CATLAS-Backend

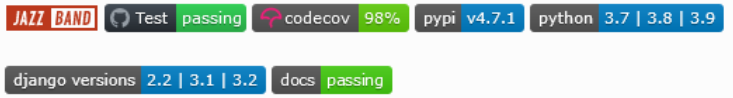
2016010873 박정욱

목차

- 프레임워크
- HTTP 구조
- 사용자 인증
  - SessionAuthentication
  - JWT
  - RSA

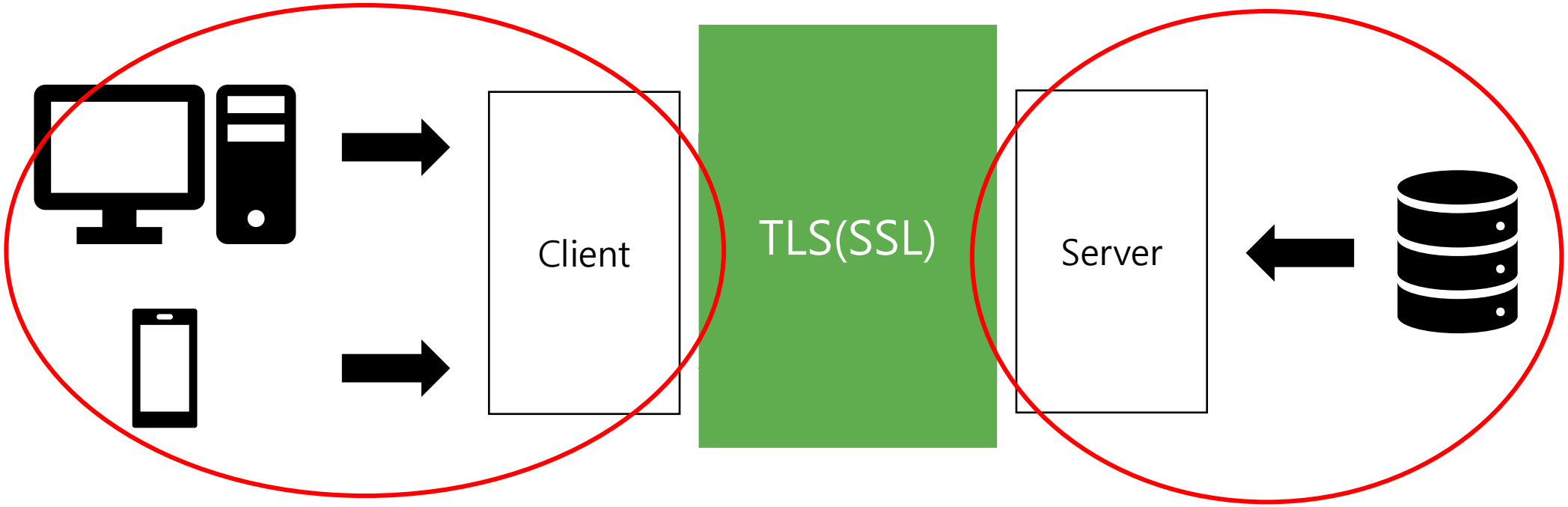
프레임워크

Simple JWT



A JSON Web Token authentication plugin for the Django REST Framework.

