TokenAuthentication

This authentication scheme uses a simple token-based HTTP Authentication scheme. Token authentication is appropriate for client-server setups, such as native desktop and mobile clients.

To use the TokenAuthentication scheme you'll need to configure the authentication classes to include TokenAuthentication, and additionally include rest_framework.authtoken in your INSTALLED_APPS setting:

```
INSTALL_APPS = [
    'rest_framework.authtoken',
]
```

Note: Make sure to run manage.py migrate after changing your settings. The rest_framework.authtoken app provides Django database migrations.

If successfully authenticated, TokenAuthentication provides the following credentials
request.user will be a Django User instance.
request.auth will be a rest_framework.authtoken.models. Token instance.
Unauthenticated responses that are denied permission will result in an HTTP 401
Unauthorized response with an appropriate WWW- header. For example:

    WWW-Authencate: Token

The http command line tool may be useful for testing token authenticated APIs. For example:

    http http://127.0.0.1:8000/student-api/' Authorization: Token 994b099c6434883834943'

Note: If you use TokenAuthentication in production you must ensure that your API is only available over https.

Generate Token

To generate token:

- Using Admin Application
- Using Django manage.py command
  - python manage.py drf_create_token <username> - This command will return API for the given user or Create a Token if the token doesn't exist for the user.
- By exposing an API endpoint.
- Using signals

How client can Ask/Create Token

When using TokenAuthentication, you may want to provide a mechanism for clients to obtain a token given the username and password.
REST framework provides a built-in view to provide this behavior. To use it, add the obtain_auth_token view to your URLconf:

```
from rest_framework.authtoken.views import obtain_auth_token
```

```
urlpatterns = [
    path('get-token', obtain_auth_token),
]
```

The Obtain_auth_token view will return a JSON response when valid username and password fields are POSTed to the view using form data or JSON:
http POST http://127.0.0.1:8000/get-token/username=”name”password=”pass”
{‘token’: ’9349238ed933484989f948993’}

It also generates a token if the token is not generated for the provided user.

Permission Classes

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

- AllowAny
- IsAuthenticated
- IsAdminUser
- IsAuthentictedOrReadOnly
- DjangoModelPermissions
- DjangoModelPermissionsOrAnonReadOnly
- DjangoModelPermissionsOrAnonRedOnly
- DjangoObjectPermissions

httpie

HTTPie (pronounced aitch-tee-tee-pi) is a command line HTTP client. Its goal is to make CLI interaction with web services as human-friendly as possible. It provides a simple http command that allows for sending arbitrary HTTPie that can be used for testing, debugging, and generally interacting with HTTP servers.
Syntax:- [flags] [METHOD] URL [ITEM [ITEM]]
How to install httpie - pip install httpie

Use httpie

GET REquest
http http://127.0.0.1:8000/studentapi/

GET Request with Auth
http http://127.0.0.1:8000/studentapi/’Authorizatin:Token 347823jsfdj342840’

POST Request/ Submitting Form
http -f POST http://127.0.0.1:8000/studentapi/ name=Jay roll=104 city=Dhanbad ’Authorizatin:Token 347823jsfdj342840’

PUT Request

```
http PUT http://127.0.0.1:8000/studentapi/1/ name=Jay roll=104 city=Kathmandu
’Authorizatin:Token 347823jsfdj342840’
```

DELETE Request

```
http DELETE http://127.0.0.1:8000/studentapi/1/ name=Jay roll=104 city=Kathmandu
’Authorizatin:Token 347823jsfdj342840’
```