ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Διπλωματική Εργασία

Price Monitoring με Web Scraping

Του φοιτητή:
Βασιλούδη Γεώργιου
Αρ. Μητρώου: 154603

Επιβλέπων:
Μιχαήλ Σαλαμπάσης
Βαθμίδα: Καθηγητής

Θεσσαλονίκη, Φεβρουάριος 2022

Τίτλος Δ.Ε. : Price Monitoring με Web Scraping
Κωδικός Δ.Ε. 21201
Ονοματεπώνυμο φοιτητή : Βασιλούδης Γεώργιος
Ονοματεπώνυμο εισηγητή : Μιχαήλ Σαλαμπασης
Ημερομηνία ανάληψης Δ.Ε. : 18-03-2021
Ημερομηνία περάτωσης Δ.Ε. : 01-02-2022

Βεβαιώνω πως είμαι ο συγγραφέας αυτής της εργασίας καθώς και κάθε βοήθεια που αξιοποιήθηκε για την εκπλήρωση της, είναι πλήρως αναγνωρισμένη.Η κάθε πηγή που χρησιμοποιήθηκε για την άντληση πληροφοριών, είτε αυτές είναι κείμενο, εικόνες είτε σχήματα αναφέρονται ακριβώς. Τέλος, η συγκεκριμένη εργασία μου ανατέθηκε ώς διπλωματική εργασία του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου Ελλάδος

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Αφιέρωση

Αφιερώνω τη διπλωματική εργασία στους γονείς μου και σε όσους ήταν κοντά μου, καθόλη τη διάρκεια των σπουδών μου, για την ηθική και οικονομική υποστήριξη που μου παρείχαν. Τους είμαι για πάντα ευγνώμων.

Πρόλογος

Με την περάτωση της παρούσας Δ.Ε. με τίτλο “Price Monitoring με Web Scraping” σηματοδοτείται και το τέλος των σπουδών μου στο τμήμα των μηχανικών πληροφορικής και ηλεκτρονικών συστημάτων του ΔΙ.ΠΑ.Ε Θεσσαλονίκης. Το θέμα της Δ.Ε. αποτελεί, στη σύγχρονη εποχή, μια δραστηριότητα δημοφιλή και συχνά ιδιαίτερα επικερδή για όσους ασχολούνται με αυτη. Είναι μια εύκολη και χρήσιμη μέθοδος που στοχεύει στην απόκτηση και συλλογή δεδομένων από ιστοσελίδες στο διαδίκτυο. Δεν είχα καμία γνώση πάνω στο αντικείμενο, και έτσι, επέλεξα αυτό το θέμα διότι, πέρα απ’το οτι μου φάνηκε ενδιαφέρον, είναι και επίκαιρο καθώς χρησιμοποιείται συχνά σε πολλές εταιρείες πληροφορικής. Μελλοντικά, ενδέχεται να επεκταθεί η συγκεκριμένη εργασία για την άντληση παραπάνω δεδομένων, προσφέροντας ακόμα μεγαλύτερη συλλογή πληροφοριών.

Περίληψη

Με την ανάπτυξη της τεχνολογίας δημιουργήθηκε ένα πλήθος εργαλείων με τα οποία συνεργάζεται το κάθε κατάλυμα, με σκοπό να προωθηθούν και να αναδειχθούν μέσω της σύγκρισης τιμών και υπηρεσιών. Η τιμή της διανυκτέρευσης ενός δωματίου είναι η πρώτη και η βασικότερη πληροφορία που λαμβάνει ένας χρήστης, ο οποίος αναζητά κατάλυμα για τη διαμονή του. Στόχος της Δ.Ε. είναι η άντληση συγκεκριμένων πληροφοριών, μέσω της πλατφόρμας booking.com, με σκοπό την παροχή τους σε ιδιοκτήτες καταλυμάτων. Οι ιδιοκτήτες θα λαμβάνουν μια καθημερινή ενημέρωση τιμών και υπηρεσιών που προσφέρουν τα γειτονικά καταλύματα της περιοχής τους, καθώς και τη σύγκριση τιμών τους. Πιο συγκεκριμένα, μέσω των κατάλληλων βιβλιοθηκών, θα γίνει η σωστή άντληση, επεξεργασία και αποθήκευση της πληροφορίας σε μια βάση δεδομένων. Μετά από αυτό το στάδιο, τα ίδια δεδομένα είναι εφικτό να εξαχθούν σε διάφορες μορφές. Το Project υλοποιείται στο περιβάλλον Visual Studio Code με χρήση της γλώσσας προγραμματισμού Python. Η βάση δεδομένων που χρησιμοποιήθηκε είναι η MongoDB η οποία είναι ανοιχτή και διαθέσιμη προς όλους τους χρήστες. Έγινε χρήση του GitHub για την αποθήκευση του project online και για την αξιοποίηση κάποιων εργαλείων του, που θα αναφερθούν παρακάτω. Τέλος, η παρούσα Δ.Ε. αποτελεί συνεργατική εργασία καθώς, αφού ολοκληρώθηκε, παραδόθηκε στον συμφοιτητή Ευάγγελο Γιατσίδη για περαιτέρω ανάπτυξη, στο κομμάτι του Price Monitoring.