HTTP 구조

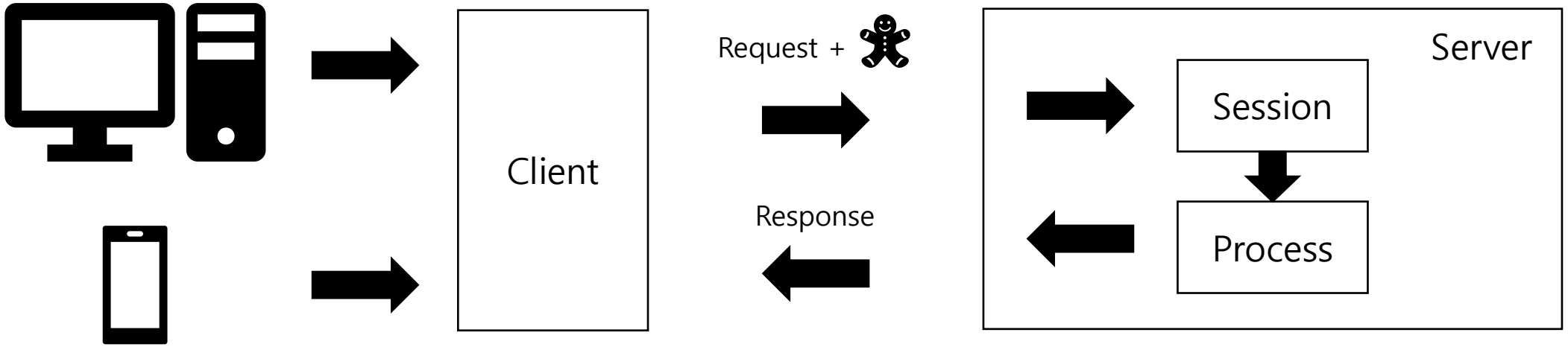


사용자 인증

```
27  DEFAULTS = {
28      # Base API policies
29      'DEFAULT_RENDERER_CLASSES': [
30          'rest_framework.renderers.JSONRenderer',
31          'rest_framework.renderers.BrowsableAPIRenderer',
32      ],
33      'DEFAULT_PARSER_CLASSES': [
34          'rest_framework.parsers.JSONParser',
35          'rest_framework.parsers.FormParser',
36          'rest_framework.parsers.MultiPartParser'
37      ],
38      'DEFAULT_AUTHENTICATION_CLASSES': [
39          'rest_framework.authentication.SessionAuthentication',
40          'rest_framework.authentication.BasicAuthentication'
41      ],
42      'DEFAULT_PERMISSION_CLASSES': [
43          'rest_framework.permissions.AllowAny',
44      ],
```

- BasicAuthentication
  - HTTP의 인증 방법 중 하나
  - RFC 7617
  - 보안 미적용
- SessionAuthentication
  - Session / Cookie
  - RFC 6265
  - 추가적인 저장공간을 필요로 함

SessionAuthentication



JWT

- JSON Web Token

- RFC 7519

aaaaaa.bbbbbb.cccccc

Header Payload Signature

JWT

- Header : 아래 정보의 base64 인코딩 값
  - typ – 토큰의 타입
  - alg – 해당 토큰을 sign하는 알고리즘(HMAC/RSA 등)

- Payload : 아래 정보의 base64 인코딩 값
  - registered/public/private
  - 발급자/대상자 정보, 만료 시간, 식별자, 추가 정보 등

