web2py Dojo

@PyCon 2009

Goal

- Write an application that allows you to post news (in markdown syntax), attach and link documents to news, restrict access, provides login, logout, registration, and exposes the posted news as multiple services.
- 29 (models) + 47 (controllers) + 45 (views) = 121 total lines of code
- type only stuff in blue

Part 0

- Download “web2py 1.59” from

  http://www.web2py.com
- Start it with:

  ```
  $ python web2py.py
  ```
- Create an application called “news”

Part 1

- Create a table “news_item” with fields:
  - title
  - body
  - posted_on

FILE: applications/news/models/db.py

```
try:
    from gluon.contrib.gql import *
except:
    db = SQLDB('sqlite://storage.db') # connect to SQLite
else:
    db = GQLDB()  # connect to Google BigTable
    session.connect(request, response, db=db)

db.define_table('news_item',
   db.Field('title',length = 128),
   db.Field('body','text'),
   db.Field('posted_on','datetime'))
```

Part 1

- Create a table “document” with fields
  - news_id which references news_item
  - name
  - uploaded file

FILE: applications/news/models/db.py

```
try:
    from gluon.contrib.gql import *
except:
    db = SQLDB('sqlite://storage.db') # connect to SQLite
else:
    db = GQLDB()  # connect to Google BigTable
    session.connect(request, response, db=db)


db.define_table('news_item',
   db.Field('title',length = 128),
   db.Field('body','text'),
   db.Field('posted_on','datetime'))

db.define_table('document',
   db.Field('news_id',db.news_item),
   db.Field('name',length = 128),
   db.Field('file','upload'))
```

Try appadmin

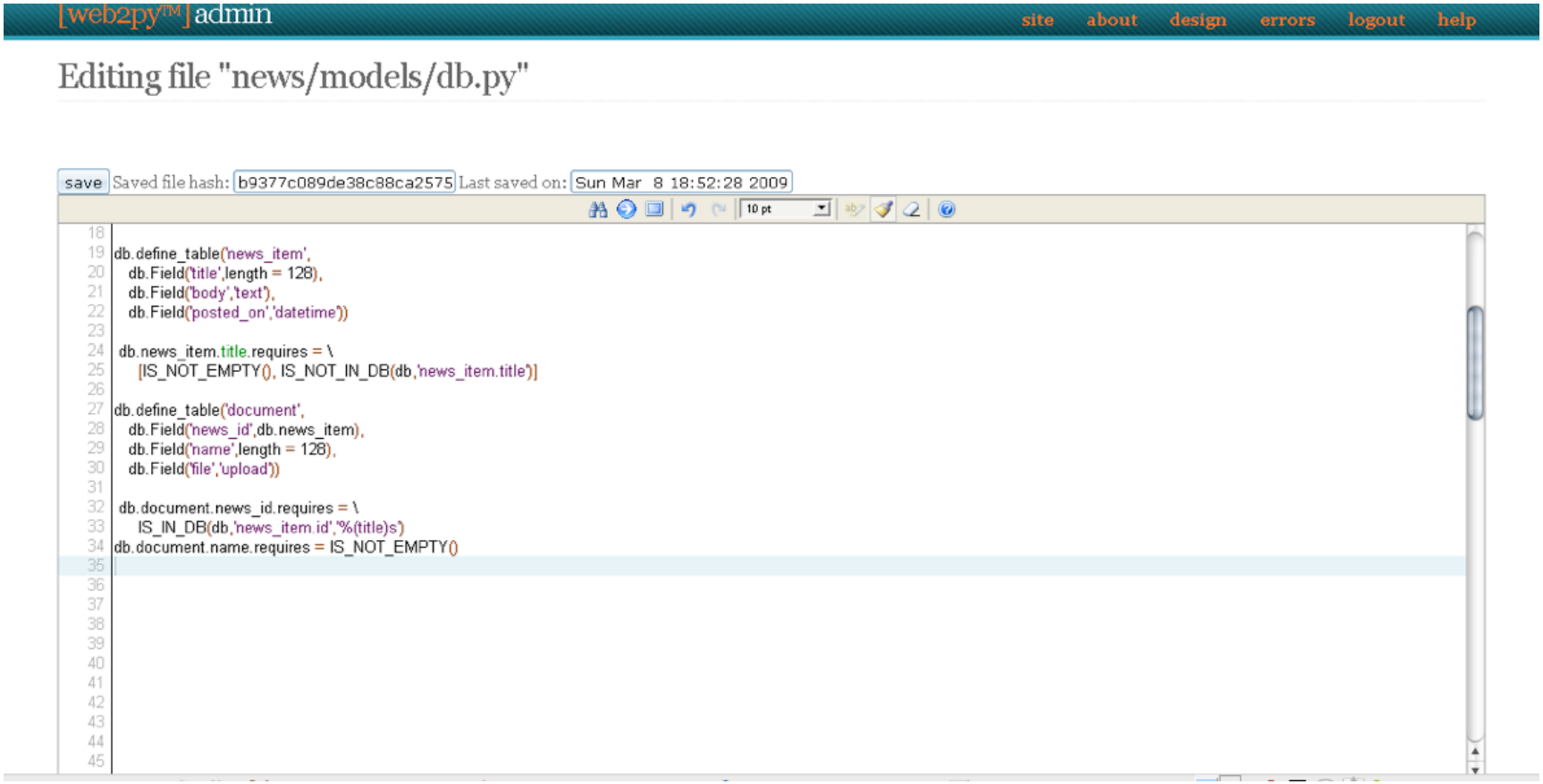
- http://127.0.0.1:8000/news/appadmin
- Try insert some records
- Try select some records

