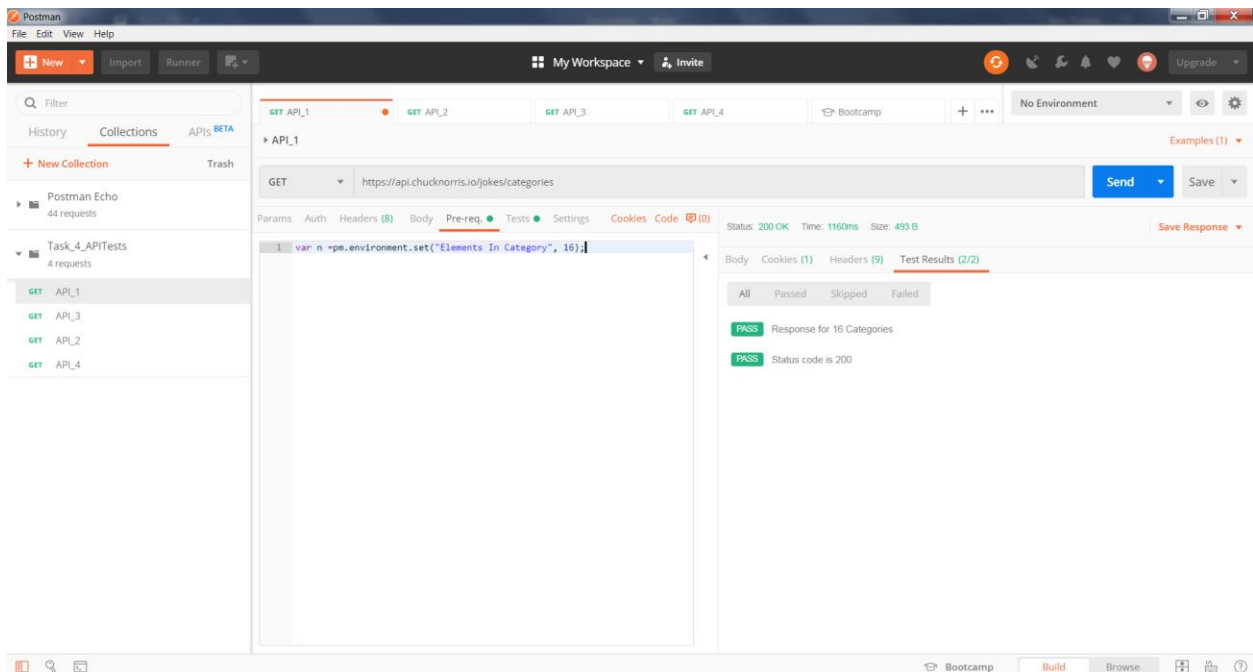
