Automation Testing

Test Script Creation

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

A Day in the Life of an Automation Testing Engineer

John has been assigned a task to add some validations to REST APIs. He is exploring the Postman tool's test script capability for this task.

After this session, John can add validations to the API responses and update variables before sending the requests via Postman.

Learning Objectives

By the end of this lesson, you will be able to:

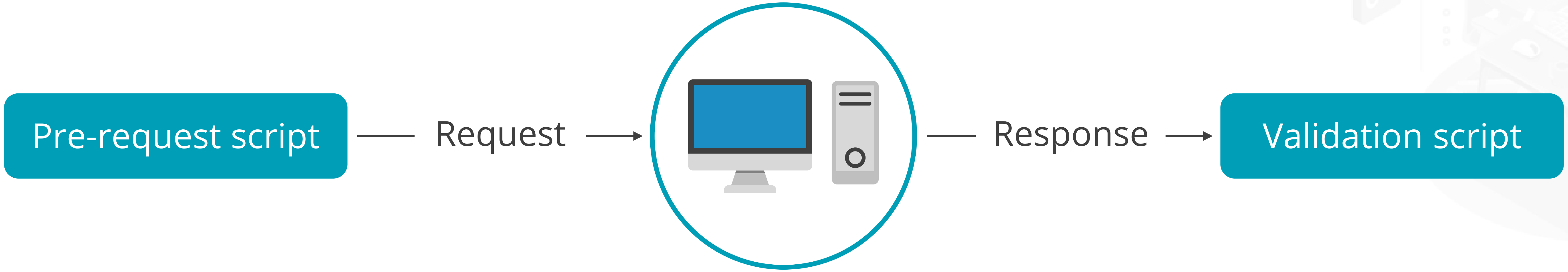
- Define test scripts in Postman
- List steps to create quick scripts in Postman
- Explain how to create test scripts in Postman
- State the debugging process of the test scripts

Test Scripts in Postman

Test Script in Postman

Postman has a powerful integration of script creation that is based on Node.js. Users can create API tests, pass data between requests, and build requests with dynamic parameters. They can add JavaScript code to execute these things. A script is run:

- Before a request is sent to the server
- After a response is received from the server



Pre-request script — Request → — Response → Validation script

Features of Test Scripts

Test scripts run after receiving the response from the server. Various tests are accessible inside the Postman Snippet.

Users can execute numerous Test scripts on a request.

Why Use Scripts in Postman?

Postman scripts allow changes in the behavior of the requests. Users can perform the following tasks using the scripts:

Setting or obtaining variable values, parameters, headers, body data, and so on directly in the request

Debugging (log output to the console)

Why Use Scripts in Postman?

Consider a situation where a user needs to retrieve the user's ID with a Get request after establishing a user with a POST request. They can use scripts to perform the task rather than manually copying and pasting the ID. The scripts help make such tasks easy.

Users can first set a variable ID in the Tests script of the POST request. The variable is then obtained from the Get request's pre-request script and is set in the argument.

It allows the users to rely on the existing data and automate this cycle utilizing Postman Collection.

How to Create Quick Scripts in Postman?

Pre-request Scripts

- Pre-request scripts are lines of code that run Collection requests before the request is executed.

Example:
Users may need to generate any key in the URL or transmit some random variables as a parameter. Pre-request scripts are necessary here.

