Token Blacklist

Prepare:

Clone the repository in your local disk.

Now create the virtualenv using below commands.

Install the virtualenv

```
pip install virtualenv
```

Initialize the venv in your local dir

```
virtualenv venv
```

Activate the virtualenv

```
source venv/bin/activate
```

Now you are good to install all of your pip packages.

```
.//venv/bin/pip install -r requirements.txt
```

Now the app is ready to test. This app is created in Flask and it has 3 API Endpoints

GET /info
POST /blacklist
DELETE /blacklist

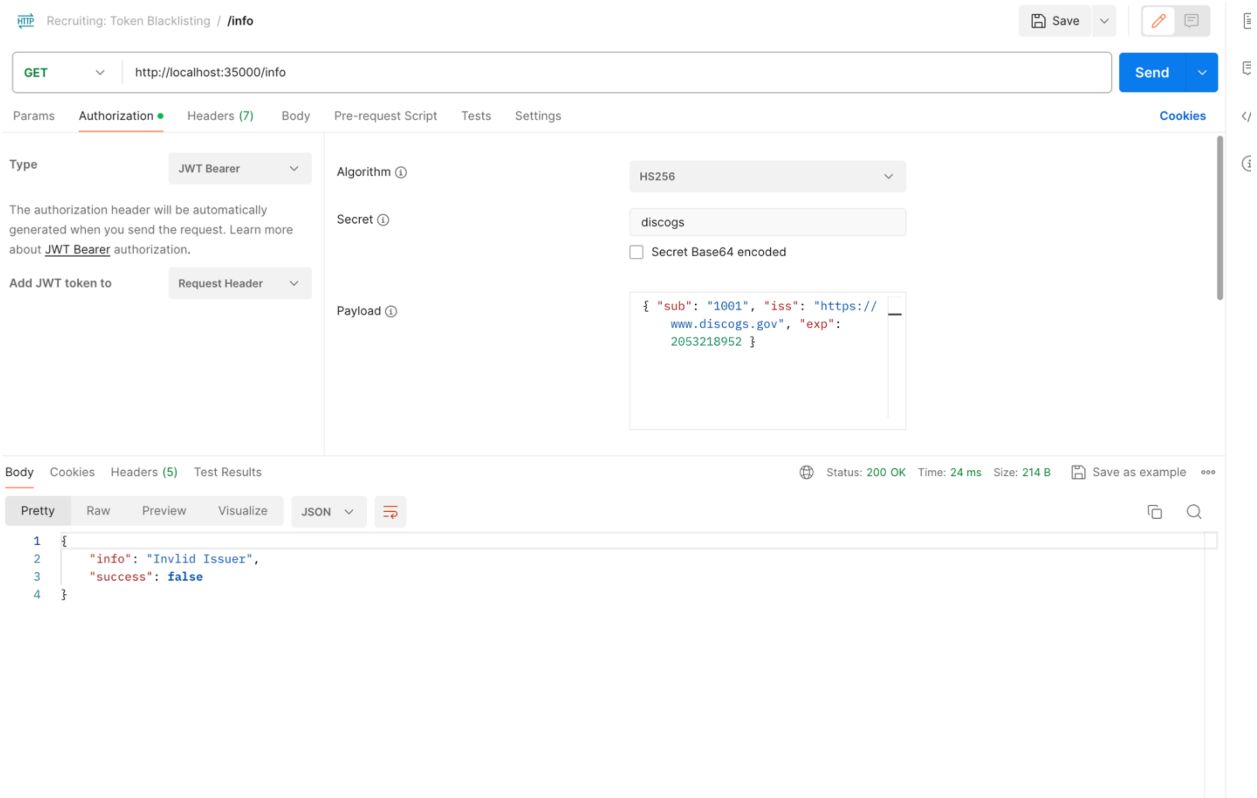
Github repo:
https://github.com/keypat1906/token_blacklist
Let me know if you any issue accessing this repository.

GET /info

This endpoint creates the JWT token using the parameters (issuer, secret, algorithm, expiry) we pass and it creates the token. This endpoint also validates the new JWT token which is created against the any error related to JWT token. If there is any error it returns error.

