно говоря https://test.com, то по адресу
// https://test.com/api/news можно отправлять запрос и этот адрес нужно
// указать для отправки в SiteSputnik

apiRouter.route('/news')
  .post( (req, res) => {

    // через переменную req.body получаем доступ к json
    // и считываем данные, обращаясь к ключам json

    const dataset = req.body['data'];
    const generalData = {
      routeApi: req.body['route-api'],
      heading: req.body['heading'],
      dateProject: req.body['date'],
      timeProject: req.body['time']
    };

    // Тут для примера определена логика работы с БД,
    // функция для сохранения данных определена
    // по пути './app/helpers/saveNews'

    const saveNews = require('./app/helpers/saveNews');
    saveNews(dataset, generalData)
      .then(
        result => {
          res.json({ message: result }) // Ответ в случае успеха
        },
        error => {
          res.json({ message: error }) // Ответ в случае ошибки
        }
      );
  });
// _____
});
```

Автор: Николай Рыбак, г. Киев, Украина, 09.01.2018г.