DJANGO DEPENDENT DROP DOWN LIST WITH JQUERY HTMX

1 . models.py

from django.db import models  
  
  
class Country(models.Model):  
  
 id = models.BigAutoField(primary\_key=True)  
  
 name = models.CharField(max\_length=255, unique=True)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 class Meta:  
 db\_table = 'country'

class State(models.Model):  
  
 id = models.BigAutoField(primary\_key=True)  
  
 country = models.ForeignKey(Country, on\_delete=models.SET\_NULL, blank=True, null=True)  
  
 name = models.CharField(max\_length=255, unique=True)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 class Meta:  
 db\_table = 'state'

class City(models.Model):  
  
 id = models.BigAutoField(primary\_key=True)  
  
 state = models.ForeignKey(State, on\_delete=models.SET\_NULL, blank=True, null=True)  
  
 name = models.CharField(max\_length=255, unique=True)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 class Meta:  
 db\_table = 'city'

class Person(models.Model):  
  
 id = models.BigAutoField(primary\_key=True)  
  
 name = models.CharField(max\_length=255)  
  
 country = models.ForeignKey(Country, on\_delete=models.SET\_NULL, blank=True, null=True)  
  
 state = models.ForeignKey(State, on\_delete=models.SET\_NULL, blank=True, null=True)  
  
 city = models.ForeignKey(City, on\_delete=models.SET\_NULL, blank=True, null=True)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 class Meta:  
 db\_table = 'person'

2 – forms.py

from django import forms  
  
from .models import Person, State, City  
  
  
class PersonCreationForm(forms.ModelForm):  
 class Meta:  
 model = Person  
 fields = '\_\_all\_\_'  
  
 def \_\_init\_\_(self, \*args, \*\*kwargs):  
 super().\_\_init\_\_(\*args, \*\*kwargs)  
   
 self.fields['state'].queryset = State.objects.none()  
   
 self.fields['city'].queryset = City.objects.none()  
  
 if 'country' in self.data:  
 try:  
 country\_id = int(self.data.get('country'))  
   
 self.fields['state'].queryset = State.objects.filter(country\_id=country\_id).order\_by('name')  
   
 except (ValueError, TypeError):  
 *# invalid input from the client; ignore and fallback to empty City queryset* pass   
 elif self.instance.pk:  
   
 self.fields['state'].queryset = self.instance.country.state\_set.order\_by('name')  
  
 if 'state' in self.data:  
 try:  
 state\_id = int(self.data.get('state'))  
   
 self.fields['city'].queryset = City.objects.filter(state\_id=state\_id).order\_by('name')  
   
 except (ValueError, TypeError):  
 *# invalid input from the client; ignore and fallback to empty City queryset* pass   
 elif self.instance.pk:  
   
 self.fields['city'].queryset = self.instance.country.city\_set.order\_by('name')

3 – views.py

from django.http import JsonResponse  
from django.shortcuts import render, redirect, get\_object\_or\_404  
  
from .forms import PersonCreationForm  
from .models import Person, State, City  
  
  
def person\_create\_view(request):  
 form = PersonCreationForm()  
 if request.method == 'POST':  
 form = PersonCreationForm(request.POST)  
 if form.is\_valid():  
 form.save()  
 return redirect('person\_add')  
 return render(request, 'backend/home.html', {'form': form})  
  
  
def person\_update\_view(request, pk):  
 person = get\_object\_or\_404(Person, pk=pk)  
 form = PersonCreationForm(instance=person)  
 if request.method == 'POST':  
 form = PersonCreationForm(request.POST, instance=person)  
 if form.is\_valid():  
 form.save()  
 return redirect('person\_change', pk=pk)  
 return render(request, 'backend/home.html', {'form': form})

*# AJAX*def load\_states(request):  
 country\_id = request.GET.get('country\_id')  
  
 context = {  
 "states": State.objects.filter(country\_id=country\_id).all()  
 }  
  
 *# http://127.0.0.1:8000/ajax/load-states/?country\_id=2  
 # return JsonResponse(list(states.values('id', 'name')), safe=False)* return render(request, 'backend/state\_dropdown\_list\_options.html', context)  
  
*# AJAX*def load\_cities(request):  
 state\_id = request.GET.get('state\_id')  
  
 context = {  
 "cities": City.objects.filter(state\_id=state\_id).all()  
 }  
  
 *# http://127.0.0.1:8000/ajax/load-cities/?state\_id=2  
 # return JsonResponse(list(cities.values('id', 'name')), safe=False)* return render(request, 'backend/city\_dropdown\_list\_options.html', context)

4 – urls.py

from django.urls import path  
  
from . import views  
  
urlpatterns = [  
 path('add/', views.person\_create\_view, name='person\_add'),  
 path('<int:pk>/', views.person\_update\_view, name='person\_change'),  
  
 path('ajax/load-states/', views.load\_states, name='ajax\_load\_states'), *# AJAX* path('ajax/load-cities/', views.load\_cities, name='ajax\_load\_cities'), *# AJAX*]

5 - state\_dropdown\_list\_options.html

<option value="">---------</option>  
{% for state in states %}  
<option value="{{ state.id }}">{{ state.name }}</option>  
{% endfor %}

6 - city\_dropdown\_list\_options.html

<option value="">---------</option>  
{% for city in cities %}  
<option value="{{ city.pk }}">{{ city.name }}</option>  
{% endfor %}

7 - home.html

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Dependent Dropdown in Django</title>  
</head>  
<body>  
<h2>Person Form</h2>  
  
<form method="post" id="personForm" data-states-url="{% url 'ajax\_load\_states' %}" data-cities-url="{% url 'ajax\_load\_cities' %}">  
 {% csrf\_token %}  
 {{ form.as\_p }}  
 <input type="submit" value="Submit">  
</form>  
  
<script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>

<script>  
$("#id\_country").change(function () {  
 // get the url of the `load\_cities` view  
 const url = $("#personForm").attr("data-states-url");  
  
 // get the selected country ID from the HTML input  
 const countryId = $(this).val();  
  
 // initialize an AJAX request  
 $.ajax({  
 // set the url of the request   
 url: url,  
 data: {  
 // add the country id to the GET parameters  
 'country\_id': countryId  
 },  
 // `data` is the return of the `load\_cities` view function  
 success: function (data) {  
 // replace the contents of the city input with the data that came from the server  
 $("#id\_state").html(data);  
 }  
 });  
  
});

$("#id\_state").change(function () {  
 // get the url of the `load\_cities` view  
 const url = $("#personForm").attr("data-cities-url");  
  
 // get the selected State ID from the HTML input  
 const stateId = $(this).val();  
  
 // initialize an AJAX request  
 $.ajax({  
 // set the url of the request   
 url: url,  
 data: {  
 // add the state id to the GET parameters  
 'state\_id': stateId  
 },  
 // `data` is the return of the `load\_cities` view function  
 success: function (data) {  
 // replace the contents of the city input with the data that came from the server  
 $("#id\_city").html(data);   
 }  
 });  
  
});  
</script>  
  
</body>  
</html>