Η Δ.Ε. απευθύνεται σε όσους θέλουν να αποκτήσουν κάποια γνώση πάνω στο W.S. Θα υπάρξουν αναλυτικά βήματα τόσο για το θεωρητικό όσο και για το πρακτικό κομμάτι της εργασίας. Επίσης θα αναφερθούν όλες οι έννοιες και οι τεχνικές καθώς και θα περιγραφούν λεπτομερώς όλοι οι μέθοδοι που χρησιμοποιήθηκαν για την επίλυση του project.

Abstract

With the development of technology, a variety of tools have been created with which each hotel collaborates in order to promote and stand out through the comparison of prices and services. The price of one night's room is the first and most basic information received by a user, who is looking for accommodation for his stay. The aim of the dissertation is to obtain specific information, through the booking.com platform, in order to provide it to accommodation owners. The owners will receive daily updates on prices and services offered by neighboring accommodations in their area, as well as price comparisons. More specifically, through the appropriate libraries, the correct extraction, processing and storage of information in a database will take place.After this step, the same data can be exported in different formats. The Project is being implemented in the environment Visual Studio Code using the programming language Python . The database used is MongoDB which is an open-source and accessible to every user. GitHub was used to store the project online and to use some of its tools that will be mentioned below. Finally, the present dissertation is a collaborative work since, after completion, it was handed over to fellow student Evangelos Giatsidis for further development, in the part of Price Monitoring.

The dissertation is addressed to those who want to gain some knowledge on Web Scraping. There will be detailed steps for both the theoretical and the practical part of the work. Also, all the concepts and techniques will be mentioned as well as all the methods used to solve the project will be described in detail.

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον κ. Μιχαήλ Σαλαμπάση για την καθοδήγηση και τις συμβουλές του καθ' όλη την διάρκεια της διπλωματικής εργασίας, οι οποίες οδήγησαν στην επιτυχή ολοκλήρωση της εργασίας.
Ένα ακόμα ευχαριστώ στους φίλους μου, που πέρα από την στήριξη, φρόντισαν να μου μείνουν αξέχαστα τα φοιτητικά μου χρόνια.
Και τέλος, θα ήθελα να δώσω το μεγαλύτερο ευχαριστώ στην οικογένεια μου, που δε σταμάτησε να με στηρίζει καθόλη τη διάρκεια των σπουδών μου.

