web2py

Massimo Di Pierro

Associate Professor

School of Computing

DePaul University, Chicago

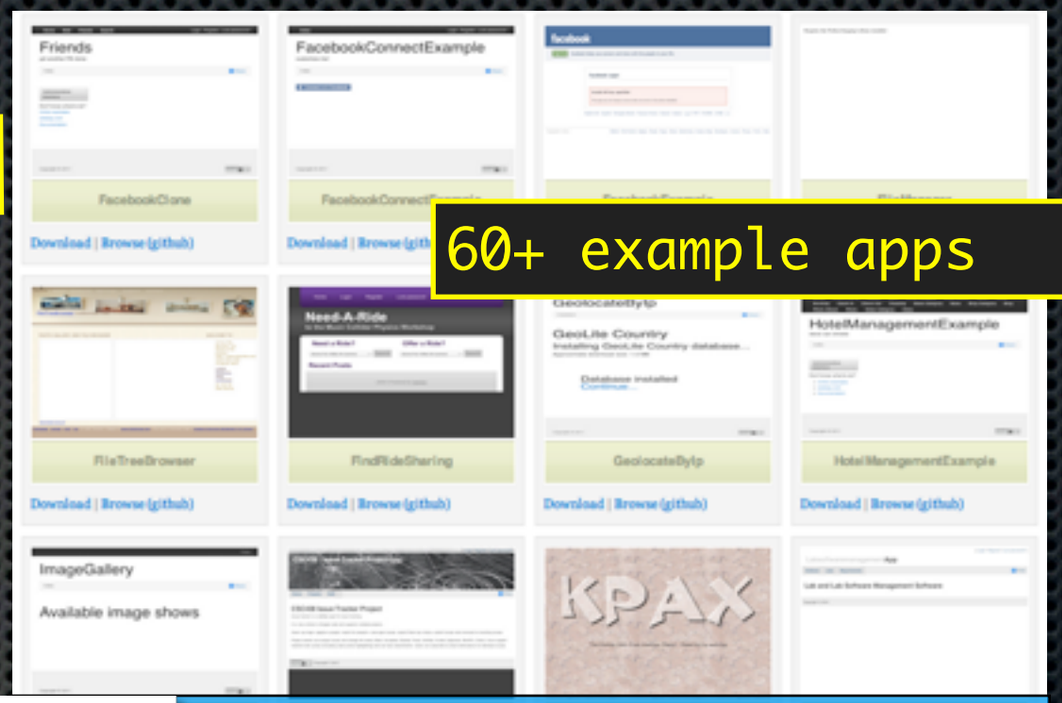
Contributors

Resources

books

videos

web2py.com



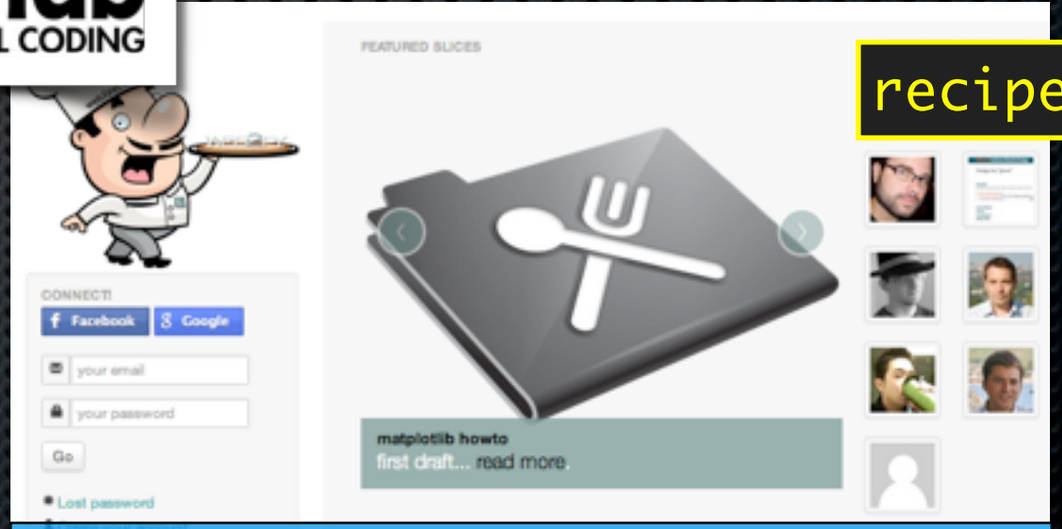
60+ example apps

web2py.com/appliances

github
SOCIAL CODING

plugins

dev.s-cubism.com/web2py_plugins



recipes

http://www.web2pyslices.com
(built with web2py-based social network software: http://movu.ca/)

web2py Goals

- Make web development easy and secure
- ... by providing a simple consistent syntax
- ... everything should have a default
- ... all tools in one box
- ... including a web based IDE
- ... takes care of all management issues (no shell)

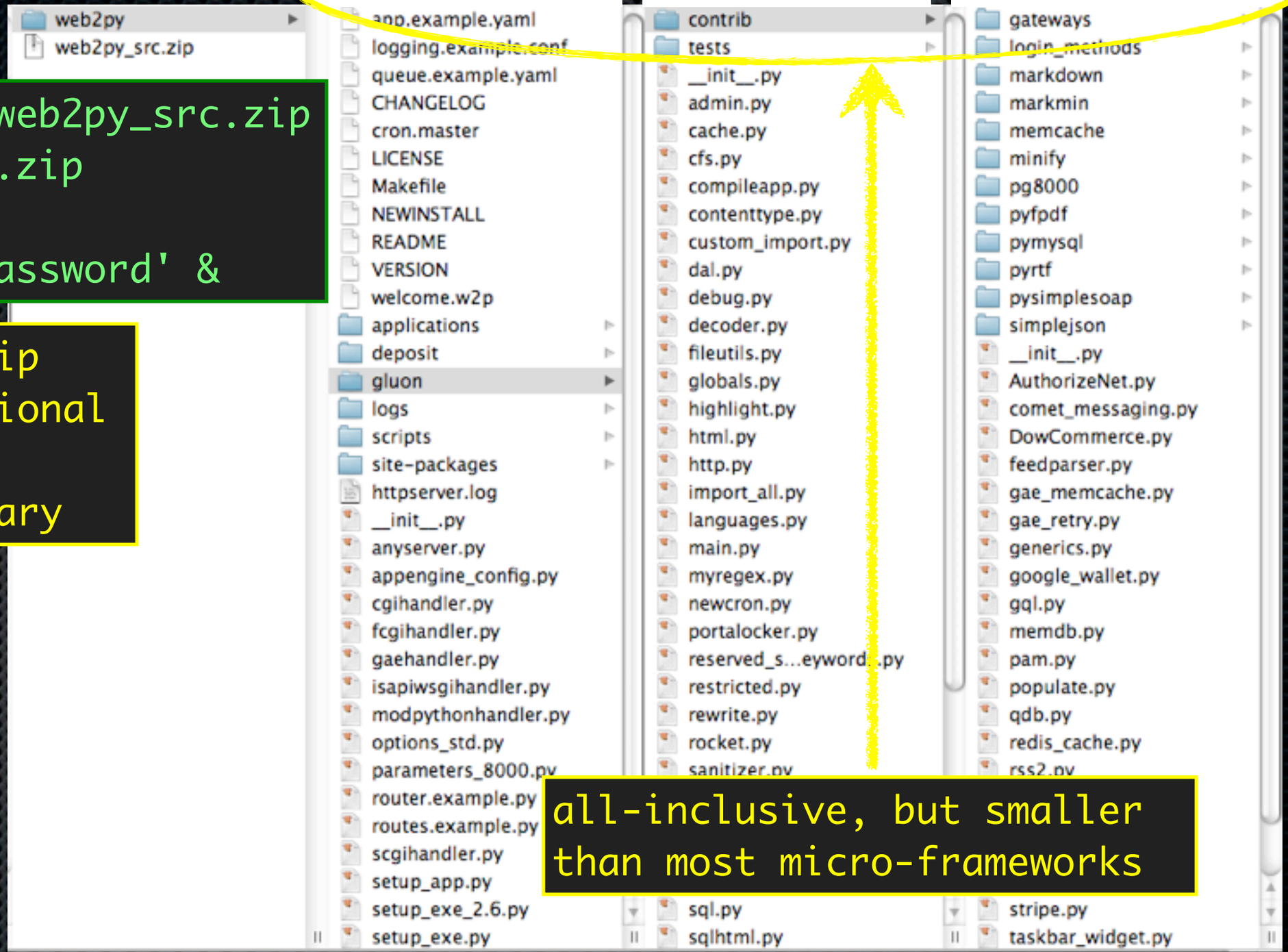
```
wget http://.../web2py_src.zip
unzip web2py_src.zip
cd web2py
web2py.py -a 'apassword' &
```

```
download and unzip
installation optional
no configuration
no coding necessary
```

