


```
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)}"
    )
```

Exception handling

```
@app.get("/api/messages")
async def get_messages(user: User = Depends(verify_token)):
    """Endpoint to retrieve messages. Requires authentication."""
    messages = [
        {
            "date": datetime.now(),
            "text": "You are free from the dungeon!"
        }
    ]
    return {"messages": messages}

@app.get("/api/public")
async def public_endpoint():
    return {"message": "Public endpoint, no auth required"}
```

Dependency Injection