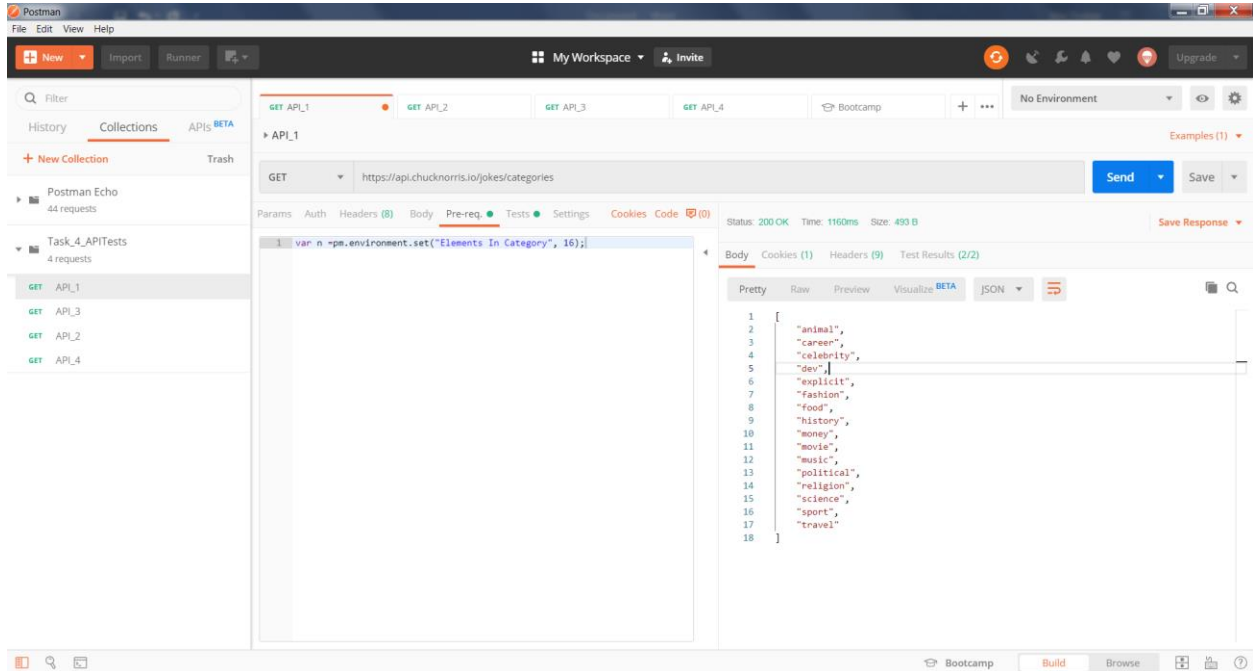
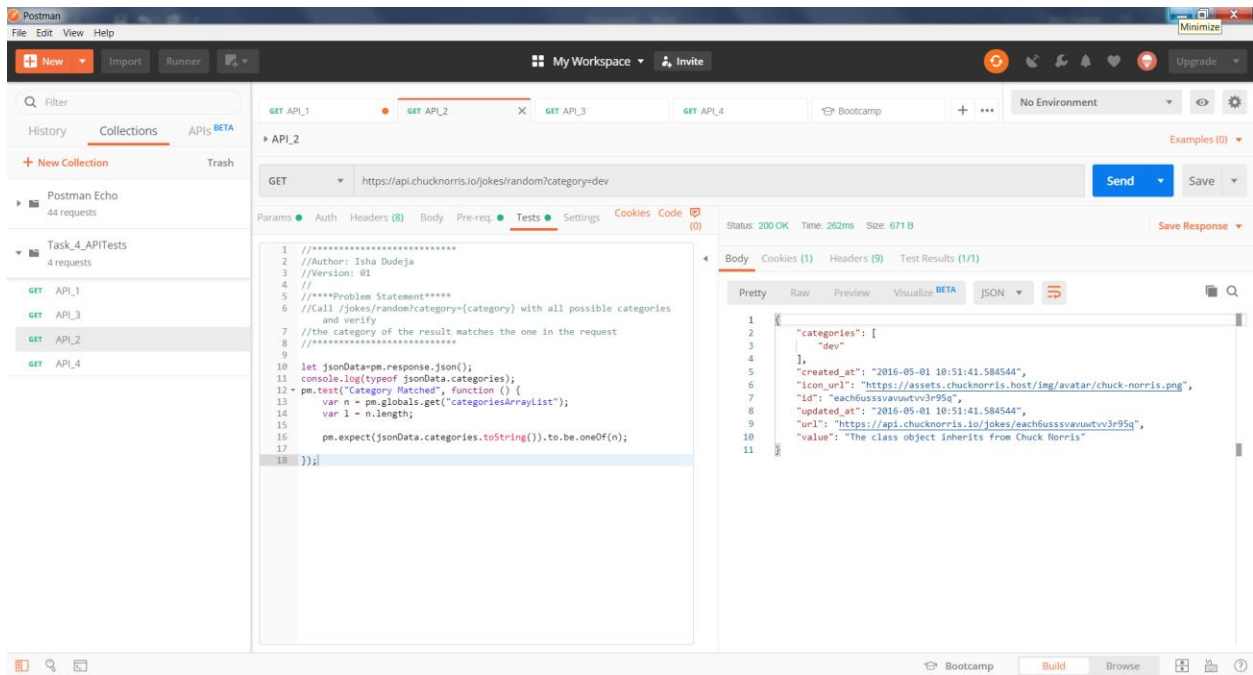