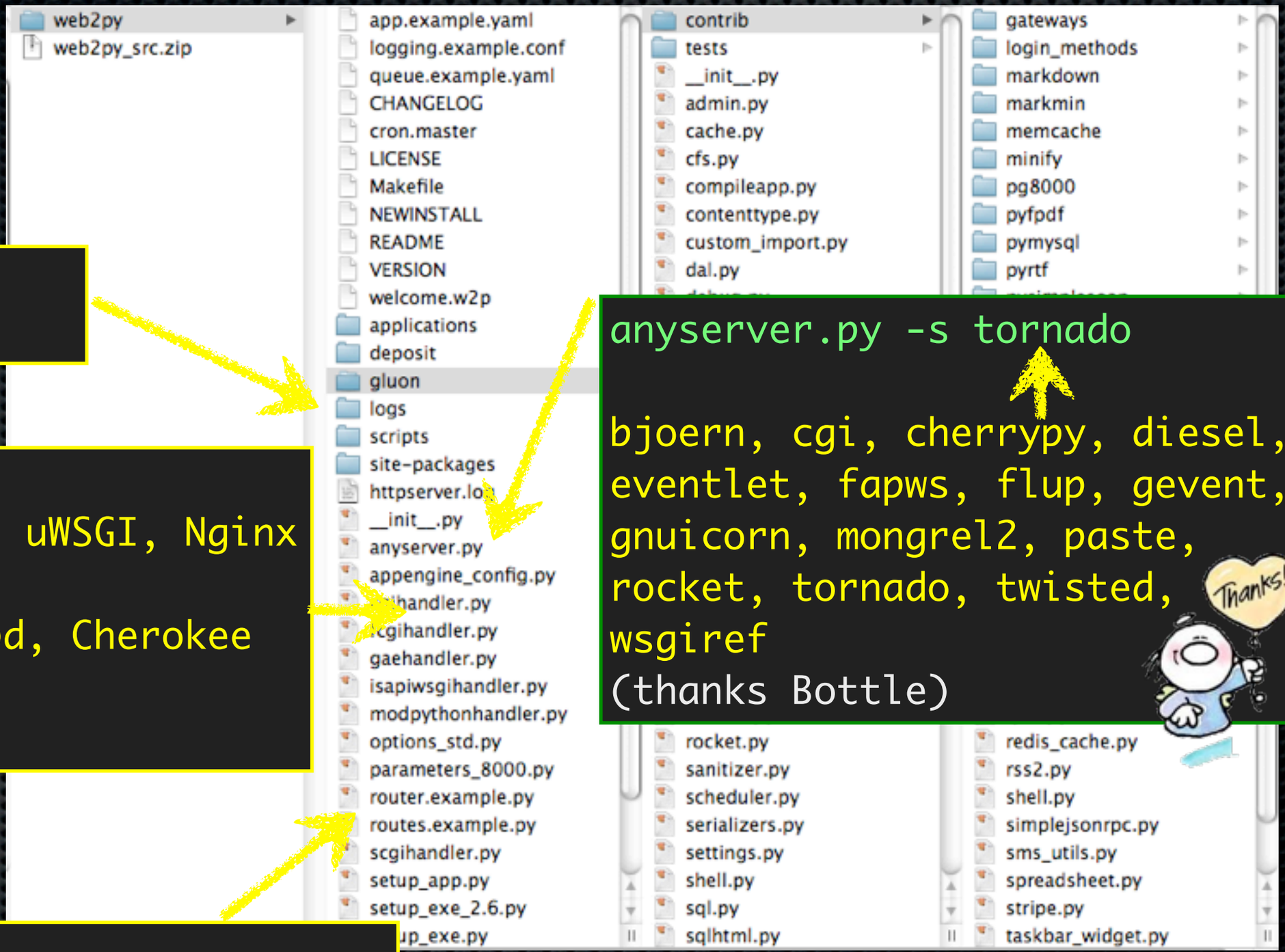
web2py/ 0.1MB

gluon/ 1.3MB

contrib/ 2.0MB




```
all-inclusive, but smaller
than most micro-frameworks
```



*.log:
log files

adapters:
- WSGI: Apache, uWSGI, Nginx
- CGI
- FCGI: Lighttpd, Cherokee
- GAE
- ISAPI: IIS

anyserver.py -s tornado

bjoern, cgi, cherrypy, diesel, eventlet, fapws, flup, gevent, gnuicorn, mongrel2, paste, rocket, tornado, twisted, wsgiref
(thanks Bottle)

routes.py:
like urls.py but optional

dal.py: Database Abstraction Layer
single file / no dependencies / framework agnostic

supports: sqlite, postgresql, mysql, db2, firebird, mssql, google datastore, interbase, ingres, imap, ~sapdb, ~cubrid, ~teradata, ~mongodb

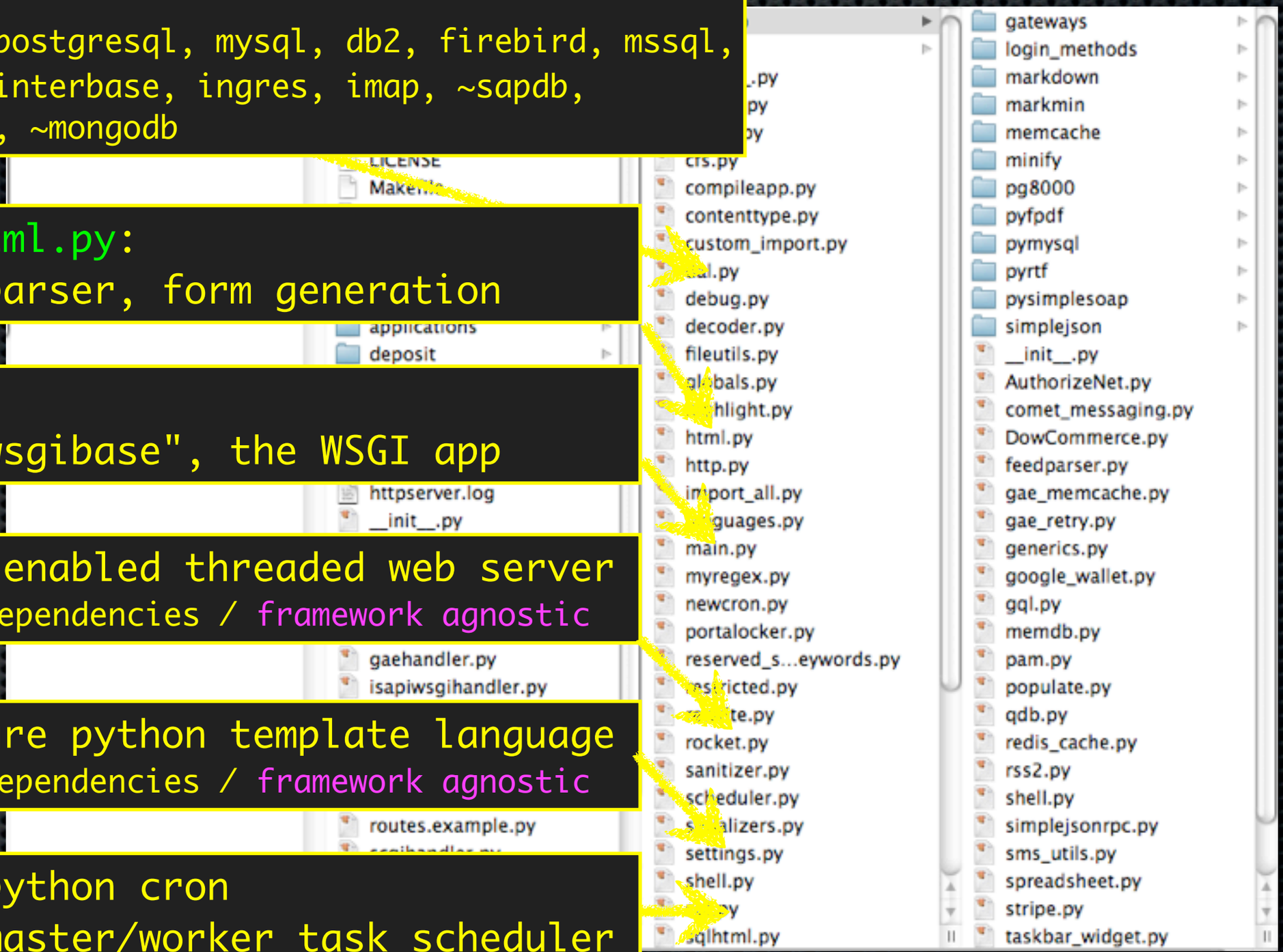
html.py & sqlhtml.py:
helpers, html parser, form generation

main.py:
contains "def wsgibase", the WSGI app

rocket.py: ssl-enabled threaded web server
single file / no dependencies / framework agnostic

template.py: pure python template language
single file / no dependencies / framework agnostic

cron.py: pure python cron
scheduler.py: master/worker task scheduler



Document processing:
pyfpdf (html to pdf), pyrtf (generates Rich Text Format)
markdown, MARKMIN (like markdown but better ;-)

Protocols:
rss, feedparser, json, PySimpleSOAP, ...

Caching Adapters:
Redis, Memcache, GAE Memcache

Payment processing:
Authorize.net, Stripe.com, DowCommerce.com

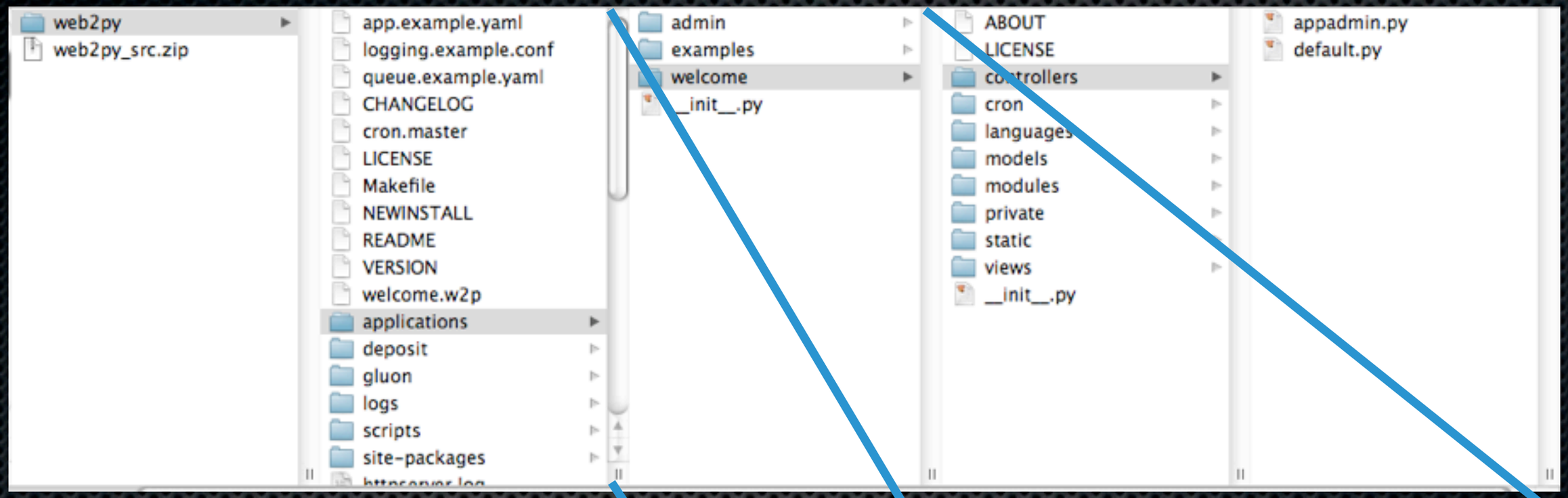
Login Methods:
Oauth 1.0, Oatuh 2.0, OpenID, CAS (consumer and provider),
PAM, LDAP, Janrain, x509, LinkedIn, Loginza, Janrain,
gmail, Dropbox, motp

