
chaos2wechat Documentation

Release 1.1.131010

Zoom.Quiet

October 17, 2013

CONTENTS

1		3
1.1	0: ?!	3
1.2	1:	8
1.3	2: ?	15
1.4	1	21
1.5		42
1.6		42
1.7		42
	<i>reStructuredText</i>	
1.8		43
1.9		47

, 42 SAE wechat ;-)

- - „ Zoom.Quiet ...
- - : <https://github.com/ZoomQuiet/chaos2wechat>
 - : <http://chaos2wechat.RTFD.org>
 - : OBP
 - : OBP
- TODO
 -
 -

1.1 0: ?!

,42 SAE wechat ,,,,

:

- 42 ?! ,,!
- 42 , Linux/Unix ~!;,,
- 42 , ,, JS ,,,,
- 42 , ,” “!
- 42 , ,/,,, ;-)
- 42 , ,(git/apt/yum/brew ...)

, ;-)

Warning:

- !

See Also:

- :
 - - Sina App Engine
 - Python Learn Python The Hard Way — Python v1.0 documentation
 - -
 - 42SAE!;-)

1.1.1

- Python ?
- SAE ?
- Bottle ?
- wechat ?

, !

```
wechat ~ : !
    • , ,,,,SNS wechat !
    • wechat ,, !
, , SINA , SAE PaaS , GAE ,,,,
    • ,,,
    • ,,,
, SAE ,, C++/JAVA/.NET ,,,
```

Note: (~_~)

- ,, Python Git
- M\$,,,,

Warning:

- SAE ?!
- ~~~ , go!

1.1.2 00:00”

! - ,: 00:00

: dev_server

- ,:
 - MacOS X 10.7.3
 - Python 2.7.1
- , Linux !
- :
- \$,;
- # ‘root’,;

SAE

```
$ git clone http://github.com/SAEPython/saepythondevguide.git
$ sudo python setup.py install
$ dev_server.py --help
Usage: dev_server.py [options]
```

Options:

```
-h, --help          show this help message and exit
-p PORT, --port=PORT Which port to listen
--mysql=MYSQL       Mysql configuration: user:password@host:port
```

```
--storage-path=STORAGE
    Directory used as local stoarge
```

SAE saecloud

```
$ saecloud version
SAE command line v0.0.1
$ saecloud -h
usage: saecloud [-h] {version,export,deploy} ...

positional arguments:
  {version,export,deploy}
                        sub commands
  export                export source code to local directory
  deploy               deploy source directory to SAE
  version              show version info

optional arguments:
  -h, --help            show this help message and exit
```

SAE, Python,,
wechat ,,,, ,

SAE !

SAE:

```
/path/2/you/wechat/
+- config.yaml
+- index.wsgi
```

config.yaml

```
---
name: wechat
version: 1
```

index.wsgi

```
1  # -*- coding:utf-8 -*-
2  from bottle import *
3  import sae
4
5  APP = Bottle()
6  application = sae.create_wsgi_app(APP)
7
8  @APP.get('/echo')
9  @APP.get('/echo/')
10 def echo_wechat():
11     print request.query.keys()
12     print request.query.echostr
13     return request.query.echostr
```

94!

,,:

```
$ dev_server.py
MySQL config not found
  * Running on http://localhost:8080/
  * Restarting with reloader
MySQL config not found
```

, SAE ,:, wechat ,

:

```
$ saecloud deploy
Deploying http://wechat.sinaapp.com
Updating cache
Finding changes
Deploying to server... done
```

,,,, ,: saecloud

?!

?!

...

高级功能 > 开发模式 > 成为开发者

成为开发者后，你才可以使用公众平台的开发功能。需要填写URL和Token，这两项信息需要你拥有自己的服务器资源。

接口配置信息

URL
必须以http://开头，目前只支持80端口

Token
必须为英文或数字，长度为3-32字符
[什么是Token?](#)

提交

Figure 1.1: 0-1

,

- wechat

- RESTful
- wechat Token
- ,5, wechat echostr ~ , web ,!
- ,

”

http://wechat.sinaapp.com

!

- http://wechat.sinaapp.com/echo
- wechat
- echostr ~

cURL

„!

- , SAE wechat ?!?!?!

,,,, cURL

```

zoomq @ MBP111216ZQ in ~/Works/@SAE/zhgdg
$ dev_server.py
MySQL config not found
* Running on http://localhost:8080/
* Restarting with reloader
MySQL config not found
['nonce', 'echostr', 'signature']
sdfsfd
127.0.0.1 - - [22/Sep/2013 17:36:47] "GET /echo/?signature=sdfsdf&echostr=sdfsfd&nonce=2341234 HTTP/1.1" 200 -

zoomq @ MBP111216ZQ in ~/Works/@SAE/zhgdg
$ curl "http://localhost:8080/echo/?signature=sdfsdf&echostr=sdfsfd&nonce=2341234"
sdfsfd
zoomq @ MBP111216ZQ in ~/Works/@SAE/zhgdg
$

```

Figure 1.2: 0-2 tumx +

, SAE sdk „!

.

7:01”

, „!

,,, SAE:

```

^ `->saecloud deploy
|           `-> curl
|           |
+-----/

```

1.1.3

, !



Figure 1.3: 0-3 “”

,:

- *POST* ?
- ?
- ?
- ?
- ...

Warning:

- ...

1.2 1:

wechat ,,, ,!

,:

- ?
- ?

Warning:

- !

1.2.1

/E, ,:

- [XML ~](#)
- [19.7. xml.etree.ElementTree — The ElementTree XML API — Python v2.7.5 documentation](#)

See Also:

- [Py](#)
- [Python](#)
- [Python](#)
- [Tutorial](#)

1.2.2 7:42”

,...

Bottle

„//!

: [bottleweb](#) | [Vimer](#)

:

```
/path/2/you/wechat/
+- config.yaml
+- index.wsgi
```

,:

```
/path/2/you/wechat/
+- config.yaml
+- config.py
+- index.wsgi
+- module/
|   +- utility.py
+- static/
|   +- js/
|   +- css/
|   +- img/
+- templates/
|   +- base.html
|   +- 404.html
|   +- ...
+- web/
    +- __init__.py
    +- mana4cli.py
    +- mana4api.py
```

```
:
index.wsgi
-> import config.py
-> from web import APP
-> __init__.py mount api.py
```

?

~web

- , Python Web!
- ? Python Web
- , ? ?
- CPyUG ,: ! ,,,...
- Bottle web ;-)

,:

index.wsgi

```
1 # -*- coding: utf-8 -*-
2 import sae
3 import config
4 from bottle import *
5 from web import APP
6
7 application = sae.create_wsgi_app(APP)
```

config.py

```
1 # -*- coding: utf-8 -*-
2 import sys
3 import os.path
4
5 app_root = os.path.dirname(__file__)
6 sys.path.insert(0, os.path.join(app_root, "module/"))
7 sys.path.insert(0, os.path.join(app_root, "web/"))
8
9 class Borg():
10     '''base http://blog.youxu.info/2010/04/29/borg
11         -
12     '''
13     __collective_mind = {}
14     def __init__(self):
15         self.__dict__ = self.__collective_mind
16
17     TPL_TEXT = ''' <xml>
18         <ToUserName><![CDATA[% (toUser)s]]></ToUserName>
19         <FromUserName><![CDATA[% (fromUser)s]]></FromUserName>
20         <CreateTime>% (tStamp)s</CreateTime>
21         <MsgType><![CDATA[% (text)s]]></MsgType>
22         <Content><![CDATA[% (content)s]]></Content>
```

