

Relazione progetto Advanced Programming Languages A.A. 21-22

Personal Stock Market

Link repository: <https://github.com/Salvo-Sipala/APL-Salvatore-Sipala-21-22.git>

Autore: *Salvatore Sipala (matricola: 055000438)*

Introduzione

Personal Stock Market è un'applicazione mobile multiplatforma Android e iOS che permette ad un utente di ricercare, visualizzare e seguire determinati Stock finanziari, per poterne osservare l'andamento in borsa.

Struttura

Il progetto è suddiviso in tre parti:

1. Un'app client per smartphone (Android/iOS) con l'utilizzo del framework Xamarin e del simulatore, implementata con il linguaggio C#.
2. Un web server backend con l'utilizzo del framework Django e collegamento con il database MongoDB, il tutto implementato con il linguaggio Python.
3. Un secondo web server (collegato con il precedente), utilizzato per effettuare dei calcoli sulle ricerche degli Stock effettuati, implementato in C++.

Dipendenze e packages utilizzati

1. App client
 - a. Visual studio (IDE) con l'installazione di Xamarin e i rispettivi sdk Android e iOS, di cui è stato utilizzato il simulatore Android,
 - b. Xamarin.Forms,
 - c. Xamarin.Essentials,
 - d. Newtonsoft.Json tramite Nuget Package
 - e. System.Net.Http.
2. Web server Python
 - a. PyCharm (IDE),
 - b. Python 3.9,
 - c. Django web framework,
 - d. Django REST framework,
 - e. MongoDB con GUI MongoDB Compass,
 - f. Django database mapper.
 - g. Software Ngrok (<https://ngrok.com/>) che ha permesso di esporre il web server tramite un tunneling (il quale gira in localhost) su un indirizzo in rete creato dallo stesso Ngrok.

3. Web server C++

- a. Visual studio (IDE),
- b. STL cpp,
- c. Cpprestsdk (Microsoft) per comunicazione client-server,

Procedura per avviare il progetto

1. App client:

- a. Per effettuare il tunneling richiesto dal simulatore di Xamarin mandare in esecuzione ngrok e digitare '\$ ngrok http 8081' esponendo così tale indirizzo per il nostro web server.
- b. Si ottiene un indirizzo IP da inserire al percorso "APL-Salvatore-Sipala-21-22\PersonalStockMarcket\C#-FrontEnd\FrontEnd\Constants\RestUrl.cs" del progetto.
- c. Installare il package tramite Nuget Package "Newtonsoft.Json".
- d. A questo punto è possibile far partire il progetto Xamarin tramite Visual Studio (dopo aver configurato un emulatore Android o iOS).

2. Web server Python

- a. Aprire progetto con PyCharm.
- b. Aprire MongoDB Compass ed effettuare la connessione con localhost.
- c. Creare un DB con lo stesso nome presente al percorso APL-Salvatore-Sipala-21-22\PersonalStockMarcket\Python_BackEnd\Python_BackEnd\settings.py nella sezione relativa al dizionario DATABASES {... 'NAME': '...'}.
- d. Eseguire i comandi per popolare il DB con le varie Tabelle e collezioni:
 - i. \$ python3 manage.py makemigrations
 - ii. \$ python3 manage.py migrate
- e. Creare il superuser:
 - i. \$ python3 manage.py createsuperuser
- f. Configurare il progetto per esporre il webserver alla porta 8081.
- g. A questo punto è possibile far partire il progetto Python tramite PyCharm.

3. Web server C++

- a. Aprire il progetto con visual studio.
- b. Installare il package tramite Nuget Package "cpprestsdk.v141".
- c. A questo punto è possibile far partire il progetto C++ tramite visual studio.

Funzionalità previste

Il progetto nella sua interezza è costituito da:

1. Gestione Login e Registrazione di un utente.
2. Ricerca di uno Stock finanziario tramite il servizio offerto da "Yahoo api finance" che fornisce delle API utili per recuperare le informazioni richieste.
3. Visualizzazione dello stock finanziario cercato e auto-inserimento dello stesso nel DB.
4. Aggiungere lo stock ad una lista di preferiti, e successivamente visualizzarli.

Il tutto è stato possibile tramite l'utilizzo di Rest API per effettuare la comunicazione tra le 3 parti del progetto, definite in linguaggi differenti.