Drivers:
pyMySQL, pg8000 (for postgresql)

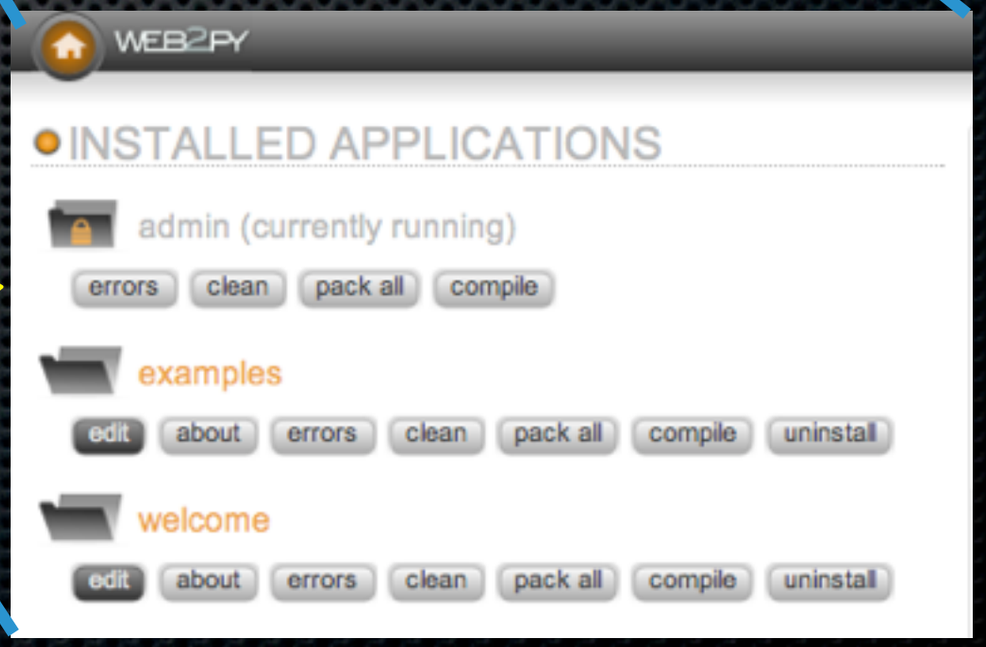
other stuff...

contrib/ 2.0MB

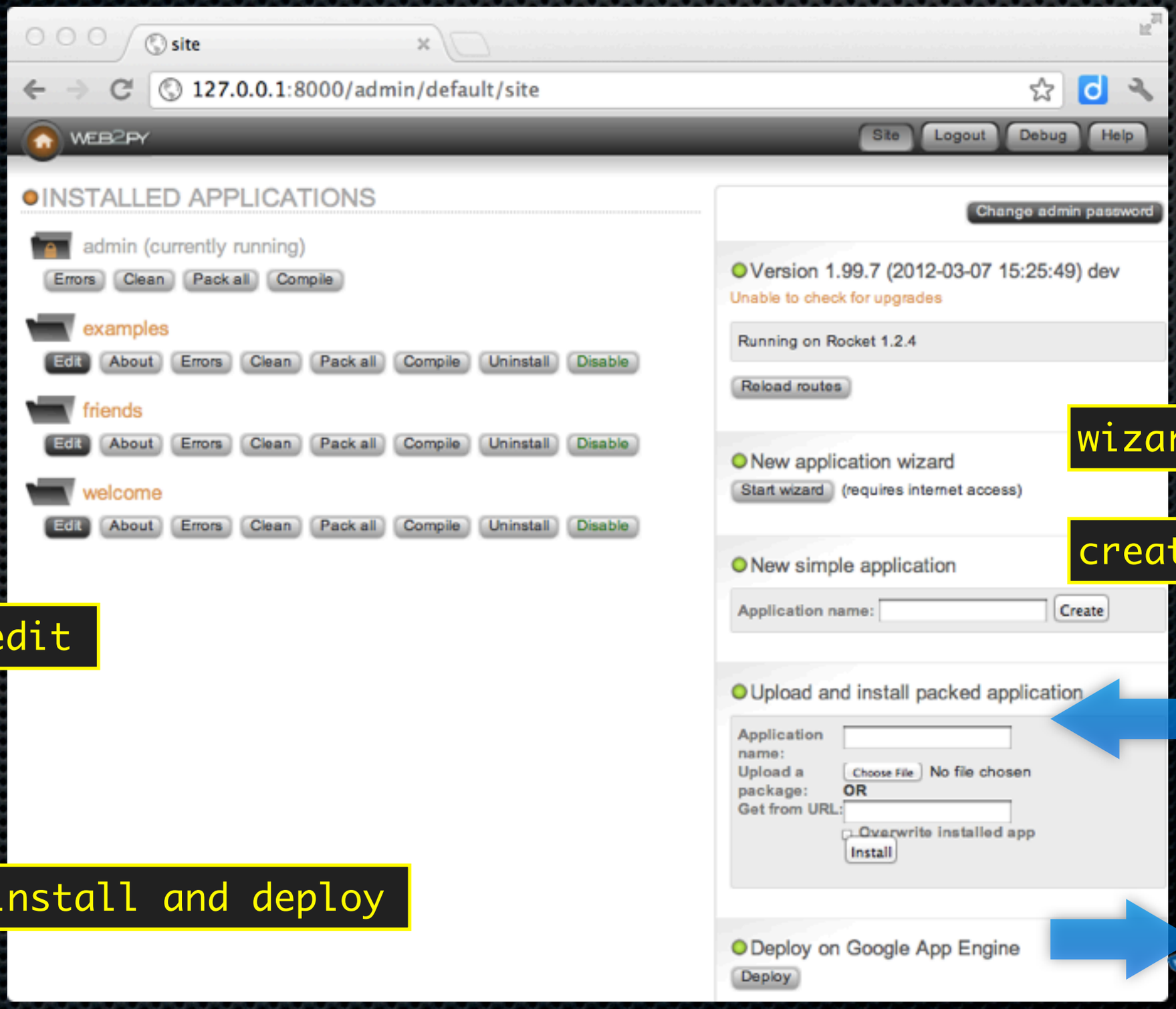
Multi-Project




```
admin:     the web based IDE
examples:  clone of the web site
welcome:   the scalffolding app
...        one web2py can run
           many projects
           without conflicts
```



Web Based IDE



wizard

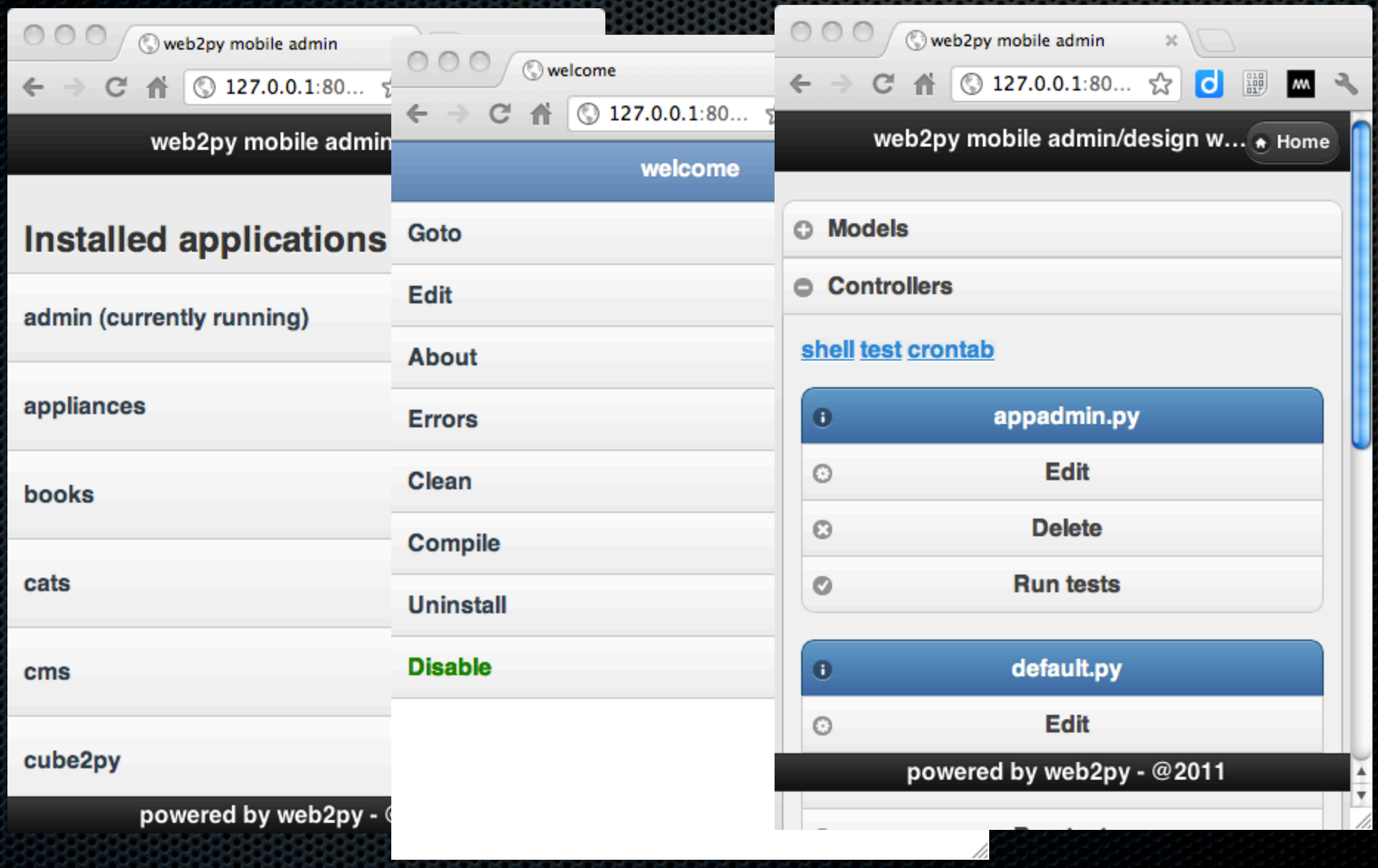
create

manage/edit

remote install and deploy

Web Based IDE

mobile version (based on jQuery mobile)



