

クエリパラメータ

パスパラメータではない関数パラメータを宣言すると、それらは自動的に "クエリ" パラメータとして解釈されます。

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

クエリはURL内で `?` の後に続くキーとバリューの組で、`&` で区切られています。

例えば、以下の様なURL内で:

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

...クエリパラメータは:

- `skip`: 値は `0`
- `limit`: 値は `10`

これらはURLの一部なので、「自然に」文字列になります。

しかしPythonの型を宣言すると (上記の例では `int` として)、その型に変換されバリデーションが行われます。

パスパラメータに適用される処理と完全に同様な処理がクエリパラメータにも施されます:

- エディターサポート (明らかに)
- データ「解析」
- データバリデーション
- 自動ドキュメント生成

デフォルト

クエリパラメータはパスの固定部分ではないので、オプションとしたり、デフォルト値をもつことができます。

上述の例では、`skip=0` と `limit=10` というデフォルト値を持っています。

したがって、以下のURLにアクセスすることは:

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

以下のURLにアクセスすることと同等になります:

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

しかし、例えば、以下にアクセスすると:

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

関数内のパラメータの値は以下の様になります:

- `skip=20`: URL内にセットしたため
- `limit=10`: デフォルト値

オプションなパラメータ

同様に、デフォルト値を `None` とすることで、オプションなクエリパラメータを宣言できます:

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

この場合、関数パラメータ `q` はオプションとなり、デフォルトでは `None` になります。

!!! check "確認" パスパラメータ `item_id` はパスパラメータであり、`q` はそれとは違ってクエリパラメータであると判別できるほど **FastAPI** が賢いということにも注意してください。

!!! note "備考" FastAPIは、`= None` があるおかげで、`q` がオプションだとわかります。

``Optional[str]`` の ``Optional`` はFastAPIでは使用されていません (FastAPIは ``str`` の部分のみ使用します)。しかし、``Optional[str]`` はエディタがコードのエラーを見つけるのを助けてくれます。

クエリパラメータの型変換

`bool` 型も宣言できます。これは以下の様に変換されます:

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

この場合、以下にアクセスすると:

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

もしくは、

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

もしくは、

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

もしくは、

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

もしくは、

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

もしくは、他の大文字小文字のバリエーション (アップパーケース、最初の文字だけアップパーケース、など)で、関数は `short` パラメータを `True` な `bool` 値として扱います。それ以外は `False` になります。

複数のパスパラメータとクエリパラメータ

複数のパスパラメータとクエリパラメータを同時に宣言できます。**FastAPI**は互いを区別できます。

そして特定の順序で宣言しなくてもよいです。

名前で判別されます:

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

必須のクエリパラメータ

パスパラメータ以外のパラメータ (今のところ、クエリパラメータのみ説明しました) のデフォルト値を宣言した場合、そのパラメータは必須ではなくなります。

特定の値を与えずにただオプションにしたい場合はデフォルト値を `None` にして下さい。

しかしクエリパラメータを必須にしたい場合は、ただデフォルト値を宣言しなければよいです:

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

ここで、クエリパラメータ `needy` は `str` 型の必須のクエリパラメータです

以下のURLをブラウザで開くと:

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

...必須のパラメータ `needy` を加えなかったので、以下の様なエラーが表示されます:

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

`needy` は必須のパラメータなので、URLにセットする必要があります:

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

...これはうまくいくでしょう:

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

そして当然、あるパラメータを必須に、別のパラメータにデフォルト値を設定し、また別のパラメータをオプションにできます:

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

この場合、3つのクエリパラメータがあります。:

- `needy`、必須の `str` 。
- `skip`、デフォルト値を `0` とする `int` 。
- `limit`、オプションな `int` 。

!!! tip "豆知識"

[パスパラメータ] ([path-params.md#predefined-values](#)) (`.internal-link target=_blank`)と同様に
`Enum` を使用できます。