References


Django source code

How to use sessions, http://www.djangoproject.com/documentation/sessions/

Reading

Authentication
Authentication

Turned on by default

Parts
Users
Permissions
Groups
Messages
Users

Django model

- username -- Required.
- first_name
- last_name
- email
- password -- Required.
- is_staff
- is_active
- is_superuser
- last_login
- date_joined

http://code.djangoproject.com/browser/django/trunk/django/contrib/auth/models.py
Adding Users

Admin Page
Script
View
# Admin Site

## Django administration

<table>
<thead>
<tr>
<th>Site administration</th>
<th>Auth</th>
<th>Groups</th>
<th>Users</th>
<th>Sites</th>
<th>Sites</th>
<th>Polls</th>
<th>Polls</th>
<th>Books</th>
<th>Books</th>
<th>Authors</th>
<th>Authors</th>
<th>Books</th>
<th>Science books</th>
<th>Forms</th>
<th>Contact infos</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Site administration</strong></td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
<td>Add</td>
</tr>
<tr>
<td><strong>My Actions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Recent Actions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Welcome, whitley. Documentation / Change
Add User Page

Need to add password in encoded form
Creating Groups via code

Al 16->manage.py shell
(InteractiveConsole)

>>> from django.contrib.auth.models import User, Group
>>> staff = Group(name='staff')
>>> staff.save()

>>> Group.objects.filter(name='staff')
[<Group: staff>]

>>> sam = User.objects.get(username='sam')
>>> sam.groups.add(staff)
>>> sam.groups.all()
[<Group: staff>]
>>> sam.save()
Permissions

Each model has three default permissions
  add
  change
  delete
Add Permissions

class Poll(models.Model):
    question = models.CharField(maxlength=200)
    pub_date = models.DateTimeField('date published')

    def __str__(self):
        return self.question

class Admin:
    pass

class Meta:
    permissions = (
        ("can_view", "Can view poll list"),
        ("can_vote", "Can vote"),
        ("can_see_results", "Can see voting results"),
    )
Accessing Permission Directly

```python
>>> from django.contrib.auth.models import Permission
>>> Permission.objects.filter(codename='can_view')
[<Permission: poll | Can view poll list>]
```
Poll Example

Polls have permissions
"can_view"
"can_vote"
"can_see_results"

Users must
  login in
  have correct permissions

Node
Model changed per previous slide

manage.py syncdb run
Poll Example - Long Way
Login

http://127.0.0.1:8000/polls/login.html → cs683.polls.views.loginPage

cs683.polls.views.loginPage

def loginPage(request):
    return render_to_response('polls/simpleLogin.html', {})
<h1>Please Login in</h1>

<form method="post" action="processLogin.html">
<table>
<tr>
    <td>Name:</td>
    <td><input type="text" name="username" size="30" value="\{name\}" maxlength="100" />
</td>
</tr>
<tr>
    <td>Password:</td>
    <td><input type="password" name="password" size="30" value="\{Password\}" maxlength="75" />
</td>
</tr>
</table>
<input type="submit" />
</form>
def processLogin(request):
    username = request.POST['username']
    password = request.POST['password']
    user = authenticate(username=username, password=password)
    if user is not None:
        login(request, user)
        # Just to show how to access default permissions
        if user.has_perm('polls.add'):
            print 'add'
            return HttpResponseRedirect('/polls/')
    else:
        return HttpResponseRedirect('/polls/login.html')
def index(request):
    if not request.user.is_authenticated():
        return HttpResponseRedirect('/polls/login.html')

    if not request.user.has_perm('Poll.can_view'):
        return HttpResponseRedirect('/polls/invalidPermission.html')

    poll_list = Poll.objects.all().order_by('-pub_date')
    return render_to_response('polls/poll_list.html', {'object_list': poll_list})
Protecting Pages

Each view of each restricted page needs to check for authentication
Accessing user and Permissions in Template

Pass them in as template variables

If use RequestContext they are passed in automatically

But then can not use render_to_response
Login Short Cuts
Could not get the examples to run
from django.db import models

class Poll(models.Model):
    question = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

    def foo(self):
        return 'function call' + self.question

    def __str__(self):
        return self.question

class Admin:
    date_hierarchy = 'pub_date'
    list_display = ('question', 'pub_date', 'foo')
    list_filter = ('pub_date', )
from django.contrib.auth import logout

def logoutView(request):
    logout(request)

    return some page
# Poll Admin View

## Select poll to change

<table>
<thead>
<tr>
<th>2006</th>
<th>Date published</th>
<th>Foo</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What's up doc?</strong></td>
<td>Sept. 14, 2006, 3:43 p.m.</td>
<td>function callWhat's up doc?</td>
</tr>
<tr>
<td><strong>Do you like Python?</strong></td>
<td>Sept. 14, 2006, 11:22 a.m.</td>
<td>function callDo you like Python?</td>
</tr>
<tr>
<td><strong>Do you like Django Reinhardt's music?</strong></td>
<td>Sept. 13, 2006, 6:45 p.m.</td>
<td>function callDo you like Django Reinhardt's music?</td>
</tr>
</tbody>
</table>

3 polls

---

**Poll Details:**
- **What's up doc?**
  - Date: Sept. 14, 2006, 3:43 p.m.
  - Question: What's up doc?
  - Function call: function callWhat's up doc?

- **Do you like Python?**
  - Date: Sept. 14, 2006, 11:22 a.m.
  - Question: Do you like Python?
  - Function call: function callDo you like Python?

- **Do you like Django Reinhardt's music?**
  - Date: Sept. 13, 2006, 6:45 p.m.
  - Question: Do you like Django Reinhardt's music?
  - Function call: function callDo you like Django Reinhardt's music?
def vote(request, poll_id):
    if request.session.get(str(poll_id), False):
        return HttpResponse('You already voted on this poll')

    poll = get_object_or_404(Poll, pk=poll_id)
    try:
        selected_choice = poll.choice_set.get(pk=request.POST['choice'])
    except (KeyError, Choice.DoesNotExist):
        # Redisplay the poll voting form.
        return render_to_response('polls/poll_detail.html', {
            'poll': poll,
            'error_message': "You didn't select a choice.",
        })
    else:
        selected_choice.votes += 1
        selected_choice.save()
        request.session[str(poll_id)] = True

        return HttpResponseRedirect('/polls/%s/results/' % poll.id)