***API***

# API

Wat zijn API’s 0 – 2u24

Dit zijn programma’s om informatie in en uit systemen, SQL te halen, te bewerken.

Basis bewerkingen : CRUD

* create posts
* read posts
  + get all
  + get (id) specifiek item
* update posts
  + put all
  + patch specifiek item
* delete posts (id) specifiek item

# Python

Installeer Python

Wincmd py -3 –version check Python version

Models.py model creëren voor de database

* Tabelnaam
* Kolommen specifiëren
* Etc…

Run py program:

* Py -3 run example.py
* Py\_DB\_URS
  + Localhost: 5432

# Visual Studio Code

Setup 35’47”

Installeer VScode

Open new folder op pc => fastapi

Alle applicaties draaien onder de global environment.

Create virtual environment:

* Maak voor elke applicatie een virtueel environment. (venv) 26’30”
* Installeer alle applicaties onder dit venv.
* Maak alle programma’s onder dit venv
* Mag altijd dezelfde naam zijn.

VSC Terminal new windows cmd deze draait onder venv

* Onze terminal geeft nu ook venv aan.

python.exe

Win cmd: -m pip install -U autopep 8

* Automatisch aanvullen lijnen en code tijdens programmatie.

Win cmd : “py -3 -m venv <naam>”

* py -3 -m venv venv

Activeer cmd line onder venv.

VSC View command palette Python select interpreter Enter interpreter path

* \venv\Scripts\python.exe

Venv cmd: venv\scripts\activate.bat

Uvicorn cmd:

Pip freeze laat geïnstalleerde pakketten zien

New folder : App package

New file in App \_\_init\_\_.py maakt van App een package, blijft leeg

* main verplaatsen in App
* main starten cmd: uvicorn App.main:app –reload
  + App = package
  + app = functie app in main

comment out : ctrl K + C # blablabla

# Fast api

## installeren

35:09

Win cmd: pip install fastapi[all]

Win cmd: pip freeze laat al de geïnstalleerde pakketten zien

Win cmd: uvicorn main: app start server

Win cmd: uvicorn main: app –reload start server auto

Optioneel :

Win cmd: uvicorn App.main: app start server

Win cmd: uvicorn App.main: app –reload start server auto

Uvicorn running on httm:\\127.0.0.1:8000 (eigen pc)

cmd: CTRL C stop server

cmd: CTRL S save program 39:23

Gegevens beheren in Python 3u39

Cmd: INSERT INTO <table> (name, ……), values( ,,,,) RETURNING \*

* returning \* laat alle rijen zien die toegevoegd zijn
* keuze \* voor alles of specifieke rij ingeven.

Cmd: DELETE FROM <table> WHERE RETURNING \*

* bij delete laat returning de rij zien die gewist is

## Fastapi ⬄ SQL

Fastapi communiceert met de database : SQL

* FASTAPI < SQL commando’s in Python > SQL

Dit kan rechtstreeks maar in realiteit zit hiertussen een ORM.

ORM object relational mapper.

FASTAPI < python commando’s > ORM psycopg/SQL SQL

Fastapi + documentatie 127.0.0.1:8000/doc

Doc versie 1

Redoc versie 2

Doc Swagger UI

Hash password 5u55

Venv cmd : Install algoritme

Passlib werkt met verschillende algoritmen voor encryptie.

Pip install “passlib[bcript]”

Path operations uitsplitsen 6u18

* CRUD operations in functie
* User operations in functie

Router

* From . import punt betekent zelfde dir
* From .. import punt punt betekent dir up

@ app vervangen door router

# Postman

Download en installeer postman.

Programma om berichten te verzenden.

## Create environment

7u42

Create environment, vb URL variabele die in alle posts kan gebruikt worden. {{URL}}

Login “set an environment variable” voor de login.

Deze variabele bevat dan de token data. En dus kan je deze variabele dan gebruiken om de token te vervangen in alle posts.

