

Тестовые сценарии

Так как тестирую на Windows curl запросы, то буду использовать cmd, чтобы не нагружать тест-кейсы, будет подразумеваться, что мы уже находимся cmd(предусловие). Также в дальнейшем использую сервисы <https://codebeautify.org/jsonviewer> для более понятного отображения результата и <https://jsoneditoronline.org/#> для отображения количества объектов внутри json.

1) Получить все имена из базы

1.Выполнить GET запрос с помощью curl:

```
curl "http://rest.test.ivu.ru/v2/characters" -u  
test@mail.com:APZrVp83vFNk5F
```

2. Скопировать полученный ответ от сервера и отформатировать в <https://codebeautify.org/jsonviewer>

Ожидаемый результат:

```
{  
  "result": [  
    {  
      "education": "High school graduate; military training  
(Chuck Chandler only)",  
      "height": 187,  
      "identity": "Secret",  
      "name": "3-D Man",  
      "other_aliases": "None",  
      "universe": "Marvel Universe",  
      "weight": 90  
    },  
    {  
      "education": "High school graduate; military training  
(Chuck Chandler only)",  
      "height": 187,  
      "identity": "Secret",
```

```

    "name": "3-D Man",
    "other_aliases": "None",
    "universe": "Marvel Universe",
    "weight": 90
  },
  {
    "education": "High school graduate; military training
(Chuck Chandler only)",
    "height": 187,
    "identity": "Secret",
    "name": "3-D Man",
    "other_aliases": "None",
    "universe": "Marvel Universe",
    "weight": 90
  }

```

Будет огромный json объект с 307 объектами внутри, первые три объекта герой с именем “3-D Man”.

2) Получить персонажа с именем состоящим больше чем одно слово

1. Из codebeautify выбрать героя с именем имеющим более чем одно слово, например "Baron Zemo (Heinrich Zemo)"
2. Выполнить GET запрос с помощью curl:

curl

"http://rest.test.ivi.ru/v2/character?name=Baron+Zemo+(Heinrich+Zemo)" -u test@mail.com:APZrVp83vFNk5F

Ожидаемый результат:

```

{"result":{"education":"College
graduate","height":177,"identity":"Publicly
known","name":"Baron Zemo (Heinrich
Zemo)","other_aliases":"Iron Cross, Citizen V, Mark
Evanier, Phoenix","universe":"Marvel
Universe","weight":82.35000000000001}}

```

3) Получить персонажа с именем состоящего не более чем из одного слова

1. Выбрать в codebeautify героя с именем состоящего из одного слова, например "Bastion"
2. Выполнить GET запрос с помощью curl:

```
curl "http://rest.test.ivy.ru/v2/character?name=Bastion" -u test@mail.com:APZrVp83vFNk5F
```

Ожидаемый результат:

```
{
  "result": {
    "education": "Inapplicable",
    "height": 190,
    "identity": "Secret",
    "name": "Bastion",
    "other_aliases": "Template, Sebastion Gilberti, Nimrod, Master Mold",
    "universe": "Marvel Universe",
    "weight": 168.75
  }
}
```

4) Получить персонажа с пустым параметром name

1. Выполнить GET запрос с помощью curl с пустым параметром name:

```
curl "http://rest.test.ivy.ru/v2/character?name=" -u test@mail.com:APZrVp83vFNk5F
```

Ожидаемый результат: {"error": "No such name"}

5) Получить персонажа с неверно указанным параметром name

1. Выбрать в codebeautify героя с именем состоящего из одного слова, например "Bastion"
2. Выполнить GET запрос с помощью curl с неверным именем в параметре name:

curl

"http://rest.test.ivi.ru/v2/character?name=Bastiin" -u test@mail.com:APZrVp83vFNk5F

Ожидаемый результат: {"error": "No such name"}

6) Получить персонажа используя произвольный регистр в параметре name

1. Выбрать в codebeautify героя с именем состоящего из одного слова, например "Bastion"
2. Выполнить GET запрос с помощью curl, используя только заглавные буквы в имени в параметре name:

curl

"http://rest.test.ivi.ru/v2/character?name=BASTION" -u test@mail.com:APZrVp83vFNk5F

Ожидаемый результат: {"error": "No such name"}

7) Получить персонажа не указав "+" в параметре name

1. Выбрать в codebeautify героя с именем состоящего из двух слов, например "Baron Strucker"
2. Выполнить GET запрос с помощью curl, не используя "+" между словами в параметре name:

curl

"http://rest.test.ivi.ru/v2/character?name=BaronStrucker" -u test@mail.com:APZrVp83vFNk5F

Ожидаемый результат: {"error": "No such name"}

DELETE

8) Удаление персонажа из базы

1. Выбрать в codebeautify героя с именем состоящего из одного слова, например "Selene"
2. Выполнить DELETE запрос с помощью curl:

```
curl -X DELETE
```

```
"http://rest.test.ivy.ru/v2/character?name=Selene" -u  
test@mail.com:APZrVp83vFNk5F
```

Ожидаемый результат: {"result":"Hero Selene is deleted"}

9) Удаление уже удаленного персонажа из базы

1. Выбрать уже удаленного персонажа из предыдущего кейса
2. Выполнить DELETE запрос с помощью curl:

```
curl -X DELETE
```

```
"http://rest.test.ivy.ru/v2/character?name=Selene" -u  
test@mail.com:APZrVp83vFNk5F
```

