



Uliweb 比较与实践 - 2013

李迎辉 (limodou@gmail.com)

2013/11/25

2013 年个人小结

Uliweb 与其它框架比较

Uliweb 的特殊功能介绍

2013 年个人小结

Uliweb 与其它框架比较

Uliweb 的特殊功能介绍

- Uliweb 发布到 0.2.2
- par 发布到 0.9.3 , parm 发布到 0.6
- Win32-Process-Watcher 发布
- 定制 alembic 并与 Uliweb 集成
- chatroom 实验使用 Uliweb+gevent-socketio 写的 websocket 的聊天程序, wshell-web shell
- desktopnotify js 库, 可以调浏览器 notification API
- 对 mmGrid 添加树, 优化多行表头, 无限翻页等功能
- 尝试 python3 , 创建 2to6 项目
- 参加何家胜组织的 codepark 编程公园活动, 组织了 3 次全天的编程活动

2013 年个人小结

Uliweb 与其它框架比较

Uliweb 的特殊功能介绍

怎么理解框架？我们需要框架吗？

（软件）框架是经验的总结

框架是把设计思想（模式）、开发习惯、常见功能组织在一起的一种 **

软件 **

使用框架要按框架的要求来组织代码，让框架来调

使用库则是由你来组织如何使用

使用框架表示你可能认同它的设计思想

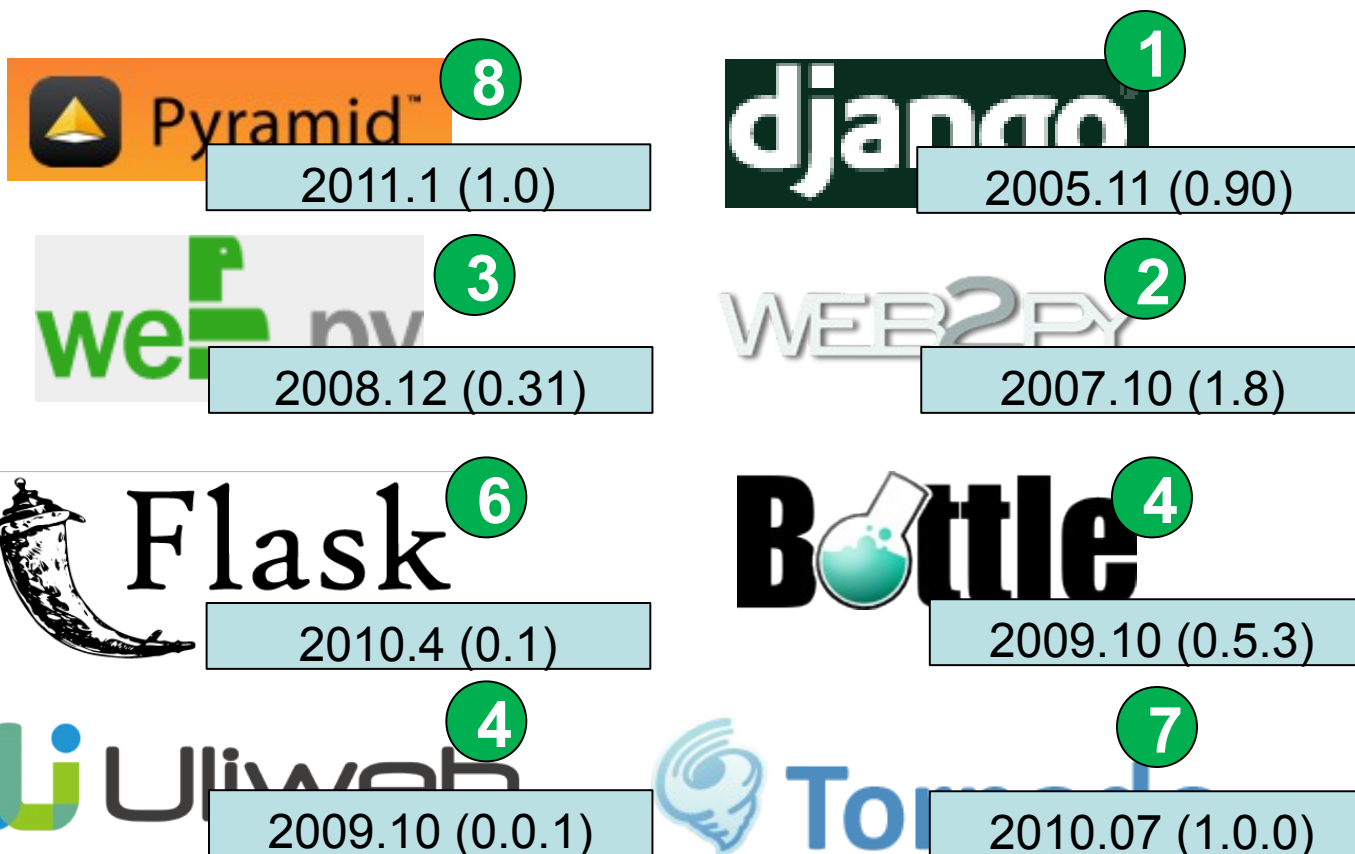
我们需要框架吗？



<http://wiki.woodpecker.org.cn/moin/FrameworksShow>

<http://simple-is-better.com/news/309>

第一个版本发布时间





v
s



组织方式



APP



自由，通过 Blueprint 来实现
APP 类似的功能



自由



APP

配置文件



settings.py



自由，多种格式



支持，要先定义，py 文件



settings.ini, local_setings.ini,
app 级别 settings.ini

配置文件语法



变量大写， Python 语法



变量大写， Python 语法， 支持类的写法



Python 语法



大小写都可以， Python 语法

程序结构的创建



makeproject



手工



手工



makeproject, makeapp

URL 的定义



集中式，urls.py，可以 include
，正则式



在程序中分散定义，使用
decorator，非正则



主程序中集中定义，正则式



在 views.py 中分散定义，使用
decorator，同 flask

URL 的反向获取



提供 `url_for`，按名字获取



`url_for`, 按名字获取



`reverse_url`，按名字



提供 `url_for`，`url_for_static`，
