

查询参数

声明不属于路径参数的其他函数参数时，它们将被自动解释为"查询字符串"参数

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

查询字符串是键值对的集合，这些键值对位于 URL 的 `?` 之后，并以 `&` 符号分隔。

例如，在以下 url 中：

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

...查询参数为：

- `skip` : 对应的值为 `0`
- `limit` : 对应的值为 `10`

由于它们是 URL 的一部分，因此它们的"原始值"是字符串。

但是，当你为它们声明了 Python 类型（在上面的示例中为 `int`）时，它们将转换为该类型并针对该类型进行校验。

应用于路径参数的所有相同过程也适用于查询参数：

- （很明显的）编辑器支持
- 数据"解析"
- 数据校验
- 自动生成文档

默认值

由于查询参数不是路径的固定部分，因此它们可以是可选的，并且可以有默认值。

在上面的示例中，它们具有 `skip=0` 和 `limit=10` 的默认值。

因此，访问 URL：

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

将与访问以下地址相同：

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

但是，如果你访问的是：

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

函数中的参数值将会是：

- `skip=20` : 在 URL 中设定的值
- `limit=10` : 使用默认值

可选参数

通过同样的方式，你可以将它们的默认值设置为 `None` 来声明可选查询参数：

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

在这个例子中，函数参数 `q` 将是可选的，并且默认值为 `None`。

!!! check 还要注意的，**FastAPI** 足够聪明，能够分辨出参数 `item_id` 是路径参数而 `q` 不是，因此 `q` 是一个查询参数。

查询参数类型转换

你还可以声明 `bool` 类型，它们将被自动转换：

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

这个例子中，如果你访问：

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

或

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

或

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

或

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

或

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

或任何其他的变化形式（大写，首字母大写等等），你的函数接收的 `short` 参数都会是布尔值 `True`。对于值为 `False` 的情况也是一样的。

多个路径和查询参数

你可以同时声明多个路径参数和查询参数，**FastAPI** 能够识别它们。

而且你不需要以任何特定的顺序来声明。

它们将通过名称被检测到：

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

必需查询参数

当你为非路径参数声明了默认值时（目前而言，我们所知道的仅有查询参数），则该参数不是必需的。

如果你不想添加一个特定的值，而只是想使该参数成为可选的，则将默认值设置为 `None`。

但当你想让一个查询参数成为必需的，不声明任何默认值就可以：

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

这里的查询参数 `needy` 是类型为 `str` 的必需查询参数。

如果你在浏览器中打开一个像下面的 URL：

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

...因为没有添加必需的参数 `needy`，你将看到类似以下的错误：

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

由于 `needy` 是必需参数，因此你需要在 URL 中设置它的值：

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

...这样就正常了：

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

当然，你也可以定义一些参数为必需的，一些具有默认值，而某些则完全是可选的：

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

在这个例子中，有3个查询参数：

- `needy`，一个必需的 `str` 类型参数。
- `skip`，一个默认值为 `0` 的 `int` 类型参数。
- `limit`，一个可选的 `int` 类型参数。

!!! tip 你还可以像在 [路径参数](#){internal-link target=_blank} 中那样使用 `Enum`。