Ожидаемый результат:
{"error":"No such name"}

10) Добавление персонажа в базу

1. Выполнить POST запрос с помощью curl:

```
curl -X POST -H "Content-type:application/json" -u  
test@mail.com:APZrVp83vFNk5F -d
```

```
{"name\":\"Postd\", \"universe\":\"Marvel  
Universe\", \"education\":\"School\",  
\"weight\":\"104\", \"height\":\"1.90\",  
\"identity\":\"Publicly known\"}"
```

```
"http://rest.test.ivy.ru/v2/character"
```

Ожидаемый результат:

```
{"result":{"education":"School","height":1.9,"identity":"Publicly known","name":"Postd","universe":"Marvel Universe","weight":104.0}}
```

11) Добавление уже существующего персонажа в базу

1. Повторим предыдущий POST запрос из 10 тест-кейса.

2. Выполнить POST запрос с помощью curl:

```
curl -X POST -H "Content-type:application/json" -u test@mail.com:APZrVp83vFNk5F -d {"name":"Postd", "universe":"Marvel Universe", "education":"School", "weight":104, "height":1.9, "identity":"Publicly known"} "http://rest.test.ivu.ru/v2/character"
```

Ожидаемый результат:

```
{"error":"Postd is already exists"}
```

12) Добавление персонажа с пустыми параметрами в теле запроса

1. Выполнить POST запрос с помощью curl, убрав все значения параметров в теле запроса:

```
curl -X POST -H "Content-type:application/json" -u test@mail.com:APZrVp83vFNk5F -d {"name":"","universe":"","education":"","weight":"","height":""} "http://rest.test.ivu.ru/v2/character"
```

Ожидаемый результат:

```
{"error":"weight: ['Not a valid number.'], education: ['Length must be between 1 and 350.'], height: ['Not a valid number.'], universe: ['Length must be
```

between 1 and 350.'], name: ['Length must be between 1 and 350.']]}

13)Добавление нового персонажа без обязательных параметров

1. Выполнить POST запрос с помощью curl, удалив все параметры и их значения кроме обязательного name:

```
curl -X POST -H "Content-type:application/json"
-u test@mail.com:APZrVp83vFNk5F -d
{"name\":\"sleeping\"}
"http://rest.test.ivu.ru/v2/character"
```

Ожидаемый результат:

```
{"result":{"name":"sleeping"}}
```

14)Изменение параметров в созданном персонаже

1. Выбрать уже созданного с помощью POST запроса персонажа
2. Выполнить PUT запрос с помощью curl,изменив необязательный параметр, например "education":

```
curl -X PUT -H "Content-type:application/json" -u
test@mail.com:APZrVp83vFNk5F -d
{"name\":\"kfssdf\", \"universe\":\"Marvel
Universe\", \"education\":\"Put school\",
\"weight\":\"104\", \"height\":\"1.90\"}
"http://rest.test.ivu.ru/v2/character"
```

Ожидаемый результат:

```
{"result":{"education":"Put
school","height":1.9,"name":"kfssdf","universe":"Mar
vel Universe","weight":104.0}}
```

15) Изменение параметров несуществующего персонажа

1. Выполнить PUT запрос с помощью curl,изменив обязательный параметр name и введя значение несуществующего имени, например “kf”:

```
curl -X PUT -H "Content-type:application/json" -u
test@mail.com:APZrVp83vFNk5F -d
{"name\":\"kf\", \"universe\":\"Marvel
Universe\", \"education\":\"Put school\",
\"weight\":\"104\", \"height\":\"1.90\"}
"http://rest.test.ivu.ru/v2/character"
```

Ожидаемый результат:

```
{"error":"No such name"}
```

16)Изменение параметров персонажа из базы

1. Выбрать персонажа, который изначально был в базе данных, например “Beak”
2. Выполнить PUT запрос с помощью curl,изменив необязательные параметры, например “weight” и “height”:

```
curl -X PUT -H "Content-type:application/json" -u
test@mail.com:APZrVp83vFNk5F -d
{"education": \"Some college-level
courses\", \"height\": \"190\", \"identity\":
\"Secret\", \"name\":\"Beak\", \"other_aliases\":\
\"Blackwing\", \"universe\":\"Marvel
Universe\", \"weight\":\"90\"}
"http://rest.test.ivu.ru/v2/character"
```

Ожидаемый результат:

```
{"result":{"education":"Some college-level
courses", "height":190.0, "identity":"Secret", "name":"
Beak", "other_aliases":"Blackwing", "universe":"Marv
el Universe", "weight":90.0}}
```


Обязательно сохраните или запомните какие параметры были до, для сравнения результатов.

17) Сброс коллекции в первоначальное состояние

Прежде чем выполнять данный кейс, нужно пройти кейсы с POST запросами, иначе нельзя будет проверить и проверить что последние записи внесены в базу.

1. Выполнить POST запрос с помощью curl:
`curl -X POST "http://rest.test.ivi.ru/v2/reset" -u test@mail.com:APZrVp83vFNk5F`
2. После выполнить GET запрос с помощью curl:
3. `curl "http://rest.test.ivi.ru/v2/characters" -u test@mail.com:APZrVp83vFNk5F`

Ожидаемый результат:

После выполнения данной команды будет выведена вся база, пролистнуть до самого низа и проверить, что нет созданных записей из предусловия.