- Signature
  - Header와 Payload를 합친 후 정해진 알고리즘으로 sign
  - 이 값이 base64 인코딩 값

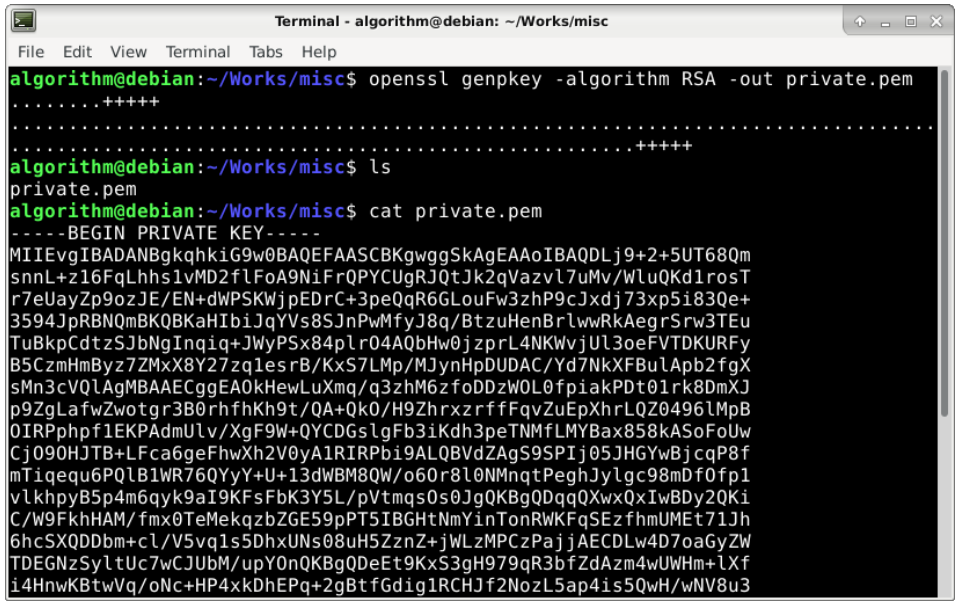
JWT

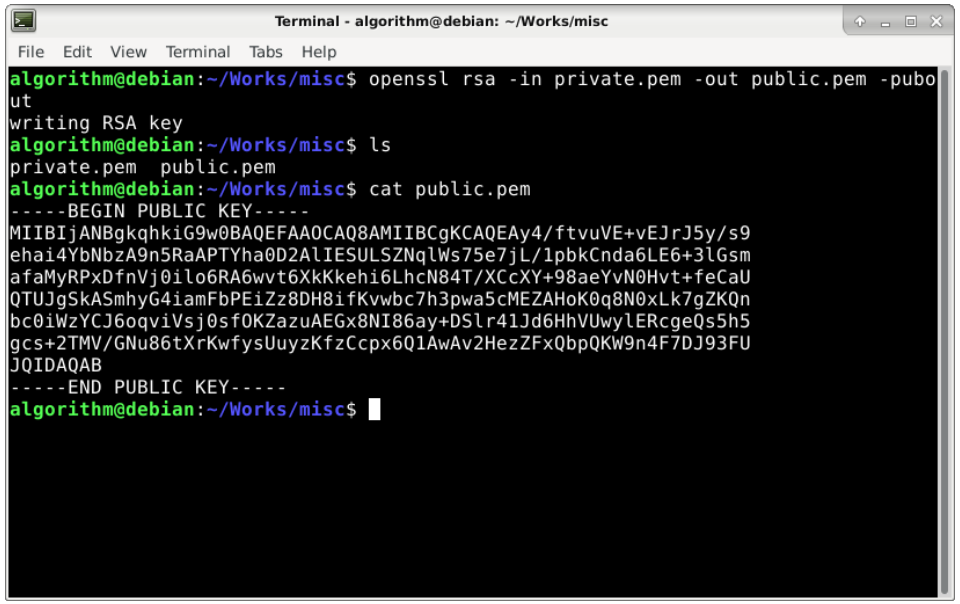

```
{
  "refresh":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cC
I6MTYyMjYzOTY4MCwianRpIjoiZWQ1OTMzMDcwMjczNDkzNGJjMWE2ZTg4YTZjNGRmZDMiLCJ1c2
