

TD N° 1: Corrigé Type

Les opérations mathématiques avec Django

1. Créer un projet appelé `core`

```
(.venv) PS D:\calculator\django-admin startproject core .
```

2. Créer une application nommée `calculator`

```
(.venv) PS D:\calculator\python manage.py startapp calculator
```

3. Ajouter l'application `calculator` au projet

fichier `core/settings.py`

```
INSTALLED_APPS = [  
    ...  
    'calculator', # Ajoutez l'application ici à la suite des applications déjà présentes  
]
```

4. Configurer les URLs du projet pour inclure celles de l'application

fichier `core/urls.py`

```
from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('calculator.urls')), # Inclure les URLs de calculator  
]
```

5. Créer un fichier urls.py pour l'application calculator

fichier calculator/urls.py

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path('json/<str:operation>/<str:num1>/<str:num2>/', views.calculate_json, name='calculate_json'),
    path('html/<str:operation>/<str:num1>/<str:num2>/', views.calculate_html, name='calculate_html'),
]
```

6. Créer les vues pour JSON et HTML

fichier calculator/views.py

```
from django.http import JsonResponse
from django.shortcuts import render
```

Vue pour JSON

```
def calculate_json(request, operation, num1, num2):
    try:
        num1 = float(num1)
        num2 = float(num2)
        result = None
        if operation == 'add':
            result = num1 + num2
        elif operation == 'sub':
            result = num1 - num2
        elif operation == 'mul':
            result = num1 * num2
        elif operation == 'div':
            if num2 == 0:
                return JsonResponse({'error': 'Division by zero is not allowed.'}, status=400)
            result = num1 / num2
        else:
            return JsonResponse({'error': 'Invalid operation.'}, status=400)
        return JsonResponse({'operation': operation, 'num1': num1, 'num2': num2, 'result': result})
    except ValueError:
        return JsonResponse({'error': 'Invalid numbers.'}, status=400)
    except Exception as e:
        return JsonResponse({'error': str(e)}, status=500)
```

Vue pour HTML

```
def calculate_html(request, operation, num1, num2):
    try:
        num1 = float(num1)
        num2 = float(num2)
        result = None

        if operation == 'add':
            result = num1 + num2
        elif operation == 'sub':
            result = num1 - num2
        elif operation == 'mul':
            result = num1 * num2
        elif operation == 'div':
            if num2 == 0:
                return render(request, 'calculator/error.html', {'error': 'Division by zero is not allowed.'})
            result = num1 / num2
        else:
            return render(request, 'calculator/error.html', {'error': 'Invalid operation.'})

        return render(request, 'calculator/result.html', {
            'operation': operation,
            'num1': num1,
            'num2': num2,
            'result': result,
        })

    except ValueError:
        return render(request, 'calculator/error.html', {'error': 'Invalid numbers.'})
    except Exception as e:
        return render(request, 'calculator/error.html', {'error': str(e)})
```

7. Créer les templates HTML

fichier calculator/templates/calculator/result.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Calculation Result</title>
</head>
<body>
  <h1>Calculation Result</h1>
  <p>Operation: {{ operation }}</p>
  <p>Number 1: {{ num1 }}</p>
  <p>Number 2: {{ num2 }}</p>
  <p>Result: {{ result }}</p>
</body>
</html>
```

fichier calculator/templates/calculator/error.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Error</title>
</head>
<body>
  <h1>An Error Occurred</h1>
  <p>{{ error }}</p>
</body>
</html>
```

8. Lancer le serveur local pour tester l'application

(.venv) PS D:\calculator\py.exe manage runserver

Test JSON :

http://127.0.0.1:8000/json/add/5/3/

Test HTML :

http://127.0.0.1:8000/html/add/5/3/