

# API ПAK BcMK

ООО "Бизнес-Мониторинг"

Version 0.4, 04 июля 2023

# Оглавление

1. API	1
1.1. Основные положения	1
1.2. Доступ	1
1.3. Ограничение доступа по IP-адресу	1
1.4. Формат взаимодействия	1
2. Авторизация	2
2.1. API метод авторизации	2
3. Работа с объектами	4
3.1. Создание нового объекта	4
3.2. Изменение объекта	6
3.3. Получение объекта	8
3.4. Получение списка объектов	9
3.5. Передача объекта на согласование	10
3.6. Отмена согласования объекта	12
3.7. Удаление объекта	13
3.8. Формат объекта	14
3.8.1. Формат поля адрес	16
3.9. Работа с файлами	16
3.9.1. Добавление файла к объекту	16
3.9.2. Получение данных	17
3.9.3. Удаление файла	18
4. Тревоги	20
4.1. Поддерживаемые запросы по работе с тревогами	20
4.2. Создание новой тревоги по объекту	20
4.3. Отправка нового события по тревоге	21
4.4. Получение списка активных тревог	22
4.5. Отмена ранее переданной тревоги	23
4.6. Форматы данных	24
4.6.1. Формат тревоги	24
4.6.2. Формат передаваемой тревоги	26
4.6.3. Тикет принятой тревоги	26
4.6.4. Формат события по тревоге	26
4.6.5. Коды событий	26
5. Регионы / Подразделения	28
5.1. Получение списка регионов	28
5.2. Формат списка регионов	29
5.3. Получение списка подразделений по региону	29
5.4. Формат подразделения	30

6. Передача изменений в мониторинговую компанию .....	32
7. Справочные данные .....	33
7.1. Статусы объекта .....	33
7.2. Статус тревоги .....	33
7.3. Действие оператора .....	33
7.4. Коды событий .....	34
7.5. Категория объекта .....	34
7.6. Тип объекта .....	34
7.7. Формат ответа при ошибке в REST API .....	34
7.8. Коды ошибок при работе через REST API .....	35
7.8.1. Коды ошибок .....	35

# Раздел 1. API

## 1.1. Основные положения

При работе с API существует ряд ограничений: Количество запросов ограничено 100 запросами в секунду. При превышении данного показателя на новые запросы будет выдан ответ сервера: 503 (Service Temporarily Unavailable)

## 1.2. Доступ

Приложение (программа, скрипт и др.) выполняет запрос к API от имени пользователя мониторинговой компании и управляет данными, принадлежащими данной компании. Пользователи из мониторинговой компании могут иметь роль **Администратор МК** или **Сотрудник МК**. Работа по API осуществляется с ролью Сотрудник МК. Для авторизации используется механизм JWT. Подробности осуществления авторизации в системе смотрите на соответствующей странице.

## 1.3. Ограничение доступа по IP-адресу

Доступ к API ограничен по IP-адресу, что повышает информационную безопасность. Список разрешенных IP-адресов можно согласовать с Администратором.

## 1.4. Формат взаимодействия

Доступ к API предоставляется через криптографический протокол SSL. Установка SSL-соединения обязательна для вызова методов API.

Взаимодействие с API ПаК ВcМК в формате REST, используя представление данных в формате JSON. Запросы в формате JSON передаются методом HTTP POST | GET | PUT | DELETE на следующий адрес: <https://pakvcmk.ru/api/>

# Раздел 2. Авторизация

Авторизация пользователя в API производится через [JWT]([https://en.wikipedia.org/wiki/JSON\\_Web\\_Token](https://en.wikipedia.org/wiki/JSON_Web_Token))

При работе с API нужно получить токен с помощью ниже приведенного метода API.

Последующие запросы должны содержать в заголовке значение токена в виде поля: 'Authorization:Bearer токен'

## 2.1. API метод авторизации

### URL

/login

### Метод

POST

### Данные

### Формат:

```
{
  "username" : "[string]",
  "password" : "[alphanumeric]"
}
```

\_Возвращает токен для дальнейшей работы с системой\_

### Код

200

### Тело ответа

```
{ token : [string] }
```

### Ответ при ошибке

Возникает в случае неправильного логина или пароля

```
Code: 401 UNAUTHORIZED
```

```
Content: { error : "Log in" }
```

### Пример использования

```

<?php

$token = "";
$base_url = "https://pakvcmk.ru/api/";
$user = "john"; $password = "password";

function curl_get($url, $post = "", $token = '', $method = "")
{
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    if ($method != "") {
        curl_setopt($ch, CURLOPT_CUSTOMREQUEST, $method);
    }
    if ($post != "") {
        curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($post));
    }
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
    $headers = array('Accept: application/json', 'Content-Type: application/json',
'Authorization:Bearer ' . $token);
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    $head = curl_exec($ch);
    $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    curl_close($ch);

    if ($httpCode != 200 ) {
        echo "Error in response: code: $httpCode, text: '$head' \n";
        return false;
    }
    return $head;
}

$token_str = curl_get($base_url . "login", array("username" => $user, "password" =>
$password));

if ($token_str!=false){
    $token = json_decode($token_str);
    echo "Token: {$token->token}";
}

