

DestroyerRobot 下 ApiTesting 全链路接口自动化测试框架

-2021 年 3 月 2 日

谢猛--vivid

简介:

此框架是基于 Python+Unittest+BeautifulReport+requests+Config+yaml+json+ddt 实现全链路接口自动化数据驱动测试。

主要流程: 解析接口数据包->生成接口基础配置 (config 项目主配置文件+yaml 子项目主配文件) ->生成测试用例 (yaml+json) ->生成测试脚本 (.py) ->运行测试场景 (unittest) ->生成测试报告 (BeautifulReport)。

测试流程: 初始化请求->除了接口基础信息->读取前置接口用例->发送前置接口->处理当前接口数据->发送当前接口->检查接口返回。

接口自动化测试几大块: 测试用例设计、测试脚本编写、测试结果校验、测试报告生成、测试配置管理。

依赖关系及处理难点: 接口之间相互依赖关系、测试数据与脚本分离、测试数据参数化处理。

而这些本框架通通已经解决, 无需编写冗余代码, 基本以 copy 和修改参数为准, 需要获取接口数据信息即可。

关于接口依赖: 只要填写前置接口对应路径即可, 如果存在数据依赖关系, 此时你可以填写前置接口对应的参数值, 本框架将自动为你调用和替换关联数据。

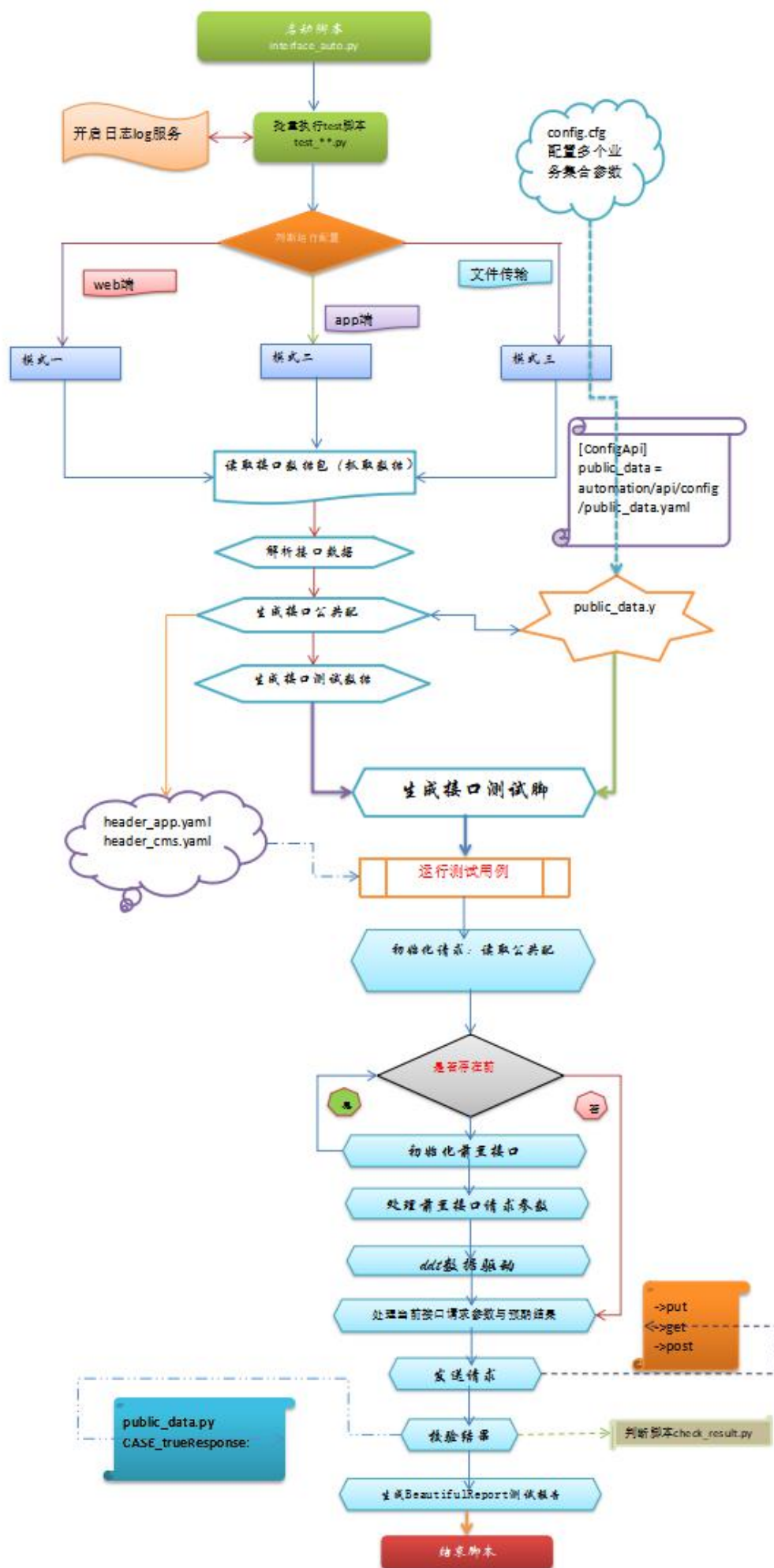
关于测试数据: 本框架采用 yaml 配置模板信息, excel 文件保存请求参数和执行结果数据, 将执行 json 信息写入到 excel 文件中, 解决各类数据错综复杂的问题。

