

Ítalo Rossi

italo@evolux.net.br

CTO Evolux

<https://br.linkedin.com/in/italorossi>



Origem, motivação, história

- Rápido
- Produtividade
- Uso de novos recursos da linguagem
- Intuitivo/Fácil
- Pequeno


FastAPI = Boas escolhas

Tirando proveito das boas e más experiências dos frameworks já existentes.

FastAPI básico

Path params - format strings

```
1  from fastapi import FastAPI
2
3  app = FastAPI()
4
5
6  @app.get("/conference/{name}")
7  async def conference(name):
8      return {"message": f"Welcome to conference {name}"}
9
10
```



Types Annotation

```
1 from fastapi import FastAPI
2
3 app = FastAPI()
4
5
6 @app.get("/conference/{name}")
7 async def conference(name):
8     return {"message": f"Welcome to conference {name}"}
9
10
```

for

not

not expr

par

(expr)

Ctrl+Down and Ctrl+Up will move caret down and up in the editor [Next Tip](#)

⋮

Types Annotation

```
1 from fastapi import FastAPI
2
3 app = FastAPI()
4
5
6 @app.get("/conference/{name}")
7 async def conference(name: str):
8     return {"message": f"Welcome to conference {name}!"}
```

for


find(self, __sub, __start, __end)	str
join(self, __iterable)	str
count(self, x, __start, __end)	str
index(self, __sub, __start, __end)	str
capitalize(self)	str
casefold(self)	str
center(self, __width, __fillchar)	str
encode(self, encoding, errors)	str
endswith(self, __suffix, __start, __end)	str
expandtabs(self, tabsize)	str
format(self, *args, **kwargs)	str

Press Ctrl+ to choose the selected (or first) suggestion and insert a dot afterwards. Next Tip

Data conversion

```
@app.get("/conference/year/{year}")  
async def conference(year: int):  
    return {"year": year}
```

```
@app.get("/conference/year/{year}")  
async def conference(year: str):  
    return {"year": year}
```




GET http://127.0.0.1:8000/conference/year/2023

Show Request

HTTP/1.1 200 OK

> (Headers) ...content-type: application/json...

```
{  
  "year": 2023  
}
```



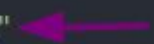
GET http://127.0.0.1:8000/conference/year/2023

Show Request

HTTP/1.1 200 OK

> (Headers) ...content-type: application/json...

```
{  
  "year": "2023"  
}
```



Data validation

```
@app.get("/conference/year/{year}")
async def conference(year: int):
    return {"year": year}
```

GET http://127.0.0.1:8000/conference/year/doismil

Show Request

HTTP/1.1 422 Unprocessable Entity

> (Headers) ...content-type: application/json...

```
{
  "detail": [
    {
      "type": "int_parsing",
      "loc": [
        "path",
        "year"
      ],
      "msg": "Input should be a valid integer, unable to parse string as an integer",
      "input": "doismil",
      "url": "https://errors.pydantic.dev/2.4/v/int_parsing"
    }
  ]
}
```



Query params

```
@app.get("/conferences/talks")
async def get_talks(limit: int):
    """List talks from conferences"""
    talks = ["talk1", "talk2", "talk3"]
    return {"talks": talks[:limit]}
```

```
GET http://127.0.0.1:8000/conferences/talks?limit=2
```

Show Request


```
HTTP/1.1 200 OK
```

```
> (Headers) ...content-type: application/json...
```

```
{
  "talks": [
    "talk1",
    "talk2"
  ]
}
```

Query params - defaults

```
@app.get("/conferences/talks")
async def get_talks(limit: int = 50):
    """List talks from conferences"""
    talks = ["talk1", "talk2", "talk3"]
    return {"talks": talks[:limit]}
```



Query params - defaults

```
GET http://127.0.0.1:8000/conferences/talks
```

Show Request

```
HTTP/1.1 200 OK
```

```
> (Headers) ...content-type: application/json...
```

```
{  
  "talks": [  
    "talk1",  
    "talk2",  
    "talk3"  
  ]  
}
```

Mypy

```
@app.get("/conferences/talks")
async def get_talks(limit: int | None = None):
    """List talks from conferences"""
    talks = ["talk1", "talk2", "talk3"]
    if limit > 50:
        limit = 50
    return {"talks": talks}
```

Mypy: Left operand is of type "int | None"

Mypy: Unsupported operand types for < ("int" and "None") [operator]

Suppress Mypy error Alt+Shift+Enter More actions... Alt+Enter

Validação Automática

Query params - validation

```
@app.get("/conferences/{name}")
async def get_conference(name: Annotated[str, Path(max_length=10)],
                          order_by: Annotated[str, Query(max_length=10)] = "name"):
    """Get conference by name"""
    return {"conference": name, "order_by": order_by}
```

Query params - validation

```
GET http://127.0.0.1:8099/conferences/gornvvvvvvvvvvvvvvvvvvvvv
```

[Show Request](#)

