

# Device Flow

```
def login(auth0_request_data):
    # . . .
    if device_code_response.status_code != 200:
        print('Error generating the device code')
        return

    print('Device code successful')
    data = device_code_response.json()

    print('1. Navigate to: ', data['verification_uri_complete'])
    print('2. Enter the following code: ', data['user_code'])
    draw_page(data)
    # . . .
```



```
from auth import device_flow as df
from auth.auth0 import config_auth0
from dino.leaderboard import set_user_score

# . . . previous code
elif display.pressed(BUTTON_B):
    print("Connecting to the internet...")
    if not display.isconnected():
        display.connect()
    clear_screen()

    print("Config Auth0")
    auth0_request_data = config_auth0()

    print("Requesting tokens")
    tokens = df.login(auth0_request_data)

    print("Setting user score with API...")
    set_user_score(tokens.get('access_token'))
```



# Device Flow

```
def login(auth0_request_data):
    # . . .
    user_confirmation = False
    elapsed_time = 0
    grant_type = (
        'urn:ietf:params:oauth:' +
        'grant-type:device_code'
    )
    device_code_data = (
        f"device_code={data['device_code']} +
        f"&client_id={auth0_request_data['client_id']} +
        f"&grant_type={grant_type}"
    )
    headers = {
        "Content-Type": "application/x-www-form-urlencoded"
    }
    url = "https://{}/oauth/token".format(
        auth0_request_data['domain'])
    # . . .
```



```
from auth import device_flow as df
from auth.auth0 import config_auth0
from dino.leaderboard import set_user_score

# . . . previous code
elif display.pressed(BUTTON_B):
    print("Connecting to the internet...")
    if not display.isconnected():
        display.connect()
    clear_screen()

    print("Config Auth0")
    auth0_request_data = config_auth0()

    print("Requesting tokens")
    tokens = df.login(auth0_request_data)

    print("Setting user score with API...")
    set_user_score(tokens.get('access_token'))
```