

跳过

查看详情

Toggle

TesterHome

HttpRunner HttpRunner 的结果校验器优化

- 社區
- Bug 曝光台
- 問答
- 社團
- 招聘
- Wiki
- 开源项目
- 酷站
- ITF榜单
- 

Toggle

- liuzhipeng17
 -
 - 個人資料設置
 - 我的收藏
 - 記事本
 -
 - 退出
- 搜索本站内容
- 295
 - 新建帖子
 -
 - 創建公司 / 團體

欢迎

HttpRunner HttpRunner 的结果校验器优化

debugtalk · 2017年12月13日 · 最後由 吴勇胜 (Vincent) 於 2019年08月26日回覆 · 7607 次閱讀

目录

- 设计之初
- 存在的局限性
- 改造结果
- 新增支持自定义结果校验器
- 结果校验器中实现变量引用
- 结果校验内容新增支持正则表达式提取
- 结果校验器的进一步简化

在测试用例中，包含预期结果这么一项，用于辅助测试人员执行测试用例时判断系统的功能是否正常。而在自动化测试中，我们的目标是让测试用例自动执行，因此自动化测试用例中同样需要包含预期结果一项，只不过系统响应结果不再由人工来进行判断，而是交由测试工具或框架来实现。

这部分功能对应的就是测试结果校验器（validator），基本上能称得上自动化测试工具或框架的都包含该功能特性。

设计之初

HttpRunner在设计之初，结果校验器（validator）的实现比较简单。

对于每一个test，可以指定0个或多个校验项，放置在validate中。在自动化测试执行的时候，会在发起HTTP请求、解析结果响应之后，逐个检查各个校验项，若存在任意校验项不通过的情况，则该test将终止并被标记为失败。

```
- test:
  name: get token
  request:
    url: http://127.0.0.1:5000/api/get-token
    method: GET
  extract:
    - token: content.token
  validate:
    - {"check": "status_code", "comparator": "eq", "expect": 200}
    - {"check": "content.token", "comparator": "len_eq", "expect": 16}
```

如上例所示，每一个校验项均为一个json结构，里面包含check、expect、comparator三个属性字段。其中，check对应着要检查的字段，expect对应着检查字段预期的值，这两项是必须指定的；comparator字段对应着比较方法，若不指定，则默认采用eq，即检查字段与预期值相等。

为了实现尽可能强大的检查功能，check属性值可通过链式操作精确指定具体的字段，comparator也内置实现了大量的检查功能。

举个例子可能会更清晰些。假如某结构的响应结果如下：

```
// status code: 200

// response headers
{
  "Content-Type": "application/json"
}

// response body content
{
  "success": False,
  "person": {
    "name": {
      "first_name": "Leo",
      "last_name": "Lee",
    },
    "age": 29,
    "cities": ["Guangzhou", "Shenzhen"]
  }
}
```

那么假如我们要检查status code，check就可以指定为status_code；假如要检查response headers中的Content-Type，check就可以指定为headers.content-type；假如要检查response body中的first_name，check就可以指定为content.person.name.first_name。可以看出，假如下一层级为字典结构，那么就可以通过 运算符指定下一层级的key，依次类推。

对于字段内容为列表list的情况略有不同，我们需要通过序号来指定具体检查哪一项内容。例如，Guangzhou对应的检查项为content.person.cities.0，Shenzhen对应的检查项为content.person.cities.1。

在比较方式（comparator）方面，HttpRunner除了eq，还内置了大量的检查方法。例如，我们可以通过gt、ge、lt、le等比较数值大小，通过len_eq、len_gt、len_lt等比较长度是否相等（列表、字典、字符串均适用），通过contains、contained_by来判断包含关系，通过startswith、endswith判断字符串的开头结尾，甚至通过regex_match来判断是否满足正则匹配等。详细的比较方式还有许多，需要时可查看[comparator](#)表格。

