Homework 03:

```
/*
   Given
       https://reqres.in/api/users/3
   When
       User sends a GET Request to the url
   Then
       HTTP Status Code should be 200
   And
       Content Type should be JSON
   And
       Status Line should be HTTP/1.1 200 OK
*/
```

"Homework 03," which involves verifying the response from a specific API endpoint, you can employ either a software application like Postman for an interactive interface or a script written in JavaScript using Node.js with the Axios library for programmatic verification. Below, detailed methodologies are presented for both approaches, ensuring compliance with the stipulated criteria: the HTTP status code, content type, and status line.

Using Postman for Interactive API Testing

1. **Initialize Postman**:
    - Launch the Postman application.
2. **Configure the GET Request**:
    - Set the request type to GET.
    - Input the URL: **https://reqres.in/api/users/3**.
3. **Implement Tests**:
    - Navigate to the "Tests" tab within the request setup.
    - Enter the following JavaScript code to validate the response against the given criteria:

// Check for the correct HTTP status code pm.test("Status code is 200", function () { pm.response.to.have.status(200); }); // Verify the content type of the response pm.test("Content-Type is JSON", function () { pm.response.to.have.header("Content-Type", "application/json; charset=utf-8"); }); // Confirm the status line correctness pm.test("Status Line is HTTP/1.1 200 OK", function () { pm.expect(pm.response.to.have.status("HTTP/1.1 200 OK")); });

4. **Execute the Request**:
    - Send the request and review the results in the "Test Results" tab to confirm compliance with the specified requirements.

Employing JavaScript and Axios for Programmatic API Testing

1. **Prepare the Testing Environment**:
    - Ensure Node.js is installed.
    - Create a new directory, initialize a Node.js project, and install the Axios library:

mkdir my-test-project cd my-test-project npm init -y npm install axios

2. **Script the Test**:
    - Create a file named **test.js**.
    - Write the following JavaScript code to programmatically test the API endpoint:

const axios = require('axios'); axios.get('https://reqres.in/api/users/3') .then(response => { console.log("Status code check:", response.status === 200); // Should output true console.log("Content-Type check:", response.headers['content-type'].includes("application/json; charset=utf-8")); // Should output true console.log("Status Line check:", response.statusText === "OK"); // Should output true }) .catch(error => { console.error('Error during API call', error); });

3. **Run the Test**:
    - Execute the script by running:

node test.js

This script conducts a GET request to the specified endpoint and logs whether each response attribute meets the specified conditions—ideal for automated testing environments or continuous integration workflows.

Both methods provide robust solutions for testing API endpoints against predefined conditions, catering to different needs and preferences in software development and testing scenarios.