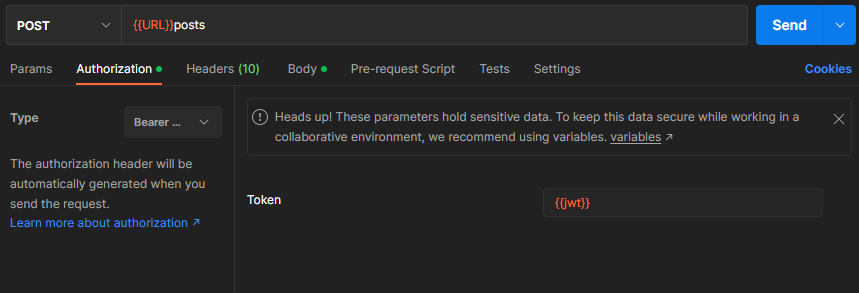
* pm.environment.set("jwt", pm.response.json().access\_token);

Token

Copy token => postman

Create post

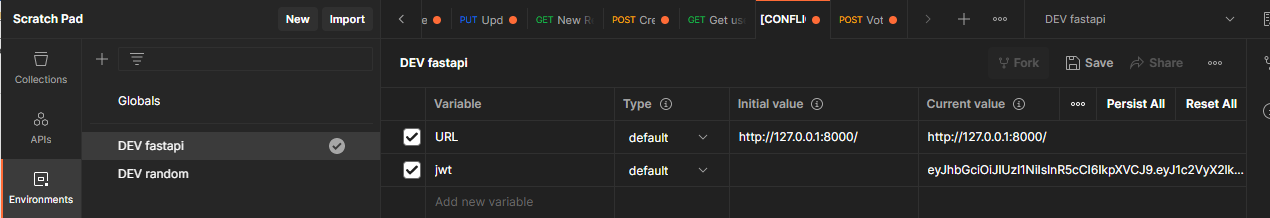
* autherisation type : bearer
* Geef de variabele in uit environment.



## Create environments

9u04

Copy token in environment.



# SQLAlchemy

Venv cmd pip install sqlalchemy

Install drivers indien nodig (mogelijks bestaan reeds)

* Database.py verbinding tussen Python en SQL 4u41
  + Via sqlalchemy

Setup foreign key 8u07

Relationship return data 8u37

Query parameters by request 8u40

* URL \_\_\_\_\_ ? \_\_\_\_ rechts van ? zijn parameters  
    
  https://www.youtube.com/watch?v=0sOvCWFmrtA

# ORM sql alchemy Python ORM

# Postgresql

Relational database.

Install postgresql :

* DB1 – app1 DB2 – app2
* Port 5432

Info data types online

Postgres vs Py

* Int, dec, precision int, float
* Varchar, text string
* Boolean Boolean
* Array list

cmd SELECT column of \* FROM db\_naam WHERE int, str =, !=, <, > etc…

cmd SELECT column of \* FROM db\_naam LIKE num, ‘str’, ‘%str’, ‘str%’, ‘%str%

cmd SELECT BY name DESC, name rangschikken, ASC default

cmd SELECT column of \* FROM db\_naam WHERE ORDER BY

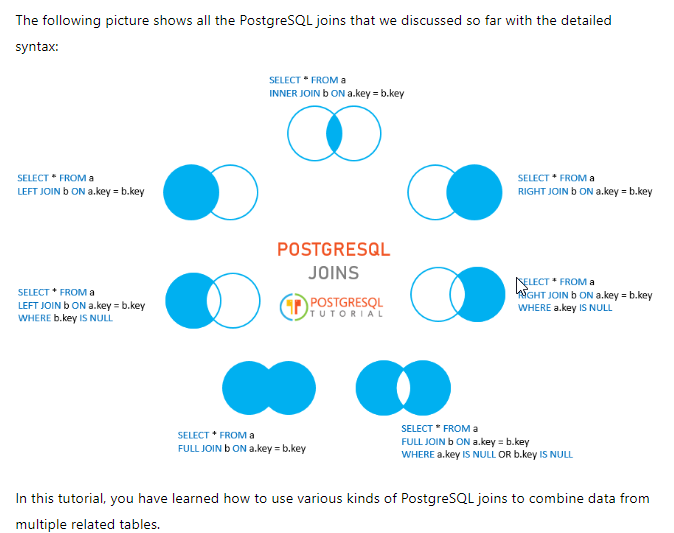
cmd SELECT column of \* FROM db\_naam WHERE LIMIT num eerste 5

