Update View – Function based Views Django

Last Updated : 17 May, 2021

Update View refers to a view (logic) to update a particular instance of a table from the database with some extra details. It is used to update entries in the database for example, updating an article at geeksforgeeks. So Update view must display the old data in the form and let user update the data from there only. Django provides extra-ordinary support for Update Views but let's check how it is done manually through a function-based view. This article revolves around Update View which involves concepts such as Django Forms, Django Models.

For Update View, we need a project with some models and multiple instances which will be displayed. Basically, Update view is a combination of Detail view

Django Update View – Function Based Views

Illustration of **How to create and use Update view** using an Example. Consider a project named geeksforgeeks having an app named geeks.

> *Refer to the following articles to check how to create a project and an app in Django.*

- *How to Create a Basic Project using MVT in Django?*
- *How to Create an App in Django ?*

After you have a project and an app, let's create a model of which we will be creating instances through our view. In geeks/models.py,

Python3

```
# import the standard Django Model
# from built-in library
from django.db import models

# declare a new model with a name "GeeksModel"
class GeeksModel(models.Model):

    # fields of the model
    title = models.CharField(max_length = 200)
    description = models.TextField()

    # renames the instances of the model
    # with their title name
    def __str__(self):
        return self.title
```

After creating this model, we need to run two commands in order to create Database for the same.

```
Python manage.py makemigrations
Python manage.py migrate
```

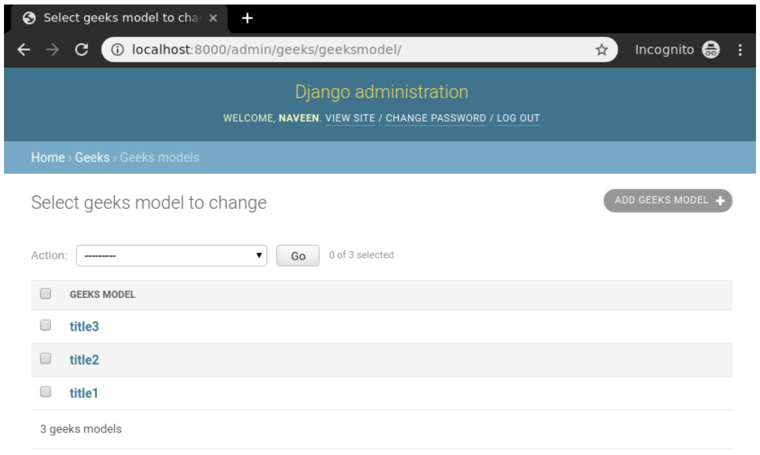
Now let's create some instances of this model using shell, run form bash,

```
Python manage.py shell
```

Enter following commands

```
>>> from geeks.models import GeeksModel
>>> GeeksModel.objects.create(
                       title="title1",
                       description="description1").save()
>>> GeeksModel.objects.create(
                       title="title2",
                       description="description2").save()
>>> GeeksModel.objects.create(
                       title="title2",
                       description="description2").save()
```

Now we have everything ready for back end. Verify that instances have been created from http://localhost:8000/admin/geeks/geeksmodel/


Select geeks model to cha
localhost:8000/admin/geeks/geeksmodel/
Incognito
Django administration
WELCOME, NAVEEN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Geeks › Geeks models
Select geeks model to change
ADD GEEKS MODEL
Action:
Go
0 of 3 selected
GEEKS MODEL
title3
title2
title1
3 geeks models

Now we will create a Django ModelForm for this model. Refer this article for more on modelform – Django ModelForm – Create form from Models. create a file forms.py in geeks folder,

Python3

```python
from django import forms
from .models import GeeksModel


# creating a form
class GeeksForm(forms.ModelForm):

    # create meta class
    class Meta:
        # specify model to be used
        model = GeeksModel

        # specify fields to be used
        fields = [
            "title",
            "description"]
```