It also checks the issuer is "`https://www.discogs.com/`

If there is wrong issuers, it will throw error



GET http://localhost:35000/info

Type: JWT Bearer

The authorization header will be automatically generated when you send the request. Learn more about JWT Bearer authorization.

Add JWT token to: Request Header

Algorithm: HS256

Secret: discogs

□ Secret Base64 encoded

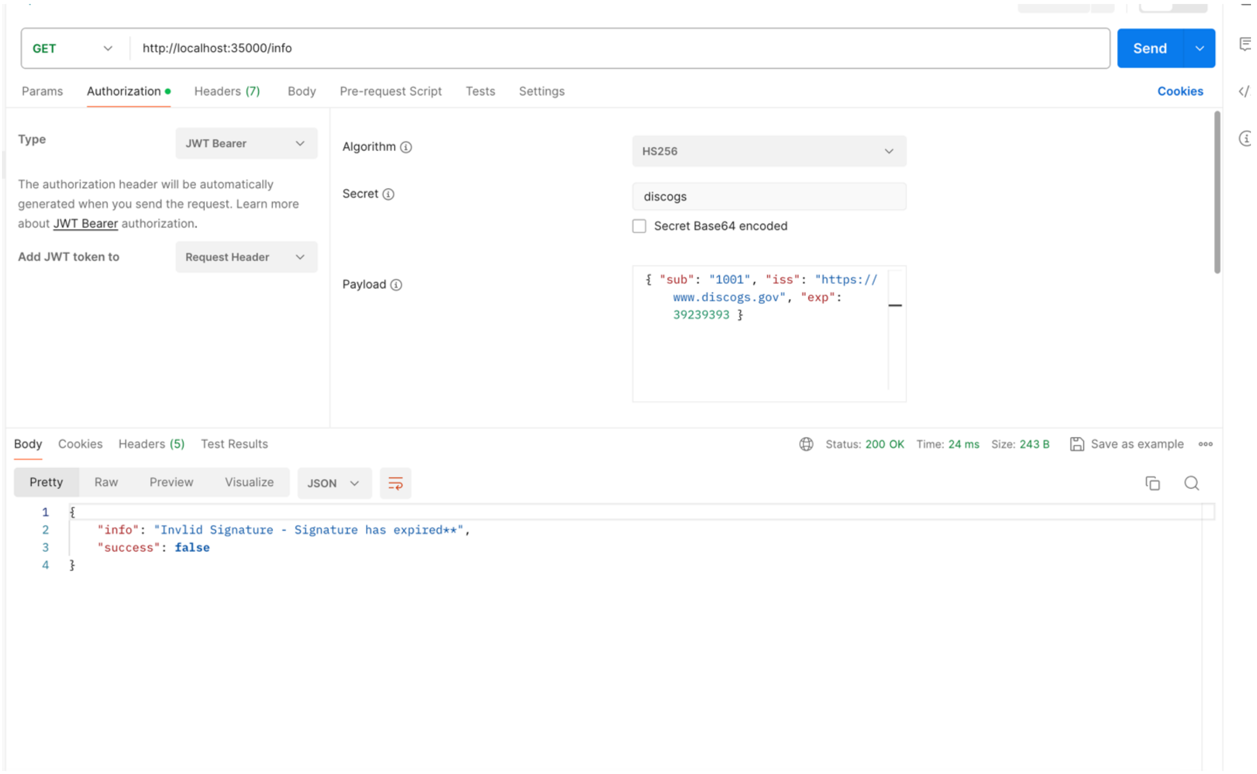
Payload:

```
{ "sub": "1001", "iss": "https://
   www.discogs.gov", "exp":
   2053218952 }
```

Status: 200 OK Time: 24 ms Size: 214 B

```
{
    "info": "Invlid Issuer",
    "success": false
}
```

If token is expired, it will throw error



GET http://localhost:35000/info

Type: JWT Bearer

The authorization header will be automatically generated when you send the request. Learn more about JWT Bearer authorization.