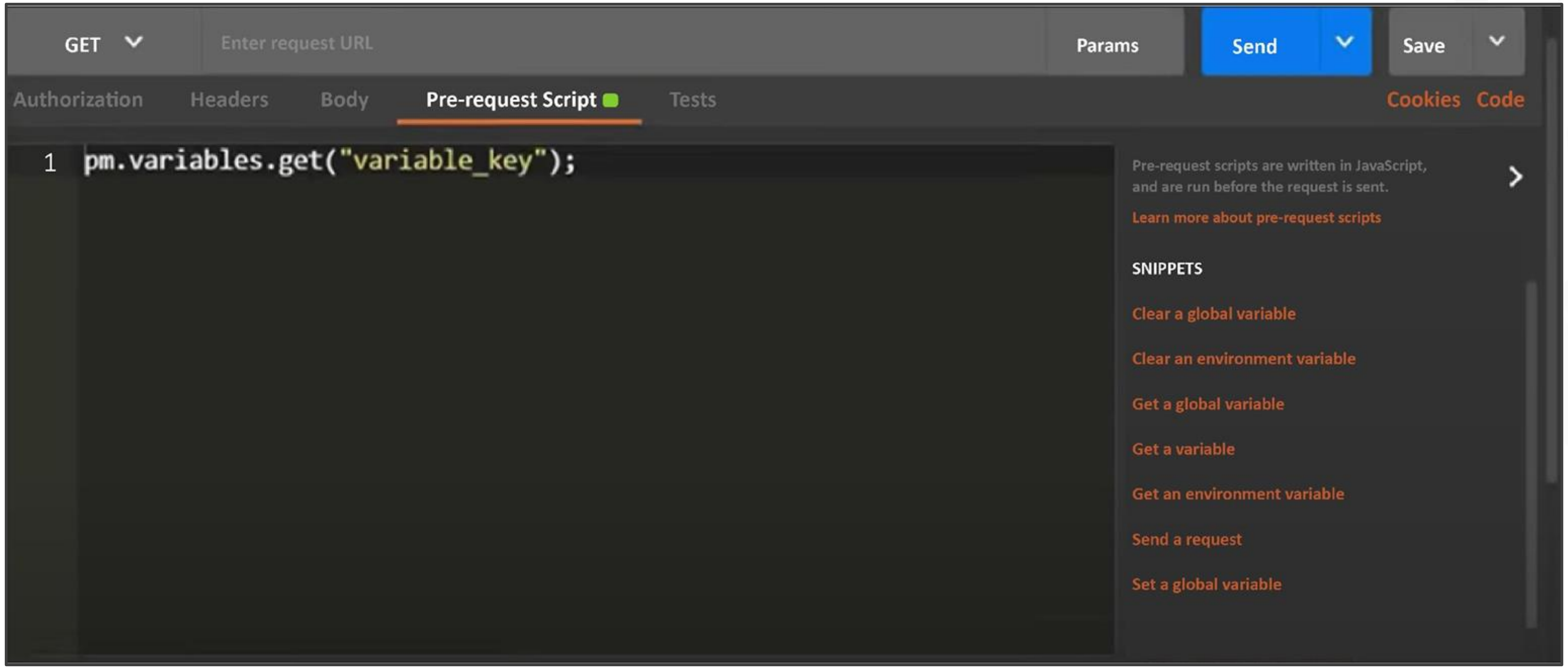
- Additionally, users can get a functional output, such as date, time, timestamp, and so on, by using the pre-request script concept.

Example:
An Environment variable can be set with the value returned from an internal Postman function to include a timestamp in the request headers.

How to Create Quick Scripts?

Postman provides snippets to create basic scripts, such as getting a variable, setting a variable, sending a request, clearing a variable value, and so on quickly.

Users can find the **Snippets** pane when moving to the **pre-request Script** or **Tests** tab.



```
pm.variables.get("variable_key");
```

Pre-request scripts are written in JavaScript, and are run before the request is sent.