关于参数化: 本框架提供了大量的 util 类来满足使用者对数据操作及采集。

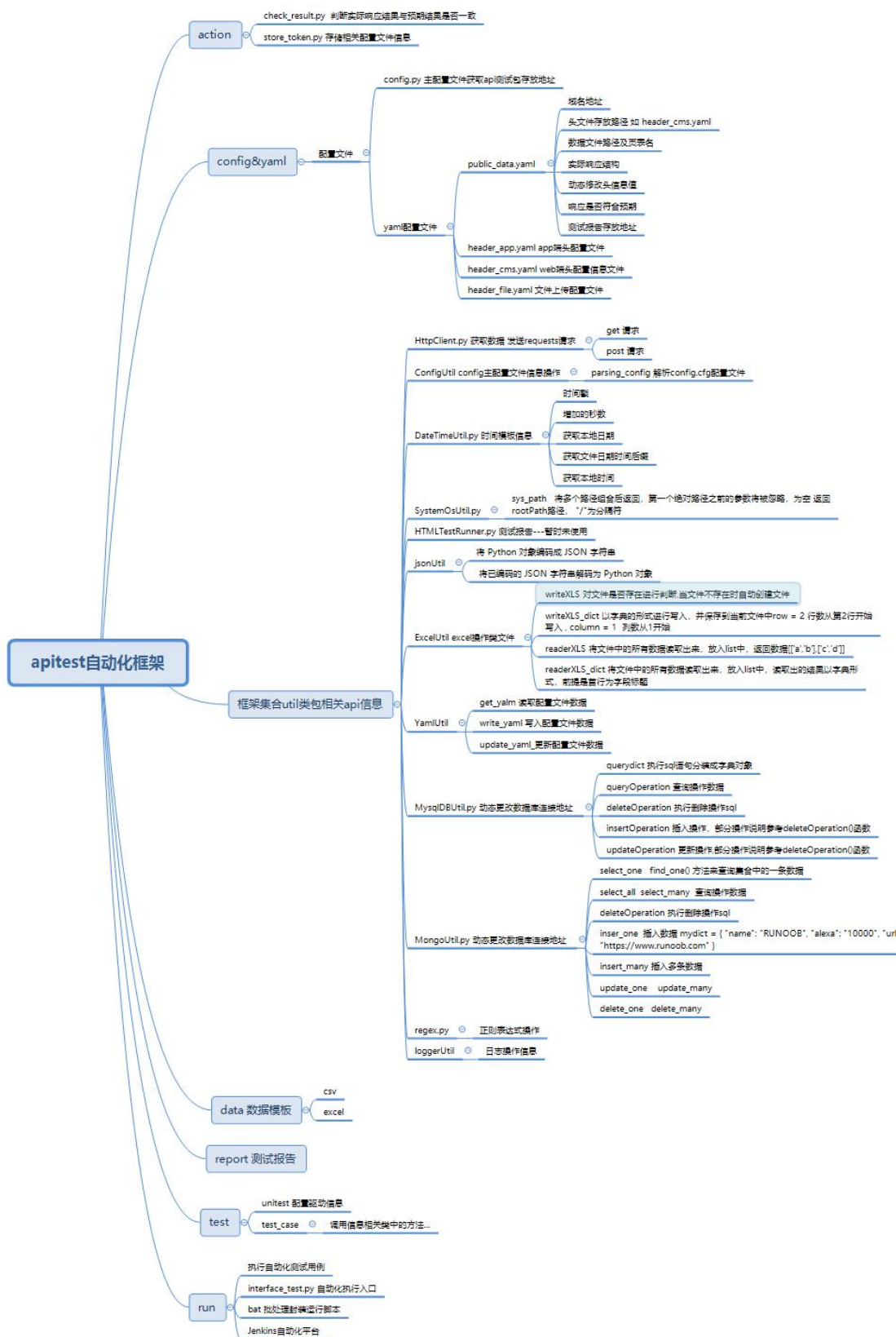
目前主要支持三种运行模型:

- >0、web 端接口请求
- >1、文件传输
- >2、app 端接口请求

框架流程图:



项目结构:



执行入口: run interface_test.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# @Time    : 2020/11/9 15:45
# @Author   : vMld
# @Email    : 331597811@163.com
# @File     : interface_auto.test.py

import unittest

from BeautifulReport import BeautifulReport

from DestroyerRobot.automation.util.LoggerUtil import Log

from DestroyerRobot.automation.util.ConfigUtil import Config

from DestroyerRobot.automation.util.SystemOsUtil import SystemOs

from DestroyerRobot.automation.util.YamlUtil import yamlUtil

from DestroyerRobot.automation.util.DateTImeUtil import TestDateTIme

class testFileReport:

    def apIs_yaml(self):

        conf = Config("ConfigApi")

        api_path = conf.get_configPath("public_data")

        api_yaml = SystemOs().sys_path(api_path)

        report = yamlUtil(api_yaml).get_yaml()

        report_path = SystemOs().sys_path(report['report'])

        test_path = SystemOs().sys_path(report['test_path'])

        # 0 报警地址 , 1 测试数据集地址

        return report_path, test_path
```

操作脚本: test test_cms.py

```
import unittest

import time

from ddt import ddt, data, unpack

from DestroyerRobot.automation.util.ConfigUtil import Config

from DestroyerRobot.automation.util.SystemOsUtil import SystemOs

from DestroyerRobot.automation.util.YamlUtil import yamlUtil

from DestroyerRobot.automation.util.ExcelUtil import xlsxoper

from DestroyerRobot.automation.util.LoggerUtil import Log

from DestroyerRobot.automation.util.HttpClient import HttpClient

from DestroyerRobot.automation.api.action.check_result import CheckResult

from DestroyerRobot.automation.api.action.store_token import store_token

@ddt

class TestCase(unittest.TestCase):

    """

    全局变量中将 excel 中首行标题提取: testdata2, yaml 数据返回结果: dataInfo

    新脚本中配置新的 excel 需要改写内容如下 : 对应新的配置文件信息

    global: 全局变量
```

```

file_path = SystemOs().sys_path(dataInfo['file_path2'])

testdata = xlsoper(file_path).readerXLS_dlist(dataInfo['SheetName2'])[0] # 数据用于 ddt 使用
testdata2 = xlsoper(file_path).readerXLS_dlist(dataInfo['SheetName2'])[1] # 首行标题

def:

执行脚本对应的 excel 写入 sheet

xls = xlsoper(file_path).writeXLS_dlist(datas['caseNo'],column,str(responseData),dataInfo['SheetName2'])

"""

global log , file_path,dataInfo,testdata2

log = Log().logger()

conf = Config("ConfigApi")

keys = conf.parsing_config("public_data")

public_data = SystemOs().sys_path(keys) #通过主配置文件 config 获取主 yamI 配置文件路径-

dataInfo = yamiUtil(public_data).get_yalm()

file_path = SystemOs().sys_path(dataInfo['file_path2'])

testdata = xlsoper(file_path).readerXLS_dlist(dataInfo['SheetName2'])[0] # 数据用于 ddt 使用
testdata2 = xlsoper(file_path).readerXLS_dlist(dataInfo['SheetName2'])[1] # 首行标题

