

Тестовое задание

Практика

Имеется API для выдачи регионов с одним методом:

GET /1.0/regions — возвращает список регионов в формате:

```
1 {
2   "total": 1, // Общее количество регионов в базе
3   "items": [
4     {
5       "id": 1, // Уникальный идентификатор региона
6       "name": "Новосибирск", // Название региона
7       "code": "novosibirsk", // Код региона
8       "country": {
9         "name": "Россия", // Название страны региона
10        "code": "ru" // Код страны региона
11      }
12    }
13  ]
14 }
```

Доступные query-параметры

Название	Описание	Пример
q	<p>Произвольная строка для нечёткого поиска* по названию региона</p> <ul style="list-style-type: none">• Минимум — 3 символа• Регистр не имеет значения• Если передан этот параметр, все остальные	<pre>GET /1.0/regions? q=Новосибирск</pre>

	параметры игнорируются	
country_code	Код страны для фильтрации <ul style="list-style-type: none"> • Может принимать значения: ru, kg, kz, cz • По умолчанию отображаются регионы из всех стран 	GET /1.0/regions?country_code=ru
page	Порядковый номер страницы <ul style="list-style-type: none"> • Минимальное значение — 1 • Значение по умолчанию — 1 	GET /1.0/regions?page=2
page_size	Количество элементов на странице <ul style="list-style-type: none"> • Может принимать значения: 5, 10, 15 • Значение по умолчанию — 15 	GET /1.0/region?page=3&page_size=5

* Нечёткий поиск — поиск по вхождению подстроки в строку. Например, если поискать «рск», в ответе должны отобразиться регионы: **Новосибирск**, **Магнитогорск**, **Орск** и т.д.

API доступно по адресу <https://regions-test.2gis.com/1.0/regions>. Для отображения JSON в браузере можно использовать расширение «[JSON Formatter](#)».

Задание 1

Протестируйте API для выдачи регионов и оформите баг-репорт с найденными ошибками.

Результат пришлите в одном из следующих форматов: Google Docs, Microsoft Word, LibreOffice Writer, Markdown, PDF.

Задание 2

Составьте тест-кейсы и напишите по ним авто-тесты, проверяющие корректность работы API для выдачи регионов. Каждый авто-тест должен состоять как минимум из двух шагов: формирование запроса к серверу и сравнение результата с ожидаемым значением проверяемого тест-кейса.

Пример самого простого авто-теста на языке Python 3:

```
1 import json
2 from urllib import request
3
4 def test_total_count():
5     res = request.urlopen('https://regions-test.2gis.com/1.0/r
6     egions')
7     body = json.loads(res.read().decode('utf-8'))
8     assert body['total'] == 22
9
10 test_total_count()
```

Для написания авто-тестов можно использовать **любой язык программирования и любые библиотеки**. Например, для Python можно использовать библиотеки [requests](#) и [unittest](#), для C# — [RestSharp](#) и [NUnit](#), для Java — [Unirest](#) и [JUnit](#), и т.д.

В качестве результата должен быть исходный код с тестами.