Django form view

ways to send information to server¶

- Can send the data in HTTP headers
- Can send the data in HTTP request params
- Can send the data in HTTP request body.

What is a form?¶

- A form is used to send the information to the server
- `GET` and `POST` are the only HTTP methods to use when dealing with forms.

Django forms¶

- Django provides a way to create the form fields in a pythonic way with/without form context
- Django forms validates the data when a data is submitted to the server.
- Django forms also helpful in rendering the error or validation messages if submitted data is not in the required format.

django contact form¶

- Let's create a simple django contact form.
- Create a `forms.py` file in the *django app* `my_app` and add the below code.

my_app/forms.py

```
from django import forms

class ContactForm(forms.Form):
    first_name = forms.CharField(max_length=10)
    last_name = forms.CharField(max_length=10)
    email = forms.EmailField(max_length=100)
```

django form as HTML¶

- Can generate `HTML` from the above django form.
- Now, open django shell generate HTML from django form with below code.

```
form = ContactForm()
print(form.as_div())
```

- It will generate below HTML

```
<div>
  <label for="id_first_name">First name:</label>
  <input type="text" name="first_name" maxlength="10" required
id="id_first_name">
</div>
<div>
  <label for="id_last_name">Last name:</label>
  <input type="text" name="last_name" maxlength="10" required
id="id_last_name">
</div>
<div>
  <label for="id_email">Email:</label>
  <input type="email" name="email" maxlength="10" required id="id_email">
</div>
```

- we can use same HTML if the django template just like `{{ form.as_div }}`
- similarly, we can use other form methods like `form.as_p()`, `form.as_ul()` and `form.as_table()`
- we render the template with form fields for `GET` request.

Django Form for data validation¶

- Django forms can be used to validate the form data as well.
- Look at the below code.

```
data = {}
form = ContactForm(data)
print(form.is_valid())
# output: False
print(form.errors)
# output: <ul class="errorlist"><li>first_name<ul class="errorlist"><li>This
field is required.</li></ul></li><li>last_name<ul class="errorlist"><li>This
field is required.</li></ul></li><li>email<ul class="errorlist"><li>This
field is required.</li></ul></li></ul>
print(form.errors.as_json())
# output: {"first_name": [{"message": "This field is required.", "code":
"required"}], "last_name": [{"message": "This field is required.", "code":
"required"}], "email": [{"message": "This field is required.", "code":
"required"}]}
```

- `form.is_valid()` method is used to check whether the form data is valid or not.
- `form.errors`, `form.errors.as_json()` are used for form errors.

- Let's use the valid data

```
data = {"email": "john@example.com", "first_name": "John", "last_name":
"Doe"}
form = ContactForm(data)
print(form.is_valid())
# output: True
print(form.cleaned_data)
# output: {'first_name': 'John', 'last_name': 'Doe'}
```

- `form.cleaned_data` returns the validated data.

Django form view¶

function based form view¶

- we can use the forms with function based views for simple functionality.
- lets write a simple form view to get the contact information.
- we will be using the `ContactForm` mentioned in this article.
- look at the below code

```
from django.http import HttpResponse
from django.shortcuts import render
from my_app.forms import ContactForm

def my_form_view(request):
    if request.method.lower() == "POST":
        form = ContactForm(request.POST)
        if form.is_valid():
            data = form.cleaned_data
            print(data)
            return HttpResponse("successfully submitted the form")
    else:
        form = ContactForm()
    context = {"form": form}
    return render(request, 'form_template.html', context)
```

form_template.html

```
<form action="" method="POST" enctype="multipart/form-data">
    {% csrf_token %}
    {{ form.as_table }}
    <input type="submit" value="submit" />
</form>
```

class based form view¶

```
from django.http import HttpResponse
from django.views.generic.edit import FormView
from my_app.forms import ContactForm

class MyFormView(FormView):
    template_name='form_template.html'
    form_class=ContactForm

    def form_valid(self, form):
      data = form.cleaned_data
      print(data)
      return HttpResponse("successfully submitted the form")
```

- `FormView` is generic class provided by django.
- It provides the generic functionality for form views.
- It comes with methods like `form_valid` and `form_invalid` to handle the requests.
- Only use it if your functionality matches with `FormView` implementation.

configure urls¶

- open `my_app/urls.py` and add below code to it.

```
from django.urls import path
from my_app import views

urlpatterns = [
    # ...
    path('fbv_form_view', views.my_form_view, name='fmy_form_view'),
    path('cbv_form_view', views.MyFormView.as_view(), name='cmy_form_view'),
    # ...
]
```

- Access the url http://localhost:8000/my_app/fbv_form_view to see the function based form view
- Access the url http://localhost:8000/my_app/cbv_form_view to see the class based form view
- Once open the form urls, fill out the form and submit the form if the form is valid then we can see the response "successfully submitted the form" otherise it shows the form with error messages.

References¶

- https://docs.djangoproject.com/en/dev/topics/forms/
- https://docs.djangoproject.com/en/dev/ref/forms/fields/
- https://docs.djangoproject.com/en/dev/topics/class-based-views/generic-editing/