

TaskApi

آشنایی با API:

Task1:

برای نمایش جوابی request بر می گردانه ابتدا جواب را به فرم json در آورده و فیلد مورد نظر را بر می گردانیم.
از آنجا که جواب ما یک دیکشنری پایتون است در تمبلیت با For مقدار های بدگردانده شده را نمایش می دهیم.

1) view.py

```
class QueryView(generic.ListView):
```

```
    template_name = 'query.html'
```

```
    context_object_name = 'query'
```

```
    def get_queryset(self):
```

```
        response = requests.get('https://rickandmortyapi.com/api/character/?page=20')
```

```
        json_var = response.json()
```

```
        return json_var['results']
```

2) query.html

```
% extends 'base.html' %}
```

```
{#{% load static %}#}
```

```
{% block body %}
```

```
    {% for j in query %}
```

```
        <br>
```

```
        <ul><li >
```

```
            <div class="myinfo">
```

```
                {% for key, value in j.items %}
```

```

{% if 'image' == key %}
    <img src={{ value}}>
{% else %}
    <p> &nbsp; {{ key|capfirst}}:{{value|capfirst }}<br></p>
{% endif %}
{% endfor %}
</div>
</li></ul>

{% endfor %}
{

{% endblock %}

```

Task2:

در ابتدا باید رست فریم ورک را به اب های اینستال شده خودمان اضافه کنیم سپس از آنجا که ما نمی خواهیم اطلاعاتی رو ذخیره کنیم برای همین از serializers.Serializer استفاده می کنیم . پس از آن از ListCreateAPIView استفاده کردیم و فانکشن (post(self, request, *args, **kwargs) و (get(self, request, *args, **kwargs) را باز نویسی می کنیم رای شناسایی تمپلیت هامون از (TemplateHTMLRenderer,) استفاده می کنیم همچنین serializer_class را هم مشخص می کنیم . از آنجا که api_rest از های جانگو پشتیبانی نمی کند در داخل template form ها را نمایش می دهیم.

1) first we add 'rest_framework' in are INSTALLED_APPS in settings.py

2) serializer.py

```

class MySerilizer1(serializers.Serializer):

    input = serializers.CharField(max_length=1000)

    delimiter = serializers.CharField(max_length=5)

```

3) views.py

```

class MyView1(ListCreateAPIView):

```

```
serializer_class = MySerilizer1
context_object_name = 'me'
renderer_classes = (TemplateHTMLRenderer,)

def post(self, request, *args, **kwargs):
    db_start = time.time()
    request_time = datetime.datetime.now()
    serializer = MySerilizer1(data=request.data)
    if serializer.is_valid():
        result = serializer["input"].value.split(serializer["delimiter"].value)
        my_response = {
            "id": 1,
            "output": result,
            "WordCount": len(result),
            "InputString": serializer["input"].value,
            "ProcessingDelay": time.time()-db_start,
            "RequestTimeDate": request_time,
            "ResponseTmeDate": datetime.datetime.now()
        }
        # request.data = my_response

    return Response(data=my_response, template_name='query1.html')
else:
    return Response(status.HTTP_203_NON_AUTHORITATIVE_INFORMATION,
template_name='query.html')
```

```
def get(self, request, *args, **kwargs):  
    return render(request, template_name='form.html')  
    pass
```

4) my templates

```
{% extends 'base.html' %}  
{% block body %}  
  
<div class="container-fluid">  
    <div class="row">  
        <div class="col-sm-12 col-md-7">  
            <div class="panel panel-default">  
                <div class="panel-body">  
                    <form class="form-horizontal" action="{% url 'task2:MyQuery' %}"  
method="post" enctype="multipart/form-data" novalidate>  
                        {% csrf_token %}  
                        {% include 'form-template.html' %}  
                    </form>  
  
                </div>  
            </div>  
        </div>  
    </div>  
    </div>  
    </div>  
    </div>  
{% endblock %}  
  
<div>
```

```
<fieldset>

  <input type="hidden" name="csrfmiddlewaretoken" value="2IttJtM6SibIPrL7Y-
iN1EUktfaU2MC7U1wJFGpKMsZCHF68mEb8aWXv30BAW8Bn1">

  <div class="form-group ">
    <label class="col-sm-2 control-label ">
      Input
    </label>

    <div class="col-sm-10">
      <input name="input" class="form-control" type="text" value>
    </div>
  </div>

  <div class="form-group">
    <label class="col-sm-2 control-label ">
      Delimiter
    </label>

    <div class="col-sm-10">
      <input name="delimiter" class="form-control" type="text" value>
    </div>
  </div>

  <div class="form-actions">
    <div class="col-sm-offset-2 col-sm-10">
      <button type="submit" class="btn btn-primary"> submit</button>
    </div>
  </div>

</fieldset>
```

Tuesday, August 7, 2018

</div>

Git Repository :