```

23         </xml>'''
24
25     CFG = Borg()

```

web/__init__.py

```

1  # -*- coding: utf-8 -*-
2  from bottle import *
3
4  APP = Bottle()
5  APP.mount('/api', __import__('mana4api').APP)
6
7  @APP.error(404)
8  def error404(error):
9      return template('404.html')
10
11 @APP.route('/favicon.ico')
12 def favicon():
13     abort(204)

```

web/mana4api.py

```

1  # -*- coding: utf-8 -*-
2  import sae
3  from bottle import *
4  from config import CFG
5  #debug(True)
6  APP = Bottle()
7
8  @APP.get('/echo')
9  @APP.get('/echo/')
10 def echo_wechat():
11     print request.query.keys()
12     print request.query.echostr
13     return request.query.echostr

```

„Python, web/__init__.py “(mount); „!

url:

- index.wsgi,
- ,: - /echo - /api/echo - @APP.get('/echo')

Warning: !, URL

XML

: XML:

```

<xml>
<ToUserName><![CDATA[toUser]]></ToUserName>
<FromUserName><![CDATA[fromUser]]></FromUserName>
<CreateTime>1348831860</CreateTime>

```

```
<MsgType><![CDATA[text]]></MsgType>
<Content><![CDATA[this is a test]]></Content>
<MsgId>1234567890123456</MsgId>
</xml>
```

Pythonic , XML , , ElementTree ;

, xml.etree.ElementTree

web/mana4api.py

```
1 #...
2 @APP.post('/echo')
3 def wechat_post():
4     print request.forms.keys()
```

- , , ?!
- , wechat h
- , SAE :

```
⊕ - [2013/09/18 17:06:41] - ['<xml><ToUserName><![CDATA[gh_5e32c47b5b23]]></ToUserName>\n<FromUserName><![CDATA[oFNShji
OhclfJ-CtOO81p2sPrBfs]]></FromUserName>\n<CreateTime>1379495200</CreateTime>\n<MsgType><![CDATA[text]]></MsgType>\n<Co
ntent><![CDATA[?]]></Content>\n<MsgId>5924886768988979552</MsgId>\n</xml>'] yq34
```

Figure 1.4: 1-1 SAE

?! POST ?! ?! , request.forms.keys() !?

.. code-block:: python

linenos

emphasize-lines 4

```
#... @APP.post('/echo') def wechat_post():
    print request.forms.keys()[0]
```

! .. !!

.....

```
1 #...
2 @APP.post('/echo')
3 @APP.post('/echo/')
4 def wechat_post():
5     #print request.forms.keys()[0]
6     xml = etree.XML(request.forms.keys()[0])
7     print xml.findtext("Content")
```

ElementTree findtext() ,!

!

```

foo bar
127.0.0.1 - - [22/Sep/2013 22:30:46] "POST /api/echo/ HTTP/1.1" 200 -
zoomq @ MBP111216ZQ in ~/Works/@SAE/zhgdg
$ curl -d '<?xml><ToUserName><![CDATA[gh_5e32c47b5b23]]></ToUserName>\n<FromUserName><![CDATA[oFNShji0hc1fJ
e>\n<CreateTime>1379495200</CreateTime>\n<MsgType><![CDATA[text]]></MsgType>\n<Content><![CDATA[foo bar]]>
8979552</MsgId>\n</xml>' "http://localhost:8080/api/echo/"

```

Figure 1.5: 1-2 XML

```

,, h :
1 #...
2 @APP.post('/echo')
3 @APP.post('/echo/')
4 def wechat_post():
5     xml = etree.XML(request.forms.keys()[0])
6     fromUser = xml.findtext("ToUserName")
7     toUser = xml.findtext("FromUserName")
8     __MsgType = xml.findtext("MsgType")
9     __Content = xml.findtext("Content")
10    if "text" == __MsgType:
11        if "h" == __Content:
12            tStamp = TSTAMP()
13            content = ""
14            print CFG.TPL_TEXT% locals()
15            return CFG.TPL_TEXT% locals()
16    return None

```

cURL, SAE

:

27:00”

, 14, wechat !!!

,? , Bottle ,

- ,
-
- ,,,,,,
- ,print ,,,,,

!

1.2.3

?!

- !

```

127.0.0.1 - - [22/Sep/2013 22:30:46] "POST /api/echo/ HTTP/1.1" 200 -
h
<xml>
  <ToUserName><![CDATA[oFNShji0hc1fJ-Ct0081p2sPrBfs]]></ToUserName>
  <FromUserName><![CDATA[gh_5e32c47b5b23]]></FromUserName>
  <CreateTime>1309222362020884</CreateTime>
  <MsgType><![CDATA[text]]></MsgType>() :
  <Content><![CDATA[是也乎]]></Content>ML(request.forms.keys()[0])
</xml>fromUser = xml.findtext("ToUserName")
127.0.0.1 - - [22/Sep/2013 22:36:20] "POST /api/echo/ HTTP/1.1" 200 -
__MsgType = xml.findtext("MsgType")
Content = xml.findtext("Content")

$ curl -d '<xml><ToUserName><![CDATA[gh_5e32c47b5b23]]></ToUserName>\n<FromUserNa
e>\n<CreateTime>1379495200</CreateTime>\n<MsgType><![CDATA[text]]></MsgType>\n<Co
2</MsgId>\n</xml>' "http://localhost:8080/api/echo/"
<xml>
  <ToUserName><![CDATA[oFNShji0hc1fJ-Ct0081p2sPrBfs]]></ToUserName>
  <FromUserName><![CDATA[gh_5e32c47b5b23]]></FromUserName>
  <CreateTime>1309222362020884</CreateTime>
  <MsgType><![CDATA[text]]></MsgType>
  <Content><![CDATA[是也乎]]></Content>
</xml>

```

Figure 1.6: 1-3



Figure 1.7: 1-4

Warning:

- ...

ps:

saecloud SAE !!

pps:

config.yaml :

name: wechat

version: 2

...

- *saecloud deploy* ,
- , OK , SAE ,...

1.3 2: ?

„ Python ; Bottle ; SAE , :

- SAE
-
- : *h* :

,: ?!

agentzh

-
- 4599

,...

1.3.1

SAE :

- {412vol.49}: SAE
- SAE , :
 - SAE ,
 - ,
 - ,!

,,, SAE !

Warning:

- !

1.3.2 27:42”

, SAE Web

- MySQL RDC
- Memcache
- Storage
- KVDBkey-value
- Cron
- Image
- FetchURL
- Mail
- TaskQueue
- DeferredJob
- Counter
- Rank

...

,,, -;

KVDB !

:

```
1 import sae.kvdb
2 kv = sae.kvdb.KVClient()
3 k = 'foo'
4 kv.set(k, 2)
5 kv.delete(k)
6
7 kv.add(k, 3)
8 kv.get(k)
9
10 kv.replace(k, 4)
11 kv.get(k)
12
13 print kv.get_info()
```

!

:

OpenID

- , *OpenID* KVDB ;
- , *0FNShjiOhc1fJ-Ct2O80p2sPrBf5 OpenID* ,
- , hashlib
- , Python ,!

: *i* ~ Info. , information

KVDB

KVDB SAE NoSQL ,,,, SDK , dev_server.py ,:

```
$ dev_server.py -kvdb-file=./logs/kv.db
```

SAE SDK ,:

```
$ dev_server.py -h Usage: dev_server.py [options]
```

Options:

- h, --help** show this help message and exit
- p PORT, --port=PORT** Which port to listen
- host=HOST** Which host to listen
- mysql=MYSQL** Mysql configuration: user:password@host:port
- storage-path=STORAGE** Directory used as local stoarge
- kvdb-file=KVDB** File to save kvdb data

web/mana4api.py :

```
1 import sae.kvdb
2 KV = sae.kvdb.KVClient()
3 #...
4 @APP.post('/echo')
5 @APP.post('/echo/')
6 def wechat_post():
7     xml = etree.XML(request.forms.keys()[0])
8     fromUser = xml.findtext("ToUserName")
9     toUser = xml.findtext("FromUserName")
10    __MsgType = xml.findtext("MsgType")
11    __Content = xml.findtext("Content")
12    tStamp = TSTAMP()
13    if "text" == __MsgType:
14        if "h" == __Content:
15            content = ""
16            return CFG.TPL_TEXT% locals()
17        elif "i" == __Content:
18            uid = hashlib.sha1(toUser).hexdigest()
19            print uid
20            usr = KV.get(uid)
21            if None == usr:
```

```

22         # ,
23         pass
24     else:
25         #
26         if "em" in usr.keys():
27             #
28             content = ": %s" % usr['em']
29         else:
30             #
31             content = "\nem:foo@bar.com"
32         print CFG.TPL_TEXT % locals()
33         return CFG.TPL_TEXT % locals()
34     elif "em" in __Content.split(":"):
35         uid = hashlib.sha1(toUser).hexdigest()
36         usr = KV.get(uid)
37         print __Content[3:]
38         usr['em'] = __Content[3:]
39         KV.replace(uid, usr)
40         content = ": %s" % usr['em']
41         print CFG.TPL_TEXT % locals()
42         return CFG.TPL_TEXT % locals()
43
44     return None

```

,cURL SAE



Figure 1.8: 2-1

...

BaZINga

42:01”

~, 30, !

- „
- „? ?
- *print* ,?
- „„„„„

„„„„„

1.3.3

,:

config.yaml

```
1 ---
2 name: wechat
3 version: 3
```

index.wsgi

```
1 # -*- coding: utf-8 -*-
2 import sae
3 import config
4 from bottle import *
5 from web import APP
6
7 application = sae.create_wsgi_app(APP)
```

config.py

```
1 # -*- coding: utf-8 -*-
2 import sys
3 import os.path
4
5 app_root = os.path.dirname(__file__)
6 sys.path.insert(0, os.path.join(app_root, "module/"))
7 sys.path.insert(0, os.path.join(app_root, "web/"))
8
9 class Borg():
10     """base http://blog.youxu.info/2010/04/29/borg
11         -
12     """
13     __collective_mind = {}
14     def __init__(self):
15         self.__dict__ = self.__collective_mind
16
```

```
17     TPL_TEXT=''' <xml>
18         <ToUserName><![CDATA[% (toUser)s]]></ToUserName>
19         <FromUserName><![CDATA[% (fromUser)s]]></FromUserName>
20         <CreateTime>% (tStamp)s</CreateTime>
21         <MsgType><![CDATA[text]]></MsgType>
22         <Content><![CDATA[% (content)s]]></Content>
23     </xml>'''
24
25     CFG = Borg()
```

web/__init__.py

```
1     # -*- coding: utf-8 -*-
2     from bottle import *
3
4     APP = Bottle()
5     APP.mount('/api', __import__('mana4api').APP)
6
7     @APP.error(404)
8     def error404(error):
9         return template('404.html')
10
11     @APP.route('/favicon.ico')
12     def favicon():
13         abort(204)
```

web/mana4api.py

```
1     # -*- coding: utf-8 -*-
2     import sae
3     from bottle import *
4     from config import CFG
5     #debug(True)
6     APP = Bottle()
7     import sae.kvdb
8     KV = sae.kvdb.KVClient()
9
10     @APP.post('/echo')
11     @APP.post('/echo/')
12     def wechat_post():
13         xml = etree.XML(request.forms.keys()[0])
14         fromUser = xml.findtext("ToUserName")
15         toUser = xml.findtext("FromUserName")
16         __MsgType = xml.findtext("MsgType")
17         __Content = xml.findtext("Content")
18         tStamp = TSTAMP()
19         if "text" == __MsgType:
20             if "h" == __Content:
21                 content = ""
22                 return CFG.TPL_TEXT% locals()
23             elif "i" == __Content:
24                 uid = hashlib.sha1(toUser).hexdigest()
25                 print uid
26                 usr = KV.get(uid)
27                 if None == usr:
```

```

28         # ,
29         pass
30     else:
31         #
32         if "em" in usr.keys():
33             #
34             content = ": %s"% usr['em']
35         else:
36             #
37             content = "\nem:foo@bar.com"
38     print CFG.TPL_TEXT% locals()
39     return CFG.TPL_TEXT% locals()
40 elif "em" in __Content.split(":"):
41     uid = hashlib.sh1(toUser).hexdigest()
42     usr = KV.get(uid)
43     print __Content[3:]
44     usr['em'] = __Content[3:]
45     KV.replace(uid, usr)
46     content = ": %s"% usr['em']
47     print CFG.TPL_TEXT% locals()
48     return CFG.TPL_TEXT% locals()
49
50 return None

```

