API PROJECT

ADA-SOFIA RANNELMA

PROJECT OVERVIEW

- RESTful API for person, home and pet management
- Express.js backend with a SQLite database
- Supports CRUD (Create, Read, Update, Delete) operations
- Three main entities: person, pet, home

USED TECHNOLOGIES

- Backend: Node.js and express.js
- Database: SQLite3 and SQL queries with LEFT JOIN to get all values (null values included)
- Middleware: express.json() and custom error handling
- Environment managing: dotenv, and to get variables from .env file in the server.js file used process.env

CHALLENGES & SOLUTIONS

- Challenge: POST route the value put in homeid went to the personid
- Solution: changed GET route SQL stament from p.* to p.id etc.

- Challenge: connecting tables with id, because the autoincrement id makes it challenging
- Solution: connected person(owner) and pet with email and person and home with address and zipcode

TIMETABLE

- 22.3.2025
- ~8h
- Working on the database and server.js

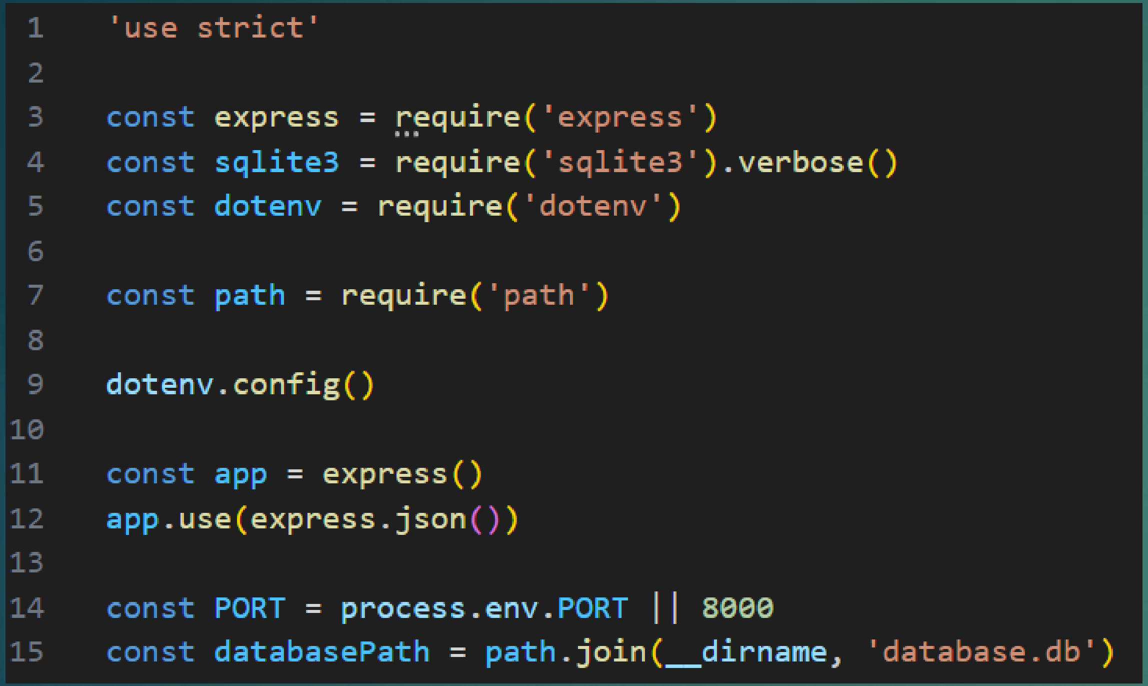
- 25.3.2025
- ~11h
- Finished database and implemented all CRUD operations

- 26.3.2025
- ~3h
- Finishing powerpoint and README

SUMMARY

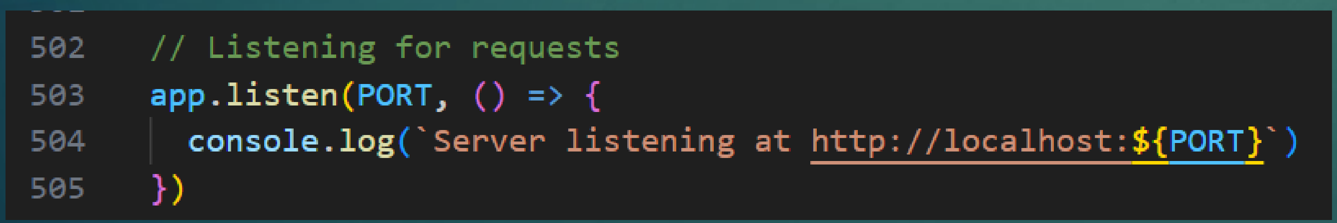
- A RESTful API built with node.js and express.js, using SQLite3 as the database
- Provides CRUD operations for managing persons, pets and homes
- Used technologies: node.js, express.js, sqlite3

PROJECT SETUP & CONFIGURATION



```
1   'use strict'
2
3   const express = require('express')
4   const sqlite3 = require('sqlite3').verbose()
5   const dotenv = require('dotenv')
6
7   const path = require('path')
8
9   dotenv.config()
10
11  const app = express()
12  app.use(express.json())
13
14  const PORT = process.env.PORT || 8000
15  const databasePath = path.join(__dirname, 'database.db')
```

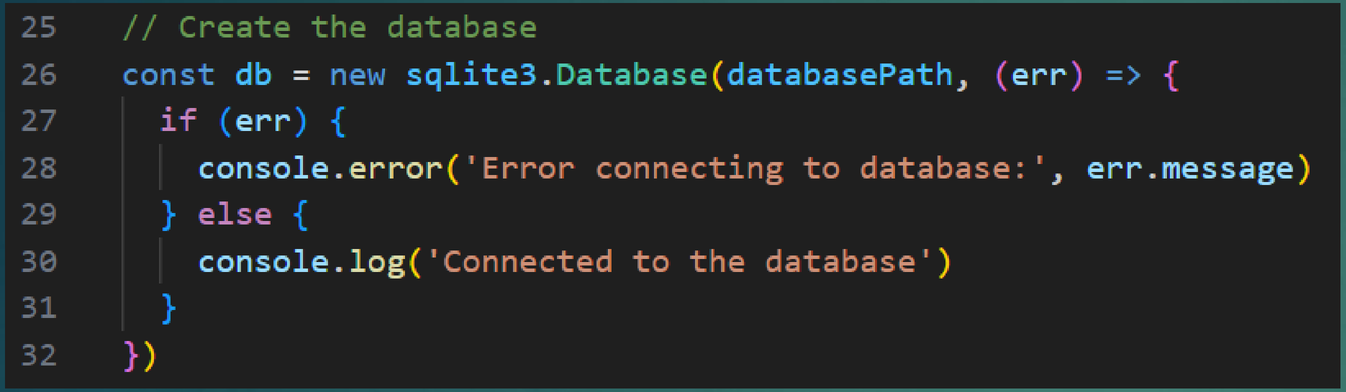
- 'use strict' for improving code quality and performance
- Rows 3 – 7: required dependencies
- Row 9: loading environment variables
- Row 12: middleware to parse JSON request bodies
- Row 14: set port from environment or default 8000
- Row 15: creates a path to the database file



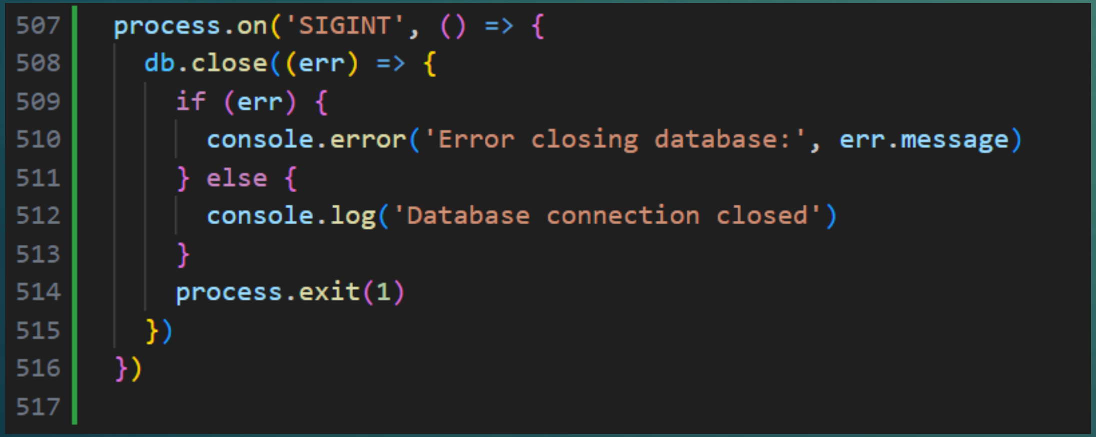
```
// Listening for requests
app.listen(PORT, () => {
  console.log(`Server listening at http://localhost:${PORT}`)
})
```