VyX2lkIjoxfQ.xh2UGkQnGpww2N-w-hs55G89jTY2jXBzZCEtbXEq5z0",
  "access":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIj
oxNjIyNTUzMzQwLCJqdGkiOiI2MjQ2YzEwZTBhNTg0NTJhODE0NGZlOGMyZTM2MjNkYiIsInVzZX
JfaWQiOjF9.NBNDogF7VyyeIvk5GJ_JHt-FCWr1QQTuKrcji3JYhs4"
}
```

RSA

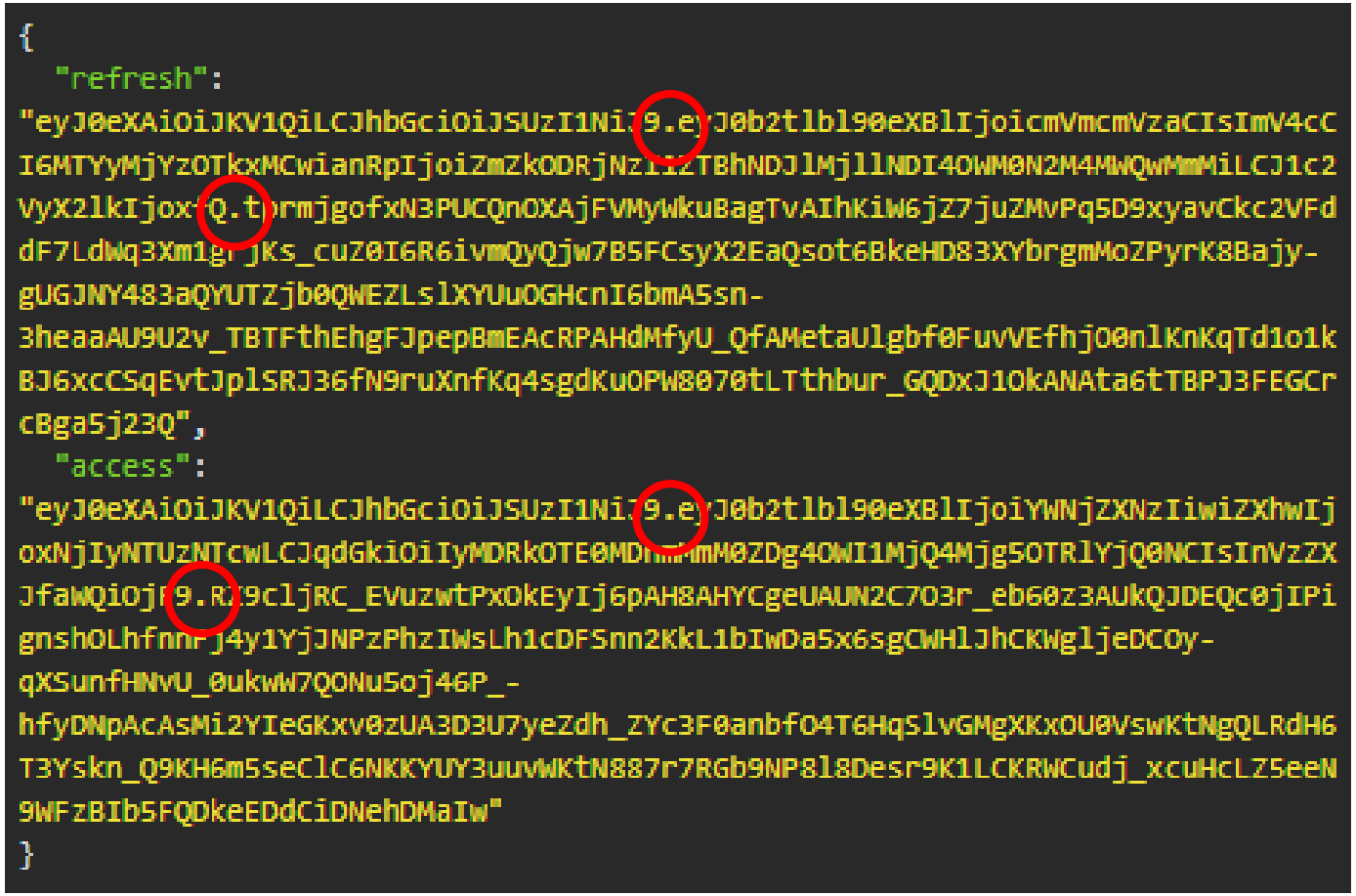
- PCKS(공개 키 암호 표준)의 일종
- RFC 3447
- 클라이언트단에서 노출되어도 안전하도록!

