flask 学习

2018-03-22 21:46:21

app.config 存储临时配置？

多应用需要使用应用上下文，那么我只是用多 blueprint 是应该用什么功能呢？

就算了解了请求上下文和应用上下文，也会有很多疑惑。

(1) 既然请求上下文和应用上下文生命周期都在线程内，其实他们的作用域基本一样，为什么还要两个级别的上下文存在呢？ (2) 既然上下文环境只能在一个请求中，而一个请求中似乎也不会创建两个以上的请求或应用上下文。那用 ThreadLocal 本地变量就行，什么要用栈呢？

(3) 为什么要放在“栈”里：在 Web 应用运行时中，一个线程同时只处理一个请求，那么 _req_ctx_stack 和 _app_ctx_stack 肯定都是只有一个栈顶元素的。那么为什么还要用“栈”这种结构？

查了一些资料后，对于第一个问题：虽然在 flask 应用中，一个 app 就能基本实现一个简单的 web 应用，我们知道对一个 Flask App 调用 app.run() 之后，进程就进入阻塞模式并开始监听请求。此时是不可能再让另一个 Flask App 在主线程运行起来的。那么还有哪些场景需要多个 Flask App 共存呢？前面提到了，一个 Flask App 实例就是一个 WSGI Application，那么 WSGI Middleware 是允许使用组合模式的，就可以支持多个 app 共存。就像 request 一样，在多 app 情况下也要保证 app 之间的隔离。那在 flask 中如何实现多个 app 呢？使用中间件 DispatcherMiddleware。这个将在以后介绍。(挖坑了。。。)

对于第二第三个问题，其实回答是一样的。在 web 环境下，确实没必要弄这么麻烦，就算多个 Flask App 同时工作也不是问题，毕竟每个请求被处理的时候是身处不同的 Thread Local 中的。不过 Flask 支持在离线环境中跑自动测试。但是 Flask App 不一定仅仅在 Web Runtime 中被使用 —— 有两个典型的场景是在非 Web 环境需要访问上下文代码的，一个是离线脚本（前面提到过），另一个是测试。这两个场景即所谓的 “Running code outside of a request”。这个将在以后介绍。(又挖坑。。。。)

---

flask 应用上下文和请求上下文

> 外一个问题，在形成这些“全局变量”的时候，使用了 werkzeug.local 模块的 LocalProxy 类。之所以要用该类，主要是为了动态地实现对栈顶元素的引用。如果不使用这个类，在生成上述“全局变量”的时候，它们因为指向栈顶元素，而栈顶元素此时为 None，所以这些变量也会被设置为 None 常量。后续即使有上下文对象被推入栈中，

相应的“全局变量”也不会发生改变。为了动态地实现对栈顶元素的引用，这里必须使用 werkzeug.local 模块的 LocalProxy 类

这个什么意思，全局变量的变化？

请求钩子

问题记录

flask templates 如何对应到不同的 blueprint 中的函数

```
url_for('todo.index')
```

直接指向 blueprint 的路由函数就可以了

Flask Blueprint AttributeError: 'module' object has no attribute 'name' error

从 [^6] 中可以看到，这个问题，主要是由于实际在 register_blueprint 时，todo 不是 Blueprint 对象导致的，添加一个 `isinstance(obj, Blueprint)` 可以有效帮助排查问题。

然后，主要是导入模块上出现的问题，由于我 Bluprint 设置的对象名与模块文件名相同，导入了自定义包里的模块 (即文件名)，而不是导入的 Blueprint，所以才会出现上述问题。

Reference

1. Flask Doc
2. Flask 项目配置
3. How to access app.config in a blueprint?
4. 一个 Flask 应用运行过程剖析
5. Flask 进阶 (一)——请求上下文和应用上下文完全解答 (下) [^6]. Flask Blueprint AttributeError: 'module' object has no attribute 'name' error