按名字获取

模板



支持 block ， 代码受限， 可以定义 filter, tag



jinja2 ， 类 django 模块， 代码限制较小



类 django ， 但是可以嵌入 python 表达式， 导入等



从 web2py 改造而来， 支持 block ， 可以直接嵌入 python ， pass 表示缩进结束

模板文件关联



在 `render_to_template` 中指定



在 `render_template` 中指定



`render` 中指定



自动套用函数，返回值为 `dict`
时或方法名或
`response.template` 中指定或
`expose` 中 `template`

View 方法



函数和类，一个类只处理一个 URL，需要从基类继承



函数，类 (需要从基类继承)，没有对类的 decorator 的修饰函数，一个类只处理一个 URL



类，一个类只处理一个 URL，需要从基类继承



函数和类，一个类可以处理多个 URL。可以在类方法上加 expose, 不需要特殊基类

常用对象的引用



request 是作为第一个参数



导入



绑定在类上



自动注入和导入

Form 处理



内置



不提供, 需要使用 WTForms



不提供



内置

ORM



内置，自己实现



不直接提供，通过 Flask-SQLAlchemy



不提供



内置，基于 SQLAlchemy 实现

数据库迁移

The Django logo, featuring the word "django" in a white, lowercase, sans-serif font on a dark green rectangular background.

间接通过 south

The Flask logo, featuring a stylized illustration of a flask on the left and the word "Flask" in a large, black, serif font on the right. Below "Flask" is the tagline "web development, one drop at a time" in a smaller, black, sans-serif font.

不提供

The Tornado logo, featuring a stylized blue icon of a tornado on the left and the word "Tornado" in a blue, sans-serif font on the right.

不提供

The Uliweb logo, featuring a stylized "U" icon composed of blue and green segments on the left and the word "Uliweb" in a black, sans-serif font on the right.