Περιεχόμενα

Πρόλογος **3**

Περίληψη **4**

Abstract **5**

Ευχαριστίες **6**

Περιεχόμενα **7**

Κατάλογος Σχημάτων **10**

Συντομογραφίες **11**

Κεφάλαιο 1ο: Web Scraping **12**

Εισαγωγή 12

Τι είναι το Web Scraping; 12

Γιατί να χρησιμοποιήσω το Web Scraping; **13**

Είναι το Web Scraping Νόμιμο; 14

Ποιός χρησιμοποιεί το Web Scraping; 15

Web Scraping vs Web Crawling 17

Web Scraping 17

Web Crawling 17

Διαφορά Web Scraping με Web Crawling **18**

Γιατί να χρησιμοποιήσεις την Python στο Web Scraping; 19

Γιατί Python 3 και οχι 2; 20

Python 2 20

Python 3 21

Python 2 vs Python 3 : Σύγκριση 21

Συμπέρασμα 22

Κεφάλαιο 2ο: MongoDB **23**

Εισαγωγή 23

Ανασκόπηση της φιλοσοφίας της MongoDB 23

Γιατι να χρησιμοποιήσει κανείς τη MongoDB; 24

Κεφάλαιο 3ο: GitHub **25**

Εισαγωγή 25

Τι είναι το git; 25

Τι είναι το GitHub; 25

Γιατί να χρησιμοποιήσει κανείς το GitHub 26
Βασικές εντολές του Git 26
GitHub Actions 27
Τι είναι τα GitHub Actions; 28
Τα δύο είδη των GitHub Actions 28
Τι είναι ένα GitHub Action workflow; 28

Κεφάλαιο 4ο: Υλοποίηση του Project **30**
Εισαγωγή 30
Επαλήθευση της διαδρομής της Python στα Windows 31
Εγκατάσταση του pip 32
Βημα 1 : Beautifulsoup και Requests 34
Βήμα 2 : Σύνδεση με τη βάση δεδομένων 35
Βήμα 3 : Δημιουργία του βασικού link 35
Βήμα 4 : Συλλογή συνδέσμου του κάθε καταλύματος 37
Βήμα 5 : Scraping των δεδομένων 40
Βήμα 6 : Κατανομή των δεδομένων στη βάση 45
Βήμα 7 : Εξαγωγή των δεδομένων 50
Βήμα 8 : Προσθήκη του GitHub Action 51
Δημιουργία του workflow 52
Προβολή των αποτελεσμάτων του workflow 52

Κεφάλαιο 5ο: Συμπέρασμα και προτάσεις βελτίωσης **58**
Συμπεράσματα 58
Προτάσεις βελτίωσης 59

Βιβλιογραφία **60**

Κατάλογος Σχημάτων

1. Web scraping
2. Web scraping vs Web crawling
3. Scraper vs Crawler
4. Python 2 vs Python 3
5. Παράδειγμα yml αρχείου
6. Εκτέλεση push/pull requests μόνο για το master branch
7. Προγραμματισμένο workflow
8. Σύνταξη cron
9. Προσθήκη της Python στα environment variables
10. pip is not recognized error
11. pip installed successfully
12. Προσθήκη του pip στα environment variables
13. pip install bs4
14. Σύνδεση με MongoDB
15. basic_datas document
16. Δημιουργία του βασικού link
17. Στιγμιότυπο του βασικού link στο booking.com
18. Στιγμιότυπο του inspect element του βασικού link
19. δεδομένα για scraping
20. πίνακας ενός καταλύματος στο booking.com
21. δεδομένα για scraping 2
22. while loop
23. Κώδικας για scraping δεδομένων
24. Inspect element της ιστοσελίδας για το hotel_id
25. Στιγμιότυπα κώδικα για το scraping των reviews
26. Στιγμιότυπο booking.com/inspect element σελίδας
27. JSON μοντέλο βάσης
28. Στιγμιότυπο του hotels collection στη MongoDB
29. document ενός καταλύματος
30. Κώδικας για εξαγωγή σε csv
31. Εγγραφο csv μιας ημέρας
32. Δημιουργία του workflow
33. Επιλογή του Actions
34. Επιλογή του επιθυμητού workflow
35. Λίστα με όλα τα workflows
36. Κλικ του workflow στα jobs
37. Βήματα του workflow
38. Σφάλμα στην εκτέλεση του workflow
39. yml αρχείο
40. requirements.txt

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
W.S.	Web Scraping

Κεφάλαιο 1ο: Web Scraping

Εισαγωγή

Κάθε σημαντικό Project προγραμματισμού επικεντρώνεται στα δεδομένα. Για την συντριπτική πλειοψηφία δεδομένων των προβλημάτων προγραμματισμού, ο προγραμματιστής θα χρειαστεί να χειριστεί τα δεδομένα ετσι ώστε να φτάσει στη λύση. Ετσι, προκειμένου να καταπολεμηθεί αυτή η 'διαμάχη' δεδομένων θα πρέπει να απαντηθούν δύο ερωτήσεις. Απο που θα παρθούν τα δεδομένα τα οποία χρειάζεται το πρόγραμμά σου; Και πώς θα παρθούν αυτα τα δεδομένα, από την πηγή τους, μέσα στο πρόγραμμά; Αν η απάντηση στο πρώτο ερώτημα είναι "από μία ιστοσελίδα", τότε μέχρι το τέλος της Δ.Ε. θα μπορεί να απαντηθεί και το δεύτερο ερώτημα. Πολλές φορές, τα δεδομένα μπορεί να δοθούν έτοιμα, σε κάποιο αρχείο ή σε κάποια βάση δεδομένων. Από την άλλη είναι πολύ πιθανό, ο προγραμματιστής να χρειαστεί να βρεί τα δεδομένα μόνος του. Ευτυχώς, με την εξέλιξη της τεχνολογίας, την άνοδο του διαδικτύου και τον κόσμο που βασίζεται όλο και περισσότερο στα δεδομένα, τα δεδομένα που χρειάζονται μπορούν να συλλεχθούν εύκολα διαδικτυακά. Δυστυχώς, τα άτομα που κατέχουν αυτά τα δεδομένα, δεν είχαν υπόψιν τους, τον κάθε προγραμματιστή, όταν αποφάσιζαν ποιός είναι ο καλύτερος τρόπος εμφάνισης τους. Τις περισσότερες φορές, οι πληροφορίες που είναι ορατές στο διαδίκτυο έχουν διαμορφωθεί έτσι ώστε, αφού υποβληθούν σε επεξεργασία και εμφανιστούν σε ένα πρόγραμμα περιήγησης ιστού, να μπορούν να διαβαστούν και να κατανοηθούν εύκολα από τους ανθρώπινους χρήστες. Το κάθε πρόγραμμά όμως, για να αποκτήσει τα δεδομένα που χρειάζεται από μια ιστοσελίδα, θα χρειαστεί όχι μόνο να κατανοήσει τη δομή της σελίδας, αλλά θα πρέπει να καταλάβει αρκετά σχετικά με το περιεχόμενό της, για να καθορίσει ποιες πληροφορίες στη σελίδα υποτίθεται ότι θα 'διαβάσει'. Αυτό δεν αποτελεί πάντα, μια εύκολη δουλειά.
Σε αυτό το κεφάλαιο θα εξηγηθεί τι είναι το W.S., γιατί να το χρησιμοποιήσει κανείς, αν είναι νόμιμο, καθώς και παραδείγματα χρήσης του. Επίσης θα τονιστούν οι διαφορές ενός scraper από έναν crawler, θα εξηγηθεί γιατί είναι ιδανική η επιλογή της Python σαν γλωσσα προγραμματισμου πάνω στο W.S., και τέλος μια σύγκριση της Python3 με την Python2.

Τι είναι το Web Scraping;

Το W.S. (επίσης ορίζεται και ως Screen Scraping, Web Data Extraction, Web Harvesting κ.α.) είναι μία τεχνική που χρησιμοποιείται για να εξάγει μεγάλα πλήθη δεδομένων από ιστοσελίδες, τα οποία αποθηκεύονται σε τοπικά αρχεία στον υπολογιστή ή σε μια βάση δεδομένων.
Τα δεδομένα που εμφανίζονται από τις περισσότερες ιστοσελίδες μπορούν να προβληθούν μόνο μέσω ενός περιηγητή Ιστού (web browser). Για παράδειγμα σε κοινωνικά δίκτυα (social networks), σε διαδικτυακά καταστήματα αγορών, βάσεις δεδομένων επικοινωνίας κ.λ.π. Οι περισσότερες ιστοσελίδες δεν προσφέρουν κάποια λειτουργικότητα για να αποθηκεύσει και να προβάλει κανείς τα

δεδομένα στον υπολογιστή του. Αρα η μόνη λύση θα ήταν να κάνει αντιγραφή και επικόλληση των δεδομένων που χρειάζεται από την ιστοσελίδα σε ένα αρχείο τοπικά στον υπολογιστή σου. Σίγουρα αυτό αποτελεί μια κουραστική δουλειά που θα πάρει ώρες, μέχρι και μέρες για να ολοκληρωθεί.
Το W.S. είναι η τεχνική αυτοματοποίησης αυτής της διαδικασίας, έτσι ώστε αντί να γίνεται αντιγραφή των δεδομένων από ιστοσελίδες χειροκίνητα, το λογισμικό του W.S. θα εκτελέσει την ίδια εργασία μέσα σε κλάσματα του χρόνου.
Το W.S. αλληλεπιδρά με τις ιστοσελίδες με τον ίδιο τρόπο που αλληλεπιδρά ένας φυλλομετρητής με τις ιστοσελίδες. Απλά αντί να προβάλλει τα δεδομένα που εξυπηρετούνται από τον ιστότοπο στην οθόνη, το λογισμικό του W.S. αποθηκεύει τα απαιτούμενα δεδομένα από μια ιστοσελίδα σε ένα τοπικό αρχείο ή σε μια βάση δεδομένων. Η αποθήκευση εξαρτάται από τον προγραμματιστή στό που θα εξάγει τα δεδομένα.