RSA

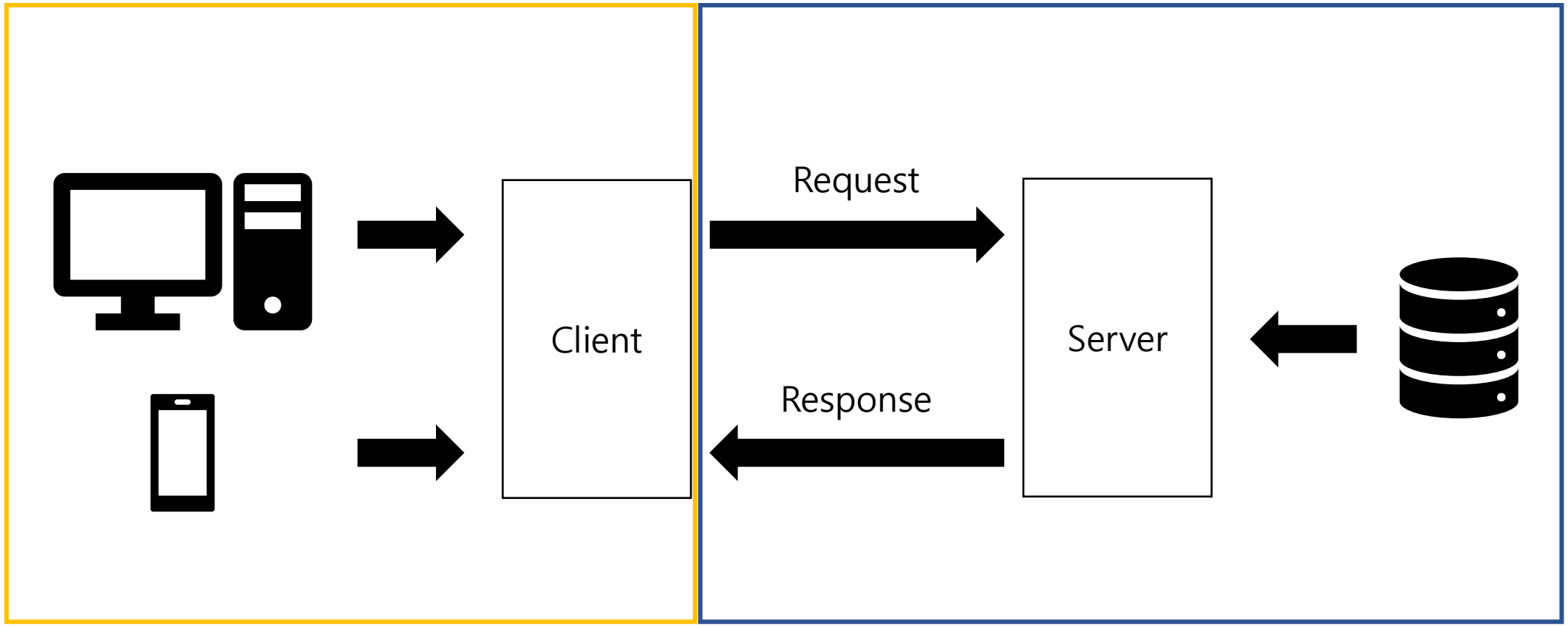

Terminal - algorithm@debian: ~/Works/misc
File Edit View Terminal Tabs Help
algorithm@debian:~/Works/misc$ openssl genpkey -algorithm RSA -out private.pem
.....+++++
.....+++++
algorithm@debian:~/Works/misc$ ls
private.pem
algorithm@debian:~/Works/misc$ cat private.pem
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDLj9+2+5UT68Qm
snnL+z16FqLhhs1vMD2flFoA9NiFrQPYCUgRJQtJk2qVazvl7uMv/WluQKd1rosT
r7eUayZp9ozJE/EN+dWPSKWjpEDrC+3peQqR6GLouFw3zhP9cJxdj73xp5i83Qe+
3594JpRBNQmBKQBKaHIbiJqYVs8SJnPwMfyJ8q/BtzuHenBrlwwRkAegrSrw3TEu
TuBkpCdtzSJbNgInqiq+JWyPSx84plrO4AQbHw0jzprL4NKWvjUl3oeFVTDKURFy
B5CzmHmByz7ZMxX8Y27zq1esrB/KxS7LMp/MJynHpDUDAC/Yd7NkXFBulApb2fgX
sMn3cVQlAgMBAAECggEAOkHewLuXmq/q3zhM6zfoDDzWOL0fpiakPDt01rk8DmXJ
p9ZgLafwZwotgr3B0rhfhKh9t/QA+QkO/H9ZhrxzrffFqvZuEpXhrLQZ0496lMpB
OIRPphpf1EKPAdmUlv/XgF9W+QYCDGslgFb3iKdh3peTNMfLMYBax858kASoFoUw
CjO9OHJTB+LFca6geFhwXh2V0yA1RIRPbi9ALQBVdZAgS9SPIj05JHGYwBjcqP8f
mTiqequ6PQlB1WR76QYyY+U+13dWBM8QW/o6Or8l0NMnqtPeghJylgc98mDfOfp1
vlkhpyB5p4m6qyk9aI9KFsFbK3Y5L/pVtmqsOs0JgQKBgQDqqQXwxQxIwBDy2QKi
C/W9FkhHAM/fmx0TeMekqzbZGE59pPT5IBGHtNmYinTonRWKFqSEzfhmUMEt71Jh
6hcSXQDDbm+cl/V5vq1s5DhxUNs08uH5ZznZ+jWLzMPCzPajjAECDLw4D7oaGyZW
TDEGNzSyltUc7wCJUbM/upYOnQKBgQDeEt9KxS3gH979qR3bfZdAzm4wUWHm+lXf
i4HnwKBtwVq/oNc+HP4xkDhEPq+2gBtfGdig1RCHJf2NozL5ap4is5QwH/wNV8u3


Terminal - algorithm@debian: ~/Works/misc
File Edit View Terminal Tabs Help
algorithm@debian:~/Works/misc$ openssl rsa -in private.pem -out public.pem -pubout
writing RSA key
algorithm@debian:~/Works/misc$ ls
private.pem public.pem
algorithm@debian:~/Works/misc$ cat public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAy4/ftvuVE+vEJrJ5y/s9
ehai4YbNbzA9n5RaAPTYha0D2AlIESULSZNqlWs75e7jL/1pbkCnda6LE6+3lGsm
afaMyRPxDfnVj0ilo6RA6wvt6XkKkehi6LhcN84T/XCcXY+98aeYvN0Hvt+feCaU
QTUJgSkASmhyG4iamFbPEiZz8DH8ifKvwbc7h3pwa5cMEZAHoK0q8N0xLk7gZKQn
bc0iWzYCJ6oqviVsj0sfOKZazuAEGx8NI86ay+DSlr41Jd6HhVUwylERcgeQs5h5
gcs+2TMV/GNu86tXrKwfysUuyzKfzCcpx6Q1AwAv2HezZFxQbpQKW9n4F7DJ93FU
JQIDAQAB
-----END PUBLIC KEY-----
algorithm@debian:~/Works/misc$

JWT in RSA-256



HTTP 구조



추가적으로 이 곳을 암호화!