```

Вернет результат, в случае правильного логина и пароля:

Token: eyJhbG...nYhJQV2C4

## Раздел 3. Работа с объектами

Метод	Параметры	Результат	Описание
GET <a href="https://pakvcmk.ru/api/objects">https://pakvcmk.ru/api/objects</a>	нет	JSON Массив объектов	Получение списка всех объектов компании
GET <a href="https://pakvcmk.ru/api/objects/:id">https://pakvcmk.ru/api/objects/:id</a>	нет	JSON объект	Получение информации по объекту
DELETE <a href="https://pakvcmk.ru/api/objects/:id">https://pakvcmk.ru/api/objects/:id</a>	нет		Удаление объекта
POST <a href="https://pakvcmk.ru/api/objects">https://pakvcmk.ru/api/objects</a>	JSON объект	code 201 - Created  { "status": "success", "object_id": "%id%" }	<a href="#">Создание нового объекта</a>
PUT <a href="https://pakvcmk.ru/api/objects/:id">https://pakvcmk.ru/api/objects/:id</a>	Аналогично POST запросу	В случае успеха:  code 200 Json объект с новыми данными.  Код ошибки 4** или 5**  { "error" : "Описание ошибки" }	<a href="#">Изменение объекта</a>
POST <a href="https://pakvcmk.ru/api/approve">https://pakvcmk.ru/api/approve</a>	{  "object_id": "ID объекта", "comment" : "Тест комментария"  }	В случае успеха:  code 200	<a href="#">Передача объекта на согласование</a>

### 3.1. Создание нового объекта

<b>Наименование</b>	Добавление нового объекта в систему
<b>URL</b>	/objects
<b>Метод</b>	POST
<b>Параметры URL</b>	нет
<b>Данные</b>	JSON <a href="#">Формат объекта</a>

<b>Ответ при удачном запросе</b>	<b>Код ответа:</b> 201 <b>Содержимое:</b> {status: "success", object_id: %id%}
<b>Ответ при ошибке</b>	<b>Код ответа:</b> 401 UNAUTHORIZED <b>Содержимое:</b> "" <div data-bbox="359 416 1458 555" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">Возникает в случае, если пользователь не авторизован, либо у него нет прав на создание объекта</div> <p>или</p> <b>Код ответа:</b> 500 Internal Server Error <b>Содержимое:</b> { error : "Описание ошибки" } <div data-bbox="359 808 1458 947" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</div>

**Пример  
использован  
ия**

```
$.ajax({
  url: "/objects",
  dataType: "json",
  data : {
    "object_num": "2",
    "address": "dfsadasdasdasd",
    "category_id": 1,
    "phone": "21321312312321",
    "name": "Аптека",
    "locked": "fdsadasdasd",
    "district_id": "5820ab00-3b53-480d-ba4a-bfdbb4eb603d",
    "type_id": 0,
    "contacts": [
      {
        "name"=> "Имя контакта",
        "phone"=> "Телефон контакта"
      }
    ],
    "zones": [
      {
        "zone_num"=> "Номер зоны",
        "name"=> "Наименование зоны",
        "description"=> "Описание зоны"
      }
    ]
  },
  type : "PUT",
  success : function(r) {
    console.log(r);
  }
});
```

## 3.2. Изменение объекта

<b>Наименование</b>	Изменение данных объекта
<b>URL</b>	/objects/:id
<b>Метод</b>	PUT
<b>Параметры URL</b>	Обязательно: <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">id:[UUID] Пример: 5213a389-ccda-11e6-9751-0cc47aac4166</div>

<b>Данные</b>	JSON <a href="#">Формат объекта</a>
<b>Ответ при удачном запросе</b>	<b>Код ответа:</b> 200 <b>Содержимое:</b> {status: "success", object_id: %id%}
<b>Ответ при ошибке</b>	<b>Код ответа:</b> 401 UNAUTHORIZED <b>Содержимое:</b> "" <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">Возникает в случае, если пользователь не авторизован, либо у него нет прав на создание объекта</div> <p>или</p> <b>Код ответа:</b> 500 Internal Server Error <b>Содержимое:</b> { error : "Описание ошибки" } <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</div>

**Пример  
использован  
ия**

```
$.ajax({
  url: "/objects/5213a389-ccda-11e6-9751-0cc47aac4166",
  dataType: "json",
  data : {
    "object_num": "2",
    "address": "dfsadasdasd",
    "category_id": 1,
    "phone": "21321312312321",
    "name": "Аптека",
    "locked": "fdsadasd",
    "district_id": "5820ab00-3b53-480d-ba4a-bfdbb4eb603d",
    "type_id": 0,
    "contacts": [
      {
        "name"=> "Имя контакта",
        "phone"=> "Телефон контакта"
      }
    ],
    "zones": [
      {
        "zone_num"=> "Номер зоны",
        "name"=> "Наименование зоны",
        "description"=> "Описание зоны"
      }
    ]
  },
  type : "PUT",
  success : function(r) {
    console.log(r);
  }
});
```

