Forms, ModelForms e Views Genéricas de Edição

Yuri Kaszubowski Lopes

Especialização em Tecnologia Python para Negócios

Métodos HTTP

- Requisições para uma mesma URL podem resultar em diferentes operações
- A arquitetura é que uma URL representa um objeto, e dependendo do método fazemos algo diferente com ele
  - GET: Ler
  - POST: Criar
  - PUT: Atualizar/Trocar
  - PATCH: Atualizar/Modificar
  - DELETE: Remover
- Na prática necessitamos mais requisições por objeto do que a quantidade de método
  - E.g., requisitar detalhes, requisitar um formulário para inserir/atualizar e postar o formulário preenchido com os valores
- Métodos mais usados GET e POST com várias URLs para um mesmo objeto

Django Forms

- Queremos aceitar dados do usuário
- Em HTML usamos `<form></form>`
  - Entrar texto, selecionar opções, ...
  - Itens simples tag HTML
  - Itens complexos (e.g. selecionar data) pode requerer JS e CSS
- O Django trás várias facilidades para trabalhar com forms
  - Usa apenas requisições com métodos GET e POST
  - GET: Carrega o HTML do form para o cliente
  - POST: Envia os dados do formulário
    - Usualmente mesma URL do GET
    - Uma URL para criar, outra para atualizar, outra excluir, ...

URL e imports da views.py

- Adicione o URL path abaixo

server/seriados/urls.py

```
1     path('serie/inserir/', views.serie_insert, name='serie_insert'),
```

- Atualize o início de views.py para incluir as importações abaixo
  - HttpResponseRedirect, reverse, DetailView, CreateView, DeleteView, UpdateView

server/seriados/views.py

```
1  from django.http import HttpResponse, HttpResponseRedirect
2  from django.shortcuts import render, get_object_or_404
3  from django.forms.models import model_to_dict
4  from django.urls import reverse
5
6  from django.views import View
7  from django.views.generic import TemplateView, ListView, DetailView
8  from django.views.generic.edit import CreateView, DeleteView, UpdateView
9
10 from .models import Serie, Temporada, Episodio
```

Criando o código de um formulário

server/seriados/forms.py

```
1 from django import forms
2
3 from .models import Serie, Temporada, Episodio
4
5 class SerieForm(forms.Form):
6     nome = forms.CharField(label="Nome da Série", max_length=70)
```

Template de um formulário base

server/seriados/templates/form_base.html

```
1 {% extends "base.html" %}
2
3 {% block content %}
4 <form action={% url target_url %} method="post">
5     {% csrf_token %}
6     {{ form }}
7     <input type="submit" value="Enviar">
8 </form>
9 {% endblock %}
```

View para tratar form

server/seriados/views.py

```
1  from .forms import SerieForm
2
3  ...
4
5  def serie_insert(request):
6      if request.method == 'GET':
7          form = SerieForm()
8      elif request.method == 'POST':
9          form = SerieForm(request.POST)
10         if form.is_valid():
11             nome = form.cleaned_data['nome']
12             obj = Serie(nome = nome)
13             obj.save()
14             return HttpResponseRedirect(reverse(
15                 'seriados:serie_details',
16                 kwargs = {'pk': obj.pk}
17             ))
18
19     return render(request, 'form_base.html', {
20         'form': form,
21         'target_url': 'seriados:serie_insert',
22     })
```

Model Form

server/seriados/forms.py

```
1 class TemporadaForm(forms.ModelForm):
2     class Meta:
3         model = Temporada
4         fields = ['numero', 'serie']
```

URL path para temporadas

server/seriados/urls.py

```
1     path('temporada/', views.TemporadaListView.as_view(),
2         name='temporada_list'),
3     path('temporada/<int:pk>/', views.TemporadaDetail.as_view(),
4         name='temporada_details'),
5     path('temporada/inserir/', views.TemporadaCreateView.as_view(),
6         name='temporada_insert'),
```

Todos os modelos devem ter o método `get_absolute_url`

views path para temporadas

server/seriados/views.py

```
1 class TemporadaListView(ListView):
2     template_name = 'temporada_list.html'
3     model = Temporada
4
5
6 class TemporadaDetail(DetailView):
7     template_name = "temporada_details.html"
8     model = Temporada
9
10
11 class TemporadaCreateView(CreateView):
12     template_name = 'form_generic.html'
13     form_class = TemporadaForm
```

Templates para Temporada

server/seriados/templates/temporada_list.html

```
1 {% extends "base.html" %}
2
3 {% block content %}
4 <h1>Lista de Temporadas</h1>
5
6 <table border="1px">
7     <tr>
8         <th>Série</th>
9         <th>Temporada</th>
10    </tr>
11
12    {% for object in object_list %}
13        <tr>
14            <td>{{ object.serie.nome }}</td>
15            <td>{{ object.numero }}</td>
16        </tr>
17    {% endfor %}
18 </table>
19 {% endblock %}
```

Templates para Temporada

server/seriados/templates/temporada_details.html

```
1 {% extends "base.html" %}
2
3 {% block content %}
4     <b>Série</b>: {{ object.serie.nome }} <br />
5     <b>Temporada</b>: {{ object.numero }} <br />
6 {% endblock %}
```

Templates para Temporada

server/seriados/templates/form_generic.html

```
1 {% extends "base.html" %}
2
3 {% block content %}
4 <form action="." method="post">
5     {% csrf_token %}
6     {{ form }}
7     <input type="submit" value="Enviar">
8 </form>
9 {% endblock %}
```

- Única diferença para `form_base.html` é o valor do atributo `action` na tag `form`

ModelForm automático

- Não necessitamos criar o model form, a view genérica de edição pode fazer

server/seriados/urls.py

```
1     path('episodio/inserir/', views.EpisodioCreateView.as_view(),
2         name='episodio_insert'),
```

server/seriados/views.py

```
1 class EpisodioCreateView(CreateView):
2     template_name = 'form_generic.html'
3     model = Episodio
4     fields = ['temporada', 'data', 'titulo']
```

Edição

server/seriados/urls.py

```
1     path('temporada/<int:pk>/editar/', views.TemporadaUpdateView.as_view(),
2         name='temporada_update'),
```

server/seriados/views.py

```
1 class TemporadaUpdateView(UpdateView):
2     template_name = 'form_generic.html'
3     model = Temporada
4     fields = ['serie', 'numero']
```

Excluir

server/seriados/urls.py

```
1     path('temporada/<int:pk>/excluir/', views.TemporadaDeleteView.as_view(),
2         name='temporada_excluir'),
```

server/seriados/views.py

```
1 class TemporadaDeleteView(DeleteView):
2     template_name = "temporada_confirm_delete.html"
3     model = Temporada
4
5     def get_success_url(self):
6         return reverse('seriados:temporada_list')
```

Templates para excluir temporada

server/seriados/templates/temporada_confirm_delete.html

```
1 {% extends "base.html" %}
2
3 {% block content %}
4     Quer deletar a temporáda {{ object.numero }}
5     da Série {{ object.serie.nome }}? <br />
6
7     <form action="." method="post">
8         {% csrf_token %}
9         <input type="submit" value="Delete">
10    </form>
11 {% endblock %}
```

Forms, ModelForms e Views Genéricas de Edição

Yuri Kaszubowski Lopes

Especialização em Tecnologia Python para Negócios