Add JWT token to: Request Header

Algorithm: HS256

Secret: discogs

□ Secret Base64 encoded

Payload:

```
{ "sub": "1001", "iss": "https://
   www.discogs.gov", "exp":
   39239393 }
```

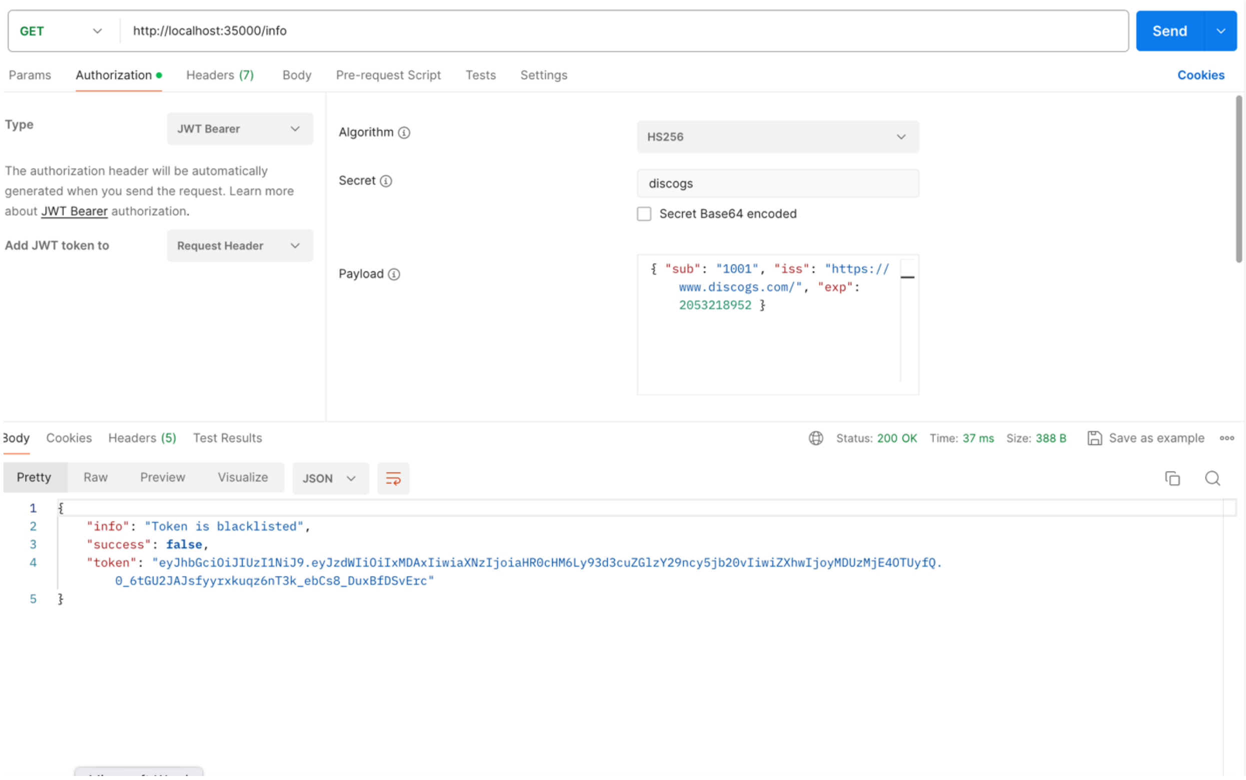
Status: 200 OK Time: 24 ms Size: 243 B

```
{
    "info": "Invlid Signature - Signature has expired**",
    "success": false
}
```

POST /blacklist

This endpoint gets the JWT token in the header and blacklist it. Internally it stores in the Set data structure so that we can check which token is blacklisted or not.

If you try to use the same JWT token on GET /info endpoint, it will throw error



Type: JWT Bearer

The authorization header will be automatically generated when you send the request. Learn more about JWT Bearer authorization.

Add JWT token to: Request Header

Algorithm: HS256

Secret: discogs

□ Secret Base64 encoded

Payload:

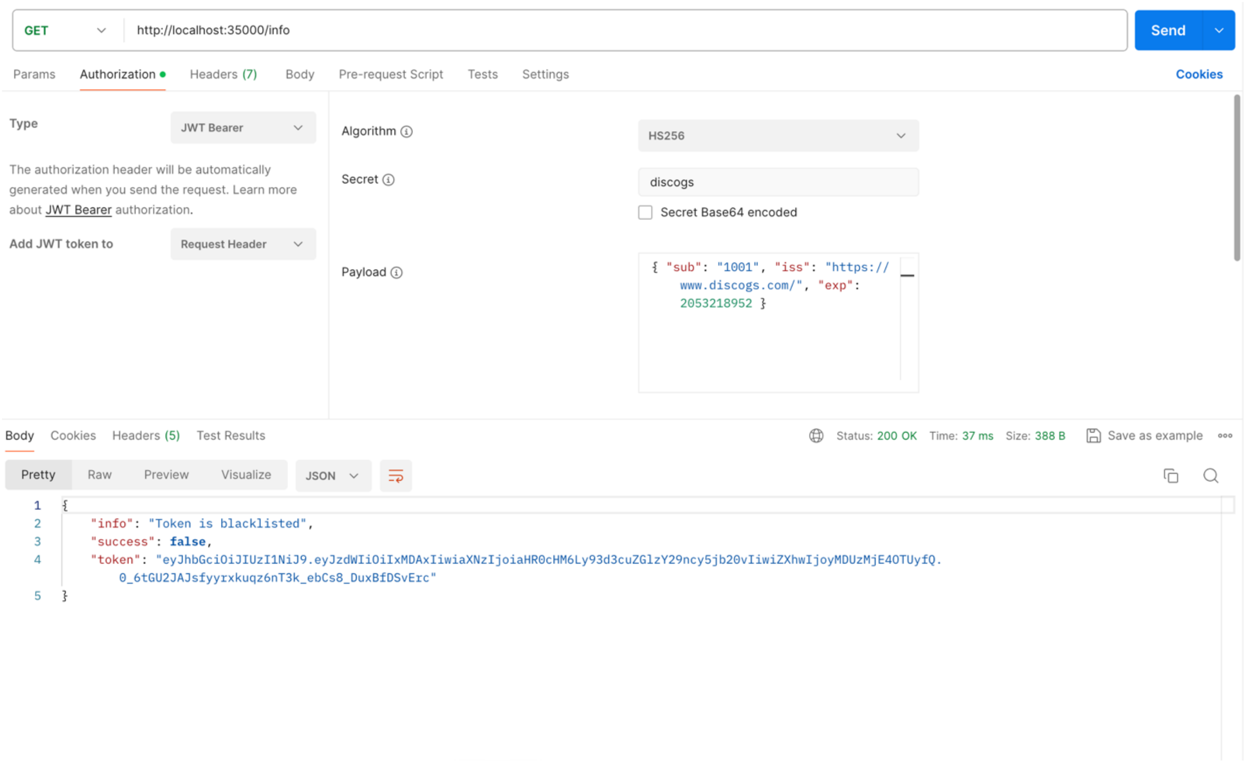
```
{ "sub": "1001", "iss": "https://
    www.discogs.com/", "exp":
    2053218952 }
```

Status: 200 OK Time: 37 ms Size: 388 B

```
{
    "info": "Token is blacklisted",
    "success": false,
    "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMDAxIiwiaXNzIjoiaHR0cHM6Ly93d3cuZGlzY29ncy5jb20vIiwiZXhwIjoyMDUzMjE4OTUyfQ.
        0_6tGU2JAJsfyyrxkuqz6nT3k_ebCs8_DuxBfDSvErc"
}
```

DELETE /blacklist

This endpoint removes the blacklisted JWT token from the Set data structure.


GET http://localhost:35000/info

Type: JWT Bearer

The authorization header will be automatically generated when you send the request. Learn more about JWT Bearer authorization.

