

# 쿼리 매개변수

경로 매개변수의 일부가 아닌 다른 함수 매개변수를 선언할 때, "쿼리" 매개변수로 자동 해석합니다.

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

쿼리는 URL에서 `?` 후에 나오고 `&` 으로 구분되는 키-값 쌍의 집합입니다.

예를 들어, URL에서:

```
http://127.0.0.1:8000/items/?skip=0&limit=10
```

...쿼리 매개변수는:

- `skip` : 값 `0` 을 가집니다.
- `limit` : 값 `10` 을 가집니다.

URL의 일부이므로 "자연스럽게" 문자열입니다.

하지만 파이썬 타입과 함께 선언할 경우(위 예에서 `int` ), 해당 타입으로 변환되고 이에 대해 검증합니다.

경로 매개변수에 적용된 동일한 프로세스가 쿼리 매개변수에도 적용됩니다:

- (당연히) 편집기 지원
- 데이터 "파싱"
- 데이터 검증
- 자동 문서화

## 기본값

쿼리 매개변수는 경로에서 고정된 부분이 아니기 때문에 선택적일 수 있고 기본값을 가질 수 있습니다.

위 예에서 `skip=0` 과 `limit=10` 은 기본값을 갖고 있습니다.

그러므로 URL로 이동하면:

```
http://127.0.0.1:8000/items/
```

아래로 이동한 것과 같습니다:

```
http://127.0.0.1:8000/items/?skip=0&limit=10
```

하지만 가령 아래로 이동한 경우:

```
http://127.0.0.1:8000/items/?skip=20
```

함수의 매개변수 값은 아래가 됩니다:

- `skip=20` : URL에서 지정했기 때문입니다
- `limit=10` : 기본값이기 때문입니다

## 선택적 매개변수

같은 방법으로 기본값을 `None` 으로 설정하여 선택적 매개변수를 선언할 수 있습니다:

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

이 경우 함수 매개변수 `q` 는 선택적이며 기본값으로 `None` 값이 됩니다.

!!! check "확인" **FastAPI**는 `item_id` 가 경로 매개변수이고 `q` 는 경로 매개변수가 아닌 쿼리 매개변수라는 것을 알 정도로 충분히 똑똑합니다.

!!! note "참고" FastAPI는 `q` 가 `= None` 이므로 선택적이라는 것을 인지합니다.

``Optional[str]``에 있는 ``Optional``은 FastAPI(FastAPI는 ``str`` 부분만 사용합니다)가 사용하는게 아니지만, ``Optional[str]``은 편집기에게 코드에서 오류를 찾아낼 수 있게 도와줍니다.

## 쿼리 매개변수 형변환

`bool` 형으로 선언할 수도 있고, 아래처럼 변환됩니다:

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

이 경우, 아래로 이동하면:

```
http://127.0.0.1:8000/items/foo?short=1
```

또는

```
http://127.0.0.1:8000/items/foo?short=True
```

또는

```
http://127.0.0.1:8000/items/foo?short=true
```

또는

```
http://127.0.0.1:8000/items/foo?short=on
```

또는

```
http://127.0.0.1:8000/items/foo?short=yes
```

또는 다른 어떤 변형(대문자, 첫글자만 대문자 등)이더라도 함수는 매개변수 `bool` 형을 가진 `short` 의 값이 `True` 임을 압니다. 그렇지 않은 경우 `False` 입니다.

## 여러 경로/쿼리 매개변수

여러 경로 매개변수와 쿼리 매개변수를 동시에 선언할 수 있으며 **FastAPI**는 어느 것이 무엇인지 알고 있습니다.

그리고 특정 순서로 선언할 필요가 없습니다.

매개변수들은 이름으로 감지됩니다:

```
{!../../../../../docs_src/query_params/tutorial004.py!}
```

## 필수 쿼리 매개변수

경로가 아닌 매개변수에 대한 기본값을 선언할 때(지금은 쿼리 매개변수만 보았습니다), 해당 매개변수는 필수적(Required)이지 않았습니다.

특정값을 추가하지 않고 선택적으로 만들기 위해선 기본값을 `None` 으로 설정하면 됩니다.

그러나 쿼리 매개변수를 필수로 만들려면 기본값을 선언할 수 없습니다:

```
{!../../../docs_src/query_params/tutorial005.py!}
```

여기 쿼리 매개변수 `needy` 는 `str` 형인 필수 쿼리 매개변수입니다.

브라우저에서 URL을 아래처럼 연다면:

```
http://127.0.0.1:8000/items/foo-item
```

...필수 매개변수 `needy` 를 넣지 않았기 때문에 아래와 같은 오류를 보게 됩니다:

```
{
  "detail": [
    {
      "loc": [
        "query",
        "needy"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

`needy` 는 필수 매개변수이므로 URL에 반드시 설정해줘야 합니다:

```
http://127.0.0.1:8000/items/foo-item?needy=sooooneedy
```

...아래처럼 작동합니다:

```
{
  "item_id": "foo-item",
  "needy": "sooooneedy"
}
```

그리고 물론, 일부 매개변수는 필수로, 다른 일부는 기본값을, 또 다른 일부는 선택적으로 선언할 수 있습니다:

```
{!../../../docs_src/query_params/tutorial006.py!}
```

이 경우 3가지 쿼리 매개변수가 있습니다:

- `needy` , 필수적인 `str` .

- `skip`, 기본값이 0 인 `int`.
- `limit`, 선택적인 `int`.

!!! tip "팁" [경로 매개변수](#){internal-link target=\_blank}와 마찬가지로 `Enum` 을 사용할 수 있습니다.