


```
if USE_INTROSPECTION:
    # Use introspection to validate the token
    introspection_result = await introspect_token(token)
    if not introspection_result.get("active"):
        raise HTTPException(
            status_code=status.HTTP_401_UNAUTHORIZED, detail="Token is inactive or invalid"
        )
    # Extract user data and scopes from introspection result
    user_data = {
        "sub": introspection_result["sub"],
        "username": introspection_result.get("username", "unknown"),
    }
    token_scopes = introspection_result.get("scope", "").split()

    # Validate the 'aud' claim
    if introspection_result.get("aud") != "api://default": # Replace with your expected audience
        raise HTTPException(
            status_code=status.HTTP_401_UNAUTHORIZED, detail="Invalid audience"
        )
```

```
# Validate required scopes
for scope in SCOPES:
    if scope not in token_scopes:
        print(f"Missing required scope: {scope}")
        raise HTTPException(
            status_code=status.HTTP_403_FORBIDDEN, detail="Insufficient scope"
        )

print(f"User data: {user_data}")
return User(**user_data)
```