

Test Cases Documentation for Server-Client-Config

Introduction

This document lists all the test cases used to validate the functionality of Server-Client-Config. It covers unit tests, integration tests, and system tests.

Test Case Categories

- **Unit Tests**
- **Integration Tests**
- **System Tests**

Test Cases

Test Case ID	Description	Category	Test Steps	Expected Result VS Actual Result	Status
TC-01	Test on valid options (input)	Unit Test	Call <code>validate_inputs(options)</code> , with the options being valid	Same	passed
TC-02	Test on no input is provided	Unit Test	Call <code>validate_inputs([])</code> , with the options being empty	Same	passed
TC-03	Test on options contains special characters, spaces	Unit Test	Call <code>validate_inputs(["options_1", "option 2"])</code> , with options contains special characters	Same	passed
TC-04	Test on duplicate inputs	Unit Test	Call <code>validate_inputs(["option1", "option1"])</code> , with options contains some duplicates	Same	passed
TC-05	Test inputs with empty string	Unit Test	Call <code>validate_inputs(["", "option1"])</code> , with options contains empty string	Same	passed
TC-06	Test generate combinations with no options	Unit Test	Call <code>generate_combinations([])</code> , with no options provided	Same	passed

TC-07	Test generate combinations with one option	Unit Test	Call generate_combinations(["option1"])	Same	passed
TC-08	Test generate combinations with multiple options	Unit Test	Call generate_combinations(["option1", "option2"])	Same	passed
TC-09	Test generate expected values with no options	Unit Test	Call generate_expected_values([])	Same	passed
TC-10	Test generate expected values with one option	Unit Test	Call generate_expected_values(combinations, options), with both valid and one option	Same	passed
TC-11	Test generate expected value with multiple options	Unit Test	Call generate_expected_values(combinations, options), with both valid and multiple options	Same	passed
TC-12	Test is valid test case with valid cases	Unit Test	Call is_valid_tc(combinations, options) with both being valid, client is not trying to change options' values	Same	passed
TC-13	Test is valid test case with Invalid cases	Unit Test	Call is_valid_tc(combinations, options) with both being valid, but client try to configure the options	Same	passed
TC-14	Test CSV output	Unit Test	Call csv_output(combinations), and compare the result with expected csv file	Same	passed
TC-15	Test CSV output with empty data	Unit Test	Call csv_output([])	Same	passed
TC-16	Test valid inputs for validate_inputs() , and generate_combinations() methods	Integration Test	Setup valid options, validate on options, then generate combinations and compare the results	Same	passed

TC-17	Test valid inputs for validate_inputs() , generate_combinations(), and generate_expected_values() methods	Integration Test	Setup valid options, validate on options, then generate combinations, and finally generate the expected values and compare the results	Same	Passed
TC-18	Test valid inputs for validate_inputs() , generate_combinations(), and generate_expected_values(), is_valid_tc() methods	Integration Test	Setup valid options, validate on options, then generate combinations, and then generate the expected values, finally check is_valid_tc()	Same	
TC-18	Test valid inputs and invalid inputs with the whole system	System Test	Setup valid and invalid options, and test the whole system functionality	Same	passed

Summary of Unit Test Cases

- **Total Number of Unit Test Cases:** 15
- **Number of Passed Test Cases:** 15
- **Number of Failed Test Cases:** 0

Summary of Integration Test Cases

- **Total Number of Integration Test Cases:** 3
- **Number of Passed Test Cases:** 3
- **Number of Failed Test Cases:** 0

Summary of System Test Cases (main.py)

- **Total Number of System Test Cases:** 1
- **Number of Passed Test Cases:** 1
- **Number of Failed Test Cases:** 0

Appendix

- **Test Environment:** Ubuntu 20.04
- **Test Data:** Differs from test to another, but in general List of all combinations of test cases
- **Tools:** Pytest