Seek - Next

Release 0.1.0

Team 18

Aug 06, 2024

CONTENTS:

This document provides details about the unit tests for the API endpoints related to code submissions. Each test case includes information about the API being tested, the inputs, expected output, actual output, and the result.

CHAPTER

ONE

1. TEST CASE: `TEST_READ_MAIN`

- **API being tested: /**
- **Inputs:** None
- **Expected Output:**

```
{
  "message": "Hello World from Seek Next!"
}
```

- **Actual Output:**

```
{
  "message": "Hello World from Seek Next!"
}
```

- **Result:** Success

CHAPTER

TWO

2. TEST CASE: `TEST_COMPUTE_CODE_PYTHON`

- **API being tested:** */compute*
- **Inputs:**


```
{
  "problem_id": "wxyz",
  "user_id": "1",
  "language": "python",
  "code": "import sys\nclass Solution:\n    def print_statement(s):\n        print(s)\n\ndef main():\n    # Read input from stdin\n    input_data = sys.stdin.read().strip()\n    Solution.print_statement(input_data)\n\nif __name__ == \"__main__\":\n    main()"
}
```


- **Expected Output:**


```
{
  "message": "Passed 1 out of 1 test cases",
  "result": [
    {"error": "Test case passed", "input_data": "hello", "expected_output": "hello", "your_output": "hello\n"}
  ]
}
```


- **Actual Output:**


```
{
  "message": "Passed 1 out of 1 test cases",
  "result": [
    {"error": "Test case passed", "input_data": "hello", "expected_output": "hello", "your_output": "hello\n"}
  ]
}
```


- **Result:** Success

CHAPTER

THREE

3. TEST CASE: `TEST_COMPUTE_CODE_JAVA`

- **API being tested:** */compute*
- **Inputs:**


```
{
  "problem_id": "wxyz",
  "user_id": "1",
  "language": "java",
  "code": "import java.util.Scanner;\n\npublic 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        System.
↪out.print(inputData);\n\n        // Close the scanner\n        scanner.close();\n␣
↪    }\n}"
}
```


- **Expected Output:**


```
{
  "message": "Passed 1 out of 1 test cases",
  "result": [
    {"error": "Test case passed", "input_data": "hello", "expected_output": "hello",
↪ "your_output": "hello"}
  ]
}
```


- **Actual Output:**


```
{
  "message": "Passed 1 out of 1 test cases",
  "result": [
    {"error": "Test case passed", "input_data": "hello", "expected_output": "hello",
↪ "your_output": "hello"}
  ]
}
```


- **Result:** Success

CHAPTER

FOUR

4. TEST CASE: `TEST_COMPUTE_CODE_INVALID_PYTHON_CODE`

- **API being tested:** */compute*
- **Inputs:**


```
{
  "problem_id": "wxyz",
  "user_id": "1",
  "language": "python",
  "code": "import sys\nclass Solution:\n    def print_statement(s):\n        ␣
↪print(s)\n\ndef main():\n    # Read input from stdin\n    input_data = sys.stdin.
↪read().strip()\n    Solution.print_statement(input_data\n\nif __name__ == \"__
↪main__\":\n    main()"
}
```


- **Expected Output:**


```
{
  "message": "Passed 0 out of 1 test cases",
  "result": [
    {"error": "SyntaxError", "input_data": "hello", "expected_output": "hello",
↪"your_output": ""}
  ]
}
```


- **Actual Output:**


```
{
  "message": "Passed 0 out of 1 test cases",
  "result": [
    {"error": "SyntaxError", "input_data": "hello", "expected_output": "hello",
↪"your_output": ""}
  ]
}
```


- **Result:** Success

CHAPTER

FIVE

5. TEST CASE: `TEST_COMPUTE_CODE_INVALID_JAVA_CODE`

- **API being tested:** */compute*
- **Inputs:**


```
{
  "problem_id": "1",
  "user_id": "1",
  "language": "java",
  "code": "import java.util.Scanner;\n\npublic 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        System.
↪out.print(inputData)\n\n        // Close the scanner\n        scanner.close();\n ␣
↪  }\n}"
}
```


- **Expected Output:**


```
{
  "message": "Passed 0 out of 1 test cases",
  "result": [
    {"error": "Compilation error", "input_data": "hello", "expected_output": "hello
↪", "your_output": ""}
  ]
}
```


- **Actual Output:**


```
{
  "message": "Passed 0 out of 1 test cases",
  "result": [
    {"error": "Compilation error", "input_data": "hello", "expected_output": "hello
↪", "your_output": ""}
  ]
}
```


- **Result:** Success

6. TEST CASE: `TEST_COMPUTE_CODE_NO_PROBLEM_INFO`

- **API being tested:** */compute*
- **Inputs:**


```
{
  "problem_id": "pro",
  "user_id": "1",
  "language": "python",
  "code": "import sys\nclass Solution:\n    def print_statement(s):\n        print(s)\n\ndef main():\n    # Read input from stdin\n    input_data = sys.stdin.read().strip()\n    Solution.print_statement(input_data)\n\nif __name__ == \"__main__\":\n    main()"
}
```


- **Expected Output:**


```
{
  "detail": "Problem not found"
}
```


- **Actual Output:**


```
{
  "detail": "Problem not found"
}
```


- **Result:** Success

CHAPTER

SEVEN

7. TEST CASE: `TEST_COMPUTE_CODE_EMPTY_PROBLEM_ID`

- **API being tested:** */compute*
- **Inputs:**


```
{
  "problem_id": "",
  "user_id": "1",
  "language": "python",
  "code": "import sys\nclass Solution:\n    def print_statement(s):\n        print(s)\n\ndef main():\n    # Read input from stdin\n    input_data = sys.stdin.read().strip()\n    Solution.print_statement(input_data)\n\nif __name__ == \"__main__\":\n    main()"
}
```


- **Expected Output:**


```
{
  "detail": "Invalid input"
}
```


- **Actual Output:**


```
{
  "detail": "Invalid input"
}
```


- **Result:** Success

CHAPTER

EIGHT

8. TEST CASE: `TEST_ADD_CODE_INFO`

- **API being tested:** */add-code-info*
- **Inputs:**

```
{
  "problem_id": "1001_test_unique_id",
  "total_test_cases": "1",
  "test_cases": [
    {"input": "hello", "expected_output": "hello"}
  ]
}
```

- **Expected Output:**

```
{
  "message": "Code information added successfully"
}
```

- **Actual Output:**

```
{
  "message": "Code information added successfully"
}
```

- **Result:** Success

CHAPTER

NINE

9. TEST CASE: `TEST_DELETE_CODE_INFO`

- **API being tested:** */delete-code-info*
- **Inputs:**

```
{
  "problem_id": "1001_test_unique_id"
}
```

- **Expected Output:**

```
{
  "message": "Code information deleted successfully"
}
```

- **Actual Output:**

```
{
  "message": "Code information deleted successfully"
}
```

- **Result:** Success