

# Django: Todo App Setup Guide (for Windows)

## 1. INSTALL PYTHON AND DJANGO

-----

- Install Python from <https://www.python.org/>.
- Make sure to check the box to add Python to your system's PATH during installation.
- Open a new Command Prompt or PowerShell and install Django:

```
pip install django
```

- Verify the installation:

```
django-admin --version
```

```
python --version
```

## 2. CREATE A NEW DJANGO APP

-----

- Create a new Django project and navigate into it:

```
django-admin startproject taskmanager
```

```
cd taskmanager
```

- Create a new app named 'tasks':

```
python manage.py startapp tasks
```

- Add 'tasks' to the INSTALLED\_APPS list in taskmanager/settings.py:

```
INSTALLED_APPS = [
```

```
    ...
```

```
    'tasks',
```

```
]
```

## 3. CREATE A TASK MODEL

-----

- Open tasks/models.py and define the Task model:

```
from django.db import models

class Task(models.Model):

    title = models.CharField(max_length=100)

    description = models.TextField()

    created_at = models.DateTimeField(auto_now_add=True)

    completed = models.BooleanField(default=False)

    def __str__(self):

        return self.title
```

- Run the following command to create the migration for the new model:

```
python manage.py makemigrations tasks
```

- Apply the migration to create the database table:

```
python manage.py migrate
```

#### 4. CREATE A FORM FOR TASKS

-----

- In tasks/forms.py, define the form for adding a new task:

```
from django import forms

from .models import Task

class TaskForm(forms.ModelForm):

    class Meta:

        model = Task

        fields = ['title', 'description']
```

#### 5. CREATE A VIEW FOR TASKS

-----

- Open tasks/views.py and add views to list tasks, create a new task, and toggle the completion

status:

```
from django.shortcuts import render, redirect

from .models import Task

from .forms import TaskForm

def task_list(request):

    tasks = Task.objects.all()

    return render(request, 'tasks/task_list.html', {'tasks': tasks})

def add_task(request):

    if request.method == 'POST':

        form = TaskForm(request.POST)

        if form.is_valid():

            form.save()

            return redirect('task_list')

    else:

        form = TaskForm()

        return render(request, 'tasks/add_task.html', {'form': form})

def task_complete(request, pk):

    task = Task.objects.get(id=pk)

    task.completed = True

    task.save()

    return redirect('task_list')

def task_delete(request, pk):

    task = Task.objects.get(id=pk)

    task.delete()

    return redirect('task_list')
```

## 6. SET UP URLS

- 
- In tasks/urls.py, create the URLs for the views:

```
from django.urls import path

from .views import task_list, add_task, task_complete, task_delete

urlpatterns = [

    path('', task_list, name='task_list'),

    path('add/', add_task, name='add_task'),

    path('complete/<int:pk>/', task_complete, name='complete_task'),

    path('delete/<int:pk>/', task_delete, name='delete_task'),

]
```

- In taskmanager/urls.py, include the tasks app URLs:

```
from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', include('tasks.urls')),

]
```

## 7. CREATE BASIC VIEWS

-----

- Create task\_list.html inside tasks/templates/tasks/:

```
<!DOCTYPE html>

<html>

<head>

    <title>Task List</title>

</head>
```

```

<body>

  <h1>Task Manager</h1>

  <form method="post">

    {% csrf_token %}

    <input type="text" name="title" placeholder="Enter task..." required>

    <textarea name="description" placeholder="Enter description"></textarea>

    <button type="submit">Add Task</button>

  </form>

  <ul>

    {% for task in tasks %}

      <li>

        {{ task.title }} - {{ task.completed|yesno:"Completed,Pending" }}

        <a href="{% url 'complete_task' task.id %}">Mark as Completed</a> |

        <a href="{% url 'delete_task' task.id %}">Delete</a>

      </li>

    {% endfor %}

  </ul>

</body>

</html>

```

- Create add\_task.html inside tasks/templates/tasks/:

```

<!DOCTYPE html>

<html>

<head>

  <title>Add Task</title>

</head>

<body>

```

```
<h1>Add a New Task</h1>

<form method="post">

    {% csrf_token %}

    {{ form.as_p }}

    <button type="submit">Save Task</button>

</form>

<br>

<a href="{% url 'task_list' %}">Back to Task List</a>

</body>

</html>
```

## 8. START THE SERVER

-----

- Run the server:

```
python manage.py runserver
```

- Open the browser and visit <http://127.0.0.1:8000/> to see your Todo app in action.

Your Django Todo App is ready! You can now create, view, complete, and delete tasks.