HTTP/1.1 422 Unprocessable Entity

Request Body - Pydantic

POST http://127.0.0.1:8000/conferences/talks

Accept: application/json

Content-Type: application/json

`{"title": "FastAPI", "speaker": "italo", "duration": "3600"}`

```
1 from fastapi import FastAPI
2 from pydantic import BaseModel
3
4 app = FastAPI()
5
6 talks = []
7
8
9 1 usage
10 class Talk(BaseModel):
11     """Talk model"""
12     title: str
13     speaker: str
14     duration: int
15
16 @app.post("/conferences/talks")
17 async def add_talk(talk: Talk):
18     """Add a talk to the list of talks"""
19     talks.append(talk)
20     return {"talk": talk}
```

Request Body

```
POST http://127.0.0.1:8000/conferences/talks

HTTP/1.1 200 OK
> (Headers) ...content-type: application/json...

{
  "talk": {
    "title": "FastAPI",
    "speaker": "Italo",
    "duration": 3600
  }
}
```

Request Body - Error

```
{"title": "FastAPI", "speaker": null, "duration": 3600}
```

```
POST http://127.0.0.1:8000/conferences/talks
```

```
Show Request
```

```
HTTP/1.1 422 Unprocessable Entity
```

```
> (Headers) content-type: application/json
```

```
{
  "detail": {
    {
      "type": "string_type",
      "loc": [
        "body",
        "speaker"
      ],
      "msg": "Input should be a valid string",
      "input": null,
      "url": "https://errors.pydantic.dev/2.4/v/string_type"
    }
  ]
}
```



OpenAPI



default

POST

/conferences/talks Add Task

GET

/conferences/talks Get Talks

GET

/conferences/{name} Get Conference

Schemas

HTTPValidationError > Expected id object

Talk > Expected id object

ValidationError > Expected id object

Search...

Add Talk

Get Talks

Get Conferences

FastAPI (0.1.0)

Download OpenAPI specificationDownload

Add Talk

Add a talk to the list of talks.

Upload OpenAPI definition

application/json

File to upload

image (File)

Speaker

image (Speaker)

Short talk

image (Short talk)

Responses

5/200 Responses by Response

3/400 Validation Error

Get Talks

Get talks from conferences

Query to fetch talks

→ Talks

Integer (0-100)

Default: 30

Responses

POST

transfer request to db

Request samples

Request

Content type

application/json

Copy

```
{
  "talker": "John Doe",
  "speaker": "John Doe",
  "short talk": 1
}
```

Response samples

200200

Content type

application/json

Copy

```
{}
```

POST

transfer request to db

Response samples

200200

Content type

application/json

Copy

```
{}
```

Headers - Cookies

```
@app.get("/conferences/talks")
async def get_talks(user_agent: Annotated[str, Header()],
                    my_cookie: Annotated[str | None, Cookie()] = None,
                    limit: int = 50):
    """List talks from conferences"""
    if limit > 50:
        limit = 50
    return {"talks": talks[:limit], "User-Agent": user_agent, "my_cookie": my_cookie}
```


Headers - Cookies - Response

```
class TalksResponse(BaseModel):
    """Talks Response model"""
    talks: list[Talk]
    UserAgent: str
    my_cookie: str | None

@app.get("/conferences/talks")
async def get_talks(user_agent: Annotated[str, Header()],
                    my_cookie: Annotated[str | None, Cookie()] = None,
                    limit: int = 50) -> TalksResponse:
    """List talks from conferences"""
    if limit > 50:
        limit = 50

    return TalksResponse(talks=talks[:limit], UserAgent=user_agent, my_cookie=my_cookie)
```

Headers - Cookies - Response

```
{
  "talks": [
    {
      "title": "Pydantic",
      "speaker": "Sebastian",
      "duration": 3600
    },
    {
      "title": "FastAPI",
      "speaker": "italo",
      "duration": 3600
    }
  ],
  "UserAgent": "Apache-HttpClient/4.5.14 (Java/17.0.8)",
  "my_cookie": "1234567890"
}
```

Python Type Annotations

Dependency

```
@app.get("/talks/paginated")
async def paginated_talks(page: int = 1, size: int = 10):
    """Paginated talks"""
    return {"page": page, "size": size}
```


Dependency

```
async def pagination(page: int = 1, size: int = 10):  
    """Pagination dependency"""  
    return {"page": page, "size": size}  
  
@app.get("/talks/paginated")  
async def paginated_talks(pagination: dict = Depends(pagination)):  
    """Paginated talks"""  
    return {"page": pagination.get("page"), "size": pagination.get("size")}
```



Dependency - Nested

```
async def get_token_optional(authorization: Annotated[str | None, Header(required=False)] = None):  
    return authorization  
  
1 usage  
  
async def pagination(page: int = 1, size: int = 10, token: str = Depends(get_token_optional)):  
    """Pagination dependency"""  
    return {"page": page, "size": size, "token": token}  
  
@app.get("/talks/paginated")  
async def paginated_talks(pagination: dict = Depends(pagination)):  
    """Paginated talks"""  
    return {"page": pagination.get("page"),  
            "size": pagination.get("size"),  
            "token": pagination.get("token")}
```



FastAPI - Mais...

Mais...

- SQLAlchemy
- Middleware
- Async e sync
- Lifespan
- Authentication
- BackgroundTasks
- Testing
- Websockets
- WSGIMiddleware

Obrigado!