

Johan Sebastian Roa Rodriguez

The image shows a development environment with a code editor on the left and a browser on the right. The code editor displays the following Python code:

```
15 # 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16
17 from django.urls import path
18 from django.http import JsonResponse
19 from django.views.decorators.csrf import csrf_exempt
20 from django.utils.decorators import method_decorator
21 from django.views.decorators.http import require_POST
22 import json
23 import math
24
25 def count_characters(request, word):
26     return JsonResponse({'word': word, 'length': len(word), 'nombre_estu
27
28 urlpatterns = [
29     path('count/<str:word/>', count_characters, name='count_characters')
30 ]
31
32 @csrf_exempt
33 @require_POST
34 def calculate_factorial(request):
35     try:
36         data = json.loads(request.body)
37         number = data.get('number')
38         if number is None or not isinstance(number, int) or number < 0:
39             return JsonResponse({'error': 'Invalid input. Please provide
40         result = math.factorial(number)
41         return JsonResponse({'number': number, 'factorial': result})
42     except json.JSONDecodeError:
43         return JsonResponse({'error': 'Invalid JSON.'}, status=400)
44
45 urlpatterns += [
46     path('factorial/', calculate_factorial, name='calculate_factorial'),
47 ]
```

The browser shows a REST client interface with a POST request to `http://localhost:8000/factorial/`. The response is a JSON object:

```
{
  "number": 7,
  "factorial": 5040
}
```