```
        expire = data.get("expires")

        if expire is None:
            raise HTTPException(
                status_code=status.HTTP_400_BAD_REQUEST,
                detail="No access token supplied"
            )
        if datetime.utcnow() > datetime.utcfromtimestamp(expire):
            raise HTTPException(
                status_code=status.HTTP_403_FORBIDDEN,
                detail="Token expired!"
            )
        return data

    except JWTError:
        raise HTTPException(
            status_code=status.HTTP_400_BAD_REQUEST,
            detail="Invalid token"
        )
```

함수가 토큰을 문자열로 받아 try 블록 내에서 여러 가지 확인 작업을 거친다. 가장 먼저 확인하는 것은 토큰의 만료 시간이 존재하는지 여부다. 만료 시간이 없으면 유효한 토큰이 존재하지 않는다고 판단한다. 두 번째로 확인하는 것은 토큰이 유효한지(만료 시간이 지나지 않았는지) 여부다. 토큰이 유효하다면 디코딩된 페이로드를 반환한다. 마지막으로 except 블록에서는 JWT 요청 자체에 오류가 있는지 확인한다.

지금까지 애플리케이션으로 전달된 토큰을 검증하는 함수를 만들었다. 이어서 사용자 인증을 검증하고 의존 라이브러리로 사용할 함수를 만들어보자.

사용자 인증

JWT 생성 및 디코딩하는 컴포넌트와 패스워드를 해싱 및 비교하는 컴포넌트를 모두 구현했다. 이번에는 의존 함수를 구현해서 이벤트 라우트에 주입해보자. 이 함수는 활성 세션에 존재하는 사용자 정보를 추출하는 단일 창구 역할을 한다.

auth/authenticate.py에 다음과 같이 코드를 작성하자.