

헤더 매개변수

헤더 매개변수를 `Query`, `Path` 그리고 `Cookie` 매개변수들과 같은 방식으로 정의할 수 있습니다.

Header 임포트

먼저 `Header` 를 임포트합니다:

```
{!../../../../../docs_src/header_params/tutorial001.py!}
```

Header 매개변수 선언

`Path`, `Query` 그리고 `Cookie` 를 사용한 동일한 구조를 이용하여 헤더 매개변수를 선언합니다.

첫 번째 값은 기본값이며, 추가 검증이나 어노테이션 매개변수 모두 전달할 수 있습니다:

```
{!../../../../../docs_src/header_params/tutorial001.py!}
```

!!! note "기술 세부사항" `Header` 는 `Path`, `Query` 및 `Cookie` 의 "자매"클래스입니다. 이 역시 동일한 공통 `Param` 클래스를 상속합니다.

``Query``, ``Path``, ``Header`` 그리고 다른 것들을 ``fastapi``에서 임포트 할 때, 이들은 실제로 특별한 클래스를 반환하는 함수임을 기억하세요.

!!! info "정보" 헤더를 선언하기 위해서 `Header` 를 사용해야 합니다. 그렇지 않으면 해당 매개변수를 쿼리 매개변수로 해석하기 때문입니다.

자동 변환

`Header` 는 `Path`, `Query` 그리고 `Cookie` 가 제공하는 것 외에 기능이 조금 더 있습니다.

대부분의 표준 헤더는 "마이너스 기호" (-)라고도 하는 "하이픈" 문자로 구분됩니다.

그러나 파이썬에서 `user-agent` 와 같은 형태의 변수는 유효하지 않습니다.

따라서 `Header` 는 기본적으로 매개변수 이름을 언더스코어(`_`)에서 하이픈(-)으로 변환하여 헤더를 추출하고 기록합니다.

또한 HTTP 헤더는 대소문자를 구분하지 않으므로 "snake_case"로 알려진 표준 파이썬 스타일로 선언할 수 있습니다.

따라서, `User_Agent` 등과 같이 첫 문자를 대문자화할 필요없이 파이썬 코드에서처럼 `user_agent` 로 사용합니다.

만약 언더스코어를 하이픈으로 자동 변환을 비활성화해야 할 어떤 이유가 있다면, `Header` 의 `convert_underscores` 매개변수를 `False` 로 설정하십시오:

```
{!../../../../../docs_src/header_params/tutorial002.py!}
```

!!! warning "경고" `convert_underscore` 를 `False` 로 설정하기 전에, 어떤 HTTP 프로시들과 서버들은 언더스코어가 포함된 헤더 사용을 허락하지 않는다는 것을 명심하십시오.

중복 헤더

중복 헤더들을 수신할 수 있습니다. 즉, 다중값을 갖는 동일한 헤더를 뜻합니다.

타입 정의에서 리스트를 사용하여 이러한 케이스를 정의할 수 있습니다.

중복 헤더의 모든 값을 파이썬 `list` 로 수신합니다.

예를 들어, 두 번 이상 나타날 수 있는 `X-Token` 헤더를 선언하려면, 다음과 같이 작성합니다:

```
{!../../../docs_src/header_params/tutorial003.py!}
```

다음과 같은 두 개의 HTTP 헤더를 전송하여 해당 경로와 통신할 경우:

```
X-Token: foo
X-Token: bar
```

응답은 다음과 같습니다:

```
{
  "X-Token values": [
    "bar",
    "foo"
  ]
}
```

요약

`Header` 는 `Query` , `Path` , `Cookie` 와 동일한 패턴을 사용하여 선언합니다.

변수의 언더스코어를 걱정하지 마십시오, **FastAPI**가 변수를 변환할 것입니다.