01

Uppgifter & Eget Arbete

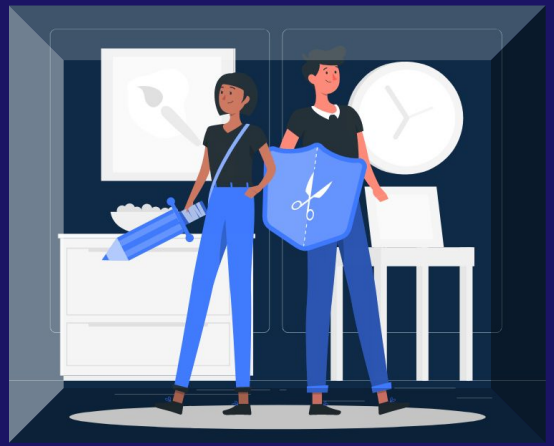
Uppgifter

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.



MINNS DU?

```
/*

    Förklara följande:
      • Mapping
      • Endpoint
      • HTTP-method
      • Query-parameter
      • Status Code

*/
```

Theory - Delete Mapping

```
/*
    På nästa sida presenteras kod som
    du ska analysera.

    Därefter får du ytterligare
    information om hur det presenterade
    problemet kan lösas.
*/
```

Theory - Delete Mapping

```python
@app.delete("/users", status_code=status.HTTP_200_OK)
def delete_user(username: str) -> dict[str, str]:
    for user in userList:
        if user.username == username:
            userList.remove(user)
            return {"message": "User deleted"}

    return {"message": "User not found"}


# Notera att nu kommer vi ALLTID att returnera 200 ok även om
# resultatet ej hittas…
```

Theory - Solution

```python
@app.delete("/users", status_code=status.HTTP_200_OK)
def delete_user(username: str) -> dict[str, str]:
    for user in userList:
        if user.username == username:
            userList.remove(user)
            return {"message": "User deleted"}

    raise HTTPException(
        status_code=status.HTTP_404_NOT_FOUND,
        detail="User not found",
    )

# Vi kan grena ut med HTTPException (glöm inte importera bara)
# 'raise' är likt 'throw' inom Java, C#, javascript/typescript, kotlin
```

```
1   // -Uppgift #1- //
2
3   /* INSTRUCTIONS
4
5       Skapa ett helt nytt projekt.
6       Installera rätt bibliotek:
7       https://fastapi.tiangolo.com/#create-it
8
9       Försök hitta hur en installerar!
10
11  */
12
13  // HINT & Examples
14  hint("Leta efter terminal kommandon")
15  hint("Installation inkluderar pydantic")
16
17
18
19
20
21
22
23
```

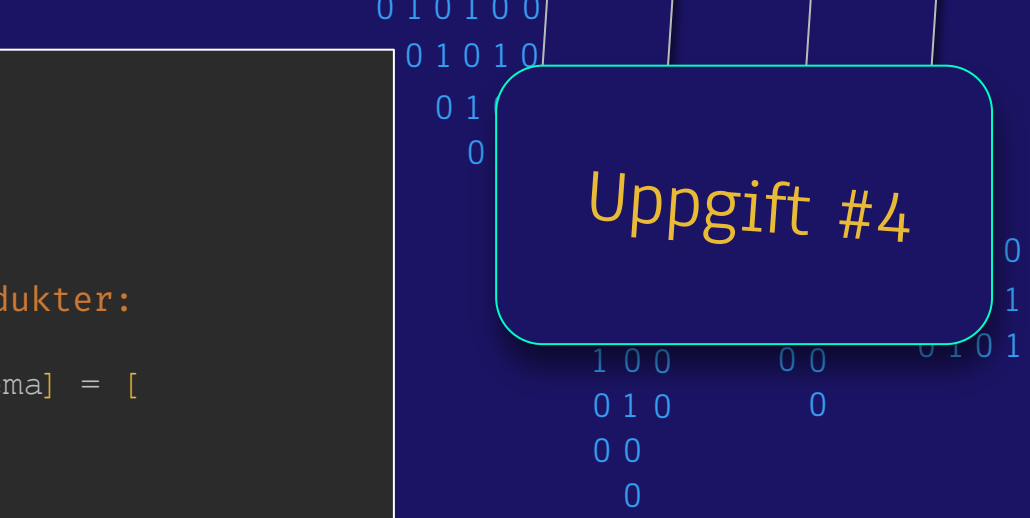
Kom igång enkelt med uppgift #1

```
1          // -Uppgift #2- //
2
3  /* INSTRUCTIONS
4
5      Skapa en 'app' variabel:
6
7      from fastapi import FastAPI
8      app = FastAPI(title="My First API")
9
10
11     Skapa nu en enkel 'Hello World'
12     GET-Mapping som använder sig av en
13     'Dictionary'
14
15 */
16
17 hint("@")
18 hint("app")
19 hint("get()")
20 hint("return {}")
21
22
23
```