Part II

- Hide some fields that should be filled automatically
- Add validators:
  - news_item.title has to be unique
  - ...
- Create a simple index controller

FILE: applications/news/models/db.py

```
db.define_table('news_item',
   db.Field('title',length = 128),
   db.Field('body','text'),
   db.Field('posted_on','datetime', writable=False))

db.news_item.title.requires = \
     [IS_NOT_EMPTY(), IS_NOT_IN_DB(db,'news_item.title')]

db.define_table('document',
   db.Field('news_id',db.news_item),
   db.Field('name',length = 128),
   db.Field('file','upload'))

db.document.news_id.requires = \
     IS_IN_DB(db,'news_item.id','%(title)s')
db.document.name.requires = IS_NOT_EMPTY()
```


[web2py™] admin
site about design errors logout help
Editing file "news/models/db.py"
save
Saved file hash: b9377c089de38c88ca2575
Last saved on: Sun Mar 8 18:52:28 2009
10 pt
db.define_table('news_item',
db.Field('title',length = 128),
db.Field('body','text'),
db.Field('posted_on','datetime'))
db.news_item.title.requires = \
[IS_NOT_EMPTY(), IS_NOT_IN_DB(db,'news_item.title')]
db.define_table('document',
db.Field('news_id',db.news_item),
db.Field('name',length = 128),
db.Field('file','upload'))
db.document.news_id.requires = \
IS_IN_DB(db,'news_item.id','%(title)s')
db.document.name.requires = IS_NOT_EMPTY()

Try appadmin again

- http://127.0.0.1:8000/news/appadmin
- Try insert a news item without title

FILE: applications/news/controllers/default.py

```
# http://127.0.0.1:8000/news/default/index
def index():
    news = db().select(db.news_item.ALL)
    return dict(news = news)
```

FILE: applications/news/views/default/index.html

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

<h1>News</h1>

<ul>
{{for item in news:}}
{{=LI(A(item.title,
        _href=URL(r=request,f='read',args=item.id)))}}
{{pass}}
</ul>
```

Try index

- http://127.0.0.1:8000/news/default/index

[web2py™] news

News

- News Today

Part III

- add CRUD
- an action to post news
- an action to read news
- an action to edit news
- an action to manage attached documents
- add some views

FILE: applications/news/models/db.py

```
try:
    from gluon.contrib.gql import *
except:
    db = SQLDB('sqlite://storage.db') # connect to SQLite
else:
    db = GQLDB()  # connect to Google BigTable
    session.connect(request, response, db=db)

from gluon.tools import *
crud=Crud(globals(),db)
```

FILE: applications/news/controllers/default.py

```
# http://127.0.0.1:8000/news/default/index
def index():
    news = db().select(db.news.ALL)
    return dict(news = news)

def post():
    db.news_item.posted_on.default = request.now
    form = crud.create(db.news_item,
                       next=URL(r=request,f='index'))
    return dict(form = form)
```

FILE: applications/news/views/default/post.html

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

<h1>Post News</h1>

{{=form}}
```

FILE: applications/news/controllers/default.py

```
# http://127.0.0.1:8000/news/default/index
def index():
    news = db().select(db.news.ALL)
    return dict(news = news)

def post():
    db.news_item.posted_on.default = request.now
    form = crud.create(db.news_item,
                       next=URL(r=request,f='index'))
    return dict(form = form)

# http://127.0.0.1:8000/news/default/read/[id]
def read():
    item = db.news_item[request.args[0]]
    return dict(item = item)
```

FILE: applications/news/controllers/default.py

```
# http://127.0.0.1:8000/news/default/edit/[id]
def edit():
    item = db.news_item[request.args[0]]
    form = crud.update(db.news_item,item,
                       next=URL(r=request,args=item.id))
    return dict(form = form)
```

FILE: applications/news/views/default/edit.html

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

<h1>Edit News</h1>

{{=form}}
```

FILE: applications/news/controllers/default.py

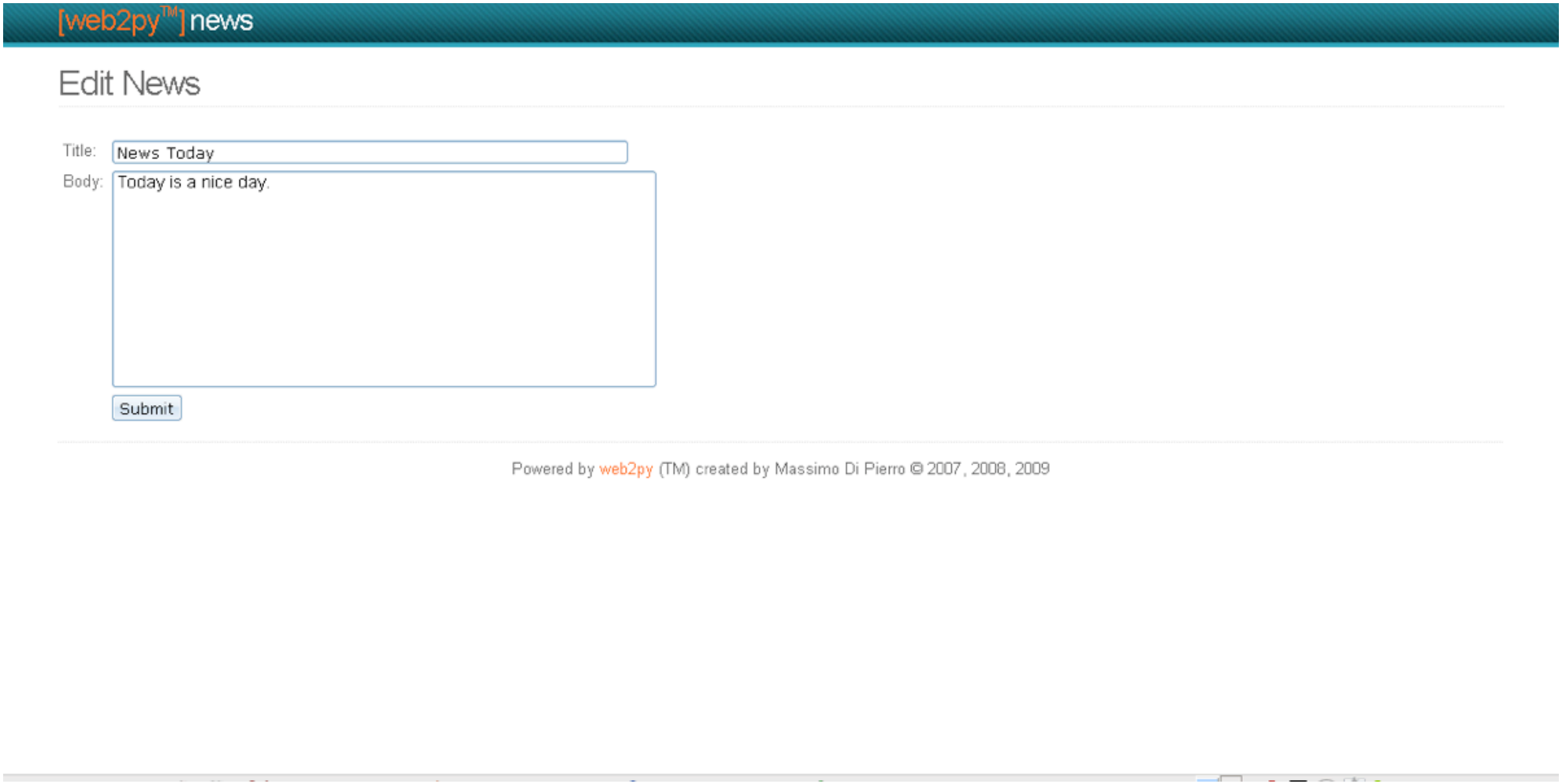
```
# http://127.0.0.1:8000/news/default/edit/[id]
def edit():
    item = db.news_item[request.args[0]]
    form = crud.update(db.news_item,item,
                       next=URL(r=request,args=item.id))
    return dict(form = form)

