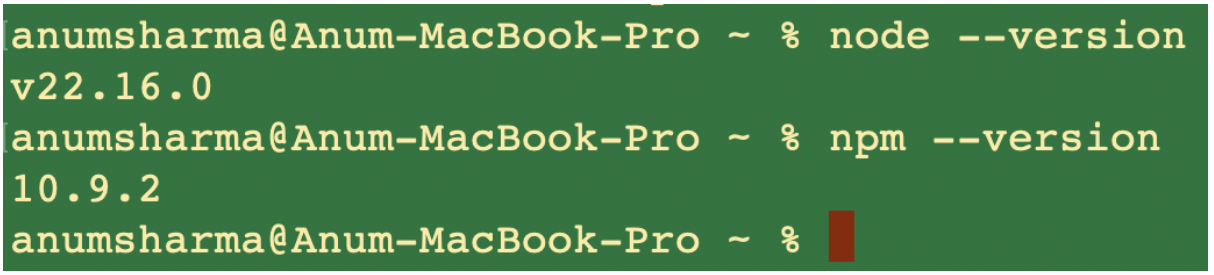
Postman Mini Project

Getting started with postman:

Installations:

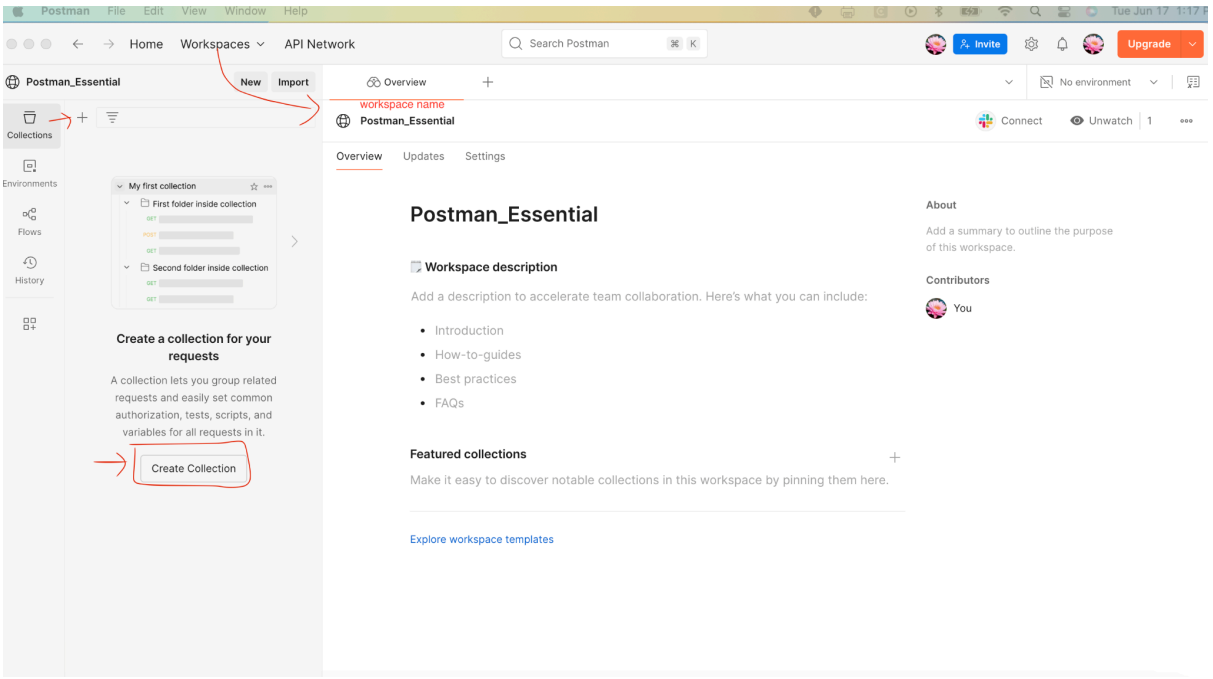
1. First download Postman : https://postman.com/downloads
2. Check to see if Node is installed using terminal : node - -version (Node.js and npm: https://nodejs.org/)
3. If not, then install the node.js
4. Once you install node.js , npm is automatically install
5. Go to terminal and type npm –version to check the version
6. Next, install Newman using terminal: npm install -g newman



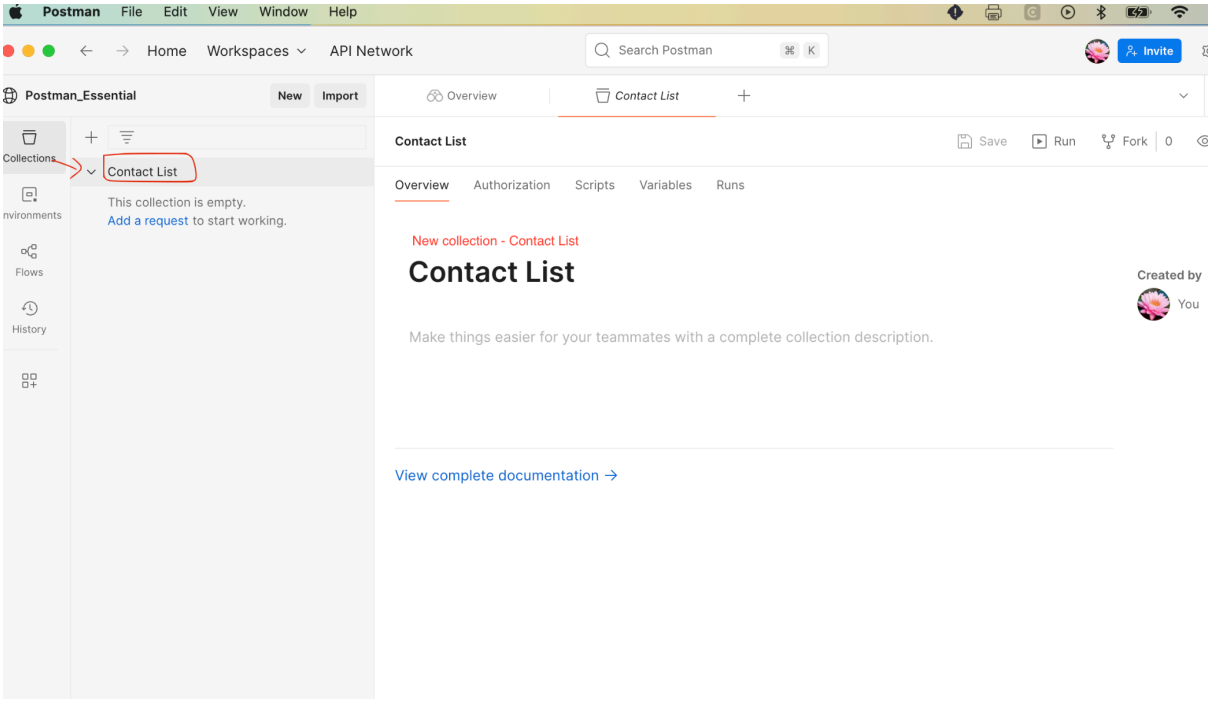
Postman Terms:

1. Collections: all the requests and assertions for an API; the top level folder
2. Folder: a group of requests and assertions that reside in the collection

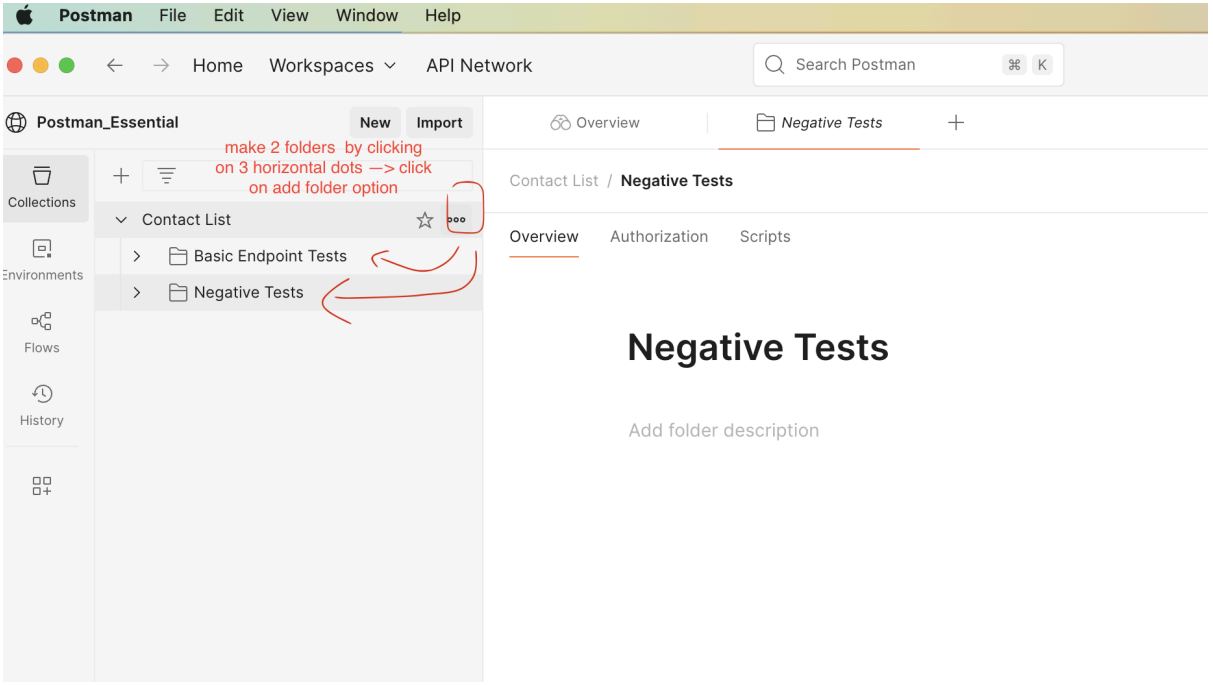
Open Postman
Create a new workspace - Postman Essential



Create a new collection - Contact List



Under New Collection - add 2 new folders :
- Basic endpoint tests
- Negative tests



How to Create Postman Requests and Assertions?

We will use API called “Contact List API”

We have created Contact List API

URL: https://thinking-tester-contact-list.herokuapp.com/

It allows users to add, edit and delete contacts like they would in an address book

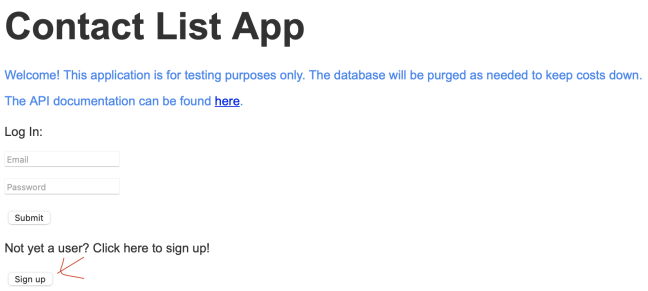
This API also has a user Interface, or UI to go with it.

Let do that → follow below steps :

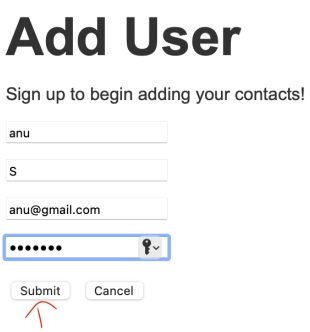
Navigate to https://thinking-tester-contact-list.herokuapp.com/

Create an account which we will use for API call

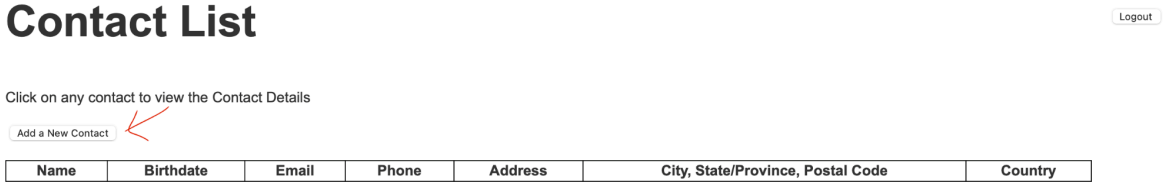
Click on sign up button



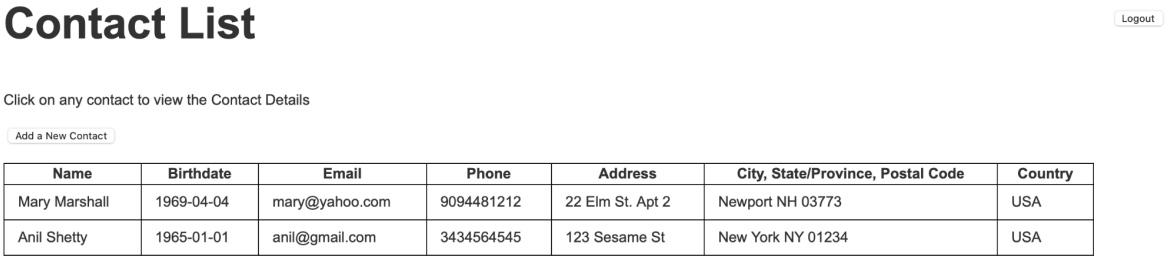
Add first name , last name , email, password and submit button
NOTE - note down the email and password . this will use in POST method



User land on contact list page



Let add new contact in order to see in our GET request

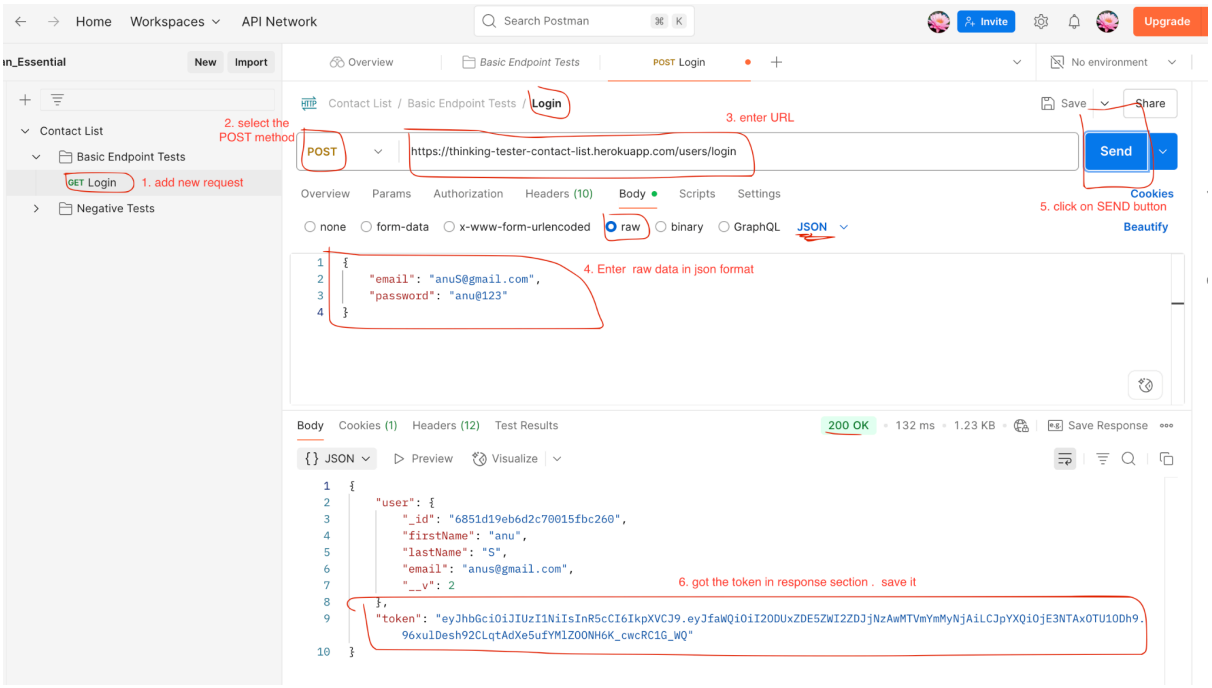


How JSON web tokens (JWTs)work?
The most common way to secure API's
Generated through a POST request
When a valid username and password are sent in the request, a JWT is generated
The JWT can then be used in all requests to the API

So let's generate JWT

Go back to POSTMAN

1. Under Contact List collection → Basic Endpoints Tests folder → add a new request
2. Give it name - Login
3. Select Method- POST
4. Enter the URL - https://thinking-tester-contact-list.herokuapp.com/users/login
5. Under body section → select Raw radio button + JSON format and paste the below JSON :
   {
   "Email" : "enter your email address you entered while signup",
   "Password": "enter the password"
   }
6. Click on "SEND"button
7. User can see response with 200 OK in response section
8. Save the token value
9. Save the login request by clicking on the save button on the top right side.