```

-> h
<-

-> i
<- \nem:foo@bar.com

-> em:bar@gmail.com
<- : bar@gmail.com

```

!

- ,?
- ?
- zoomquiet+gdg@gmail.com + ?!
- ?
- ?

Note: (~~)

- ! ,,!
-

1.4 1

„ Python ; Bottle ; SAE , :

- SAE
-
- : *h* :
- : ...

! , ,,,

Warning:

- !...

1.4.1

, 3,!

,,, //,!

,,, ,,:

-
- ,,?

,?

:

1. : s
- 2.
- 3.
4. ,,,,
- 5.
- 6.

? ?

- ,,,?
- ./,?
- ,?
- ,,,?
- ...

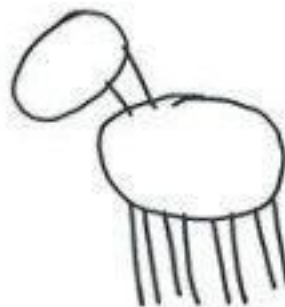
,,, Pythonic ,:

!

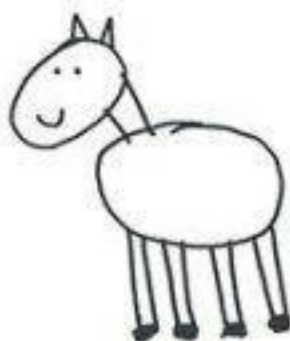
怎样画马



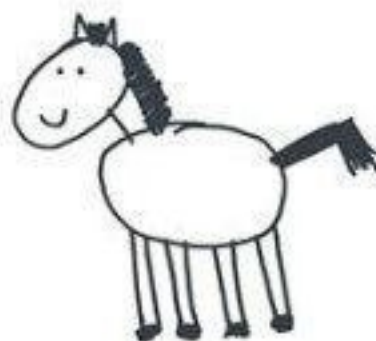
① 画两个圆圈



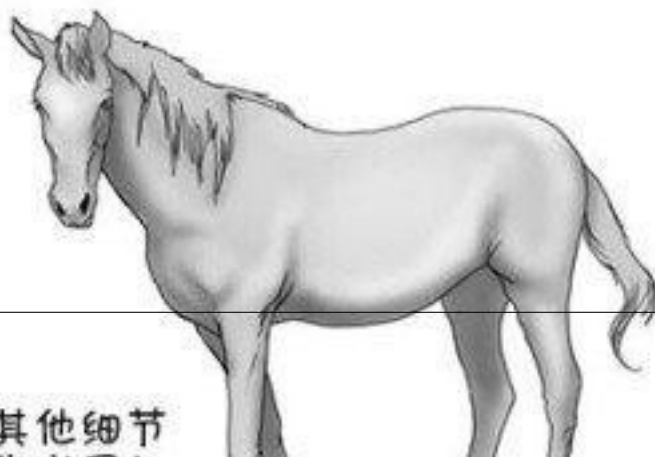
② 画上脚



③ 画上脸



④ 画上毛发



⑤

再添加其他细节

```
""" , , , , ,
, Pythonner Pythonic ,/, ,,!
, wecaht ,:
    • twinsant/pyweixin
    • jeffkit/wechat
"""""
```

Warning: (^o^)
!

1.4.2

```
,: ?!
    • , dev_server.py ,
    • cURL XML
! XML , ,!!
!

!

,:
curl -d '[XML]' http://localhost:8080/api/echo

[XML] :

<xml>
<ToUserName><![CDATA[toUser]]></ToUserName>
<FromUserName><![CDATA[fromUser]]></FromUserName>
<CreateTime>1348831860</CreateTime>
<MsgType><![CDATA[text]]></MsgType>
<Content><![CDATA[this is a test]]></Content>
<MsgId>1234567890123456</MsgId>
</xml>

,<Content><![CDATA[]]></Content>

CLI.py:
1 # -*- coding: utf-8 -*-
2 import sys
3 from time import time
4 from subprocess import Popen
5 from xsettings import XCFG
6
7 if __name__ == "__main__":
8     if 2 != len(sys.argv):
```

```

9         print """ Usage::
10             $ python CLI.py []
11             """
12     else:
13         TPL_TEXT='''<xml>
14             <ToUserName><![CDATA[% (toUser)s ]]></ToUserName>
15             <FromUserName><![CDATA[% (fromUser)s ]]></FromUserName>
16             <CreateTime>% (tStamp)s</CreateTime>
17             <MsgType><![CDATA[text]]></MsgType>
18             <Content><![CDATA[% (content)s ]]></Content>
19             </xml>'''
20
21         toUser = XCFG.AS_SRV
22         fromUser = XCFG.AS_USR
23
24         tStamp = int(time())
25         content = sys.argv[1]
26         xml = TPL_TEXT % locals()
27
28         cmd = "curl -d '%s' http://localhost:8080/api/echo"% xml
29         print cmd
30         #Popen(cmd, shell=True, close_fds=True)

```

```

,,:

```

```

$ python chaos_CLI.py
Usage::
    $ python CLI.py []

```

```

$ python chaos_CLI.py h
$ python chaos_CLI.py s
...

```

,XML,!

```

from xsettings import XCFG

```

```

,,, config.py

```

```

, GitHub, .gitignore

```

```

.ropeproject
*.dump
*.pyc
*.db
*.log
.svn
.DS_Store
*.wsgic
logs
xsettings.py

```

```

xsettings.py

```

web/mana4api.py :

```
1 import sae.kvdb
2 KV = sae.kvdb.KVClient()
3 #...
4 @APP.post('/echo')
5 @APP.post('/echo/')
6 def wechat_post():
7     xml = etree.XML(request.forms.keys()[0])
8     fromUser = xml.findtext("ToUserName")
9     toUser = xml.findtext("FromUserName")
10    __MsgType = xml.findtext("MsgType")
11    __Content = xml.findtext("Content")
12    tStamp = TSTAMP()
13    if "text" == __MsgType:
14        if "h" == __Content:
15            content = ""
16            return CFG.TPL_TEXT% locals()
17        elif "i" == __Content:
18            uid = hashlib.shal(toUser).hexdigest()
19            print uid
20            usr = KV.get(uid)
21            if None == usr:
22                # ,
23                pass
24            else:
25                #
26                if "em" in usr.keys():
27                    #
28                    content = ": %s"% usr['em']
29                else:
30                    #
31                    content = "\nem:foo@bar.com"
32                print CFG.TPL_TEXT% locals()
33                return CFG.TPL_TEXT% locals()
34        elif "em" in __Content.split(":"):
35            uid = hashlib.shal(toUser).hexdigest()
36            usr = KV.get(uid)
37            print __Content[3:]
38            usr['em'] = __Content[3:]
39            KV.replace(uid, usr)
40            content = ": %s"% usr['em']
41            print CFG.TPL_TEXT% locals()
42            return CFG.TPL_TEXT% locals()
43
44    return None

if elif else:

1 #...
2 @APP.post('/echo')
3 @APP.post('/echo/')
4 def wechat_post():
5     xml = etree.XML(request.forms.keys()[0])
6     fromUser = xml.findtext("ToUserName")
7     toUser = xml.findtext("FromUserName")
8     __MsgType = xml.findtext("MsgType")
```