# http://127.0.0.1:8000/news/default/attachments/[id]
def attachments():
    item = db.news_item[request.args[0]]
    db.document.news_id.writable = False
    db.document.news_id.default = item.id
    form = crud.create(db.document,
                       next=URL(r=request,args=item.id))
    documents = db(db.document.news_id==item.id).select()
    return dict(item=item,form=form,documents=documents)

def download(): return response.download(request,db)
```

FILE: applications/news/views/default/attachments.html

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

<h1>Attachments to New Item</h1>

<h2>{{=item.title}}</h2>

<ul>
{{for doc in documents:}}
{{=LI(A(doc.name,
        _href=URL(r=request,f='download',args=doc.file)))}}
{{pass}}
</ul>

<h2>Post new document</h2>

{{=form}}
```

Try post, list, edit

- http://127.0.0.1:8000/news/default/post
- http://127.0.0.1:8000/news/default/index
- http://127.0.0.1:8000/news/default/edit/[id]
- http://.../news/default/attachments/[id]


[web2py™] news
Edit News
Title:
News Today
Body:
Today is a nice day.
Submit
Powered by web2py (TM) created by Massimo Di Pierro © 2007, 2008, 2009

[web2py™] news

Attachments to New Item

News Today

Post new document

Name:

File: Browse...

Submit

Part IV

- add Authentication and basic Authorization
- change the news_item.created_by fields to a reference and see a migration “happen”
- markup actions that require login

FILE: applications/news/models/db.py

```
from gluon.tools import *

#mail = Mail()
#mail.settings.server = 'smtp.gmail.com:587'
#mail.settings.sender = 'you@gmail.com'
#mail.settings.login = 'username:password'

auth = Auth(globals(),db)
#auth.captcha = Recaptcha('public_key','private_key')
#auth.settings.mailer = mail
auth.define_tables()

crud=Crud(globals(),db)

db.define_table('news_item',
   db.Field('title',length = 128),
   db.Field('body','text'),
   db.Field('posted_on','datetime',writable=False),
   db.Field('posted_by',db.auth_user,writable=False))
```

FILE: applications/news/controllers/default.py

```
# http://127.0.0.1:8000/news/default/user/login
# http://127.0.0.1:8000/news/default/user/logout
# http://127.0.0.1:8000/news/default/user/register
# http://127.0.0.1:8000/news/default/user/change_password
# http://127.0.0.1:8000/news/default/user/profile
# http://127.0.0.1:8000/news/default/user/retrieve_password
def user(): return dict(form=auth())
```