- Creates an HTTP server and listens to requests

DATABASE



```
// Create the database
const db = new sqlite3.Database(databasePath, (err) => {
  if (err) {
    console.error('Error connecting to database:', err.message)
  } else {
    console.log('Connected to the database')
  }
})
```

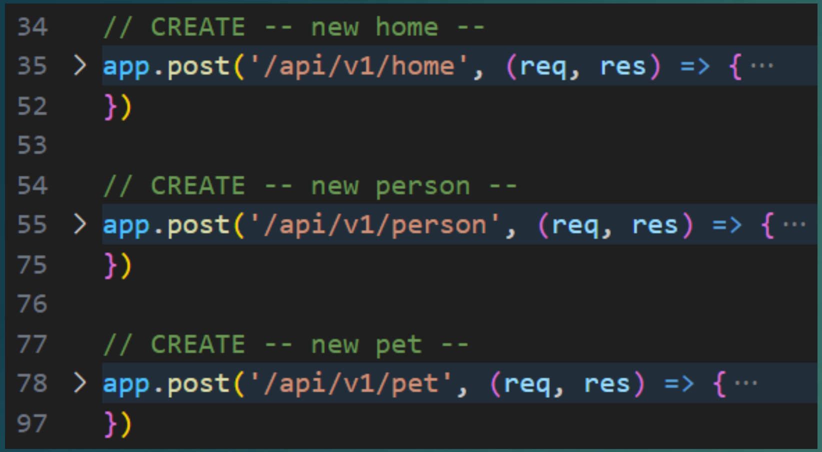


```
process.on('SIGINT', () => {
  db.close((err) => {
    if (err) {
      console.error('Error closing database:', err.message)
    } else {
      console.log('Database connection closed')
    }
    process.exit(1)
  })
})

```

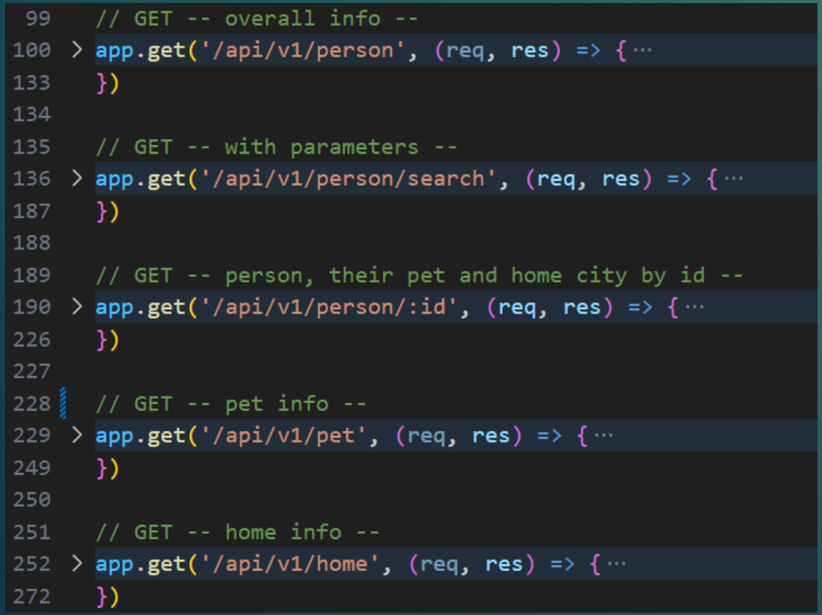
- Uses sqlite3.Database to create a database connection

- Closes the database on termination signal

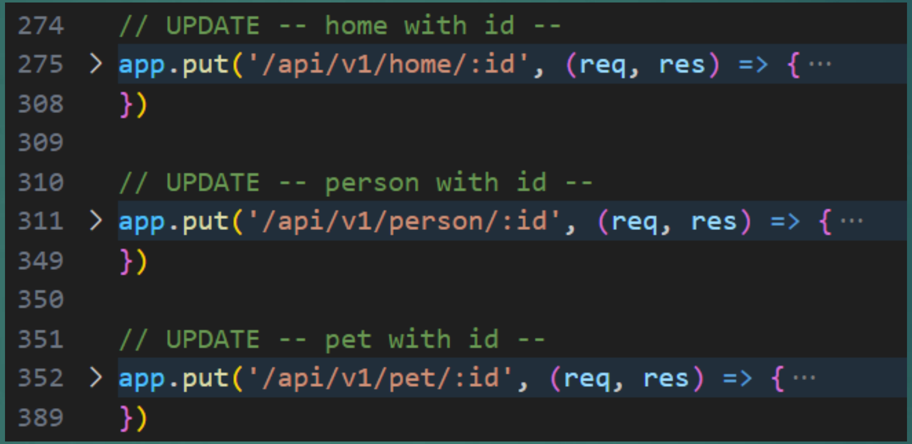
CRUD



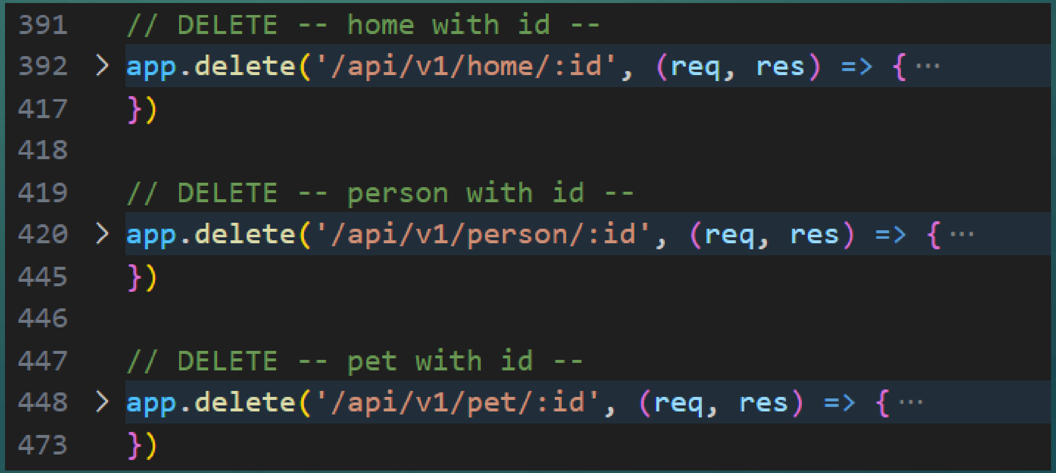
```
// CREATE -- new home --
app.post('/api/v1/home', (req, res) => {⋯
})

// CREATE -- new person --
app.post('/api/v1/person', (req, res) => {⋯
})

// CREATE -- new pet --
app.post('/api/v1/pet', (req, res) => {⋯
})
```



```
// GET -- overall info --
app.get('/api/v1/person', (req, res) => {⋯
})

// GET -- with parameters --
app.get('/api/v1/person/search', (req, res) => {⋯
})

// GET -- person, their pet and home city by id --
app.get('/api/v1/person/:id', (req, res) => {⋯
})

// GET -- pet info --
app.get('/api/v1/pet', (req, res) => {⋯
})

// GET -- home info --
app.get('/api/v1/home', (req, res) => {⋯
})
```



```
// UPDATE -- home with id --
app.put('/api/v1/home/:id', (req, res) => {⋯
})

// UPDATE -- person with id --
app.put('/api/v1/person/:id', (req, res) => {⋯
})

// UPDATE -- pet with id --
app.put('/api/v1/pet/:id', (req, res) => {⋯
})
```

