Authentication

- BasicAuthentication
- SessionAuthentication
- TokenAuthentication
- RemoteUserAuthentication
- Custom authentication

Authentication

Third party packages:-

- Django OAuth Toolkit
- JSON Web Token Authentication
- Hawk HTTP Authentication
- HTTP Signature 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 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'),
]
```

Configure Simple JWT

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 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
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,
```

JWT Default Settings

```
    '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),
}
```

JWT Default Settings

ACCESS_TOKEN_LIFETIME - A datetime.timedelta object which specifies how long access tokens are valid.

REFRESH_TOKEN_LIFETIME - A datetime.timedelta object which specifies how long refresh tokens are valid.

Use JWT

GET Token

```
http POST http://127.0.0.1:8000/gettoken/ username="user1"
password="geekyshows“
```

Verify Token

```
http POST http://127.0.0.1:8000/verifytoken/
token="eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZ
XNzIiwiZXhwIjoxNjAzNjIxOTAxLCJqdGkiOiI2NmM4ZWJiYjUwMWM0MzA3
YWJjMGNjNTY2ZmNmNTJiMyIsInVzZXJfaWQiOjJ9.cwUSWrkFnFdO8fP45aE
P6GDa3yaybSVYAG6vGUlkFOo"
```

Use JWT

Refresh Token

```
http POST http://127.0.0.1:8000/refreshtoken/ refresh="eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTYwMzcwODAwMSwianRpIjoiYzYzODBmYjVjMDk3NDVhNjkyYzA5YWRmMGI1ZDQ5OWIiLCJ1c2VyX2lkIjoyfQ.Q-E-8N8VvSZof5IjoNIL-2KECRLqlYzBojbTCj_4dBc"
```

Permission Classes

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

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