Figure 1 Web Scraping

Γιατί να χρησιμοποιήσει κανείς το Web Scraping;

Οι πιθανοί λόγοι για να χρησιμοποιήσει κάποιος την τεχνική του W.S. είναι ατελείωτοι. Η γενική απάντηση σε αυτό το ερώτημα είναι η εξής: Το W.S. χρησιμοποιείται οποτεδήποτε χρειάζεται ο χρήστης, μια μηχανή για να έχει πρόσβαση σε πληροφορίες που είναι άμεσα διαθέσιμες στον ιστό αλλα δεν έχουν παρουσιαστεί με τρόπο που να είναι εύκολα προσβάσιμες από μηχανές. Υπάρχουν αρκετοί λόγοι που τα δεδομένα στον Ιστό ενδέχεται να μην παρουσιάζονται με ενα φιλικό προς τη μηχανή (machine-friendly) τρόπο. Πιθανών ο κύριος λόγος είναι ότι οι κάτοχοι των δεδομένων (δηλαδή αυτοί που έβαλαν τα δεδομένα στο διαδίκτυο) απλώς δεν περίμεναν ή δε σκόπευαν τα δεδομένα να διαβαστούν από ένα πρόγραμμα υπολογιστή. Για παράδειγμα, η Wikipedia είναι μια φανταστική πηγή από όλων των ειδών πληροφορίες αλλα ο HTML κώδικας που παράγει δεν είναι πάντα φιλικός προς τη μηχανή. Αλλος ένας λόγος που ένας ιστότοπος μπορεί να μην είναι φιλικός προς τη μηχανή, είναι ότι οι κάτοχοι των δεδομένων έχουν κάποιο συγκεκριμένο λόγο που δεν θέλουν τα προγράμματα να διαβάζουν τον ιστότοπό τους. Για παράδειγμα, ένας ιστότοπος αγορών μπορεί να θέλει να δυσκολέψει ένα αυτοματοποιημένο σύστημα να αναζητήσει τις καταχωρήσεις για τις τιμές των προϊόντων του, ετσι ώστε να αποτρέψει τους ανταγωνιστές του να σχεδιάσουν κάποιο σύστημα που θα τους ωφελεί, με το να βάζουν χαμηλότερες τιμές. Αυτό το παράδειγμα αναδεικνύει ένα

σημαντικό ζήτημα. Είναι πάντα σημαντικό να είναι γνωστοί οι ισχύοντες νόμοι καθώς και οι συμφωνίες χρηστών όταν χρησιμοποιείται το W.S. για ένα project. Η επίγνωση τυχόν πιθανών νομικών ζητημάτων και, γενικότερα, το να είναι κανείς υπεύθυνος και προσεκτικός "διαδικτυακός"πολίτης είναι ευθύνη του κάθε προγραμματιστή. Σύμφωνα με τα παραπάνω, είναι λοιπόν το W.S. εντελώς ηθικό και νόμιμο;

Είναι το Web Scraping Νόμιμο;

