

# HTTPBearer scheme

```
async def verify_token(credentials: HTTPAuthorizationCredentials = Depends(bearer_scheme)) → User:
    """Verifies the incoming access token using FastAPI's HTTPBearer."""
    token = credentials.credentials # Extract the token from the Authorization header
    # print(f"Extracted token: {token}")
```

# Introspect endpoint (remote validation w/IdP)

```
async def introspect_token(token: str, token_type_hint: str = "access_token") → dict:
    """Introspects the token using the authorization server's introspection endpoint."""
    async with httpx.AsyncClient() as client:
        response = await client.post(
            INTROSPECTION_ENDPOINT,
            data={
                "token": token,
                "token_type_hint": token_type_hint,
                "client_id": CLIENT_ID,
            },
            headers={
                "Content-Type": "application/x-www-form-urlencoded",
            },
        )
        response.raise_for_status()
    return response.json()
```