### 3.3. Получение объекта

<b>Наименование</b>	Получение списка всех объектов
<b>URL</b>	/objects/:id
<b>Метод</b>	GET
<b>Параметры URL</b>	Обязательно: <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">id:[UUID] Пример: 5213a389-ccda-11e6-9751-0cc47aac4166</div>

<b>Данные</b>	нет
<b>Ответ при удачном запросе</b>	<p>Код ответа: 200</p> <p>Содержимое: JSON <a href="#">Формат объекта</a></p>
<b>Ответ при ошибке</b>	<p>Код ответа: 401 UNAUTHORIZED</p> <p>Содержимое: ""</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на получение данных объекта</p> </div> <p>или</p> <p>Код ответа: 500 Internal Server Error</p> <p>Содержимое: { error : "Описание ошибки" }</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p> </div>
<b>Пример использования</b>	<pre>\$.ajax({   url: "/objects/5213a389-ccda-11e6-9751-0cc47aac4166",   type: "GET",   success: function(r) {     console.log(r);   } });</pre>

## 3.4. Получение списка объектов

Получение списка объектов с основными данными. Расширенные данные можно получить по запросу [Получение объекта](#)

<b>Наименование</b>	Получение списка всех объектов
<b>URL</b>	/objects
<b>Метод</b>	GET
<b>Параметры URL</b>	нет

<b>Данные</b>	нет
<b>Ответ при удачном запросе</b>	<p><b>Код ответа:</b> 200</p> <p><b>Содержимое:</b> JSON Массив объектов</p> <p>[ <a href="#">Формат объекта</a>, <a href="#">Формат объекта</a> ]</p>
<b>Ответ при ошибке</b>	<p><b>Код ответа:</b> 401 UNAUTHORIZED</p> <p><b>Содержимое:</b> ""</p> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на создание объекта</p> <p>или</p> <p><b>Код ответа:</b> 500 Internal Server Error</p> <p><b>Содержимое:</b> { error : "Описание ошибки" }</p> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p>
<b>Пример использования</b>	<pre>\$.ajax({   url: "/objects",   dataType: "json",   type : "GET",   success : function(r) {     console.log(r);   } });</pre>

### 3.5. Передача объекта на согласование

<b>Наименование</b>	Передача объекта на согласование с подразделением УВО
<b>URL</b>	/approve
<b>Метод</b>	POST

<b>Параметры URL</b>	нет
<b>Данные</b>	<p>JSON объект</p> <pre>{   "object_id": "ID объекта",   "comment" : "Тест комментария" }</pre>
<b>Ответ при удачном запросе</b>	<p><b>Код ответа:</b> 200  <b>Содержимое:</b> ""</p>
<b>Ответ при ошибке</b>	<p><b>Code:</b> 400 Bad Request  <b>Content:</b> {Error:"описание ошибки"}</p> <p>Возникает в случае, если согласование объекта невозможно по каким-либо причинам</p> <p><b>Code:</b> 401 UNAUTHORIZED  <b>Content:</b> {Error:"no access"}</p> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на изменение объекта</p> <p><b>Code:</b> 500 Internal Server Error  <b>Content:</b> { error : "Описание ошибки" }</p> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p>
<b>Пример использования</b>	<pre>\$.ajax({   url: "/approve",   dataType: "json",   data : {     "object_id": "5213a389-ccda-11e6-9751-0cc47aac4166",     "comment": "Первичное согласование объекта"   },   type : "POST",   success : function(r) {     console.log(r);   } });</pre>

## 3.6. Отмена согласования объекта

<b>Наименование</b>	Снятие согласования объекта с подразделением УВО
<b>URL</b>	/unapprove
<b>Метод</b>	POST
<b>Параметры URL</b>	нет
<b>Данные</b>	JSON объект <pre>{   "object_id": "ID объекта",   "comment" : "Тест комментария" }</pre>
<b>Ответ при удачном запросе</b>	<b>Код ответа:</b> 200 <b>Содержимое:</b> ""
<b>Ответ при ошибке</b>	<b>Code:</b> 400 Bad Request <b>Content:</b> {Error:"описание ошибки"} <p>Возникает в случае, если согласование объекта невозможно по каким-либо причинам</p> <b>Code:</b> 401 UNAUTHORIZED <b>Content:</b> {Error:"no access"} <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на изменение объекта</p> <b>Code:</b> 403 Forbidden <b>Content:</b> {Error:"Exists alerts on this object"} <b>Code:</b> 500 Internal Server Error <b>Content:</b> { error : "Описание ошибки" } <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p>

<b>Пример использования</b>	<pre>\$.ajax({   url: "/unapprove",   dataType: "json",   data : {     "object_id": "5213a389-ccda-11e6-9751-0cc47aac4166",     "comment": "Закрытие договора"   },   type : "POST",   success : function(r) {     console.log(r);   } });</pre>
-----------------------------	--

### 3.7. Удаление объекта

Получение списка объектов с основными данными. Расширенные данные можно получить по запросу GET /objects/:id

