QuickBurro 中间件 V8.00

Python 语言框架安装方法

一、 Windows 版安装方法

1.1. Python for Windows 安装

需要先在应用服务器上安装 Python for Windows 软件,32 位的 Windows Server,安装 32 位的 Python 系统;64 位的 Windows Server,则安装 64 位的 Python。Python 的版本建议 选择 V3.x,比如 V3.9。



安装时,需要指定将 Python 目录添加到环境变量 path 中,以便在 Windows 命令行下能 找到 Python 主程序:

🖕 Python 3.6.4 (64-bit) Si	etup	– 🗆 X
	Advanced Options	
	Install for <u>a</u> ll users	
	Associate files with Python (requires the	py launcher)
	Create shortcuts for installed application	S
	Add Python to environment variables	将Python添加到环境变量中
	Precompile standard library	
	Download debugging symbols	
	Download debug binaries (requires VS 2	015 or later)
	Customize install location	
	C:\Python36	B <u>r</u> owse
python	You will require write permissions for the se	elected location.
windows	Back	Install Cancel

安装完成后(可能需要重启计算机),在 Windows 的 cmd 窗口输入 "python",能顺利 打开 Python,表示安装成功:



1.2. 启动中间件服务

将中间件 Windows 版中的 Server32 或 Server64 复制到应用服务器的合适目录下,双击 其中的 burrocontrol.exe,密码空,点"确定"按钮,打开服务管理器主界面,再点"启动" 按钮启动中间件服务。如正式用户,再进行中间件的注册,注册后重启服务,显示如下界面:

磁気 組内 円白 子书点 数据库 抽件 计划任务 web/K名 Web/K2 Web/K2		QuickB The Set	urro多层分; rvice Mana;	布式中间f ger for G	牛 年费会员版Vi uickBurro Mide	00 节点服 Lleware Node	务管理器 a Server					① 关于		(升级	3	③ Web预览	也遇出
服务控制 原目名称 原目値 原目金 原目値 原目値 ・ ・	监控	选项	组网	用户	子节点	数据库	插件	计划任务	Web服务	Web域名	Web代理	Web脚本	安全性	认证	网盘	长连接	日志
	服务控制	RS IT			页目名称 系统账户数 下属于有点数 %。场面就是一个数 %。场面就是一个数 %。场面就是一个数 后的数据的。 是一个数 后的数据。 是一个数 后的数据。 是一个数 后的数据。 是一个数 后的数据。 是一个数 后的。 是一个句子。 是一句子。 是一句子。 是一句子。 是一句子。 是一句子。 是一句子。 是一句子。 是一句子。 是一句子。 是一句子。 是一句子。 是一句子。 是 一字 是 一句子。 是 一句子。 是 一子 一句子。 是 一句子。 是 一句子。 是 一句子。 是 一句子。 是 一句子。 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一	X	項 4 4 0/2 2 7 0 2 0 2 0 0 1 0 0 4 1 5 0 0 4 1 5 0 0 2 0 0 2 0 0 7 4 1 5 7 2 7 7 0 7 0 7 7 0 0 2 2 0 0 7 7 7 0 2 2 0 7 0 2 2 0 7 0 2 2 0 7 0 2 2 0 7 0 2 2 0 7 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0 0 2 2 0 0 0 2 2 0 0 0 2 0 0 1 0 1	目値	0(64 bits) stServer64 Studio 3516C-955251	¥7\	项向活业。 一、一、一、一、一、一、一、一、一、一、一、一、一、一、一、 有一、一、一、一、一、一、一、一、一、一、一、一、一、 有一、一、一、一、一、一、一、一、一、一、一、 有一、一、一、 有一、一、一、一、一、一、一、一、 有一、一、一、一、一、一、一、 有一、一、一、一、一、一、一、 有一、一、一、一、一、一、 有一、一、一、一、一、一、 有一、一、一、一、一、一、 有一、一、一、一、一、一、 有一、一、一、一、一、 一、二、一、一、一、一、 一、二、一、一、 一、二、一、一、一、 一、二、一、一、一、 一、二、一、一、 一、二、一、 一、二、一、 一、二、一、 一、二、一、 一、二、一、 一、二、一、 一、二、一、 一、二、一、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 一、二、二、 二、二、二、 一、二、二、 二、二、二、 二、二、二、 一、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、 二、二、二、 二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、二、 二、二、 二、二、二、 二、	動 会構の数 動 合構の数 数 数 数 数 数 数 数 数 数 数 数 数 数	15 17 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	目前 15 15 15 15 15 15 15 15 15 15 15 15 15			
	21	停止	6) #04	HÎ	→ AC IT = 1700R 许可长连接并发 许可Web会话并发 当前服务器时间 服务持续运行时	数 交数 间	500 500 200 1:5	2000 2000 23-03-10 12:57 58:27	5:29		文件高速缓 流媒体点播 Web验证码: 文件上传会	存项数 会话数 会话数	0, 0, 0, 0, 0	~ ′9 ′0		2	🔊 Rigi

