|  |  |  |
| --- | --- | --- |
| Лабораторная работа №2  «Работа с Redis» | Дата | 26.05.2024 |
| Группа | ИВТ-363 |
| ФИО | Дементьев .А. |
| Преподаватель | Тарасов П.С. |

Скрипт на Lua:

-- Проверка и обновление счетчиков

local function check\_and\_update\_counters(user\_id, config\_key)

    local key = "rate\_limit:user:" .. user\_id

    local max\_requests = tonumber(redis.call("HGET", config\_key, "max\_requests")) or 0

    local interval = tonumber(redis.call("HGET", config\_key, "interval")) or 0

    local current = tonumber(redis.call("GET", key) or 0)

    if current >= max\_requests then

        return 0  -- Лимит запросов исчерпан

    else

        redis.call("INCR", key)

        if current == 0 then

            redis.call("EXPIRE", key, interval)

        end

        return 1  -- Запрос разрешен

    end

end

-- KEYS[1] - ключ, по которому будем отслеживать запросы

-- ARGV[1] - лимит запросов

-- ARGV[2] - время окна в секундах

local current = redis.call("GET", KEYS[1])

if current and tonumber(current) and tonumber(current) >= tonumber(ARGV[1]) then

    return 0

else

    current = redis.call("INCR", KEYS[1])

    if tonumber(current) == 1 then

        redis.call("EXPIRE", KEYS[1], ARGV[2])

    end

    return 1

end

-- Инициализация лимитера

local function initialize\_rate\_limiter(user\_id, max\_requests, interval)

    redis.call("SET", "rate\_limit:user:" .. user\_id, 0, "EX", interval)

    redis.call("HMSET", "rate\_limit\_config:user:" .. user\_id, "max\_requests", max\_requests, "interval", interval)

end

-- Блокировка пользователя

local function block\_user(user\_id, block\_duration)

    redis.call("SET", "block:user:" .. user\_id, 1, "EX", block\_duration)

end

-- Разблокировка пользователя

local function unblock\_user(user\_id)

    redis.call("DEL", "block:user:" .. user\_id)

end

return {

    initialize\_rate\_limiter = initialize\_rate\_limiter,

    check\_and\_update\_counters = check\_and\_update\_counters,

    block\_user = block\_user,

    unblock\_user = unblock\_user

}

Код на Python для загрузки Lua-скрипта на Redis:

import redis

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

with open('rate\_limit.lua', 'r', encoding='utf-8') as file:

    lua\_script = file.read()

script\_sha = client.script\_load(lua\_script)

def initialize\_rate\_limiter(user\_id, max\_requests, interval):

    client.evalsha(script\_sha, 0, "initialize\_rate\_limiter", user\_id, max\_requests, interval)

def check\_and\_update\_counters(user\_id):

    config\_key = "rate\_limit\_config:user:" + str(user\_id)

    response = client.evalsha(script\_sha, 0, "check\_and\_update\_counters", user\_id, config\_key)

    if response == 0:

        # Бан на 5 минут

        block\_user(user\_id, 300)

        print("Лимит запросов превышен. Доступ заблокирован.")

    elif response == 1:

        print("Запрос разрешен.")

    else:

        print("Произошла ошибка при выполнении Lua-скрипта.")

def block\_user(user\_id, block\_duration):

    client.evalsha(script\_sha, 0, "block\_user", user\_id, block\_duration)

def unblock\_user(user\_id):

    client.evalsha(script\_sha, 0, "unblock\_user", user\_id)

# Пользователь с ID 123, лимит 10 запросов за 60 секунд

initialize\_rate\_limiter(123, 10, 60)

# Проверка и обновление счетчиков

check\_and\_update\_counters(123)

Результат работы:

