

Тестовое задание состоит из двух частей, которые показывают разные умения:

1. Общение со студентами (важнее всего)
2. Работа с автотестами постмана (вы или уже умеете это делать, или умеете гуглить)

Начните с 1 части тестового задания. Когда сделаете ее, пришлите результат мне на эл почту ok.molechka@gmail.com. Заголовок письма: «Тестовое по постману от <Имя Фамилия>»

Если по первой части всё ок, тогда делайте вторую. Для второй части я взяла самые сложные задания с курса. Если вы справитесь с ними, то и простые осилите)) Если просто хочется попробовать свои силы в автотестах, можете сделать эту часть «для себя». Но меня она будет волновать только в том случае, если вы хотите и умеете учить, то есть давать хорошую обратную связь студентам.

1 часть

Задача для студента — проверить, что в методе <http://users.bugred.ru/tasks/rest/getuserfull> (post-запрос) правильно возвращаются поля «cat» и «name». Вот, например, для запроса

```
{  
  
  "email": "test_cu_11@mail.com"  
  
}
```

Это «Красавчик» и «Маруся» (на момент написания этого сообщения). Надо написать pass + fail тесты (роняем тест, чтобы убедиться, что он вообще работает), сдать скриншот прогона.

Ответ студента:

POST

▼

http://users.bugred.ru/tasks/rest/getuserfull

Params

Authorization

Headers (11)

Body ●

Pre-request Script

Tests ●

Settings

```

1  pm.test("Your test name", function () {
2      var jsonData = pm.response.json();
3      pm.expect(jsonData.cat).to.eql("Маруся");
4  });
5
6  pm.test("Your test name 2", function () {
7      var jsonData = pm.response.json();
8      pm.expect(jsonData.name).to.eql("Красавчик");
9  });
10
11 pm.test("Your test name 3", function () {
12     var jsonData = pm.response.json();
13     pm.expect(jsonData.cat).to.eql("Манго");
14 });

```

Body

Cookies (1)

Headers (9)

Test Results (2/3)

⌐ Status: 200 OK

All

Passed

Skipped

Failed

PASS

Your test name

PASS

Your test name 2

FAIL

Your test name 3 | AssertionError: expected 'Маруся' to deeply equal 'Манго'

Какие будут ваши комментарии к работе? Пришлите мне ответ для студента. Не ваши размышления для меня, а ровно то, что вы в системе дистанционного обучения скажете студенту.

Помните, что студента надо сначала за что-то похвалить, а потом указать, что необходимо исправить.

=====

2 часть

1. Тестируем Shop — <https://testbase.atlassian.net/wiki/spaces/SHOP/overview>

Ваша задача — проверить, что товар реально создается. Но это должен быть полный цикл правильных тестов — которые не зависят от текущего состояния базы и не плодят лишние сущности, а чистят за собой. Так что цикл включает в себя:

- Создать товар через метод *create*. (*мы не тестируем создание товара целиком, прогоняя разные варианты заполнения полей, поэтому это просто должен быть успешно выполняющийся запрос*). Сделать тест на *id* и *name* (план-минимум), а то вдруг упадет с ошибкой, а мы и не в курсе?
- Проверить, что товар создан — дернуть по нему метод *get*. И в тесте убедиться, что *id* и *name* правильные
- Удалить товар — чистим за собой!
- Проверить, что товар удален — снова *get* и снова тест на ответ.
- Не забыть удалить за собой переменную, в которую вы сохранили *id* товара!

Что сдавать

- Коллекцию тестов в формате json. Название коллекции — «ДЗ по постману»

2. Тестируем Shop

Ваша задача — проверить, что поиск вернет несколько товаров, а не только один. Для этого делаем коллекцию по аналогии с *Student_MagicSearch*, которую я показывала в лекции. В коллекции надо:

- Подготовить данные — случайное число от 1 до 10. Создаем товар через метод *create*. Отдельно метод звать не надо, делаем через *SendRequest* в *Pre-request* скриптах поиска
- Проверить, что поиск *search* вернет:
 - а) правильное количество товара;
 - б) правильные данные внутри
- Удалить тестовые данные — чистим за собой!