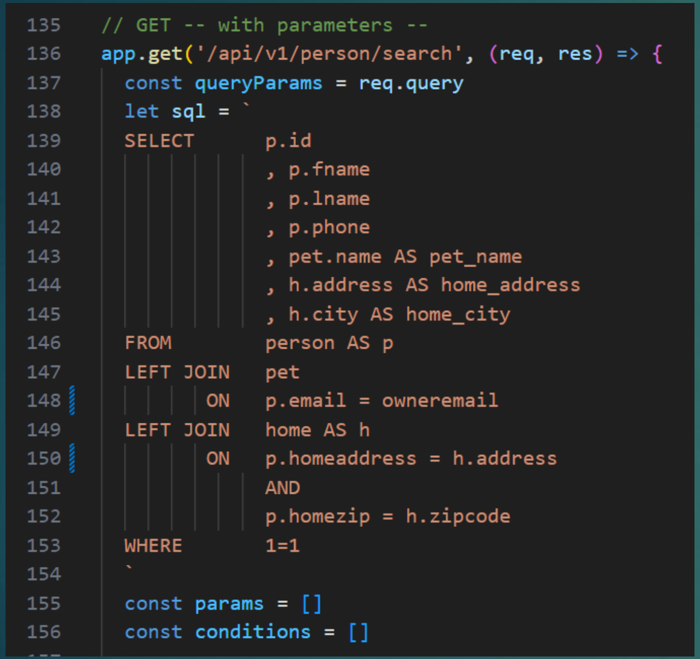


```
// DELETE -- home with id --
app.delete('/api/v1/home/:id', (req, res) => {⋯
})

// DELETE -- person with id --
app.delete('/api/v1/person/:id', (req, res) => {⋯
})

// DELETE -- pet with id --
app.delete('/api/v1/pet/:id', (req, res) => {⋯
})
```

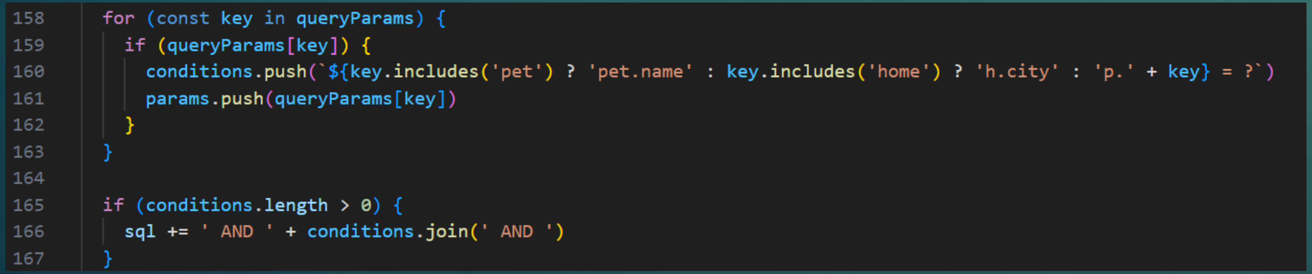
- RESTful routing
- Consistent API version routing
- Has three entities

GET – with parameters

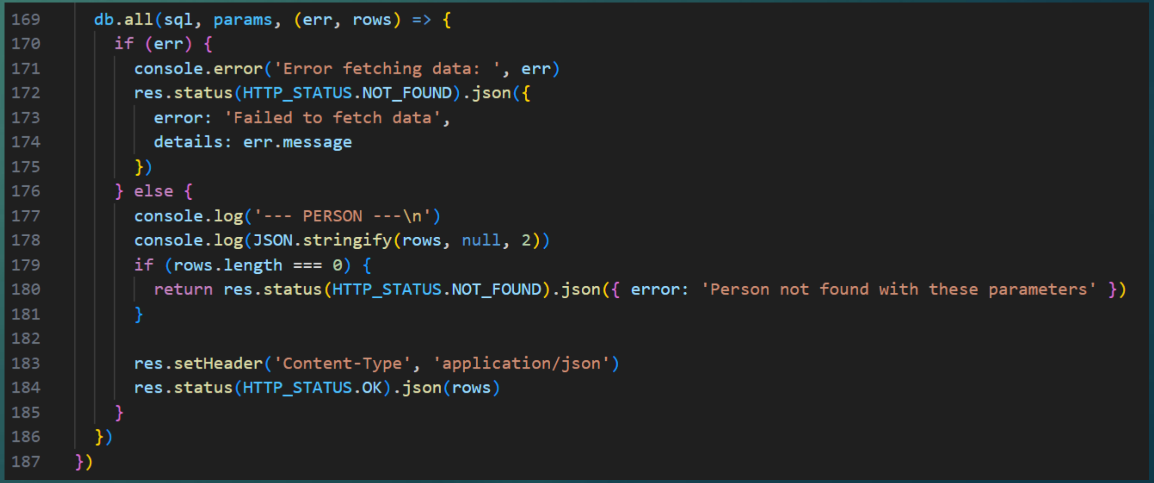
- Allows search with all person parameters plus petname and home city parameters
- Joins tables with LEFT JOIN to get all values
- Dynamically builds SQL statements based on input



```
// GET -- with parameters --
app.get('/api/v1/person/search', (req, res) => {
  const queryParams = req.query
  let sql = `
  SELECT      p.id
              , p.fname
              , p.lname
              , p.phone
              , pet.name AS pet_name
              , h.address AS home_address
              , h.city AS home_city
  FROM        person AS p
  LEFT JOIN   pet
         ON   p.email = owneremail
  LEFT JOIN   home AS h
         ON   p.homeaddress = h.address
              AND
              p.homezip = h.zipcode
  WHERE       1=1
  `
  const params = []
  const conditions = []
```

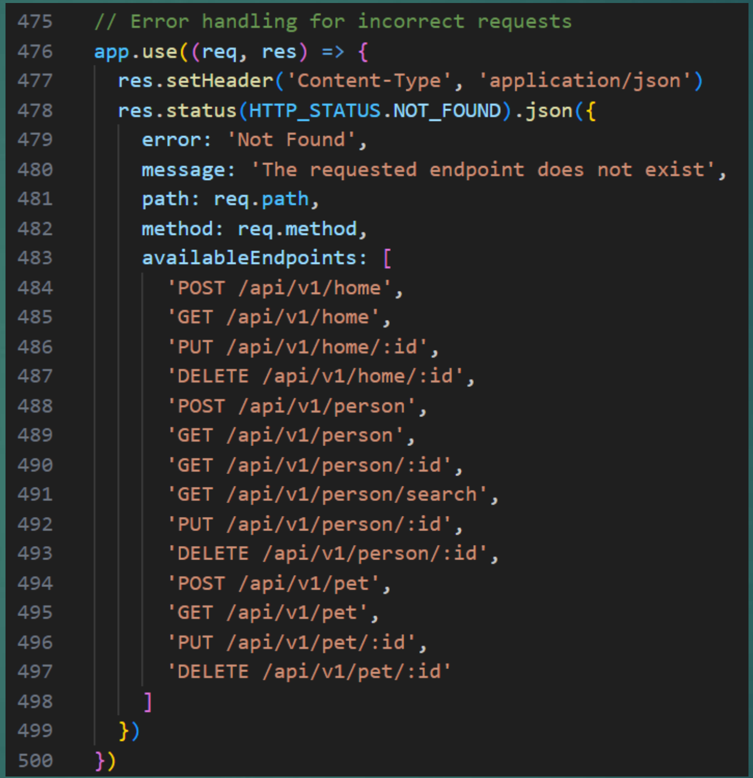


```
  for (const key in queryParams) {
    if (queryParams[key]) {
      conditions.push(`${key.includes('pet') ? 'pet.name' : key.includes('home') ? 'h.city' : 'p.' + key} = ?`)
      params.push(queryParams[key])
    }
  }

  if (conditions.length > 0) {
    sql += ' AND ' + conditions.join(' AND ')
  }
```

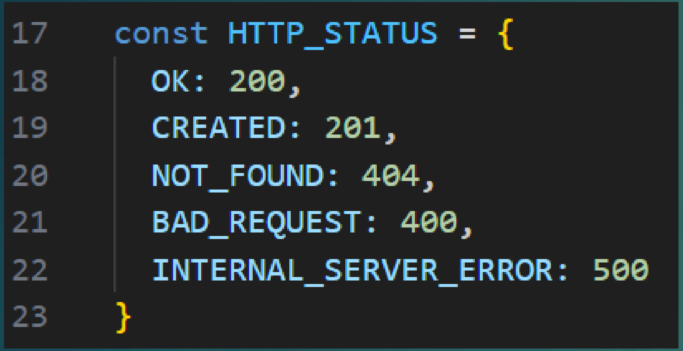


```
  db.all(sql, params, (err, rows) => {
    if (err) {
      console.error('Error fetching data: ', err)
      res.status(HTTP_STATUS.NOT_FOUND).json({
        error: 'Failed to fetch data',
        details: err.message
      })
    } else {
      console.log('--- PERSON ---\n')
      console.log(JSON.stringify(rows, null, 2))
      if (rows.length === 0) {
        return res.status(HTTP_STATUS.NOT_FOUND).json({ error: 'Person not found with these parameters' })
      }

      res.setHeader('Content-Type', 'application/json')
      res.status(HTTP_STATUS.OK).json(rows)
    }
  })
})
```