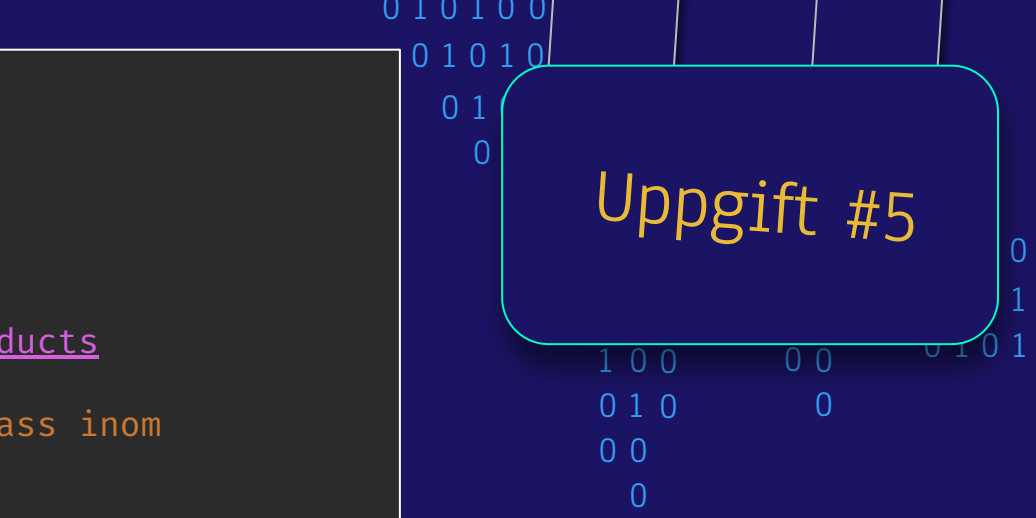
Uppgift #3

```
1   // -Uppgift #3- //
2
3   /* INSTRUCTIONS
4
5       Skapa ett nytt 'package'
6       Döp den till 'schema'
7
8       Skapa en ny .py fil: Product
9       class ProductSchema(BaseModel):
10          TBD: tbd
11
12      Inkludera
13        • id,
14        • title,
15        • price,
16        • description,
17        • category,
18        • image
19  */
20
21  // HINT & Examples
22  hint("Glöm inte data typer: str, int etc…")
23
```

```
1          // -Uppgift #4- //
2
3   /* INSTRUCTIONS
4
5        Inom main.py
6        Skapa en enkel array med produkter:
7
8        productList: list[ProductSchema] = [
9           ProductSchema(...),
10          ProductSchema(...),
11          ProductSchema(...),
12          ProductSchema(...),
13          ProductSchema(...),
14       ]
15
16       Hämta ut alla produkter inom en
17       'getProducts mapping'
18  */
19
20  // HINT & Examples
21  hint("Glöm inte att returnera listan inom 'def'
22  också för bättre struktur")
23
```



```
1   // -Uppgift #5- //
2
3   /* INSTRUCTIONS
4
5       Kolla på URL'n
6       Fakestore API
7       https://fakestoreapi.com/products
8
9       Skapa nu en till 'Schema' klass inom
10      product.py
11
12      Vi har redan definierat de 6 första värden…
13  */
14
15  // HINT & Examples
16  hint("Ignorera index elementen. Detta är enbart en
17  lista med objekt.
18  Fokusera enbart på objekt")
19  hint("rating")
20
21
22
23
```



```
1   // -Uppgift #6- //
2      Tough nut
3   /* INSTRUCTIONS
4
5        Konsumera API:et
6        https://fakestoreapi.com/products
7
8        Använd datan för att visa upp alla produkter.
9        (Notera att allt är en enda stor array)
10
11       FACIT finns på nästkommande sidor!
12  */
13
14  // HINT & Examples
15  hint("List + For loop")
16
17  hint("Denna uppgift är svår, ta gärna hjälp av
18  google eller andra externa resurser")
19
20
21
22
23
```

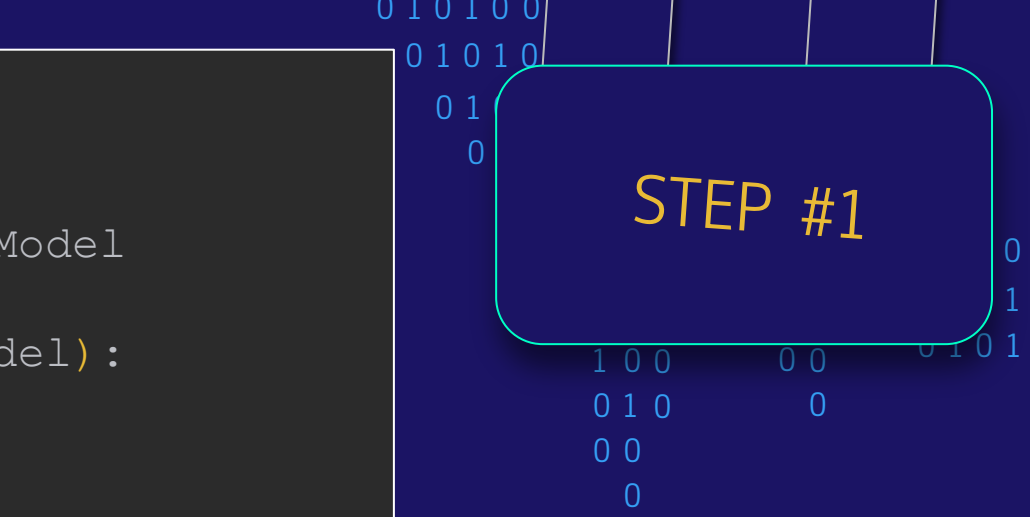
```python
        # -FACIT- #
           Tough nut


from pydantic import BaseModel

class RatingSchema(BaseModel):
    rate: float
    count: int

class ProductSchema(BaseModel):
    id: int
    title: str
    price: float
    description: str
    category: str
    image: str
    rating: RatingSchema

```



```
// -FACIT- //
  Tough nut


@app.get("/products", response_model=list[ProductSchema])
def get_products() -> list[ProductSchema]:
    result =
requests.get("https://fakestoreapi.com/products" )
    response_json = result.json()

    products: list[ProductSchema] = []

    for item in response_json:
        product = ProductSchema(**item)
        products.append(product)

    return list(products)
```