Requests to the Contacts List API

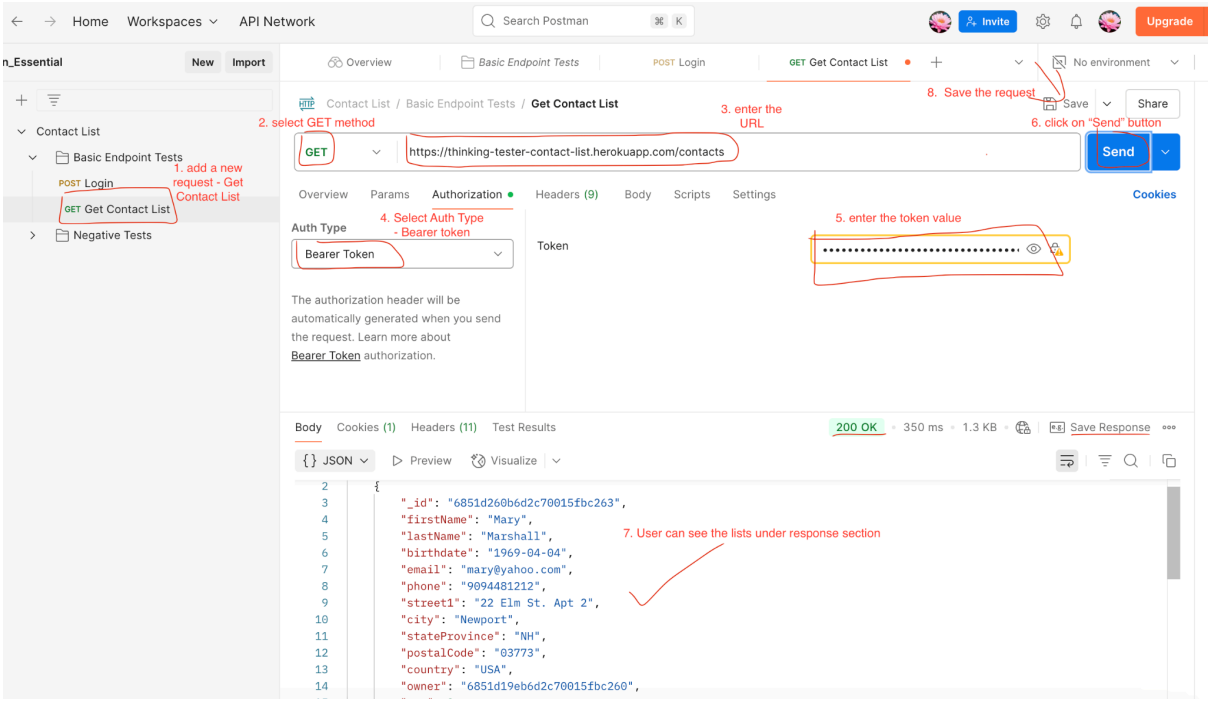
1. GET request : returns all contacts
2. GET/{contactID} request: returns one specific contact
3. POST request - adds a contact
4. PUT/{contactId} request - updates a specific contact
5. DELETE/{contactId} request - deletes a specific contact

Postman Documentation: https://documenter.getpostman.com/view/4012288/2s8YRiKDbu (helps in getting URL link)

—----------Back to Postman—----------

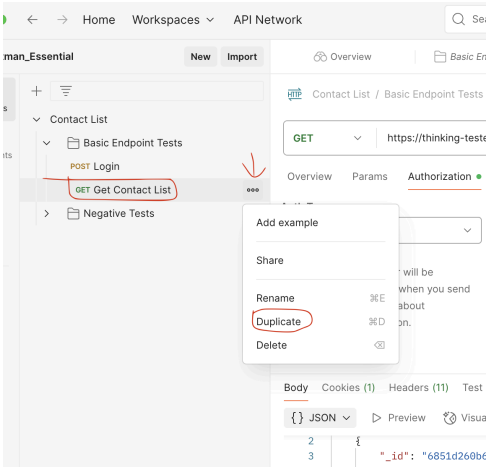
GET method - get all contacts using postman :

1. Create a new request - Get Contact List
2. Select the method - GET
3. Enter the URL-
4. Go to Authorization tab
5. Select the Auth Type - Bearer token
6. Enter the token value you save in above steps
7. Click on “send” button
8. User will see lists in response section
9. Save the request by clicking on Save button on the top right

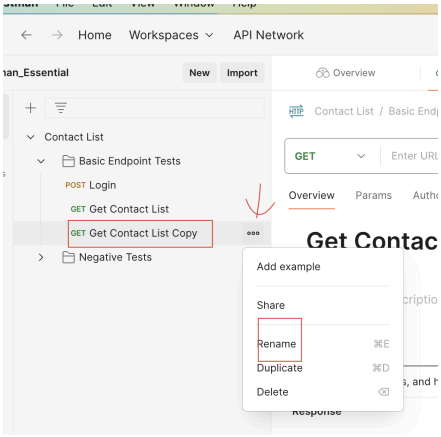


GET method - get specific contact from list using postman :

1. Create a new request in easy way
2. Hover over Get Contact List request
3. Click on 3 dots menu
4. Select the duplicate options



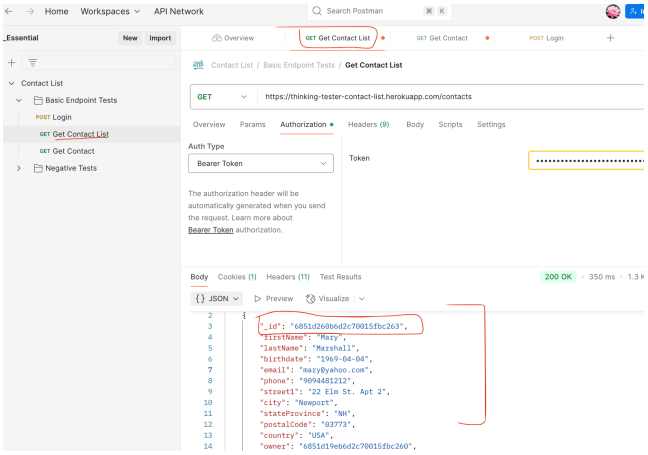
5. New request is added
6. Rename the request by clicking on 3 dots menu and select rename option



7. Make a request a - Get Contacts
8. Select the method - GET
9. Enter the URL - https://thinking-tester-contact-list.herokuapp.com/contacts/6851d260b6d2c70015fbc263

Where 6851d260b6d2c70015fbc263 is contact ID

10. Get the contacts ID value from Get Contacts List response



11.Go to Authorization tab

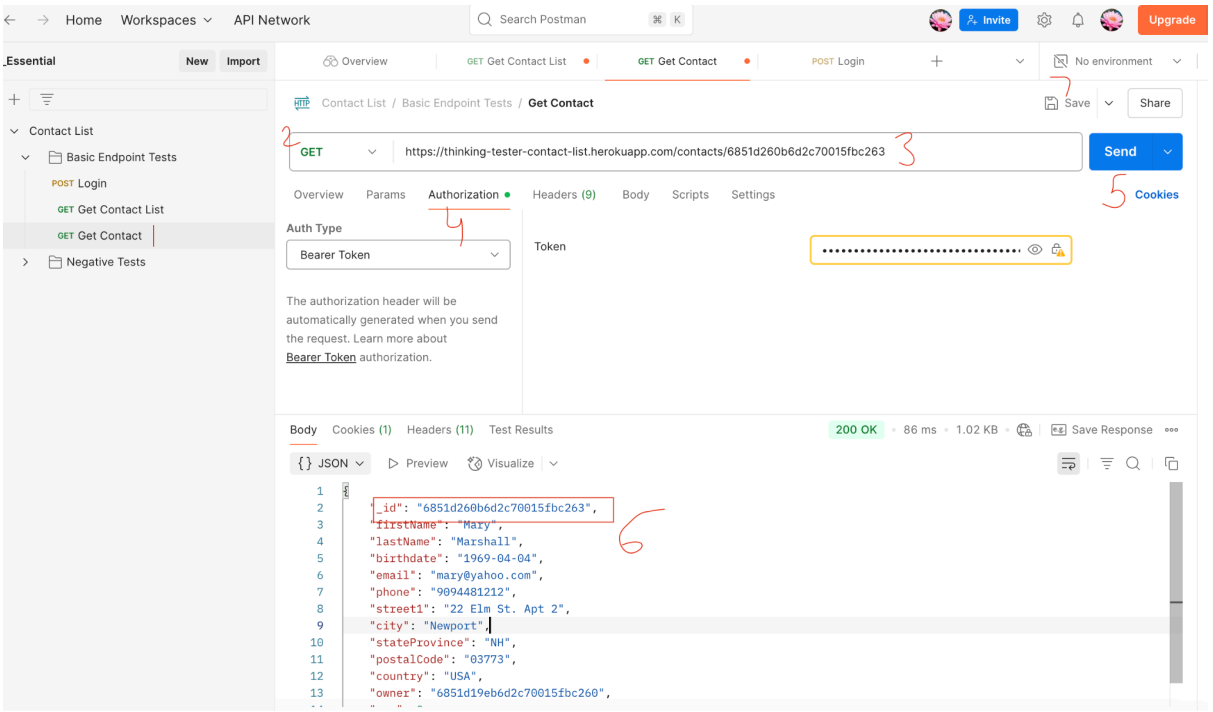
12 Select the Auth Type - Bearer token

13. Enter the token value you save in above steps

14. Click on “send” button

15. User will see lists in response section

16. Save the request by clicking on Save button on the top right

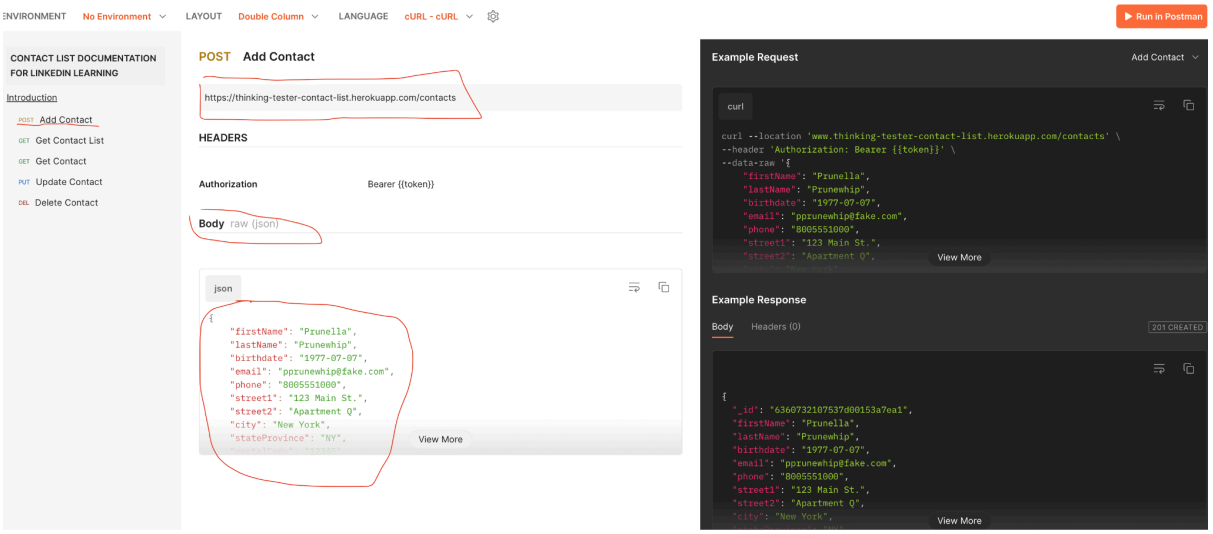


POST method - add contacts using postman :

1. Create a new request - Add Contact
2. Select the method - POST
3. Enter the URL- https://thinking-tester-contact-list.herokuapp.com/contacts and under body tab enter raw data in json format

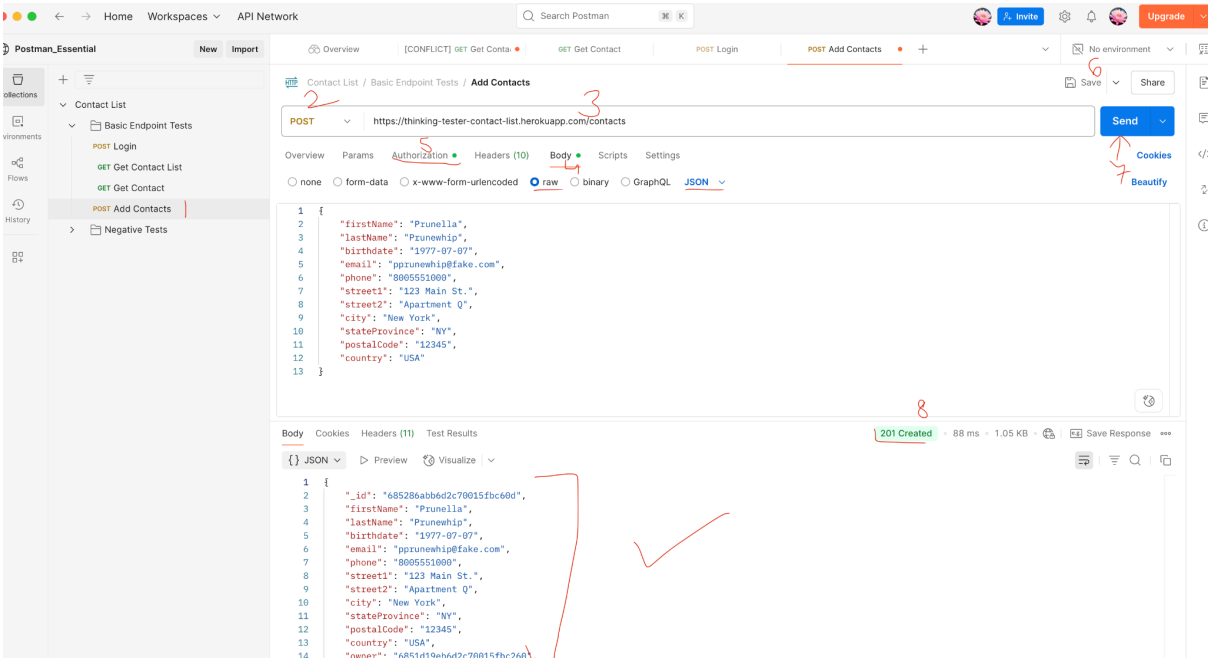
```
{
    "firstName": "Prunella",
    "lastName": "Prunewhip",
    "birthdate": "1977-07-07",
    "email": "pprunewhip@fake.com",
    "phone": "8005551000",
    "street1": "123 Main St.",
    "street2": "Apartment Q",
    "city": "New York",
    "stateProvince": "NY",
    "postalCode": "12345",
    "country": "USA"
}
```

NOTE: User can get the URL from postman documentation https://documenter.getpostman.com/view/4012288/2s8YRiKDbu#3cf734e9-f970-4431-b7d2-2cd04a3614be



4. Go to Authorization tab
5. Select the Auth Type - Bearer token
6. Enter the token value you save in above steps
7. Save the request by clicking on Save button on the top right
8. Click on “send” button
9. User will see lists in response section

.