laatste 5

cmd SELECT column of \* FROM db\_naam WHERE LIMIT num OFFSET num

## Join tables

9u53



SELECT \* COUNT (\*) AS user\_post\_count FROM posts LEFT JOIN user ON posts.owner\_id = users\_id WHERE post\_id = 10 GROUP\_BY

* SELECT kolommen
  + \* alle
  + posts.\* alle kolommen van één bepaalde tabel
  + posts.id, posts.email meerdere kolommen
  + tabelnaam, veldnaam specifiëren in geval van identiek veldnamen
* COUNT() tellen
  + \* alle
  + Posts.id specifieke kolom
* AS nieuw ontstane kolom door COUNT een naam geven
  + Vb. user\_post\_count
* FROM tabelnaam ingeven posts
* LEFT JOIN links is hoofdtabel, dus alles van de hoofdtabel, vult aan met nullen als rechtse tabel geen data heeft
* ON velden specifiëren voor de JOIN
  + Posts.owner\_id = users.id
* WHERE zoek specifiek veld post.id = 10
* GROUP\_BY

# PGadmin

Pgadmin4 installeren voor Windows :

* Pgadmin port 5050

GEEN @ sign in paswoorden !!!

Register Server

* General naam local\_postgres
* Connection hostname localhost port: 5432
* Postgres postgres

Opbouw van de database :

* Steeds 1 kolom voorzien met prim key.
* Constraints – uitzonderingen instellen
  + Unique colomn
  + Null, not null kan niet null zijn

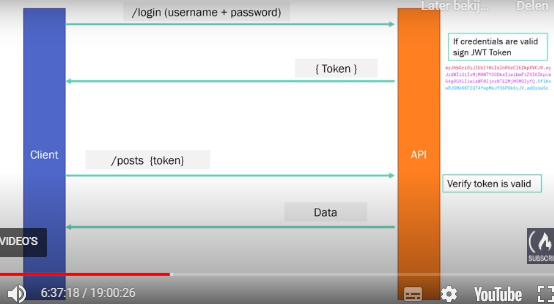
Info i.v.m. data types online: postgres datatypes

# Session based authentication

6u32

* Token
* User login/out
* Storidge of that information

JWT stateless, no storage off information.

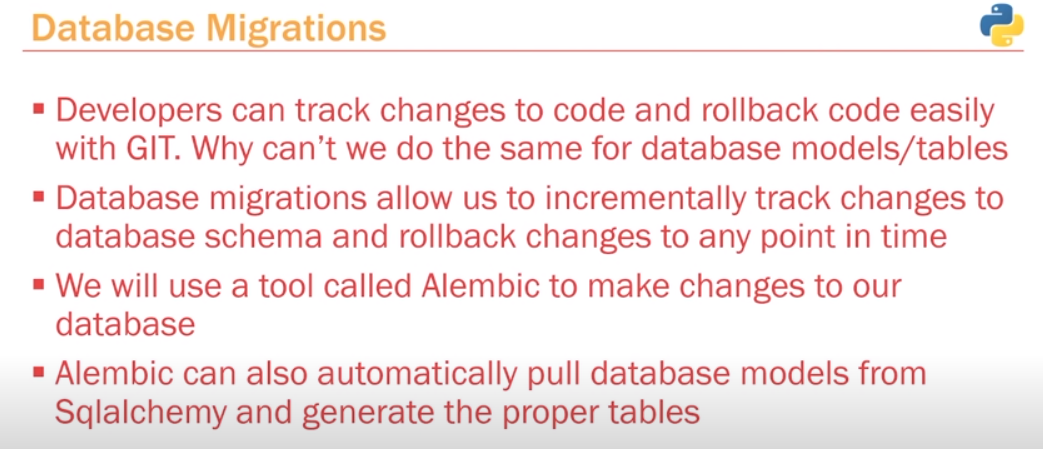


## Token :

* Header
* Payload: data, algemene info
* Verify signature: header, payload, secret info