间接，集成了 alembic



支持，通过 xgettext 提取



支持，通过 babel 处理



支持，通过 xgettext 提取



支持，通过修改 pygettext.py

命令行工具



提供，可以按 **app** 来自定义



第三方



无



提供，可以按 **app** 来自定义

异步处理支持



第三方



第三方



内置



第三方

websocket

django

第三方

Flask
web development,
one drop at a time

第三方

Tornado

内置

Uliweb

第三方

比较总结（一）

比较内容	Django	Flask	Tornado	Uliweb
组织方式	APP	Free/Blueprint		APP
配置文件	py	多种	py	ini
配置文件语法	py	py	py	py
程序结构创建	cmd			cmd
URL 定义	集中	分散	集中	分散
URL 反向获取	√	√	√	√
模板	继承，受限	继承，受限	继承，自由	继承，自由
模板调用	手动	手动	手动	套用 / 手动
View	函数 / 类	函数 / 类	类	函数 / 类
Url/View-Class	1:1	1:1	1:1	n:1
request 等对象	参数	导入	绑定类	注入 / 导入
Form	√			√
ORM	√			√

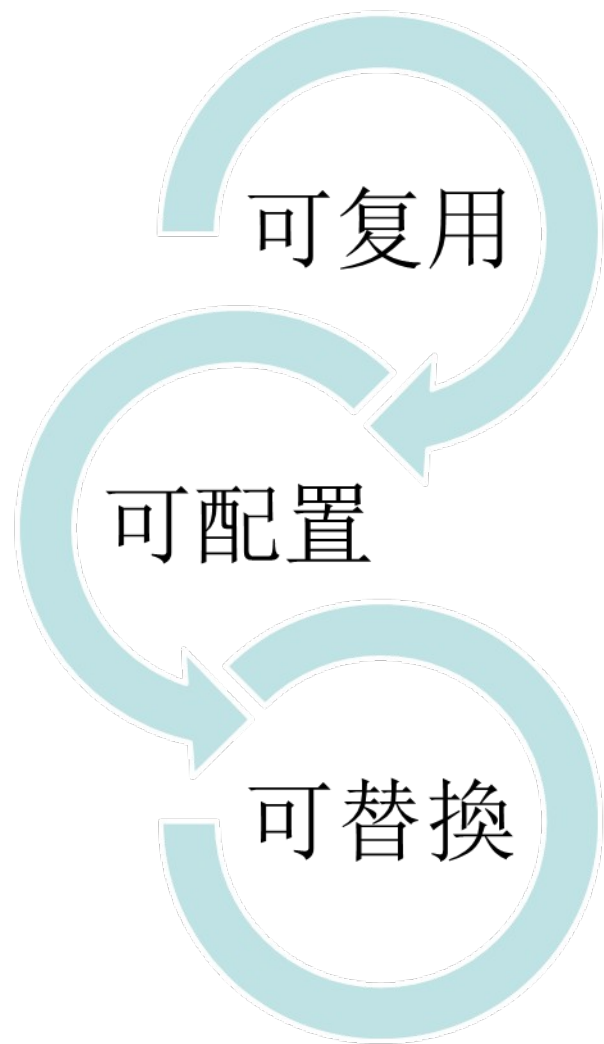
比较总结（二）

比较内容	Django	Flask	Tornado	Uliweb
数据库迁移	-/south			-/Alembic
i18n	√	√	√	√
命令行工具	√			√
异步处理			√	
websocket			√	

2013 年个人小结

Uliweb 与其它框架比较

Uliweb 的特殊功能介绍



API

- 如何简单提供统一的 **API** 的定义和使用方式？

functions

```
[FUNCTIONS]  
get_model = 'uliweb.orm.get_model'  
get_object = 'uliweb.orm.get_object'  
set_echo = 'uliweb.orm.set_echo'
```

```
from uliweb import functions
```

```
User = functions.get_model('user')
```

Model

- Model 能替换吗？
- Model 变了怎么方便处理？
- 如何方便调试？

配置化

```
[MODELS]  
wikiPage = 'wiki.models.WikiPage'  
wikichangeset = 'wiki.models.WikiChangeSet'
```

get_model()

```
from uliweb import functions
```

```
User = functions.get_model('user')
```