```

9  __Content = xml.findtext("Content")
10 tStamp = TSTAMP()
11 if "text" == __MsgType:
12     if "h" == __Content:
13         content = ""
14         return CFG.TPL_TEXT% locals()
15     elif "s" == __Content:
16         #
17         pass
18     elif "dd" == __Content:
19         # dd ~ D
20         pass
21     elif "" == __Content:
22         #
23         pass
24     elif "i" == __Content:

```

Warning: (\o/)!!!
• !!!!

if else !

公众号文章查阅 {流程分析}

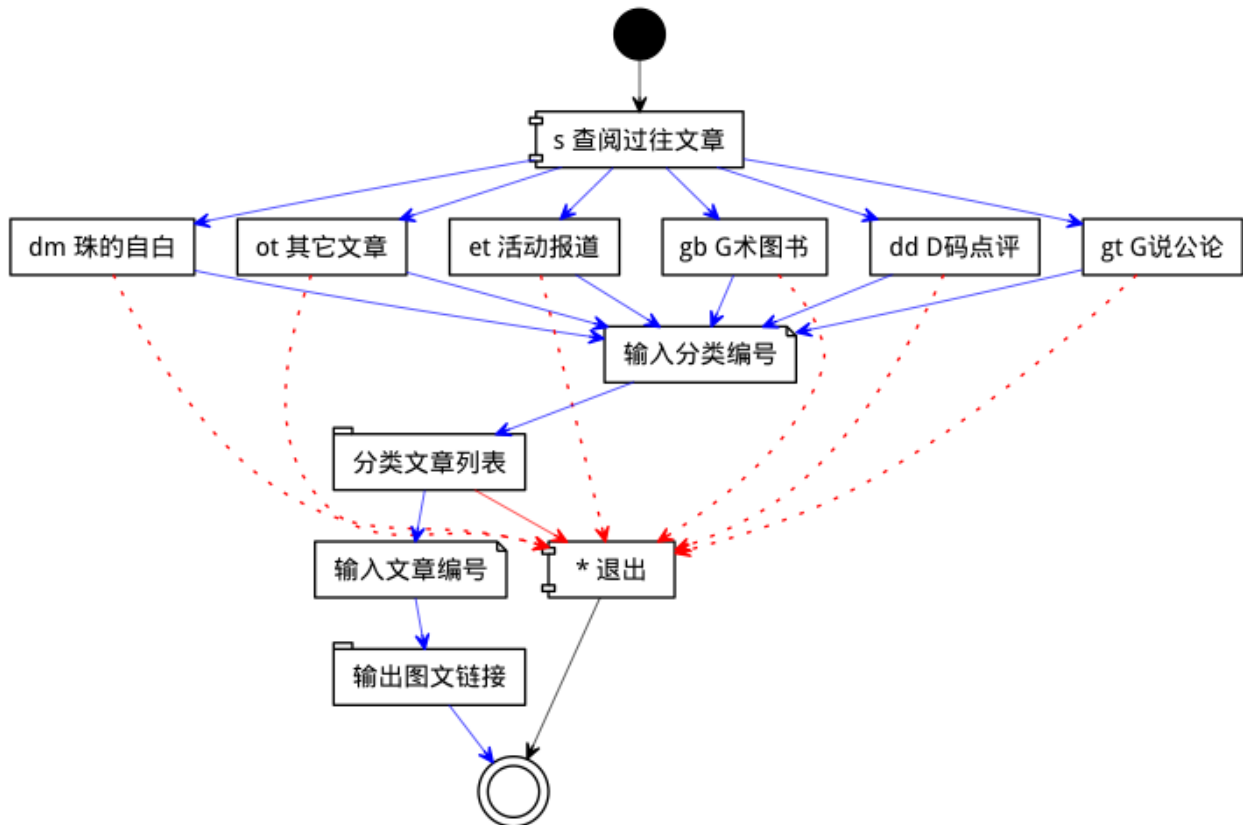


Figure 1.10: 3-2 *

```

”
, * ,”
: if elif else :
    • ,web
    • ”
    • ””
    • ” ,” !
!

```

BazingA

....

1.4.3 FSM

```

!
    • FSM : (Flying Spaghetti Monsterism)
    • FSM : (Free software movement)
    • FSM : (Finite State Machine)
:
    •
    •
    •

```

```

, ,”
, .
    •
    • ,
    • ,
    • , FSM !
“

```

Google Python [FSM](#) ,:

- [Finite State Machines in Python – No Pity for the Masses](#)
- [python - Code Golf: Finite-state machine! - Stack Overflow](#)
- [fsm - What are the best Python Finite State Machine implementations - Stack Overflow](#)
- [FiniteStateMachine - Python Wiki](#)
 - [AT&T’s Weighted FSM in python](#)

- FSME , A graphical FSM Editor w/ python target
- FSA
- fsm.py module
- fsm.py , Skip Montanero's FSM
- fysom ,A slick FSM implementation that provides function callbacks for each state.

....

!!

- , fysom JavaScript ,! , ,,,!
- !
- ,,!

, github :

- 1.
- 2.
3. ,+,!

pyfsm

pyfsm - Pure Python Implementation of a Finite State Machine

!!

```
@state('hello_world') # task 'hello_world'
@transition(1, 'goodbye') # transitions on 1 to the goodbye state
def say_hello(tsk): # state name is say_hello, the task is the second object
    ...
```

Python FSM , Pythonic !-)

samples.py:

```
import pyfsm
from pyfsm import state, transition

@state('say_hello')
@transition('goodbye', 'goodbye')
def meet_and_greet(self):
    print 'hello, world!'
    @self.callback('hi')
    def print_hello(event):
        print 'hello, again'

@state('say_hello')
def goodbye(self):
    print 'leaving so soon?'

say_hello = pyfsm.Registry.get_task('say_hello')
say_hello.start('meet_and_greet')
say_hello.send('hi')
say_hello.send('dead message')
#say_hello.send('hi')
```

```
say_hello.send('goodbye')

'''
seek->list->reg->conf->bye
'''
@state('my_reg')
@transition('end', 'end')
@transition('r', 'reg')
def seek(self):
    print 'seek crt. events...'
    @self.callback('l')
    def list(event):
        print 'list events:...'

@state('my_reg')
@transition('c', 'conf')
def reg(self):
    print 'reg some one event?!'

@state('my_reg')
@transition('b', 'bye')
def conf(self):
    print 'confirmed reg the event!'

@state('my_reg')
def bye(self):
    print 'bye and c u there!'

@state('my_reg')
def end(self):
    print 'broken flow!'

my_reg = pyfsm.Registry.get_task('my_reg')
my_reg.start('seek')
#my_reg.send('end')
my_reg.send('l')
my_reg.send('r')
#my_reg.send('c')
my_reg.send('b')

print "\tOTHER SFM matter..."
crt_reg = pyfsm.Registry.get_task('my_reg')
my_reg.start('reg')
my_reg.send('c')
```

Bottle! :

1. import pyfsm, Bottle ,,,,!
2. pyfsm.Registry.get_task('say_hello')
3. say_hello.start('meet_and_greet')
4. say_hello.send('hi') , ,,

Pythonic:

```
from pyfsm import state, transition
...
```

```

@state('say_hello')
@transition('goodbye', 'goodbye')
def meet_and_greet(self):
    ...

@state('say_hello')
def goodbye(self):
    ...