表示中间件服务已经准备就绪。

1.3. 设置中间件服务的登录方式

由于 Python 的安装是在登录用户模式下进行的,其 Path 等环境变量也只在登录用户模 式下才有效。为了让服务方式运行的中间件能使用 Python 路径等环境,需要将中间件服务 设置为以 Windows 用户方式登录,比如,以 administrator 身份登录。

方法是在 Windows 的"控制面板—系统与安全—管理工具—服务"里找到中间件的服务 "Node Service for QuickBurro Network",鼠标右键点击—选属性,指定以 Windows 用户(比 如 administrator 用户)方式登录,再保存,再重启中间件服务:

Node S	Service Fo	or Quick	Burro Network 的属性(本地计算机)	×
常规	登录	恢复	依存关系	
登录	身份:			
0本	地系统帐	户(L)		
	一允许服务	务与桌面3	5回(W)	
⊚⊯	,帐户(T):		.\administrator 浏览(B)
평	密码(P):		•••••	
矿	制密码((C):	•••••	
			确定取消	应用(A)

1.4. 启用 qup 脚本语言引擎

中间件发布包里默认的 Python 脚本语言引擎是禁用的,可以在中间件的服务管理器的 "Web 脚本"配置页里进行启用设置:

	注土1 石			7#5	粉裙库	14-04	江利在冬		11.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	1.1.421冊	人」 Walk的木) HHP 324	315T		上法按	
:	远坝	粗州 用户		, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	资料1月1年	间十	IT ANT 5	neo1095	wep138-0	rep1公王	n e ollohvite	女主性	NUL	Prim	下庄按	
ſ	💡 Lua	脚本引擎(qua):														
				☑启用。	ua引擎											
		quat	密密钥	IVw30-3	28+22 j@sk? j	sha!`kspo	~kja	?	qua文件后	22 . qu	8					
		quaž	始标记	[\$qua\$]					qua结束标	-\$J 5	qua\$]					
		· 注音, 太 祖無叛約	おら. 4	《活金 白明	冬甘能生动						1	保存		还原		
		注意:本组参数修	改后,小	必须重启服	务才能生效						[保存		还原		
		注意:本组参数修	改后,小	必须重启服	务才能生效							保存		还原		
	V Pyt	注意:本组参数修 hon 脚本引擎(qu	改后,,):	必须重启服 	务才能生效						[保存		还原		
	💡 Pyt	注意:本组参数修 hon 脚 本引擎(quj	改后, ,	2.须重启服	务才能生效 				qup对象池大	J. 16	[保存		还原		
[💡 Pyt	注意:本组参数修 hon 脚本引擎(qu y qup)	改后, ,): 密密钥	必须重启服 ☑ 启用。 [sn_80s/	寄才能生效 mp引擎	jma98`a. p	q	?	qup对象地大 qup文件后	∫\ 16	P	保存		还原		
	💡 Pyt	注意:本组参数修 hon 脚本引擎(qu j qup <i>t</i> Pvtho	改后, 」):)密密钥	必须重启服 ☑ 启用。 sn_80s/ F:\0uic	済才能生效 mp引擎 jp[j1\$asu+s kBurro\Test	jma98`a. p Server64	'q V7\plugins\p	?	qup对象地大 qup文件后 Pvthon库文	小 16 鶸 . qu 業 . pvt	P hon39. dl1	保存		还原		
	9 Pyt	注意:本组参数修 hon 脚本引擎(quy 加 quy加 Pytho	改后, 」):))))) ()))))	 须重启服 ☑ 启用。 sn_8%s/ F:\Quie [f:	傍才能生效 up引擎 jp{j1\$asu+s kBurro\Test	jma98`a. p Server64_	'q V7\plugins\p	? python\	qup对象池大 qup文件后 Python库文	小 16	₽ hon39. dl1	保存		还原 		
	💡 Pyt	注意:本组参数修 hon 脚本引擎(qu qup Pytho qup	改后, ,):)密密钥 公库路径 验标记	 ○ 倉用 q sn_8%s/ F:\Quio [\$qup\$] 	済才能生效 mp引撃 jp{j1\$asu*s kBurro\Test	jma98`a.j Server64_	P⊈ V7\plugins\p	? bython\	qup对象池大 qup文件后 Python库文 qup结束标	小 16 鶸 .qu 牛 pyt 2 [\$/	P hon39. dll qup\$]	保存		还原 		
	💡 Pyt	注意:本组参数修 hon 脚本引擎(quy qupf qupf	改后, ,):))))))))))))))))))	※须重启服 ✓ 启用 q sn_80s/ F:\Qui o [\$qup\$]	务才能生效 uup引擎 kBurro\Test	jna98`a. p Server64_	Q Y7\plugins\p	? python\	qup对象地大 qup文件后 Python库文 qup结束标	小 16 鶸 . qu 井 pyt 己 [\$/	P hon39. dll qup\$]	保存		还原 		

