Full Stack React & Django

- https://youtu.be/0d7clfiydAc

Django Token Authentication

settings.py

snippet.python

```
INSTALLED_APPS = [
  'know',
  'accounts'
]

REST_FRAMEWORK = {
  'DEFAULT_AUTHENTICATION_CLASSES' : ('knox.auth.TokenAuthentication',
) # Tuple
}
```

snippet.shell

```
python manage.py startapp accounts
```

accounts

serializers.py

snippet.python

```
from rest_framework import serializers
from django.contrib.auth.models import User
from django.contrib.auth import authenticate

# User Serializer
class UserSerializer(serializers.ModelSerializer):
  class Meta:
    model = User
    fields = ('id', 'username', 'email')

# Register Serializer
class RegisterSerializer(serializers.ModelSerializer):
  class Meta:
    model = User
```

```
    fields = ('id', 'username', 'email', 'password')
    extra_kwargs = {'password': {'write_only': True}}

  def create(self, validated_data):
    user = User.objects.crate_user(validated_data['username'],
validated_data['email'], validated_data['password'])
    return user

# Login Serializer
class LoginSerializer(serializer.Serializer):
  username = serializer.CharField()
  password = serilizaer.CharField()

  def validate(self, data):
    user = authenticate(**data)
    if user and suer.is_active:
      return user
    rais serializer.ValidateionError('Incorrect Credentials')
```

api.py

snippet.python

```
from rest_framework import generics, permissions
from rest_framework.response import Response
from knox.models import AuthToken
from .serializers import UserSerializer, registerSerializer

# Register API
class RegisterAPI(generics.GenericAPIVIew):
  serializer_class = RegisterSerializer

  def post(self, request, *args, **kwargs):
    serializer = self.get_serializer(data=request.data)
    serializer.is_valid(raise_exception=True)
    user = serializer.save()
    return Response({
      "user": UserSerizlier(user,
context=self.get_serilizer_context()).data,
      "token": AuthToken.objects.create(user)
    })


# Login API
class LoginAPI(generics.GenericAPIView):
  serializer_class = LoginSerializer

  def post(self, request, *args, **kwargs):
```

```
    serializer = self.get_serializer(data=request.data)
    serializer.is_valid(raise_exception=True)
    user = serializer.validated_data

    return Response({
      "user": UserSerizlier(user,
context=self.get_serilizer_context()).data,
      "token": AuthToken.objects.create(user)
    })

# Get User API
class UserAPI(generics.RetrieveAPIView):
  permission_classes = [
    permissions.IsAuthenticated,
  ]
  serializer_class = UserSerializer

  def get_object(self):
    return self.request.user
```

urls.py

snippet.python

```
from django.urls import path, include
from .api import RegisterAPI
from knox import views as knox_views

urlpatterns = [
  path('api/auth', include('knox.urls))
  path('api/auth/register', RegisterAPI.as_view())
  ...
  path('api/auth/logout', knox_views.LogoutView.as_view(),
name='knox_logout')
]
```

Auth State & Private Routes

- https://youtu.be/EmAc4wQikwY

Frontend Authentication

- https://youtu.be/kfpY5BsIoFg

관련 문서

Plugin Backlinks: 아무 것도 없습니다.