POST method - add contacts using postman :

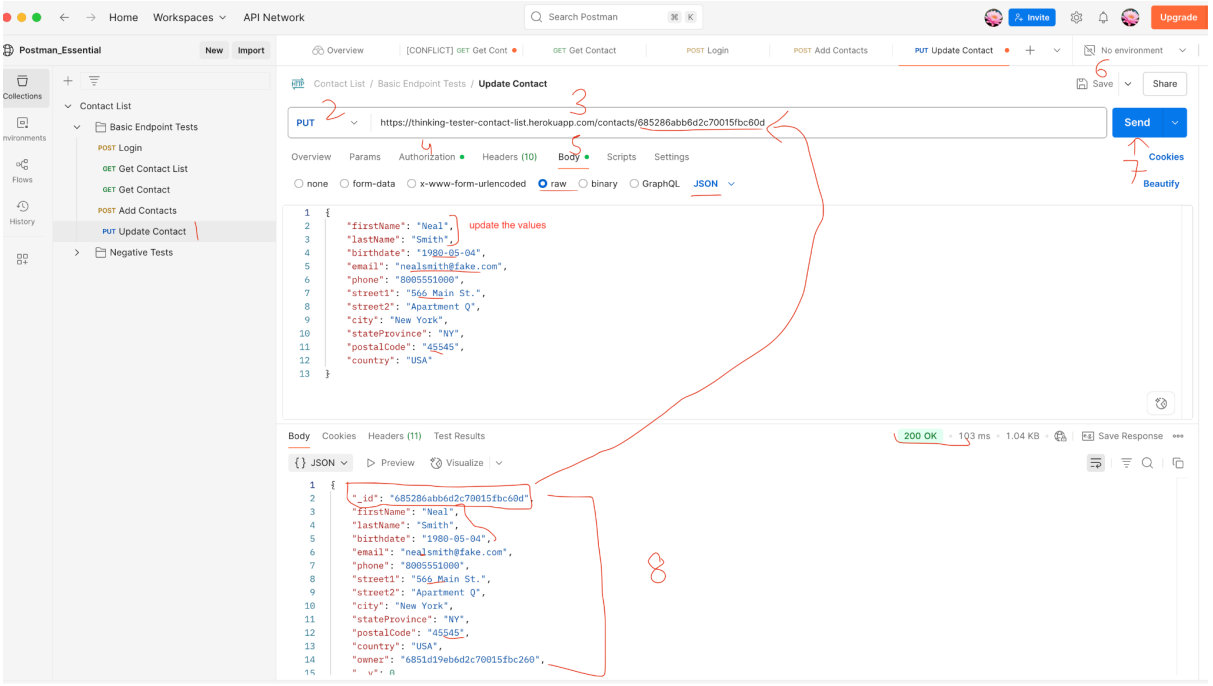
1. Create a new request - Update Contact
2. Select the method - PUT
3. Enter the URL- https://thinking-tester-contact-list.herokuapp.com/contacts/685286abb6d2c70015fbc60d and under body tab update raw data(first name, last name, email, address , postal code) in json format

```
{
    "firstName": "Neal",
    "lastName": "Smith",
    "birthdate": "1980-05-04",
    "email": "nealsmith@fake.com",
    "phone": "8005551000",
    "street1": "566 Main St.",
    "street2": "Apartment Q",
    "city": "New York",
    "stateProvince": "NY",
    "postalCode": "45545",
    "country": "USA"
}
```

NOTE: User can get the URL from postman documentation https://documenter.getpostman.com/view/4012288/2s8YRiKDbu#3cf734e9-f970-4431-b7d2-2cd04a3614be

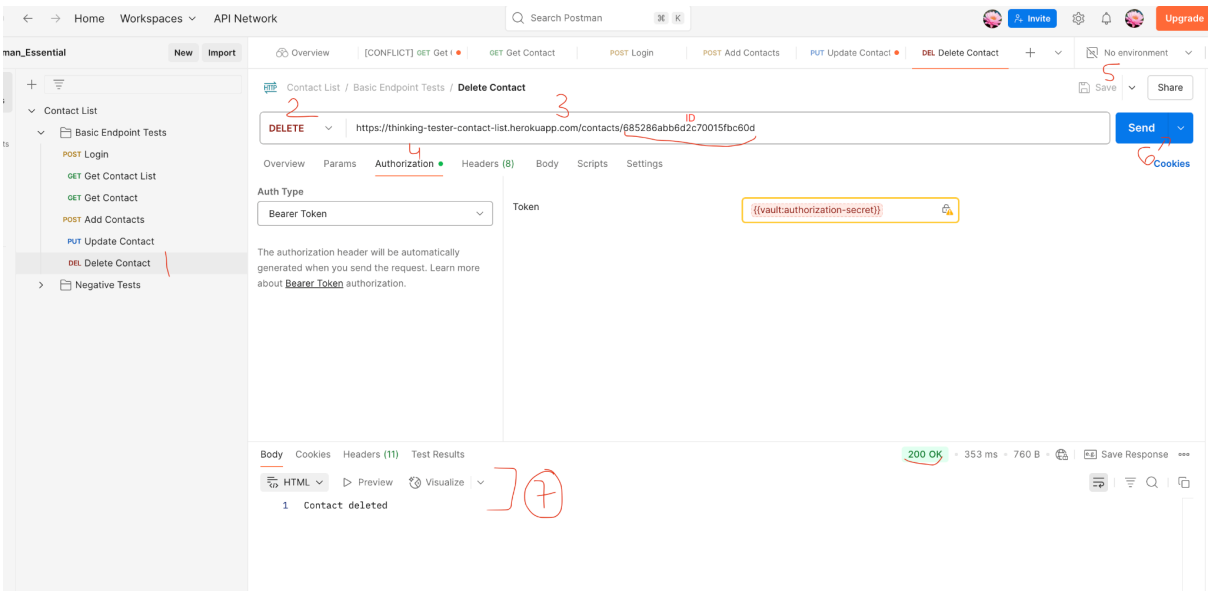
4. Go to Authorization tab
5. Select the Auth Type - Bearer token
6. Enter the token value you save in above steps

7. Save the request by clicking on Save button on the top right
8. Click on “send” button
9. User will see lists in response section



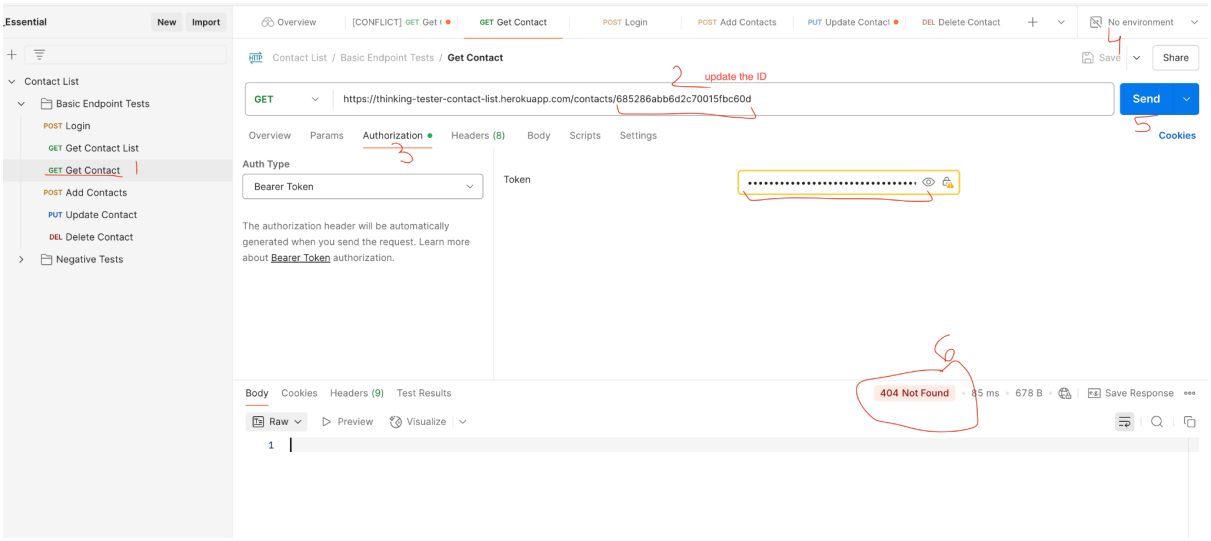
DELETE method - delete contacts using postman :

1. Create a new request - Delete Contact
2. Select the method - DELETE
3. Enter the URL-
https://thinking-tester-contact-list.herokuapp.com/contacts/685286abb6d2c70015fbc60d
Where 685286abb6d2c70015fbc60d is the ID number of contact
4. Go to Authorization tab
5. Select the Auth Type - Bearer token
6. Enter the token value you save in above steps
7. Save the request by clicking on Save button on the top right
8. Click on “send” button
9. User will see lists in response section



In order to check that contact is deleted , go to Get Contact request and update the

ID- [685286abb6d2c70015fbc60d](685286abb6d2c70015fbc60d) and when click on send button , user got the response status = 404 NOT FOUND

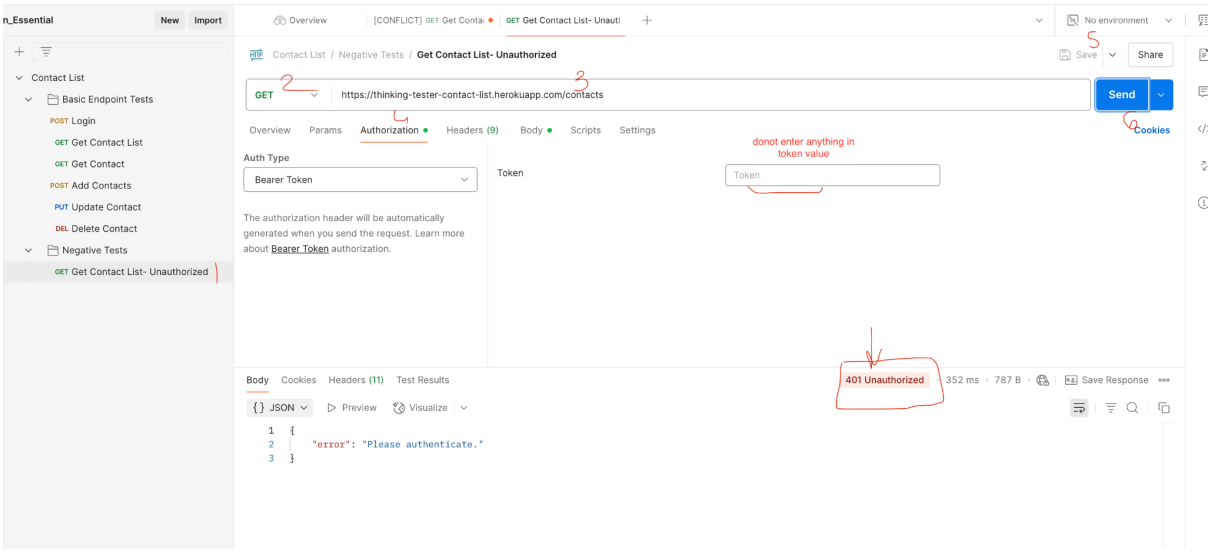


We have 6 saved requests . Each request is successful.

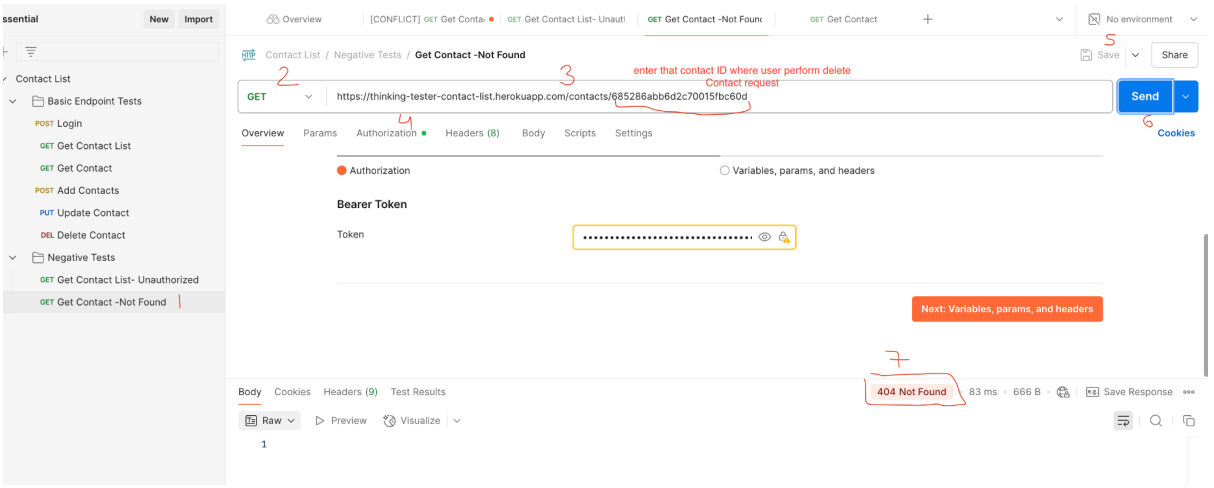
Negative Tests

Let make unsuccessful request in negative Tests folder with 5 different negative testing scenarios

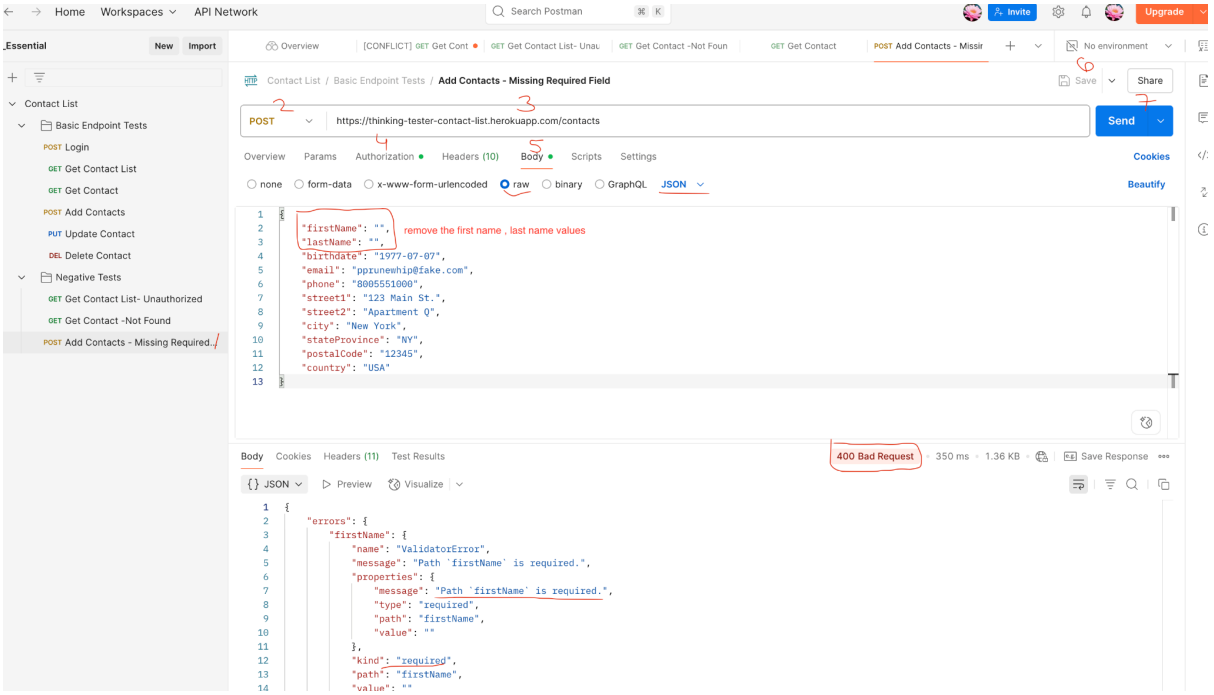
1. A request is sent with a missing authentication token