서비스 제공자가 신경쓸 수 있음!

예시

```
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': ['rest_framework_simplejwt.authentication.JWTAuthentication']
}

SIMPLE_JWT = {
    'ACCESS_TOKEN_LIFETIME': timedelta(minutes=1),

    'ALGORITHM': 'RS256',
    'SIGNING_KEY': parse_file(getcwd() + "/private_key.pem")[0].as_text(),
    'VERIFYING_KEY': parse_file(getcwd() + "/public_key.pem")[0].as_text()
}
```

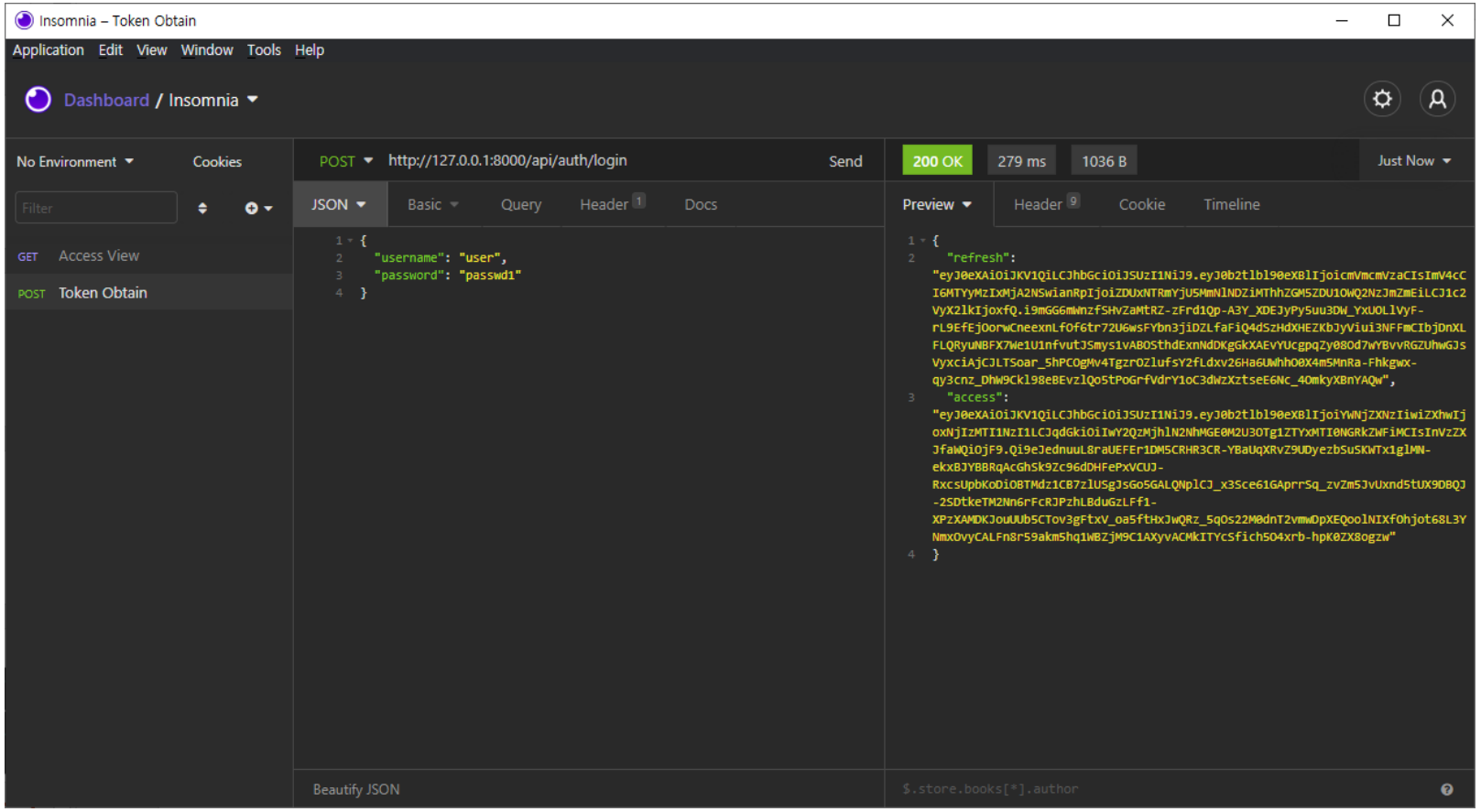
catlas > api > urls.py > ...

```
from django.contrib import admin
from django.urls import path, include

from rest_framework_simplejwt.views import TokenObtainPairView, TokenVerifyView

from .views import MyListClass

urlpatterns = [
    path('auth/login', TokenObtainPairView.as_view(), name='login'),
    path('auth/verify', TokenVerifyView.as_view(), name='verify'),

    path('list', MyListClass.as_view())
]
```