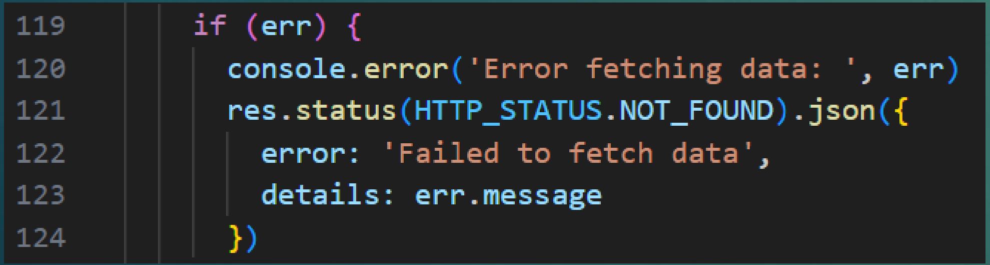
ERROR HANDLING



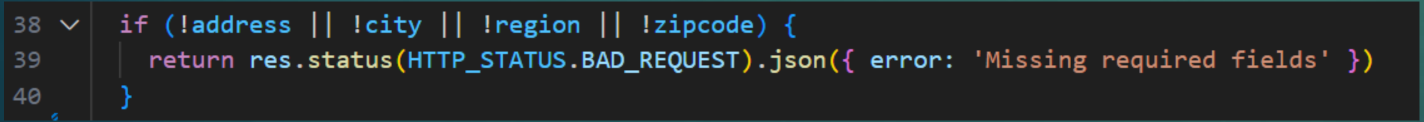
```
// Error handling for incorrect requests
app.use((req, res) => {
  res.setHeader('Content-Type', 'application/json')
  res.status(HTTP_STATUS.NOT_FOUND).json({
    error: 'Not Found',
    message: 'The requested endpoint does not exist',
    path: req.path,
    method: req.method,
    availableEndpoints: [
      'POST /api/v1/home',
      'GET /api/v1/home',
      'PUT /api/v1/home/:id',
      'DELETE /api/v1/home/:id',
      'POST /api/v1/person',
      'GET /api/v1/person',
      'GET /api/v1/person/:id',
      'GET /api/v1/person/search',
      'PUT /api/v1/person/:id',
      'DELETE /api/v1/person/:id',
      'POST /api/v1/pet',
      'GET /api/v1/pet',
      'PUT /api/v1/pet/:id',
      'DELETE /api/v1/pet/:id'
    ]
  })
})
```



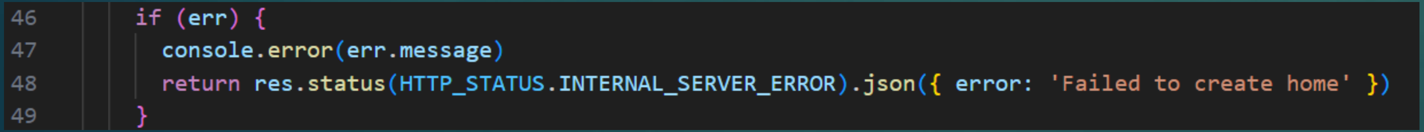
```
const HTTP_STATUS = {
  OK: 200,
  CREATED: 201,
  NOT_FOUND: 404,
  BAD_REQUEST: 400,
  INTERNAL_SERVER_ERROR: 500
}
```



```
      if (err) {
        console.error('Error fetching data: ', err)
        res.status(HTTP_STATUS.NOT_FOUND).json({
          error: 'Failed to fetch data',
          details: err.message
        })
```



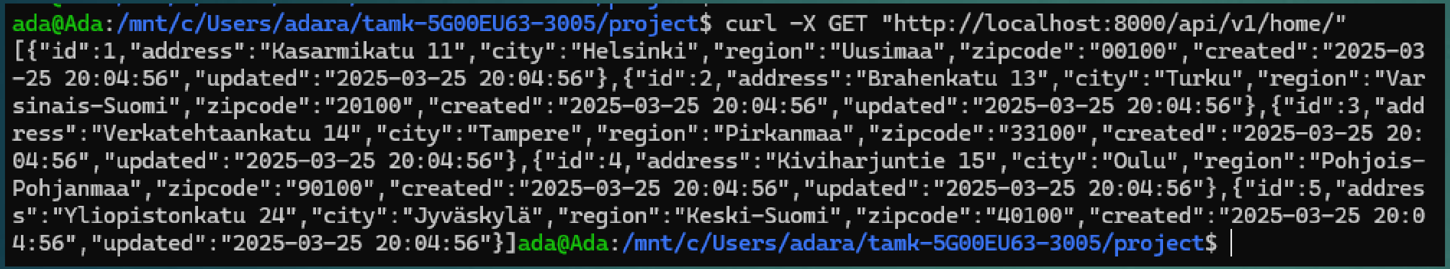
```
if (!address || !city || !region || !zipcode) {
  return res.status(HTTP_STATUS.BAD_REQUEST).json({ error: 'Missing required fields' })
}
```



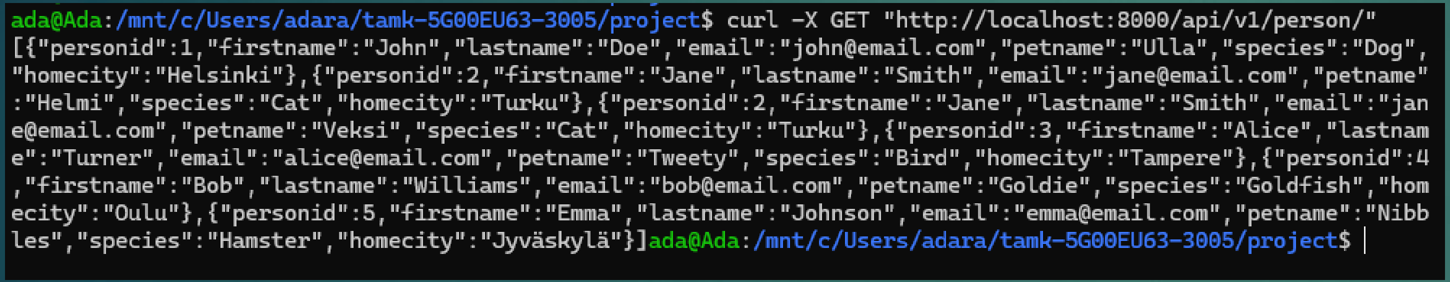
```
    if (err) {
      console.error(err.message)
      return res.status(HTTP_STATUS.INTERNAL_SERVER_ERROR).json({ error: 'Failed to create home' })
    }
```

- HTTP status code management
- Informative error messages
- Validates input before database operations
- Error handling for curl commands