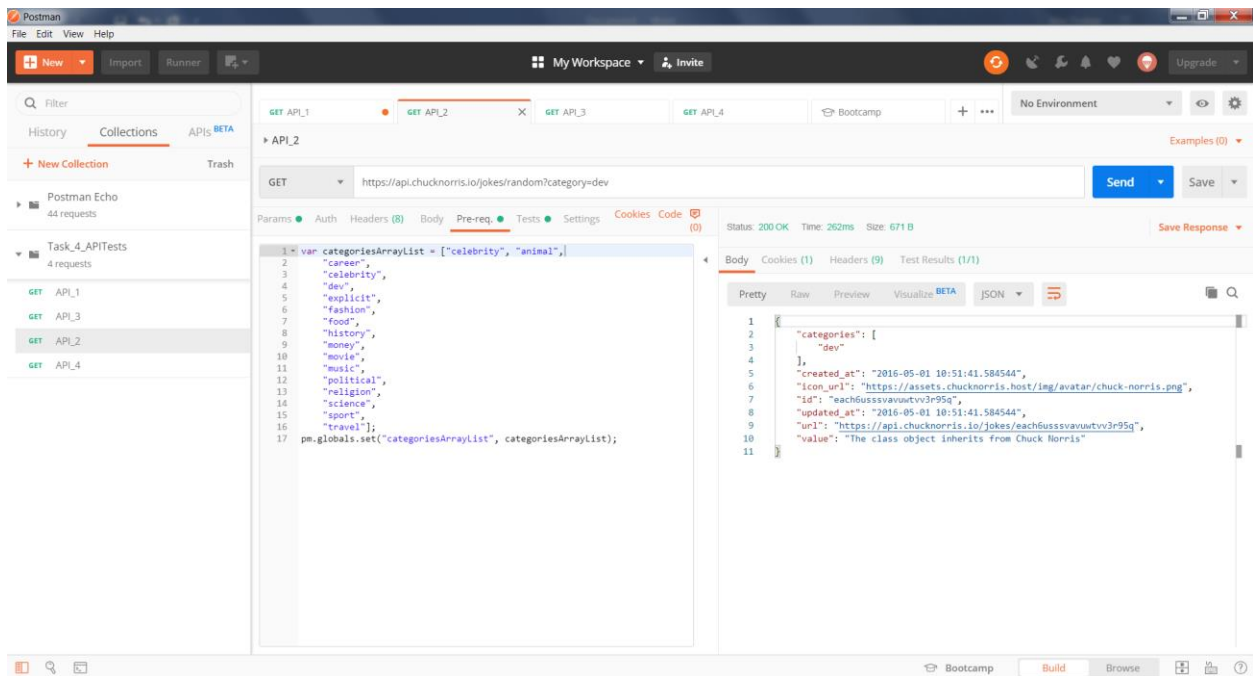