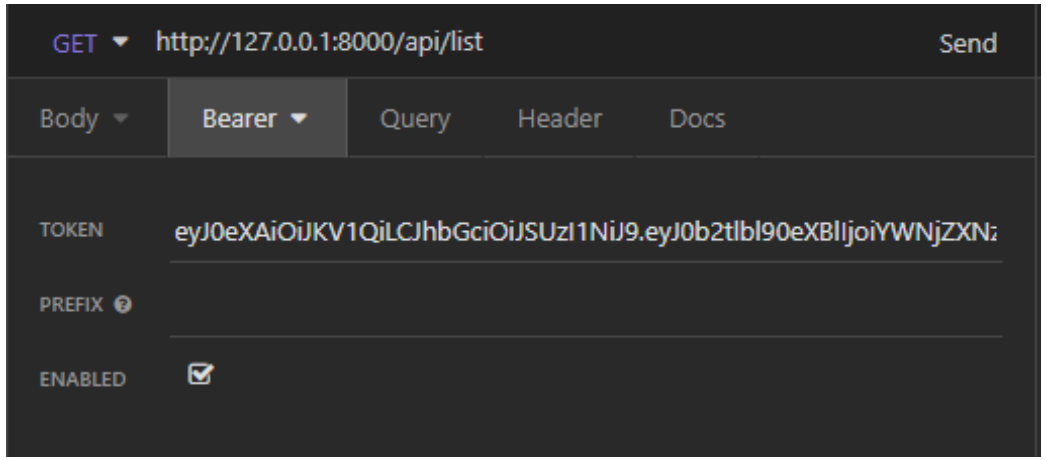
catlas > api > views.py > ...

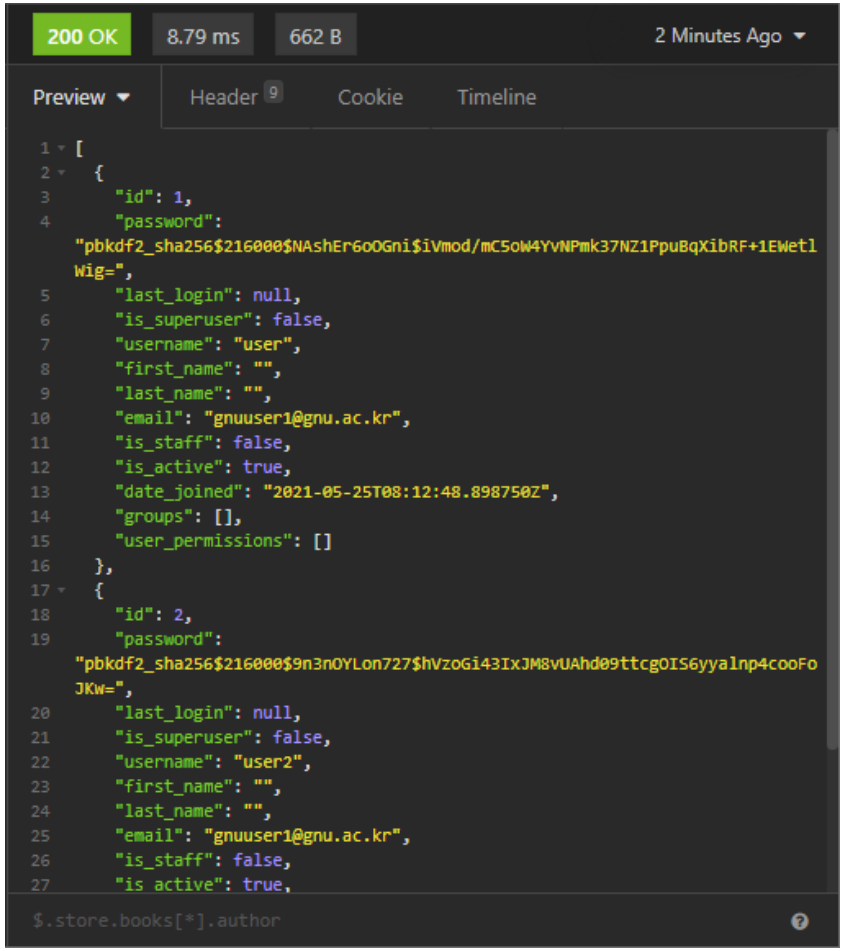
```
from django.shortcuts import render
from rest_framework import permissions
from rest_framework.generics import ListAPIView
from django.contrib.auth.models import User
from rest_framework.permissions import IsAuthenticated

from .serializers import UserSerializer

# Create your views here.

class MyListClass(ListAPIView):
    queryset = User.objects.all()
    serializer_class = UserSerializer
    permission_classes = [IsAuthenticated]
```

