

Django Rest

Práctica

Crear entorno virtual

- django
- djangorestframework

Crear el proyecto djangoRest

Crear app projects

Agregar la app al settings.py

```
33 INSTALLED_APPS = [  
34     'django.contrib.admin',  
35     'django.contrib.auth',  
36     'django.contrib.contenttypes',  
37     'django.contrib.sessions',  
38     'django.contrib.messages',  
39     'django.contrib.staticfiles',  
40     'projects',  
41     'rest_framework',  
42 ]  
43
```

Crear el modelo

```
models.py U X  
DiplomadoDAW > Módulo7 > source > solutions > djangoRest > projects > m  
1 from django.db import models  
2  
3 class Project(models.Model):  
4     title = models.CharField(max_length=100)  
5     description = models.TextField()  
6     created = models.DateTimeField(auto_now_add=True)  
7
```

Realizar las migraciones a la base de datos

Crear el super usuario

Crear el serializador

```
serializers.py U X
DiplomadoDAW > Módulo7 > source > solutions > djangoRest > projects > serializers.py
1  from rest_framework import serializers
2  from .models import Project
3
4  class ProjectSerializer(serializers.ModelSerializer):
5      class Meta:
6          model = Project
7          fields = ('id', 'title', 'description', 'created')
8          read_only_fields = ('created',)
```

Crear el archivo api.py

```
api.py U X
DiplomadoDAW > Módulo7 > source > solutions > djangoRest > projects > api.py
1  from .models import Project
2  from rest_framework import viewsets, permissions
3  from .serializers import ProjectSerializer
4
5  class ProjectViewSet(viewsets.ModelViewSet):
6      queryset = Project.objects.all()
7      #Cualquier cliente puede hacer solicitudes.
8      permission_classes = [permissions.AllowAny]
9      serializer_class = ProjectSerializer
```

No es necesario crear las rutas en el archivo urls.py uno a uno.. para ello utilizamos “routers”

```

urls.py  U X
DiplomadoDAW > Módulo7 > source > solutions > djangoRest > projects > urls.py >
1  from rest_framework import routers
2  from .api import ProjectViewSet
3
4  router = routers.DefaultRouter()
5
6  #Se generan un conjunto de rutas para todo el CRUD
7  router.register('api/projects', ProjectViewSet, 'projects')
8
9  urlpatterns = router.urls

```

Modificar el urls.py principal

```

urls.py  U X
DiplomadoDAW > Módulo7 > source > solutions > djangoRest
1  from django.contrib import admin
2  from django.urls import path, include
3
4  urlpatterns = [
5      path('admin/', admin.site.urls),
6      path('', include('projects.urls'))
7  ]
8

```

Verifique su funcionamiento

Django REST framework

Api Root

Api Root

The default basic root view for DefaultRouter

GET /

HTTP 200 OK
 Allow: GET, HEAD, OPTIONS
 Content-Type: application/json
 Vary: Accept

```
{
  "api/projects": "http://127.0.0.1:8000/api/projects/"
}
```

Title

Description

POST

```
Content-Type: application/json
Vary: Accept

{
  "id": 1,
  "title": "Proyecto 1",
  "description": "Descripción del proyecto 1",
  "created": "2022-11-24T16:25:33.739152Z"
}
```

Cliente AJAX

Crear un cliente mediante AJAX, desde otra aplicación en otro puerto.

Configuración

Instalar la librería: “django-cors-headers”

En el archivo settings.py

Agregar la aplicación

```
39     'django.contrib.staticfiles',
40     'projects',
41     'corsheaders',
42     'rest_framework',
43 ]
```

Agregar el middleware

```
45 MIDDLEWARE = [
46     'django.middleware.security.SecurityMiddleware',
47     'django.contrib.sessions.middleware.SessionMiddleware',
48     'corsheaders.middleware.CorsMiddleware',
49     'django.middleware.common.CommonMiddleware',
50     'django.middleware.csrf.CsrfViewMiddleware',
```

Agregar los orígenes permitidos

```
55
56 CORS_ALLOWED_ORIGINS = ['http://localhost:8001', 'http://127.0.0.1:8001']
57
```

Crear un archivo "index.html"

```
22 <body>
23   <div class="container grid-lg">
24     <h2>Test</h2>
25     <hr />
26     <div class="columns" id="test">
27       <div class="column col-3">
28         <button class="btn btn-primary" id='testGetButton'>Projects (Get)</button>
29       </div>
30       <div class="column col-9">
31         <div id='testGetResponse'></div>
32       </div>
33     </div>
34   </div>
35
```

Recuerde incluir los scripts para incluir JQuery y Bootstrap

```
<script>
  var g_urls = {
    'test_get': 'http://127.0.0.1:8000/api/projects/',
  };

  $(function () {
    $('#testGetButton').click(function() {
      $.ajax({
        url: g_urls.test_get,
        method: "GET",
        beforeSend: function(request) {
        }
      }).done(function(data) {
        $('#testGetResponse').html("<span class='text-primary'>Ok! Response: </span>" + data);
      }).fail(function(data) {
        $('#testGetResponse').html("<span class='text-danger'>Fail! Response: " +
          data.responseText + " (status: " + data.status+">");
      });
    });
  });
</script>
```

Verifique su funcionamiento

Arranque un pequeño servidor haciendo uso de Python

```
python -m http.server 8001
```

Práctica

Haciendo uso de JQuery y un ciclo, muestre una tabla con cada uno de los proyectos recibidos.

```
var row = <tr><td>id</td><td>name</td></tr>
$.each(data, function(i) {
  // ...
})
```

Por ejemplo:

Projects (Get)

Ok! Response:

id	title	description	fecha
1	Proyecto 1	Descripción del proyecto 1	2022-11-24T16:25:33.739152Z
2	Proyecto 1	Descripción del proyecto 1	2022-11-24T16:45:10.783189Z
3	Proyecto 1	Descripción del proyecto 1	2022-11-24T16:50:51.080423Z

Práctica POST

De la misma manera, como se realizó el GET, realizar una ejecución para el POST, enviando los datos para un registro nuevo.

```
$('#testPostButton').click(function() {  
    var str_json = '{ "title": "dddd", "description": "cccccccccc" }';  
    var obj_json = jQuery.parseJSON(str_json);  
    $.ajax({  
        url: g_urls.test_post,  
        method: "POST",  
        dataType: "json",  
        data: obj_json,  
        beforeSend: function(request) {  
        }  
    }).done(function(data) {  
        $('#testPostResponse').html("<span class='text-primary'>Ok! Response: </span>" + data.title);  
    }).fail(function(data) {  
        $('#testGetResponse').html("<span class='text-danger'>Fail! Response: " + data.responseText  
        + " (status: " + data.status+ "</span>");  
    });  
});
```

Modifique el HTML para crear una interfaz y que sea amigable al usuario.