



College of Engineering, Construction and Living Sciences
Bachelor of Information Technology
IN608: Intermediate Application Development Concepts
Level 6, Credits 15
Practical 14 Django 8: Django REST Framework

Due Date: 07-05-2021 at 5pm

In this practical, you will complete a series of tasks covering today's lecture. This practical is worth 2% of the final mark for the IN608: Intermediate Application Development Concepts course.

Before you start, in your practicals repository, create a new branch called **14-practical**.

Task

Create a Django project called `quizcreator`. `cd` to `quizcreator`, create a virtual environment & install Django. Create an app called `practical14quizcreator`. Please ensure you configure your app in `quizcreator/settings.py` & `quizcreator/urls.py`.

In order to create a Django REST API, the `django-rest-framework` Python module must be installed. Run the command `pipenv install django-rest-framework` to install the `django-rest-framework` Python module. In `settings.py`, declare `rest_framework` in `INSTALLED_APPS`.

In the `models.py`, copy & paste the `User` & `Quiz` model from **Practical 12 Django 6: Authentication**.

Create two files called `serializer.py` & `urls.py`. In `serializer.py`, create two classes called `UserSerializer` & `QuizSerializer` which implement `serializers.ModelSerializer`. For each class, create a nested class called `Meta`. In the `Meta` class, declare the appropriate model & fields.

In `view.py`, create two classes called `UserViewSet` which implements `viewsets.ReadOnlyModelViewSet` & `QuizViewSet` which implements `viewsets.ModelViewSet`. For each class, declare the appropriate `queryset` & `serializer_class`, i.e., `User.objects.all()` & `UserSerializer`.

In `urls.py`, create two URLs which map to `UserViewSet` & `QuizViewSet`. The URL patterns should be as follows:

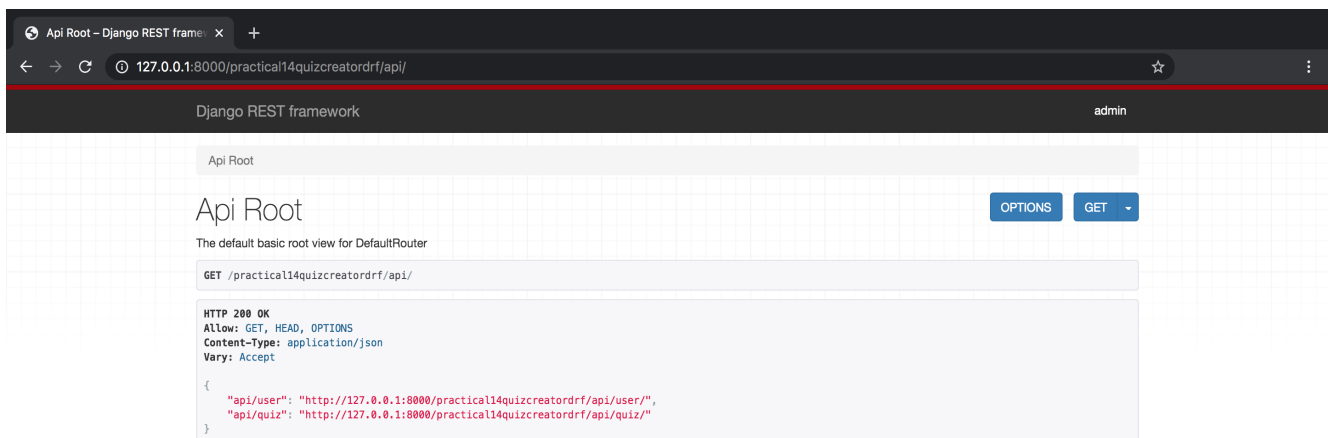
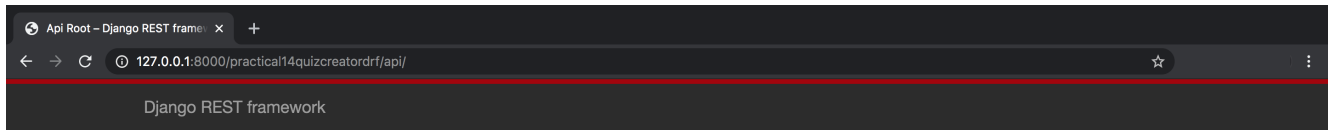
- `api/user`

- `api/quiz`

In `settings.py`, set permission policy so that the API is restricted to only admin users.