Web Translation

WEB2PY

EDITING LANGUAGE FILE "WELCOME/LANGUAGES/

ORIGINAL/TRANSLATION

"update" is an optional expression like "field1='newvalue'". You cannot update or delete the results of a

"update" è un'espressione opzionale come "campo1='nuovo valore'".
Non si può fare "update" o "delete" dei risultati di un JOIN

delete

%s rows deleted

%s righe ("record") cancellate

delete

%s rows updated

%s righe ("record") modificate

delete

%Y-%m-%d

%d/%m/%Y

delete

%Y-%m-%d %H:%M:%S

%d/%m/%Y %H:%M:%S

delete

Administrative interface

Interfaccia amministrativa

delete

Tickets

no distiction debug vs production

logs all errors and issues tickets to users by default

WEB2PY

ERROR LOGS FOR "A1"

check all | uncheck all | delete all checked

Click row to expand traceback

lists by ticket

groups common errors

Delete	Count	File	Error
☐	3	default.py	ZeroDivisionError: integer division or modulo by zero + details
☐	1	default.py	NameError: name 'asasdf' is not defined + details

WEB2PY

ERROR TICKET FOR "A1"

web debugger

d06-fbab-4518-beeb-8f3cda692432

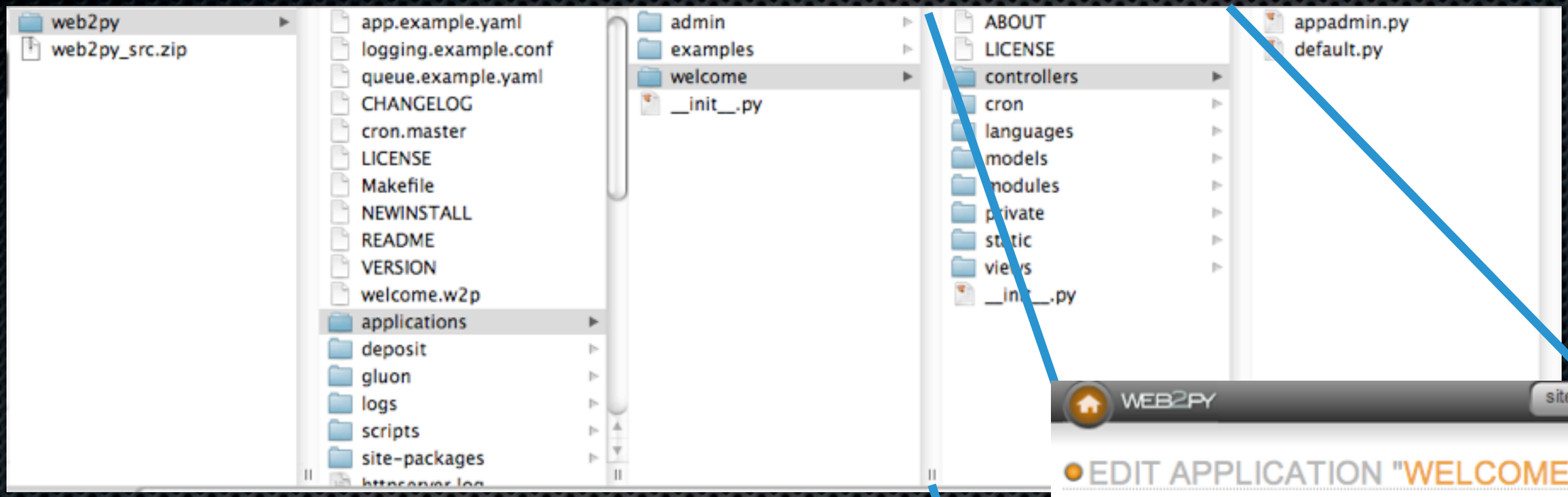
04 09:10:10)

neworks/Python.framework/Versions/2.5/Resources/Python.app/

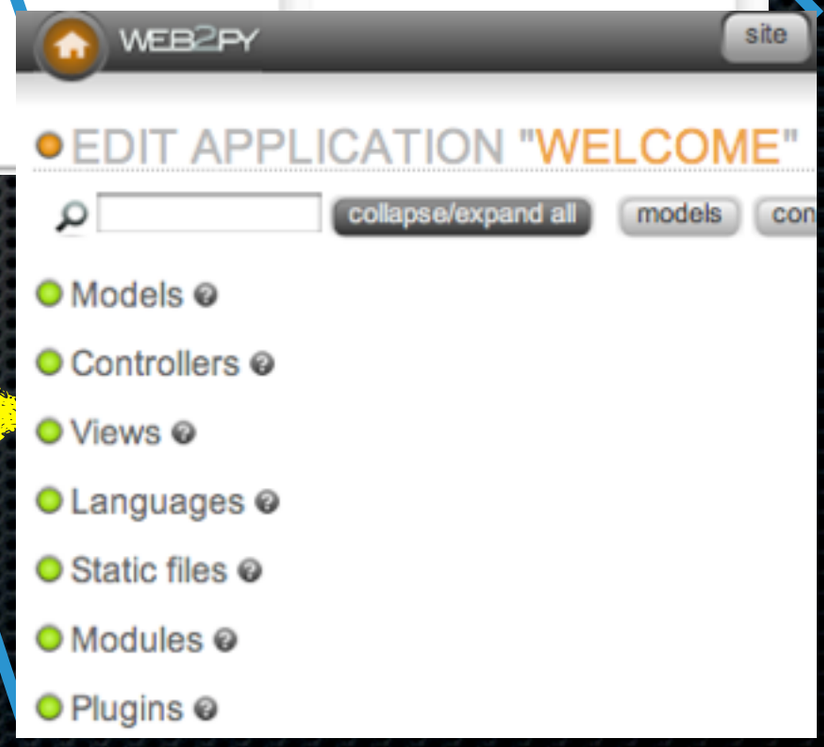
traceback

```
1.  Traceback (most recent call last):
2.    File "/Users/mdipierro/web2py/gluon/restricted.py", line 192, in restricted
3.      exec ccode in environment
4.    File "/Users/mdipierro/web2py/applications/a1/controllers/default.py", line 12, in <modul
5.      1/0
6.  ZeroDivisionError: integer division or modulo by zero
7.
```

Model-View-Controller




```
cd applications
mkdir friends
cp -r welcome/* friends
emacs friends/models/db.py
emacs friends/controllers/default.py
emacs friends/views/default/index.html
```



shell access or web access or both (no metadata, no restart)

Overall Design

Most Python Frameworks

your program

import

framework libraries

runme.py

```
import bottle

@bottle.route('/index')
def index():
    return 'hello world'

bottle.run()
```

web2py

framework

execute

your program

```
def index():
    return "hello world"
```

web2py.py

cron

scheduler

```
from gluon.rocket import Rocket
from gluon.main import wsgibase

Rocket(app_info =
  {'wsgi_app':wsgibase}).start()
```

+ routes + session + cache + ...

"do-not-repeat-yourself" > "explicit is better than implicit"

Conventions

http://.../friends/main/index

```
web2py/
   applications/
      friends/
         models/

         controllers/main.py:

            def index():
               return "hello world"

         views/
```

similar to Rails
but multi-app

Conventions

http://.../friends/main/index

web2py/
applications/
friends/
models/

controllers/main.py:

```
def index():
    message = "hello world"
    return dict(message=message)
```

views/main/index.html:

```
{{extend 'layout.html'}}

<h1>Example</h1>
{{=message}}
```

default layout.html is mobile friendly, based on flexible grid (skeleton.css)

Conventions

http://.../friends/main/index

web2py/
applications/
friends/

models/friends.py:

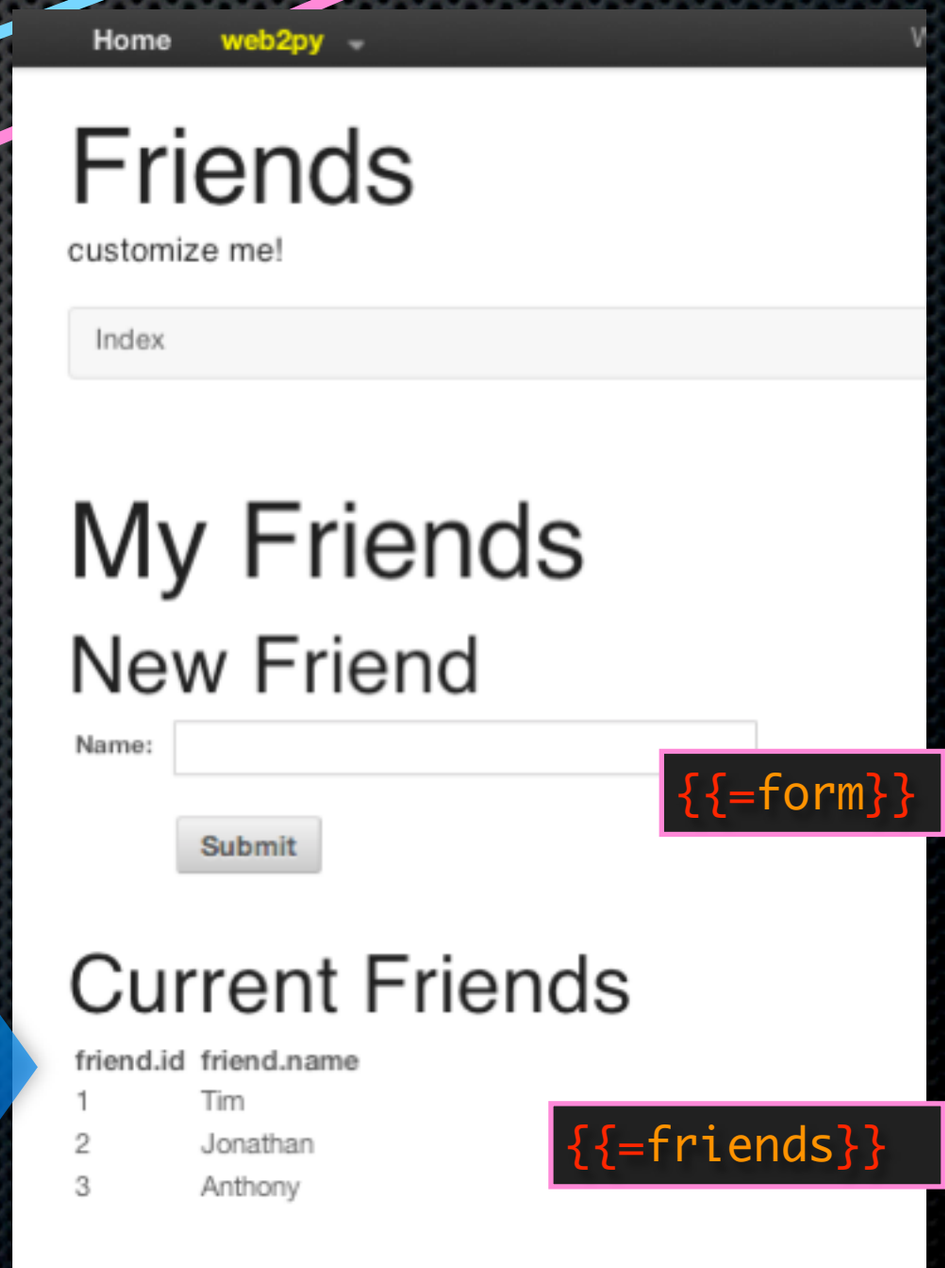
```
db.define_table('friend',Field('name'))
```

