학습을 마치면 해시hash를 사용해 패스워드를 보호하고 FastAPI 애플리케이션에 인증 계층을 추가할 수 있다. 허가되지 않은 사용자로부터 라우트를 보호하는 방법도 알 수 있다.

7.1 FastAPI의 인증 방식

FastAPI는 다양한 인증 방식을 지원한다. 그 중에서도 일반적인 인증 방법인 기본 HTTP 인증, 쿠키, bearer 토큰 인증에 관해 먼저 간단히 알아보자.

기본 HTTP 인증

사용자 인증 정보(일반적으로 사용자명과 패스워드를 사용한다)를 `Authorization` HTTP 헤더를 사용해 전송하는 방식이다. `Basic`값을 포함하는 `WWW-Authenticate` 헤더와 인증 요청을 처리한 리소스를 나타내는 영역realm 매개변수가 반환된다.

쿠키

데이터를 클라이언트 측(웹 브라우저 등)에 저장할 때 사용된다. FastAPI 애플리케이션도 쿠키를 사용해서 사용자 정보를 저장할 수 있으며 서버는 이 정보를 추출해 인증 처리에 사용한다.

bearer 토큰 인증

bearer 토큰이라는 보안 토큰을 사용해 인증하는 방식이다. 이 토큰은 `Bearer` 키워드와 함께 요청의 `Authorization` 헤더에 포함돼 전송된다. 가장 많이 사용되는 토큰은 JWT이며 사용자 ID와 토큰 만료 기간으로 구성된 딕셔너리 형식이 일반적이다.

이 방법들은 모두 장단점이 있으며 사용되는 곳도 다르다. 여기서는 bearer 토큰 인증을 사용한다. 인증에 사용되는 메서드는 런타임 시 호출되는 의존 라이브러리로 FastAPI 애플리케이션에 주입된다. 즉, 정의한 인증 메서드는 실제로 사용되기 전까지 휴면 상태에 있는 것이다. 이것을 의존성 주입이라고 한다.