

Source: C:\Users\Admin\Desktop\CMC\ms1_email_ingestion\utils\token_manager.py

```
"""
Async token management
"""

import os
from msal import ConfidentialClientApplication
from concurrent_storage.redis_manager import get_redis_storage
from utils.config import settings

KEY_REFRESH_TOKEN = "auth:refresh_token"

async def get_token() -> str:
    """
    Get access token (async)

    Returns:
        Access token string
    """
    redis = await get_redis_storage()

    # Get refresh token from Redis
    refresh_token = await redis.get(KEY_REFRESH_TOKEN)

    if not refresh_token:
        raise Exception("Refresh token not found. Run login first.")

    # Get new access token (blocking MSAL call)
    # Run in thread pool to avoid blocking event loop
    import asyncio
    loop = asyncio.get_event_loop()

    def _get_token():
        client = ConfidentialClientApplication(
            client_id=settings.CLIENT_ID,
            client_credential=settings.CLIENT_SECRET
        )

        result = client.acquire_token_by_refresh_token(
            refresh_token,
            scopes=settings.SCOPES
        )

        if "access_token" not in result:
            raise Exception(f"Token refresh failed: {result.get('error_description')}")

        return result["access_token"]

    access_token = await loop.run_in_executor(None, _get_token)
    return access_token

async def save_refresh_token(refresh_token: str):
    """Save refresh token to Redis"""
    redis = await get_redis_storage()
    await redis.set(KEY_REFRESH_TOKEN, refresh_token)
    print("[TokenManager] Refresh token saved")

def login_interactive():
    """
    Interactive login (run once to get refresh token)
    This is SYNC - run separately
    """
    client = ConfidentialClientApplication(
        client_id=settings.CLIENT_ID,
        client_credential=settings.CLIENT_SECRET
    )

    auth_url = client.get_authorization_request_url(
        scopes=settings.SCOPES,
        redirect_uri="http://localhost:8000/callback"
    )
```

```
print(f"Open this URL to login:\n{auth_url}\n")
auth_code = input("Paste authorization code: ").strip()

result = client.acquire_token_by_authorization_code(
    code=auth_code,
    scopes=settings.SCOPES,
    redirect_uri="http://localhost:8000/callback"
)

if "refresh_token" not in result:
    print(f"Login failed: {result}")
    return None

return result[ "refresh_token" ]
```