Alembic

Database migratie tools 10u32



Afbeelding met tekst

Automatisch gegenereerde beschrijving

Afbeelding met diagram

Automatisch gegenereerde beschrijving

## User login

Api connects to database and get back hashed password.

Hash the login password and check if equal with the hash password from the database.

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Generate en verify tokens in Python. 7u01

Pip install “Python-jose[cryptograhy]

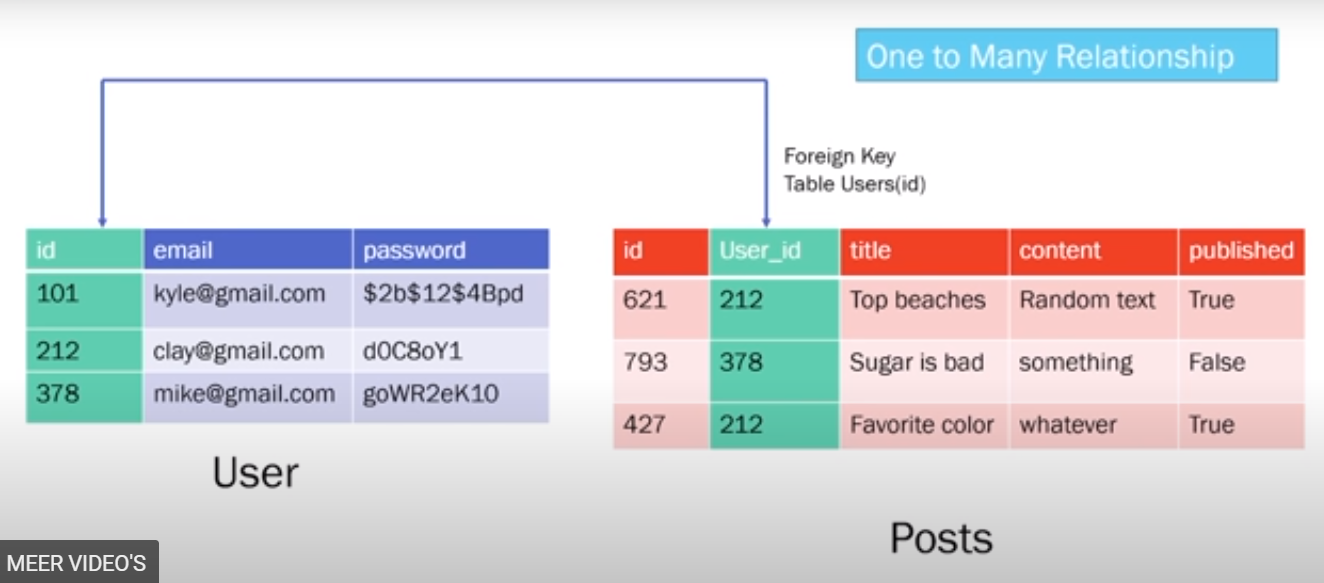
URL Jwt.io website to encrypt token

# Relationships

7u50

Relation between databases, User & Posts.

* Use Foreign Key
* One to many relationship.



Afbeelding met tekst

Automatisch gegenereerde beschrijving

Afbeelding met tekst

Automatisch gegenereerde beschrijving

# Toevoegen van parameters in de URL commando’s.

URL\\.......?........

Alles rechts van ? zijn parameters.

* Limit beperk aantal tot of stel aantal in
* Skip is gelijk aan offset
* Search zoek functie

Afbeelding met tekst

Automatisch gegenereerde beschrijving

# Environment variables

8u54

Database hardcode paswoord & URL, server of database waar dit ook mag zijn.

