

علیرضا سلحشور

9931308

پروژه اول درس اصول طراحی پایگاه داده

Redis Database

استاد درس : دکتر حمیدرضا شهریاری

اسفند 1402

در این پروژه قصد ایجاد یک پایگاه داده Redis با استفاده از یک python client را داریم که از redis-py استفاده کردیم.

```
Alireza > pip install redis
```

ابتدا برای نصب این کتابخانه از دستور

استفاده میکنیم.

```
Alireza > pip install "redis[hiredis]"
```

یا از

برای برقراری ارتباط با client ابتدا redis-server.exe را اجرا کرده و مشاهده میکنیم که سرور آماده برقراری ارتباط از طریق پورت 6379 می باشد.

```
[8224] 14 Mar 12:21:49.630 # Server started, Redis version 3.2.100
[8224] 14 Mar 12:21:49.630 * The server is now ready to accept connections on port 6379
```

برای اطمینان این ارتباط همچنین میتوانیم redis-cli.exe را اجرا کنیم. و میبینیم که سرور از پورت 6379 در حال ارتباط می باشد.

```
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> ping "hello"
"hello"
```

سپس برای برقراری از طریق python یک client ایجاد کرده و از طریق آن با پایگاه کار میکنیم.

```
client = redis.Redis(host='localhost', port=6379, db=0)
```

برای شروع پروژه ابتدا نیازمندی های پروژه را نصب میکنیم.

```
Alireza > pip install -r "requirements.txt"
Requirement already satisfied: cryptography in c:\users\alireza\appdata\local\programs\python\python311\lib\site-packages
s (from -r requirements.txt (line 1)) (42.0.5)
Requirement already satisfied: cffi>=1.12 in c:\users\alireza\appdata\local\programs\python\python311\lib\site-packages
(from cryptography->r requirements.txt (line 1)) (1.15.1)
Requirement already satisfied: pycparser in c:\users\alireza\appdata\local\programs\python\python311\lib\site-packages (
from cffi>=1.12->cryptography->r requirements.txt (line 1)) (2.21)
```

پیاده سازی متد ها:

CREATE •

در ابتدا بررسی میکنیم که کلید در پایگاه داده موجود است یا نه، در صورت وجود پیامی به این منظور نمایش میدهیم در غیر اینصورت ابتدا رمز را رمزنگاری کرده و تمام موارد کلید را در یک دیکشنری ذخیره و شی password را به صورت یک json ایجاد کرده و آن را ذخیره میکنیم.

```
def create(self, name, description, key, exp):
    """
    encrypt the given key
    create a Password object
    store the json format of object in redis
    print the encrypted key
    """
    if client.exists(name):
        print(f"The key '{name}' already exists.")
    else:
        encrypted_key = self._cipher.encrypt(key)

        value = {"name" : name,
                  "description": description,
                  "key": encrypted_key,
                  "exp": exp}
        password = json.dumps(value)

        client.set(name, password)
        client.expire(name, exp)

        print(encrypted_key)
```

برای اطمینان از درستی این تابع دستور نمونه ای را به صورت زیر اجرا می کنیم :

```
● 425ms Alireza > python main.py create -n portal -c my student portal key -key 1234 -exp 300
//GDocmAn3TEN3Zuz4CdrZgTZdx8VcfxGBCbAG9E6Q=

● 453ms Alireza > python main.py create -n portal -c my student portal key -key 1234 -exp 300
The key 'portal' already exists.
```

READ •

با دریافت یک نام، از طریق client درخواست برای دریافت مقدار ارسال میکنیم و بررسی میکنیم در صورت عدم وجود نام پیامی نمایش داده شود.

```
def read(self, name):
    """
    retrieves the password from redis
    prints the password info
    """
    value = client.get(name)
    if value:
        for key, value in json.loads(value).items():
            print(f"{key}: {value}")
    else :
        print(f"The key '{name}' does not exist.")
```

برای اطمینان از درستی این تابع دستور نمونه ای را به صورت زیر اجرا می کنیم :

```
• Alireza > python main.py read portal
name: portal
description: my student portal key
key: wIDrJ1TaFz0adx4HaJaq9ek6TvkNuSkTYMQNGFvgwe8=
exp: 300

• 390ms Alireza > python main.py read port
The key 'port' does not exist.
```

