

Contents

1	Einleitung	3
1.1	Wer wir sind	3
2	Netzwerkprotokolle	3
3	ISO/OSI Modell	3
4	Transport-Ebene	4
4.1	TCP-Protokoll	4
4.1.1	Wie funktioniert das?	4
4.2	UDP-Protokoll	4
4.2.1	Programme, die UDP benutzen	4
5	Client-Server-Anwendung	5
5.1	Was ist	5
6	DNS	5
6.1	Wozu dient der DNS?	6
7	HTTP	6
7.1	Wie HTTP Request und Response funktioniert:	6
8	CSV	6
8.1	Wie lese ich eine CSV-Datei?	6
8.2	Wie kann ich eine CSV-Datei schreiben?	7
9	API	7
10	Bot-Telegramm	7
10.1	Erstellen	7
10.2	Klass	7
10.3	Vertiefung	8
11	JSON	8
11.1	Was ist JSON	8
11.2	Wie JSON funktioniert	8
12	Programmiersprache Java	8
12.1	Vorteile und Nachteile	8
13	Python-Programmiersprache	9
13.1	Vor- und Nachteile	9
14	L^AT_EX	9
14.1	Was ist L ^A T _E X?	9
14.2	Wie wir L ^A T _E X verwendet haben	9

FinanceBot

L^AT_EX

PROJEKTGEBIET *Deutsche Sprache*

Klasse 4J

1 Einleitung

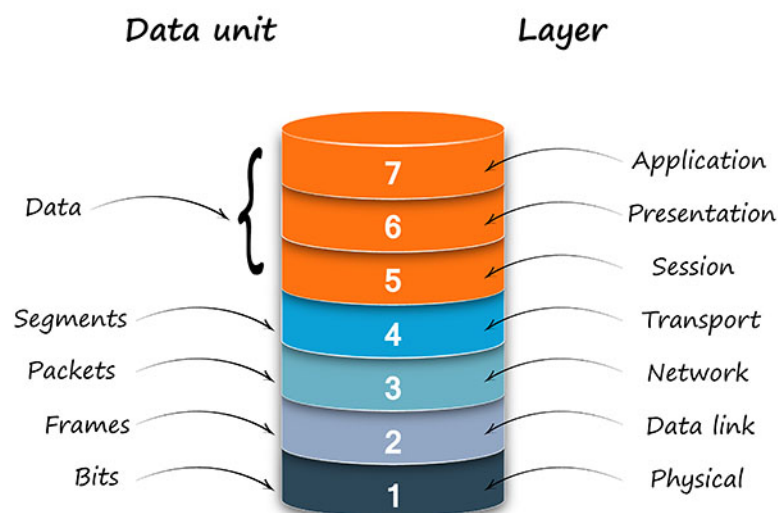
1.1 Wer wir sind

Wir sind eine Klasse von einem Computer-Adressing-Technik-Institut. Gemeinsam haben wir ein Programm entwickelt, das NASDAQ-Wertpapiere über Yahoo Finance API überwacht. Die verarbeiteten Daten werden dann von einem Telegramm-Bot verarbeitet.

2 Netzwerkprotokolle

Netzwerkprotokolle umfassen 5 Protokolle, die wichtigsten sind IP-, ARP- und ICMP-Protokolle. Die Übertragungsebene ermöglicht es Anwendungen, die auf entfernten Terminals laufen, zu kommunizieren. Es enthält zwei Protokolle, die es den beiden Anwendungen ermöglichen, Daten unabhängig von den unteren Ebenen auszutauschen. Internet Protocol (IP) ist ein Netzwerkprotokoll, das sich mit Adressierung/Weiterleitung beschäftigt und zur TCP/IP-Internetprotokoll-Suite gehört, auf der das Internet-Netzwerk basiert. Moderne Protokolle für Computernetzwerke verwenden im Allgemeinen die Paketwitching-Technologie, um Nachrichten in Form von Paketen zu senden und zu empfangen, die am Bestimmungsort gesammelt und wieder zusammengebaut werden.

