

Jalolov Tursunbek Sadriddinovich
Osiyo xalqaro universiteti assistenti

ts_jalolov@oxu.uz

DJANGODA API LAR BILAN ISHLASH

Annotatsiya: Mazkur maqolada **Django** veb-ramkasi orqali API'lar bilan ishlash masalasi yoritiladi. Django Rest Framework (DRF) API'lar yaratishda asosiy vosita hisoblanib, API interfeyslari orqali ilovalar va tizimlar o'rtasida qanday qilib samarali va xavfsiz aloqani ta'minlash mumkinligi ko'rib chiqiladi. Ushbu maqola API'larning umumiy tushunchasi, Django Rest Framework orqali API yaratish jarayoni, autentifikatsiya va avtorizatsiya kabi masalalarga qaratiladi.

Kirish

API (Application Programming Interface) dasturiy interfeyslar turli tizimlar, ilovalar va xizmatlar o'rtasida o'zaro aloqa o'rnatish uchun ishlatiladi. API'lar orqali ma'lumot almashish zamonaviy dasturlashning ajralmas qismi hisoblanadi. Django ramkasi – veb-ilovalarni tez va samarali rivojlantirish uchun keng foydalaniladigan vosita bo'lib, uning asosida RESTful API'larni yaratish juda qulaydir. Django'da API'larni yaratishda **Django Rest Framework (DRF)** kutubxonasidan foydalaniladi. DRF orqali qulay va xavfsiz API'larni yaratish, ularni autentifikatsiya qilish va foydalanuvchi huquqlarini boshqarish imkoniyati mavjud.

API'larning asosiy tushunchasi

API (Dasturiy interfeys) dasturlar, serverlar yoki xizmatlar o'rtasida ma'lumotlar almashish va o'zaro aloqani ta'minlovchi ko'prikdir. API orqali tashqi ilovalar yoki tizimlar ma'lumotlarni olish yoki o'zgarish kiritish uchun foydalanadi. API'lar RESTful (Representational State Transfer) tamoyillari asosida quriladi, bu esa oddiy HTTP protokollari orqali amalga oshiriladi.

API'ning asosiy afzalliklari:

- **Mustaqillik:** API orqali ma'lumot almashish bir-biriga bog'liq bo'lmagan tizimlar o'rtasida ham amalga oshiriladi.
- **Qayta foydalanish:** Bir marta yaratilgan API boshqa tizimlar va dasturlar tomonidan foydalanilishi mumkin.
- **Sodda o'zaro aloqa:** API'lar orqali HTTP metodlar (GET, POST, PUT, DELETE) yordamida ma'lumot olish va yuborish mumkin.

Django Rest Framework (DRF) haqida umumiy tushuncha

Django Rest Framework (DRF) Django dasturiy ramkasiga asoslangan va API yaratishda keng qo'llaniladigan kuchli kutubxonadir. DRF RESTful API'larni yaratish, boshqarish va rivojlantirish uchun yuqori darajada moslashuvchan vositalarni taqdim etadi. DRF'dan foydalanishning asosiy afzalliklari:

- Django ORM bilan chuqur integratsiya.
- Foydalanuvchi autentifikatsiyasi va avtorizatsiyasi uchun moslashuvchan tizim.
- JSON formatidagi ma'lumotlar bilan oson ishlash.
- Modellar va serializerlar yordamida ma'lumotlarni qayta ishlash.

Django Rest Framework orqali API yaratish bosqichlari:

1. Loyiha yaratish va o'rnatish

Django va DRF kutubxonasini o'rnatish va asosiy loyihani yaratish uchun quyidagi buyruqlar ishlatiladi:

```
bash
Копировать код
pip install django
pip install djangorestframework
django-admin startproject api_project
```

2. App yaratish

Django ichida API bilan ishlovchi dastur yaratish uchun quyidagi buyruqdan foydalaniladi:

```
bash
Копировать код
python manage.py startapp api_app
```