Expected Output

Run the command `python manage.py runserver` then navigate to <http://127.0.0.1:8000/practical14quizcreatordrf/api>



User List - Django REST frame x +

127.0.0.1:8000/practical14quizcreatordrf/api/user/

Django REST framework admin

Api Root / User List

User List

OPTIONS GET

GET /practical14quizcreatordrf/api/user/

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

[
  {
    "id": 1,
    "username": "admin",
    "email": "admin@admin.com",
    "first_name": "Admin",
    "last_name": "User"
  }
]
```

Quiz List - Django REST frame x +

127.0.0.1:8000/practical14quizcreatordrf/api/quiz/

Django REST framework admin

Api Root / Quiz List

Quiz List

OPTIONS GET

GET /practical14quizcreatordrf/api/quiz/

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

[
  {
    "id": 1,
    "creator": 1,
    "name": "Quiz 1: General Knowledge - Easy",
    "category": "gk",
    "difficulty": "easy"
  },
  {
    "id": 2,
    "creator": 1,
    "name": "Quiz 2: General Knowledge - Medium",
    "category": "gk",
    "difficulty": "medium"
  },
  {
    "id": 3,
    "creator": 1,
    "name": "Quiz 3: General Knowledge - Hard",
    "category": "gk",
    "difficulty": "hard"
  }
]
```

Raw data HTML form

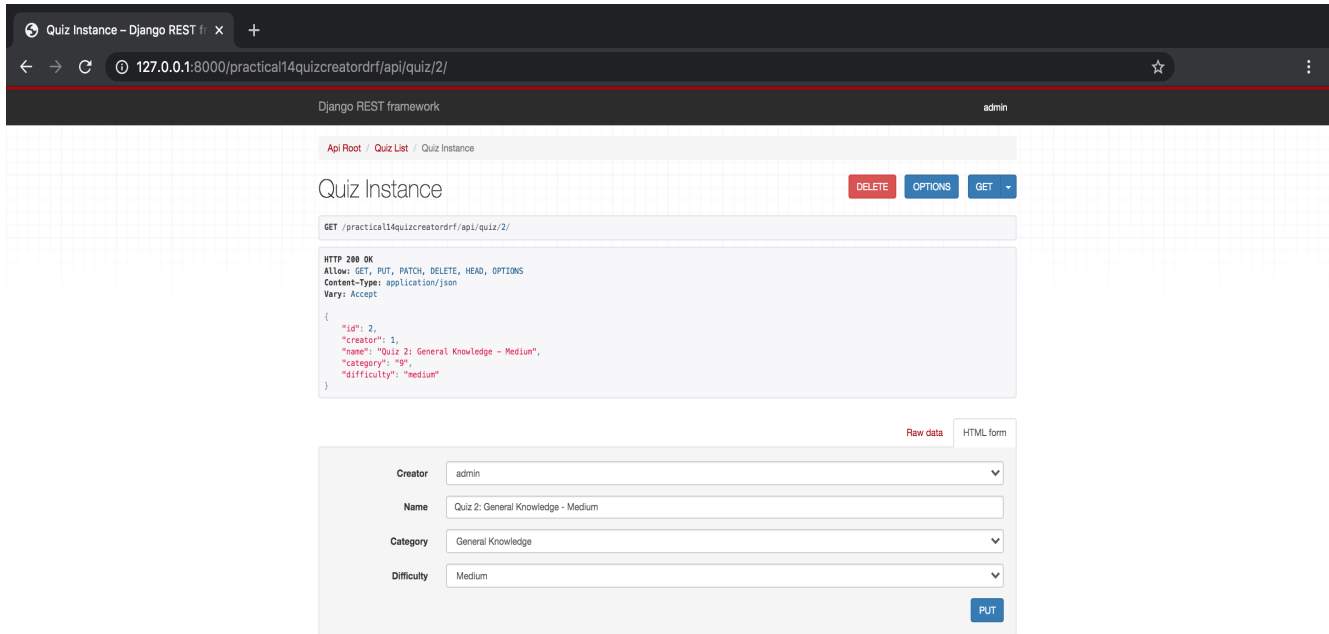
Creator admin

Name

Category Any Category

Difficulty Any Difficulty

POST



Deployment link: <https://int-app-dev-practical-14.herokuapp.com/practical14quizcreatordrf/api>

Resources

- [Django REST Framework](#)