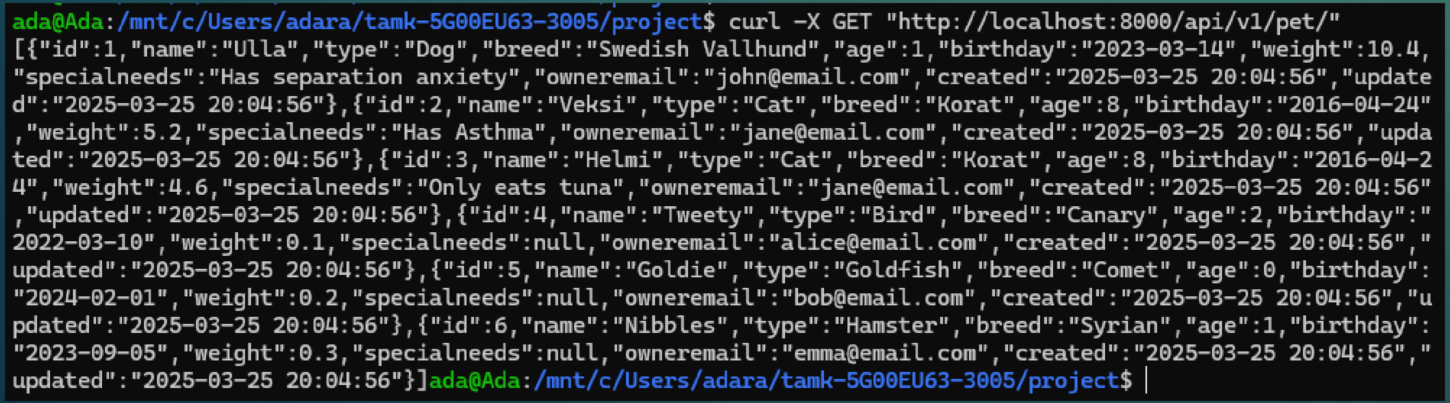
TEST CALLS - GET



```
ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ curl -X GET "http://localhost:8000/api/v1/home/"
[{"id":1,"address":"Kasarmikatu 11","city":"Helsinki","region":"Uusimaa","zipcode":"00100","created":"2025-03
-25 20:04:56","updated":"2025-03-25 20:04:56"},{"id":2,"address":"Brahenkatu 13","city":"Turku","region":"Var
sinais-Suomi","zipcode":"20100","created":"2025-03-25 20:04:56","updated":"2025-03-25 20:04:56"},{"id":3,"add
ress":"Verkatehtaankatu 14","city":"Tampere","region":"Pirkanmaa","zipcode":"33100","created":"2025-03-25 20:
04:56","updated":"2025-03-25 20:04:56"},{"id":4,"address":"Kiviharjuntie 15","city":"Oulu","region":"Pohjois-
Pohjanmaa","zipcode":"90100","created":"2025-03-25 20:04:56","updated":"2025-03-25 20:04:56"},{"id":5,"addres
s":"Yliopistonkatu 24","city":"Jyväskylä","region":"Keski-Suomi","zipcode":"40100","created":"2025-03-25 20:0
4:56","updated":"2025-03-25 20:04:56"}]ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$
```



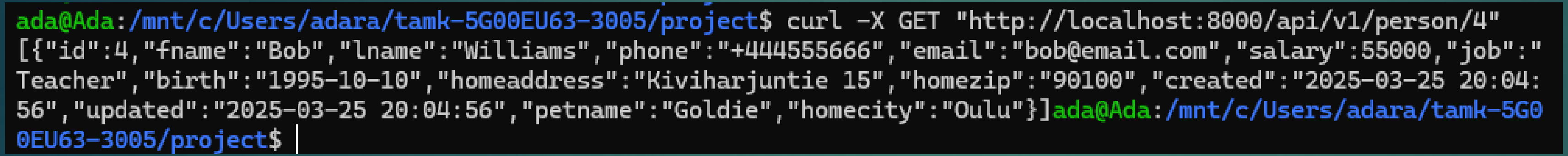
```
ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ curl -X GET "http://localhost:8000/api/v1/person/"
[{"personid":1,"firstname":"John","lastname":"Doe","email":"john@email.com","petname":"Ulla","species":"Dog",
"homecity":"Helsinki"},{"personid":2,"firstname":"Jane","lastname":"Smith","email":"jane@email.com","petname"
:"Helmi","species":"Cat","homecity":"Turku"},{"personid":2,"firstname":"Jane","lastname":"Smith","email":"jan
e@email.com","petname":"Veksi","species":"Cat","homecity":"Turku"},{"personid":3,"firstname":"Alice","lastnam
e":"Turner","email":"alice@email.com","petname":"Tweety","species":"Bird","homecity":"Tampere"},{"personid":4
,"firstname":"Bob","lastname":"Williams","email":"bob@email.com","petname":"Goldie","species":"Goldfish","hom
ecity":"Oulu"},{"personid":5,"firstname":"Emma","lastname":"Johnson","email":"emma@email.com","petname":"Nibb
les","species":"Hamster","homecity":"Jyväskylä"}]ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$
```



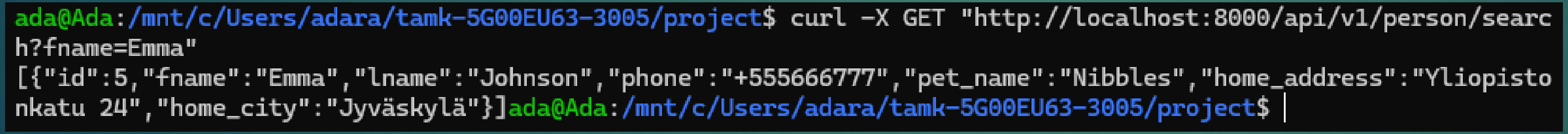
```
ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ curl -X GET "http://localhost:8000/api/v1/pet/"
[{"id":1,"name":"Ulla","type":"Dog","breed":"Swedish Vallhund","age":1,"birthday":"2023-03-14","weight":10.4,
"specialneeds":"Has separation anxiety","owneremail":"john@email.com","created":"2025-03-25 20:04:56","update
d":"2025-03-25 20:04:56"},{"id":2,"name":"Veksi","type":"Cat","breed":"Korat","age":8,"birthday":"2016-04-24"
,"weight":5.2,"specialneeds":"Has Asthma","owneremail":"jane@email.com","created":"2025-03-25 20:04:56","upda
ted":"2025-03-25 20:04:56"},{"id":3,"name":"Helmi","type":"Cat","breed":"Korat","age":8,"birthday":"2016-04-2
4","weight":4.6,"specialneeds":"Only eats tuna","owneremail":"jane@email.com","created":"2025-03-25 20:04:56"
,"updated":"2025-03-25 20:04:56"},{"id":4,"name":"Tweety","type":"Bird","breed":"Canary","age":2,"birthday":"
2022-03-10","weight":0.1,"specialneeds":null,"owneremail":"alice@email.com","created":"2025-03-25 20:04:56","
updated":"2025-03-25 20:04:56"},{"id":5,"name":"Goldie","type":"Goldfish","breed":"Comet","age":0,"birthday":
"2024-02-01","weight":0.2,"specialneeds":null,"owneremail":"bob@email.com","created":"2025-03-25 20:04:56","u
pdated":"2025-03-25 20:04:56"},{"id":6,"name":"Nibbles","type":"Hamster","breed":"Syrian","age":1,"birthday":
"2023-09-05","weight":0.3,"specialneeds":null,"owneremail":"emma@email.com","created":"2025-03-25 20:04:56","
updated":"2025-03-25 20:04:56"}]ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$
```

- curl -X GET "http://localhost:8000/api/v1/home/"
- curl -X GET "http://localhost:8000/api/v1/person/"
- curl -X GET "http://localhost:8000/api/v1/pet/"

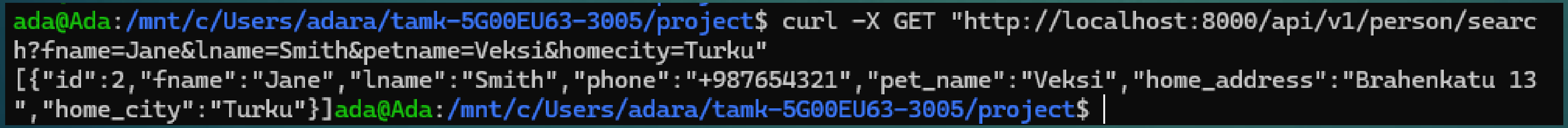
TEST CALLS – GET with parameters



```
ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ curl -X GET "http://localhost:8000/api/v1/person/4"
[{"id":4,"fname":"Bob","lname":"Williams","phone":"+444555666","email":"bob@email.com","salary":55000,"job":"
Teacher","birth":"1995-10-10","homeaddress":"Kiviharjuntie 15","homezip":"90100","created":"2025-03-25 20:04:
56","updated":"2025-03-25 20:04:56","petname":"Goldie","homecity":"Oulu"}]ada@Ada:/mnt/c/Users/adara/tamk-5G0
0EU63-3005/project$ |
```



```
ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ curl -X GET "http://localhost:8000/api/v1/person/searc
h?fname=Emma"
[{"id":5,"fname":"Emma","lname":"Johnson","phone":"+555666777","pet_name":"Nibbles","home_address":"Yliopisto
nkatu 24","home_city":"Jyväskylä"}]ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ |
```