controllers/main.py:

```
def index():
    form = SQLFORM(db.friend).process()
    friends = db(db.friend).select()
    return locals()
```

views/main/index.html:

```
{{extend 'layout.html'}}
<h1>{{=T('My Friends')}}</h1>
<h2>New Friend</h2>
{{=form}}
<h2>Current Friends</h2>
{{=friends}}
```



Controllers

Django (view in MTV)

```
def get_friend(request):
    friend_id = request.GET['id']
    friend = Friend.objects.get(pk=friend_id)
    output = friend.name
    return HttpResponse(output)
```

web2py (controller in MVC)

```
def get_friend():
    friend_id = request.get_vars['id']
    friend = Friend(friend_id)
    output = friend.name
    return dict(output=output)
```

Routes

Django (url.py)

```
urlpatterns = patterns('',
  (r'^friends/$', 'friends.views.index'),
  (r'^friends/(\d{4})/$', 'friends.views.read'),
)
```

web2py (routes.py)

```
routes_in = [
  (r'friends/', '/friends/default/index'),
  (r'friends/(\d{4})', '/friends/default/get_friend?id=\1')
  (r'friends/$id', /friends/default/get_friend?id=\1'),
  (r'127.0.0.*:http://domain.com friends/(\d{4})','/otherapp')
]

routes_out = [...]

routes_onerror = [
  (r'friends/400', r'/friends/default/login'),
  (r'*/*', r'/friends/static/fail.html')]
```

always optional

reverse routes

routes on error

Models (DAL not ORM)

Django

```
class Friend(models.Model):
    name      = models.CharField(max_length=255,null=False)
    email     = models.EmailField()
    info      = models.TextField()
    image     = models.ImageField()
    birthdate= models.DateField()
```


```
django-admin.py syncdb
```

web2py

```
Friend = db.define_table('friend',
    Field('name',length=255,notnull=True),
    Field('email',requires=IS_EMAIL()),
    Field('info','text'),
    Field('image','upload',requires=IS_IMAGE()),
    Field('birthdate','date'))
```

```
automatic migrations
CREATE / ALTER tables
```

Meta Models (weird stuff)

web2py

```
db.define_table('meta_thing',Field('names','list:string'))

meta_thing = db.meta_thing(session.id)

Thing = db.define_table('thing_type_%i' % meta_thing.id,
     *[Field(field.name) for field in meta_thing.fields])
```

Database queries

Django

```
q = Friend.objects.filter(name__startswith="T")
q = q.filter(birthdate__lte=datetime.now())
q = q.exclude(info__icontains="python")
print q
```

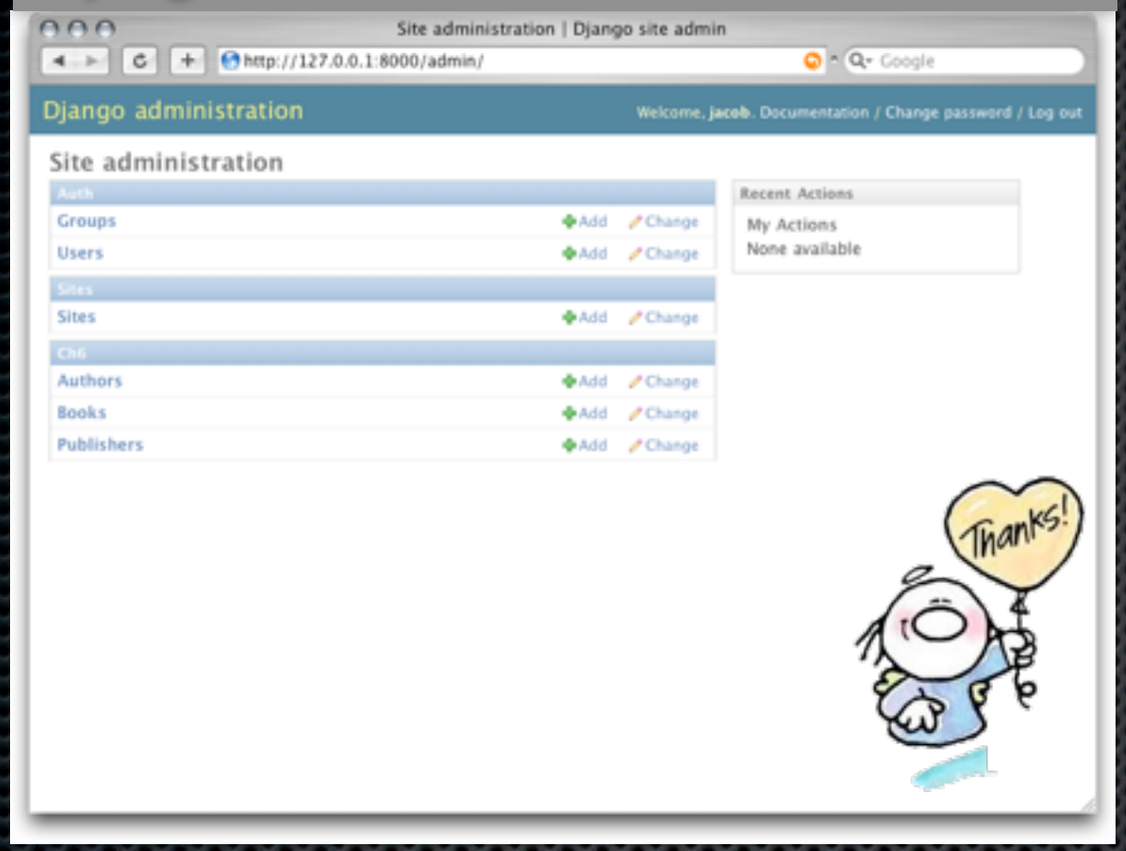
web2py

```
q = Friend.name.startswith("T")
q = q & (Friend.birthdate<datetime.now())
q = q & (!Friend.info.contains("python"))
print db(q).select()
```

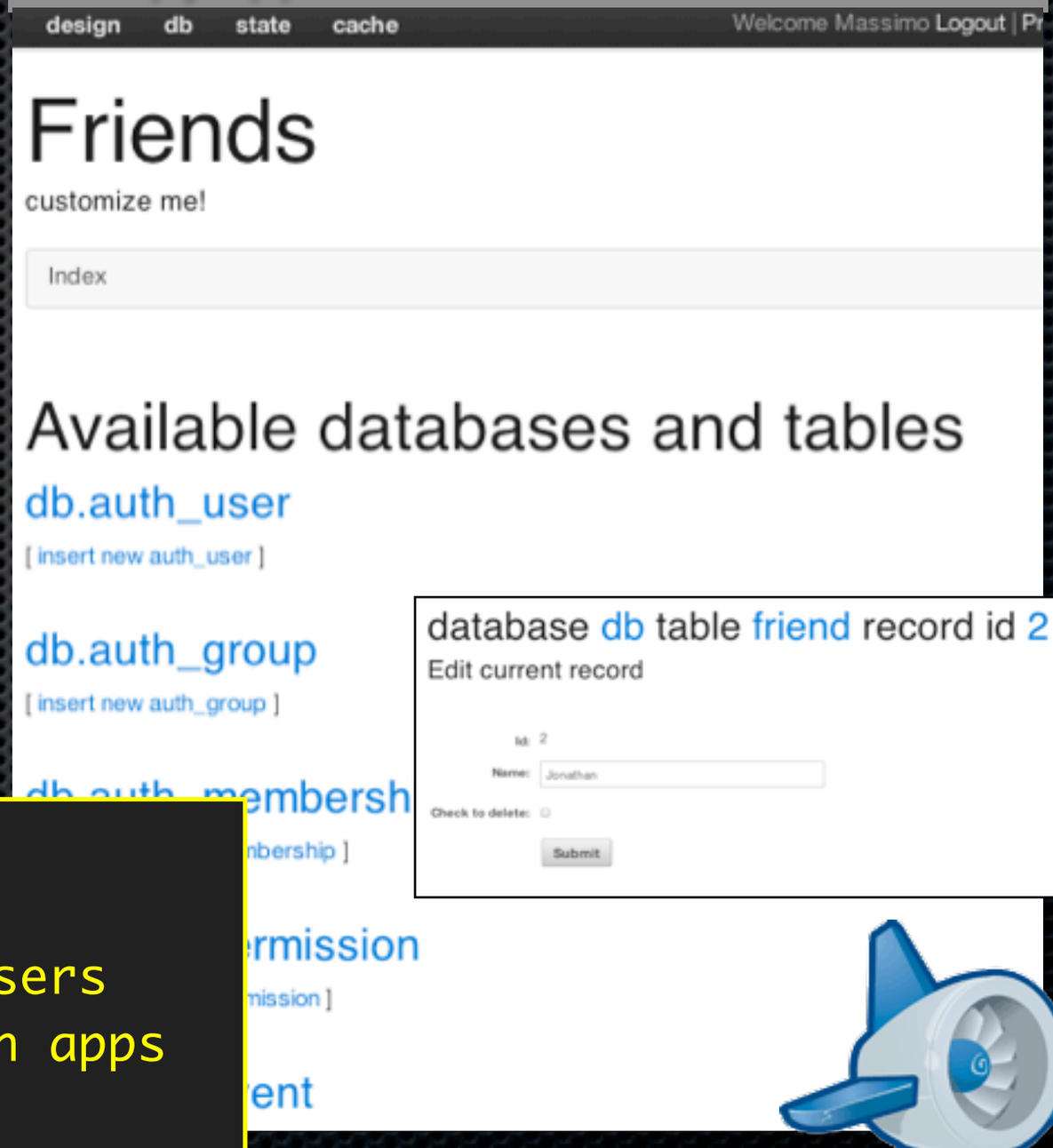
overloaded operators

appadmin (database admin)

Django admin



web2py appadmin

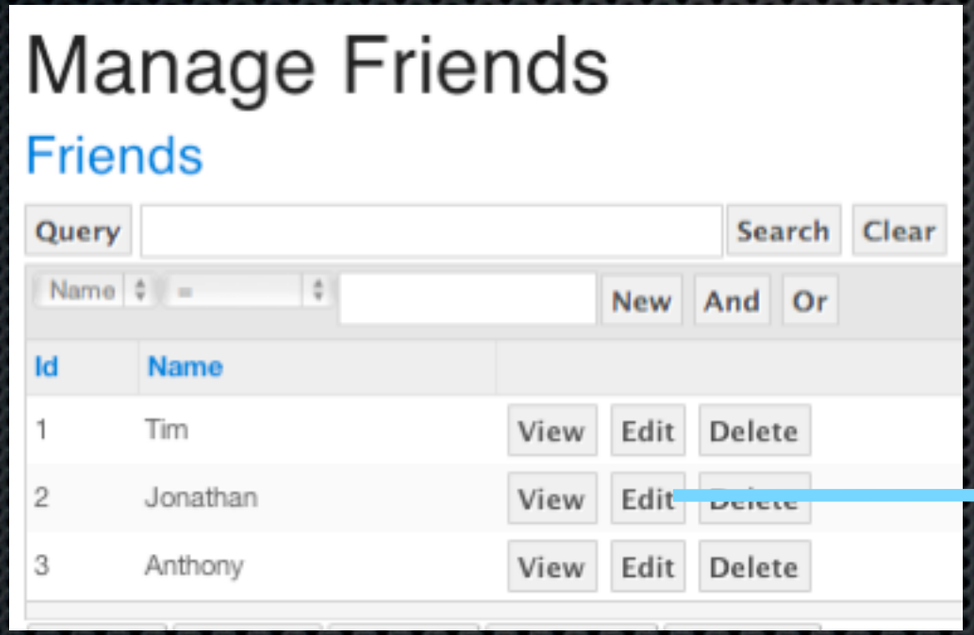


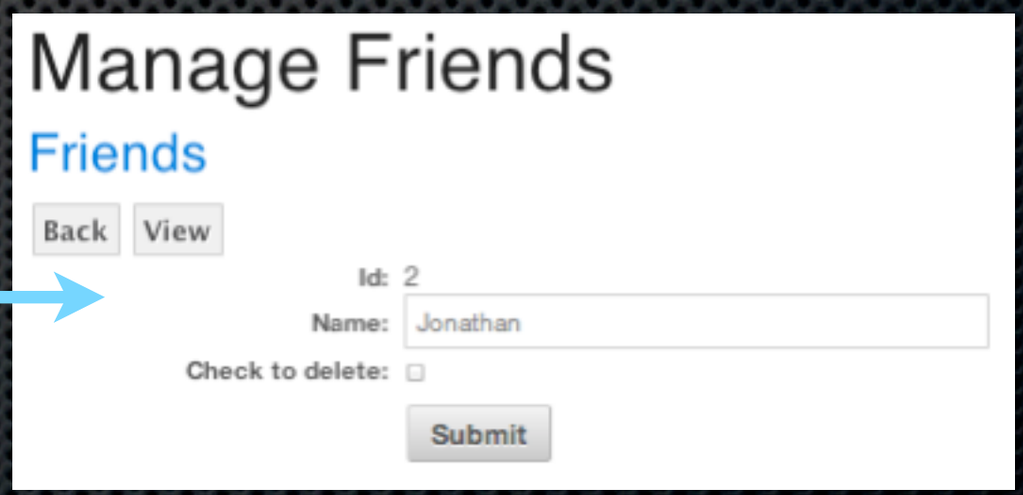
every app has its own
same look and feel as the app
designed for administrators not users
crud components can be embedded in apps
runs on GAE

CRUD, Grid, SmartGrid

web2py

```
def manage_friends():
    grid = SQLFORM.smartgrid(db.friend)
    return locals()
```



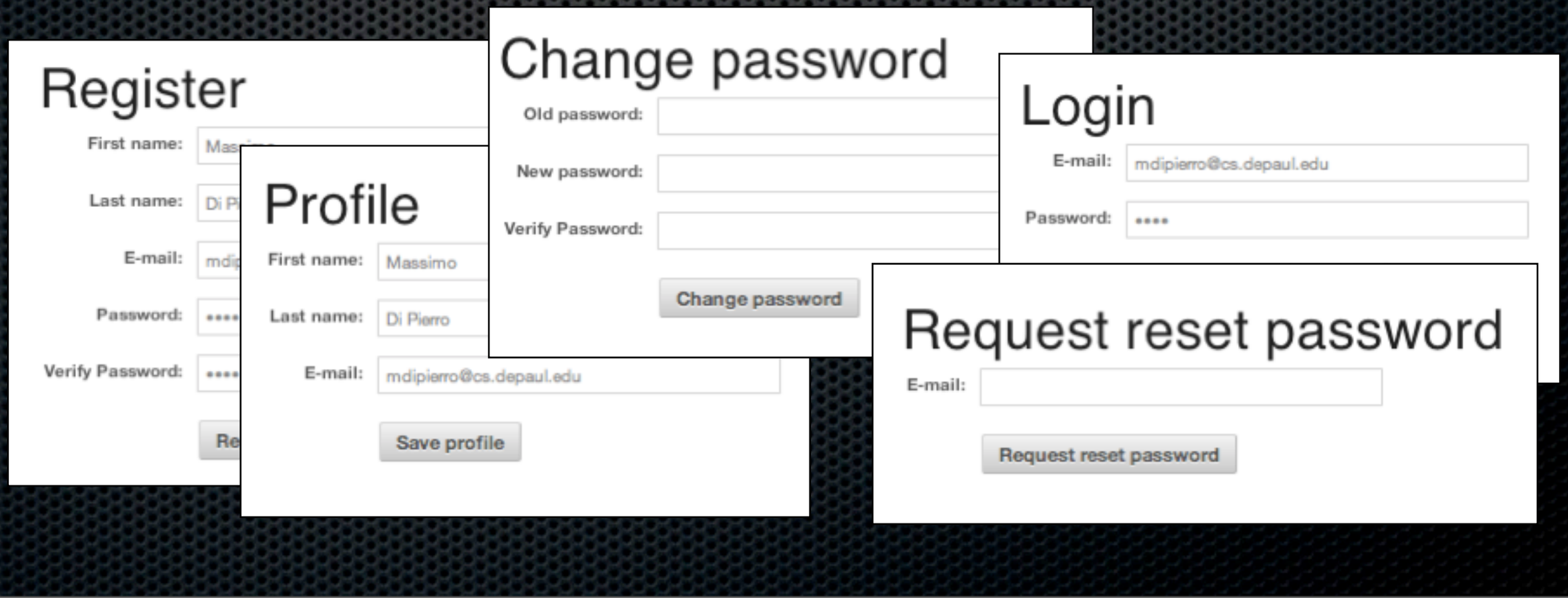


breadcrumbs | query builder | pagination | reference links

Role Based Access Control

web2py

```
@auth.requires_login()
@auth.requires_membership(role='friend manager')
@auth.requires_permission('delete', 'friend')
def manage_friends():
      ...
```



Federated Authentication

(with built-in CAS consumer and provider in every app)

Provider app

any app can delegate authentication to another local or remote web2py app (CAS 2.0)

Consumer apps

```
auth = Auth(db,cas_provider='http://.../friends/default/user/cas')
```

Record Versioning (all tables)

web2py

```
auth.enable_record_versioning(db)
```

Will record all changes to all records

Templates

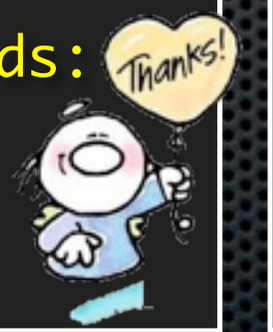
{{...python...}}

Mako

```
<%inherit file="base.html"/>

<%def name="show(friend)">
  <tr>
    <td>${friend.name}</td>
    <td>${friend.email}</td>
  </tr>
</%def>

<table>
    % for friend in friends:
        ${show(friend)}
    % endfor
</table>
```



web2py

```
{{extend "base.html}}

{{ def show(friend): }}
  <tr>
    <td>{{=friend.name}}</td>
    <td>{{=friend.email}}</td>
  </tr>
{{ return }}

<table>
    {{ for friend in friends: }}
        {{ show(friend) }}
    {{ pass }}
</table>
```

return closes def

pass closes for

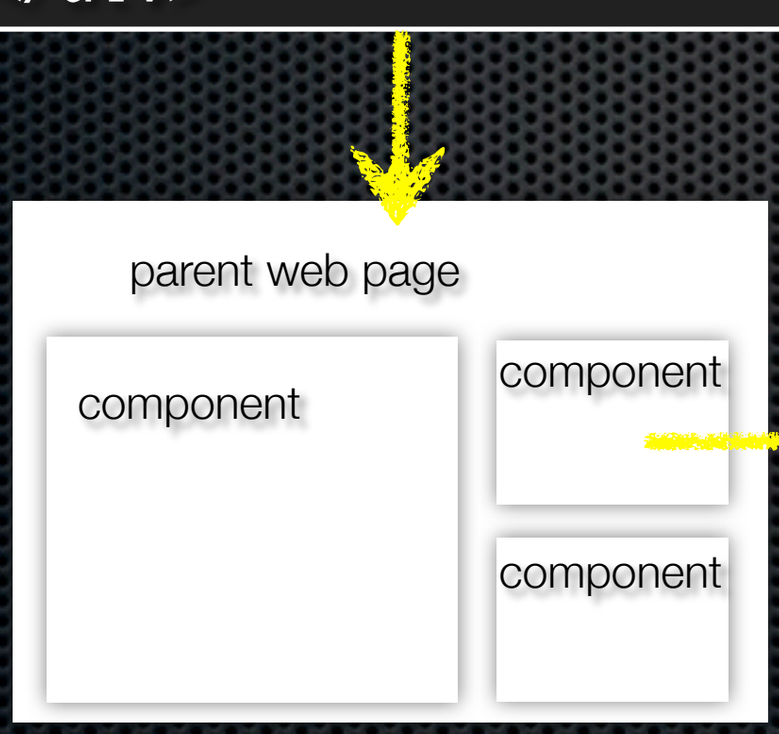
everything escaped by default except XML('...',sanitize=True)

Beyond MVC

(ajax components with signed URL)

web2py (parent template)

```
<div>
{{=LOAD('plugin','component1',user_signature=True,ajax=True)}}
{{=LOAD('plugin','component2',user_signature=True,ajax=True)}}
{{=LOAD('plugin','component3',user_signature=True,ajax=True)}}
</div>
```




```
@auth.requires_signature()
def component1():
    form = SQLFORM(db.friend)
    if form.process().accepted:
        response.flash = 'done!'
    return locals()
```

Modules and thread locals

Flask (proxies to objects that are local to a specific context)

```
from flask import request

with app.request_context(environ):
    assert request.method == 'POST'
```

web2py (thread-locals)

```
from gluon import current

assert current.request.env.http_method == 'POST'
```

thread local

request obj

wsgi environ

No Library Conflicts

web2py (each app can ship with its own version of libraries)

```
# app 1
import mylib     # from applications/app1/modules/

# app 2
import mylib     # from applications/app2/modules/
```

https://github.com/mitsuhiko/multiversion

```
import multiversion
multiversion.require_version('mylib', '1.0')
import mylib
```

Web Services

```
@service.json
@service.xml
@service.jsonrpc
@service.xmlrpc
@service.soap
@service.amfrpc3('domain')
def all(key):
    return db(db.friend.name.contains(key)).select()
```

```
http://.../friends/default/call/json/add?key=tim
http://.../friends/default/call/xml/add?key=tim
...
```

```
from xmlrpclib import ServerProxy
server = ServerProxy('http://.../friends/default/call/xmlrpc')
print server.key('tim')
```

RESTful Web Services

```
@request.restful()
def myAPI():
    def POST(**vars):
        return db.friend.insert(**vars)
    def GET(*args,**vars):
        patterns = [
            "/friends[friend]",
            "/{friend.name.startswith}",
            "/{friend.name}/:field",
            ]
        return db.parse_as_rest(patterns,args,vars).response
    def DELETE(id):
        return db(db.friend.id==id).delete()
    def PUT(id,**vars):
        return db(db.friend.id==id).update(**vars)
    return locals()
```

```
http://.../friends/default/myAPI/John/email
```

Scheduler

web2py (in a model file for example)

```
from gluon.scheduler import Scheduler

def email_friend(friend_id,subject,message):
    email = db.friend(friend_id).email
    mailer.send(to=email, subject=subject, message=message)
    return 'done!'

scheduler = Scheduler(db,dict(mytask=email_friend))
```

Start workers, local or remote

```
web2py.py -K friends &
web2py.py -K friends &
web2py.py -K friends &
```

```
db.scheduler_task.insert(task='mytask',args=[2,'hi','...'])
```

Manage tasks from appadmin interface

Multi-tenancy

web2py

```
db._common_fields.append(
    Field('request_tenant',default=request.env.host_name))
```

`http://domain1/friends`

`http://domain2/friends`

`http://domain3/friends`

share database but see different data
records filtered by domain (or other condition)

Let's build something

- URL shortening service
- Bookmarking service with tagging
- Click counter
- Url rating (WOT service)

models/myapp.py

```
Link = db.define_table(
    'link',
    Field('url',unique=True),
    Field('visits','integer',default=0),
    Field('screenshot','upload',writable=False),
    format = '%(url)s')

Bookmark = db.define_table(
    'bookmark',
    Field('link','reference link',writable=False),
    Field('category',
          requires=IS_IN_SET(['work','personal'])),
    Field('tags','list:string'),
    auth.signature)
```

controllers/default.py

```
def index():
    return dict()
```

models/myapp.py

```
def toCode(id):
    s,c = 'GKys67LJPDAFvcEp9rkwRd43fCjbxSXzMTQ28hgeUWuNYmqZt5VanBH',''
    while id: c,id = c+s[id % 55], id//55
    return c

def toInt(code):
    s,id = 'GKys67LJPDAFvcEp9rkwRd43fCjbxSXzMTQ28hgeUWuNYmqZt5VanBH',0
    for i in range(len(code)): id += s.find(code[i])*55**i
    return id

def shorten(id, row):
    s = URL('visit',args=toCode(id),scheme=True)
    return A(s,_href=s)

Bookmark.link.represent = shorten
Bookmark.id.readable = False
Bookmark.is_active.readable = False
Bookmark.is_active.writable = False
```

models/myapp_utils.py

```
from urllib import urlopen

def thumbalizr(url):
    """ free service to download web page thumbnail """
    return Bookmark.screenshot.store(
     urlopen('http://api1.thumbalizr.com/?url=%s&width=250'%url),
         filename = 'shot.png')

def wotrate(url):
    """ Web Of Trust service for web site rating """
    from gluon.tools import fetch
    from gluon.contrib.simplejson import loads
    wot = 'http://api.mywot.com/0.4/public_link_json?hosts=%s/'
    url = url.split('/')[2]
    return loads(urlopen(wot % url).read())[url]
```

models/menu.py

```
response.menu = [
    (T('Home'), False, URL('default','index')),
    (T('My Bookmarks'), False, URL('default','bookmarks')),
    ]
```

controllers/default.py

http://127.0.0.1:8000/myapp/default/bookmark?url=http://www.google.com

```
@auth.requires_login()
def bookmark():
    """ allows users to bookmark a page an get short url """
    url = request.vars.url
    link = Link(url=url) or Link.insert(url=url)
    link.update_record(screenshot=thumbalizr(url))
    bid = Bookmark(link=link) or Bookmark.insert(link=link)
    rating = cache.ram(link.url,lambda:wotrate(link.url),3600)
    form = SQLFORM(Bookmark,bid).process(next='bookmarks')
    return locals()

def visit():
    """ tracks visitors """
    link = Link(toInt(request.args(0)))
    link.update_record(visits=link.visits+1)
    redirect(link.url)
```

controllers/default.py

```
@auth.requires_login()
def bookmarks():
    """ allow users to manage their bookmarks """
    query = (Bookmark.created_by==auth.user.id)&\
            (Bookmark.link==Link.id)
    grid = SQLFORM.grid(query, create=False)
    return locals()

# stuff required for authenticatio
def user(): return dict(form=auth())
def download(): return response.download(request,db)
```

routes.py

localhost/K2 > localhost/myapp/default/visit/K2 > www.google.com

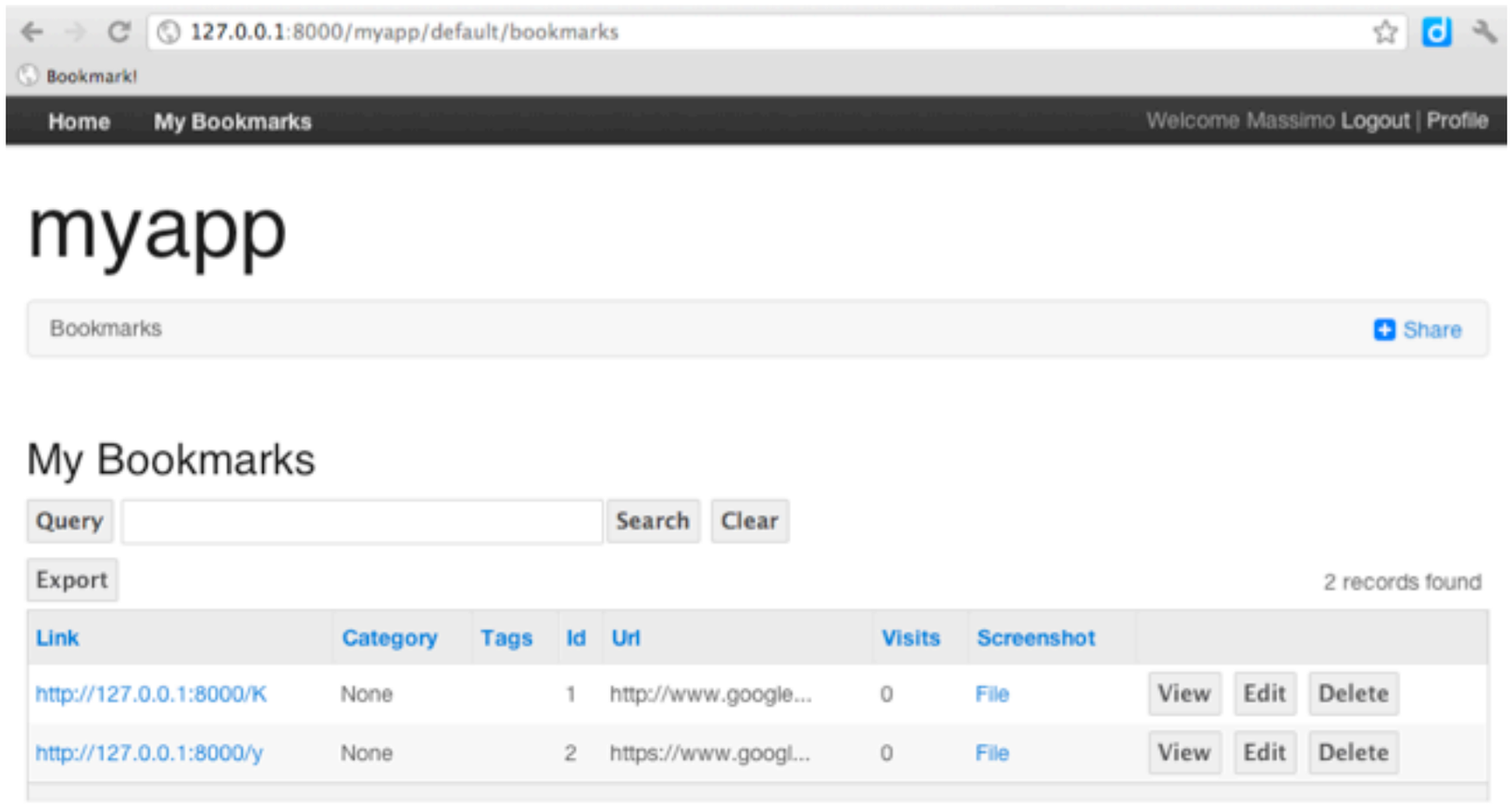
```
routes_in = [
    ('/(?P<code>\w+)','/myapp/default/visit/\g<code>'),
    ]

routes_out = [
    ('/myapp/default/visit/$code','/$code'),
    ]

routes_onerror=[
    ('myapp/*', 'myapp/default/error'),
    ]
```

views/default/bookmarks.py

```
{{extend 'layout.html'}}
<h3>My Bookmarks</h3>
{{=grid}}
```



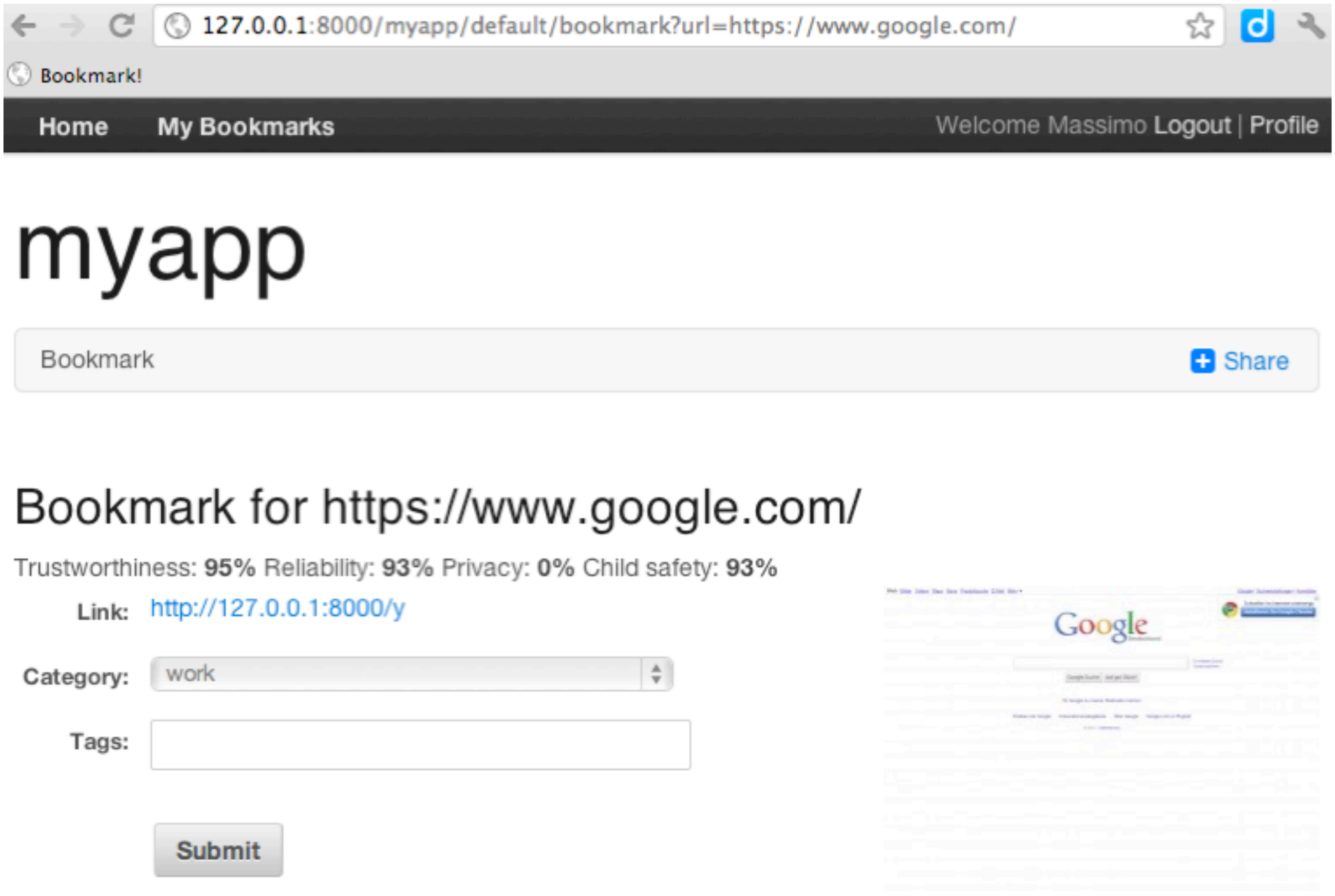
views/default/bookmark.py

```
{{extend 'layout.html'}}

<h3>Bookmark for {{=url}}</h3>

{{def show(rating,k):}}
    <strong>{{=rating.get(k,[0])[0]}}%</strong>{{return}}
<div>
Trustworthiness: {{show(rating,'1')}}
Reliability: {{show(rating,'2')}}
Privacy: {{show(rating,'3')}}
Child safety: {{show(rating,'4')}}
</div>
<img src="{{=URL('default','download',
                 args=link.screenshot)}}" align="right"/>
{{=form}}
```

views/default/bookmark.py

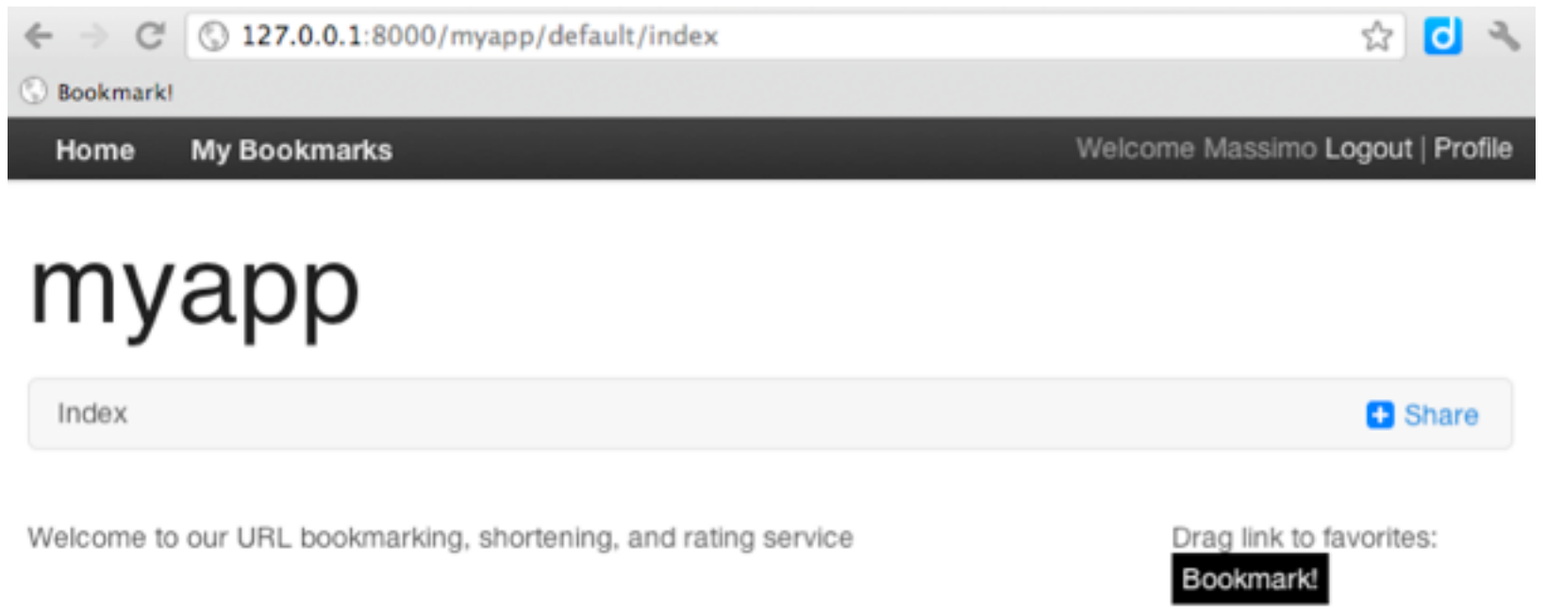


views/default/index.html

```
{{right_sidebar_enabled=True}}
{{extend 'layout.html'}}

Welcome to our bookmarking & url shortening service

{{block right_sidebar}}
Drag link to favorites:
<a style="padding:5px;color:white;background-color:black;"
   title="Bookmark"
   href="javascript:window.location='{{=URL('bookmark',
   scheme=True)}}?url='+window.location">Bookmark!</a>
{{end}}
```



models/services.py

http://127.0.0.1:8000/myapp/services/call/xml/linksby?key=google
http://127.0.0.1:8000/myapp/services/api/google.json

```
@service.json
@service.xml
@service.jsonrpc
@service.xmlrpc
@service.amfrpc3('domain')
@service.soap()
def linksby(key=''):
    return db(Link.id==Bookmark.link)\
        (Bookmark.tags.contains(key)).select(Link.ALL,distinct=True)

@request.restful()
def api():
    def POST(**vars):
        raise HTTP(501)
    def GET(key=''):
        return dict(result=linksby(key).as_list())
    return locals()

def call(): return service()
```

models/mytasks.py

```
from gluon.scheduler import Scheduler

def emailme():
    me = 'mdipierro@cs.depaul.edu'
    message = '%s users' % db(db.auth_user).count()
    mailer.send(to=me, subject='stats', message=message)
    return 'done!'

scheduler = Scheduler(db,dict(emailme=emailme))
if db(db.scheduler_task).isempty():
    db.scheduler_task.insert(
        application_name=request.application,
        task_name='email me user count',
        function_name='emailme',args=[],vars={},
        period=3600*24,repeats=0)
```

```
web2py.py -K friends &
web2py.py -K friends &
web2py.py -K friends &
```

Give it a try!

3700 registered users