命令行

uliweb syncdb -v

```
[default] Creating [1/40, blog] blog_category...EXISTED
[default] Creating [2/40, blog] blog...EXISTED
[default] Creating [3/40, blog] blog_blog_tag_tags...EXISTED
[default] Creating [4/40, blog] blog_tag...EXISTED
[default] Creating [5/40, classes] class_teacher...EXISTED
[default] Creating [6/40, classes] class_info...EXISTED
```

命令行

```
uliweb dump [appname,...]  
uliweb dumptable tablename[, tablename]  
uliweb reset [appname,...]  
uliewb resetttable tablename[, tablename]  
uliweb load [appname,...]  
uliewb loadtable tablename[, tablename]
```

命令行

uliweb sql
uliweb sqltable




```
CREATE TABLE forumcategory (  
    name VARCHAR(100),  
    description TEXT,  
    ordering INTEGER,  
    created_on DATETIME,  
    updated_on DATETIME,  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    PRIMARY KEY (id)  
) CHARSET=utf8;
```


命令行

uliweb sqlhtml

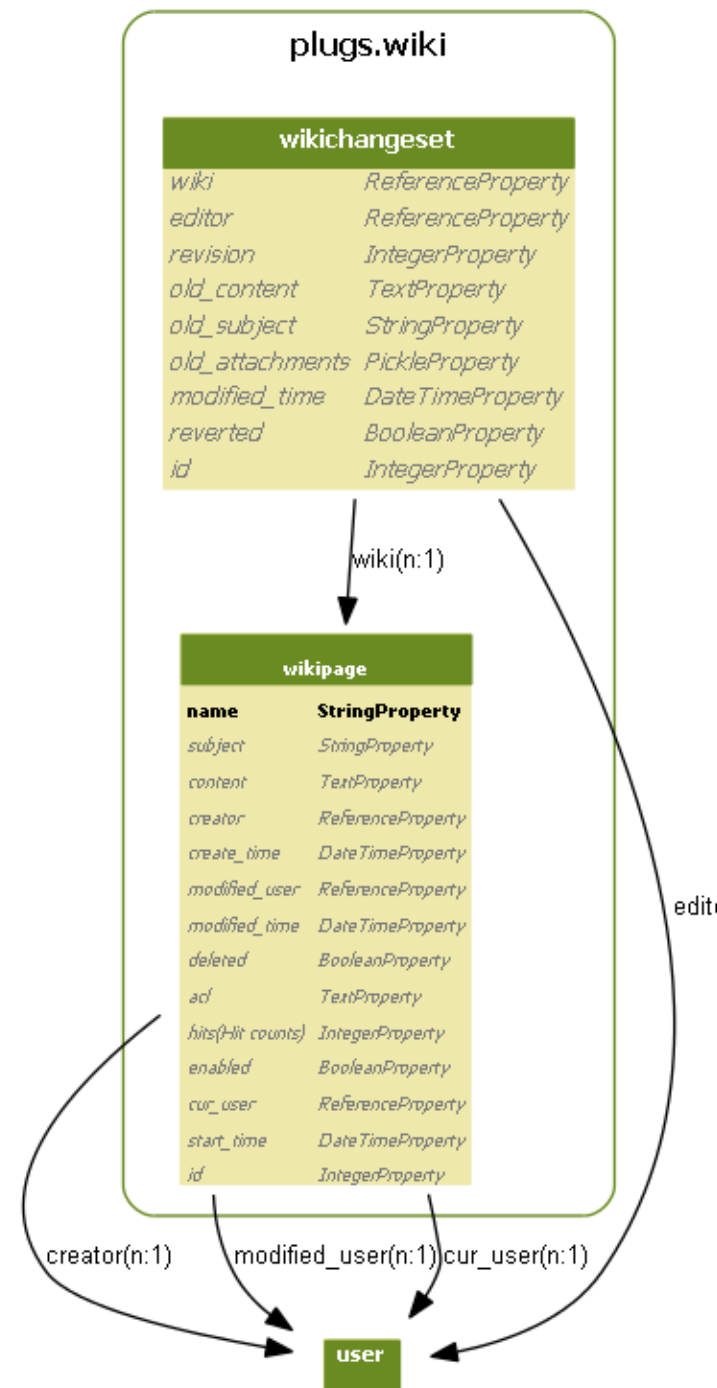
Database Tables Sheet

blog	blog blog
<ul style="list-style-type: none">blog_categoryblogblog_tag	
classes	
<ul style="list-style-type: none">class_teacherclass_infoclass_issueclass_studyrecordclassclass_category	