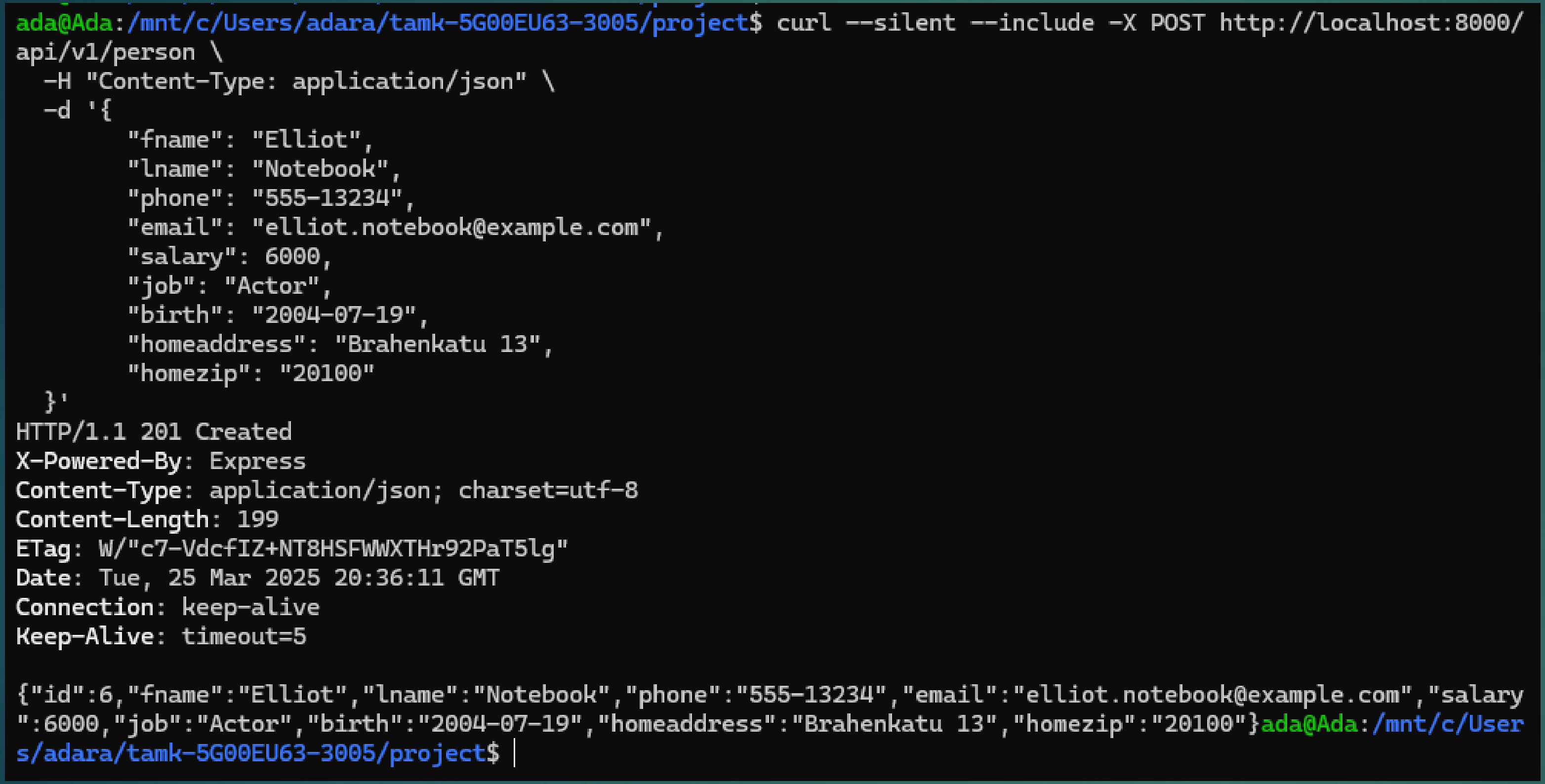
```
ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ curl -X GET "http://localhost:8000/api/v1/person/searc
h?fname=Jane&lname=Smith&petname=Veksi&homecity=Turku"
[{"id":2,"fname":"Jane","lname":"Smith","phone":"+987654321","pet_name":"Veksi","home_address":"Brahenkatu 13
","home_city":"Turku"}]ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ |
```

○ curl -X GET "http://localhost:8000/api/v1/person/search?fname=Emma"

○ curl -X GET "http://localhost:8000/api/v1/person/search?fname=Jane&lname=Smith&petname=Veksi&homecity=Turku"

○ curl -X GET "http://localhost:8000/api/v1/person/4"

TEST CALLS - POST

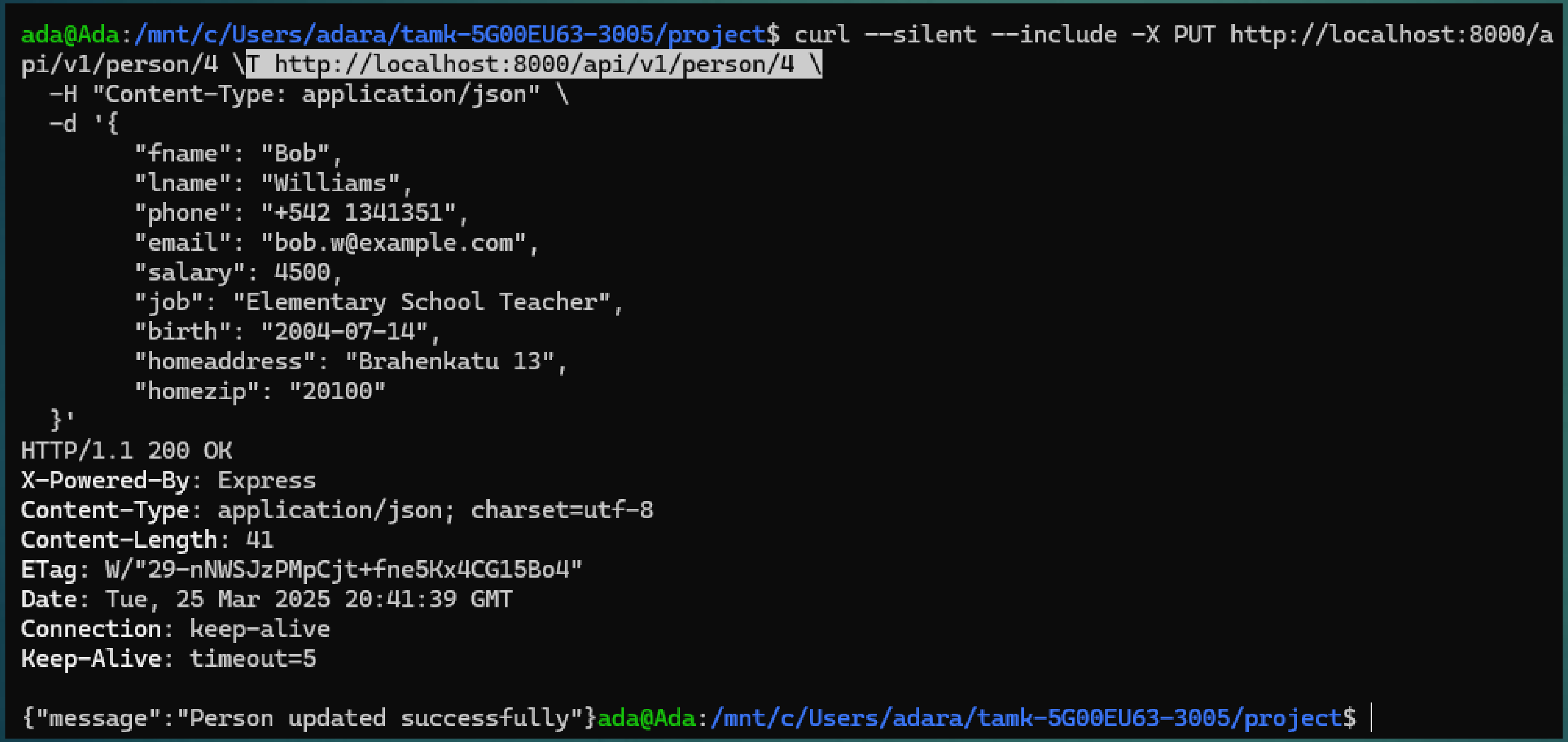


```
ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ curl --silent --include -X POST http://localhost:8000/
api/v1/person \
  -H "Content-Type: application/json" \
  -d '{
        "fname": "Elliot",
        "lname": "Notebook",
        "phone": "555-13234",
        "email": "elliot.notebook@example.com",
        "salary": 6000,
        "job": "Actor",
        "birth": "2004-07-19",
        "homeaddress": "Brahenkatu 13",
        "homezip": "20100"
  }'
HTTP/1.1 201 Created
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 199
ETag: W/"c7-VdcfIZ+NT8HSFWWXTHr92PaT5lg"
Date: Tue, 25 Mar 2025 20:36:11 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{"id":6,"fname":"Elliot","lname":"Notebook","phone":"555-13234","email":"elliot.notebook@example.com","salary
":6000,"job":"Actor","birth":"2004-07-19","homeaddress":"Brahenkatu 13","homezip":"20100"}ada@Ada:/mnt/c/User
s/adara/tamk-5G00EU63-3005/project$
```