<b>Наименование</b>	Удаление объекта
<b>URL</b>	/objects/:id
<b>Метод</b>	DELETE
<b>Параметры URL</b>	Обязательно: <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <pre>id:[UUID] Пример: 5213a389-ccda-11e6-9751-0cc47aac4166</pre> </div>
<b>Данные</b>	нет
<b>Ответ при удачном запросе</b>	Код ответа: 200 Content: <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <pre>{   "status": 0,   "object_id": %id%,   "deleted": true }</pre> </div>

<p><b>Ответ при ошибке</b></p>	<p><b>Код ответа:</b> 401 UNAUTHORIZED</p> <p><b>Содержимое:</b> ""</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на получение данных объекта</p> </div> <p>или</p> <p><b>Код ответа:</b> 500 Internal Server Error</p> <p><b>Содержимое:</b> { error : "Описание ошибки" }</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p> </div>
<p><b>Пример использования</b></p>	<pre>\$.ajax({   url: "/objects/5213a389-ccda-11e6-9751-0cc47aac4166",   type: "DELETE",   success: function(r) {     console.log(r);   } });</pre>

## 3.8. Формат объекта

Table 1. Список полей

Идентификатор	Тип	Наименование	Доп.	Обяз.
object_num	Строка	Пультный номер	Уникальное поле,	да
address	Объект <a href="#">Формат поля адрес</a>	Адрес объекта		да
category_id	Число	Категория объекта	Варианты поля Категория объекта(category_id): <a href="#">Категория объекта</a>	да
phone	Строка	Телефон на объекте		нет
name	Строка	Наименование объекта		да

Идентификатор	Тип	Наименование	Доп.	Обяз.
locked	Строка	Характеристика объекта, уязвимые места и места возможного проникновения		нет
district_id	UUID	Идентификатор подразделения		да
type_id	Число	Тип объекта	Варианты поля Тип объекта(type_id): <a href="#">Тип объекта</a>	да
contacts	Массив	Доверенные лица	Массив доверенных лиц	нет
zones	Массив	Зоны	Массив зон объекта	нет

Listing 1. Пример данных в json

```
{
  "object_num": "2", //Пультный номер. Обязательное уникальное поле.
  "address": { // Адрес объекта в виде строки или в формате поля адрес
    "text": "Адрес объекта", //Пультный номер. Обязательное уникальное поле.
    "lat": 0.0, // Координаты объекта, широта
    "lng": 0.0, // Координаты объекта, долгота
    "comment": "Комментарий" // Комментарий к адресу, например уточнение как
    ое здание или где вход
  },
  "category_id": 1, // Категория объекта по справочнику
  "phone": "21321312312321", // Телефон на объекте
  "name": "Аптека", // Наименование объекта
  "locked": "fdsadasdasd", // Заблокированные и уязвимые места
  "district_id": "5820ab00-3b53-480d-ba4a-bfdbb4eb603d", //Код Подразделения
  "type_id": 0, // Тип объекта по справочнику
  "contacts": [
    {
      "name": "Имя контакта",
      "phone": "Телефон контакта"
    }
  ],
  "zones": [
    {
      "zone_num": "Номер зоны", // Произвольный номер зоны, используется при пе
      редаче тревоги
      "name": "Наименование зоны",
      "description": "Описание зоны"
    }
  ]
}
```

### 3.8.1. Формат поля адрес

Table 2. Список полей

Идентификатор	Тип	Наименование	Доп.	Обяз.
text	Строка	Адрес объекта	Адрес в человеко-читаемом виде, например: г Москва, ул Ленина, д 1, кв 1. В случае отсутствия координат будет произведена попытка их определения на основе адреса.	да
lat	Float 64	Координаты объекта, широта		нет
lng	Float 64	Координаты объекта, долгота		нет
comment	Строка	Комментарий	Комментарий к адресу, например: уточнение какое здание или где вход, этаж и т.д.	нет

Listing 2. Пример данных в json

```
{
  "text": "Адрес объекта",
  "lat": 55.56789,
  "lng": 55.56789,
  "comment": "Комментарий"
}
```

## 3.9. Работа с файлами

### 3.9.1. Добавление файла к объекту

Наименование	Добавление файла к объекту
URL	/objects/:id/file
Метод	POST
Параметры URL	id - Идентификатор объекта

<b>Данные</b>	<p>JSON объект</p> <pre data-bbox="357 181 1461 434"> {   "name": "Наименование файла",   "description": "Описание файла",   "file": "base64 кодированный файл" } </pre>
<b>Ответ при удачном запросе</b>	<p><b>Код ответа:</b> 200</p> <p><b>Содержимое:</b></p> <pre data-bbox="357 629 1461 882"> {   "description": "Описание файла",   "id": "e9952ce6-e87e-11e6-8069-40167e6733b0", // идентификатор файла   "name": "Наименование файла" } </pre>
<b>Ответ при ошибке</b>	<p><b>Код ответа:</b> 401 UNAUTHORIZED</p> <p><b>Содержимое:</b> ""</p> <p data-bbox="357 1122 1461 1256">Возникает в случае, если пользователь не авторизован, либо у него нет прав на создание объекта</p> <p data-bbox="357 1290 416 1323">или</p> <p><b>Код ответа:</b> 500 Internal Server Error</p> <p><b>Содержимое:</b> { error : "Описание ошибки" }</p> <p data-bbox="357 1514 1461 1648">Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p>
<b>Пример использования</b>	

