Seek - Next

Release 0.1.0

Team 18

Aug 05, 2024

CONTENTS:

CHAPTER

ONE

TEST CASES FOR */COMPUTE* ENDPOINT

Python Code Test

```python
def test_compute_code_python():
    response = client.post(
        "/compute",
        json={
            "code": (
                "import sys\n"
                "class Solution:\n"
                "    def print_statement(s):\n"
                "        print(\"s\",s)\n"
                "def main():\n"
                "    # Read input from stdin\n"
                "    input_data = sys.stdin.read().strip()\n"
                "    Solution.print_statement(input_data)\n"
                "if __name__ == \"__main__\":     main()"
            ),
            "user_id": "1",
            "language": "python",
            "problem_id": "1"
        }
    )
    assert response.status_code == 200
    data = response.json()
    assert "message" in data
    assert "result" in data
    assert data["message"] == "Passed 0 out of 2 test cases"
    assert len(data["result"]) == 2
    assert data["result"][0]["error"] == "Test case failed"
    assert data["result"][0]["input_data"] == "hello"
    assert data["result"][0]["expected_output"] == "hello"
    assert data["result"][0]["your_output"] == "s hello\n"
    assert data["result"][1]["error"] == "Test case failed"
    assert data["result"][1]["input_data"] == "world"
    assert data["result"][1]["expected_output"] == "world"
    assert data["result"][1]["your_output"] == "s world\n"
```

Java Code Test

```java
public class solution {
    public static void main(String[] args) {
```

(continues on next page)

(continued from previous page)

```
        // Create a Scanner object to read input from stdin
        Scanner scanner = new Scanner(System.in);

        // Read all input from stdin
        StringBuilder inputBuilder = new StringBuilder();
        while (scanner.hasNextLine()) {
            inputBuilder.append(scanner.nextLine());
            if (scanner.hasNextLine()) {
                inputBuilder.append("\n");
            }
        }
        String inputData = inputBuilder.toString().trim();

        // Print the input
        // Print an additional statement
        // Use the input in another statement
        // Close the scanner
        scanner.close();
        System.out.print(inputData);
    }
}
```

CHAPTER

TWO

TEST CASES FOR */COMPUTE* ENDPOINT

Java Code

```
def test_compute_code_java():
    response = client.post(
        "/compute",
        json={
            "code": (
                "import java.util.Scanner;\n\n"
                "public class solution {\n"
                "    public static void main(String[] args) {\n"
                "        // Create a Scanner object to read input from stdin\n"
                "        Scanner scanner = new Scanner(System.in);\n"
                "        \n"
                "        // Read all input from stdin\n"
                "        StringBuilder inputBuilder = new StringBuilder();\n"
                "        while (scanner.hasNextLine()) {\n"
                "            inputBuilder.append(scanner.nextLine());\n"
                "            if (scanner.hasNextLine()) {\n"
                "                inputBuilder.append(\"\\n\");\n"
                "            }\n"
                "        }\n"
                "        String inputData = inputBuilder.toString().trim();\n"
                "        \n"
                "        // Print the input\n"
                "        // Print an additional statement\n"
                "        // Use the input in another statement\n"
                "// Close the scanner\n"
                "        scanner.close();\n"
                "        System.out.print(inputData);\n"
                "    }\n"
                "}"
            ),
            "user_id": "1",
            "language": "java",
            "problem_id": "1"
        }
    )
    assert response.status_code == 200
    data = response.json()
    assert "message" in data
```

(continues on next page)

(continued from previous page)

```
    assert "result" in data
    assert data["message"] == "Passed 2 out of 2 test cases"
    assert len(data["result"]) == 2
    assert data["result"][0]["error"] == "Test case passed"
    assert data["result"][0]["input_data"] == "hello"
    assert data["result"][0]["expected_output"] == "hello"
    assert data["result"][0]["your_output"] == "hello"
    assert data["result"][1]["error"] == "Test case passed"
    assert data["result"][1]["input_data"] == "world"
    assert data["result"][1]["expected_output"] == "world"
    assert data["result"][1]["your_output"] == "world"
```