Learn more about pre-request scripts

SNIPPETS

Clear a global variable

Clear an environment variable

Get a global variable

Get a variable

Get an environment variable

Send a request

Set a global variable

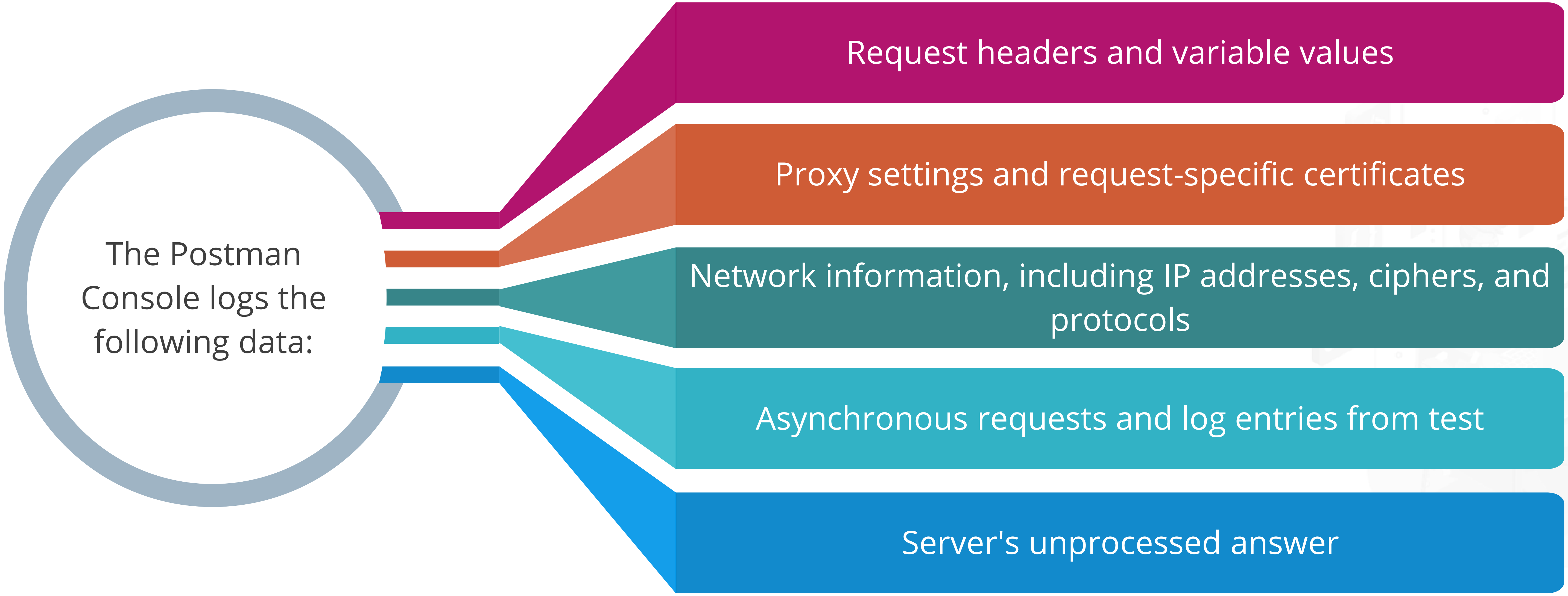
How to Debug Scripts in Postman?

Debugging Scripts in Postman

Postman logs every request it sends in the console. It lets the users know the changes' details when they send a request. The users can use the Postman console to help troubleshoot the issue when an API is not behaving as it should.

It is best to keep the console open when debugging, as it will enable network calls and make log messages more visible.

Debugging Scripts in Postman



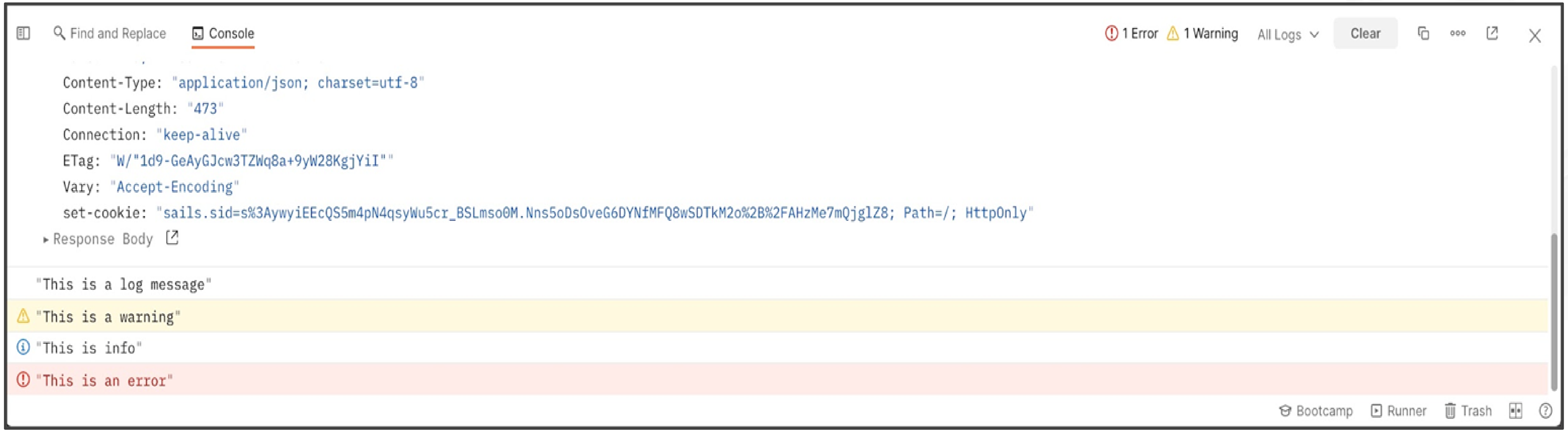
The Postman Console logs the following data:

- Request headers and variable values
- Proxy settings and request-specific certificates
- Network information, including IP addresses, ciphers, and protocols
- Asynchronous requests and log entries from test
- Server's unprocessed answer

How to Debug Scripts in Postman?

Users can implement different statements mentioned below to debug tests and check the unexpected behaviors:

1. console.log()
2. console.info()
3. console.warn()
4. console.error()



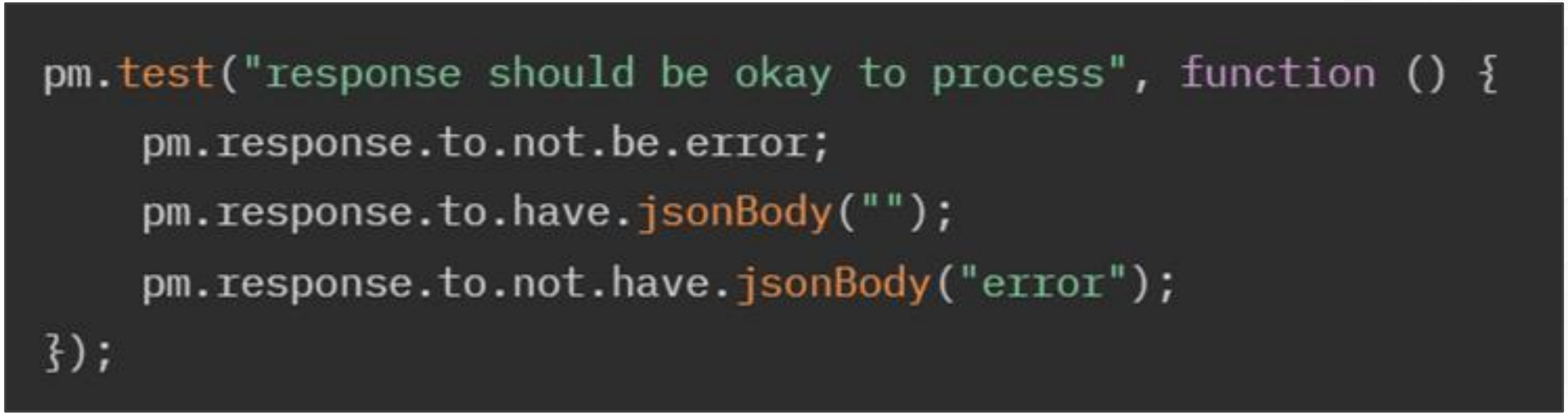
```
    Content-Type: "application/json; charset=utf-8"
    Content-Length: "473"
    Connection: "keep-alive"
    ETag: "W/"1d9-GeAyGJcw3TZWq8a+9yW28KgjYiI""
    Vary: "Accept-Encoding"
    set-cookie: "sails.sid=s%3AywyiEEcQS5m4pN4qsyWu5cr_BSLmso0M.Nns5oDsOveG6DYNfMFQ8wSDTkM2o%2B%2FAHzMe7mQjglZ8; Path=/; HttpOnly"
  ▸ Response Body

"This is a log message"
⚠ "This is a warning"
ⓘ "This is info"
ⓘ "This is an error"
```

FULL STACK

How to Create the First Test Script in Postman?