### 3.9.2. Получение данных

<b>Наименование</b>	Получение данных
---------------------	------------------

<b>URL</b>	/file/:id
<b>Метод</b>	GET
<b>Параметры URL</b>	<p>Обязательно:</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>id:[UUID] - Идентификатор файла Пример: 5213a389-ccda-11e6-9751-0cc47aac4166</p> </div>
<b>Данные</b>	нет
<b>Ответ при удачном запросе</b>	<p><b>Код ответа:</b> 200 <b>Содержимое:</b> Содержимое файла</p>
<b>Ответ при ошибке</b>	<p><b>Код ответа:</b> 401 UNAUTHORIZED <b>Содержимое:</b> ""</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на получение данных объекта</p> </div> <p>или</p> <p><b>Код ответа:</b> 500 Internal Server Error <b>Содержимое:</b> { error : "Описание ошибки" }</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p> </div>
<b>Пример использования</b>	

### 3.9.3. Удаление файла

<b>Наименование</b>	Получение данных
<b>URL</b>	/file/:id
<b>Метод</b>	DELETE

<b>Параметры URL</b>	<p>Обязательно:</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>id:[UUID] - Идентификатор файла  Пример: 5213a389-ccda-11e6-9751-0cc47aac4166</p> </div>
<b>Данные</b>	<p>нет</p>
<b>Ответ при удачном запросе</b>	<p><b>Код ответа:</b> 200  <b>Содержимое:</b> Содержимое файла</p>
<b>Ответ при ошибке</b>	<p><b>Код ответа:</b> 401 UNAUTHORIZED  <b>Содержимое:</b> ""</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на получение данных объекта</p> </div> <p>или</p> <p><b>Код ответа:</b> 500 Internal Server Error  <b>Содержимое:</b> { error : "Описание ошибки" }</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p> </div>
<b>Пример использования</b>	

## Раздел 4. Тревоги

### 4.1. Поддерживаемые запросы по работе с тревогами

Метод	Параметры	Результат	Описание
GET GET <a href="https://pakvcmk.ru/api/alerts">https://pakvcmk.ru/api/alerts</a>	нет	JSON Массив текущих тревог	Получение списка всех активных тревог
POST <a href="https://pakvcmk.ru/api/alert">https://pakvcmk.ru/api/alert</a>	JSON Объект: <a href="#">Формат передаваемой тревоги</a>	JSON объект <a href="#">Тикет принятой тревоги</a>	Создание новой тревоги по объекту
POST <a href="https://pakvcmk.ru/api/alert/:id/event">https://pakvcmk.ru/api/alert/:id/event</a>	JSON Объект <a href="#">Формат события по тревоге</a>	JSON объект со временем передачи события:  <pre>{ "event_id": "ID события" "delivery_time":"0001-01-01T00:00:00Z" }</pre>	Отправка события по тревоге
POST <a href="https://pakvcmk.ru/api/alert/:id/cancel">https://pakvcmk.ru/api/alert/:id/cancel</a>	JSON объект с комментарием о причине отмены:  <pre>{ "comment": "причина отмены" }</pre>	code 200 В случае успеха, code 4**, 5** в случае ошибки и JSON объект:  <pre>{ "Error": "Сообщение об ошибке" }</pre>	Отмена ранее переданной тревоги

### 4.2. Создание новой тревоги по объекту

<b>Наименование</b>	Создание новой тревоги по объекту
<b>URL</b>	/alert
<b>Метод</b>	POST
<b>Параметры URL</b>	нет
<b>Данные</b>	JSON объект: <a href="#">Формат передаваемой тревоги</a>
<b>Ответ при удачном запросе</b>	Код ответа: 200  Содержимое: JSON объект: <a href="#">Тикет принятой тревоги</a>

<b>Ответ при ошибке</b>	<p><b>Код ответа:</b> 401 UNAUTHORIZED</p> <p><b>Содержимое:</b> ""</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на создание объекта</p> </div> <p>или</p> <p><b>Код ответа:</b> 500 Internal Server Error</p> <p><b>Содержимое:</b> { error : "Описание ошибки" }</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p> </div>
<b>Пример использования</b>	

### 4.3. Отправка нового события по тревоге

<b>Наименование</b>	Отправка нового события по тревоге
<b>URL</b>	/alert/:id/event
<b>Метод</b>	POST
<b>Параметры URL</b>	:id - Идентификатор тревоги
<b>Данные</b>	JSON объект: <a href="#">Формат события по тревоге</a>
<b>Ответ при удачном запросе</b>	<p><b>Код ответа:</b> 200</p> <p><b>Содержимое:</b> JSON объект со временем передачи события:</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <pre> {   "event_id": "ID события",   "delivery_time": "0001-01-01T00:00:00Z" } </pre> </div>

<b>Ответ при ошибке</b>	<p><b>Код ответа:</b> 401 UNAUTHORIZED</p> <p><b>Содержимое:</b> ""</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на создание объекта</p> </div> <p>или</p> <p><b>Код ответа:</b> 500 Internal Server Error</p> <p><b>Содержимое:</b> { error : "Описание ошибки" }</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p> </div>
<b>Пример использования</b>	

