

----- DJANGO TUTORIAL 7 -----

In this video I'll show you how to allow users to delete articles. I'll also cover a bunch of tests we can use to check those changes we made in the last few videos.

1. Create article_delete.html

```
{% extends 'base.html' %}
{% block content %}
<h1>Delete Article</h1>
<form action="" method="post">{% csrf_token %}
  <p>Do you want to delete
    "{{ article.title }}"?</p>
  <input type="submit" value="Delete" />
</form>
{% endblock content %}
```

2. Add the DeleteView to articles/views.py

```
from django.views.generic import ListView, DetailView, UpdateView, DeleteView
from django.urls import reverse_lazy
```

```
class ArticleDeleteView(DeleteView):
    model = Article
    template_name = 'article_delete.html'
    # Use reverse_lazy instead of reverse so that it won't
    # load the landing page until after the DB has been updated
    success_url = reverse_lazy('home')
```

3. Update articles/urls.py with the ArticleDeleteView url

```
from .views import HomePageView, ArticleDetailView, ArticleCreateView, ArticleUpdateView,
ArticleDeleteView
```

```
urlpatterns = [
    path('article/<int:pk>/delete/', ArticleDeleteView.as_view(), name='article_delete'),
```

4. Add a link to the article_delete.html page in the article_detail.html page

```
<a href="{% url 'article_edit' article.pk %}" class="btn btn-primary btn-sm">Edit Article</a>
<a href="{% url 'article_delete' article.pk %}" class="btn btn-primary btn-sm">Delete Article</a>
```

5. Add tests to verify we can create, update and delete articles in articles/tests.py

```
def test_article_creation(self):
    response = self.client.post(reverse('add_article'), {
        'title': 'Sample Title',
        'text': 'Sample Text',
        'author': self.user,
    })
    self.assertEqual(response.status_code, 200)
    self.assertContains(response, 'Sample Title')
    self.assertContains(response, 'Sample Text')

def test_article_update(self):
    response = self.client.post(reverse('article_edit', args='1'), {
        'title': 'New Title',
        'text': 'New Text',
    })
    # Confirm a 302 redirect
    self.assertEqual(response.status_code, 302)

def test_article_delete(self):
    response = self.client.get(reverse('article_delete', args='1'))
    self.assertEqual(response.status_code, 200)
```

Run tests python manage.py test