Create Scripts in Postman

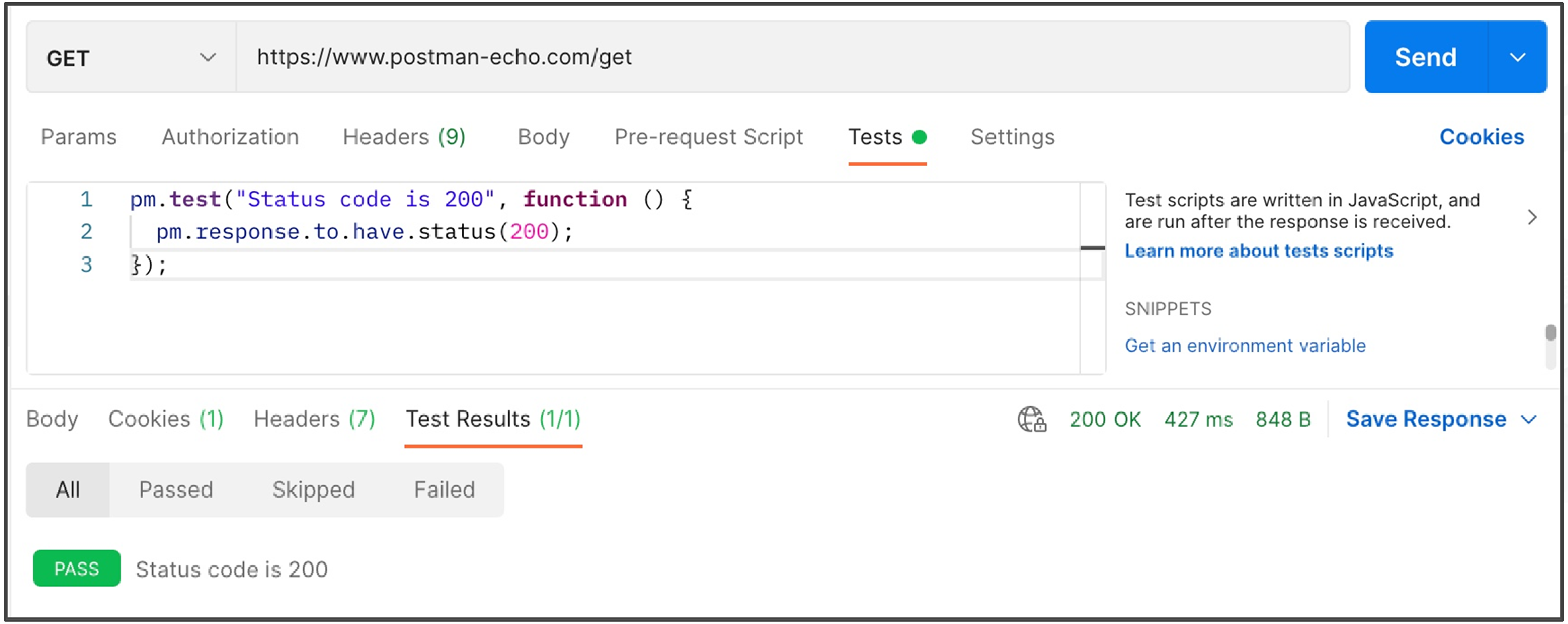
Scripts can carry out assertions on the response received and pass data between requests. Advanced JavaScript codes can perform complex operations that are not readily available with snippets.



```
pm.test("response should be okay to process", function () {
    pm.response.to.not.be.error;
    pm.response.to.have.jsonBody("");
    pm.response.to.not.have.jsonBody("error");
});
```

How to Create the First Test?

Users can create the first script in Postman by creating a **Get** request and then accessing the **Tests** tab to write the JavaScript code and perform any validation.



```
pm.test("Status code is 200", function () {
  pm.response.to.have.status(200);
});
```

Test scripts are written in JavaScript, and are run after the response is received.

Learn more about tests scripts

SNIPPETS

Get an environment variable

PASS Status code is 200

FULL STACK

How to Create Test Scripts in Postman?

Script Hierarchy in Postman

Scripts run in the following order for each request in a Collection:

Every request in a Collection is preceded by executing a pre-request script linked to the Collection.

Every request in a folder is preceded by executing a pre-request script linked with that folder.

After each request in a Collection, a Test script connected with that Collection is executed.

After each request in a folder, a Test script connected with that folder is executed.

Key Takeaways

- There are two types of scripts: a pre-request script and a Validation script.
- Pre-request scripts are primarily used to get or set specific data in the request body before sending the request.
- Validation scripts or Post Response scripts are used to validate the response received from the server.
- Snippets can be used to create quick test scripts.
- A JavaScript code can be manually typed to write advanced scripts.