

Scope check for both remote and local validation

```
# 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"
        )
# Returns a user object when all checks pass
print(f"User data: {user_data}")
return User(**user_data)
```

Exception handling

```
except ExpiredTokenError as e:
    raise HTTPException(
        status_code=status.HTTP_401_UNAUTHORIZED, detail="Token has expired"
    )
except DecodeError as e:
    raise HTTPException(
        status_code=status.HTTP_401_UNAUTHORIZED, detail=f"Token decoding failed: {str(e)}"
    )
except JoseError as e:
    raise HTTPException(
        status_code=status.HTTP_401_UNAUTHORIZED, detail=f"Invalid token: {str(e)}"
    )
except Exception as e:
    raise HTTPException(
        status_code=status.HTTP_401_UNAUTHORIZED, detail=f"Token verification failed: {str(e)}"
    )
```