存在的局限性

在大多数情况下，HttpRunner的结果校验器（validator）是够用的。不过问题在于，框架不可能为用户实现所有的检查方法，假如用户需要某些特殊的检查方法时，HttpRunner就没法实现了。

这的确是一个问题，之前Junho2010提的issue [#29](#)中举了一个例子，应该也算比较有代表性。

发送请求时的数据使用了随机生成，然后需要比较结果中的数据是否是和这个相关（通过某个算法转换）。比如我输入的是321，我的结果是 $(3+2+1) * \text{avg}(3+2+1)$ 这种转化，目前的comparator是比较难于实现的。

要解决这个问题，最好的方式应该是在HttpRunner中实现自定义结果校验器的机制；用户在有需要的时候，可以自己编写校验函数，然后在validate中引用校验函数。之前也介绍过HttpRunner的热加载机制，[《约定大于配置：ApiTestEngine实现热加载机制》](#)，自定义结果校验器应该也是可以采用这种方式来实现的。

第二个需要优化的点，HttpRunner的结果校验器还不支持变量引用，会造成某些场景下的局限性。例如，testwangchao曾提过一个issue [#52](#)：

接口response内，会返回数据库内的自增ID。ID校验的时候，希望expected为参数化的值。

```
validate:
  - {"check": "content.data.table_list.0.id", "expected": "$id"}
```

另外，在[《ApiTestEngine，不再局限于API的测试》](#)一文中有介绍过，结果提取器（extract）新增实现了通过正则表达式对任意文本响应内容的字段提取。考虑到结果校验器（validate）也需要先从结果响应中提取出特定字段才能与预期值进行比较，在具体实现上完全可以复用同一部分代码，因此在validate的check部分也可以进行统一化处理。

经过前面的局限性问题描述，我们的改造目标也明确了，主要有三个方面：

- 新增支持自定义结果校验器
- 结果校验器中实现变量引用
- 结果校验内容新增支持正则表达式提取

改造结果

具体的改造过程就不写了，有兴趣的同学可以直接阅读源码，重点查看[httprunner/context.py](#)中的parse_validator、do_validation和validate三个函数。

经过优化后，改造目标中的三项功能都实现了。为了更好地展现改造后的结果校验器，此处将结合实例进行演示。

新增支持自定义结果校验器

先来看第一个优化项，新增支持自定义结果校验器。

假设我们需要使用HTTP响应状态码各个数字的和来进行校验，例如，201状态码对应的数字和为3，503状态码对应的数字和为8。该实例只是为了演示用，实际上并不会用到这样的校验方式。

首先，该种校验方式在HttpRunner中并没有内置，因此需要我们自己来实现。实现方式与热加载机制相同，只需要将自定义的校验函数放置到当前YAML/JSON文件同级或者父级目录的debugtalk.py中。

对于自定义的校验函数，需要遵循三个规则：

- 自定义校验函数需放置到debugtalk.py中
- 参数有两个：第一个为原始数据，第二个为原始数据经过运算后得到的预期结果值

- 在校验函数中通过`assert`将实际运算结果与预期结果值进行比较

对于前面提到的演示案例，我们就可以在`debugtalk.py`中编写如下校验函数。

```
def sum_status_code(status_code, expect_sum):
    """ sum status code digits
        e.g. 400 => 4, 201 => 3
    """
    sum_value = 0
    for digit in str(status_code):
        sum_value += int(digit)

    assert sum_value == expect_sum
```

然后，在YAML/JSON格式测试用例的`validate`中，我们就可以将校验函数名称`sum_status_code`作为`comparator`进行使用了。

```
- test:
  name: get token
  request:
    url: http://127.0.0.1:5000/api/get-token
    method: GET
  validate:
    - {"check": "status_code", "comparator": "eq", "expect": 200}
    - {"check": "status_code", "comparator": "sum_status_code", "expect": 2}
```