Add JWT token to: Request Header

Algorithm: HS256

Secret: discogs

□ Secret Base64 encoded

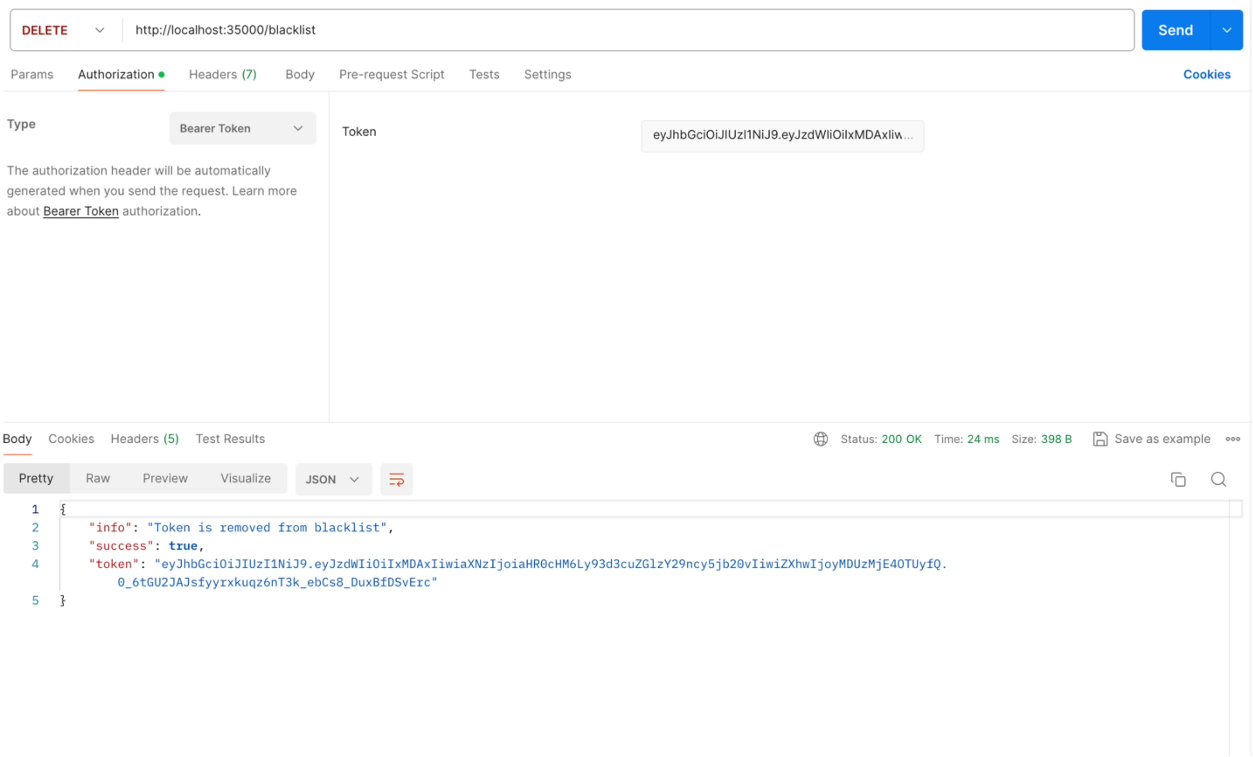
Payload:

```
{ "sub": "1001", "iss": "https://
  www.discogs.com/", "exp":
  2053218952 }
```

Status: 200 OK Time: 37 ms Size: 388 B

```
{
    "info": "Token is blacklisted",
    "success": false,
    "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMDAxIiwiaXNzIjoiaHR0cHM6Ly93d3cuZGlzY29ncy5jb20vIiwiZXhwIjoyMDUzMjE4OTUyfQ.
        0_6tGU2JAJsfyyrxkuqz6nT3k_ebCs8_DuxBfDSvErc"
}
```

And now try the endpoint GET /info and you should see the token validated again.


DELETE http://localhost:35000/blacklist

Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about Bearer Token authorization.

Token: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMDAxIiw...

Status: 200 OK Time: 24 ms Size: 398 B

```
{
    "info": "Token is removed from blacklist",
    "success": true,
    "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMDAxIiwiaXNzIjoiaHR0cHM6Ly93d3cuZGlzY29ncy5jb20vIiwiZXhwIjoyMDUzMjE4OTUyfQ.
        0_6tGU2JAJsfyyrxkuqz6nT3k_ebCs8_DuxBfDSvErc"
}
```

Improvement:

There is definitely area of improvement here. Currently it stores all the JWT token in the Set data structure. If there is server reload happen, it will lose all the blacklisted tokens. For improvement, I can store these JWT token in the Sqllite or Postgres database and keep it in one table.

If any JWT token is blacklisted, it will contain an entry in table with Blacklisted=True and if that JWT token is removed from blacklist, I will update the table row with Blacklisted=False.
That way we have all the history of the JWT token blacklisted

Any feedback or criticism welcome. You can email me at keyurpatel80@gmail.com

Thanks