Παρά τις πολυάριθμες αποφάσεις που πάρθηκαν τα τελευταία 20 χρόνια, το W.S. και το αν είναι νομικά επιτρεπόμενο ακόμα καθορίζεται. Εάν τα 'σκραπαρισμένα' δεδομένα χρησιμοποιούνται για προσωπικούς και ιδιωτικούς σκοπούς και εμπίπτουν στο πεδίο εφαρμογής της θεμιτής χρήσης σύμφωνα με τη νομοθεσία περί πνευματικών δικαιωμάτων, συνήθως δεν υπάρχει πρόβλημα. Ωστόσο, εάν τα δεδομένα πρόκειται να αναδημοσιευτούν, εάν το scraping είναι αρκετά δυνατό ώστε να 'ριξει' ολόκληρο τον ιστότοπο ή εάν το περιεχόμενο προστατεύεται από πνευματικά δικαιώματα και το scraping παραβιάζει τους όρους παροχής υπηρεσιών, τότε υπάρχουν πολλά νομικά προηγούμενα που πρέπει να γνωρίζετε.
Στο *Feist Publications, Inc. v. Rural Telephone Service Co*., το ανώτατο δικαστήριο των Ηνωμένων Πολιτειών αποφάσισε πως το scraping και η αναδημοσίευση γεγονότων είναι επιτρεπτά. Σε μια παρόμοια υπόθεση στην Αυστραλία, η εταιρία *Telstra Corporation Limited v. Phone Directories Company Pty Ltd*, ανακοίνωσε πως μόνο τα δεδομένα με ταυτοποιήσιμο συγγραφέα μπορούν να προστατεύονται από πνευματικά δικαιώματα. Άλλη μία υπόθεση ιδιου περιεχομένου στις Ηνωμένες Πολιτείες, στην οποία η επαναχρησιμοποίηση των ιστοριών της *Associated Press* (εταιρεία ειδήσεων) για μία είδηση, κρίθηκε ως παραβίαση των πνευματικών δικαιωμάτων. Τέλος,σε μια υπόθεση της Ευρωπαϊκής Ένωσης στη Δανία,
ofir.dk vs home.dk, αποφασίστηκε πως το τακτικό scraping είναι επιτρεπτό.
Υπήρξαν επίσης αρκετές περιπτώσεις στις οποίες εταιρείες έχουν κατηγορήσει τον ενάγοντα για επιθετικό scraping και προσπάθησαν να το σταματήσουν μέσω νομικής διαταγής. Το πιό πρόσφατο συμβάν, *το QVC v. Resultly* (τηλεοπτικό δίκτυο), αποφάσισε πως, εάν το scraping δεν είχε ως αποτέλεσμα ζημιά ιδιωτικής ιδιοκτησίας, δεν θα μπορούσε να θεωρηθεί ως σκόπιμη βλάβη.
Αυτές οι περιπτώσεις υποδηλώνουν πως όταν τα σκραπαρισμένα δεδομένα αποτελούν δημόσια γεγονότα (π.χ. τοποθεσίες επιχειρήσεων και τηλεφωνικοί καταχωρητές), μπορούν να αναδημοσιευτούν σύμφωνα με τους κανόνες δικαιωμάτων. Ομως, αν τα δεδομένα είναι πρωτότυπα (όπως απόψεις και κριτικές ή προσωπικά δεδομένα χρήστη), σχεδόν πάντα, δεν μπορούν να αναδημοσιευθούν για λόγους πνευματικών δικαιωμάτων. Σε κάθε περίπτωση, αν ποτε χρειαστεί να γίνει scraping δεδομένων από κάποια ιστοσελίδα, πρέπει ο καθένας να γνωρίζει τα δικαιώματα του. Ειδάλλως, πιθανόν να προκύψει ban της διεύθυνσης IP ή μέχρι και να υπάρξουν νομικές διαδικασίες. Αυτό σημαίνει, ότι θα πρέπει να υποβληθεί αίτημα λήψης και να οριστεί ένα *user agent* για να αναγνωρίσει το scraping. Επίσης θα πρέπει να ληφθούν στα υπόψιν, οι όροι χρήσης της ιστοσελίδας για να είναι βέβαιο ότι τα δεδομένα που θα συλλεχθούν απο το W.S. δεν θεωρούνται ιδιωτικά ή με πνευματικά δικαιώματα. Για αμφιβολίες ή τυχόν ερωτήσεις, υπάρχουν διαθέσιμοι δικηγόροι μέσων ενημέρωσης, για οποιαδήποτε συμβουλή σχετικά με τα προηγούμενα.