由此可见，自定义的校验函数`sum_status_code`与HttpRunner内置的校验方法`eq`在使用方式上完全相同，应该没有理解上的难度。

结果校验器中实现变量引用

对于第二个优化项，结果校验器中实现变量引用。在使用方式上我们应该与`request`中的变量引用一致，即通过`$var`的方式来引用变量`var`。

```
- test:
  name: get token
  request:
    url: http://127.0.0.1:5000/api/get-token
    method: GET
  variables:
    - expect_status_code: 200
    - token_len: 16
  extract:
    - token: content.token
  validate:
    - {"check": "status_code", "comparator": "eq", "expect": "$expect_status_code"}
    - {"check": "content.token", "comparator": "len_eq", "expect": "$token_len"}
    - {"check": "$token", "comparator": "len_eq", "expect": "$token_len"}
```

通过以上示例可以看出，在结果校验器`validate`中，`check`和`expect`均可实现变量的引用；而引用的变量，可以来自四种类型：

- 当前`test`中定义的`variables`，例如`expect_status_code`
- 当前`test`中提取（`extract`）的结果变量，例如`token`
- 当前测试用例集`testset`中，先前`test`中提取（`extract`）的结果变量
- 当前测试用例集`testset`中，全局配置`config`中定义的变量

而`check`字段除了可以引用变量，以及保留了之前的链式操作定位字段（例如上例中的`content.token`）外，还新增了采用正则表达式提取内容的方式，也就是第三个优化项。

结果校验内容新增支持正则表达式提取

假设如下接口的响应结果内容为`LB123abcRB789`，那么要提取出`abc`部分进行校验，就可以采用如下描述方式。

```
- test:
  name: get token
  request:
    url: http://127.0.0.1:5000/api/get-token
    method: GET
  validate:
    - {"check": "LB123(.*)RB789", "comparator": "eq", "expect": "abc"}
```

可见在使用方式上与在结果提取器（`extract`）中完全相同。

结果校验器的进一步简化

最后，为了进一步简化结果校验的描述，我在`validate`中新增实现了一种描述方式。

简化后的描述方式与原始方式对比如下：

```
validate:
  - comparator_name: [check_item, expect_value]
  - {"check": check_item, "comparator": comparator_name, "expect": expect_value}
```

同样是前面的例子，采用新的描述方式后会更加简洁。而两种方式表达的含义是完全等价的。

```
- test:
  name: get token
  request:
    url: http://127.0.0.1:5000/api/get-token
    method: GET
  validate:
    - eq: ["status_code", $expect_status_code]
    - sum_status_code: ["status_code", 2]
    - len_eq: ["$token", $token_len]
    - len_eq: ["content.token", 16]
    - eq: ["LB123(.*)RB789", "abc"]
```

当然，此次优化保证了与历史版本的兼容，之前编写的测试用例脚本的运行是完全不会受到任何影响的。

「原创声明：保留所有权利，禁止转载」

[10 个赞](#) [关注](#) [收藏](#) [举报](#)

TesterHome 为用户提供「保留所有权利，禁止转载」的选项。除非获得原作者的单独授权，任何第三方不得转载标注了「原创声明：保留所有权利，禁止转载」的内容，否则均视为侵权。具体请参见[TesterHome 知识产权保护协议](#)。

如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

[打赏支持](#)

共收到 21 条回复 [时间](#) [点赞](#)



[尹全旺](#) · #1 · [2017年12月14日](#)

仅楼主可见



[debugtalk](#) #2 · [2017年12月14日](#) 作者



对 [尹全旺](#) #1 回复

测试报告会采用allure重新实现下，现在的测试报告太简陋了



[yyy](#) #3 · [2017年12月14日](#)

[@debugtalk](#) 楼主,现在是只支持格式为json的请求报文吗?