#testdata,testdata2 = xlsoper(file_path).readerXLS_dlist(dataInfo['SheetName2']) #等于 testdata,testdata2

#*testdata 只获取数据

@data(*testdata)

def test_01_demo(self, datas):

    """

    接口测试

    :param datas: 从 yamI 文件中获取 excel 路径, 读取数据集, 通过 ddt 形式

    """

    isSkip = datas['isSkip'] #获取 excle 中 isSkip

    if isSkip == 'y':

        log.info("当 isSkip==y 跳过此用例")

    else:

        #print("datas:",datas)

        webapp = datas['setApp']

        setApp=dataInfo[webapp] #获取 excel 中设备类型 setApp 是网页还是手机

        log.info('setApp: %s ',setApp)

        setpath = SystemOs().sys_path(setApp) # 获取 yamI 子 配置文件 路径 信息 如 :

C:/Users/vivid/PycharmProjects/untitled/DestroyerRobot/yamI_file/header cms.yamI

        header = yamiUtil(setpath).get_yalm() #获取 header 信息

        uri = dataInfo[datas['domain']] +datas['requestUri']

        # "请求方式 = %s",datas['method'], "请求参数 = %s",datas['param'] "请求格式 = %s",datas['paramType'], "请求

头部信息 = %s",header

        log.info("请求地址 = %s ",uri)

```

```

        # log.info("请求方式= %s",datas['method'])

        # log.info("请求格式= %s",datas['paramType'])

        # log.info("请求头信息= %s",header)

        responseData = HttpClient().request(datas['method'],uri,datas['paramType'],datas['param'],header) # 实际
结果

        log.info("实际响应结果: %s",responseData)

        responseResult = CheckResult.check(responseData, datas["exceptResponse"])

        log.info("判断预期结果与实际结果是否一致: 0 为 False 1 为 True = %s ",responseResult)

        # 判断是否将 token 写入 yml 对应的文件中

start_token(responseData,webapp,setpath,responseResult,datas['istoken'],dataInfo['token'],dataInfo['Authorization']).to
ken()

        #查找 CASE_trueResponse 中 value 值对应 excel 的列位置
        column = testdata2.index(dataInfo['CASE_trueResponse'])

        # 写入 excel 中

        log.info("将实际响应结果集数据写入 excel 中对应列")

        #写入 excel 数据必须是字符串, 不能是 dict

        xls = xlsoper(file_path).writeXLS_dict(datas['caseNo'],column,str(responseData),dataInfo['SheetName2'])

        time.sleep(1)

if __name__ == '__main__':

    unittest.main()

```

判断脚本 action check_result.py store_token.py

```

class CheckResult():

    """

    responseData:判断实际响应结果与预期结果是否一致

    exceptResponse :

    """

    @classmethod

    def check(self, responseData, exceptResponse):

        If isinstance(responseData ,str):

            responseData = eval(responseData)

        If isinstance(exceptResponse , str):

            exceptResponse = eval(exceptResponse)

        for key, value in exceptResponse.items():

            If key in responseData:

                If responseData[key] == value:

                    responseResult = 1

            else:

                responseResult = 0

```

```
        return responseResult

    else:

        responseResult = 0

        return responseResult

    return responseResult
```

```
class start_token:
```

```
    """
```

```
    start_token 只是为了满足权限系统而作
```

```
    """
```

```
    def __init__(self,responseData,setApp,setpath,responseResult,lstoken,token_value,Authorization):
```

```
        self.responseData = responseData
```

```
        self.lstoken = lstoken
```

```
        self.setpath = setpath
```

```
        self.setApp = setApp
```

```
        self.responseResult = responseResult
```

```
        self.tokenvalue =token_value
```

```
        self.Authorization = Authorization
```

```
        self.log = Log().logger()
```

```
    def token(self):
```

```
        #print(self.responseData,"\n\t",self.lstoken                ,"\n\t",self.setpath                ,"\n\t",
```

