

8:38:30

Query Parameters :-

Ex :- www.example.com/search ? find_loc=Delhi
query parameter

[localhost:8000/posts ?limit=25]

We are naming our query parameter as limit.
To obtain this parameter data in our path operation, we set/make a same named parameter in the path operation func. definition.

```
@router.get("/", response_model=...)  
def get_posts(db: Session..., ..., limit: int = 10):  
    -- some named variable  
    -- to accept data from query  
    -- parameters.  
    | setting a  
    | default value
```

• Thus, in FastAPI, query parameters are passed as function arguments to path operations.

• The limit can be applied in sqlalchemy as follow :-

```
[--]
posts = db.query(models.Post).limit(limit).all()
```

we have limit() function, to apply limit to the no. of records fetched.

• Suppose, we want to provide user a way, that he/she can see specified no. of posts, by skipping some posts from the top.

For that, we need another query parameter named skip :-

```
[localhost:8000/posts?limit=5&skip=3]
```

```
[--]
def get_posts(--, limit: int=10, skip: int=0):
    post = db.query(models.Post).limit(limit).
            ... offset(skip).all()
    -- -- --
```

by default, no posts
should be skipped

for excluding some results from top, we use the offset() method of the query object.

• This skipping feature is useful, for implementing pagination, i.e., when going to second page, certain

posts, say 20 post will be skipped & shown, on 3rd page, 40 posts from top will be skipped, and next 20 will be shown, and so on...

⇒ Search Functionality Using

Query Parameters :-

⇒ User must be able to search posts, based on keywords, in the title, or in the content.

⇒ Here, we want the user to search posts based on keywords in the title of the posts:-

```
from typing import Optional
```

```
def get_post(..., search: Optional[str] = ""):
```

```
    posts = db.query(models.Post).filter(models...
```

```
... Post.title.contains(search).limit(limit).offset(skip).all()
```

contains() method searches the caller object string, if it contains the passed parameter as substring. If substring is present, then returns the record.

localhost:8000/posts/search=abcd $\% 20$ for 'space'

localhost:8000/posts/search=hello $\% 20$ word