[debugtalk](#) #4 · [2017年12月14日](#) 作者



对 [yyy](#) #3 回复

不限，只要是HTTP协议的都支持，翻下我之前写的文章吧

5楼 已删除

6楼 已删除



[ChelseaZhang](#) #7 · [2018年03月12日](#)

comparator表格的链接无法打开了



[debugtalk](#) #8 · [2018年03月12日](#) 作者



对 [ChelseaZhang](#) #7 回复

更新了



[haha](#) #9 · [2018年04月11日](#)

我有一个疑问，数据调用已经发生变化，怎么能重复使用进行回归测试呢



我信你个鬼，你这个老
头坏的很

[梦梦GO](#) #10 · [2018年04月25日](#)

今天在检测测试用例时候，突然想到个函数调用的问题。记得之前看到函数调用的方式是`${fun_name}`，去函数调用那章复看理我一下的确如此，为什么结果检验这块就直接是

- `fun_name`: `xx xx` 了，怕自己疏忽写错又回来翻了一遍，果然结果校验优化器这块可以这么写。



[星](#) 在 [HttpRunner 的测试用例分层机制 \(已过期\)](#) 中提及了此贴 05月31日 18:21



[yjg](#) #12 · [2018年06月01日](#)

[@debugtalk](#) 大大，在validate部分，不能使用`$response`作为参数么？想自己实现函数做json schema校验，但是不知道怎么把response传到函数中。



[debugtalk](#) #13 · [2018年06月02日](#) 作者

对  [yjq #12](#) 回复

支持的，直接使用 `$response` 就可以引用了。
<http://cn.httprunner.org/advanced/request-hook/>



[meiyao](#) #14 · [2018年07月26日](#)

对于字段内容为列表 `list` 的情况略有不同，我们需要通过序号来指定具体检查哪一项内容。例如，`Guangzhou` 对应的检查项为 `content.person.cities.0`，`Shenzhen` 对应的检查项为 `content.person.cities.1`。

testerhome.com

你好，对于列表，可以倒序引用吗？类似`alist=[1,2,3]`，`alist[-1]`输出3



[debugtalk](#) #15 · [2018年07月26日](#) 作者

对  [meiyao #14](#) 回复

没有支持哈

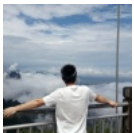


[meiyao](#) #16 · [2018年07月26日](#)

对  [debugtalk #15](#) 回复

我尝试了一下，支持了呀。

```
validate:
  - eq: ["content.result.success", True]
  - eq: ["content.data.-1.operatorName", $name]
```



[Sutune](#) #17 · [2018年07月27日](#)

对  [meiyao #16](#) 回复

谢谢提醒，我这边尝试了一下，也是ok的！



[debugtalk](#) #18 · [2018年07月28日](#) 作者

[@mickey2017](#) [@shuqing2017](#) 尴尬了，的确是支持的



[Jesmine](#) #19 · [2018年08月14日](#) 1 个赞

[@debugtalk](#) 作者大大，如果校验一个接口返回结果里不存在某个字段，比如接口返回是{
"photold": "3xfv72ni24j4nzm",
"count": 20
} 想判断一下这个返回里是不存在living这个字段的，可以直接写- eq:["content.living",null] 这么写吗？感谢感谢 ~



[我叫不迷糊](#) #20 · [2018年09月21日](#)



大神，弱弱问下，校验的内容是中文有编码问题，会报错 “Traceback (most recent call last):
UnicodeEncodeError: 'ascii' codec can't encode characters in position 0-3: ordinal not in range(128)” ，这个要怎么解决呀？
Python 2.7.9
HttpRunner 1.5.13
Windows



[我的眉毛会跳舞](#) #21 · [2019年01月09日](#)

在结果校验中，有考虑支持输出一些信息么，例如 [check_item, expect_value, msg]



[张逸帆](#) #22 · [2019年02月21日](#)

