1. Install tinymce

Step 1: $ pip install django-tinymce

Step 2: INSTALLED\_APPS = (

...

'tinymce',

)

Step 3: urlpatterns = [

...

path('tinymce/', include('tinymce.urls')),

]

Step 4:

from django.db import models

from tinymce.models import HTMLField

class MyModel(models.Model):

...

content = HTMLField()

Challenges:

**Challenge:**

* Describe the specific problem you encountered, such as "TemplateSyntaxError: Invalid filter: 'length\_is'" in your Django project.
* Explain the context of the problem, like the specific template file or code snippet where the error occurred.
* Mention any relevant versions of Django, libraries, or other dependencies.

**Solution:**

* Provide a clear and concise explanation of the cause of the error.
* Detail the steps you took to resolve the issue, including the code changes made.
* Highlight the specific fix, such as replacing the length\_is filter with the count method.
* If applicable, share any additional tips or best practices related to the problem.

**Example:**

**Challenge:**

While working on a Django project using the material template library, I encountered a TemplateSyntaxError: Invalid filter: 'length\_is'. This error occurred in the fieldset.html template, preventing the correct rendering of form fields.

**Solution:**

The length\_is filter was deprecated in Django 4.2 and removed in Django 5.1. To resolve the issue, I replaced the length\_is filter with the count method to check the number of fields in a line. Here's the updated code snippet:

HTML

{% if line.fields.count == 1 %}

{{ line.errors }}

{% endif %}

Use code [with caution.](/faq#coding)

By using count, we can accurately determine if there's only one field in the line and display errors accordingly.

**Additional Tips:**

* Consider adding screenshots or code snippets to visually illustrate the problem and solution.
* If you encountered any specific error messages or logs, include them for reference.
* Test the solution thoroughly to ensure it works as expected.