FILE: applications/news/views/default/user.html

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

<h1>{{=request.args[0].replace('_',' ').capitalize()}}</h1>

{{=form}}

{{if request.args[0]=='login':}}
{{=A('register',_href=URL(r=request,f='user/register'))}}<br/>
{{=A('lost password?',_href=URL(r=request,f='user/
retrieve_password'))}}<br/>
{{pass}}
```

FILE: applications/news/controllers/default.py

```
# http://127.0.0.1:8000/news/default/index
def index():
    news = db().select(db.news_item.ALL)
    return dict(news = news)

@auth.requires_login()
def post():
    db.news_item.posted_on.default = request.now
    db.news_item.posted_by.default = auth.user.id
    form = crud.create(db.news_item,
                       next=URL(r=request,f='index'))
    return dict(form = form)

# http://127.0.0.1:8000/news/default/read/[id]
def read():
    item = db.news_item[request.args[0]]
    return dict(item = item)
```

FILE: applications/news/controllers/default.py

```
# http://127.0.0.1:8000/news/default/edit/[id]
@auth.requires_login()
def edit():
    item = db.news_item[request.args[0]]
    if item.posted_by!=auth.user.id:
        redirect(URL(r=request,f='index'))
    form = crud.update(db.news_item,item,
                       next=URL(r=request,args=item.id))
    return dict(form = form)

# http://127.0.0.1:8000/news/default/attachments/[id]
@auth.requires_login()
def attachments():
    item = db.news_item[request.args[0]]
    if item.posted_by!=auth.user.id:
       redirect(URL(r=request,f='index'))
    db.document.news_id.writable = False
    db.document.news_id.default = item.id
    form = crud.create(db.document,
                       next=URL(r=request,args=item.id))
    documents = db(db.document.news_id==item.id).select()
    return dict(item=item,form=form,documents=documents)
```

FILE: applications/news/views/default/read.html

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

<h1>{{=item.title}}</h1>

{{from gluon.contrib.markdown import WIKI}}
{{=WIKI(item.body)}}

{{if auth.is_logged_in() and item.posted_by==auth.user.id:}}
{{=A('edit',
     _href=URL(r=request,f='edit',args=item.id))}} |
{{=A('attachments',
     _href=URL(r=request,f='attachments',args=item.id))}}
{{pass}}
```

Try register, login, logout

- http://.../news/default/user/login
- http://.../news/default/user/register
- http://.../news/default/user/logout
- http://.../news/default/user/profile
- http://.../news/default/user/change_password
- http://.../news/default/user/retrieve_password
- check out sql.log for the migration

[web2py™] news

Login

Email:

Password:

Submit

register
lost password?

Part V

- add a service that to download news in
  - CSV
  - XML
  - JSON
- allows to request the news via
  - JSONRPC
  - XMLRPC
  - AMFRPC for flash

FILE: applications/news/controllers/default.py

```
service=Service(globals())
@service.csv
@service.xml
@service.json
@service.jsonrpc
@service.xmlrpc
@service.amfrpc
def news():
    return db().select(db.news_item.ALL).as_list()

# http://127.0.0.1:8000/news/default/call/csv/news
# http://127.0.0.1:8000/news/default/call/xml/news
# http://127.0.0.1:8000/news/default/call/json/news
# http://127.0.0.1:8000/news/default/call/jsonrpc -> news()
# http://127.0.0.1:8000/news/default/call/xmlrpc  -> news()
# http://127.0.0.1:8000/news/default/call/amfrpc  -> news()
def call(): return service()
```

Try your services

- http://.../news/default/call/xml/news
- http://.../news/default/call/csv/news
- http://.../news/default/call/json/news
- ...

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
- <items>
  - <item>
    - <record>
      - <field name="body">
          <atom type="str">Today is a nice day.</atom>
        </field>
      - <field name="id">
          <atom type="str">2</atom>
        </field>
      - <field name="posted_on">
          <atom type="str">2009-03-31 09:56:29</atom>
        </field>
      - <field name="title">
          <atom type="str">News Today</atom>
        </field>
      </record>
    </item>
  </items>
```

Part VI

- download dev_appserver
- python dev_appserver.py /path/to/web2py
- look for the .yaml files under web2py/