- curl --silent --include -X POST http://localhost:8000/api/v1/person \

```
 -H "Content-Type: application/json" \
 -d '{
        "fname": "Elliot",
        "lname": "Notebook",
        "phone": "555-13234",
        "email": "elliot.notebook@example.com",
        "salary": 6000,
        "job": "Actor",
        "birth": "2004-07-19",
        "homeaddress": "Brahenkatu 13",
        "homezip": "20100"
 }'
```

TEST CALLS – UPDATE (PUT)



```
ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ curl --silent --include -X PUT http://localhost:8000/a
pi/v1/person/4 \T http://localhost:8000/api/v1/person/4 \
  -H "Content-Type: application/json" \
  -d '{
        "fname": "Bob",
        "lname": "Williams",
        "phone": "+542 1341351",
        "email": "bob.w@example.com",
        "salary": 4500,
        "job": "Elementary School Teacher",
        "birth": "2004-07-14",
        "homeaddress": "Brahenkatu 13",
        "homezip": "20100"
  }'
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 41
ETag: W/"29-nNWSJzPMpCjt+fne5Kx4CG15Bo4"
Date: Tue, 25 Mar 2025 20:41:39 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{"message":"Person updated successfully"}ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ 
```

- curl --silent --include -X PUT http://localhost:8000/api/v1/person/4 \

  -H "Content-Type: application/json" \

  -d '{

  "fname": "Bob",

  "lname": "Williams",

  "phone": "+542 1341351",

  "email": "bob.w@example.com",

  "salary": 4500,

  "job": "Elementary School Teacher",

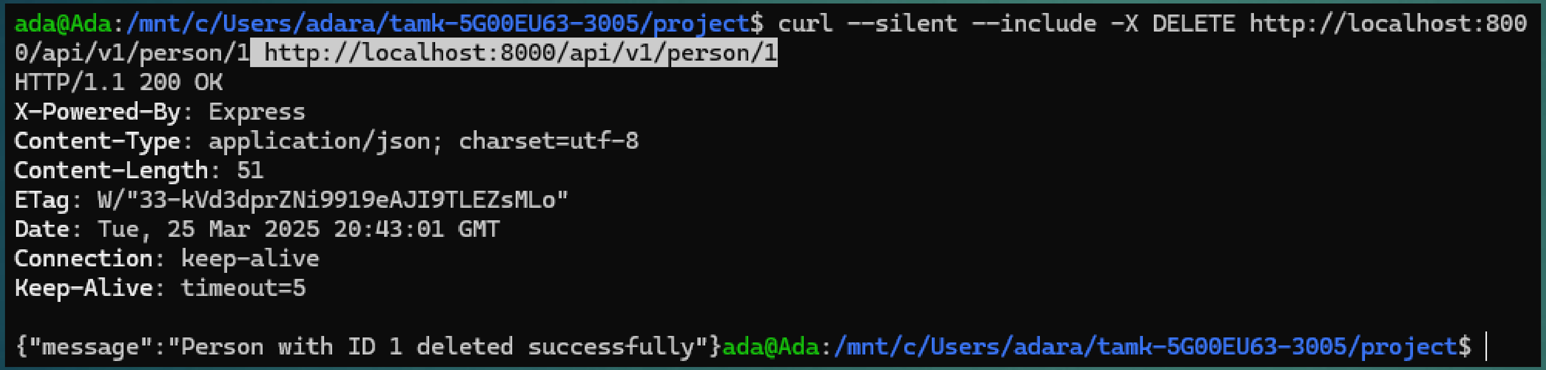
  "birth": "2004-07-14",

  "homeaddress": "Brahenkatu 13",

  "homezip": "20100"

  }'

TEST CALLS - DELETE

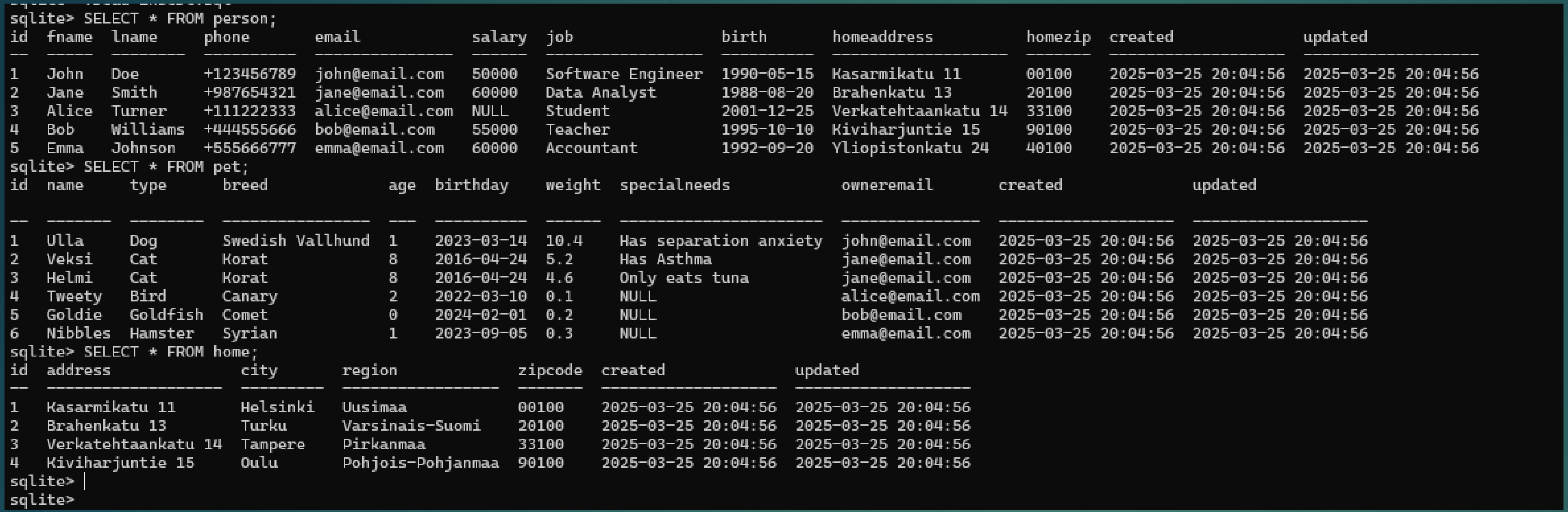


```
ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ curl --silent --include -X DELETE http://localhost:800
0/api/v1/person/1 http://localhost:8000/api/v1/person/1
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 51
ETag: W/"33-kVd3dprZNi9919eAJI9TLEZsMLo"
Date: Tue, 25 Mar 2025 20:43:01 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{"message":"Person with ID 1 deleted successfully"}ada@Ada:/mnt/c/Users/adara/tamk-5G00EU63-3005/project$ |
```

