



FastAPI

MSE ITSec



Introduction to FastAPI

- Python 3.6+
- Released in December 2018
- Facilitates the development of REST-API's
- Easy to learn
- Some features:
 - Swagger API documentation
 - Validation
 - Security



Introduction to SQLAlchemy

- Python 2.7 | 3.6+
- Released in February 2006
- SQL-Toolkit:
 - Communication with databases
 - Object relational mapper (ORM)



Application Overview

- Small REST API to manage users and related items
- Different API routes for different purposes

FastAPI 0.1.0 OAS3
/openapi.json

default ^

GET	/users/	Read Users	▼
POST	/users/	Create User	▼
GET	/users/{user_id}	Read User By Id	▼
GET	/users/safe1/{name}	Read User By Name Safe1	▼
GET	/users/safe2/{name}	Read User By Name Safe2	▼
GET	/users/unsafe/{name}	Read User By Name Unsafe	▼
POST	/users/{user_id}/items/	Create Item For User	▼
GET	/items/	Read Items	▼

Possible Attacks

- Bad coded logic can lead to possible SQL injection

```
# Route
```

```
@app.get("/users/unsafe/{name}")
```

```
def read_user_by_name_unsafe(name: str, db: Session = Depends(get_db)):
```

```
    db_user = crud.get_user_by_name_unsafe(db=db, name=name)
```

```
    if db_user is None:
```

```
        raise HTTPException(status_code=404, detail="User not found")
```

```
    return db_user
```

```
# GET INFO FROM DB
```

```
def get_user_by_name_unsafe(db: Session, name: str):
```

```
    return db.execute(f"SELECT * FROM users WHERE name = '{name}'").all()
```

Mitigate Attacks (Safe1)

- Write Clean Code

Route

```
@app.get("/users/safe1/{name}", response_model=schemas.User)
def read_user_by_name_safe1(name: str, db: Session = Depends(get_db)):
    db_user = crud.get_user_by_name_safe1(db=db, name=name)
    if db_user is None:
        raise HTTPException(status_code=404, detail="User not found")
    return db_user
```

GET INFO FROM DB

```
def get_user_by_name_safe1(db: Session, name: str):
    return db.query(models.User).filter(models.User.name == name).first()
```

Mitigate Attacks (Safe2)

- Write Clean Code

Route

```
@app.get("/users/safe2/{name}", response_model=schemas.User)
def read_user_by_name_safe2(name: str, db: Session = Depends(get_db)):
    db_user = crud.get_user_by_name_safe2(db=db, name=name)
    if db_user is None:
        raise HTTPException(status_code=404, detail="User not found")
    return db_user
```

GET INFO FROM DB

```
def get_user_by_name_safe2(db: Session, name: str):
    sql_statement: str = "SELECT * FROM users WHERE name = :name "
    return db.execute(sql_statement, {"name": name}).first()
```


DEMO





Sebastián Ramírez


@tiangolo

...

I saw a job post the other day. 

It required 4+ years of experience in FastAPI. 

I couldn't apply as I only have 1.5+ years of experience since I created that thing. 

Maybe it's time to re-evaluate that "years of experience = skill level". 

3:40 nachm. · 11. Juli 2020 · Twitter Web App