P	Display	Fieldname	Data Type	Can be Null	Relation
	id	id	INTEGER		
	Author	author	INTEGER	✓	 user
	Category	category	INTEGER	✓	 blog_category
	Content	content	TEXT	✓	
	Enable Comment	enable_comment	BOOLEAN	✓	

命令行

uliweb sqldot



Alambic 集成

uliweb alembic init

uliweb alembic diff

uliweb alembic upgrade

```
D:\project\mywork\uliwebzone>uliweb alembic diff -f
INFO [alembic.migration] Context impl MySQLImpl.
INFO [alembic.migration] Will assume non-transactional DDL.
INFO [alembic.autogenerate.compare] Detected removed column 'forum.private'
INFO [alembic.autogenerate.compare] Detected server default changed from sa.text('0') to '0' on column 'tutorials.comments_count'
INFO [alembic.autogenerate.compare] Detected server default changed from sa.text('0') to '0' on column 'tutorials_chapters.chars_count'
INFO [alembic.autogenerate.compare] Detected server default changed from sa.text('0') to '0' on column 'tutorials_chapters.comments_count'
Generating
D:\project\mywork\uliwebzone\alembic\default\versions\25370ab1feb8_.py...done
```

调试

set_echo

```
functions.set_echo(True, time=0.001, explain=True)
===>>> (d:/project/mywork/uliweb-git/uliweb/utils/generic:1916:objects)
SELECT tutorials.title, tutorials.creator, tutorials.create_date, ...,
tutorials.comments_count, tutorials.id
FROM tutorials
WHERE tutorials.deleted = false ORDER BY tutorials.modified_date DESC
LIMIT 0, 10
----
Explain: id=1L, select_type=u'SIMPLE', table=u'tutorials', type=u'ALL',
possible_keys=None, key=None, key_len=None, ref=None, rows=8L,
Extra=u'Using where; Using filesort',
===<<< time used 0.024000s
```

调试

SQLMonitorMiddle

```
[MIDDLEWARES]
```

```
sqlmonitor =
```

```
'uliweb.contrib.orm.middle_sqlmonitor.SQLMonitorMiddle'
```

```
[ORM]
```

```
SQL_MONITOR = True
```

```
===== sql execution count 10 </tutorial> =====
```

```
SELECT tutorials.title, tutorials.creator, tutorials.create_date,  
t...      1   0.100
```

```
SELECT count(tutorials.id) AS tbl_row_count FROM tutorials WHERE  
tu...     2   0.002
```

```
SELECT user.username, user.nickname, user.email, user.password,  
use...    7   0.012
```

模板

- 如何方便引用外部资源，不想写 `<script>` 或 `<link>`
- 在模板继承时，如何处理新增资源
- 模板如何调试？继承，查找？

定义 1

app/template_plugins/xxxx.py

```
def call(app, var, env, version=None):  
    from uliweb import settings  
  
    a = []  
    version = version or  
settings.UI_CONFIG.angularjs_version  
    a.append('angularjs/%s/angular-%s.min.js' % (version,  
version))  
    a.append('jsutils/json2.js')  
    return {'toplinks':a}
```

定义 2

settings.ini

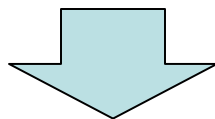
```
[TEMPLATE_USE]
avalon = {'toplinks': ['avalon/avalon.js',
'avalon/avalon_init.js']}
```