For Update_view one would need some identification to get a particular instance of the model. Usually it is unique primary key such as **id**. To specify this identification we need to define it in urls.py. Go to geeks/urls.py,

Python3

```
from django.urls import path

# importing views from views..py
from .views import update_view, detail_view

urlpatterns = [
    path('<id>/', detail_view ),
    path('<id>/update', update_view ),
]
```

Let's create these views with explanations. In geeks/views.py,

Python3

```
from django.shortcuts import (get_object_or_404,
                              render,
                              HttpResponseRedirect)

# relative import of forms
from .models import GeeksModel
from .forms import GeeksForm

# after updating it will redirect to detail_View
def detail_view(request, id):
    # dictionary for initial data with
    # field names as keys
    context ={}

    # add the dictionary during initialization
    context["data"] = GeeksModel.objects.get(id = id)

    return render(request, "detail_view.html", context)

# update view for details
def update_view(request, id):
    # dictionary for initial data with
    # field names as keys
    context ={}

    # fetch the object related to passed id
    obj = get_object_or_404(GeeksModel, id = id)

    # pass the object as instance in form
    form = GeeksForm(request.POST or None, instance = obj)

    # save the data from the form and
    # redirect to detail_view
    if form.is_valid():
        form.save()
        return HttpResponseRedirect("/"+id)

    # add form dictionary to context
    context["form"] = form

    return render(request, "update_view.html", context)
```

Now create following templates in templates folder,
In geeks/templates/update_view.html,

HTML

```
<div class="main">
    <!-- Create a Form -->
    <form method="POST">
        <!-- Security token by Django -->
        {% csrf_token %}

        <!-- form as paragraph -->
        {{ form.as_p }}

        <input type="submit" value="Update">
    </form>

</div>
```

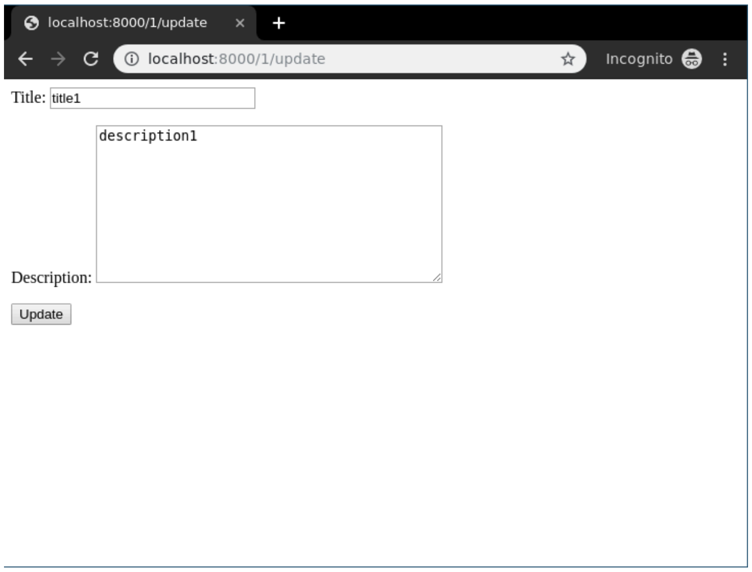
In geeks/templates/detail_view.html,

HTML

```
<div class="main">
    <!-- Display attributes of instance -->
```

```
        {{ data.title }} <br/>
        {{ data.description }}
    </div>
```

Let's check if everything is working, visithttp://localhost:8000/1/update.

