

目录

Python教程

Python简介

安装Python

第一个Python程序

Python基础

函数

高级特性

函数式编程

模块

面向对象编程

面向对象高级编程

错误、调试和测试

IO编程

进程和线程

正则表达式

常用内建模块

常用第三方模块

virtualenv

图形界面

网络编程

电子邮件

访问数据库

Web开发

异步IO

实战

Day 1 - 搭建开发环境

Day 2 - 编写Web App骨架

Day 3 - 编写ORM

Day 4 - 编写Model

Day 5 - 编写Web框架

Day 6 - 编写配置文件

Day 7 - 编写MVC

Day 8 - 构建前端

Day 9 - 编写API

Day 10 - 用户注册和登录

Day 11 - 编写日志创建页

Day 12 - 编写日志列表页

Day 13 - 提升开发效率

Day 14 - 完成Web App

Day 15 - 部署Web App

Day 16 - 编写移动App

FAQ

期末总结

关于作者



廖雪峰

北京 朝阳区

加关注

廖雪峰

自己的Python课程

Python商业爬虫全解密

让天下没有爬不到的数据!

Python爬虫 + 数据分析

Python机器学习 + 深度学习

找廖雪峰老师

廖雪峰老师

自己的Java课程

Java高级架构师

更专业 更权威

源码分析专题 + 微服务架构专题

高并发分布式专题 + 性能优化专题

找廖雪峰老师

python免费公开课

编程学习网

授课模式：在线直播+课后视频，从零基础到中级开发工程师

查看详情

python免费公开课

编程学习网

授课模式：在线直播+课后视频，从零基础到中级开发工程师

查看详情

Day 6 - 编写配置文件

阅读 33411

有了Web框架和ORM框架，我们就可以开始装配App了。

通常，一个Web App在运行时都需要读取配置文件，比如数据库的用户名、口令等，在不同的环境中运行时，Web App可以通过读取不同的配置文件来获得正确的配置。

由于Python本身语法简单，完全可以直接用Python源代码来实现配置，而不再需要解析一个单独的 .properties 或者 .yaml 等配置文件。

默认的配置文件的应该完全符合本地开发环境，这样，无需任何设置，就可以立刻启动服务器。

我们把默认的配置文件的命名为 `config_default.py`：

```
# config_default.py

configs = {
    'db': {
        'host': '127.0.0.1',
        'port': 3306,
        'user': 'www-data',
        'password': 'www-data',
        'database': 'awesome'
    },
    'session': {
        'secret': 'Aa50aE'
    }
}
```

上述配置文件简单明了。但是，如果要部署到服务器时，通常需要修改数据库的host等信息，直接修改 `config_default.py` 不是一个好办法，更好的方法是编写一个 `config_override.py`，用来覆盖某些默认设置：

```
# config_override.py

configs = {
    'db': {
        'host': '192.168.0.100'
    }
}
```

把 `config_default.py` 作为开发环境的标准配置，把 `config_override.py` 作为生产环境的标准配置，我们就可以既方便地在本地开发，又可以随时把应用部署到服务器上。

应用程序读取配置文件需要优先从 `config_override.py` 读取。为了简化读取配置文件，可以把所有配置读取到统一的 `config.py` 中：

```
# config.py
configs = config_default.configs

try:
    import config_override
    configs = merge(configs, config_override.configs)
except ImportError:
    pass
```

这样，我们就完成了App的配置。

参考源码

day-06

读后有收获可以请作者喝咖啡，读后有疑问请加群讨论：

还可以分享给朋友：

分享到微博

廖雪峰

官方 独家

Python

商业爬虫全解密

找廖雪峰老师

ACM金牌得主

全球顶尖名企一线数据科学家倾力指导

人工智能与自然语言/计算机视觉课程培训

Artificial Intelligence For NLP/CV Courses

无offer退全款

廖雪峰推荐

JAVA进阶教程

原价1599元

0元领取

爆款云产品拼购2折起

1核云主机低至199元/年，降低上云门槛

立即查看

Python全栈实战课程限时免费领取

120天腾讯课堂老师带你从零基础到项目实战，全系统学习爬虫、数据分析、Web开发等Python开发技术

Python全栈实战课程限时免费领取

120天腾讯课堂老师带你从零基础到项目实战，全系统学习爬虫、数据分析、Web开发等Python开发技术

评论

if user.passwd != sha1.hexdigest()不明白这两个怎么能相等，打断点看了几次都不相等啊？？？

我蒙尘难红 created at November 9, 2018 11:11 AM, Last updated at November 9, 2018 11:11 AM

@post('/api/authenticate')

def authenticate(*, email, passwd): if not email: raise APIValueError('email', 'Invalid email.') if not passwd: raise APIValueError('passwd', 'Invalid password.') users = yield from User.findAll('email=?', [email]) if len(users) == 0: raise APIValueError('email', 'Email not exist.') user = users[0]

check passwd: sha1 = hashlib.sha1() sha1.update(user.id.encode('utf-8')) sha1.update(b':') sha1.update(passwd.encode('utf-8')) print(sha1.hexdigest()) # if user.passwd != sha1.hexdigest(): # raise APIValueError('passwd', 'Invalid password.') # authenticate ok, set cookie: r = web.Response() r.set_cookie(COOKIES_NAME, user2cookie(user, 86400), max_age=86400, httponly=True) user.passwd = 'aaaaa' r.content_type = 'application/json' r.body = json.dumps(user, ensure_ascii=False).encode('utf-8') return r

全部评论 0 回复

配置文件很理解很简单，但是看不出来这个配置文件在做什么作用啊

用户44xvaxq2 created at May 3, 2017 9:07 PM, Last updated at September 28, 2018 3:28 PM

用这段代码好像有问题

def toDict(d): D = Dict() for k, v in d.items(): D[k] = toDict(v) if isinstance(v, dict) else v return D

...

看这个意思是把d的每一个非dict的值付给D[k],但是测试并没有起到应有的作用。还是我理解错了？

手是冰冷的

Created at June 20, 2017 11:14 AM

class Dict(dict): """ 重写属性设置，获取方法 支持通过属性名访问键值的值，属性名将被当键名 """ def __init__(self, names=(), values=(), **kw): super(Dict, self).__init__(**kw) # zip函数将参数数拆分组合成[(arg1[0], arg2[0], arg3[0]...), (arg1[1], arg2[1], arg3[1]...),...] # 以参数中元素数量最少的集合长度为返回列表长度 for k, v in zip(names, values): self[k] = v def __getattr__(self, item): try: return self[item] except KeyError: raise AttributeError(f"Dict' object has no attribute '{k}' % item) def __setattr__(self, key, value): self[key] = value

Read More

黑味奶

Created at September 28, 2018 3:28 PM

for k, v in zip(names, values): self[k] = v

把这两行去了好像也没什么影响啊

全部评论 0 回复

打卡

黑味奶 created at September 28, 2018 3:25 PM, Last updated at September 28, 2018 3:25 PM

dug 打断点，多看几遍就明白了。。 Configuration.把default的host 替换成override的host，

toDict(d)最后返回的是：

{'db': {'host': '192.168.0.100'}}

上一节的还有点懵

全部评论 0 回复

搞不懂 merge 返回的就是一个字典，为什么还要定义一个 toDict()

深圳科迅智能科技有限公司 created at May 30, 2018 8:53 AM, Last updated at August 11, 2018 12:06 AM

python定义空字典的方法有2种，1. D=dict() 2. d={}

帝吧一罗书

Created at August 11, 2018 12:06 AM

Simple dict but support access as x.y style. 也就是把这个dic转化成自己定义的Dic类型好处是可以通过d.key 这种方法来取值

全部评论 0 回复

flasky项目配置文件只有一个config，与上面写法有什么优劣呢

goldct created at December 29, 2017 4:59 PM, Last updated at December 29, 2017 4:59 PM

我看到flasky里面的config.py是这样的: class Config: pass class DevelopmentConfig(Config): pass class ProductionConfig(Config): pass config = {'development':DevelopmentConfig, 'product':ProductionConfig }

在实际开发中，配置文件应该怎么写比较好呢？

全部评论 0 回复

没搞明白为啥要多加一个toDict函数处理

lingbo2008 created at October 10, 2016 2:22 PM, Last updated at January 3, 2017 8:10 AM

如题，没搞明白为啥要多加一个toDict函数处理，我打印了一下处理前后的变量，貌似是一模一样的呢

灰手

Created at October 10, 2016 5:22 PM

这个toDic的主要功能是添加一种取值方式 `a_dict.key`，相当于 `a_dict['key']`，这个功能不是必要的，我的项目放弃实现这一功能

Wizard_Zyklop

Created at January 3, 2017 8:10 AM

从函数名字上就很清楚了，就是把从经过merge函数处理写后的dic对象configs转变为文件中从dic基类派生出的Dic类对象，从而实现xxx.key的取值功能。

全部评论 0 回复

开始懂了...

我的数据博士 created at March 12, 2016 2:10 PM, Last updated at March 12, 2016 2:10 PM

good get this

全部评论 0 回复

发表评论

登录后发表评论

廖雪峰的官方网站©2019 Powered by iTranswarp 本网站运行在阿里云上并使用阿里云CDN加速。

意见反馈 使用许可

友情链接: 中华诗词 - 阿里云 - SICP - 4closure