:
1. @state('say_hello')
2. @transition('goodbye', 'goodbye'),
3. def goodbye(self):
, pyfsm,...
,

```

公众号文章查阅 {流程分析}

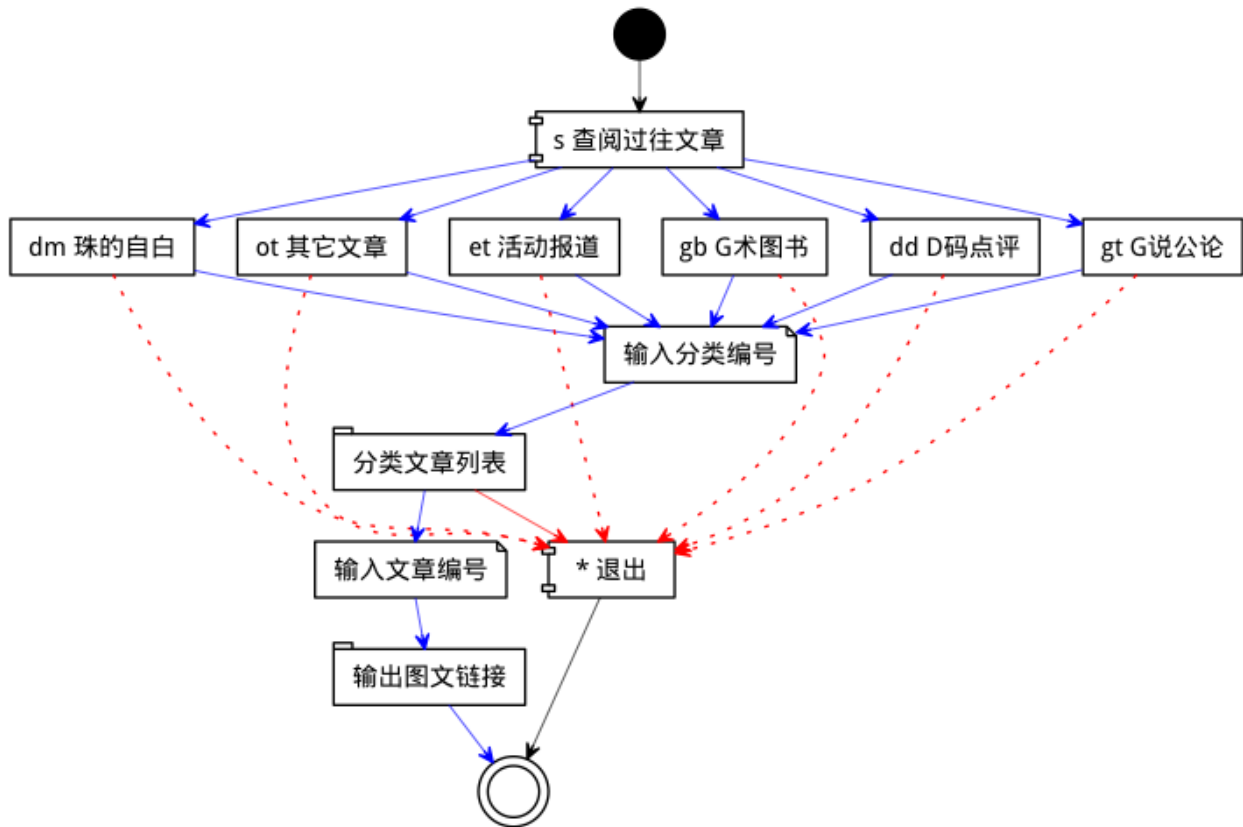


Figure 1.11: 3-2 *

```

:
@state('weknow')
@transition('s', 'seek')
def setup(self):

```

```

    print "start->"

    @state('weknow')
    @transition('dd', 'papers')
    ...
    @transition('ot', 'papers')
    @transition('*', 'end')
    def seek(self):
        print "start->seek->"

    @state('weknow')
    @transition('no', 'no_paper')
    @transition('*', 'end')
    def papers(self):
        print "start->seek->papers"

    @state('weknow')
    @transition('end', 'end')
    @transition('*', 'end')
    def number_paper(self):
        print "start->seek->papers->paper"

    @state('weknow')
    def end(self):
        print "->end"

    if elif else!:

1  #...
2  @APP.post('/echo')
3  @APP.post('/echo/')
4  def wechat_post():
5      xml = etree.XML(request.forms.keys()[0])
6      fromUser = xml.findtext("ToUserName")
7      toUser = xml.findtext("FromUserName")
8      __MsgType = xml.findtext("MsgType")
9      __Content = xml.findtext("Content")
10     tStamp = TSTAMP()
11     if "text" == __MsgType:
12         weknow = pyfsm.Registry.get_task('weknow')
13         weknow.start("")
14         return weknow.send(__Content)

!

• , !

• , @transition() ,

• , ?

!

, request , Python XML !!!

! ?!

! ...

:

```

- [twinsant/pyweixin](#) / ,
- [jeffkit/wechat](#) **Jeff**, **iOS/Android** , **Ukulele** „! !:
 - [gztechparty/techparty](#) !
 - !

[jeffkit/wechat](#) !

,?! ?!

- ! Django !
- „„
- [official.py](#)
- demo : [wechat/demo at master · jeffkit/wechat](#)

:

```
from wechat.official import WxApplication, WxRequest, WxTextResponse, WxNewsResponse, WxArticle
```

```
wxreq = WxRequest(request) #
```

```
...
```

```
WxTextResponse(, wxreq).as_xml() #XML
```

```
...
```

```
p1 = WxArticle(Url=,PicUrl=) #
```

```
p2 = WxArticle(Url=,PicUrl=)
```

```
WxNewsResponse([p1, p2]), wxreq).as_xml() # XML
```

:

```
xml = etree.XML(request.forms.keys()[0])
```

```
fromUser = xml.findtext("ToUserName")
```

```
toUser = xml.findtext("FromUserName")
```

```
__MsgType = xml.findtext("MsgType")
```

```
__Content = xml.findtext("Content")
```

```
...
```

```
return CFG.TPL_TEXT% locals()
```

,**Pythonic** !

!

:

1. **Bottle** , **wechat** ,:

```
@APP.post('/echo')
```

```
def wechat_post():
```

```
    print request.forms.keys()[0]
```

```
    ...
```

2. **pyfsm** **Pythonic** , **FSM** /,

3. [jeffkit/wechat](#) , **XML** /

:

1. **FSM** ?

2. pyfsm jeffkit/wechat wxreq?

SAE,?! ~ KVDB !

- , fsm ?!
- , ,!

:

```
1 @APP.post('/echo')
2 def wechat_post():
3     wxreq = WxRequest(request.forms.keys()[0])
4     uid = hashlib.shal(toUser).hexdigest()
5     G_CRT_USR = KV.get(uid)
6     wxreq.crt_usr = G_CRT_USR
7     # usage pyfsm as FSM echo all kinds of usr ask
8     weknow = pyfsm.Registry.get_task('weknow')
9     if G_CRT_USR['fsm']:
10         weknow.start2(G_CRT_USR['fsm'], wxreq)
11     else:
12         G_CRT_USR['fsm'] = "setup"
13         __update_usr(G_CRT_USR)
14         weknow.start2('setup', wxreq)
15     return weknow.send2(wxreq.Content.strip(), wxreq)
```

:

- wxreq.crt_usr = G_CRT_USR, jeffkit/wechat WxRequest ,,,!
- **weknow.start2('setup', wxreq) pyfsm**
 - (start)(send)
 - , Bottle, FSM

?

pyfsm :

- Changes - pyfsm - Pure Python Implementation of a Finite State Machine - Google Project Hosting
- „
- „
- „!

!