## 4.4. Получение списка активных тревог

<b>Наименование</b>	Получение списка активных тревог
<b>URL</b>	/alerts
<b>Метод</b>	GET
<b>Параметры URL</b>	нет
<b>Данные</b>	нет
<b>Ответ при удачном запросе</b>	<p><b>Код ответа:</b> 200</p> <p><b>Содержимое:</b> JSON Массив из <a href="#">Формат тревоги</a></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>[ <a href="#">Формат тревоги</a>, <a href="#">Формат тревоги</a> ]</p> </div>

<b>Ответ при ошибке</b>	<p><b>Код ответа:</b> 401 UNAUTHORIZED</p> <p><b>Содержимое:</b> ""</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на создание тревоги по объекту</p> </div> <p>или</p> <p><b>Код ответа:</b> 500 Internal Server Error</p> <p><b>Содержимое:</b> { error : "Описание ошибки" }</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p> </div>
<b>Пример использования</b>	

## 4.5. Отмена ранее переданной тревоги

<b>Наименование</b>	Отмена ранее переданной тревоги
<b>URL</b>	<a href="https://pakvcmk.ru/api/alert/:id/cancel">https://pakvcmk.ru/api/alert/:id/cancel</a>
<b>Метод</b>	POST
<b>Параметры URL</b>	:id - Идентификатор тревоги
<b>Данные</b>	<p>JSON объект причиной отмены:</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <pre>{   "comment": "причина отмены" }</pre> </div>
<b>Ответ при удачном запросе</b>	<p><b>Код ответа:</b> 200</p> <p><b>Содержимое:</b> нет</p>

<p><b>Ответ при ошибке</b></p>	<p><b>Код ответа:</b> 401 UNAUTHORIZED</p> <p><b>Содержимое:</b> ""</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на отмену тревоги</p> </div> <p>или</p> <p><b>Код ответа:</b> 500 Internal Server Error</p> <p><b>Содержимое:</b> { Error : "Описание ошибки" }</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает при указании ошибочных данных, неправильного формата JSON или UUID в полях</p> </div>
<p><b>Пример использования</b></p>	<pre>\$.ajax({   url: "/api/alert/5213a389-ccda-11e6-9751-0cc47aac4166/cancel",   dataType: "json",   data : {     "comment": "Получен ответ от собственника"   },   type : "POST",   success : function(r) {     console.log(r);   } });</pre>

## 4.6. Форматы данных

### 4.6.1. Формат тревоги

```
{
  "alert_id": "b68eea07-51b3-11e7-b659-0cc47aac4166", // Уникальный идентификатор
тревоги
  "object_id": "bfe91973-5117-11e7-9a59-0cc47aac4166", // Уникальный идентификатор
объекта
  "status_id": 2, // Статус тревоги
  "delivery_time": "2017-06-15T10:16:36.19Z", // Время доставки тревоги в систему
  "take_time": "2017-06-15T11:38:32.038Z", // Время взятия тревоги в работу ДО У
ВО
  "close_time": "0001-01-01T00:00:00Z", // Время закрытия тревоги ДО УВО или о
тмены МК
  "is_false": false, // Флаг ложной тревоги
```

```

"comment": "", // Комментарий по тревоге
"district_id": "d8d7ef2c-5112-11e7-9a55-0cc47aac4166", // Подразделение, в кото
рое была передана тревога
"region_id": "c2608f76-7129-414c-aff6-646423bc298a", // Регион, в который была
передана тревога
"company_id": "8b9c1094-1516-11e7-aa20-0cc47aac4166", // Уникальный идентификат
ор компании
"eventList": [ // Массив событий по объекту
{
"event_id": "b69804f8-51b3-11e7-b65a-0cc47aac4166",
>alert_id": "b68eea07-51b3-11e7-b659-0cc47aac4166",
>delivery_time": "2017-06-15T13:16:36.250132925+03:00",
>comment": "",
>type_id": 2,
>zone_id": "fb409dab-5117-11e7-9a68-0cc47aac4166",
},
{
"event_id": "b698059e-51b3-11e7-b65b-0cc47aac4166",
>alert_id": "b68eea07-51b3-11e7-b659-0cc47aac4166",
>delivery_time": "2017-06-15T13:16:36.250150176+03:00",
>comment": "",
>type_id": 7,
>zone_id": "fb409da8-5117-11e7-9a67-0cc47aac4166"
}
],
// Массив действия дежурного
"reactionList": [
{
"reaction_type": 1, // Тип действия
"reaction_group_name": "Гбр 1", // Наименование экипажа
"delivery_time": "2017-06-15T15:36:16.487133724+03:00", // Время события
"comment": "Тест", // Комментарий либо доклад Г.З.
},
{
"reaction_type": 2,
"reaction_group_name": "Гбр 1",
"delivery_time": "2017-06-15T15:36:24.021047133+03:00",
"comment": "тест",
},
{
"reaction_type": 3,
"reaction_group_name": "Гбр 1",
"delivery_time": "2017-06-15T15:36:30.432696998+03:00",
"comment": "доклад",
}
]
}

