

# OperationalError

```
psycopg2.OperationalError: connection to server at "mydatabase.ct15rimbfjcl.us-east-1.rds.amazonaws.com" (34.237.129.195), port 5430 failed: Connection timed out (0x0000274C/10060)
```

Is the server running on that host and accepting TCP/IP connections?

## Traceback (most recent call last)

```
File "C:\Users\Golian\Downloads\project1-omar\.venv\lib\site-packages\flask\app.py", line 2095, in __call__
```

```
    return self.wsgi_app(environ, start_response)
```

```
File "C:\Users\Golian\Downloads\project1-omar\.venv\lib\site-packages\flask\app.py", line 2080, in wsgi_app
```

```
    response = self.handle_exception(e)
```

```
File "C:\Users\Golian\Downloads\project1-omar\.venv\lib\site-packages\flask_cors\extension.py", line 165, in wrapped_function
```

```
    # Wrap exception handlers with cross_origin
```

```
    # These error handlers will still respect the behavior of the route
```

```
    if options.get('intercept_exceptions', True):
```

```
        def _after_request_decorator(f):
```

```
            def wrapped_function(*args, **kwargs):
```

```
                return cors_after_request(app.make_response(f(*args,
**kwargs)))
```

```
            return wrapped_function
```

```
        if hasattr(app, 'handle_exception'):
```

```
            app.handle_exception = _after_request_decorator(
```

```
                app.handle_exception)
```

```
File "C:\Users\Golian\Downloads\project1-omar\.venv\lib\site-packages\flask\app.py", line 2077, in wsgi_app
```

```
    response = self.full_dispatch_request()
```

```
File "C:\Users\Golian\Downloads\project1-omar\.venv\lib\site-packages\flask\app.py", line 1525, in full_dispatch_request
```

```
    rv = self.handle_user_exception(e)
```

```
File "C:\Users\Golian\Downloads\project1-omar\.venv\lib\site-packages\flask_cors\extension.py", line 165, in wrapped_function
```

```
    return cors_after_request(app.make_response(f(*args, **kwargs)))
```

```
File "C:\Users\Golian\Downloads\project1-omar\.venv\lib\site-packages\flask\app.py", line 1523, in full_dispatch_request
```

```
    rv = self.dispatch_request()
```

File "C:\Users\Golian\Downloads\project1-omar\.venv\lib\site-packages\flask\app.py", line 1509, in dispatch\_request

```
return self.ensure_sync(self.view_functions[rule.endpoint])(**req.view_args)
```

File "c:\Users\Golian\downloads\project1-omar\app.py", line 37, in login

```
user=select_user_by_username(username)
```

File "c:\Users\Golian\downloads\project1-omar\repository\userlogin\_dao.py", line 50, in select\_user\_by\_username

```
connection = get_connection()
```

File "c:\Users\Golian\downloads\project1-omar\repository\connection.py", line 4, in get\_connection

```
connection = psycopg2.connect(
```

File "C:\Users\Golian\Downloads\project1-omar\.venv\lib\site-packages\psycopg2\\_\_init\_\_.py", line 122, in connect

```
    kwasync['async'] = kwargs.pop('async')
```

```
    if 'async_' in kwargs:
```

```
        kwasync['async_'] = kwargs.pop('async_')
```

```
    dsn = _ext.make_dsn(dsn, **kwargs)
```

```
    conn = _connect(dsn, connection_factory=connection_factory, **kwasync)
```

```
    if cursor_factory is not None:
```

```
        conn.cursor_factory = cursor_factory
```

```
    return conn
```

psycopg2.OperationalError: connection to server at "mydatabase.ctl5rimbfjcl.us-east-1.rds.amazonaws.com" (34.237.129.195), port 5430 failed: Connection timed out (0x0000274C/10060)

Is the server running on that host and accepting TCP/IP connections?

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

- `dump()` shows all variables in the frame
- `dump(obj)` dumps all that's known about the object

Brought to you by **DON'T PANIC**, your friendly Werkzeug powered traceback interpreter.