

Django PimenTech Howto

Author : Guillaume Savary
Contact : guillaume@nospam.pimentech.net
Revision : 401
Date : 2007-03-30
Copyright : This document has been placed in the public domain.
Tags : django

Memcached

Memcached est d'après nos tests 4 fois plus rapide que le cache 'file ://'. Il n'est pourtant pas utilisable directement : il faut installer le client Python que l'on peut télécharger ici : <ftp://ftp.tummy.com/pub/python-memcached/> . Il s'installe de façon standard avec :

```
python setup.py install
```

Pour le backend de cache, il est possible d'utiliser des noms d'hôte à la place de l'IP qu'indique la doc officielle :

```
CACHE_BACKEND = 'memcached://localhost:11211'
```

Newforms, real life example

Template form, simple version :

```
<form action="." method="post">
    {{ form }}
    <input class="envoyer" type="submit" value="{% trans "Envoyer" %}"/><br />
</form>
```

Template form, expanded :

```
<form action="." method="post">

    <label for="nom" class="required">{% trans "Nom" %}</label>{{ form.nom }}{{ form.errors.nom }}
    <label for="prenom" class="required">{% trans "Pré&eacute;nom" %}</label>{{ form.prenom }}{{ form.errors.prenom }}
    <label for="email" class="required">{% trans "e-mail" %}</label>{{ form.email }}{{ form.errors.email }}
    <label for="web">{% trans "Web" %}</label>{{ form.web }}{{ form.errors.web }}

    <label for="societe" class="required">{% trans "Soci&eacute;t&eacute;" %}</label>{{ form.societe }}{{ form.errors.societe }}
    <label for="fonction" class="required">{% trans "Fonction" %}</label> {{ form.fonction }}{{ form.errors.fonction }}

    <label for="telephone">{% trans "T&eacute;l&eacute;phone" %}</label>{{ form.telephone }}{{ form.errors.telephone }}

    <label for="adresse">{% trans "Adresse" %}</label>{{ form.rue }}{{ form.errors.rue }}
    <label for="codepostal">{% trans "Code Postal" %}</label>{{ form.cp }}{{ form.errors.cp }}
    <label for="ville">{% trans "Ville" %}</label>{{ form.ville }}{{ form.errors.ville }}
```

```
<label for="pays">{% trans "Pays" %}</label>{{ form.pays }}{{ form.errors.pays }}  
  
<input class="envoyer" type="submit" value="{% trans "Envoyer" %}"/><br />
```

```
</form>
```

In views.py :

```
class ContactForm(forms.Form):  
    nom = forms.CharField(required=True)  
    prenom = forms.CharField(required=True)  
    email = forms.EmailField(required=True)  
    web = forms.URLField()  
    societe = forms.CharField(required=True)  
    type_fonction_id = forms.ChoiceField(choices = [ (t.uid, t.nom) for t in TypeFonction.objects.all() ])  
    tel = forms.CharField()  
    rue = forms.CharField()  
    cp = forms.CharField()  
    ville = forms.CharField()  
    pays = forms.CharField()  
  
def newsletter_subscription(request):  
    form = ContactForm(request.POST or None)  
    template = 'site/newsletter_subscription.html'  
    if form.is_valid():  
        contact = Contact.objects.filter(email=form.data['email'])  
        if contact:  
            contact = contact[0]  
        else:  
            # Direct model initialisation with form datas  
            contact = Contact(**form.cleaned_data)  
        contact.subscribe_newsletter = True  
        contact.save()  
        template = 'site/newsletter_subscription_ok.html'  
    return render_to_response(template,  
                              { 'form': form },  
                              context_instance=RequestContext(request))
```

Passing form in hidden inputs

```
{% for field in form %}  
    {{ field.as_hidden }}  
{% endfor %}
```