예시


Insomnia – Token Obtain
Application Edit View Window Tools Help
Dashboard / Insomnia
No Environment
Cookies
Filter
GET Access View
POST Token Obtain
POST http://127.0.0.1:8000/api/auth/login
Send
JSON Basic Query Header 1 Docs
{
"username": "user",
"password": "passwd1"
}
Beautify JSON
200 OK
279 ms
1036 B
Just Now
Preview Header 9 Cookie Timeline
{
"refresh":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTYyMzIxMjA2NSwianRpIjoiZDUxNTRmYjU5MmNlNDZiMThhZGM5ZDU1OWQ2NzJmZmEiLCJ1c2VyX2lkIjoxfQ.i9mGG6mWnzfSHvZaMtRZ-zFrd1Qp-A3Y_XDEJyPy5uu3DW_YxUOLlVyF-rL9EfEjOorwCneexnLfOf6tr72U6wsFYbn3jiDZLfaFiQ4dSzHdXHEZKbJyViui3NFFmCIbjDnXLFLQRyuNBFX7We1U1nfvutJSmys1vABOSthdExnNdDKgGkXAEvYUcgpqZy08Od7WYBvvRGZUhwGJsVyxciAjCJLTSoar_5hPCOgMv4TgzrOzlufsY2fLdxv26Ha6UWhhO0X4m5MnRa-Fhkgwx-qy3cnz_DhW9Ckl98eBEvzlQo5tPoGrfVdrY1oC3dWzXztseE6Nc_4OmkyXBnYAQw",
"access":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjIzMTI1NzI1LCJqdGkiOiIwY2QzMjhlN2NhMGE0M2U3OTg1ZTYxMTI0NGRkZWFiMCIsInVzZXJfaWQiOjF9.Qi9eJednuuL8raUEFEr1DM5CRHR3CR-YBaUqXRvZ9UDyezbSuSKWTx1glMN-ekxBJYBBRqAcGhSk9Zc96dDHFePxVCUJ-RxcsUpbKoDiOBTMdz1CB7zlUSgJsGo5GALQNplCJ_x3Sce61GAprrSq_zvZm5JvUxnd5tUX9DBQJ-2SDtkeTM2NN6rFcRJPzhLBduGzLFf1-XPzXAMDKJouUUb5CTov3gFtxV_oa5ftHxJwQRz_5qOs22M0dnT2vmwDpXEQoolNIXfOhjot68L3YNmxOvyCALFn8r59akm5hq1WBZjM9C1AXyvACMkITYcSfich5O4xrb-hpK0ZX8ogzw"
}
$.store.books[*].author

예시


GET
http://127.0.0.1:8000/api/list
Send
Body
Bearer
Query
Header
Docs
TOKEN
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXN
PREFIX
ENABLED


200 OK
8.79 ms
662 B
2 Minutes Ago
Preview
Header 9
Cookie
Timeline
[
{
"id": 1,
"password": "pbkdf2_sha256$216000$NAshEr6oOGni$iVmod/mC5oW4YvNPmk37NZ1PpuBqXibRF+1EWetlWig=",
"last_login": null,
"is_superuser": false,
"username": "user",
"first_name": "",
"last_name": "",
"email": "gnuuser1@gnu.ac.kr",
"is_staff": false,
"is_active": true,
"date_joined": "2021-05-25T08:12:48.898750Z",
"groups": [],
"user_permissions": []
},
{
"id": 2,
"password": "pbkdf2_sha256$216000$9n3nOYLon727$hVzoGi43IxJM8vUAhd09ttcgOIS6yyalnp4cooFoJKW=",
"last_login": null,
"is_superuser": false,
"username": "user2",
"first_name": "",
"last_name": "",
"email": "gnuuser1@gnu.ac.kr",
"is_staff": false,
"is_active": true,
$.store.books[*].author

이후 계획

- 서비스 완성 후 Dockerize
- SLAM 및 ORB-SLAM 관련 논문 확인
- ORB-SLAM 빌드 및 예제 확인