

# Wsgi工具简介

林义涵 ( 学号 : 13331158 )

## 背景

---

Python Web 开发中，服务端程序可以分为两个部分，一是服务器程序，二是应用程序。前者负责把客户端请求接收，整理，后者负责具体的逻辑处理。为了方便应用程序的开发，我们把常用的功能封装起来，成为各种Web开发框架，例如 Django, Flask, Tornado。不同的框架有不同的开发方式，但是无论如何，开发出的应用程序都要和服务器程序配合，才能为用户提供服务。这样，服务器程序就需要为不同的框架提供不同的支持。这样混乱的局面无论对于服务器还是框架，都是不好的。对服务器来说，需要支持各种不同框架，对框架来说，只有支持它的服务器才能被开发出的应用使用。

这时候，标准化就变得尤为重要。我们可以设立一个标准，只要服务器程序支持这个标准，框架也支持这个标准，那么他们就可以配合使用。一旦标准确定，双方各自实现。这样，服务器可以支持更多支持标准的框架，框架也可以使用更多支持标准的服务器。

Python Web开发中，这个标准就是 The Web Server Gateway Interface, 即 WSGI. 这个标准在PEP 333中描述，后来，为了支持 Python 3.x, 并且修正一些问题，新的版本在PEP 3333中描述。

## WSGI是什么

---

WSGI 是服务器程序与应用程序的一个约定，它规定了双方各自需要实现什么接口，提供什么功能，以便二者能够配合使用。

WSGI 不能规定的太复杂，否则对已有的服务器来说，实现起来会困难，不利于WSGI的普及。同时WSGI也不能规定的太多，例如cookie处理就没有在WSGI中规定，这是为了给框架最大的灵活性。要知道WSGI最终的目的是为了更方便服务器与应用程序配合使用，而不是成为一个Web框架的标准。

另一方面，WSGI需要使得middleware ( 是中间件么？ ) 易于实现。middleware处于服务器程序与应用程序之间，对服务器程序来说，它相当于应用程序，对应用程序来说，它相当于服务器程序。这样，对用户请求的处理，可以变成多个 middleware 叠加在一起，每个middleware实现不同的功能。请求从服务器来的时候，依次通过middleware，响应从应用程序返回的时候，反向通过层层middleware。我们可以方便地添加，替换middleware，以便对用户请求作出不同的处理。