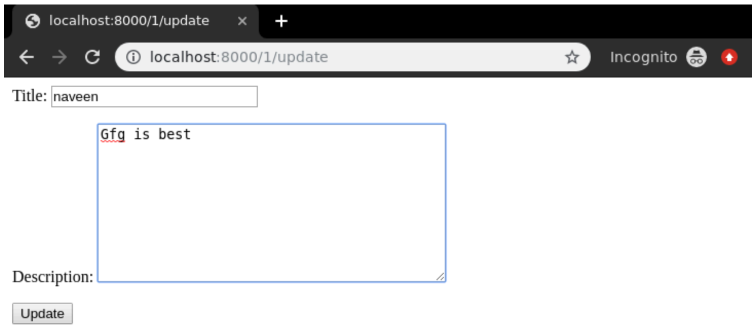


Title: title1

Description: description1

Update

Here you can see the form with data already filled from the instance, Now one can edit this data and update it easily, let's check it out

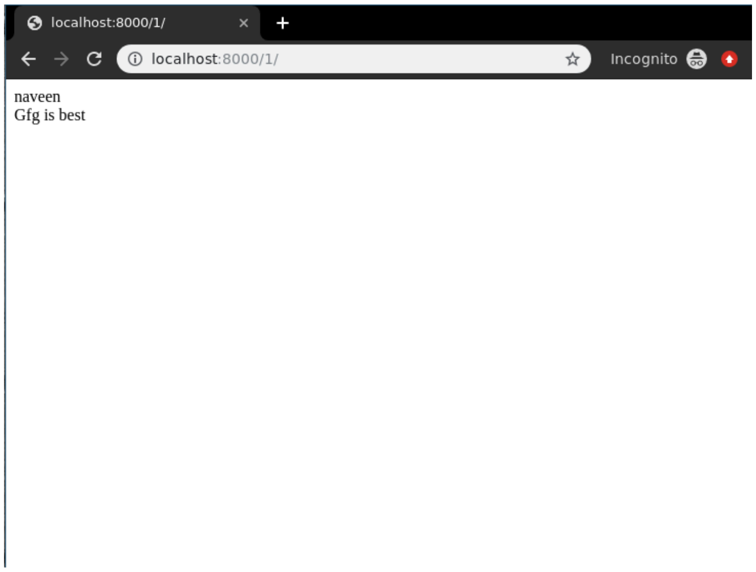


Title: naveen

Description: Gfg is best

Update

Hit update and done.



```
naveen
Gfg is best
```

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our Mastering Django Framework - Beginner to Advanced Course on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through **hands-on projects** and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

N Nave... 10

Previous Article

Detail View - Function based Views Django

Next Article

Delete View - Function based Views Django

Similar Reads

Create View - Function based Views Django

Create View refers to a view (logic) to create an instance of a table in the database. It is just like taking an input from a user and storing it in a specified...

3 min read

Detail View - Function based Views Django

Detail View refers to a view (logic) to display a particular instance of a table from the database with all the necessary details. It is used to display multiple types o...

3 min read

Delete View - Function based Views Django

Delete View refers to a view (logic) to delete a particular instance of a table from the database. It is used to delete entries in the database for example, deleting a...

3 min read

List View - Function based Views Django

List View refers to a view (logic) to list all or particular instances of a table from the database in a particular order. It is used to display multiple types of data on ...

3 min read

Class Based vs Function Based Views - Which One is Better to Use in Django?

Django...We all know the popularity of this python framework all over the world. This framework has made life easier for developers. It has become easier for...

7 min read

Function based Views - Django Rest Framework

Django REST Framework allows us to work with regular Django views. It facilitates processing the HTTP requests and providing appropriate HTTP...

13 min read

Django Function Based Views

Django is a Python-based web framework which allows you to quickly create web application without all of the installation or dependency problems that you...

7 min read

Createview - Class Based Views Django

Create View refers to a view (logic) to create an instance of a table in the database. We have already discussed basics of Create View in Create View –...

3 min read

ListView - Class Based Views Django

List View refers to a view (logic) to display multiple instances of a table in the database. We have already discussed the basics of List View in List View –...

4 min read

UpdateView - Class Based Views Django

UpdateView refers to a view (logic) to update a particular instance of a table from the database with some extra details. It is used to update entries in the databas...

3 min read

Article Tags : Python Django-views Python Django

Practice Tags : python

In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles