

????? ????????

Разберем принцип взаимодействия с API на самом простом примере – поиске региона в справочнике регионов. Например, нам требуется найти регион Яндекса с названием “Москва”. Для поиска указанного региона по названию необходимо выполнить GET-запрос на конечную точку <https://moab-apis.ru/api/v1/region/yandex>. В GET-параметре query нужно указать название (или часть названия) региона (регистр символов значения не имеет), например <https://moab-apis.ru/api/v1/region/yandex?query=москва>. Запросы к справочнику регионов, как и остальные запросы, требуют наличия заголовка X-API-Key, поэтому просто выполнить в браузере этот запрос не получится. В результате успешного выполнения запроса вы получите следующий результат:

```
application/json
1  [
2    {
3      "name": "Аничково, Москва и Московская область, Щелковский район",
4      "code": "117175"
5    },
6    {
7      "name": "Барвиха, Москва и Московская область, Одинцовский район",
8      "code": "21652"
9    },
10   {
11     "name": "Быково, Москва и Московская область, Раменский район",
12     "code": "10718"
13   },
14   {
15     "name": "Железнодорожный, Москва и Московская область",
16     "code": "21622"
17   },
18   {
19     "name": "Заречье, Москва и Московская область, Одинцовский район",
20     "code": "21653"
21   },
```

Как видно из скриншота, вам возвращается JSON-массив со списком всех регионов, в названии которых было найдено слово “москва”. Нужный вам регион вы можете найти путем сравнения названий в вашем клиентском коде. Чем точнее вы укажете название региона – тем меньше результатов будет вам выдано.

JSON-структура ответа обычным текстом выглядит вот так:

```
[
  {
    "name": "Аничково, Москва и Московская область, Щелковский район",
    "code": "117175"
  },
  {
    "name": "Барвиха, Москва и Московская область, Одинцовский район",
    "code": "21652"
  },
  {
    "name": "Быково, Москва и Московская область, Раменский район",
    "code": "10718"
  },
  {
    "name": "Железнодорожный, Москва и Московская область",
    "code": "21622"
  },
  {
    "name": "Заречье, Москва и Московская область, Одинцовский район",
    "code": "21653"
  },
  {
    "name": "Монино, Москва и Московская область, Щелковский район",
    "code": "21636"
  },
  {
    "name": "Москва",
    "code": "213"
  },
  ...
]
```

Найдя необходимый регион в справочнике, вы можете использовать значение поля code в дальнейших запросах к Yandex Wordstat и другим сервисам, которые связаны с Яндексом.

В ответ на запрос вам может вернуться какой-то HTTP-код ответа из этого списка:

Responses

> 200 OK

> 400 Bad Request

> 401 Unauthorized

> 404 Not Found

Все коды ответа, кроме 200, являются кодами ошибок. Каждый код ошибки означает ту или иную проблему, в таблице ниже смотрите описание возможных проблем.

Код ответа	Потенциальная проблема	Текст ответа
200	Проблем нет	В тексте ответа возвращается результат запроса.
400	Неверно указаны параметры запроса.	Проверьте параметры запроса на соответствие спецификации API.
401	Не указан или указан неизвестный заголовок X-API-Key.	Пустой текст ответа. Не относится к справочникам.
404	Фраза не найдена	Соответствует ответу Wordstat "Нет подходящих запросов. Попробуйте изменить формулировку." Возвращает стандартный формат ошибки (см. ниже)..
404	Метод не найден	Пустой текст ответа. Указана неверная конечная точка API.
422	Неверно задан запрос	Неверно указана проверяемая фраза - вордстат сам возвращает ошибку

Во всех случаях, когда возврат ошибки подразумевает возврат поясняющего сообщения - вам будет возвращена JSON-конструкция вида:

```
{
  "id": "fa18aecb-ac5c-443b-95a2-ef195949ecc7",
  "error_message": "NotFound",
  "instance": "/api/v1/region/check",
  "invalid_data": null
}
```

Здесь:

- **id** – уникальный идентификатор ошибки. Сообщите его вместе с сообщением об ошибке для облегчения сотрудникам техподдержки поиска конкретного случая возникновения ошибки.
 - **error_message** – текстовое обозначение кода ошибки
 - **instance** – название конечной точки API, к которой было обращение, приведшее к ошибке
 - **invalid_data** – массив строк, может содержать несколько строк, описывающих одну или несколько возникших ошибок. Nullable.
-

Revision #5

Created 7 October 2025 09:45:52 by Артем

Updated 12 October 2025 18:11:40 by Артем