Documentazione del progetto

"Borsa azioni osservatore di Poors&homeless people company"

di Tubia Tommaso, Perazetta Davide

1. Descrizione del progetto

* Il nostro progetto consiste nel ottenere dal webservice "alpha vantage" le azioni di un'azienda nelle ultima 72 ore con intervalli personalizzati, e successivamente disegnare un piccolo grafico seguendo l'andamento della borsa, visualizzando un valore se è aumentato o diminuito.
* L'idea è quella di creare 3 interfacce (configurazione, principale e visualizza borsa) usando un'interfaccia swing statica; la schermata borsa dovrebbe disegnare un grafico manuale con una linea che segna per ogni ora quanto era il valore della borsa

1. Architettura del sistema

* L'architettura del sistema è suddivisa nel modello MWCO (model, view, control, other) che

comprende tutto il necessario, usando swing e jaxb di supporto.

* I componenti chiave sono il file "MainApplication" (file avviatore dell'applicazione) , e il file "TimeSeriesData", che contiene l'indice e la struttura del file XML come dovrebbe essere tradotto da JSON
* Le interazioni dei componenti avviene attraverso i controller, le chiamate alle funzioni e un file txt contenente le opzioni.
* i parametri di input sono 3 : nome dell'azienda, il tuo nome utente e la chiave di accesso di alpha vantage

1. Descrizione del web service

* lo scopo del web service è quello che data una chiave e il nome dell'azienda in borsa (microsoft -> msft), ti restituisce un json con scritto l'ora e il valore numerico dell'azione convertito poi con l'aiuto di JAXB e un file java guida (istruzioni su come fare il marshall dell'JSON a XML)
* la funzionalità è una: consegnarti un file json con scritto ora e azione della determinata ora,

come parametri di input abbiamo "nome azienda", "chiave di accesso" e "intervallo di

tempo per il conteggio"

* un'esempio di richiesta chiedendo le ultime 72 ore dell'azienda IBM in formato json con intervallo di 60 minuti

https://www.alphavantage.co/query?function=TIME\_SERIES\_INTRADAY&symbol=<nome\_azienda\_in\_borsa>&interval=<minuti\_per\_intervallo>min&apikey=<chiave\_api> "

-- RISULTATO RICHIESTA -->

{

"Meta Data": {

"1. Information": "Intraday (60min) open, high, low, close prices and volume",

"2. Symbol": "IBM",

"3. Last Refreshed": "2024-03-01 19:00:00",

"4. Interval": "60min",

"5. Output Size": "Compact",

"6. Time Zone": "US/Eastern"

},

"Time Series (60min)": {

"2024-03-01 19:00:00": {

"1. open": "188.2000",

"2. high": "188.3000",

"3. low": "188.0000",

"4. close": "188.2000",

"5. volume": "516999"

},

"2024-03-01 18:00:00": {

"1. open": "188.2000",

"2. high": "188.2900",

"3. low": "188.0000",

"4. close": "188.0500",

"5. volume": "516726"

},

**.…**

}

…

1. Requisiti progetto

* funzionante
* la sicurezza è poca, una chiave di riserva è contenuta dentro il programma
* nessuna integrazione, il servizio è solo 1

1. Implementazione

* abbiamo usato java e JAXB
* abbiamo usato swing e jaxbc
* abbiamo iniziato facendo tutto in parallelo e correggendo dopo, scrittura in termine

casuale dove era necessario

* la struttura e divisa nel modello MWCMO (model, view, controller, main\_application, other)
* abbiamo scelto una struttura per preferire la modularità e la semplicità

1. Test e validazione

* Per provare la nostra app abbiamo prima verificato che non ci fossero problemi nel codice, per poi provare con un "fuzz testing" tutte le variabili ed opzioni del caso e vedendo se il programma si rompeva
* i risultati sono che l'applicazione non funzionava

1. Deployment e gestione del progetto