```
1 --- 3party/pyfsm.py 2013-09-26 17:34:56.000000000 +0800
2 +++ ../_3party/pyfsm/pyfsm.py 2013-09-24 09:24:26.000000000 +0800
3 @@ -266,60 +266,6 @@
4         if trans:
5             self.start(trans)
6
7 -     def start2(self, name, obj):
```

```

8 -     """
9 -     Starts the task with the given state name.
10 -
11 -     @param name: state name.
12 -     @param obj: data obj. for func.
13 -     @type name: C{str}
14 -     """
15 -     for x in self.exit:
16 -         x(self, obj)
17 -
18 -     self.current_state = self.states[name]
19 -     self.callbacks = {}
20 -     self.exit = []
21 -     self._locals = {}
22 -     return self.current_state.enter2(self, obj)
23 -
24 - def send2(self, event, obj):
25 -     """
26 -     Sends an event to this task.
27 -
28 -     It determines what key to use to identify the event by
29 -     calling the appropriate getattr function.
30 -
31 -     If any callbacks are registered for this event, then they are
32 -     invoked first.
33 -
34 -     If any transitions are registered for this event, a state transition
35 -     is invoked after completing the callbacks.
36 -
37 -     @param event: event to send to the state machine
38 -     @param obj: data obj. for func.
39 -     """
40 -     assert self.current_state, 'state machine is not running'
41 -
42 -     # recover the key for this event
43 -     for getattr in (self.getattr, Registry.getattr, lambda x: x):
44 -         try: key = getattr(event)
45 -             except: pass
46 -             else: break
47 -
48 -     # check callbacks first
49 -     callback = self.callbacks.get(key, [])
50 -     for x in callback:
51 -         x(event, obj)
52 -
53 -     # if a transition exists, change the state
54 -     trans = self.current_state.transitions.get(key, None)
55 -     if trans:
56 -         return self.start2(trans, obj)
57 -     else:
58 -         # when state transited, call old current_state func again!
59 -         return self.current_state.enter2(self, obj)
60 -
61 - def add_state(self, name, state):
62 -     """
63 -     Adds a state to this task.
64 - @@ -417,13 +363,11 @@
65 -     def __init__(self, name):

```

```

66         self.task = Registry.get_task(name)
67         self.transitions = {}
68     -
69     def __call__(self, func):
70         self.func = func
71         self.transitions.update(getattr(func, 'transitions', {}))
72         self.task.add_state(func.__name__, self)
73         return self
74     -
75     def enter(self, task):
76         """
77         Entrance function to this state.
78     @@ -432,15 +376,3 @@
79         @type task: L{pyfsm.task}
80         """
81         return self.func(task)
82     -
83     - def enter2(self, task, obj):
84     -         """
85     -         Entrance function to this state.
86     -
87     -         @param task: task this state is contained within
88     -         @param obj: data obj. for func.
89     -         @type task: L{pyfsm.task}
90     -         """
91     -         return self.func(task, obj)
92     -
93
94 **2 ,,!

```

Note: (~~)

- ! !,!
-

1.4.4 CLI

,,:

- !:
 - ?
 - /?
 - /?
 - ?
 - ?
 - ...
- ,SAE ,:
 -

-
-
-
- ...
- !
- /
- ?
- , Django
- ! Django MySQL , OMG !

? CLI !

CLI ~ Command-Line Interface

- : -5: « 4G spaces
- , CLI !
- :
- CLI , Linux MAC, M\$ Windows ,,, ,!
- CLI Shell ,// ,,,!
- CLI , HTML/JS/CSS ,,,!

- , !
- , ? !
- ,,,!
 - : 42 SAE !-)
 - , :
 - ZoomQuiet / ok.URIsA / source / index.wsgi — Bitbucket
 - !

- :
1.
 -
 - APPKEY
 - SECRET ()
 - md5 ,
 - sign
 2.
 - APPKEY SECRET

```

    •
    • ,sign
    •
    • ,, 0.42
    • ,/!
, APPKEY SECRET :

```

```

    • ,
    •
    • /,

```

RESTful

RESTful !

Bottle RESTful http , , :

```

@APP.get ('/cli/revert/<matter>')
...
@APP.put ('/cli/revert/<matter>')
...
@APP.post ('/cli/revert/<matter>')
...
@APP.delete ('/cli/revert/<matter>')

```

RESTful ,, , ,/, http , !

,,!

:

- [jkbr/httpie](#) cURL-like tool for humans
- [:\[httpie\] - Diary](#)

RESTful :

```

$ http GET http:///api/cli
$ http DELETE http:///api/cli
$ http -f PUT http:///api/cli
$ http -f POST http:///api/cli
$ http -f POST http:///api/cli json@path/2/

```

,:

```

<form enctype="multipart/form-data"
  method="post"
  action="http:///api/cli">
  <input type="file" name="json" />
</form>

```

CLI

,, CLI !

- ,!
- Pythonic CLI !

„ Pythonic :

- hfeeki/cmdln
- argparse – Command line option and argument parsing. - Python Module of the Week
- OptionParser –
- 13.2. ConfigParser — Configuration file parser — Python v2.7.5 documentation
- whtsky/parguments
- halst/docopt
- ...

, github :

- 1.
- 2.
3. ,+,!

, halst/docopt !

: docopt—language for description of command-line interfaces

!

- docopt , !!!
- „ Pythonic !
- ””

–

– //!

!

”,, ,...

Note: (~_~)

- ! ,,!
 - ...
-

1.4.5

: ZHGDG/zhgdg

: GDG-ZhuHai

珠海GDG公众号自动应答 SFM {状态变迁图谱}

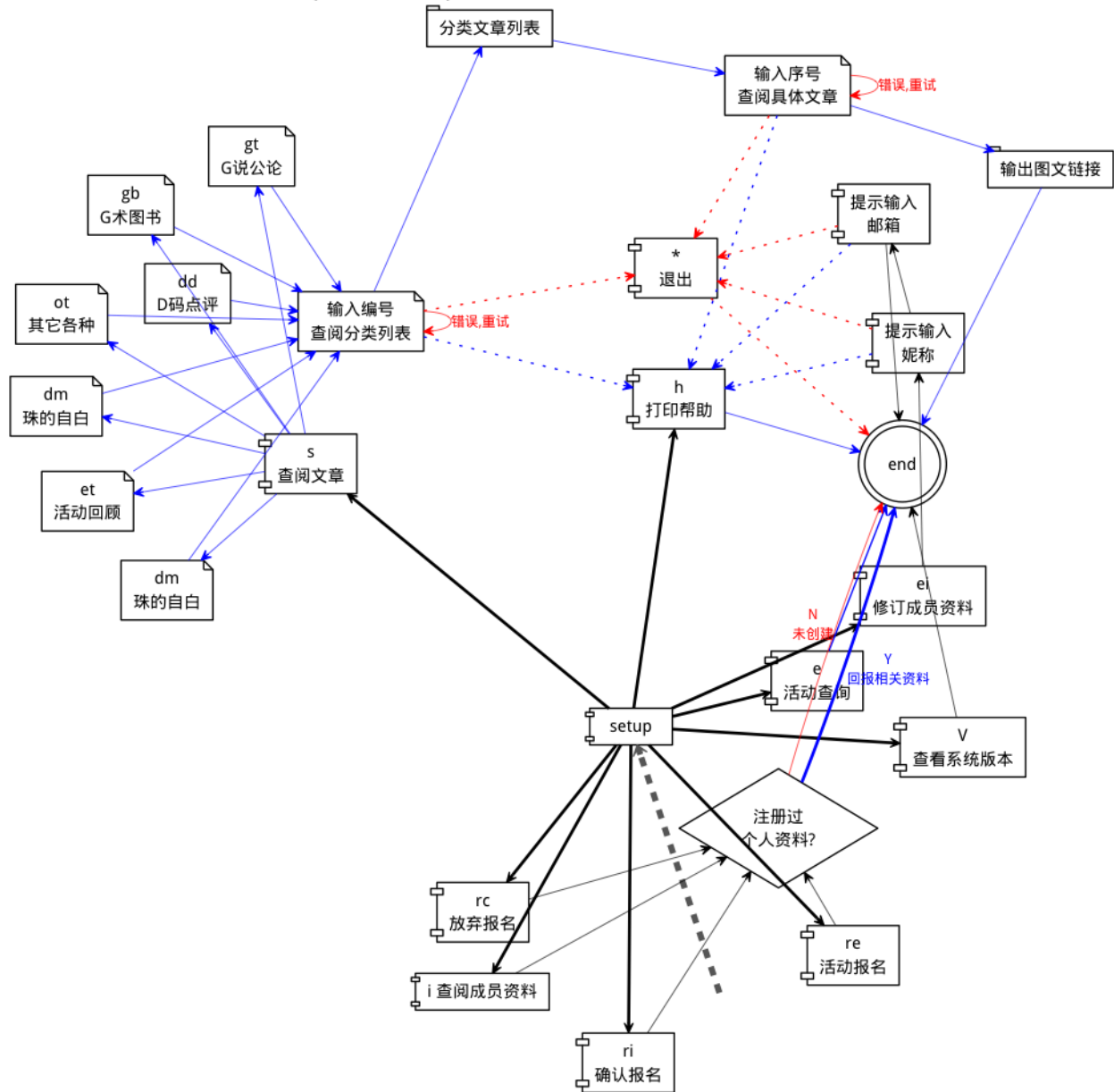


Figure 1.12: 3-3 GDG FSM



Figure 1.13: GDG

,!

fix/p/:TAG,?! SAE:

```
- Traceback (most recent call last):
  File "/data1/www/htdocs/466/weknow/3/bottle.py", line 764, in _handle
    return route.call(**args)
  File "/data1/www/htdocs/466/weknow/3/bottle.py", line 1575, in wrapper
    rv = callback(*a, **ka)
  File "/data1/www/htdocs/466/weknow/3/web/mana4api.py", line 509, in wechat_post
    return weknow.send2(wxreq.Content.strip(), wxreq)
  File "/data1/www/htdocs/466/weknow/3/3party/pyfsm.py", line 318, in send2
    return self.start2(trans ,obj)
  File "/data1/www/htdocs/466/weknow/3/3party/pyfsm.py", line 284, in start2
    return self.current_state.enter2(self, obj)
  File "/data1/www/htdocs/466/weknow/3/3party/pyfsm.py", line 444, in enter2
    return self.func(task, obj)
  File "/data1/www/htdocs/466/weknow/3/web/mana4api.py", line 758, in papers
    p_list = "      ".join(["%s: %s\n"%(p[0], p[1]) for p in papers4tag])
UnicodeDecodeError: 'ascii' codec can't decode byte 0xe5 in position 9: ordinal not in range(128)

: [python - How do I check if a string is unicode or ascii? - Stack Over-
flow](http://stackoverflow.com/questions/4987327/how-do-i-check-if-a-string-is-unicode-or-ascii)

:
Google
<type 'str'>
```

<type 'unicode'>

- !, .json
- CLI
- KVDB !

...

1.5



- Zoom.Quiet
– Python / ""OBP

1.5.1

- api Pythonbottle_
- APIPython |
- SAE - -
- PythonBottleAPISAE -
- python - kevin -
- jeffkit/wechat
-

1.6

Creative Commons — - 2.5 — CC BY-SA 2.5

1.7

- 131010 Zoom.Quiet FSM CLI
- 130923 Zoom.Quiet GDG(: GDG-ZhuHai) ,
- 130918 Zoom.Quiet

1.8 ~ *reStructuredText*

1.8.1 Shinx

:

Sphinx - pymotwcn

- <http://code.google.com/p/pymotwcn/wiki/SphinxprojectHowto>

1.8.2

.

URL,,;URL;

- : <http://wiki.woodpecker.org.cn/moin/ObpLovelyPython>
- : <http://bit.ly/2QA425>
- ! <http://bit.ly> ;
- :,,URL;

```
def foo():
    print "Love Python, Love FreeDome"
    print "E,.0123456789,,. "
```

.. code-block:: :

```
.. code-block:: python
   :linenos:
   :emphasize-lines: 2
```

```
def foo():
    print "Love Python, Love FreeDome"
    print "E,.0123456789,,. "
```

:

```
1 def foo():
2     print "Love Python, Love FreeDome"
3     print "E,.0123456789,,. "
```

:

```
.. literalinclude:: example.py
   :language: python
```

```
:
@route('%s/' %ini.urlprefix)
def index():
    __urlog("INFO","idx++")
    return template('index.tpl',urlprefix=ini.urlprefix)
```

,:
No matter where you go, there you are.
—Buckaroo Banzai

:

Note: (~_~)

- This icon signifies a tip, suggestion, or general note.

Warning: (#_#)

- ...

See Also:

(^.^)

- ...

:

'''

:

PCS*

-
-
-
-
-

1.8.3 rST

- `..html`
- `rST` :
 -
 - :

```

index.rst
,:
.. _chapter2index:

,:
:ref: `<chapter2index>`
!

/

• ./
• rST :
  -
  -

.. _fig_2_4:
.. figure:: _static/figs/tmux-curl-test.png

    2-4

, 2-4 , “2-4”

```

```

$ dev_server.py
MySQL config not found
* Running on http://localhost:8080/
* Restarting with reloader
MySQL config not found
127.0.0.1 - - [25/Apr/2012 22:51:24] "GET / HTTP/1.1" 200 -

40 app = Bottle()
41
42 @app.route('/')
43 def hello():
44     return "Hello, world! - Bottle @ SAE 4 urisaok!{v12.0
1.12} for YY talking.."
45
46 @app.route('/chk/', method="POST")
47 def chk():
48     uri = request.forms.get('uri')
49     print uri

zoomq @ ZQ111216MBP in ~/Works/@SAE/urisaok on SAE4urisaok
$ curl -i localhost:8080
HTTP/1.0 200 OK
Content-Length: 67
Content-Type: text/html; charset=UTF-8
Server: Werkzeug/0.8.2 Python/2.7.1
Date: Wed, 25 Apr 2012 14:51:24 GMT

Hello, world! - Bottle @ SAE 4 urisaok!{king..

```

Figure 1.14: 2-4

/, html /(<sub> , <sup>), rST :

H\ :sub: '2' \ 0
E = mc\ :sup: '2'

:

H₂O

E = mc²

Note: :

”””

rST

```
=====
A      not A
=====
False  True
True   False
=====
```

,:

```
.. list-table::
   :widths: 15 10 30
   :header-rows: 1

   * - Treat
     - Quantity
     - Description
   * - Albatross
     - 2.99
     - On a stick!
   * - Crunchy Frog
     - 1.49
     - If we took the bones out, it wouldn't be
       crunchy, now would it?
   * - Gannet Ripple
     - 1.99
     - On a stick!
```

Table 1.1:

Treat	Quantity	Description
Albatross	2.99	On a stick!
Crunchy Frog	1.49	If we took the bones out, it wouldn't be crunchy, now would it?
Gannet Ripple	1.99	On a stick!

