

- DMS 合规性测试工具
 - 1. 运行测试
 - 2. 解读测试报告 (test_report.json)

DMS 合规性测试工具

1. 运行测试

本工具用于测试您的 API 是否符合 DMS 的相关规范。

前提条件:

- 确保获取本工具。
- 准备好您的 API 定义文件 (例如 OpenAPI 规范的 JSON 文件, 如 [井筒API示例_simple.json](#))。
- 准备好包含自定义的测试用例的目录。

运行命令:

打开您的终端或命令行工具, 使用以下命令格式运行测试:

```
ddms_compliance_tool --base-url <您的API基础URL> --yapi <您的API定义文件路径> [--custom-test-cases-dir <您的自定义测试用例目录>] [--verbose] [--output <报告输出路径>]
```

参数说明:

- **--base-url <您的API基础URL>**: 必填项。指定被测 API 服务的基础 URL。例如: <http://127.0.0.1:8080> 或 <https://api.example.com>。
- **--yapi <您的API定义文件路径>**: 必填项。指定描述 API 接口的 JSON 文件路径。此文件应符合 OpenAPI 规范。例如: [assets/doc/井筒API示例_simple.json](#)。
- **--custom-test-cases-dir <您的自定义测试用例目录>**: 必填项。如果您编写了自定义的测试用例 Python 文件, 请通过此参数指定它们所在的目录。例如: [./custom_testcases](#)。
- **--verbose**: 可选项。添加此参数会输出更详细的执行日志, 方便调试。

- **--output <报告输出路径>**: 可选项。指定测试报告 JSON 文件的输出路径和文件名。默认为 `test_report.json`。

示例:

假设您的 API 服务运行在 `http://127.0.0.1:4523/m1/6389742-6086420-default`, API 定义文件是位于 `assets/doc/` 目录下的 `井筒API示例_simple.json`, 并且您在当前目录下的 `custom_testcases` 文件夹中存放了自定义测试用例, 希望生成名为 `my_test_report.json` 的报告并查看详细日志, 那么运行命令如下:

```
ddms_compliance_tool --base-url http://127.0.0.1:4523/m1/6389742-6086420-default --yapi assets/doc/井筒API示例_simple.json --custom-test-cases-dir ./custom_testcases --verbose --output my_test_report.json
```

执行完毕后, 您将在指定的输出路径找到 JSON 格式的测试报告。

2. 解读测试报告 (`test_report.json`)

测试报告是一个 JSON 文件, 包含了详细的测试结果。以下是主要部分的解读:

- **summary_metadata**: 测试的元数据。
 - `start_time`: 测试开始时间。
 - `end_time`: 测试结束时间。
 - `duration_seconds`: 测试总耗时 (秒)。
- **endpoint_stats**: API 端点 (Endpoint) 的统计信息。
 - `total_defined`: API 定义文件中总共定义的端点数量。
 - `total_tested`: 本次测试实际执行的端点数量。
 - `passed`: 完全通过所有测试用例的端点数量。
 - `failed`: 未能通过至少一个测试用例的端点数量。
 - `partial_success`: (根据您的实际报告结构, 此字段可能表示部分用例通过, 但整体端点有失败, 或者有特定含义)
 - `error`: 在测试过程中发生错误的端点数量 (例如无法连接服务)。
 - `skipped`: 被跳过的端点数量。
 - `success_rate_percentage`: 端点测试的成功率。
- **test_case_stats**: 测试用例 (Test Case) 的统计信息。

- **total_applicable**: 在所有被测端点中, 总共适用的测试用例数量。
- **total_executed**: 实际执行的测试用例数量。
- **passed**: 通过的测试用例数量。
- **failed**: 失败的测试用例数量。
- **error_in_execution**: 执行过程中发生错误的测试用例数量。
- **skipped_during_endpoint_execution**: 在端点执行期间被跳过的测试用例数量。
- **success_rate_percentage**: 测试用例的成功率。

- **detailed_results**: 一个数组, 包含了每个被测端点的详细测试结果。
 - **endpoint_id**: 端点的唯一标识, 通常是 HTTP方法 + 路径, 例如 POST /api/dms/{dms_instance_code}/v1/message/push/{schema}/{version}。
 - **endpoint_name**: 端点的描述性名称 (如果 API 定义中提供)。
 - **overall_status**: 该端点的总体测试状态 (例如: "通过", "失败")。
 - **duration_seconds**: 测试该端点所花费的时间 (秒)。
 - **start_time**: 该端点测试开始时间。
 - **end_time**: 该端点测试结束时间。
 - **executed_test_cases**: 一个数组, 包含在该端点上执行的所有测试用例的详情。
 - **test_case_id**: 测试用例的唯一标识符 (例如 "TC-STATUS-001")。
 - **test_case_name**: 测试用例的描述性名称 (例如 "基本状态码 200 检查")。
 - **test_case_severity**: 测试用例的严重程度 (例如 "严重")。
 - **status**: 该测试用例的执行结果 ("通过", "失败")。
 - **message**: 关于测试结果的简要信息。
 - **duration_seconds**: 执行该测试用例所花费的时间。
 - **timestamp**: 测试用例执行完成的时间戳。
 - **validation_points**: (可选) 一个数组, 包含更细致的验证点结果。
 - **passed**: (true/false) 该验证点是否通过。
 - **message**: 该验证点的具体信息。