UPDATE •

مانند موارد قبل ابتدا وجود کلید را بررسی و در صورت عدم وجود آن را به روز رسانی میکنیم.

```
def update(self, name, new_key):
    """
    encrypts new key
    saves the password with updated attributes
    prints new value for encrypted key
    """

    if client.exists(name):
        key = self._cipher.encrypt(new_key)
        client.set(name, key)
        print(f"The key '{name}' has been successfully updated.")
        print(f"updated key {key}")
    else :
        print(f"The key '{name}' does not exist.")
```

نمونه :

```
386ms Alireza > python main.py create -n portal -c my student portal key -key 1234 -exp 300
aIAbPdIYc6b1ufY70e/bnjzsVZSX8hAueR2jG1sGIItI=
codes ↓ 100

463ms Alireza > python main.py update portal -key 54321
The key 'portal' has been successfully updated.
updated key p/HkUkHaeLNQx1SNKk3+iwAQLcibtE0STnvtW51hR6os=
codes ↓ 100

465ms Alireza > python main.py update portal1 -key 54321
The key 'portal1' does not exist.
codes ↓ 100
```

DELETE •

در ابتدا بررسی می کنیم که کلید موجود است یا نه. در صورت وجود ابتدا آن را پاک کرده و دوباره بررسی میکنیم که پاک شده یا خیر و سپس پیام متناظر را نمایش می دهیم.

```
def delete(self, name):  
    """  
    deletes the password from redis  
    prints a success message  
    """  
    if client.exists(name):  
        client.delete(name)  
        if not client.exists(name):  
            print(f"The key '{name}' has been successfully deleted.")  
        else:  
            print(f"Failed to delete the key '{name}'.")  
    else:  
        print(f"The key '{name}' does not exist.")
```

نمونه :

```
• 387ms Alireza > python main.py delete portal  
The key 'portal' has been successfully deleted.  
  
• 347ms Alireza > python main.py delete portal  
The key 'portal' does not exist.
```

LIST •

```
def list(self):
    """
    gets all the saved passwords from redis
    prints each password info
    """
    keys = client.keys()
    values = client.mget(keys)

    if not values:
        print("No passwords found.")
    else :
        print("List of all key and values ")
        print("=====")
        for value in values:
            for key, val in json.loads(value).items():
                print(f"{key}: {val}")
        print("=====")
```

نمونه :

```
• 415ms Alireza > python main.py list
No passwords found.

• 394ms Alireza > python main.py create -n portal -c my student portal key -key 1234 -exp 300
B8c5/1NAj8mmhkc9WDAtAZXR4EV1PBMUKaUctv1xWw8=

• 386ms Alireza > python main.py create -n portal1 -c my student portal key -key 11234 -exp 300
ds8ydQ9Jukmv02QbJl+YDsLhcHfH99phK0JSMZp/wmA=

• 431ms Alireza > python main.py create -n portal2 -c my student portal key -key 1123 -exp 300
YXIXE0TV8bP60B7RB98fiJjf1/nTSXQqZEoPOSK2HwK=

• 405ms Alireza > python main.py list
List of all key and values
=====
name: portal1
description: my student portal key
key: ds8ydQ9Jukmv02QbJl+YDsLhcHfH99phK0JSMZp/wmA=
exp: 300
=====
name: portal2
description: my student portal key
key: YXIXE0TV8bP60B7RB98fiJjf1/nTSXQqZEoPOSK2HwK=
exp: 300
=====
name: portal
description: my student portal key
key: B8c5/1NAj8mmhkc9WDAtAZXR4EV1PBMUKaUctv1xWw8=
exp: 300
=====
```

• get_remaining_time

```
def get_remaining_time(self, name):  
    """  
    retrieve the remaining time of password existence on redis  
    prints the remaining time  
    """  
    print(client.ttl(name))
```

نمونه :

```
● 356ms Alireza > python main.py rem portal  
210  
● 395ms Alireza > python main.py rem portal  
204  
● 382ms Alireza > python main.py rem portal  
199
```