3 ISO/OSI Modell



OSI model

Figure 1: modello ISO/OSI

4 Transport-Ebene

4.1 TCP-Protokoll

Das tcp-Protokoll (transmission control protocol) ist eine standardisierte Vereinbarung zur Datenübertragung zwischen verschiedenen Nutzern eines Computernetzwerks. Der heutige Entwicklungsstand des TCP-Protokolls ermöglicht es zwei Endpunkten innerhalb eines gemeinsamen Computernetzwerks, eine Verbindung herzustellen, über die Daten in beide Richtungen übertragen werden können. Im Rahmen dieser Verbindung werden eventuelle Datenverluste automatisch erkannt und korrigiert; daher wird das TCP auch als zuverlässiges Protokoll bezeichnet. In der Familie der Internet-Protokolle bildet das TCP zusammen mit UDP und SCTP die Gruppe der Transportprotokolle, die auf der Grundlage des OSI-Modells in der Netzwerkarchitektur auf Transportebene klassifiziert werden. Da das TCP-Protokoll in fast allen Fällen auf dem Internet-Protokoll (IP) basiert und diese Kombination die Grundlage für die meisten öffentlichen und lokalen Netzwerke und Netzwerkdienste darstellt, spricht man oft auch von einem TCP/IP-Protokollstapel und meint damit die Familie der Internet-Protokolle im weiteren Sinne.

TCP-Header				
Bits	0-15			16-31
0	Quell-Port			Ziel-Port
32	Sequenznummer			
64	Bestätigungsnummer			
96	Offset	Reserviert	Flags	Fenster-/Empfangsgröße
128	Prüfsumme			Urgent-Pointer
160	Optionen			

Figure 2: TCP

4.1.1 Wie funktioniert das?

TCP ermöglicht die Übertragung von Informationen in beide Richtungen. IT-Systeme, die über TCP kommunizieren, können daher Daten gleichzeitig senden und empfangen, genau wie bei einem Telefongespräch. Die wichtigsten Übertragungseinheiten des Protokolls sind Segmente (Pakete), die neben der Nutzlast (d. h. der tatsächlichen Nachricht) auch Steuerinformationen enthalten können und auf eine Grösse von 1500 Bytes beschränkt sind. Der Aufbau und das Unterbrechen von Verbindungen, die als Punkt-zu-Punkt-Verbindungen klassifiziert werden können, sowie die eigentliche Datenübertragung werden von der TCP-Software im Netzwerkprotokollstapel des jeweiligen Betriebssystems erfasst.

4.2 UDP-Protokoll

4.2.1 Programme, die UDP benutzen

Netzwerkanwendungen, die einen zuverlässigen Datentransfer benötigen, verlassen sich natürlich nicht auf UDP, während Anwendungen, die in Bezug auf Datenverluste elastisch und streng zeitabhängig sind, sich auf UDP verlassen. Ausserdem wird UDP für Broadcast-Kommunikation (Senden an alle Endgeräte eines lokalen Netzwerks) und Multicast-Kommunikation (Senden an alle Endgeräte eines Dienstes) verwendet.

Anwendung	Protokoll Status Anwendung	Transportprotokoll Schicht
E-Mail	SMTP	TCP
Remote-Terminal-Zugang	telnet	TCP
Dateiübertragung	FTP	TCP
Web	HTTP	TCP
SAudio-/Video-Treaming	RTSP/RTP	TCP (Betätigungseinrichtungen) + UDP (Strömung)
Remote-Dateiserver	NFS	typischerweise UDP
Telefonie im Internet (VoIP)	SIP, H.323, Sonstige	typischerweise UDP
Netzverwaltung	SNMP	typischerweise UDP
Routing-Protokoll	RIP	typischerweise UDP
Namensauflösung	DNS	typischerweise UDP

Table 1: Liste der wichtigsten Internet-Dienste und der von ihnen verwendeten Protokolle

5 Client-Server-Anwendung

5.1 Was ist

Eine Client-/Server-Anwendung ist eine verteilte Anwendung, die in zwei Teile unterteilt ist: eine Server-Anwendung (Backend), die Dienste anbietet, und eine Client-Anwendung (Frontend), die die Benutzeroberfläche verwaltet und die Serverdienste anfordert.

