



Présentation ADRAR Pôle Numérique

Suivez-nous... LinkedIn  et Twitter  @Adrar_Numerique



ADRAR PÔLE NUMERIQUE

PÔLE NUMERIQUE DU CENTRE DE FORMATION ADRAR

- > SUPPORT, ADMINISTRATION SYSTEMES & RESEAUX
- > DEVELOPPEMENT D'APPLICATIONS WEB & MOBILES
- > WEBDESIGN & DEVELOPPEMENT INTERFACES
- > TRANSFORMATION NUMERIQUE DES ENTREPRISES

<http://www.adrar-numerique.com>

Suivez-nous sur LinkedIn  Twitter  @Adrar_Numerique

... et sur le web : www.adrar-numerique.com



Suivez-nous sur LinkedIn 

Twitter  @Adrar_Numerique

... et sur le web : www.adrar-numerique.com

FORMULAIRES

Suivez-nous sur LinkedIn Twitter @Adrar_Numerique

... et sur le web : www.adrar-numerique.com

Mettons maintenant en place un formulaire pour que les utilisateurs puissent voter pour les sondages. Nous allons faire ça en modifiant notre template *detail.html*.

```
<!-- HTML de base -->
<form action="{% url 'sondage:vote' question.id %}" method="post">
  {% csrf_token %}
  <fieldset>
    <legend><h1>{{ question.question_text }}</h1></legend>
    {% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}
    {% for choice in question.choice_set.all %}
      <input type="radio" name="choice" id="choice{{ forloop.counter }}" value="{{ choice.id }}">
      <label for="choice{{ forloop.counter }}">{{ choice.choice_text }}</label><br>
    {% endfor %}
  </fieldset>
  <input type="submit" value="Vote">
</form>
```

Quelques précisions : le gabarit `{% csrf_token %}` est à utiliser dès lors que vous mettez en place un formulaire de type post. La variable `{{ forloop.counter }}` permet de récupérer le numéro de tour de boucle, on s'en sers ici pour générer des id différents pour nos boutons radios. La variable `{{ error_message }}` est une variable que nous allons créer dans notre vue si l'utilisateur clique sur *Vote* sans avoir sélectionné de choix. Voyons maintenant notre vue

Suivez-nous sur LinkedIn Twitter **@Adrar_Numerique**

... et sur le web : **www.adrar-numerique.com**

Voici donc notre vue est les imports nécessaire dans notre fichier jusqu'à maintenant :

```
from django.shortcuts import render, get_object_or_404
from django.http import HttpResponseRedirect, HttpResponseRedirect
from django.urls import reverse
from .models import Question, Choice
```

render et *get_object_or_404* sont nos raccourci de syntaxe
HttpResponse est notre ancienne méthode utilisé sur la vue *results*
HttpResponseRedirect est la méthode à utiliser lors d'une redirection d'une méthode post et *reverse* permet de rediriger vers une page nécessitant une partie variable.

```
def vote(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    try:
        selected_choice = question.choice_set.get(pk=request.POST['choice'])
    except (KeyError, Choice.DoesNotExist):
        return render(request, 'sondage/detail.html', {'question': question,
                                                         'error_message': 'Vous devez choisir!'},)
    else:
        selected_choice.votes += 1
        selected_choice.save()
        return HttpResponseRedirect(reverse('sondage:results', args=(question_id,)))
```

Ici nous voyons l'utilisation de *request.POST* permettant de récupérer la valeur passé via un formulaire de type POST.
 Nous gérons ensuite dans le try/except le cas où l'utilisateur ne clique pas sur un choix avant de valider.
 Nous incrémentons la valeur du choix choisi et redirigeons l'utilisateur.

Suivez-nous sur LinkedIn  Twitter  @Adrar_Numerique

... et sur le web : www.adrar-numerique.com

Voici à quoi ressemble donc notre formulaire :

Quel langage de programmation préférez vous ?

- ☐ Python
- ☐ Php

Vote

Quel langage de programmation préférez vous ?

Vous devez choisir!

- ☐ Python
- ☐ Php

Vote

Suivez-nous sur LinkedIn Twitter @Adrar_Numerique

... et sur le web : www.adrar-numerique.com

Gérons maintenant la page d'affichage des votes de nos sondages en redéfinissant sa vue et en créant son template:

```
def results(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'sondage/results.html', {'question': question})
```

```
<!-- HTML de base -->
<h1>{{ question.question_text }}</h1>

<ul>
    {% for choice in question.choice_set.all %}
        <li>{{ choice.choice_text }} -- {{ choice.votes }} vote{{ choice.votes|pluralize }}</li>
    {% endfor %}
</ul>

<a href="{% url 'sondage:detail' question.id %}">Voter à nouveau?</a>
```

Ici nous pouvons voir quelques chose d'intéressant : la variable `{{ choice.votes|pluralize }}` permettant de rajouter un 's' si il y a plusieurs votes pour ce choix.

Et nous rajoutons un liens vers la page de détail de la question pour voter à nouveau

Suivez-nous sur LinkedIn  Twitter  @Adrar_Numerique

... et sur le web : www.adrar-numerique.com

Et voila à quoi ressemble notre pages de résultats de votes

Quel langage de programmation préférez vous ?

- Python -- 2 votes
- Php -- 0 votes

[Voter à nouveau?](#)