3. Model yaratish

Django'da API uchun avval ma'lumotlar modeli (Model) yaratiladi. Masalan, quyidagi modelda "Post" ma'lumot turi ko'rsatilgan:

```
python
Копировать код
from django.db import models

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

4. Serializer yaratish

Modelni JSON formatiga o'g'irish uchun serializerdan foydalaniladi:

```
python
Копировать код
from rest_framework import serializers
from .models import Post

class PostSerializer(serializers.ModelSerializer):
    class Meta:
        model = Post
        fields = '__all__'
```

5. View'lar va URL'lar

API'ga HTTP so'rovlari (GET, POST, PUT, DELETE) yuborish uchun view va URL'lar yaratiladi:

```
python
Копировать код
```

```
from rest_framework.decorators import api_view
from rest_framework.response import Response
from .models import Post
from .serializers import PostSerializer
```

```
@api_view(['GET'])
def get_posts(request):
    posts = Post.objects.all()
    serializer = PostSerializer(posts, many=True)
    return Response(serializer.data)
```

URL:

```
python
Копировать код
from django.urls import path
from . import views
```

```
urlpatterns = [
    path('posts/', views.get_posts),
]
```

Autentifikatsiya va Avtorizatsiya

API'larni xavfsiz qilish muhim masalalardan biridir. Django Rest Framework bir nechta autentifikatsiya usullarini taqdim etadi:

- **Token autentifikatsiyasi:** Har bir foydalanuvchiga noyob token ajratiladi, bu token API'lar orqali so'rov yuborishda ishlatiladi.
- **Session autentifikatsiyasi:** Django sesiyasi orqali autentifikatsiya amalga oshiriladi.
- **OAuth 2.0:** Bu standart autentifikatsiya mexanizmi bo'lib, DRF'da ham qo'llanilishi mumkin.

Token autentifikatsiyasini qo'llash uchun quyidagi amallar bajariladi:

1. Token o'rnatish:

```
bash
Копировать код
pip install djangorestframework-simplejwt
```

2. DRF sozlamalariga quyidagilar qo'shiladi:

```
python
Копировать код
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': (
        'rest_framework_simplejwt.authentication.JWTAuthentication',
    ),
}
```

3. Token olish uchun yangi view yaratish:

```
python
Копировать код
from rest_framework_simplejwt.views import TokenObtainPairView

urlpatterns = [
    path('api/token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
]
```

Token bilan autentifikatsiya qilinadigan so'rovlar yuborish jarayonida har bir so'rovga tokenni qo'shish kerak bo'ladi:

```
bash
Копировать код
Authorization: Bearer <token>
```

API'lar uchun test qilish

Django Rest Framework yaratgan API'larni test qilish uchun **Postman** yoki **curl** kabi vositalardan foydalanish mumkin. Bu vositalar yordamida HTTP so'rovlari yuborish, javoblarni tekshirish va API'larni sinovdan o'tkazish qulaydir.

Misol uchun, curl yordamida GET so'rov yuborish:

```
bash
Копировать код
curl -X GET http://127.0.0.1:8000/posts/
```

Xulosa

Django'da API'lar bilan ishlash zamonaviy veb-illovalar va mikroxizmatlar arxitekturasida muhim o'rin tutadi. Django Rest Framework bu jarayonni soddalashtirib, mustahkam va xavfsiz API'lar yaratishga imkon beradi. Mazkur maqolada DRF yordamida API yaratish, autentifikatsiya va avtorizatsiya kabi muhim jarayonlar tushuntirildi. Django'da API'lar orqali o'zaro aloqa qilish imkoniyatlari foydalanuvchilarga keng imkoniyatlar yaratadi va dasturlarning ko'p funktsional bo'lishini ta'minlaydi.

Foydalanilgan adabiyotlar.

1. Jalolov, T. S. (2024). ПОРЯДОК СОЗДАНИЯ ПСИХОЛОГИЧЕСКИХ ТЕСТОВЫХ ПРОГРАММ. PEDAGOG, 7(6), 145-152.
2. Jalolov, T. S. (2024). BOSHLANG'ICH SINIF O'QUVCHILARIDA MULTIMEDIA TEXNOLOGIYALARI ORQALI IJODIY FIKRLASHNI KUCHAYTIRISH. BIOLOGIYA VA KIMYO FANLARI ILMIY JURNALI, 2(5), 64-70.
3. Jalolov, T. S. (2024). EXPLORING THE MATHEMATICAL LIBRARIES OF PYTHON: A COMPREHENSIVE GUIDE. WORLD OF SCIENCE, 7(5), 121-127.
4. Jalolov, T. S. (2024). THE IMPORTANCE OF ENGLISH IN PROGRAMMING. WORLD OF SCIENCE, 7(5), 128-134.
5. Jalolov, T. S. (2024). ИЗУЧЕНИЕ МАТЕМАТИЧЕСКИХ БИБЛИОТЕК PYTHON: ПОДРОБНОЕ РУКОВОДСТВО. MASTERS, 2(5), 48-54.



6. Jalolov, T. S. (2023). MATH MODULES IN C++ PROGRAMMING LANGUAGE. Journal of Universal Science Research, 1(12), 834-838.
7. Jalolov, T. S. (2024). DASTURLASHDA INGLIZ TILINING AHAMIYATI. BIOLOGIYA VA KIMYO FANLARI ILMIY JURNALI, 2(5), 78-84.
8. Jalolov, T. S. (2024). ENHANCING CREATIVE THINKING IN ELEMENTARY SCHOOL STUDENTS THROUGH MULTIMEDIA TECHNOLOGIES. WORLD OF SCIENCE, 7(5), 114-120.
9. Akbarovna, I. S. (2024). THE IMPORTANCE OF PEDAGOGICAL MANAGEMENT IN EDUCATION. PSIXOLOGIYA VA SOTSIOLOGIYA ILMIY JURNALI, 2(4), 40-46.
10. Akbarovna, I. S. (2024). PSIXOLOGIK MASLAHAT BOSQICHLARI. Multidisciplinary Journal of Science and Technology, 4(3), 860-866.
11. Икромова, С. А. (2024). ЭТАПЫ ПСИХОЛОГИЧЕСКОЙ КОНСУЛЬТАЦИИ. Multidisciplinary Journal of Science and Technology, 4(3), 875-881.
12. Akbarovna, I. S. (2024). MATEMATIK TUSHUNCHALARNI SHAKLLANISHINING AHAMIYATI. Multidisciplinary Journal of Science and Technology, 4(3), 900-905.
13. Akbarovna, I. S. (2024). THE IMPORTANCE OF FORMING MATHEMATICAL CONCEPTS. Multidisciplinary Journal of Science and Technology, 4(3), 912-917.
14. Akbarovna, I. S. (2024). PEDAGOGIK MAHORAT ASOSLARINI O'ZLASHTIRISH. Multidisciplinary Journal of Science and Technology, 4(3), 888-893.
15. Akbarovna, I. S. (2024). MASTERING THE BASICS OF PEDAGOGICAL SKILLS. Multidisciplinary Journal of Science and Technology, 4(3), 882-887.
16. Ikromova, S. (2024). FORMS AND CONTENT OF PEDAGOGICAL TECHNOLOGY. Multidisciplinary Journal of Science and Technology, 4(3), 933-939.
17. Ikromova, S. (2023). CONCEPT OF IDEOLOGY AND FORMATION OF IDEOLOGICAL IMMUNITY IN YOUTH STUDENTS. Modern Science and Research, 2(6), 1223-1226.
18. Ikromova, S. (2024). PEDAGOGIK TEXNOLOGIYA SHAKLLARI VA MAZMUNI. Multidisciplinary Journal of Science and Technology, 4(3), 918-924.
19. Ikromova, S. (2023). Formation of ideological immunity to destructive information in teenagers. Modern science and research, 2(5), 1009-1014.
20. Akbarovna, I. S. (2024, April). BOSHLANG'ICH TA'LIMDA TA'LIM MASALASI. In International conference on multidisciplinary science (Vol. 2, No. 4, pp. 21-26).