Die Client- und Server-Komponenten der Anwendung befinden sich in der Regel auf zwei verschiedenen Stationen (die Server-Komponente befindet sich in der Regel auf einem viel leistungsfähigeren Computer), können aber auch auf derselben Station liegen.

Die Client/Server-Architektur kann auch auf drei Ebenen aufgebaut werden: eine Ebene für die Verwaltung der Benutzeroberfläche, eine Ebene für die Verarbeitung und eine Ebene für die Verwaltung der Informationen (normalerweise in einer Datenbank gespeichert).

Es ist ein unsymmetrisches Modell, das auch Request/Replay genannt wird, weil es auf einer Abfolge von Anfragen und Antworten basiert.

Der Server kann so ausgelegt werden, dass er jeweils nur eine Anfrage oder mehrere Anfragen gleichzeitig erfüllt.

Der Client, um den Server zu kontaktieren, muss seine Adresse kennen; die Adresse kann von vornherein bekannt sein, es kann einen Prozess-Server geben, an den der Client Anfragen stellt, die dann an den gewünschten Server weitergeleitet werden, oder es kann einen Verzeichnis-Server geben, der basierend auf dem Namen des gewünschten Servers die Adresse des Servers zurückgibt.

Die gebräuchlichste Technik zur Realisierung von Client/Server-Anwendungen ist die Verwendung von TCP/IP-Sockets.

6 DNS

DNS ist die Abkürzung für Domain Name System und, um es kurz zu erklären, es ist das System, das IP-Adressen in Domainnamen übersetzt. Wenn wir im Internet surfen, geben wir in der Adresszeile des Browsers die Domain einer Website ein, zum Beispiel sos-wp.it.

Was auf der technischen Ebene geschieht, ist folgendes:

- der Browser übernimmt die von Ihnen eingegebene Domain.
- Diese Domain ist mit einer IP-Adresse verknüpft.
- Die IP-Adresse ist eine Reihe von Zahlen, die einen bestimmten Server kennzeichnen, d.h. einen Computer, auf dem sich die Webseite befindet.
- DNS übersetzt den Domainnamen in die IP-Adresse und umgekehrt.



Figure 3: DNS

6.1 Wozu dient der DNS?

DNS dienen dazu, den Namen (Domain) in die Nummer (IP-Adresse) zu übersetzen, wie z.B. das Telefonbuch eines Telefons. Anstatt also die gesamte IP-Adresse einzugeben, geben Sie einfach den Namen der Person ein, die Sie anrufen möchten. Ein weiteres Beispiel kann eine Website sein, anstatt die gesamte IP-Adresse geben Sie einfach den Namen der Website, wie Amazon.com sein kann.

7 HTTP

HTTP (Hypertext Transfer Protocol) ist eine Textsprache, die die Kommunikation zwischen Client und Server über das Internet ermöglicht. Jedes Mal, wenn wir eine Website besuchen, ermöglicht uns HTTP, die Ressourcen dieser Website anzuzeigen.

7.1 Wie HTTP Request und Response funktioniert:

Nach einer kurzen Einführung des HTTP-Protokolls sehen wir uns nun die Funktionsweise des HTTP-Protokolls Schritt für Schritt an, indem wir analysieren, wie eine HTTP-Anfrage gestellt wird und wie der Server darauf reagiert. Request Der Client sendet eine Anfrage: Jede Konversation zwischen Client und Server im Web beginnt mit einer Anfrage, einer Textnachricht, die vom Client in einem speziellen Format namens HTTP erstellt wird.

Eine HTTP-Anfrage besteht aus den folgenden Teilen:

[Methode] [URI] [Version]

[Kopfzeilen]

[Körper]

Die erste Zeile einer HTTP-Anfrage ist die wichtigste und besteht aus zwei Hauptteilen:

1. die HTTP-Methode (GET)
2. die URL (/blog/protokoll-http-was-und-wie-funktioniert)

HTTP-Methoden definieren, wie der Client mit der Ressource interagieren kann. Die folgende Tabelle zeigt einige der Methoden, die während einer HTTP-Anfrage verwendet werden.

