Fibonacci试用记录

# # Part 1-问题

## ## 1.1-环境变量Key大小写问题

## ### 1.1.1 问题说明

环境变量Key的命名与实际约束不符。如下图，大些的环境变量无法设置



## --本周修复

## ## 1.2-环境变量命名问题

### ### 1.2.1 问题说明

没有对环境变量的命名进行约束和限制。

文档中（页10）提到：

*fib\_\*的环境变量是保留的，不能设置*

但实际可以设置了fib\_\*系列的环境变量。如下图，我设置了fib\_loc和fib\_wy7两个环境变量，并成功的在函数中被使用。

## --文档临时增加的规则，为了防止与用户环境变量冲突而设计，计划会增加功能校验



#### #### Code

|  |
| --- |
| import os import json  def handle(req):     """handle a request to the function     Args:         req (str): request body     """     print("FaaS-wy7test2")     print("Msg 1\n")     print("Msg 2")     val = 36     print("val: {}".format(val))     val += 20      print("val + 20 : {}".format(val))     print("Test Env")     env\_username = os.getenv("username", "default\_env\_username")     print("Env [username]:#{}#".format(env\_username))     env\_username = os.getenv("USERNAME", "default\_env\_username")     print("Env [USERNAME]:#{}#".format(env\_username))     print("New Version")     return\_msg\_dict = {         "return\_key\_1": "return\_val\_1",         "env\_username": env\_username,         "list": [1, 2, 3, 4],     }     env\_fib\_wy7 = os.getenv("fib\_wy7", "default\_env\_fib\_wy7")     print("Env [fib\_wy7]:#{}#".format(env\_fib\_wy7))     env\_fib\_loc = os.getenv("fib\_loc", "default\_env\_loc")     print("Env [fib\_loc]:#{}#".format(env\_fib\_loc))     print(json.dumps(return\_msg\_dict)) |

#### #### OutPut



## ## 1.3 代码包组织与dependencies配置

### ### 问题描述

文档页14提到支持代码包上传模式。但没有详细说明代码包的要求。我在没有dependencies的情况下，上传python3的代码包运行成功。

但是当我想用python的requests包时，一直没有成功。

代码包方面，我试了以下组合(wy7test4是我函数的名字，也是代码包的名字)

方式1 是使用faas-cli的格式

方式2 是AWS lambda的代码包组织形式（即通过pip install *module-name* -t */path/to*/project-dir 将要用的module下载好。我使用 pip install requests -t . 获得如下结构）

|  |
| --- |
| 方式1:  wy7test4 ├── handler.py └── requirements.txt  方式2:  wy7test4  ├── bin  │   └── chardetect  ├── certifi  │   |  |   ...  ├── certifi-2018.4.16.dist-info  |   ...  ├── chardet  |   ...  ├── chardet-3.0.4.dist-info  |   ...  ├── handler.py  ├── idna  |   ...  ├── idna-2.7.dist-info  |   ...  ├── requests  |   ...  ├── requests-2.19.1.dist-info  |   ...  ├── requirements.txt  ├── urllib3  |   ...  └── urllib3-1.23.dist-info     ... |

另外AWS Lambda 对zip方式也有要求（解压后直接是包内的文件，而不包含root\_directory），对此我也进行了尝试，依然没有成功。

因此，我想问下，对这种需要dependencies的代码包，应该如何组织？

## --方式1使用是正确的，我们的Dockfile有点问题，本周修复

# # Part 2-建议

这部分我按照文档顺序展开

## ## 2.1 函数名称

文档（页8）提到：

*函数名称（必填）：小写字母开头的4-10 位的字母与数字组合，在整个平台中唯一。*

由于不允许重复且限制较多，感觉起名有些困难，可否：

1. 放宽命名长度限制，
2. 增加特殊字符‘-’方（OpenFaaS允许‘-’，而测试平台目前不允许，是因为冲突吗,还是别的考虑 ？）
3. 统一命名规范，如类似Jar包命名, jd-tig-fibtest-py3

## --与jdos平台对接，规则与jdos应用标识一致，暂不调整

## ## 2.2 函数模版

文档（页10） 谈到了函数模版，除了这些函数模版外，除了基本的函数模版外，能否提供一些典型的样例, 让不太熟悉FaaS的人快速了解:

* 如何管理代码包的结构
* 如何引入要用的包
* 如何调用Function以及如何进行FaaS间的各类调用

## --完善使用文档

## ## 2.3 发布与修改

* 当点击发布后会弹出提示框，产生一个版本号。但当用户修改保存，回到主界面后，却缺乏相应的提示。不知道目前运行的是不是修改后的版本。 对此，能否提示用户有修改后的版本待发布？以及能否增加查看当前运行版本按钮？
* 点击发布后提示“部署函数成功”，但试用过程中，实际仍在部署中，需等待一段时间后方可调用。

## --功能调整完善

## ## 2.4 调用函数

### ### 调用地址

如文档页16所示，网页端提供了三类HTTP调用地址

1. 能否对下图提供的三个HTTP调用地址： 增加一个 "Copy to clipboard" 按钮？
2. 所提供的三个地址都是Server Side的调用地址（gateway开头）。开发者无法在本地使用这些地调用调试所写function？能否增加供client调用的地址？
3. OpenFaaS异步调用在header内设置callback地址。但这里没有设置callback的相应提示

## --1、新需求；2、通过http是可以调用的；3、功能是有的，没有对外开放

### ### 调用参数

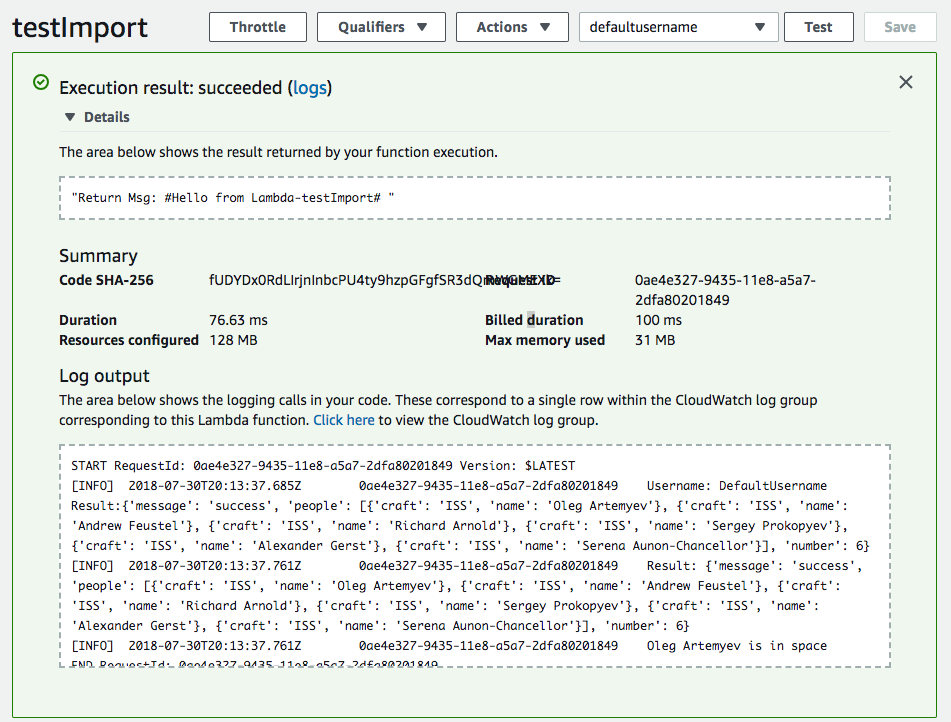
后期可以像lambda一样提供req模版和配置保存各类调用参数，便于用户测试

### ### 调用结果

感觉 “调用结果” 在这里比较有误导性。

我对“调用结果”的直观理解是我所写函数调用的返回值（即最后return语句返回的内容）。而目前测试平台点击“执行”按钮后显示的“调用结果”只包括print语句的值和logging的值，并不包括return 语句的值。

是否可以像Lambda一样将return结果和log/print分开？将log/print输出到日志中去？（如下图所示）最终将Execution Result 和Log Output分开。



## --"调用结果”文案改为“输出结果”， 显示return值目前没有提供