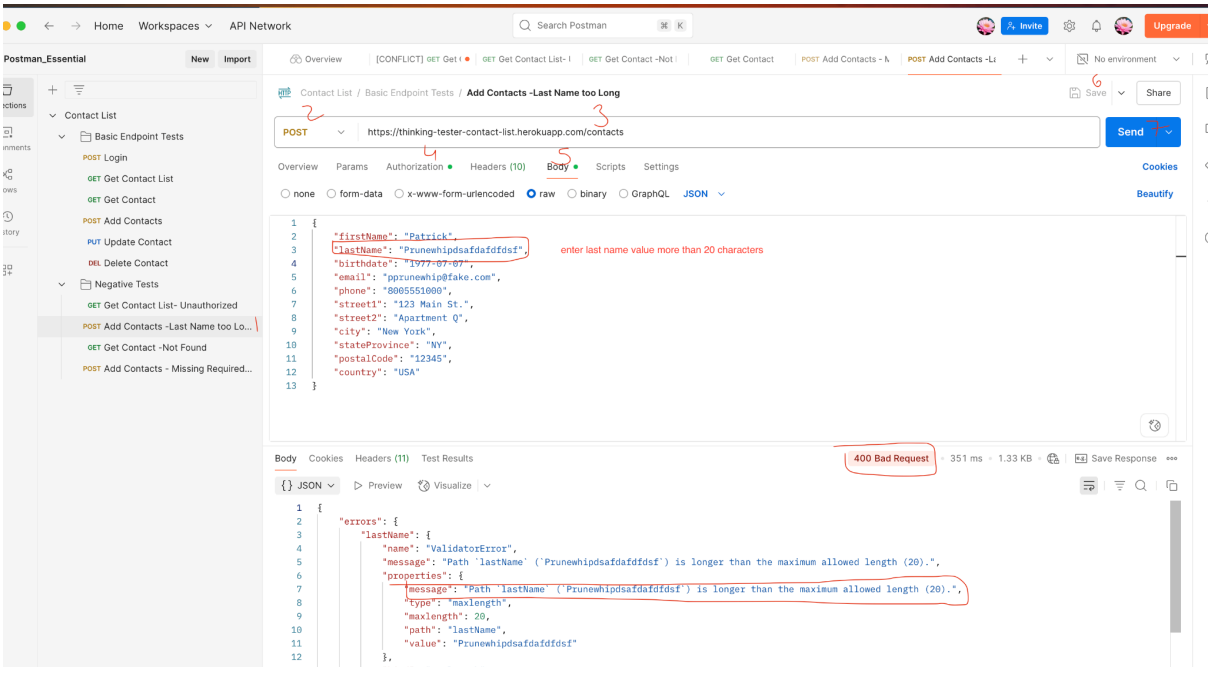
2. A record is not found - enter the ID whose record is already deleted



3. A request is sent with missing required information - first name , last name value is not entered



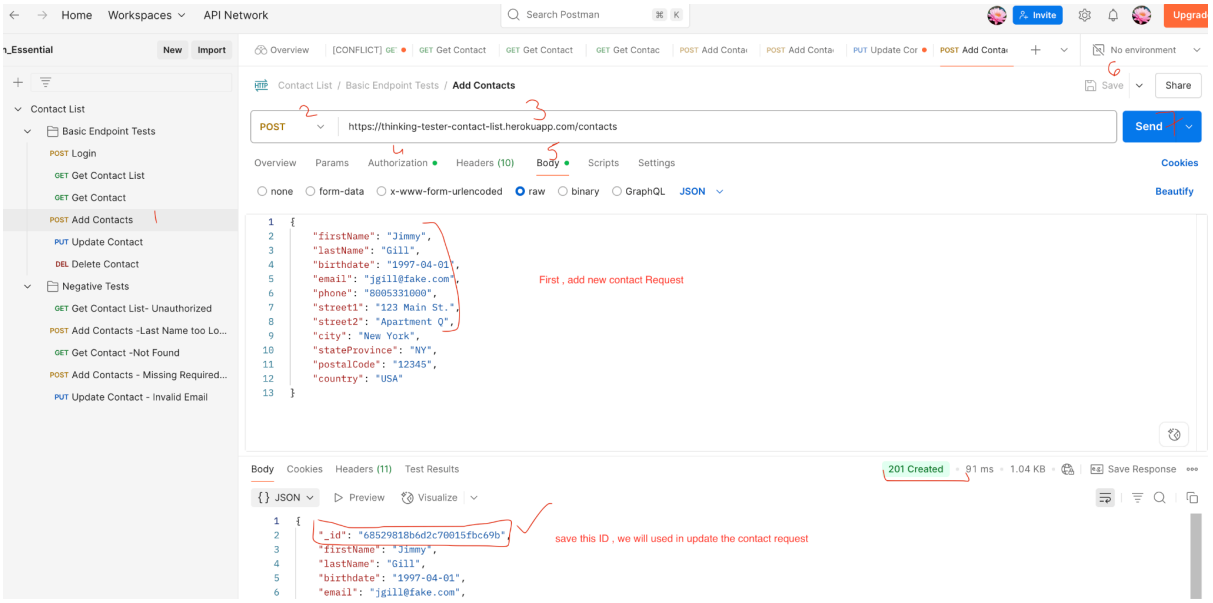
4. A value is sent with too many characters - Last name with more than 20 characters



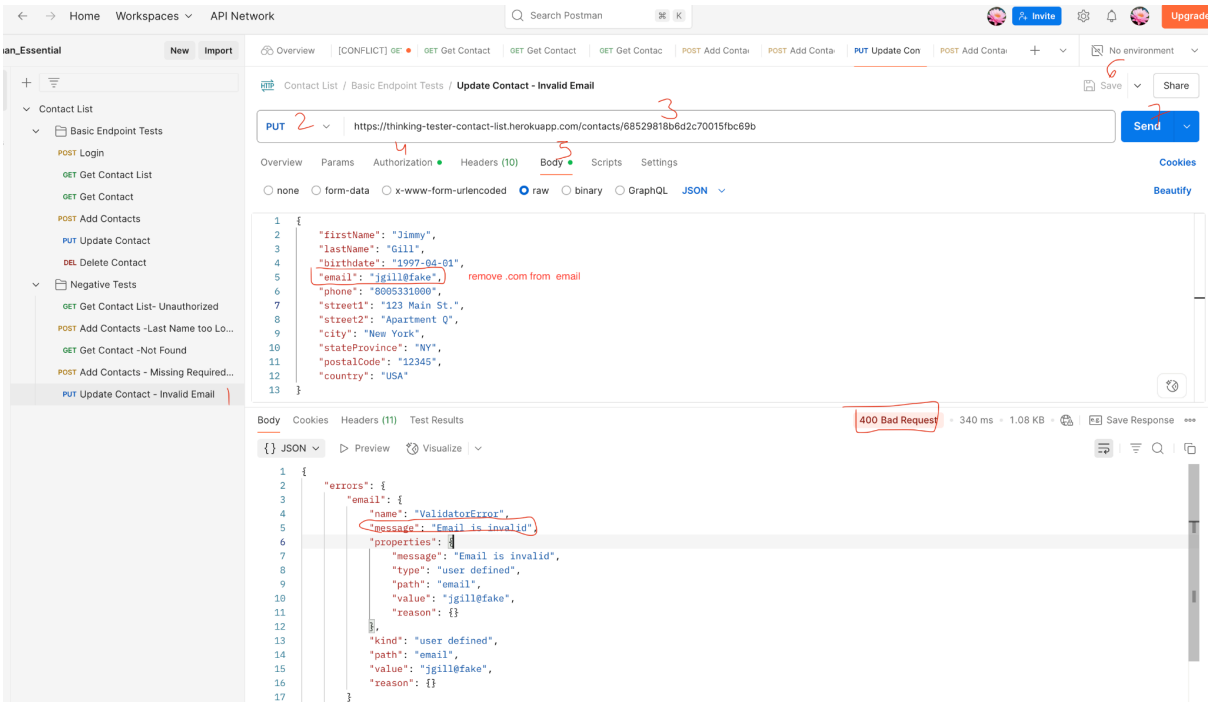
5. A value is sent that is not in correct format- invalid email format

This scenario needs 2 steps :

First - add a new contact and then save the ID



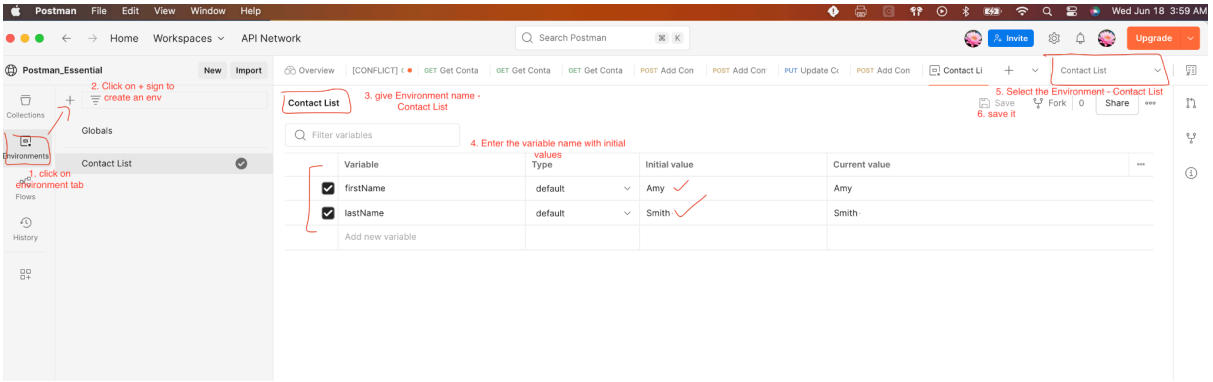
Second - update the contact with invalid email format



Environment Variables

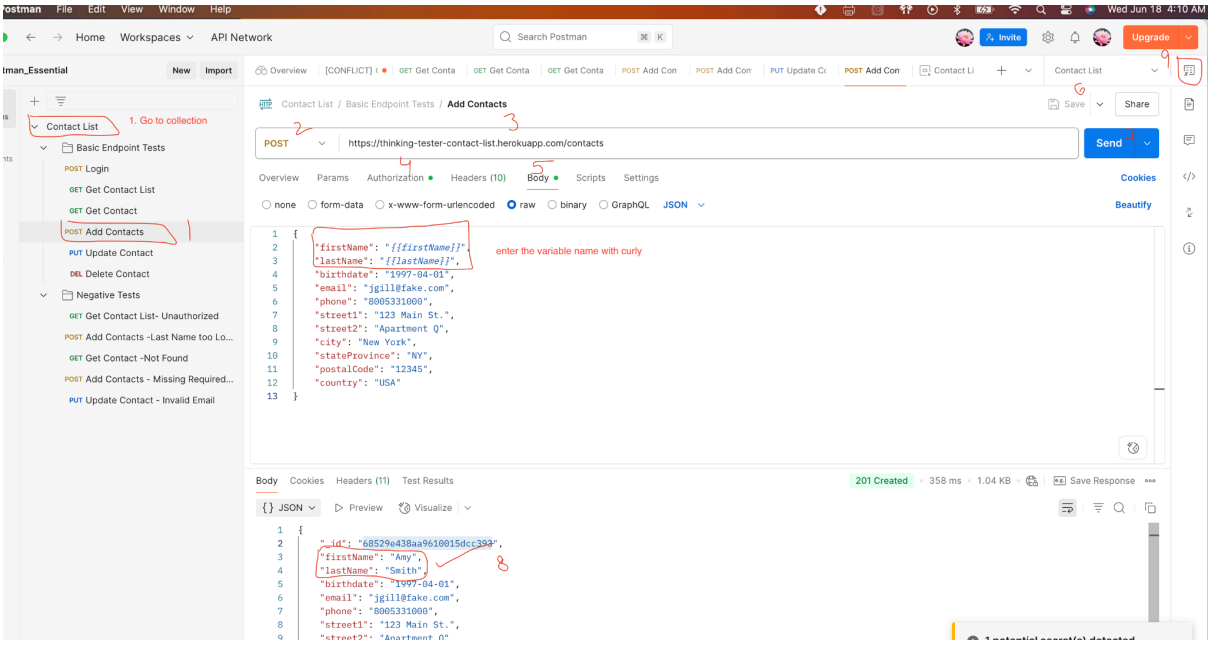
Environment - a group of variables in postman
Initial value - the variable's value that is saved in env file

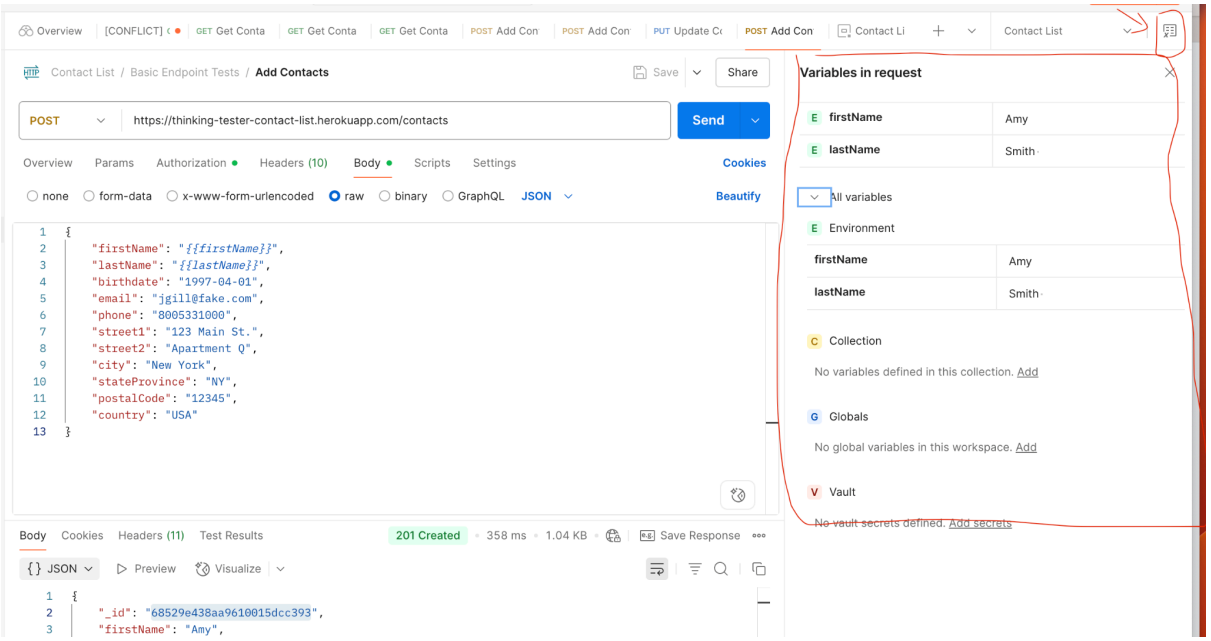
Current value - a temporary variable value