请问一下，我要测的接口的content下有很多值，也不在同一个层级下，httprunner可以支持将content下的全部字段都转化成case里的结果校验吗？



chunya #23 · 2019年04月08日

如果我在校验器里加入函数（从数据库里面取值\${lift_check_attributes_goto_db('\$sn')}）怎么处理？下面的代码，函数lift_check_attributes_goto_db('\$sn')取不到MYSQL数据库的值

- test: name: 校验上传 api: api/vvvvv.yml variables:
 - sn: \${lift_get_radom_time_string()}
 - value: \${lift_check_attributes_goto_db('\$sn')} validate:
 - { "check": "status_code", "comparator": "eq", "expect": 200 }
 - { "check": "content.success", "comparator": "eq", "expect": 0 }
 - { "check": "\$value", "comparator": "eq", "expect": "\$sn" }

报错如下：

AssertionError: validate: \$value equals gongyoulift2019040817423338(str) ==> fail

\${lift_check_attributes_goto_db('gongyoulift2019040817423338')}(str) equals gongyoulift2019040817423338(str)



吴勇胜 (Vincent) #24 · 2019年08月26日

结果校验器中实现变量引用，validate 中的 expect 中 不能调用 debugtalk 中的方法，可以优化一下

Test Report:

Request and Response data

check	comparator	expect value	actual value
status_code	equals	200	200
content.code	equals	0	0
content.msg	equals	SUCCESS	SUCCESS
content.data.list	IN	LazyString(\${query_sql()})	

不能调用debugtalk中定义的方法

Statistics:

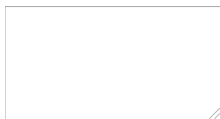
content_size(bytes)	1169
response_time(ms)	85.21
elapsed(ms)	81.054

回帖

- 请注意单词拼写，以及中英文排版，[参考此页](#)
- 支持 Markdown 格式, ****粗体****、~~删除线~~、`单行代码`, 更多语法请见这里 [Markdown 语法](#)
- 支持表情, 使用方法请点击☺
- 上传图片, 支持拖拽和剪切板黏贴上传, 格式限制 - jpg, png, gif
- 发布框支持本地存储功能, 会在内容变更时保存, 「提交」按钮点击时清空

- [Ruby](#)
- [HTML / ERB](#)
- [CSS / SCSS](#)
- [JavaScript](#)
- [YAML](#)
- [CoffeeScript](#)
- [Nginx / Redis \(.conf\)](#)
- [Python](#)
- [PHP](#)
- [Java](#)
- [Objective-c](#)
- [Erlang](#)
- [Shell / Bash](#)

- [編輯](#)
- [預覽](#)
-



提交回復 Command + Enter

[排版说明](#)

☐ 僅樓主可見

相关话题

- [【我是一本正经的广告】2020 MTSC-北京大会征稿啦!](#)
- [HttpRunner 的测试用例分层机制\(已过期\)](#)
- [HttpRunner 使用小结](#)
- [HttpRunner 实现参数化数据驱动机制](#)
- [HttpRunner 工具使用说明书](#)
- [探索 HttpRunner 最佳体现形式_设想篇](#)

作者



高級會員

debugtalk (debugtalk)

<https://testerhome.com/topics/11207>

第 6109 位會員 / 2015-11-24
ByteDance



TTF基金会



新秀群



霍格沃兹测试学院

微信公众号: DebugTalk

10 个赞

[关注](#) [收藏](#)

共收到 21 条回复

[打赏支持](#)

[有新回复!](#) [点击这里立即载入](#)



[关于](#) / [活跃用户](#) / [中国移动互联网测试技术大会](#) / [反馈](#) / [Github](#) / [API](#) / [帮助推广](#)

TesterHome 移动测试社区，由众多移动测试工作者维护，致力于推进国内测试技术。Inspired by RubyChina

友情链接 [WeTest](#) [腾讯质量开放平台](#) / [InfoQ](#) / [SegmentFault](#) / [测试窝](#) / [百度测试吧](#) / [IT大咖说](#)

[简体中文](#) / [正體中文](#) / [English](#)