```

#### 4.6.2. Формат передаваемой тревоги

```
{
  "object_id": "ID объекта в системе",
  "events":
    [
      {
        "zone_id": "ID зоны", //Необязательное поле
        "type_id": 0 //Тип события,
        "comment": "Комментарий события"
      }
    ]
}
```

#### 4.6.3. Тикет принятой тревоги

```
{
  "alert_id": "ID Тревоги",
  "delivery_time": "0001-01-01T00:00:00Z" //Время доставки тревоги в систему,
  "district_id": "ID подразделения, обрабатывающего тревогу"
}
```

#### 4.6.4. Формат события по тревоге

```
{
  "zone_id": "ID зоны",
  "type_id": 1, //Тип события
  "comment": "Комментарий события"
}
```

#### 4.6.5. Коды событий

- 2  
Нападение
- 4  
Пожар
- 5  
Принуждение
- 7  
Подбор кода
- 9



## Раздел 5. Регионы / Подразделения

Метод	Параметры	Результат	Описание
GET <a href="https://pakvcmk.ru/api/regions">https://pakvcmk.ru/api/regions</a>	нет	JSON Массив регионов	Получение списка регионов, доступных для данной компании
GET <a href="https://pakvcmk.ru/api/districts/:id">https://pakvcmk.ru/api/districts/:id</a>	нет	JSON массив подразделений	Получение списка подразделений по региону

### 5.1. Получение списка регионов

Получение списка регионов, доступных для данной компании

<b>Наименование</b>	Получение списка регионов, доступных для данной компании
<b>URL</b>	/regions
<b>Метод</b>	GET
<b>Параметры URL</b>	нет
<b>Данные</b>	нет
<b>Ответ при удачном запросе</b>	<p>Код ответа: 200</p> <p>Содержимое: JSON Массив регионов</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"><a href="#">Формат списка регионов</a></div>

<p><b>Ответ при ошибке</b></p>	<p><b>Код ответа:</b> 401 UNAUTHORIZED</p> <p><b>Содержимое:</b> ""</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на создание объекта</p> </div> <p>или</p> <p><b>Код ответа:</b> 500 Internal Server Error</p> <p><b>Содержимое:</b> { error : "Описание ошибки" }</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p> </div>
<p><b>Пример использования</b></p>	<pre>\$.ajax({   url: "/regions",   dataType: "json",   type: "GET",   success: function(r) {     console.log(r);   } });</pre>

## 5.2. Формат списка регионов

```
[
  {
    "region_id": "Идентификатор региона [uuid]",
    "name": "Наименование подразделения"
  },...
]
```

## 5.3. Получение списка подразделений по региону

Получение списка регионов, доступных для данной компании

<p><b>Наименование</b></p>	<p>Получение списка всех объектов</p>
<p><b>URL</b></p>	<p>/districts/:id</p>

<b>Метод</b>	GET
<b>Параметры URL</b>	:id - Идентификатор региона
<b>Данные</b>	нет
<b>Ответ при удачном запросе</b>	<p>Код ответа: 200</p> <p>Содержимое: JSON Массив регионов</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>[ <a href="#">Формат подразделения</a>, <a href="#">Формат подразделения</a> ]</p> </div>
<b>Ответ при ошибке</b>	<p>Код ответа: 401 UNAUTHORIZED</p> <p>Содержимое: ""</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Возникает в случае, если пользователь не авторизован, либо у него нет прав на создание объекта</p> </div> <p>или</p> <p>Код ответа: 500 Internal Server Error</p> <p>Содержимое: { error : "Описание ошибки" }</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Возникает при указании ошибочных данных, неправильного формата JSON, даты или UUID в полях</p> </div>
<b>Пример использования</b>	<pre>\$.ajax({   url: "/districts/41643c07-ee81-4d68-bf15-c4ff4a2c8a06",   dataType: "json",   type : "GET",   success : function(r) {     console.log(r);   } });</pre>