Set the Env variables - go to Env tab

Set the env variable for collection in POST request

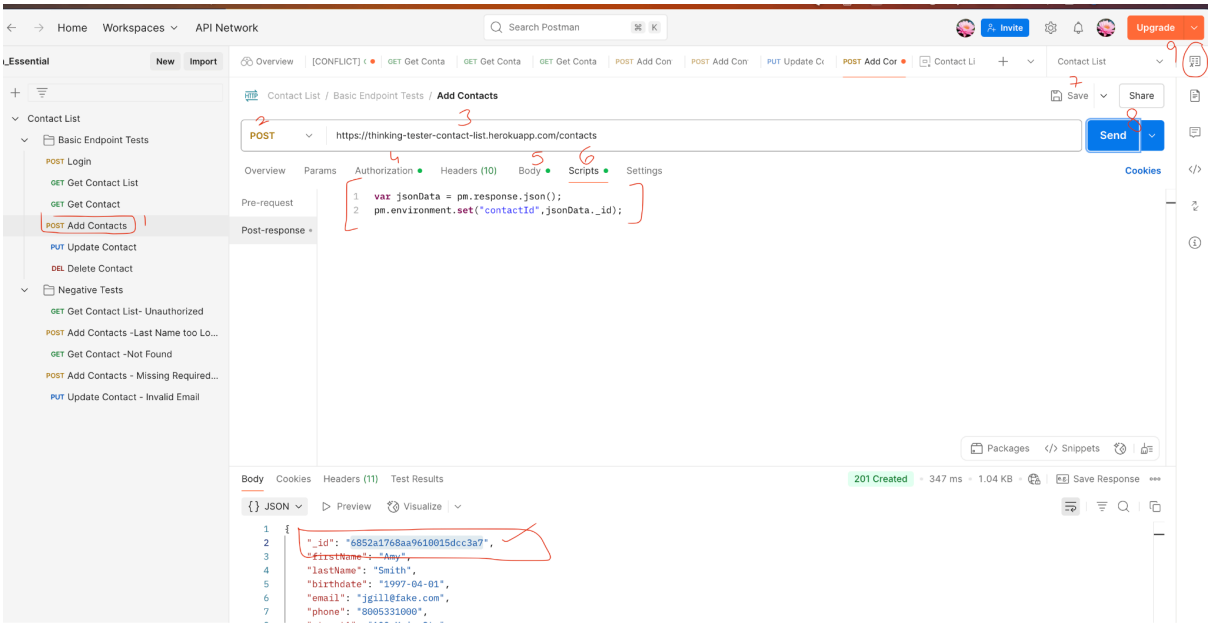


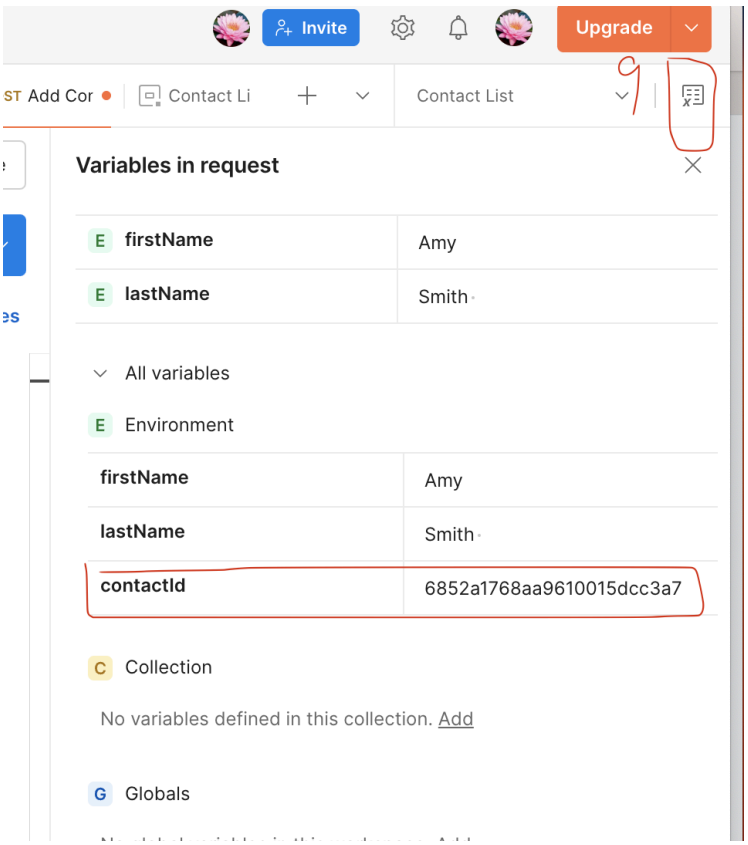


Saving response data as a variable

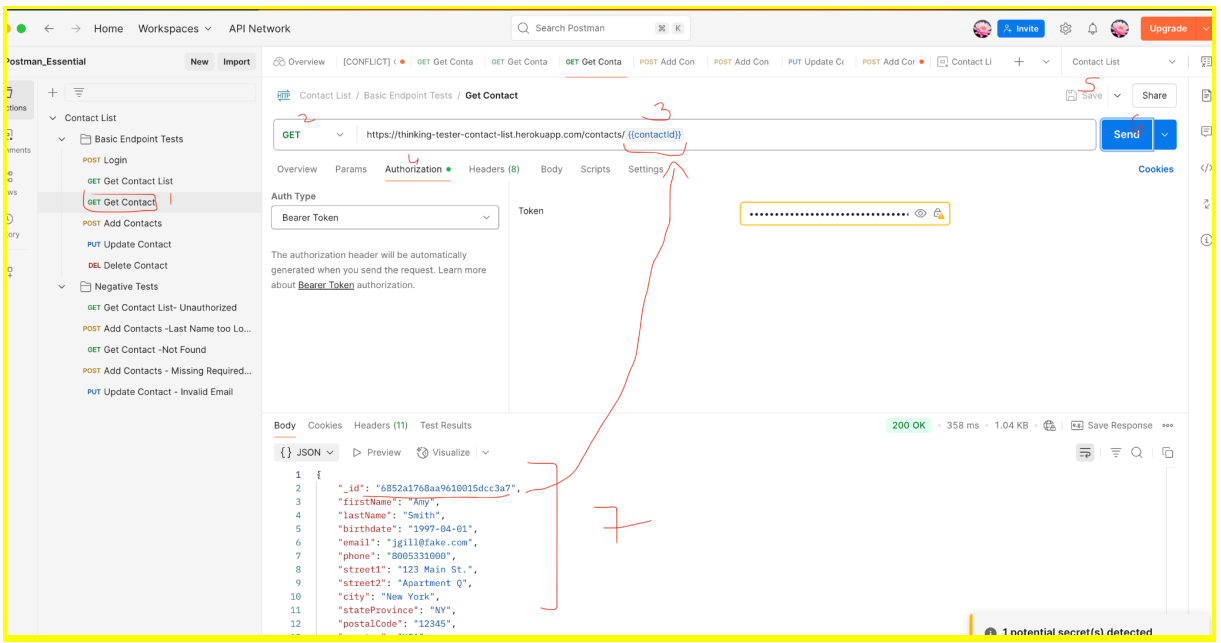
Variables can be programmatically saved using the Test tab of the postman request once the variable has been saved, it can be used in all other requests

Save Contact ID for reuse
Save token for reuse





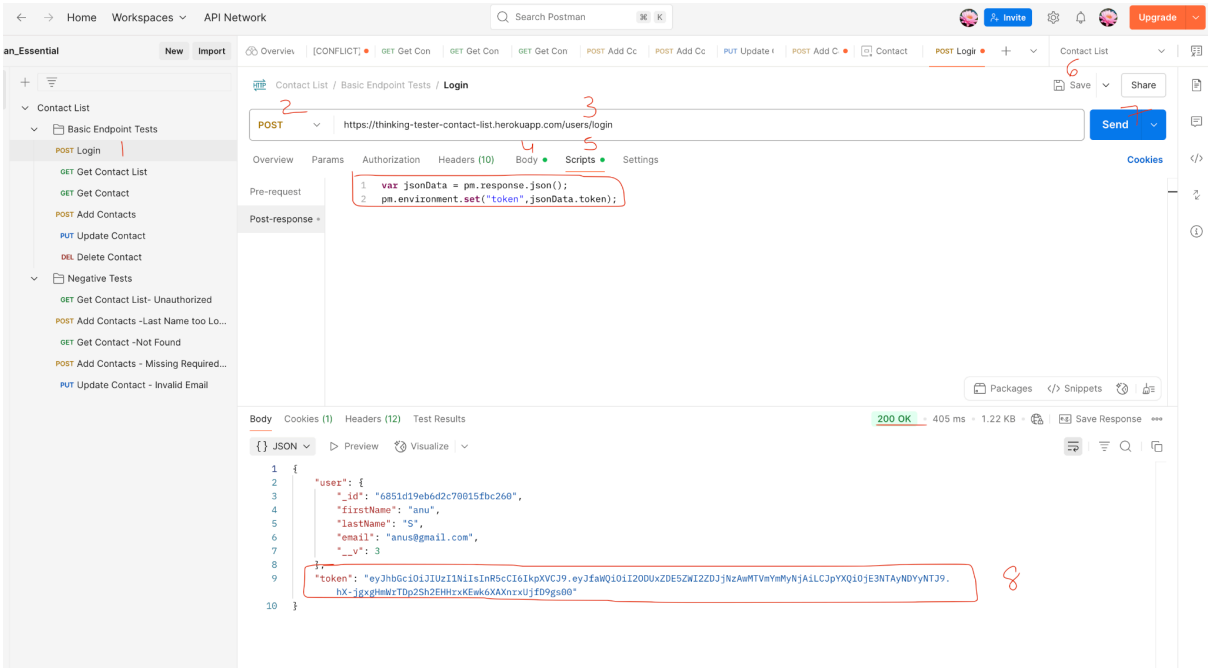
Now go to GET Contacts request and change the URL



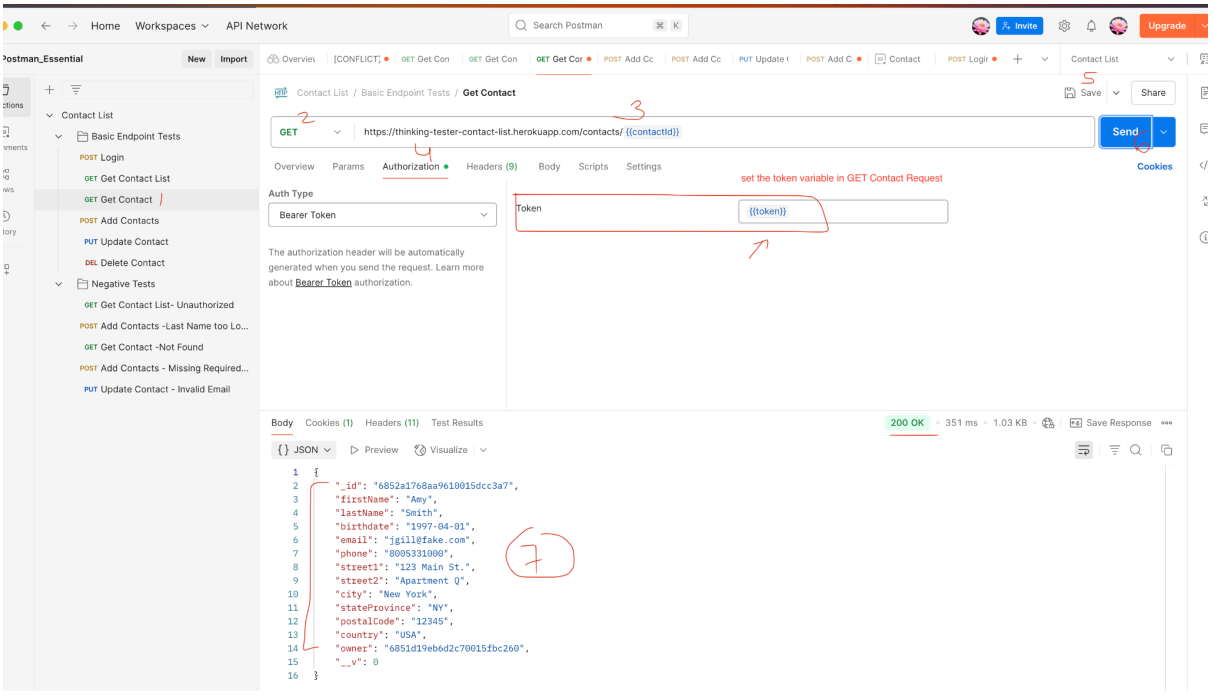
Let do it for token value too

Step 1

Write the script for setting the token value



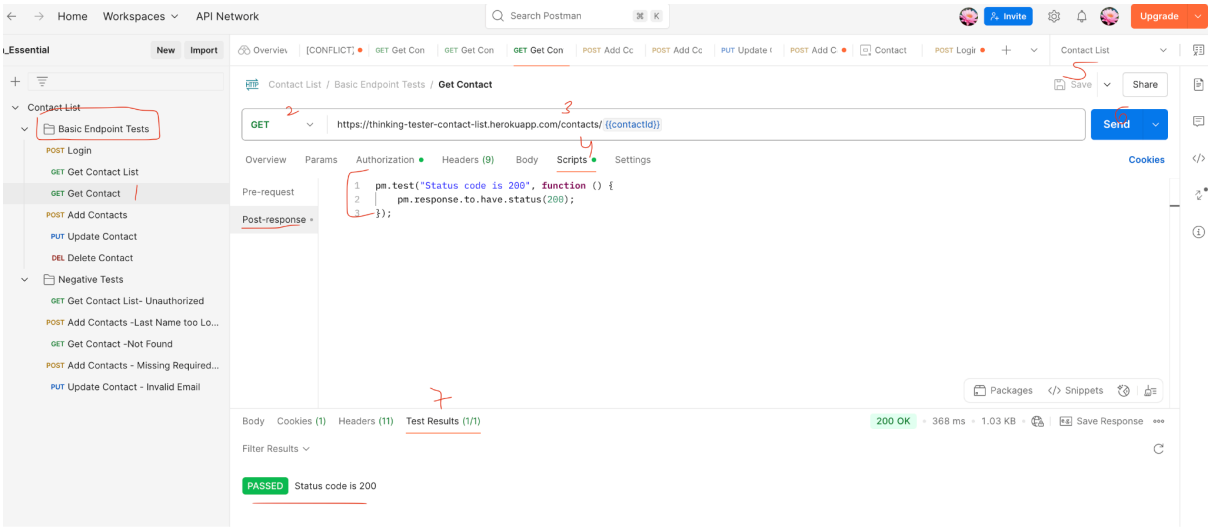
Step 2 - set token value as variable value

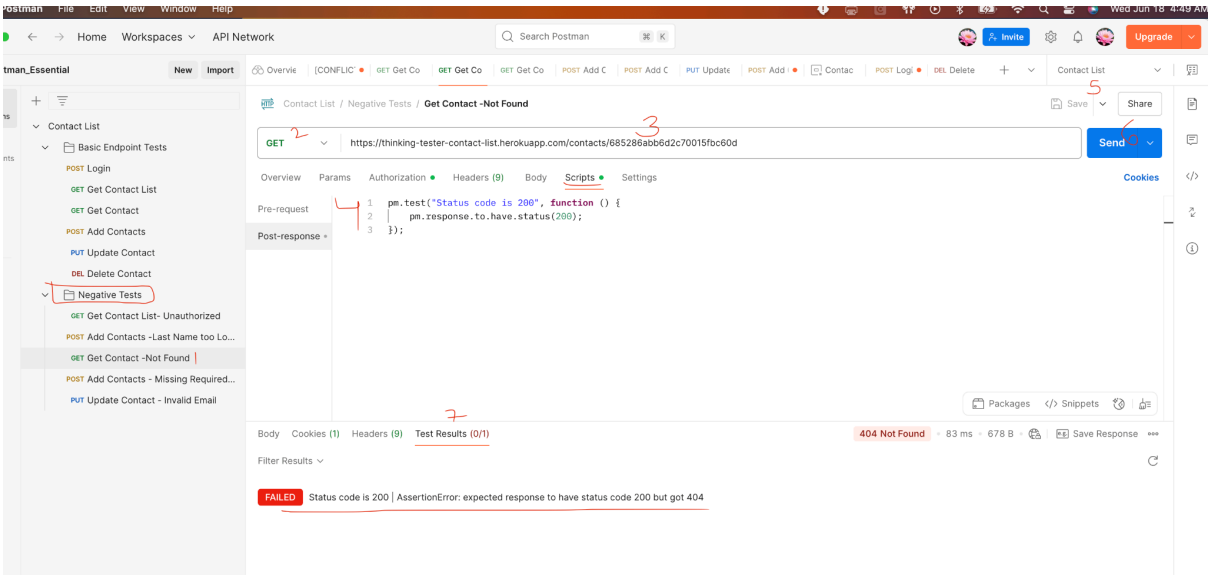


Assertions :

Status type assertions

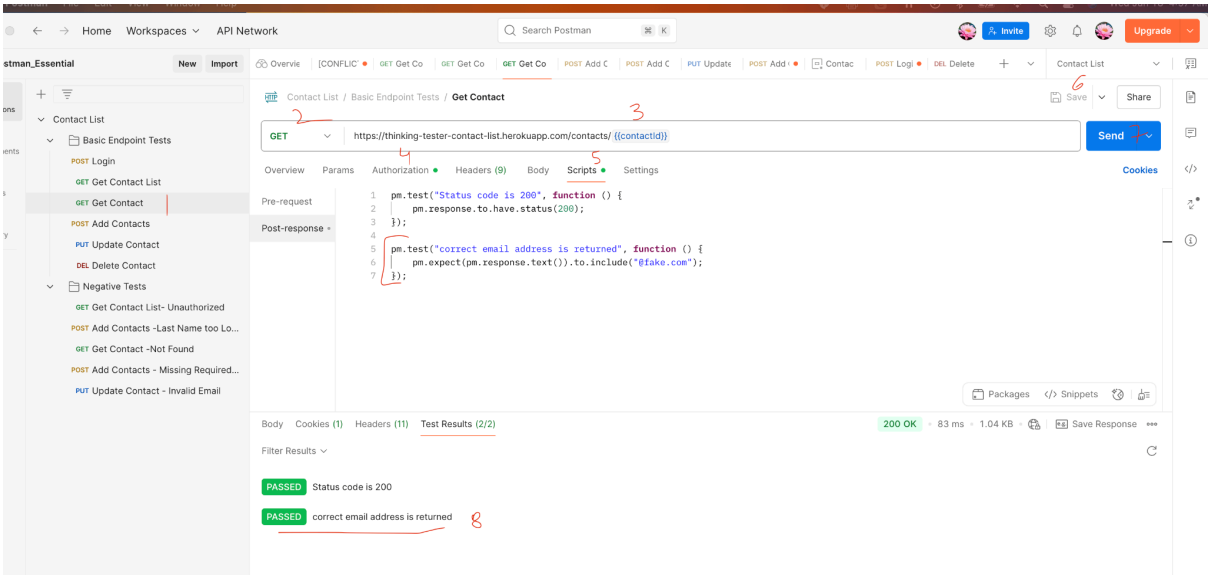
200- the request was completed as expected
201 - a new response was created
401 - the user is not authenticated
404- the resource was not found



