

Django workshop16

DRF(Django REST Framework) 에 대한 이해

1. Model & Admin

Article : title, content, created_at , updated_at 컬럼이 존재한다

관리자 페이지 혹은 djang seed 라이브러리를 활용하여 5 개의 게시글을 생성한다

```
from django.db import models

# Create your models here.

class Article(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
```

```
python manage.py seed articles --number=5
```

2. serializers

ArticleListSerializer : 모든 게시글 정보를 반환하기 위한 Serializer / id, title 필드 정의

ArticleSerializer : 상세 게시글 정보를 반환 및 생성하기 위한 Serializer / id, title, content, created_at , updated_at 필드 정의

```
from rest_framework import serializers
from .models import Article

# 모든 게시글 정보를 반환하기 위한 serializer
class ArticleListSerializer(serializers.ModelSerializer):

    class Meta:
        model = Article
        fields = ('id', 'title', )

# 상세 게시글 정보를 반환 및 생성하기 위한 serializer

class ArticleSerialilizer(serializers.ModelSerializer):

    class Meta:
        model = Article
        fields = '__all__'
```

3. url & views

- **GET** api/v1/articles/
 - 모든 게시글의 id와 title 컬럼을 JSON으로 응답한다.
- **POST** api/v1/articles/
 - 검증에 성공하는 경우 새로운 게시글의 정보를 DB에 저장하고 저장된 게시글의 정보를 응답한다.
 - 검증에 실패하는 경우 400 Bad Request 오류를 발생시킨다.
- **GET** api/v1/articles/<article_pk>/
 - 특정 게시글의 모든 컬럼을 JSON으로 응답한다.
- **PUT & DELETE** api/v1/articles/<article_pk>/
 - PUT 요청인 경우 특정 게시글의 정보를 수정한다.
 - 검증에 성공하는 경우 수정된 게시글의 정보를 DB에 저장한다.
 - 검증에 실패할 경우 400 Bad Request 오류를 발생시킨다.
 - 수정이 완료되면 수정한 게시글의 정보를 응답한다.
 - DELETE 요청인 경우 특정 게시글의 정보를 삭제한다.
 - 삭제가 완료되면 삭제한 게시글의 id를 응답한다.

❖ 각 요청에 해당하는 JSON 응답 결과와

views.py, serializers.py 코드를 마크다운에 작성하여 제출하시오.

3-1 & 3-2

```
### api/urls.py -- urlpatterns
path('api/v1/', include('articles.urls')),

### articles/urls.py -- urlpatterns
path('articles/', views.article_list_create),

### articles/views.py
from django.shortcuts import get_list_or_404, get_object_or_404

from rest_framework.decorators import api_view
from rest_framework.response import Response
from rest_framework import status

from .models import Article
from .serializers import ArticleListSerializer, ArticleSerializer

@api_view(['GET', 'POST'])
def article_list_create(request):
    if request.method == 'GET':
        # 인스턴스 정보
        articles = get_list_or_404(Article)
        # 시리얼라이저 인스턴스 생성
        serializer = ArticleListSerializer(articles, many=True)
        # JSON으로 응답
        return Response(serializer.data)
```

```

elif request.method == 'POST':

    serializer = ArticleSerializer(data=request.data)
    # 검증실패 -> 400
    if serializer.is_valid(raise_exception=True):
        serializer.save()
        return Response(serializer.data, status=status.HTTP_201_CREATED)

```

127.0.0.1:8000/api/v1/articles/

Django REST framework

Article List Create

Article List Create

GET /api/v1/articles/

```

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

[
  {
    "id": 1,
    "title": "Free seat money together interview process single base."
  },
  {
    "id": 2,
    "title": "History a pick surface house note."
  },
  {
    "id": 3,
    "title": "Personal think especially speak."
  },
  {
    "id": 4,
    "title": "Professional situation front administration war."
  },
  {
    "id": 5,
    "title": "Cup partner woman break maybe change laugh."
  }
]

```

3-3 & 3-4

```

### articles/urls.py -- urlpatterns

# GET ~/articles/<int:article_pk>/ 특정 게시물 모든 컬럼 JSON
# PUT ~/articles/<int:article_pk>/ : 특정 게시물 정보 수정
# DELETE~/articles/<int:article_pk>/ : 특정 게시물 삭제
path('articles/<int:article_pk>/', views.article_detail_delete_update),

### articles/views.py
@api_view(['GET', 'DELETE', 'PUT'])
def article_detail_delete_update(request, article_pk):

```

```

# 특정 게시물
article = get_object_or_404(Article, pk=article_pk)

# JSON 보여줘야됨.
if request.method == 'GET':
    serializer = ArticleSerializer(article)
    return Response(serializer.data)

# 삭제 해야됨.
elif request.method == 'DELETE':
    article.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)

# 수정 해야됨
elif request.method == 'PUT':
    serializer = ArticleSerializer(article, data=request.data)

    # 디폴트로 통과못하면 200 띄움
    if serializer.is_valid(raise_exception=True):
        serializer.save()
    # default 200
    return Response(serializer.data)

```

127.0.0.1:8000/api/v1/articles/1/ Django REST framework

Article List Create / Article Detail Delete Update

Article Detail Delete Update

DELETE OPTIONS GET -

GET /api/v1/articles/1/

```

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

{
  "id": 1,
  "title": "Free seat money together interview process single base.",
  "content": "Rock record sea occur.\nForeign edge none. Always no during writer themselves. Life girl wide election. Art play land herself house quality method.",
  "created_at": "2009-10-28T22:11:28Z",
  "updated_at": "1986-06-04T04:49:41Z"
}

```