Omgevingsvariabelen kunnen geconfigureerd worden op elke computer.

CMD echo %Path% print het Path af op het scherm (naam variabele ingeven)

Cmd:

* Systeemeigenschappen
  + Omgevingsvariabelen
    - Gebruiker variabelen
    - Systeem variabelen

Cmd: echo %<naam>% <enter>

Create environment 9u04

Py.config setup example:

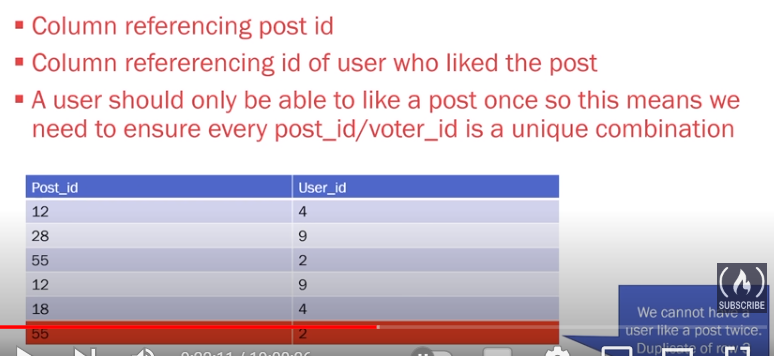
Afbeelding met tekst

Automatisch gegenereerde beschrijving

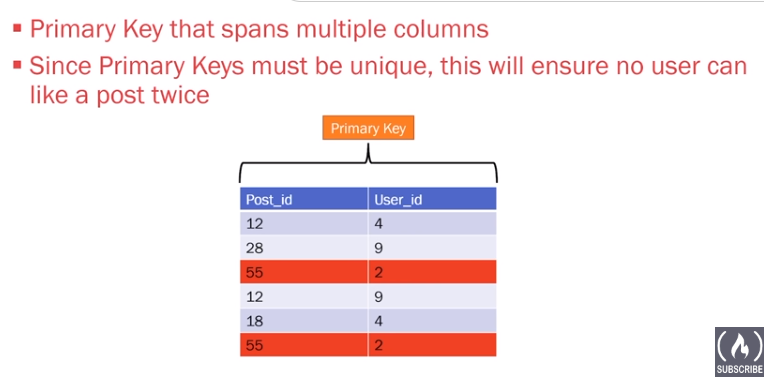
# Vote / Likes system requirements

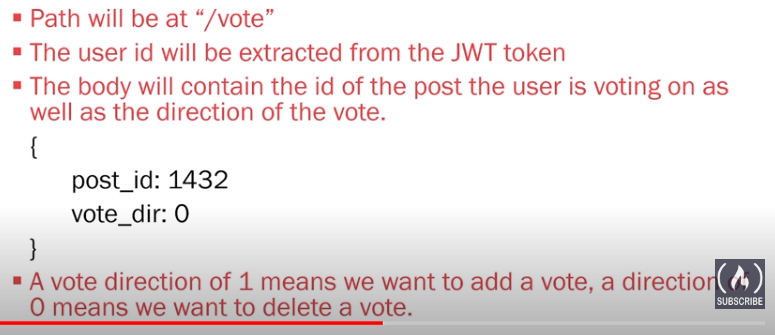
9u21





Composite key





# Cors

11u15

CORS or "Cross-Origin Resource Sharing" refers to the situations when a frontend running in a browser has JavaScript code that communicates with a backend, and the backend is in a different "origin" than the frontend.

Origin¶

An origin is the combination of protocol (http, https), domain (myapp.com, localhost, localhost.tiangolo.com), and port (80, 443, 8080).

So, all these are different origins:

http://localhost

https://localhost

http://localhost:8080

Even if they are all in localhost, they use different protocols or ports, so, they are different "origins".

Use CORSMiddleware¶

You can configure it in your FastAPI application using the CORSMiddleware.