8 CSV

Eine CSV-Datei (Comma Separated Values) gehört zu den gängigsten Formaten zum Speichern von Tabellendaten und wird hauptsächlich verwendet, um grosse Datenmengen zu verwalten. Die csv-Elemente werden üblicherweise durch ein Komma getrennt. Normalerweise werden in der ersten Zeile die Spalten unserer Tabelle dargestellt, während in den folgenden Zeilen die Werte dargestellt werden.

8.1 Wie lese ich eine CSV-Datei?

Die häufigste Praxis für die Interaktion mit einer csv-Datei in Python ist die Verwendung der gleichnamigen Bibliothek. Beginnen wir mit dem Importieren in unser Skript: `import csv`. Um den Inhalt unserer Datei zu lesen, können wir die `csv.reader()`-Methode verwenden.

Falls unsere Datei.csv ein anderes Zeichen als Komma als Trennzeichen verwendet, müssen wir dies angeben, indem wir einen zweiten Parameter an die Methode csv.reader () übergeben. Dieser Parameter heisst Trennzeichen und wird standardmässig mit einem Komma gesetzt. Auch durch Aufruf der csv-Methode.DictReader () anstelle von csv.reader () erhalten wir den Inhalt von csv in einem Wörterbuch.

8.2 Wie kann ich eine CSV-Datei schreiben?

- Um eine CSV-Datei in Python zu schreiben, können Sie die write_row () -Methode des Writer-Objekts verwenden, die durch Aufruf der csv.writer () -Methode wiederhergestellt wird.
- Um das Ergebnis zu sehen, müssen Sie das Skript wieder aufnehmen, um die Datei zu lesen, den csv-Namen mit dem, was wir erzeugt haben, ändern und es ausführen.
- Wie beim Lesen kann auch beim Schreiben das Begrenzungszeichen in der csv.writer-Methode definiert werden (file, delimiter=>>).
- Die csv-Methode wird benutzt.DictWriter () kann auf diese Weise ein csv aus einem Wörterbuch erzeugen.
- Beachten Sie, dass Sie in diesem Fall die Spalten definieren müssen, die Sie verwenden möchten.

9 API

```
# utilizzo del API
querystring = {"region":"US","symbols":titolo[i]}
response = requests.request("GET", url, headers=headers, params=querystring)
status_code = response.status_code

correctJson = removeJson(response.text)
writeToFile(path, correctJson)

with open(path, "r") as json_file:
```

Figure 4: API

API ist die Abkürzung für Anwendungsprogrammierschnittstelle (Application Programming Interface), d.h. eine Sammlung von Definitionen und Protokollen für die Erstellung und Integration von Anwendungssoftware. Intuitiv sind sie daher die Grundlage für eine Vielzahl von Anwendungen, Dienstleistungen, Websites, Geräte. APIs wurden mit der Geburt der Informatik geboren, lange vor den PCs.

10 Bot-Telegramm

10.1 Erstellen

Der Bot wurde mit Hilfe der Programmierumgebung «IntelliJ» in Java geschrieben. Enthält 3 Klassen von Objekten und eine XML-Datei. Um einen Telegram Bot zu erstellen, müssen Sie den Bot namens «BotFather» verwenden, der eine API gibt, um das Programm mit Telegram zu synchronisieren.

10.2 Klass

- Main: Ermöglicht das Starten des Bots und den Betrieb des gesamten Programms.
- SimpleBot: Gibt die Telegramm-Ausgabe für den gewünschten Befehl zurück.
- LockFile: Erstellt und löscht die Lock-Dateien.

10.3 Vertiefung

Wir haben uns für die Verwendung von .lock-Dateien entschieden, weil sie verhindern, dass das Python- und das Java-Programm gleichzeitig an derselben Datei arbeiten. Diese Funktion hätte ersetzt werden können, wenn wir uns entschieden hätten, Sockets zu verwenden, um die beiden Programme zu synchronisieren.

Tutte le classi sono immagazzinate in un apposito file .jar per permettere a tutto il programma di essere utilizzato e con la possibilità di essere avviato anche da terminale.

11 JSON

11.1 Was ist JSON

JSON ist ein textbasiertes Serialisierungsformat für den Datenaustausch, vor allem zwischen Server und Webanwendung. JSON steht für JavaScript Object Notation und verwendet die Dateiendung .json. JSON ist ein Konkurrent von XML, hat aber eine einfachere und kompaktere Syntax als sein Konkurrent.