TASK-4 Testing API endpoints (tests/api)

URL: <https://api.chucknorris.io/>





Postman interface showing a REST client request for GET API_2. The URL is `https://api.chucknorris.io/jokes/random?category=dev`. The request is a GET method. The response status is 200 OK, Time: 262ms, Size: 671 B. The response body contains a JSON object with a single joke. The test results show a single test passing: "Category Matched".

```
1 //*****
2 //Author: Isha Dudeja
3 //Version: 01
4 //
5 //****Problem Statement****
6 //Call /jokes/random?category={category} with all possible categories
7 //the category of the result matches the one in the request
8 //*****
9
10 let jsonData=pm.response.json();
11 console.log(typeof jsonData.categories);
12 pm.test("Category Matched", function () {
13   var n = pm.globals.get("categoriesArrayList");
14   var l = n.length;
15   pm.expect(jsonData.categories.toString()).to.be.oneOf(n);
16 }
17 );
```

Test Results (1/1)

All	Passed	Skipped	Failed
1	1	0	0

PASS Category Matched

Postman interface showing a REST client request for GET API_3. The URL is `https://api.chucknorris.io/jokes/1BYqNs0MSzmt09vZikisA`. The request is a GET method. The response status is 200 OK, Time: 251ms, Size: 678 B. The response body contains a JSON object with a single joke. The test results show three tests passing: "Validating Entire Song Name", "Validating Last Modified Date greater than January 2016", and "Validating Valid Image".

```
1 pm.environment.set("value", "Chuck Norris once listened to an entire song....on mute");
```

Test Results (3/3)

All	Passed	Skipped	Failed
3	3	0	0

PASS Validating Entire Song Name
PASS Validating Last Modified Date greater than January 2016
PASS Validating Valid Image

Postman interface showing a REST client request for GET API_3. The URL is `https://api.chucknorris.io/jokes/1BYqNs0MSzmt9vZikisA`. The request is a GET method. The response status is 200 OK, Time: 251ms, Size: 678 B. The response body contains a JSON object with a joke. The tests section shows three tests: "Validating Entire Song Name", "Validating Last Modified Date greater than January 2016", and "Validating Valid Image". All tests passed.

```
1 //
2 //*****Problem Statement*****
3 //Call /jokes/1BYqNs0MSzmt9vZikisA, verify the value of this joke contains "entire song" and
4 //that the last modification was later
5 //than January 2016.
6 //Please also ensure, that the provided icon url is a valid image url
7 //*****
8
9
10
11 let jsonData = pm.response.json();
12 console.log(jsonData);
13
14 pm.test("Validating Entire Song Name", function () {
15
16     pm.expect(jsonData.value.toString()).to.eql(pm.environment.get("value"));
17
18 });
19
20 pm.test("Validating Last Modified Date greater than January 2016 ", function () {
21
22
23     var d = new Date(jsonData.updated_at);
24     var receivedDate = d.getTime();
25     var referenceDate = new Date('2016-01-01').getTime();
26     pm.expect(receivedDate).to.above(referenceDate);
27
28 });
29
30 pm.test("Validating Valid Image", function () {
31
32     pm.expect(jsonData.icon_url).to.eql('https://assets.chucknorris.host/img/avatar/chuck-norris.png');
33
34 });
35
36
```

Postman interface showing a REST client request for GET API_4. The URL is `https://api.chucknorris.io/jokes/1BYqNs0MSzmt9vZikisAxxxxxx`. The request is a GET method. The response status is 404 Not Found, Time: 282ms, Size: 536 B. The response body contains a JSON object with an error message. The tests section shows one test: "Validating Status code is 204", which failed.

```
1 //*****
2 //Author: Isha Dudeja
3 //Version: 01
4 //
5 //*****Problem Statement*****
6 //Call /jokes/1BYqNs0MSzmt9vZikisAxxxxxx, calling this endpoint should
7 //NOT return a joke but a 404 status code (it will fail but that is what we want)
8
9
10
11 pm.test("Validating Status code is 204", function () {
12
13     pm.response.to.have.status(204);
14
15 });
16
```

```
1 {
2   "timestamp": "2019-09-16T21:17:22.149Z",
3   "status": 404,
4   "error": "Not Found",
5   "message": "Joke with id
6     \"1BYqNs0MSzmt9vZikisAxxxxxx\" not found.",
7   "path": "/jokes/1BYqNs0MSzmt9vZikisAxxxxxx"
8 }
```

