Django update view

Django Update View¶

- Django update view is used to update a single record of the database table.
- It shows an editable form to the user with the current row data.
- User can make the changes to the form and on submit it then it will save the data to the database table.

Contact Model¶

my_app/models.py

```
from django.db import models

class Contact(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    email = models.EmailField(max_length=255)
    phone = models.CharField(max_length=10, null=True)

    class Meta:
        db_table = "contact"
```

form template¶

templates/form_template.py

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

function based update view¶

- It's simple function handles both `GET` and `POST` requests
  - `GET` - displays the form to the user.
  - `POST` - recieves the user submitted info
- Let's looke at code below.

```
from django.http import HttpResponse
from django.shortcuts import render, get_object_or_404
from my_app.models import Contact
from my_app.forms import ContactModelForm


def fbv_update_view(request, pk):
    obj = get_object_or_404(Contact, id=pk)
    if request.method.lower() == 'post':
        form = ContactModelForm(request.POST, instance=obj)
        if form.is_valid():
            form.save()
            return HttpResponse("<h1>Updated</h1>")
    else:
        form = ContactModelForm(instance=obj)
    context = {'form': form}
    return render(request, 'form_template.html', context)
```

- `get_object_or_404` gets the object from the db if found otherwise throws HTTP 404 error.
- For `GET` request
    - render the html template with form
    - `ContactModelForm(instance=obj)` loads the initial data
- For `POST` request
    - validates the form `ContactModelForm(request.POST, instance=obj)`
    - save the form if it's valid otherwise renders the form template with errors.

class based update view¶

- Django provides a generic view `UpdateView` to update the table row.

```
from django.http import HttpResponse
from django.views.generic.edit import UpdateView
from my_app.models import Contact
from my_app.forms import ContactModelForm

class CBVUpdateView(UpdateView):
    form_class = ContactModelForm
    template_name = 'form_template.html'
    pk_url_kwarg = 'pk'
    model = Contact

    def form_valid(self, form):
        form.save()
        return HttpResponse("<h1>Updated</h1>")
```

- Django provides the generic view class `UpdateView`.
- It contains the inbuilt funcitonality which we implemented in the function based view.
- In order to work with `UpdateView` we need to provider attributes `form_class`, `template_name`, `pk_url_kwarg` and `model` or methods `get_object()` or `get_queryset()`
    - `template_name` - used to render the template
    - `pk_url_kwarg` - used to identify the object in the database
    - `model` or `get_object()` or `get_queryset()` - used to get the object using `pk_url_kwarg`

configure the 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_update_view/<int:pk>', views.fbv_update_view,
name='fbv_update_view'),
    path('cbv_update_view/<int:pk>', views.CBVUpdateView.as_view(),
name='cbv_update_view'),
    # ...
]
```

- Open url http://127.0.0.1:8000/my_app/fbv_update_view/1 to update the object using function based update view.
- Open url http://127.0.0.1:8000/my_app/cbv_update_view/1 to update the object using class based update view.
- Replace the `1` with any id in the contact table.
- It throws `404` status code if id does not exist in the table.

References¶

- https://docs.djangoproject.com/en/dev/ref/class-based-views/generic-editing/#django.views.generic.edit.UpdateView
- https://ccbv.co.uk/projects/Django/4.1/django.views.generic.edit/UpdateView/