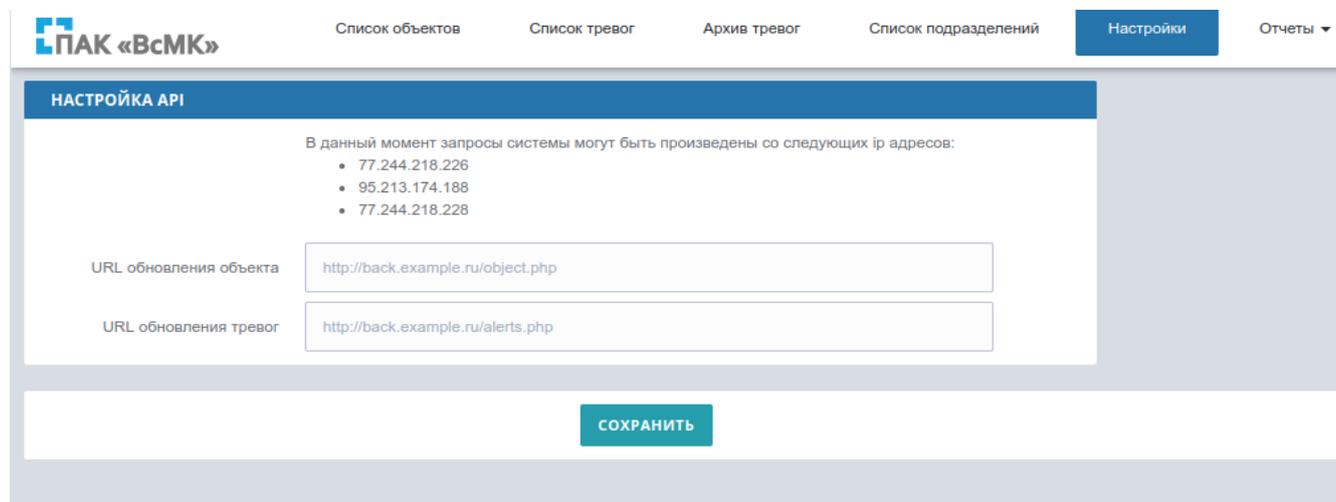
## 5.4. Формат подразделения

```
{  
  "district_id": "[uuid]", //Идентификатор подразделения  
  "region_id": "[uuid]", //Идентификатор региона  
  "name": [string], //Наименование подразделения  
  "address": [string], // Адрес подразделения  
  "phone":[string], // Телефоны подразделения  
  "time_zone" :[integer] // Смещение часового пояса подразделения относительно GMT  
},
```

## Раздел 6. Передача изменений в мониторинговую компанию

Передача изменений состояния объекта в ПО мониторинговой компании осуществляется вызовом POST запроса по URL, указанном в настройках АРМ.

Настройка url производится в веб интерфейсе на странице <https://pakvcmk.ru/#!/settings>



The screenshot shows the 'ПАК «ВсМК»' web interface. The top navigation bar includes 'Список объектов', 'Список тревог', 'Архив тревог', 'Список подразделений', 'Настройки', and 'Отчеты'. The main content area is titled 'НАСТРОЙКА API'. It contains the following information:

- Text: 'В данный момент запросы системы могут быть произведены со следующих ip адресов:'
- List of IP addresses:
  - 77.244.218.226
  - 95.213.174.188
  - 77.244.218.228
- Form field: 'URL обновления объекта' with value 'http://back.example.ru/object.php'
- Form field: 'URL обновления тревог' with value 'http://back.example.ru/alerts.php'
- Button: 'СОХРАНИТЬ'

Вызов данных url производится системой при любом изменении объекта или тревоги.

Для тревог вызов будет при взятии в работу, вызове ГЗ, закрытии тревоги и всех событиях.

Формат данных совпадает с форматом [Формат объекта](#) и [Формат тревоги](#)

# Раздел 7. Справочные данные

## 7.1. Статусы объекта

Поле status в карточках объектов

- 1 Новый объект
- 2 Объект на согласовании
- 3 Объект на охране
- 4 Отклонен
- 5 Отключен
- 6 На охране / проверка
- 7 На охране / изменения отклонены

## 7.2. Статус тревоги

Поле status в данных тревог

- Статусы > 90 Означают закрытую тревогу
- 0: Новая тревога
- 1: Принята
- 2: В работе
- 98: Отменена МК без взятия в работу Д.О.
- 99: Закрыта Д.О.

## 7.3. Действие оператора

Поле reaction\_type в reactionList тревоги

- 1: Вызов группы задержания
- 2: Прибытие группы задержания
- 3: Доклад группы задержания
- 4: Отмена вызова группы

## 7.4. Коды событий

- 2: Нападение
- 4: Пожар
- 5: Принуждение
- 7: Подбор кода
- 9: Проникновение

## 7.5. Категория объекта

Всего возможно 2 варианта:

- 0: Обычный объект
- 1: Особо важный объект

## 7.6. Тип объекта

- 1: квартира
- 2: объект
- 3: МХИГ

## 7.7. Формат ответа при ошибке в REST API

При ошибке обработки запроса через http api возвращается структура в json:

```
{
  "Error": "Тестовое описание ошибки",
  "ErrorCode": integer // Код ошибка по справочнику
}
```

## 7.8. Коды ошибок при работе через REST API

Актуальные данные по кодам ошибок можно получить по адресу:

```
https://pakvcmk.ru/api/errors
```

Запрос может быть выполнен без авторизации

Данные коды ошибок передаются, если в запросе передан валидный токен, иначе запрос вернет код ответа http 401 и json объект вида :

```
{ "Error": "Not Authorized" }
```

### 7.8.1. Коды ошибок

- 1: "Нет доступа"
- 2: "Некорректный запрос"
- 3: "Недостаточно данных"
- 4: "Пультный повторяется"
- 5: "Не найден объект или тревога"
- 6: "Объект на проверке"
- 7: "Объект не на охране"
- 8: "Отсутствует договор по региону"
- 9: "Отсутствует подразделение"
- 10: "Отсутствует регион"
- 11: "По объекту есть тревоги"
- 12: "Тревога закрыта"
- 13: "Тревога отменена"
- 14: "Зона не найдена"
- 15: "Ошибка при обработке запроса"