```
self.setApp ,"\n\t",self.responseResult ,"\n\t",self.tokenvalue ,"\n\t",self.Authorization )
```

```
    If self.lstoken :
```

```
        If self.responseResult :
```

```
            If "data" in self.responseData.keys():
```

```
                If self.responseData["data"] == None:
```

```
                    self.log.info("data 属于 null")
```

```
                else:
```

```
                    If self.tokenvalue  in self.responseData['data'].keys():
```

```
                        reToken = self.responseData["data"][self.tokenvalue]
```

```
                        #在权限系统中生成"Bearer "+applicationToken 对应 Authorization
```

```
                        reAuthorization = "Bearer "+self.responseData["data"]["applicationToken"]
```

```
                        self.log.info("reToken:%s" % reToken)
```

```
                    If self.setApp == 1 :
```

```
                        data = yamUtil(self.setpath).get_yalm()
```

```
                        If self.tokenvalue in data.keys():
```

```
                            yamUtil(self.setpath).update_yam(self.tokenvalue,reToken)
```

```
                        else:
```

```
                            Token = {self.tokenvalue: reToken}
```

```
                            yamUtil(self.setpath).write_yam(Token)
```

```
                    If self.Authorization in data.keys():
```

```
                        yamUtil(self.setpath).update_yam(self.Authorization,reAuthorization)
```

```
                    else:
```

```

        Authorization = {self.Authorization: reAuthorization}
        yamlUtil(self.setpath).write_yaml(Authorization)

    elif self.setApp == 2 :
        data = yamlUtil(self.setpath).get_yaml()
        if self.tokenvalue in data.keys():
            yamlUtil(self.setpath).update_yaml(self.tokenvalue,reToken)
        else:
            reToken = {self.tokenvalue: reToken}
            yamlUtil(self.setpath).write_yaml(reToken)

    else:
        self.log.info("登录没有返回 token")

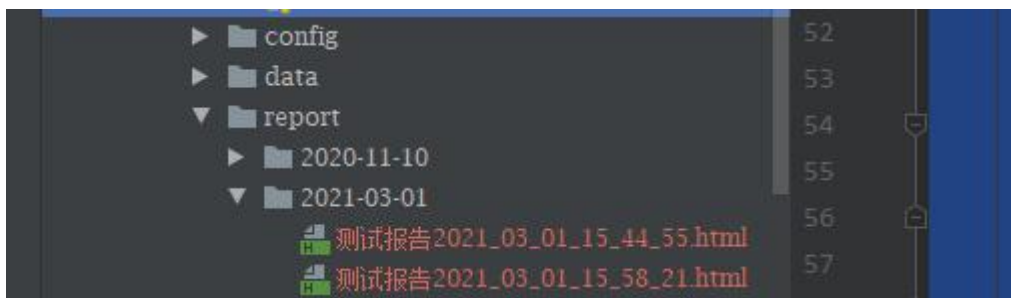
    else:
        self.log.info("response 没有 data 层")

    else:
        self.log.info("登录失败")

    else:
        self.log.info("不是登录")

```

测试报告实例：按照日期生成对应测试报告



测试报告



执行 excel 数据结果：异常测试以报告为准

caseNo	project	api	setApp**method	istoken	domain	requestUrl	param	iamTyp	iSskip	*exceptRespttrueResponse
							{ 'mobile': '18716759404', 'appFlag': 'easylife-cms-api-gateway', 'afsSessionId': 'WjFckIWDpHT9odN', 'afsSig': 'QUanqgQrMvXNN', 'afToken': 'FFFFF000000000009 19415916866073830 5844042761792562', 'afsScene': 'nc_login', 'password': '123456' }	json	n	{ 'code': 200, 'msg': '成功' }
1	new_cms	登录成功	1 post	1 uatPmsSo	/api/login		{ 'code': 200, 'msg': '成功' }	n		{ 'code': 200, 'msg': '成功', 'data': { 'token': 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50b3RhdGEiLCJ0eXciOiJ1cm9vdG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LCJ1b2N0b3R5IjoiYm9keSIsImVudCI6Im90dG9zaXQyIiwiaWF0IjoxNjE1OTQ0MjE5Lm50LC