使用

use, link

`'uliweb.contrib.template'`

```
{{use "avalon"}}  
{{link "test.js"}}
```



```
<script type="text/javascript"  
src="/static/avalon/avalon.js?ver=18"></script>  
<script type="text/javascript"  
src="/static/avalon/avalon_init.js?ver=18"></script>
```

模板继承的处理

```
#layout.html
<html>
  <head>
    {{use "bootstrap"}}
  </head>
  <body>{{block content}}{{end}}</body>
</html>
```

```
#layout.html
{{extend "layout.html"}}
{{block content}}
  {{use "avalon"}}
{{end}}
```

处理结果

```
#layout.html
<html>
  <head>
    {{use "bootstrap"}}
    {{use "avalon"}}
  </head>
  <body>{{block content}}{{end}}</body>
</html>
```

命令行调式

uliweb find -t template

```
D:\project\cc\ctasks-master>uliweb find -t index.html  
apps/newportal/templates/index.html  
apps/myportal/templates/index.html  
apps/Portal/templates/index.html
```

命令行调式

uliweb find -t template --tree

```
D:\project\cc\ctasks-master>uliweb find -t index.html --tree
apps/newportal/templates/index.html

----- Tree -----
    apps/theme/templates/theme/skeleton.html
        (extend) apps/theme/templates/theme/zebra.html
-----> (extend) apps/newportal/templates/index.html
        (include) apps/theme/templates/include/inc_userinfo.html

(include) apps/theme/templates/include/inc_checklogin.html
        (include) apps/theme/templates/theme/utils.html
        (include) d:/project/mywork/plugs-
git/plugs/ui/jquery/pnotify/templates/
inc_show_flashes.html
```

命令行调式

uliweb find -t template-block -with-filename

```
----- Blocks -----  
title      (apps/theme/templates/theme/zebra.html)  
meta       (apps/theme/templates/theme/skeleton.html)  
_css       (apps/newportal/templates/index.html)  
body_class (apps/newportal/templates/index.html)  
before_header (apps/theme/templates/theme/skeleton.html)  
header      (apps/theme/templates/theme/zebra.html)  
    header_extra_class (apps/theme/templates/theme/zebra.html)
```

CSS, JS 合并

初始

```
<link href="/static/bootstrap/2.2.0/bootstrap.min.css?ver=18"/>
<script src="/static/bootstrap/2.2.0/js/bootstrap.min.js?ver=18"></script>
<link href="/static/jquery/ui/css/redmond/jquery-ui-1.8.16.custom.css?
ver=18"/>
<script src="/static/jquery/ui/js/jquery-ui-1.8.16.custom.min.js?
ver=18"></script>
<script src="/static/jquery/ui/js/jquery.ui.datepicker.zh.js?ver=18"></script>
<link href="/static/theme/blove/index.css?ver=18"/>
<script src="/static/jqutils/jquery.cookie.js?ver=18"></script>
<link href="/static/poshytip/tip-twitter/tip-twitter.css?ver=18"/>
```

CSS, JS 合并

配置

```
[STATIC_COMBINE]
1 = ['jquery/jquery-1.7.2.min.js', 'jsutils/json2.js']
#jqutls
2 = ['jquery/ui/js/jquery-ui-1.8.16.custom.min.js',
'jquery/ui/js/jquery.ui.datepicker.zh.js',
'jsmenu/menu.js',
'poshytip/jquery.poshytip.js',
'jqutils/jqrselect.js',
'jqutils/jqutils.js',
'jqutils/jquery.hotkeys.js',
'jqutils/jquery.form.js',
'pnotify/1.2.0/jquery.pnotify.min.js',
]
```


CSS, JS 合并

uliweb exportstatic -auto static

```
<link href="/static/_cmb_3ab42e4ed430d339f6390330aa3e77b7.css?ver=15"/>  
<script src="/static/_cmb_9ed3a8d3597e13abdd529ea9d4e6270f.js?  
ver=15"></script>  
<link href="/static/_cmb_88466a4a9313f6857d61aa632abf5054.css?ver=15"/>  
<script src="/static/_cmb_c8a04bdfa7d2195d5f05c76a87a8767a.js?  
ver=15"></script>
```

Generic

- 如何快速处理 CRUD

操作流程定义为类

AddView

EditView

DetailView

DeleteView

ListView

...

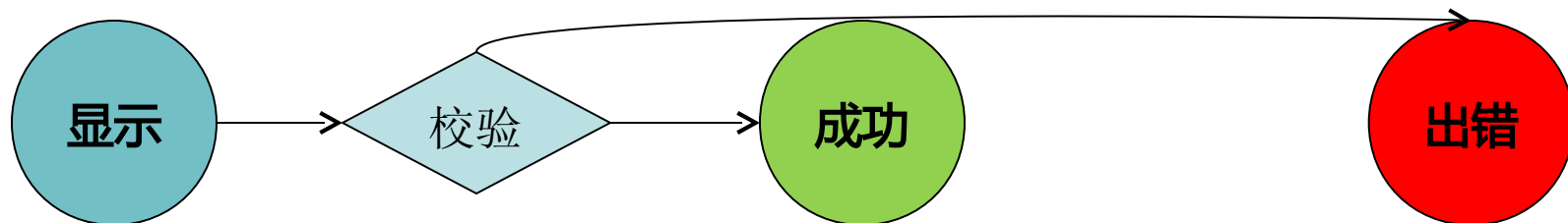
操作流程定义为类

AddView

```
@expose('/blog')
class BlogAdd(object):
    def add(self):
        Blog = functions.get_model('blog')
    def get_url(id):
        return url_for(self.__class__.list)
    view = functions.AddView(Blog, ok_url=get_url)
    return view.run()
```

操作流程定义为类

```
form = make_form(Model) # 根据 Model 自动生成 Form
if method == 'GET':
    return {'form': form} # 显示页面
else: # POST
    if form.validate(request.values): # 表单校验
        # 成功
        save() # 保存数据
        return redirect(ok_url) # 成功后跳转
    else:
        return {'form': form} # 带有出错信息返回
```



通过参数与回调来进行控制

```
class AddView(object):
    def __init__(self, model, ok_url=None, ok_template=None, form=None,
        success_msg=None, fail_msg=None, use_flash=True,
        data=None, default_data=None, fields=None, form_cls=None,
        form_args=None, static_fields=None, hidden_fields=None,
        pre_save=None, post_save=None, post_created_form=None,
        layout=None, file_replace=True, template_data=None,
        success_data=None, fail_data=None, meta='AddForm',
        get_form_field=None, post_fail=None, types_convert_map=None,
        fields_convert_map=None, json_func=None,
        file_convert=True, upload_to=None, upload_to_sub=None,
        fileserving_config='UPLOAD', protect=False, protect_field_name=None):
```

自动生成代码框架

```
generic_app_blog>uliweb generic
Appname:blog
Table Name:blog
Creation Theme([a]angularjs, [h]html), [e]sayui)[a]:h
View Class Name [BlogView]:
Save views to [views_blog.py]:
Class View URL prefix [/blog]:
Enable pagination(Y/n/q) [Y]:
Enable query(Y/n/q) [Y]:n
Enable download(Y/n/q) [Y]:n
Add View using popup(Y/n/q) [Y]:n
Add View using ajax(Y/n/q) [Y]:n
Edit View using popup(Y/n/q) [Y]:n
Edit View using ajax(Y/n/q) [Y]:n
Delete View using ajax(Y/n/q) [Y]:
```

生成的目录及文件

```
├─apps
│   └─blog
│       ├──static
│       ├──templates
│       │   └─BlogView
│       │       ├──add.html
│       │       ├──edit.html
│       │       ├──list.html
│       │       └─view.html
│       └─views_blog.py
```


欢迎使用 Uliweb

谢谢

邮件列表 <https://groups.google.com/forum/#!forum/uliweb>
论坛 <http://uliweb.clkg.org>
文档 <http://limodou.github.io/uliweb-doc/>
QQ 群 162487035

邮箱: limodou@gmail.com
微博: <http://weibo.com/u/1689940061>