操作时注意需要选定版本正确的 Python 语言的动态库文件 python??.dll,如下图示:

目织 ▼ 新建文件夹				== -
^	~	修改日期	类型	大小
▶ 快速访问				
logs	DLLs	2023-01-30 20:17	文件夹	
QuxBuilder	Doc	2023-01-30 20:17	文件夹	
OuxBuilder	Include	2023-01-30 20:17	又件夹	
OuvRuilder	Lib	2023-01-30 20:17	文件夹	
_ Quxbuilder	libs	2023-01-30 20:17	文件夹	
OneDrive	Scripts	2023-01-30 20:17	又件夹	
业由版	tcl	2023-01-30 20:17	又件夹	
	Tools	2023-01-30 20:17	又件夹	
3D 对象	python3.dll	2021-06-28 16:08	应用程序扩展	59 KB
- 视频	ypython39.dll	2021-06-28 16:08	应用程序扩展	4,381 KB
📰 图片	vcruntime140.dll	2021-06-28 16:09	应用程序扩展	95 KB
	vcruntime140_1.dll	2021-06-28 16:09	应用程序扩展	37 KB
↓下载				
▶ 音乐				
三 桌面				
▲ 本地磁盘 (C:) ¥				
	(7.0)			

指定启用 qup 引擎、选定了 Python 动态库、设置好 qup 引擎的工作参数后点击"保存" 按钮,再重启中间件服务,嵌入 QuickBurro 中间件的 Python 脚本语言引擎就开始工作了!

1.5. 运行示例程序

可以将发布包里 Demo601-QupDemo 中的大量示例文件复制到中间件的 homepage 目录下,就可在浏览器中用 url 来访问它们。有些示例模块需要使用数据库、文件等资源,请您事先查看说明进行准备。当然,更重要的是,您可以使用发布包的"3-Public\7-Tools\QuxBuilder"中的在线编辑发布工具来浏览、查看、编辑、发布这些示例程序:



测试用数据库,可以使用古老的 MS SQL Server 中的 Northwind 库,注册为中间件里的 "testdb"数据库。

二、 Linux 版安装方法

2.1. Python for Linux 安装

Linux 下 Python 脚本语言引擎的安装方法与 Windows 下基本一样,但是,由于有些 Linux 版中,Python 语言环境是标配(比如 UBuntu),故安装 Python 这一步可以视具体情况不同 而灵活对待。如 Linux 下未安装 Python,或安装的版本太低,则可以先安装。

Linux 下安装 Python,不同的 Linux 版本方法不一样,可以先搜索安装方法后根据其他 人的介绍按步进行。Python 的官网是 <u>https://www.python.org/</u>,可先下载,再解包安装。