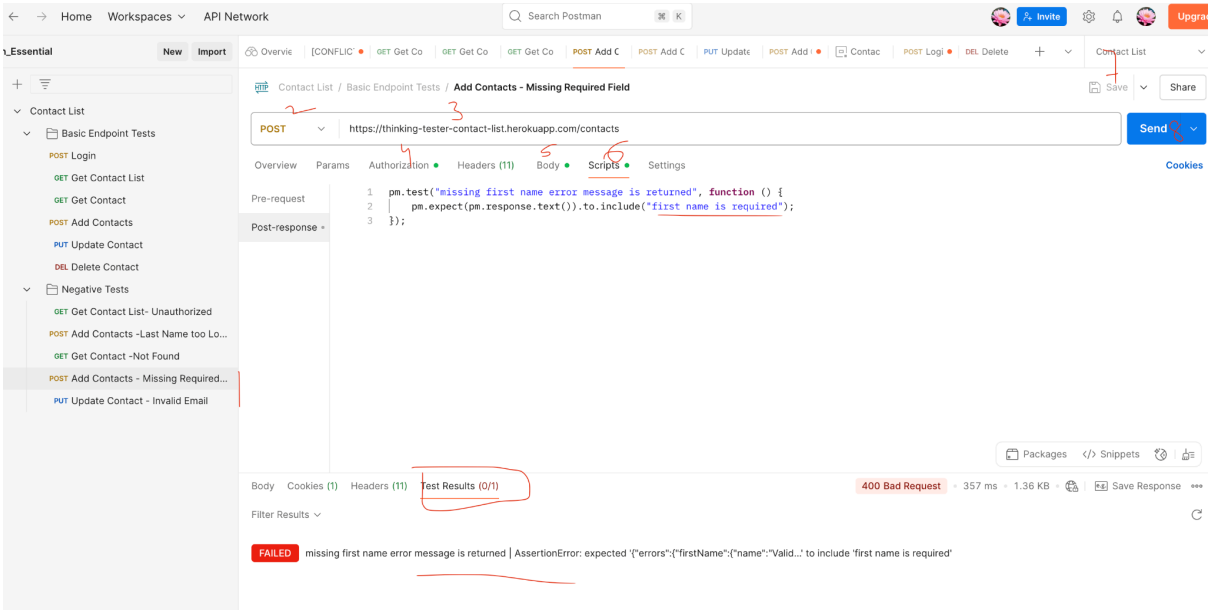


BODY Assertions:
A body assertion verifies that body of response contains the text that we were expecting.
Response body: **contains string:** contains string: the text is in the body of response, but may not be the entire response
Response body: **is equal to string:** contains string: the text matches with entire body of the response

Positive

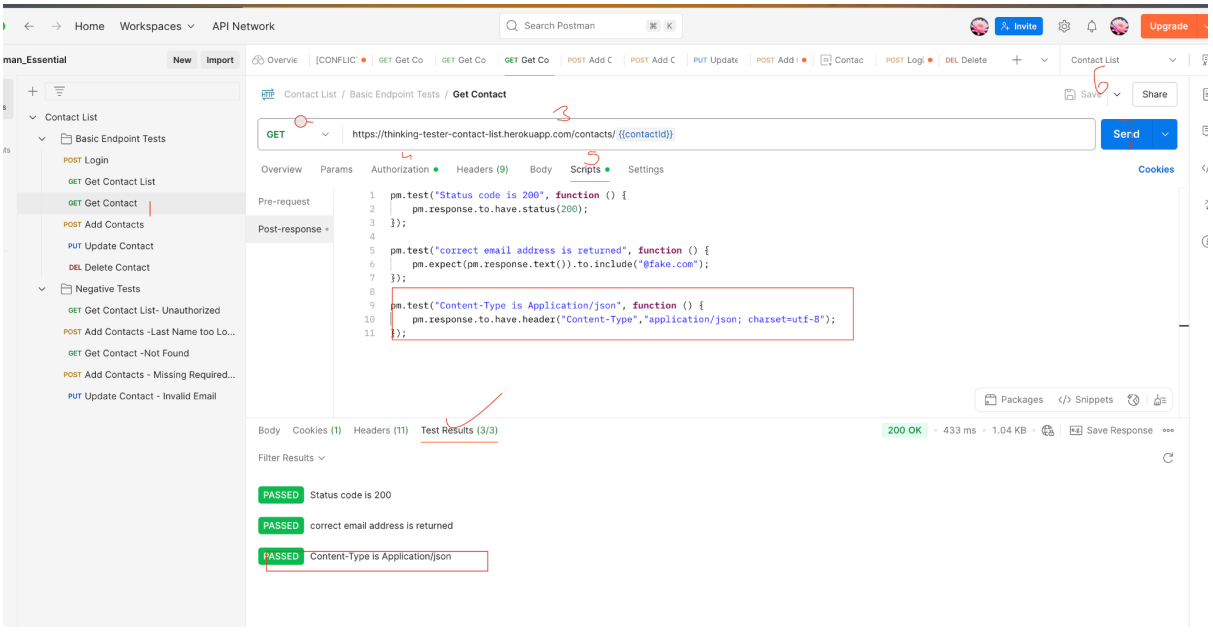


Negative

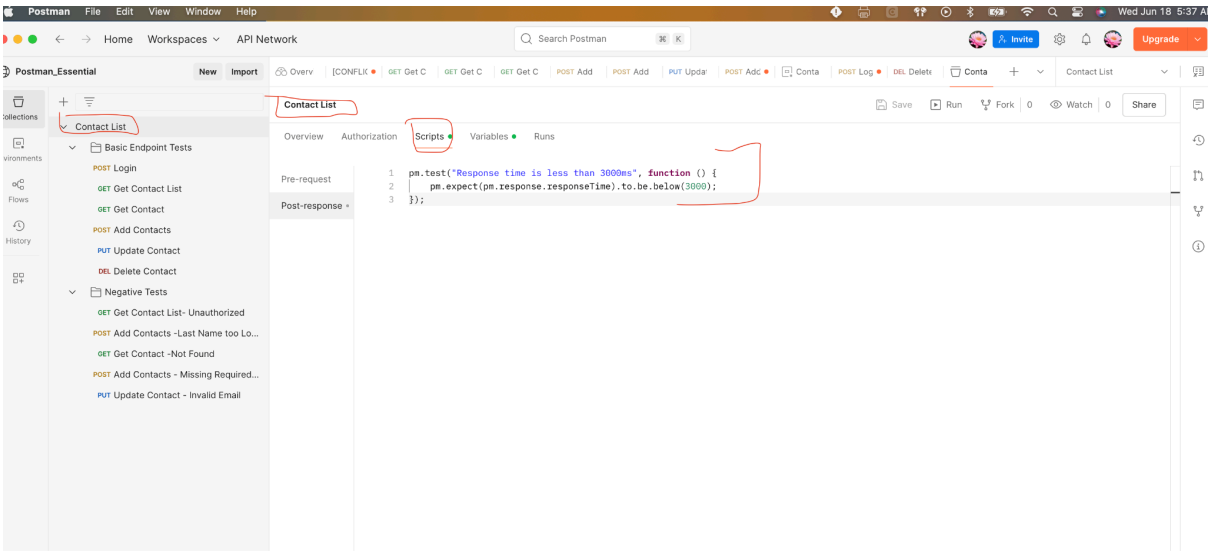


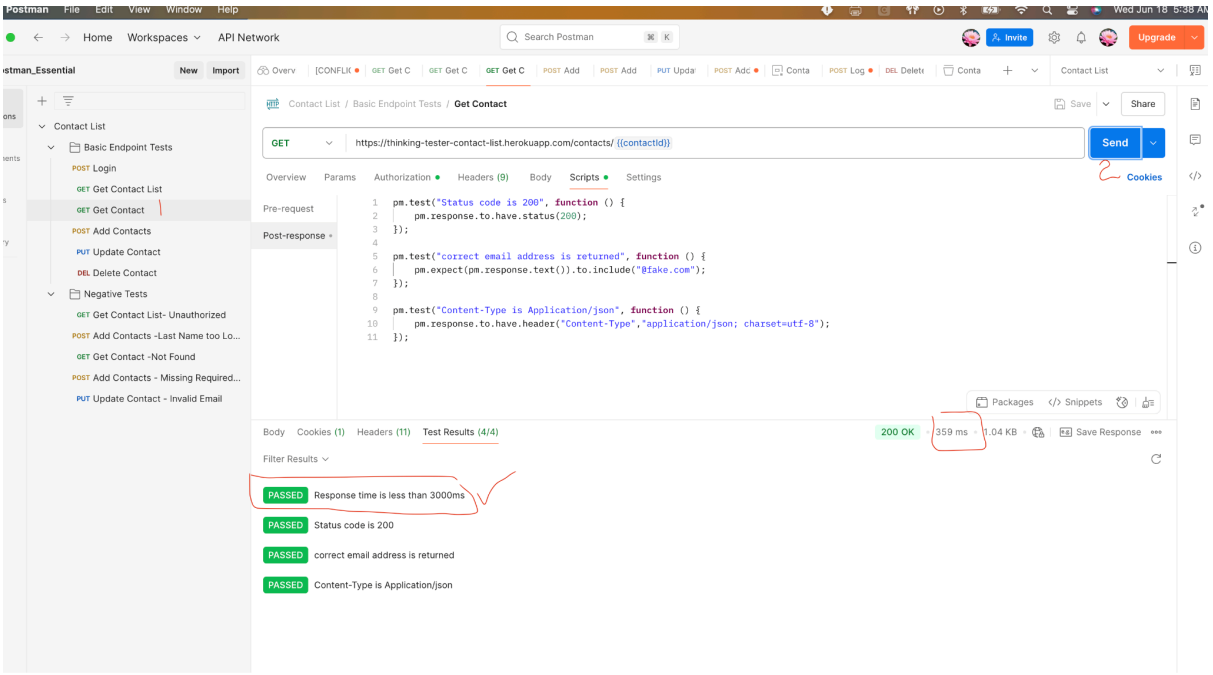
Header and response time assertion

Response header : information passed with an API response that includes additional information about the response, such as the format of the response or any security controls



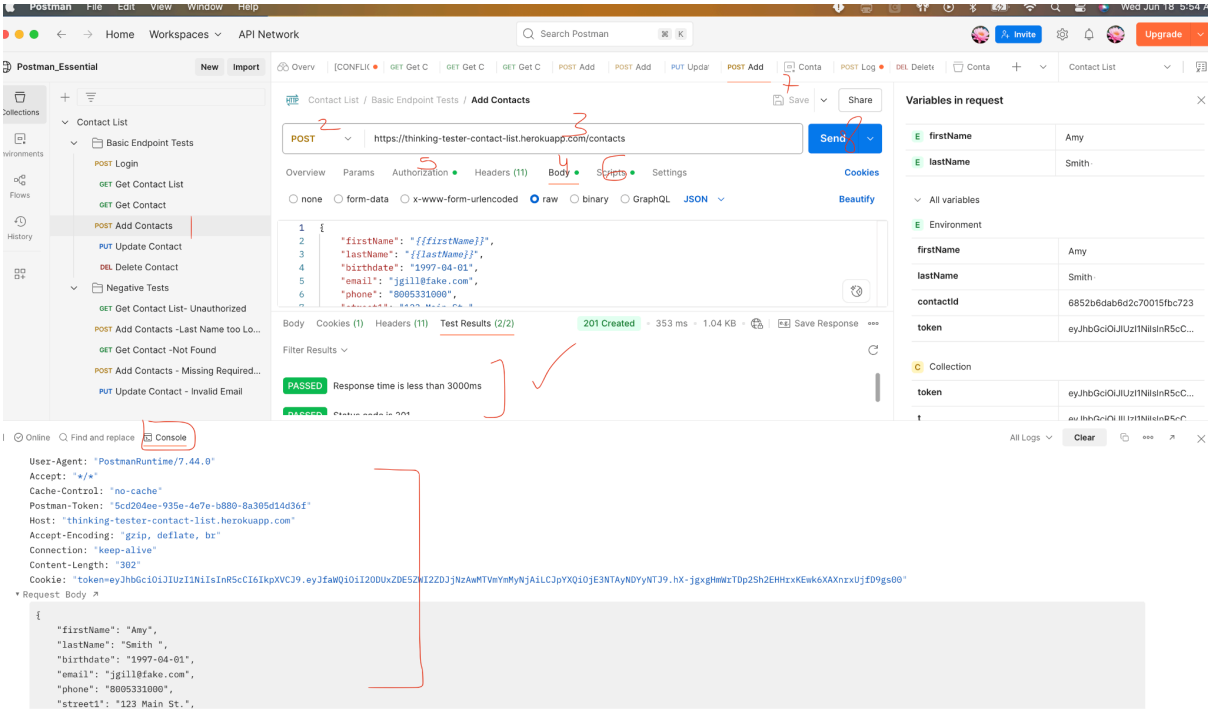
Response time : the time it takes for a request to reach the server and return a response





Debugging with postman console:

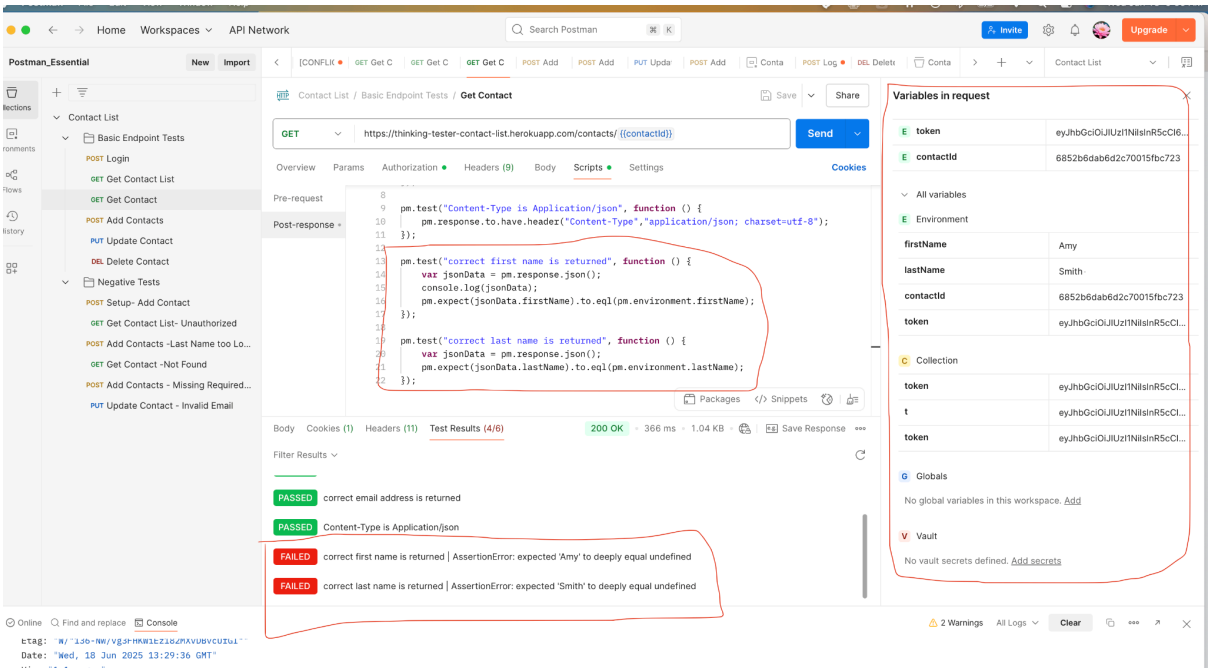
Check that your HTTP verb is correct
Check that authentication token is sent
Check that URL of the request is correct
Check that all variables have populated correctly in the request

