01

Uppgifter & Eget Arbete

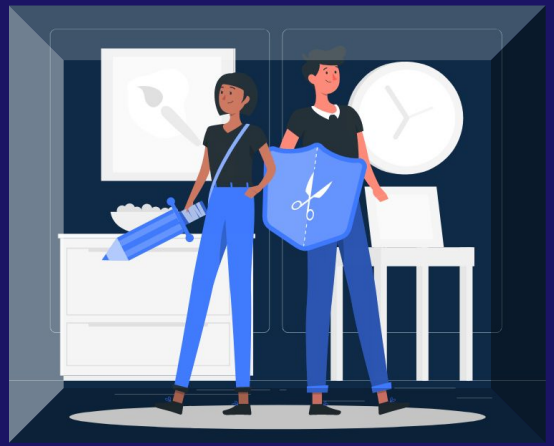
Uppgifter

Välkommen till första uppgiften!

Uppgifterna är till för att testa dina färdigheter och kunskaper för att både öva och repetera på det vi har arbetat med under föreläsningarna.

Dessa är **INTE** obligatoriska.

Men är ämnen ni kommer testas mot.



Code Analysis

```
// Question: Where does the connection OPEN and where does it CLOSE?
// What does: 'with' and 'as' mean in this context?

import psycopg
from psycopg_pool import ConnectionPool

pool = ConnectionPool("postgresql://postgres:password@localhost:5432/demo_5" )

def store_value(value: str):
    with pool.connection() as conn:

        conn.execute(
            "INSERT INTO demo_table (value) VALUES (%s) ",
            (value,)
        )
        conn.commit()
```

Code Analysis #2

```
// Question: What is 'product.model_dump()' and why is it necessary?

@app.post("/products", response_model=ProductSchema,
status_code=status.HTTP_201_CREATED)
def products(product: ProductSchema) -> ProductSchema:
    with pool.connection() as conn:
        insert_product(conn, product.model_dump())
        conn.commit()

    return product
```

Code Analysis #3

```
// Question: Just by analyzing the code… what do you think this does?

@app.post("/products/bulk")
def products_bulk(products: list[ProductSchema]):
    with pool.connection() as conn:
        with conn.cursor() as cur:
            cur.executemany(
                "INSERT INTO products_raw (product) VALUES (%s)",
                [(Json(product.model_dump()),) for product in products]
            )
        conn.commit()
    return {"inserted": len(products)}
```

```
1   // –Uppgift #1– //
2
3   /* INSTRUCTIONS
4
5       Utgå från dagens lektion
6       Ändra nu ‘ProductSchema’ så att den innehåller
7       ett extra värde:
8         • tags: Union[list[str], None]
9
10      Prova kör koden - fungerar det?
11  */
12
13  // HINT & Examples
14  hint("Hint: Ja")
15
16
17
18
19
20
21
22
23
```

Uppgift #1

Kom igång enkelt med uppgift #1

```
1   // -Uppgift #2- //
2
3   /* INSTRUCTIONS
4
5       Inom ditt schema lägg till ett objekt
6       Exempelvis:
7
8       class DimensionsSchema(BaseModel):
9           width_cm: float
10          height_cm: float
11          depth_cm: float
12
13      class ProductSchema(BaseModel):
14      // old values from before remain the same
15      dimensions: Union[DimensionsSchema, None]
16
17
18      Påverkar detta koden när du kör?
19  */
20
21  hint("hint: nej")
22  hint("Nästa uppgift visar hur du lägger till
23  objektet in i databasen via postman")
```

Uppgift #2

```
{
        "product_id": "SKU-123",
        "name": "Wireless Mouse",
        "price": 299.0,
        "currency": "SEK",
        "category": null,
        "brand": null,
        "tags": null,
        "dimensions": {
                "width_cm": 6.2,
                "height_cm": 3.8,
                "depth_cm": 10.1
        }
}
```

```
                        // -Uppgift #3- //

/* INSTRUCTIONS

    Använd följande kod


    @app.get("/products")
    def get_products():
        with pool.connection() as conn:
            with conn.cursor() as cur:
                cur.execute("SELECT product FROM products_raw" )
                rows = cur.fetchall()

        # rows = [(product_dict,), (product_dict,), ...]
        return [row[0] for row in rows]

    Testkör koden - vad tror du att 'fetchall()' returnerar?
    */

// HINT & Examples
hint("Lista av dictionaries")
```

THANKS !

Do you have any questions?
kristoffer.johansson@sti.se