

## Authentication

### Third party packages:-

- Django OAuth Toolkit
- JSON Web Token Authentication
- Hawk HTTP Authentication
- Djoser
- django-rest-auth / dj-rest-auth
- django-rest-framework-social-oauth2
- django-rest-knox
- drfpasswordless

## JSON Web Token (JWT)

JSON Web Token is a fairly new standard which can be used for token-based authentication. Unlike the built-in TokenAuthentication scheme, JWT Authentication doesn't need to use a database to validate a token. <https://jwt.io/>

## Simple JWT

Simple JWT provides a JSON Web Token authentication backend for the django REST Framework. It aims to cover the most common use cases of JWTs by offering a conservative set of default features. It also aims to be easily extensible in case a desired feature is not present. <https://django-rest-framework-simplejwt.readthedocs.io/en/latest/>

## How to Install Simple JWT

```
pip install.djangorestframework-simplejwt
```

## Configure simple JWT

### [settings.py](#)

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

### [urls.py](#)

```
from django.urls import path
from rest_framework_simplejwt.views import (
    TokenObtainPairView,
    TokenRefreshView,
)
urlpatterns = [
    path('gettoken/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
    path('refreshtoken/', TokenRefreshView.as_view(), name='token_refresh'),
]
```

```
]
```

You can also include a route for Simple JWT's TokenVerifyView if you wish to allow API users to verify HMAC-signed tokens without having access to your signing key.

#### [urls.py](#)

```
from django.urls import path
from rest_framework_simplejwt.views import (
    TokenObtainPairView,
    TokenRefreshView,
    TokenVerifyView,
)
urlpatterns = [
    path('gettoken/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
    path('refreshtoken/', TokenRefreshView.as_view(), name='token_refresh'),
    path('verifytoken/', TokenVerifyView.as_view(), name='token_verify'),
]
```

#### JWT Default Settings

```
from datetime import timedelta
from django.conf import settings
SIMPLE_JWT = {
    'ACCESS_TOKEN_LIFETIME': timedelta(minutes=5),
    'REFRESH_TOKEN_LIFETIME': timedelta(days=1),
    'ROTATE_REFRESH_TOKENS': False,
    'BLACKLIST_AFTER_ROTATION': True,

    'ALGORITHM': 'HS256',
    'SIGNING_KEY': settings.SECRET_KEY,
    'VERIFYING_KEY': None,
    'AUDIENCE': None,
    'ISSUER': None,

    'AUTH_HEADER_TYPES': ('Bearer',),
    'USER_ID_FIELD': 'id',
    'USER_ID_CLAIM': 'user_id',

    'AUTH_TOKEN_CLASSES': (
        'rest_framework_simplejwt.tokens.AccessToken',
    )
}
```

```
),
'TOKEN_TYPE_CLAIM': 'token_type',

'JTI_CLAIM': 'jti',

'SLIDING_TOKEN_REFRESH_EXP_CLAIM': 'refresh_exp',
'SLIDING_TOKEN_LIFETIME': timedelta(minutes=5),
'SLIDING_TOKEN_REFRESH_LIFETIME': timedelta(days=1),
}
```

**ACCESS\_TOKEN\_LIFETIME** - A datetime.timedelta object which specifies how long access tokens are valid

**REFRESH\_TOKEN\_LIGETIME** - A datetime.timedelta object which specifies how long refresh tokens are valid.

## User JWT

## GET Token

http POST http://127.0.0.1:8000/token/ username="user1" password="password1"

## Verify Token

http POST http://127.0.0.1:8000/varifytoken/

token="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1bi90eXBlljoicmVmcmVzaCI6ImV4cCI6MTc2NzZmMDI1NCwiaWF0IjoxNzY3MjQzODU0LCJqdGkiOiJiOWJmZTEzNTM1M2M0ODc5YjBiYWU4N2QxYjk5YTU4ZCIsInVzZXJfaWQiOiIyLn0.vnzdORNorEtP9zDcEOVnTMWA5jiCo5S8lYeeAF6Rmzo"

## Refresh Token

http POST http://127.0.0.1:8000/refresh\_token/

refresh="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1bG90eXBIIjoicmVmcmVzaCI6ImV4cCI6MTc2NzZmMDI1NCwiaWF0IjoxNzY3MjQzODU0LCJqdGkiOiJiOWJmZTEzNTM1M2M0ODc5YjBiYWU4N2QxYjk5YTtyZCIsInVzZXJfaWQiOiIyIn0.vnzdORNorEtP9zDcEOVnTMWA5jiCo5S8IYeeAF6Rmzo"

## Get Data

```
http http://127.0.0.1:8000/students/ 'Authorization: Bearer
```

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1bi90eXBlljoiYWVjZXNzIiwiaWF0IjoxNzY3MjQ1OTkwLCJpYXQiOiE3NjcyNDUyOTAsImp0aSI6IjViZTA5NGMyYTAAZjRhMGU4NDg1YjE3MjRiZWZjOGU2liwidXNlcl9pZC16IjlfQ.-VmmPq3LL0u75TpQ8D4h0GE9qRo7QRnqHYWxCf73SEI'

### **Insert Data**

http -f POST http://127.0.0.1:8000/students/ name=Raj roll=105 city=ktm 'authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1bI90eXBlljoiYWVjZXNzliwiZXhwIjoxNzY3MjQ2MzQ4LCJpYXQiOiE3NjcyNDYwNDgsImp0aSI6IjU2YWlyZGlzYThiODQyMTBiMmM4YzY3OTVIMzY4N2RkIiwidXNlcl9pZCI6IjlfQ.8pAsZX7dv96eDYBiYR3yCoHRRzDLKROD0tQlfsDXh1M'

### **Update Data**

http PUT http://127.0.0.1:8000/students/3/ name=Raj roll=103 city=Pokhara 'authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1bI90eXBlljoiYWVjZXNzliwiZXhwIjoxNzY3MjQ2Njg3LCJpYXQiOiE3NjcyNDYzODcsImp0aSI6ImUwNDQ1NmUzZDU3ZTRmN2JhMTQ2NmIxNDBiY2FmZDNmIiwidXNlcl9pZCI6IjlfQ.vDLWbbNc9OLZRxiQqBn\_5gQSlA7HqOnmtvri85CuVeM'

### **Delete Data**

http DELETE http://127.0.0.1:8000/students/3/ 'authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1bI90eXBlljoiYWVjZXNzliwiZXhwIjoxNzY3MjQ2Njg3LCJpYXQiOiE3NjcyNDYzODcsImp0aSI6ImUwNDQ1NmUzZDU3ZTRmN2JhMTQ2NmIxNDBiY2FmZDNmIiwidXNlcl9pZCI6IjlfQ.vDLWbbNc9OLZRxiQqBn\_5gQSlA7HqOnmtvri85CuVeM'

## **Permission Classes**

Permissions in the REST framework are always defined as a list of permission classes.

- AllowAny
- IsAuthenticated
- IsAdminUser
- IsAuthenticatedOrReadOnly
- DjangoModelPermissions
- DjangoModelPermissionsOrAnonReadOnly
- DjangoModelPermissionsOrAnonRedOnly
- DjangoObjectPermissions
- Custom Permissions