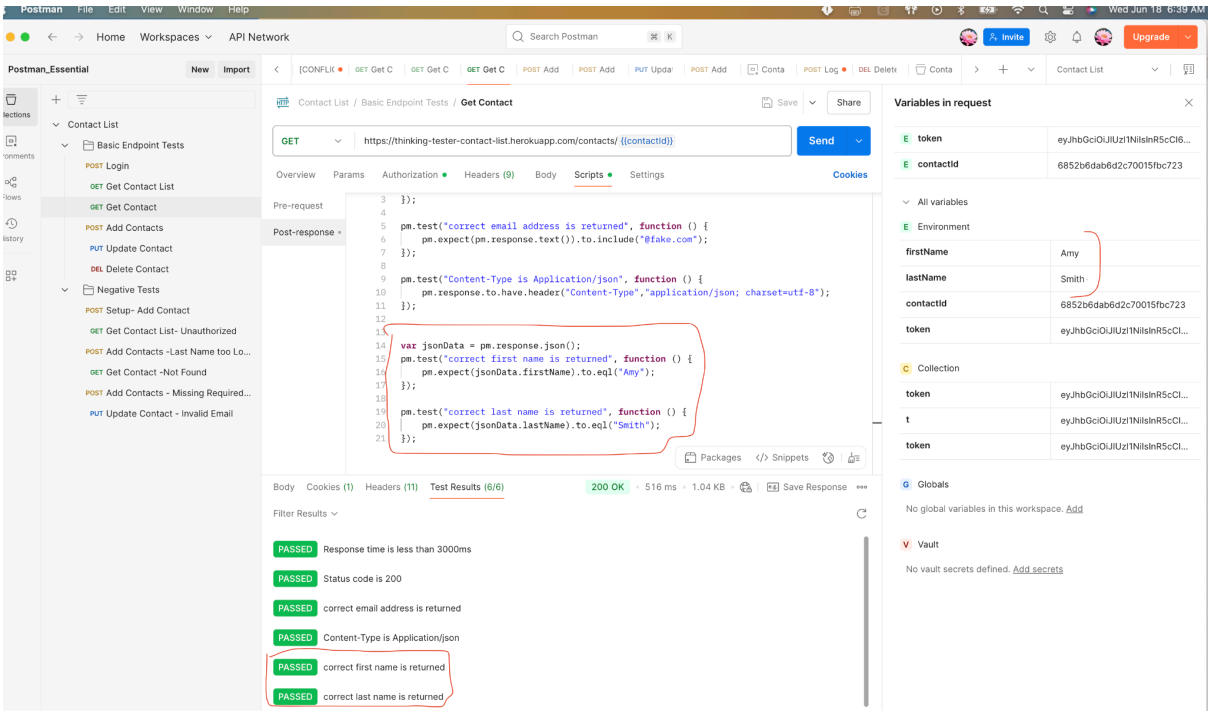


JSON assertions

JSON- javascript object notation

API requests and responses are often in JSON format





Body Assertion	JSON assertion
We are looking for value somewhere in the body	We are looking for specific field value and validating what you expected

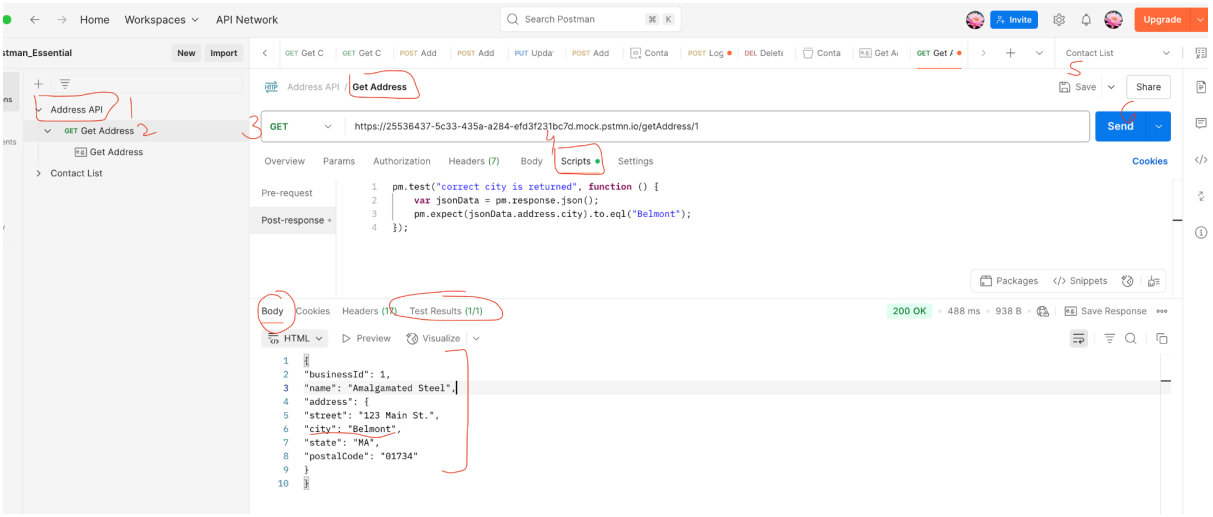
Nested JSON Assertions

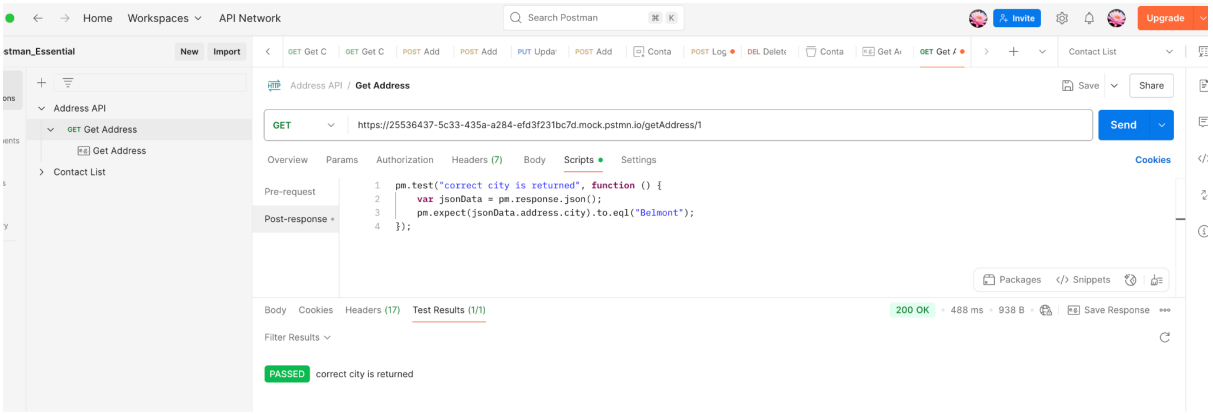
Nested JSON: JSON that has values nested inside another object

For example :

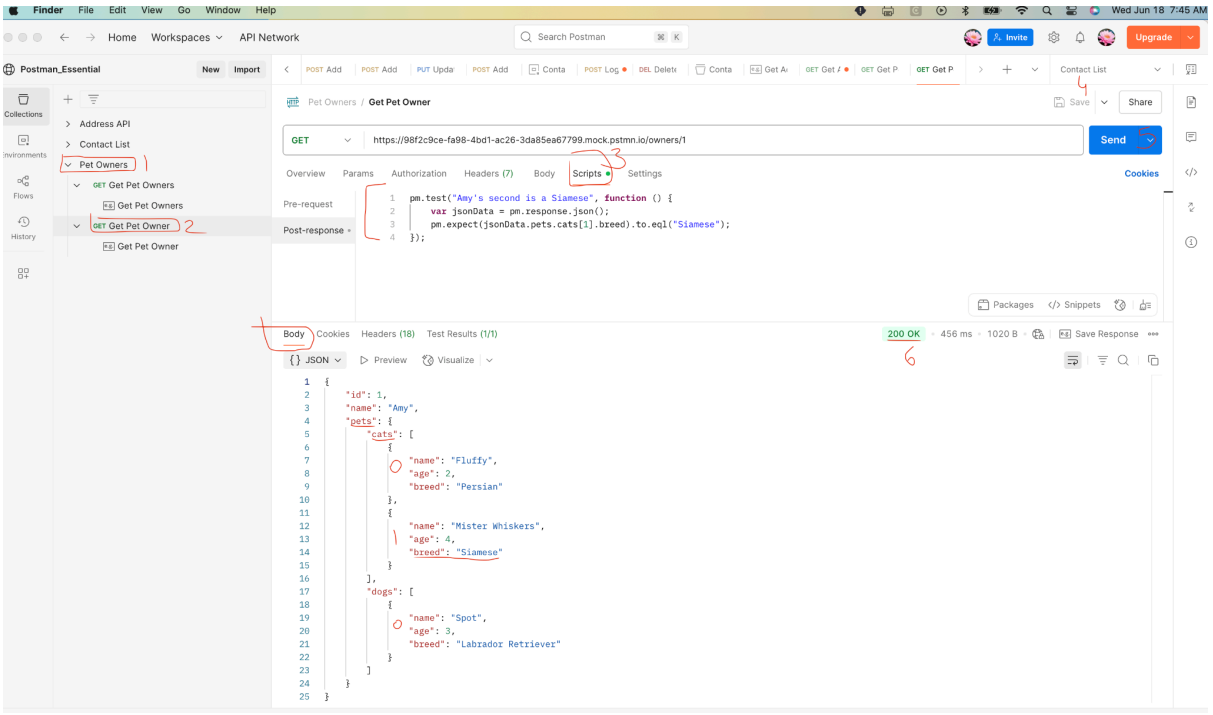
“Address” : {
“Street” : “123 Main St”,
“City” : “belmont”,
“State” : “MA”,
“postalcode”:”23243”

}



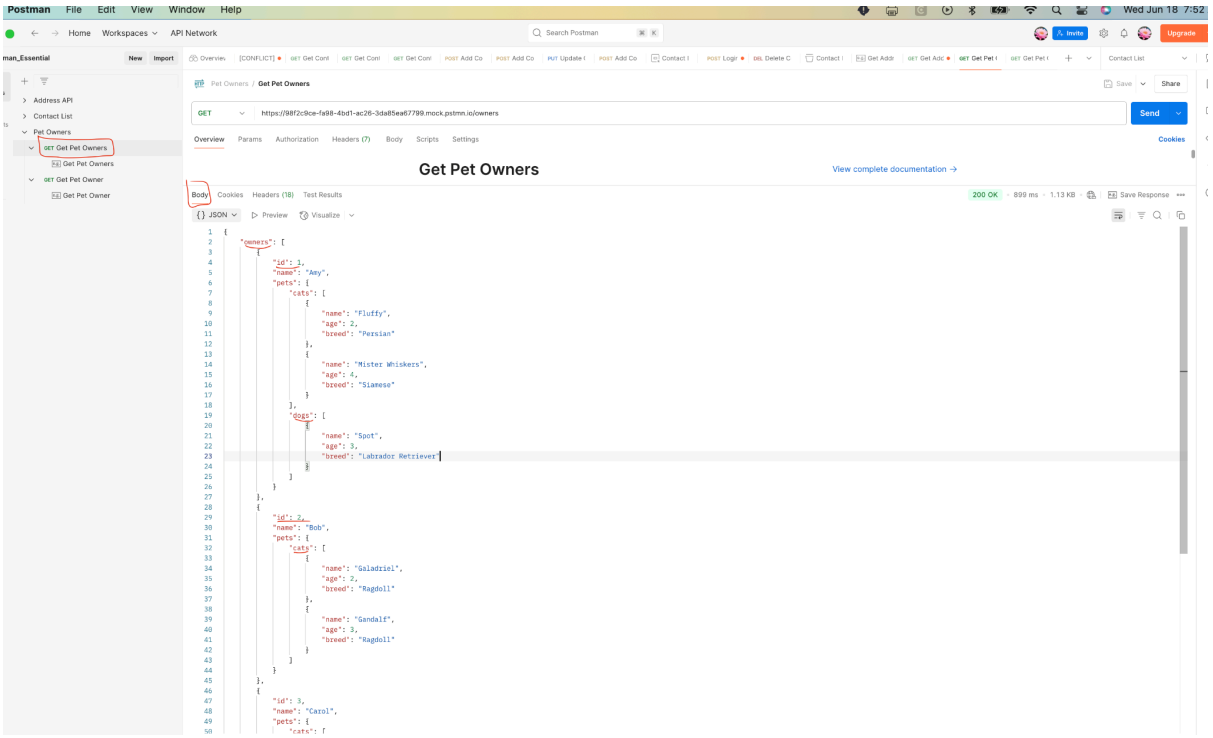


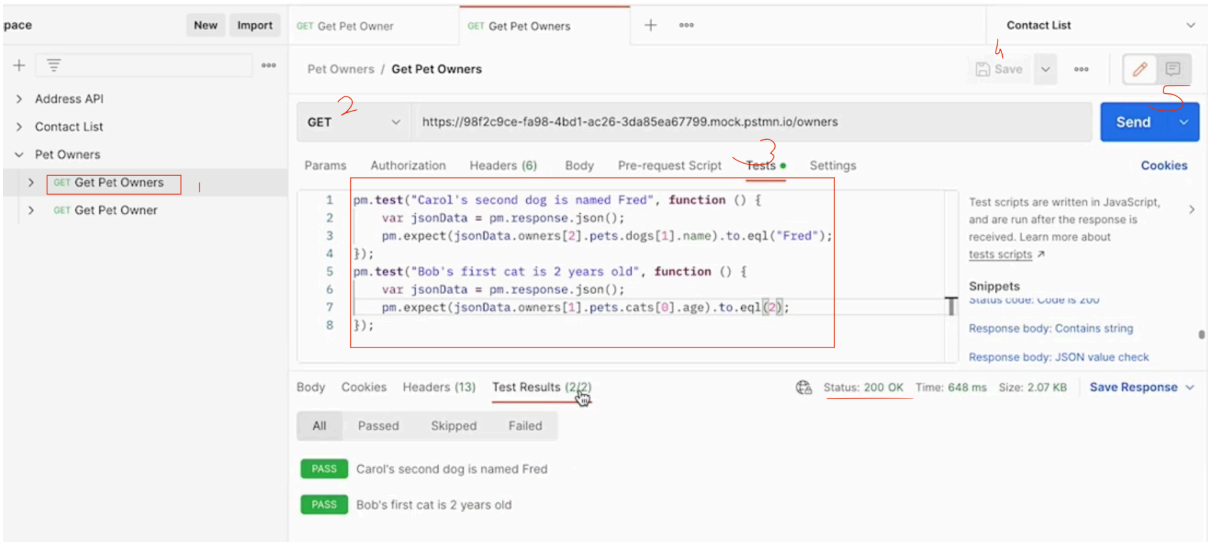
Asserting on Complicated JSON


Pet Owners / Get Pet Owner
GET
https://98f2c9ce-fa98-4bd1-ac26-3da85ea67799.mock.pstmn.io/owners/1
Send
Overview Params Authorization Headers (7) Body Scripts Settings
Pre-request
Post-response
pm.test("Amy's second is a Siamese", function () {
var jsonData = pm.response.json();
pm.expect(jsonData.pets.cats[1].breed).to.eql("Siamese");
});
Body Cookies Headers (18) Test Results (1/1)
200 OK 456 ms 1020 B
{
"id": 1,
"name": "Amy",
"pets": {
"cats": [
{
"name": "Fluffy",
"age": 2,
"breed": "Persian"
},
{
"name": "Mister Whiskers",
"age": 4,
"breed": "Siamese"
}
],
"dogs": [
{
"name": "Spot",
"age": 3,
"breed": "Labrador Retriever"
}
]
}
}


Pet Owners / Get Pet Owner
GET
https://98f2c9ce-fa98-4bd1-ac26-3da85ea67799.mock.pstmn.io/owners/1
Send
Overview Params Authorization Headers (7) Body Scripts Settings
Pre-request
Post-response
pm.test("Amy's second is a Siamese", function () {
var jsonData = pm.response.json();
pm.expect(jsonData.pets.cats[1].breed).to.eql("Siamese");
});
Body Cookies Headers (18) Test Results (1/1)
200 OK 456 ms 1020 B
Filter Results
PASSED Amy's second is a Siamese