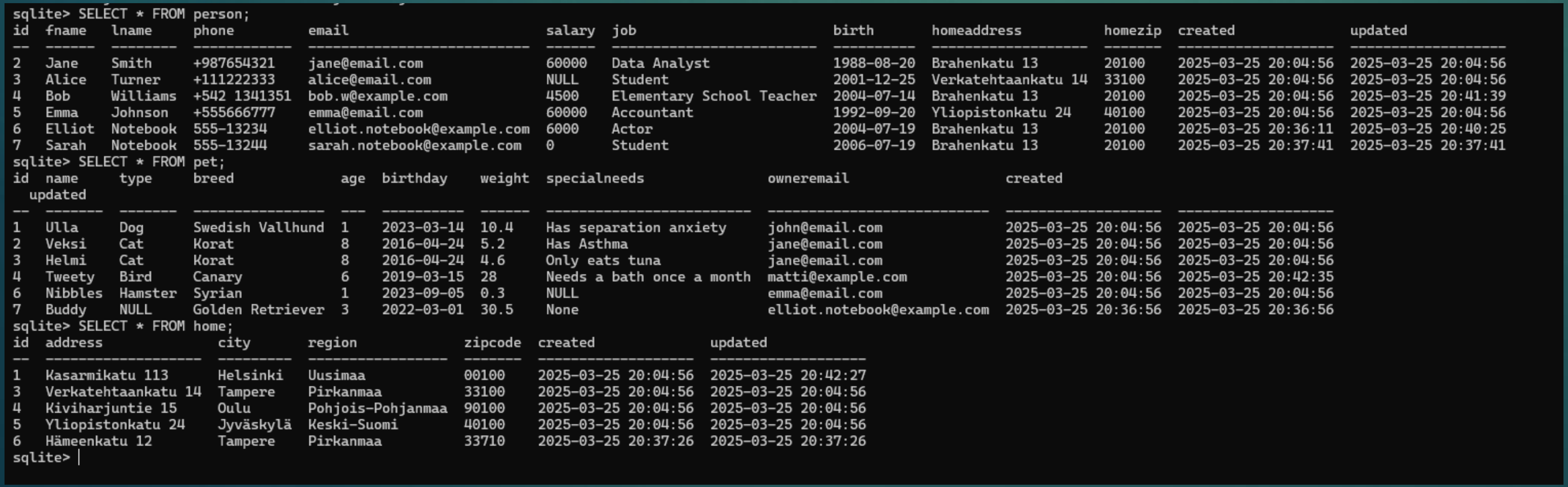
- curl --silent --include -X DELETE http://localhost:8000/api/v1/person/1

DATABASE TABLES

BEFORE AND AFTER TEST CALLS



```
sqlite> SELECT * FROM person;
id  fname  lname     phone       email            salary  job                birth       homeaddress          homezip  created              updated
--  -----  --------  ----------  ---------------  ------  -----------------  ----------  -------------------  -------  -------------------  -------------------
1   John   Doe       +123456789  john@email.com   50000   Software Engineer  1990-05-15  Kasarmikatu 11       00100    2025-03-25 20:04:56  2025-03-25 20:04:56
2   Jane   Smith     +987654321  jane@email.com   60000   Data Analyst       1988-08-20  Brahenkatu 13        20100    2025-03-25 20:04:56  2025-03-25 20:04:56
3   Alice  Turner    +111222333  alice@email.com  NULL    Student            2001-12-25  Verkatehtaankatu 14  33100    2025-03-25 20:04:56  2025-03-25 20:04:56
4   Bob    Williams  +444555666  bob@email.com    55000   Teacher            1995-10-10  Kiviharjuntie 15     90100    2025-03-25 20:04:56  2025-03-25 20:04:56
5   Emma   Johnson   +555666777  emma@email.com   60000   Accountant         1992-09-20  Yliopistonkatu 24    40100    2025-03-25 20:04:56  2025-03-25 20:04:56
sqlite> SELECT * FROM pet;
id  name     type      breed             age  birthday    weight  specialneeds             owneremail       created              updated

--  -------  --------  ----------------  ---  ----------  ------  -----------------------  ---------------  -------------------  -------------------
1   Ulla     Dog       Swedish Vallhund  1    2023-03-14  10.4    Has separation anxiety   john@email.com   2025-03-25 20:04:56  2025-03-25 20:04:56
2   Veksi    Cat       Korat             8    2016-04-24  5.2     Has Asthma               jane@email.com   2025-03-25 20:04:56  2025-03-25 20:04:56
3   Helmi    Cat       Korat             8    2016-04-24  4.6     Only eats tuna           jane@email.com   2025-03-25 20:04:56  2025-03-25 20:04:56
4   Tweety   Bird      Canary            2    2022-03-10  0.1     NULL                     alice@email.com  2025-03-25 20:04:56  2025-03-25 20:04:56
5   Goldie   Goldfish  Comet             0    2024-02-01  0.2     NULL                     bob@email.com    2025-03-25 20:04:56  2025-03-25 20:04:56
6   Nibbles  Hamster   Syrian            1    2023-09-05  0.3     NULL                     emma@email.com   2025-03-25 20:04:56  2025-03-25 20:04:56
sqlite> SELECT * FROM home;
id  address              city      region             zipcode  created              updated
--  -------------------  --------  -----------------  -------  -------------------  -------------------
1   Kasarmikatu 11       Helsinki  Uusimaa            00100    2025-03-25 20:04:56  2025-03-25 20:04:56
2   Brahenkatu 13        Turku     Varsinais-Suomi    20100    2025-03-25 20:04:56  2025-03-25 20:04:56
3   Verkatehtaankatu 14  Tampere   Pirkanmaa          33100    2025-03-25 20:04:56  2025-03-25 20:04:56
4   Kiviharjuntie 15     Oulu      Pohjois-Pohjanmaa  90100    2025-03-25 20:04:56  2025-03-25 20:04:56
sqlite> |
sqlite>
```



```
sqlite> SELECT * FROM person;
id  fname   lname     phone        email                        salary  job                        birth       homeaddress          homezip  created              updated
--  ------  --------  -----------  ---------------------------  ------  -------------------------  ----------  -------------------  -------  -------------------  -------------------
2   Jane    Smith     +987654321   jane@email.com               60000   Data Analyst               1988-08-20  Brahenkatu 13        20100    2025-03-25 20:04:56  2025-03-25 20:04:56
3   Alice   Turner    +111222333   alice@email.com              NULL    Student                    2001-12-25  Verkatehtaankatu 14  33100    2025-03-25 20:04:56  2025-03-25 20:04:56
4   Bob     Williams  +542 1341351 bob.w@example.com            4500    Elementary School Teacher  2004-07-14  Brahenkatu 13        20100    2025-03-25 20:04:56  2025-03-25 20:41:39
5   Emma    Johnson   +555666777   emma@email.com               60000   Accountant                 1992-09-20  Yliopistonkatu 24    40100    2025-03-25 20:04:56  2025-03-25 20:04:56
6   Elliot  Notebook  555-13234    elliot.notebook@example.com  6000    Actor                      2004-07-19  Brahenkatu 13        20100    2025-03-25 20:36:11  2025-03-25 20:40:25
7   Sarah   Notebook  555-13244    sarah.notebook@example.com   0       Student                    2006-07-19  Brahenkatu 13        20100    2025-03-25 20:37:41  2025-03-25 20:37:41
sqlite> SELECT * FROM pet;
id  name     type     breed             age  birthday    weight  specialneeds               owneremail                   created
  updated
--  -------  -------  ----------------  ---  ----------  ------  -------------------------  ---------------------------  -------------------  -------------------
1   Ulla     Dog      Swedish Vallhund  1    2023-03-14  10.4    Has separation anxiety     john@email.com               2025-03-25 20:04:56  2025-03-25 20:04:56
2   Veksi    Cat      Korat             8    2016-04-24  5.2     Has Asthma                 jane@email.com               2025-03-25 20:04:56  2025-03-25 20:04:56
3   Helmi    Cat      Korat             8    2016-04-24  4.6     Only eats tuna             jane@email.com               2025-03-25 20:04:56  2025-03-25 20:04:56
4   Tweety   Bird     Canary            6    2019-03-15  28      Needs a bath once a month  matti@example.com            2025-03-25 20:04:56  2025-03-25 20:42:35
6   Nibbles  Hamster  Syrian            1    2023-09-05  0.3     NULL                       emma@email.com               2025-03-25 20:04:56  2025-03-25 20:04:56
7   Buddy    NULL     Golden Retriever  3    2022-03-01  30.5    None                       elliot.notebook@example.com  2025-03-25 20:36:56  2025-03-25 20:36:56
sqlite> SELECT * FROM home;
id  address              city       region             zipcode  created              updated
--  -------------------  ---------  -----------------  -------  -------------------  -------------------
1   Kasarmikatu 113      Helsinki   Uusimaa            00100    2025-03-25 20:04:56  2025-03-25 20:42:27
3   Verkatehtaankatu 14  Tampere    Pirkanmaa          33100    2025-03-25 20:04:56  2025-03-25 20:04:56
4   Kiviharjuntie 15     Oulu       Pohjois-Pohjanmaa  90100    2025-03-25 20:04:56  2025-03-25 20:04:56
5   Yliopistonkatu 24    Jyväskylä  Keski-Suomi        40100    2025-03-25 20:04:56  2025-03-25 20:04:56
6   Hämeenkatu 12        Tampere    Pirkanmaa          33710    2025-03-25 20:37:26  2025-03-25 20:37:26
sqlite> |
```

THANK YOU!