与 Windows 版一样, Linux 下的 Python 安装完成后,在命令行下输入 python 后能进入 python 语言环境,表示安装就绪:

```
root@ubuntu:~/server64# python
Python 3.6.9 (default, Feb 28 2023, 09:55:20)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

2.2.启动中间件服务

按 Linux 版中间件的操作说明部署并启动中间件:



2.3.启用 qup 脚本语言引擎

对于带 GUI 界面的 Linux 服务器, 启用 Linux 版中间件的 Python 脚本语言引擎与 Windows 版的类似,只要打开 guiconsole 工具,在"Web 脚本"页指定启用 qup 框架、指定 Python 动态库、保存参数、重启服务,即可。

对于命令行的 Linux 服务器,首先需要先找到 libpython??.so 动态库,将它复制到中间 件的 plugins/python 目录下:

root@ubuntu:/	/usr/lib	/x86_64-linux-gnu#	ls libpython*	
libpython2.7.	so.1	libpython3.6m.a	libpython3.6m.so.1	
libpython2.7.	so.1.0	libpython3.6m.so	libpython3.6m.so.1.0	
root@ubuntu:/	usr/lib	/x86_64-linux-gnu#	cp libpython3.6m.so.1	~/server64/plugins/python
root@ubuntu:/	/usr/lib	/x86_64-linux-gnu#		

然后编辑 config/qupengine.sys 文件,指定"QupEnabled"为 True,指定 "PyLibPath"及 "PyLibFileName",保存后,重启中间件的服务:



2.4.运行示例程序

可以将发布包的 "Demo601-QupDemo" 目录下的大量示例文件上传到 Linux 版中间件 的 homepage 下,再用 QuxBuilder 进行在线查看、编辑、试运行:



三、 扩展模块引用方法

3.1.qup 扩展模块索引

QuickBurro 中的 Python 脚本语言引擎中,我们专门扩展了 18 个模块 400 余个函数,供应用开发使用,模块名称与功能如下:

'qup_webio'	{Web 接口模块}
'qup_encoding'	{编码与压缩算法模块}
'qup_parcel'	{Parcel 容器对象模块}
'qup_json'	{Json 容器对象模块}
'qup_xml'	{XML 容器对象模块}
'qup_dataset'	{数据集容器对象访问模块}
'qup_dbaccess'	{数据库读写模块}
'qup_rpc'	{远过程调用(RPC)模块}
'qup_websession'	{Web 会话管理模块}
'qup_jwt'	{JWT 令牌模块}
'qup_wssession'	{WebSocket 会话管理模块}
'qup_connections'	{TCP 长连接会话管理模块}

'qup_message'	{各类消息传输模块}
'qup_localfile'	{中间件本地文件访问模块}
'qup_file'	{远程文件目录访问模块}
'qup_binfile'	{远程二进制文件访问模块}
'qup_config'	{中间件配置管理功能模块}
'qup_misc'	{杂项功能模块}

3.2.模块引用与函数调用

可以用 Python 中的 import 语句进行以上扩展模块的引用, import 的全部语法都支持, 比如使用 "from qup_misc import *"引用 qup_misc 模块。每个模块分别有哪些函数,请各 位自行参考 QuxBuilder 工具主界面右侧 "qup API"列表中的详细说明。

3.3.编写 Python 库函数

可以用 Python 语言编写专门的.py 库函数单元供 qup 模块调用。.py 文件编写完成,应 保存到中间件主目录下的 "python"子目录中。比如,我们编写了一个 "testfunc.py" 库函 数文件,将它保存到中间件的 python 子目录下,其内容:

def hello():

import qup_misc
print('
这是 testfunc.py 中的 hello 函数的输出!
>')
print('中间件主文件目录: ',qup_misc.mainpath(),'
')
print('中间件的 python 子目录用于保存第三方库文件,
')
print('作为测试: testfunc.py 模块位于此目录下! 若本段信息
')
print('在调用页面正常输出,说明 qup 页面调用 Python 模块成功!
>
>

在其他.qup 文件里,可以方便地 import 此库文件,再调用其中的函数 hello(),比如: <html><head>

<title>qup 调用 Python 目录中的 py 文件</title>

<meta http-equiv="content-type" content="text/html; charset=UTF-8">

</head>

<body>

[\$qup\$]

def testmain():

print('qup 调用中间件 Python 子目录中的 py 文件:模块间调用示例

>')

import testfunc

print('-----开始调用 testfunc.hello------
')

testfunc.hello()

print('-----结束调用 testfunc.hello------
')

print('
')

print('Copyright(C) Jopher Software Studio 2023 年 2 月')

testmain()

[\$/qup\$]

</body></html>

下图是此.qup 文件执行后的效果:

Editor Previewer
http://192.168.1.104:1880/qupdemo/misc/qupcallfile.qup
qup调用中间件Python子目录中的py文件:模块间调用示例
开始调用 testfunc.hello
这是testfunc.py中的hello函数的输出!
中间件主文件目录: /root/server64/ 中间件的python子目录用于保存第三方库文件, 作为测试: testfunc.py模块位于此目录下! 若本段信息 在调用页面正常输出, 说明qup页面调用Python模块成功!
结束调用testfunc.hello
Copyright(C) Jopher Software Studio 2023年2月

四、 其他说明

由于时间紧张,V8中的Python脚本语言引擎的安装、使用说明由本文档担当,后续版本会进行更深入细致地编写,并纳入到《QuickBurro中间件从入门到精通》资料中。

樵夫软件工作室 2023 年 3 月 10 日