2nd example 👍



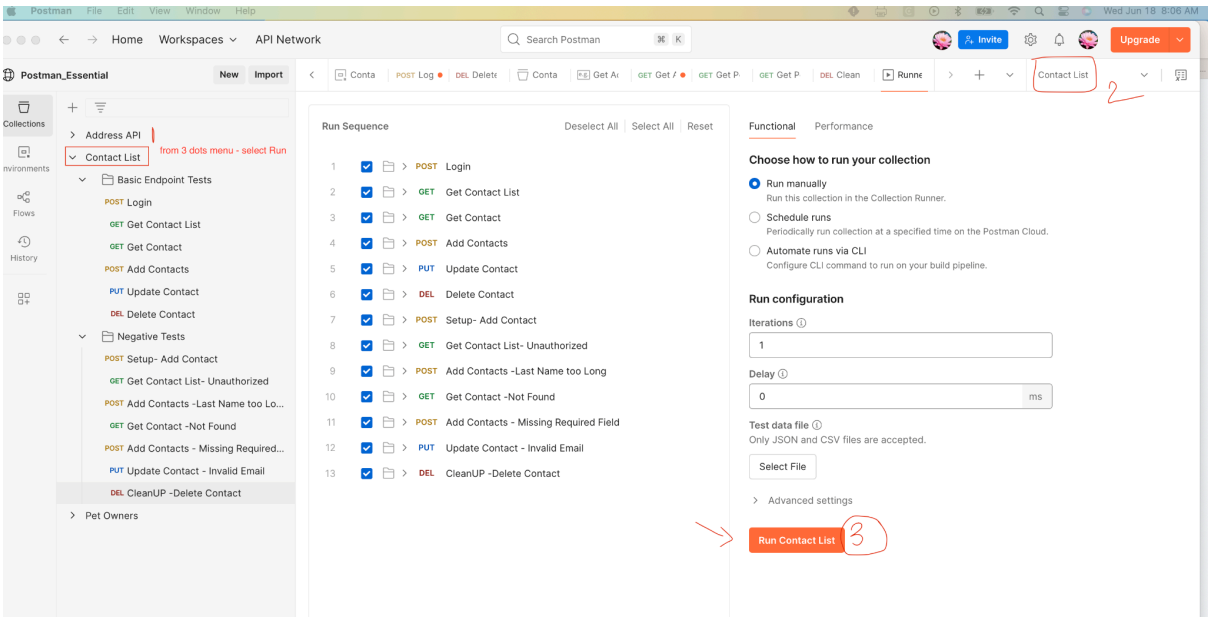


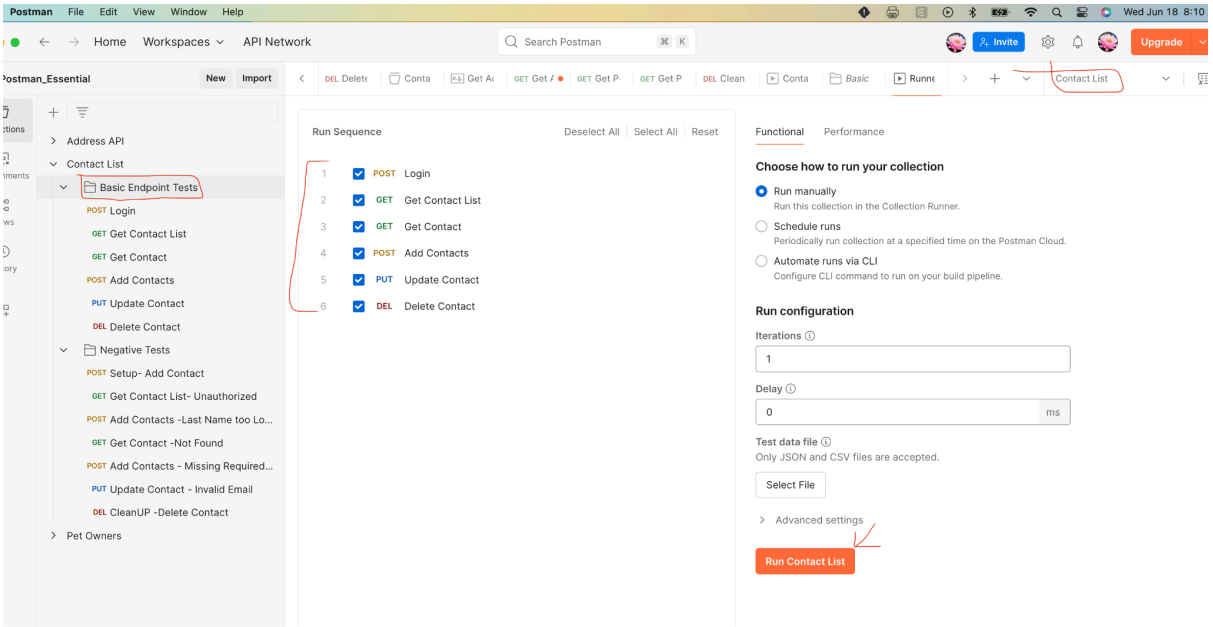
Tips for navigating complicated assertions :

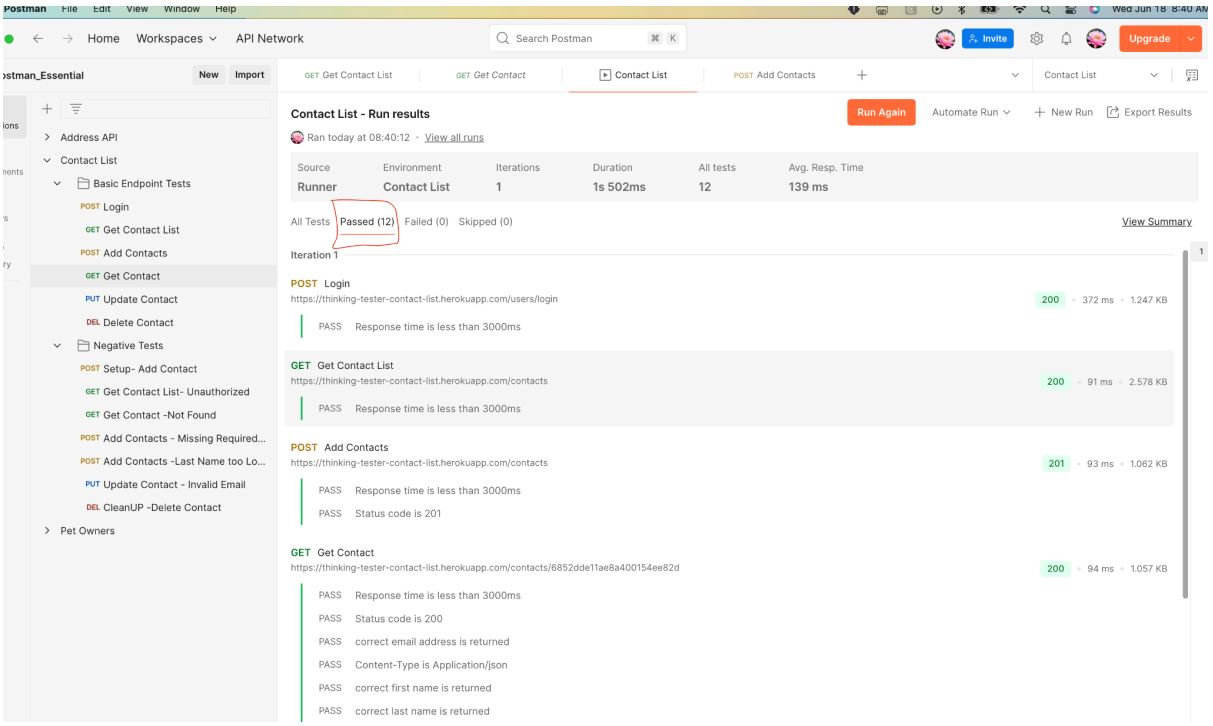
1. Objects are in curly brackets {}
2. Arrays are in square brackets []
3. Arrays will always need a bracketed number, even if there;s only one item in the array; then the number will be [0]
4. Work from left to right. Using the indentations to determine the order of navigation.
5. Anytime there is an indentation, you need a new dot in your navigation

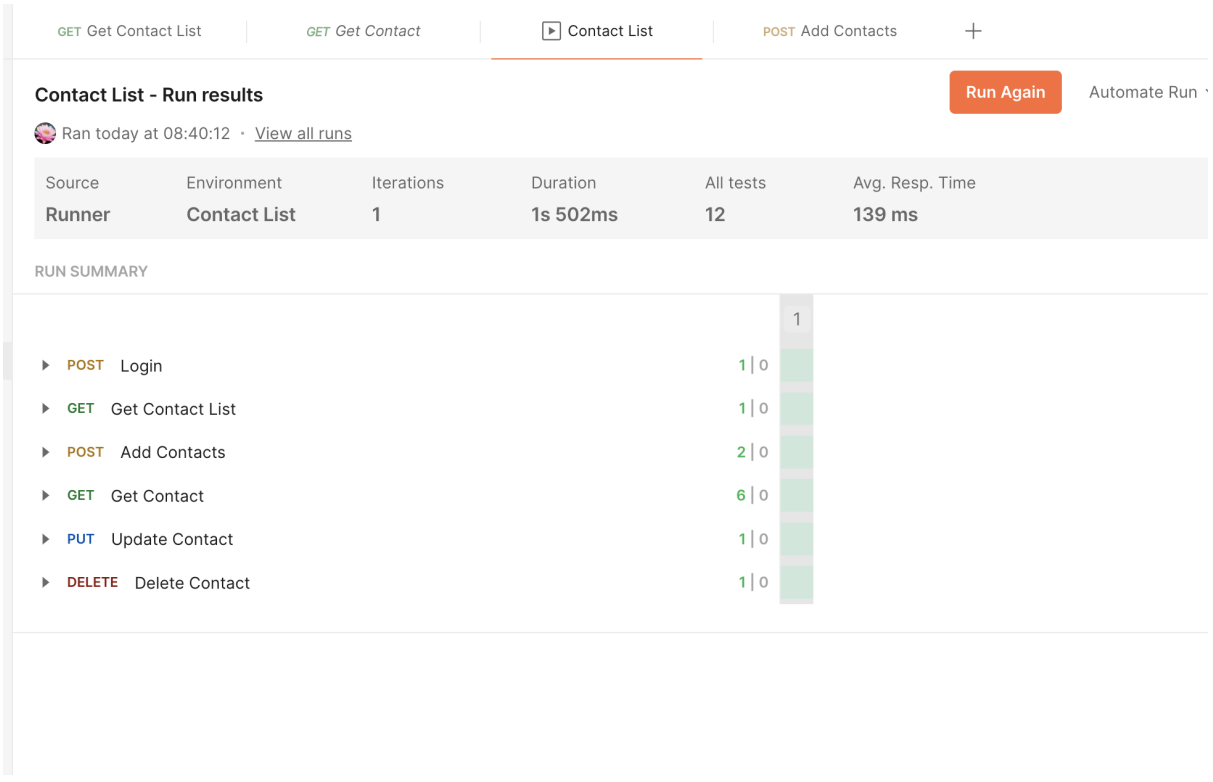
Running the Test Collections

The postman collection runner : A feature of postman that will send in each request of your collection automatically.






Postman File Edit View Window Help
Home Workspaces API Network
Search Postman
Invite
Upgrade
New Import
Address API
Contact List
Basic Endpoint Tests
POST Login
GET Get Contact List
POST Add Contacts
GET Get Contact
PUT Update Contact
DEL Delete Contact
Negative Tests
POST Setup- Add Contact
GET Get Contact List- Unauthorized
GET Get Contact -Not Found
POST Add Contacts - Missing Required...
POST Add Contacts -Last Name too Lo...
PUT Update Contact - Invalid Email
DEL CleanUP -Delete Contact
Pet Owners
GET Get Contact List
GET Get Contact
Contact List
POST Add Contacts
Contact List - Run results
Run Again
Automate Run
New Run
Export Results
Ran today at 08:40:12 · View all runs
Source Environment Iterations Duration All tests Avg. Resp. Time
Runner Contact List 1 1s 502ms 12 139 ms
All Tests Passed (12) Failed (0) Skipped (0)
View Summary
Iteration 1
POST Login
https://thinking-tester-contact-list.herokuapp.com/users/login
200 372 ms 1.247 KB
PASS Response time is less than 3000ms
GET Get Contact List
https://thinking-tester-contact-list.herokuapp.com/contacts
200 91 ms 2.578 KB
PASS Response time is less than 3000ms
POST Add Contacts
https://thinking-tester-contact-list.herokuapp.com/contacts
201 93 ms 1.062 KB
PASS Response time is less than 3000ms
PASS Status code is 201
GET Get Contact
https://thinking-tester-contact-list.herokuapp.com/contacts/6852dde11ae8a400154ee82d
200 94 ms 1.057 KB
PASS Response time is less than 3000ms
PASS Status code is 200
PASS correct email address is returned
PASS Content-Type is Application/json
PASS correct first name is returned
PASS correct last name is returned

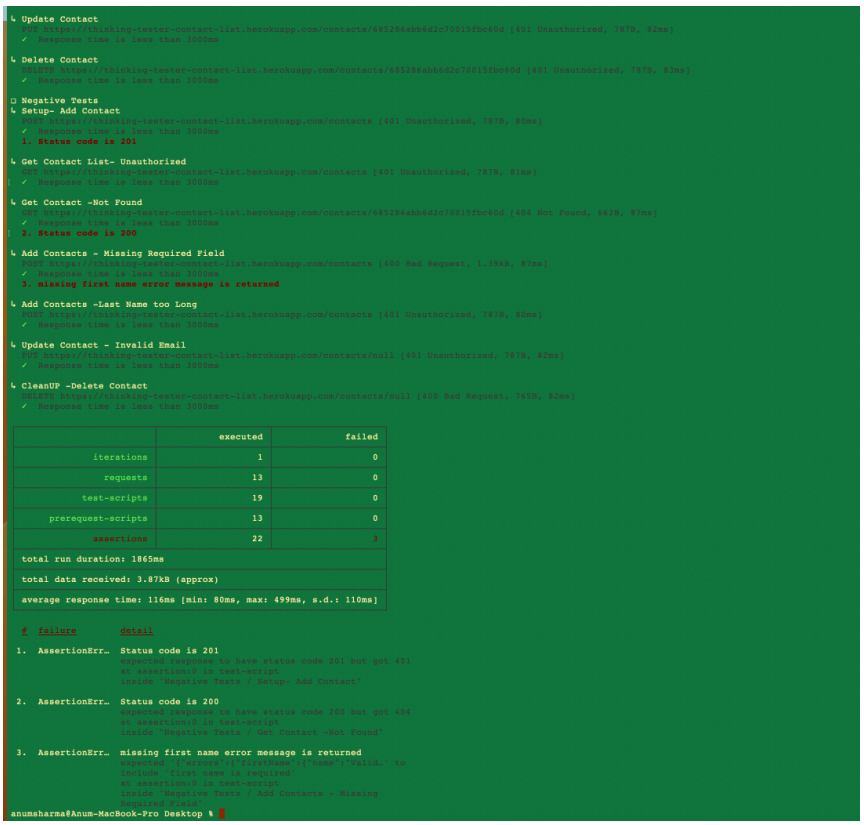

GET Get Contact List
GET Get Contact
Contact List
POST Add Contacts
Contact List - Run results
Run Again
Automate Run
Ran today at 08:40:12 · View all runs
Source Environment Iterations Duration All tests Avg. Resp. Time
Runner Contact List 1 1s 502ms 12 139 ms
RUN SUMMARY
1
POST Login 1 | 0
GET Get Contact List 1 | 0
POST Add Contacts 2 | 0
GET Get Contact 6 | 0
PUT Update Contact 1 | 0
DELETE Delete Contact 1 | 0

Running the collection in newman :

Install [node.js]
Check with command if nodeJS is installed or not
node -v
Check with if npm is installed or not
npm -v
Install newman using command:
npm install -g newman
Check with command if newman is install or not
newman —version
Export the collection and environment variables from postman by exporting in desktop
Go to terminal and run the following command :

newman run ContactList.postman_collection.json -e ContactList.postman_environment.json

To see other commands options, type
newman run -h



If your files are in a different location - for example Projects/Newman/ContactList-

You can navigate to that location :

Newman run Projects/Newman/ContactList contactList.collection.json -e Projects/Newman/ContactList/contactList.environment.json

Practise API:

Contact List app:
https://thinking-tester-contact-list.herokuapp.com

Pet Store API: https://petstore.swagger.io
Poke API: https://pokeapi.co

Integrate the postman tests into your workplace CI/CD platform

Jenkins documentation:
https://learning.postman.com/docs/running-collections/using-newman-cli/integration-with-jenkins

GitLab documentation:
https://learning.postman.com/docs/integrations/available-integrations/ci-integrations/gitlab-ci

Travis CI documentation:
https://learning.postman.com/docs/running-collections/using-newman-cli/integration-with-travis

Circle CI documentation:
https://learning.postman.com/docs/integrations/available-integrations/ci-integrations/circleci

Postman CLI tool:
https://learning.postman.com/docs/postman-cli/postman-cli-overview