11.2 Wie JSON funktioniert

1. Die Daten werden als Schlüssel-Wert-Paar geschrieben.
2. Die Daten werden durch Komma getrennt.
3. Klammern enthalten Objekte.
4. Die eckigen Klammern enthalten Arrays.
5. Zeichenketten müssen mit doppelten Spitzen umschlossen werden.

12 Programmiersprache Java

Java ist eine hochqualifizierte, klassen- und methodenbasierte und objektorientierte Programmiersprache. Daten werden als Objekte und Operationen als Methoden dargestellt, die auf sie wirken. Diese Programmiersprache wurde für die Entwicklung von Anwendungen in Netzwerken entwickelt und so unabhängig wie möglich von der Hardware-Plattform laufen.

12.1 Vorteile und Nachteile

Vorteile:

- Multi-Plattform-Kompatibilität (dank Virtual Machine)
- hohe Abstraktion von der physikalischen Maschine
- Entwicklungsgeschwindigkeit
- grosse Verfügbarkeit von Bibliotheken
- Kompatibilität mit Palm- und Embedded-Systemen
- hohe Integration mit dem Web

Nachteile:

- Langsamkeit (dank Virtual Machine)
- hohe Abstraktion von der physikalischen Maschine (die es nicht erlaubt, alles zu tun, was man will)
- ist dekompilebar.

13 Python-Programmiersprache

Python ist eine übergeordnete Programmiersprache als die meisten anderen Sprachen, auch sie ist objektbasiert. Python eignet sich unter anderem für die Entwicklung von verteilten Anwendungen, Skripting, numerischen Berechnungen, Systemtests, Websites, Software usw. Darüber hinaus sind viele Python-Bibliotheken für Bildung und Wissenschaft verfügbar.

13.1 Vor- und Nachteile

Einer der grössten Nachteile ist sicherlich das Fehlen von Typen, also ein Mangel an statischer Kontrolle. Ausserdem ist der Python-Compiler nicht in der Lage, viele Fehler und mangelnde Kontrolle über die von den Funktionen zurückgegebenen Werte zu finden. In der Tat sind wir gezwungen, alle Fälle zu durchsuchen, und oft sind die Codes, mit denen Fehler behandelt werden, schwer zu erreichen. Einer der grössten Vorteile ist die einfache und saubere Syntax, die den Code sofort sehr klar macht, und dank der Unterstützung der Python Software Foundation nutzt es eine sehr grosse eingebaute Bibliothek, die es zu einer der reichhaltigsten und praktischsten Programmiersprachen macht. Python kann als eine Pseudo-komplizierte und portable Sprache bezeichnet werden, da ein Interpreter den Quellcode analysieren und ihn dann ausführen kann, wenn er es für richtig hält, da es keine separate Kompilierphase gibt, aus der eine ausführbare Datei erzeugt werden kann.

14 L^AT_EX

14.1 Was ist L^AT_EX?

L^AT_EX ist eine Programmiersprache, die 1985 von Leslie Lamport entwickelt wurde. Die neueste Version stammt aus dem Jahr 2011. L^AT_EX ist eine Auszeichnungssprache, in diesem Fall ist die am häufigsten verwendete Markierung der (Backslash). L^AT_EX wird hauptsächlich zur Erstellung von PDF-Dokumenten verwendet. Diese Sprache zur Erzeugung von PDF-Dateien muss kompiliert werden.

14.2 Wie wir L^AT_EX verwendet haben

Um unser Dokument zu erstellen, haben wir einen Editor verwendet und das Dokument von Grund auf neu erstellt. Wir haben dann einen Index und Kapitel und Unterkapitel hinzugefügt, um das PDF übersichtlicher zu gestalten. Wir haben Optionen wie den Index hinzugefügt und um alles gut zu verteilen, haben wir uns für die Verwendung von Kapiteln und Unterkapiteln entschieden. Für die Weiterentwicklung des Codes haben wir einige Pakete verwendet, z. B. die für Sonderzeichen, die die Möglichkeit beinhalten, das System Sonderzeichen wie Buchstaben mit Akzent erkennen zu lassen.