

פרויקט סיום בפייתון (v1.1)

יצירת שרת עם מסד נתונים שמדבר עם קליינטים שמייצגים "תחנות המים"

הפרויקט מורכב מארבעת הקבצים הבאים:

server.py - מייצג את השרת

data.sqlite - מסד נתונים מבוסס SQLite3 שישמור מידע על התחנות. (נמצא באותה התיקייה כמו server.py)

client.py - מייצג את הקליינט

status.txt - מכיל מידע כטקסט של התחנה. נמצא באותה התיקייה כמו client.py

יש להגיש לפחות את השלושה הבאים: server.py, client.py, status.txt

(ניתן להגיש כקובץ zip שיכיל את הקבצים)

(את מסד הנתונים יש לתת ל-server.py ליצור במידה ולא קיים כפי שמוסבר בהמשך המסמך)

הסבר עבור הקליינט

עבור הקליינט, ראשית עליו להתחבר לשרת (מידע על השרת יפורט בהמשך המסמך) ואז עליו לרוץ בלולאה וכל דקה עליו לפתוח (מחדש) ולקרוא את המידע שב-status.txt ולשלוח את המידע לשרת.

קובץ הטקסט status.txt יכיל שלוש שורות.

- השורה הראשונה תכיל את ה-station_id (מספר int כלשהו)
- השורה השנייה תכיל את מצב אזעקה 1 (הטקסט 0 או הטקסט 1).
- והשורה השלישית תכיל את מצב אזעקה 2 (הטקסט 0 או הטקסט 1)

למשל אם הקובץ מכיל את השורות הבאות:

```
123
0
1
```

מדובר בתחנה מספר 123, אזעקה 1 כבויה (0) ואזעקה 2 דלוקה (1)

ניתן יהיה לערוך את קובץ הטקסט (עם תוכנה כגון notepad או gedit) ולשמור תוך כדי הריצה של client.py, וכאשר client.py יפתח ויקרא את הקובץ הוא יישלח את המידע המעודכן לשרת.

טיפ - את המידע לשרת אפשר לשלוח בסגנון של המחרוזת הבאה:

"ID ALARM1 ALARM2"

למשל:

"123 1 0"

אם נפצל את הטקסט הזה בצד השרת עם מתודת split (של str) נקבל 3 מחרוזות כ-list
האיבר ה-0 מכיל את ה-station_id
האיבר ה-1 מכיל את alarm1
האיבר ה-2 מכיל את alarm2

הסבר עבור השרת

כשהשרת מתחיל לפעול, הוא קודם יוצר טבלה בשם station_status במידה ולא קיימת בתוך database.
לשם כך ניתן להשתמש בשאילתה הבאה:

```
CREATE TABLE IF NOT EXISTS station_status (  
    station_id INT,  
    last_date TEXT,  
    alarm1 INT,  
    alarm2 INT,  
    PRIMARY KEY(station_id) );
```

כפי שרשום לעיל הטבלה תכיל את ארבעת העמודות הבאות:
station_id - הוא המספר המזהה של התחנה
last_date - הוא התאריך האחרון בו נשלחה ההודעה (עבור תחנה עם אותו מספר מזהה)
alarm1 - הוא מצב אזעקה 1 (הערך שלו אמור להיות 1 אם האזעקה דלוקה או 0 אם היא לא)
alarm2 - הוא מצב אזעקה 2 (הערך שלו אמור להיות 1 אם האזעקה דלוקה או 0 אם היא לא)

לאחר מכן השרת ממתיין יוצר socket כדי להאזין להודעות בלולאה מהקליינטיים, והוא מעדכן את ה-database שלו בהתאם.

העדכון יכלול את מספר התחנה, מצב אזעקה 1, מצב אזעקה 2, וכן את התאריך (last_date).

את התאריך יש ליצור בצד השרת בעת קבלת ההודעה ולרשום במסד הנתונים כטקסט, לפי הפורמט הבא:
"YYYY-MM-DD HH:mm"
למשל:

"2019-01-20 10:30"

את הפורמט הזה ניתן לקבל באמצעות הפונקציה strftime של datetime להלן קטע שמדגים זאת:

```
import datetime  
  
last_date = datetime.datetime.now().strftime('%Y-%M-%d %H:%m')  
print(last_date)
```

יש להעדיף שעבור כל תחנת מים תהיה שורה אחת בלבד במסד הנתונים (ז"א בפעם הראשונה שהתחנה מספר 123 שולחת מידע, תתווסף שורה עבודה. אך בפעמים הנוספות, לא תתווסף שורה אלא תתעדכן השורה הקיימת.

כדי לאפשר את האמור לעיל ניתן להעזר בשאילת insert or replace במקום שאילת insert רגילה:

```
INSERT OR REPLACE INTO station_status  
VALUES (?, ?, ?, ?)
```

נקודות חשובות:

- השרת משתמש ב-port קבוע
- יש לספק הגנות מ-exceptions בהתאם לצורך
- יש לוודא שהשרת מוגן מהתקפה של code injection
- יש לוודא שהשרת מסוגל להתמודד עם מספר קליינטים שפעילים ושולחים מידע. לצורך בדיקה של נקודה זו יש להעתיק את client.py ואת status.txt לתיקיה נוספת, לשנות את התוכן של status.txt כך יכיל מספר זהות אחר ולהריץ את העותק של client במקביל עם הגרסה המקורית.

בהצלחה!