Στις ακόλουθες ιστοσελίδες υπάρχουν λεπτομέρειες σχετικά με τις παραπάνω υποθέσεις:

- **QVC v. Resulty** (https://www.paed.uscourts.gov/documents/opinions/16D0129P.pdf)
- **Associated Press v.Meltwater** (http://www.nysd.uscourts.gov/cases/show.php?db=special&id=279)
- **ofir.dk vs home.dk** (http://www.bvhd.dk/uploads/tx_mocarticles/S_-_og_Handelsrettens_afg_relse_i_Ofir-sagen.pdf)
- **Telstra Corporation Limited v. Phone Directories Company Pvt Ltd** (http://www.austlii.edu.au/au/cases/cth/FCA/2010/44.html)
- **Feist Publications Inc. v. Rural Telephone Service Co.** (http://caselaw.lp.findlaw.com./scripts/getcase/pl?court=US&vol=499&invol=340)

Ποιός χρησιμοποιεί το Web Scraping;

Υπάρχουν πολλές πρακτικές εφαρμογές για την πρόσβαση και τη συλλογή δεδομένων από τον Ιστό, πολλές από τις οποίες εμπίπτουν στο πεδίο της επιστήμης δεδομένων. Ακολουθεί μια λίστα με μερικές ενδιαφέρουσες περιπτώσεις χρήσης στην πραγματική ζωή:

- Πολλά προϊόντα της Google έχουν επωφεληθεί απο το crawling στον ιστο. Για παράδειγμα το Google Translate αξιοποιεί το κείμενο που βρίσκεται στον ιστό για να εκπαιδευτεί και να βελτιωθεί.
- Πολύ σύνηθες είναι, οι ψηφιακοί καλλιτέχνες και το ψηφιακό μάρκετινγκ να χρησιμοποιεί δεδομένα από τον ιστό για κάθε είδους, ενδιαφέροντα και δημιουργικά project. Για παράδειγμα το "We Feel Fine" του Jonathan Harris και του Sep Kamvar, σκραπάρει ποικίλες blog σελίδες για να βρεί φράσεις που ξεκινούν με "i feel". Αυτό είχε ως αποτέλεσμα να απεικονίσει πως νιώθει ο κόσμος κατα την διάρκεια της ημέρας.
- Στο HR (τμήμα Ανθρώπινου Δυναμικού) και στις αναλύσεις εργαζομένων αξιοποιείται αρκετά το scraping. Η startup εταιρεία hiQ στον San Francisco ειδικεύεται στο να πουλάει αναλύσεις εργαζομένων, συλλέγοντας και εξετάζοντας πληροφορίες δημόσιων προφίλ. Για παράδειγμα, απ' το LinkedIn (https://www.bloomberg.com/news/features/2017-11-15/the-brutal-fight-to-mine-your-data-and-sell-it-to-your-boss).
- Σε μια άλλη έρευνα, μία ομάδα σκράπαρε μηνύματα από το Twitter, απο blogs και από άλλα μέσα κοινωνικής δικτύωσης με σκοπό να φτιάξουν ένα dataset, το οποίο χρησιμοποιήθηκε για να φτιαχτεί ένα μοντέλο πρόβλεψης κατάθλιψης και αυτοκτονικών σκέψεων. (https://www.sas.com/en_ca/insights/articles/analytics/using-big-data-to-predict-suicide-risk-canada.html)
- Σε ένα paper ονόματι "The Billion Prices Project: Using Online Prices for Measurement and Research", χρησιμοποιήθηκε το W.S. για να συλλέξει ενα dataset

για πληροφορίες από διαδικτυακές τιμές, που θα παρείχαν μία καθημερινή ένδειξη τιμής για πολλαπλές χώρες. (http://www.nber.org/papers/w22111)

- Τράπεζες και άλλα χρηματοπιστωτικά ιδρύματα χρησιμοποιούν το W.S. για ανάλυση ανταγωνιστών. Για παράδειγμα, συχνά οι τράπεζες σκραπάρουν δεδομένα από τους ανταγωνιστές τους για να δουν που θα ανοίξει ή θα κλείσει κάποιο υποκατάστημα, ή για να παρακολουθούν τα επιτόκια δανείων που προσφέρουν.
- Στο project "Analyzing 1000+ Greek Wines With Python" ο Florents Tselai σκραπάρει πληροφορίες για χιλιάδες ποικιλίες κρασιών από ένα ελληνικό οινοπολείο, με σκοπό να αναλύσει την προέλευση, τη βαθμολογία, τον τύπο και άλλα στοιχεία. (https://tselai.com/greek-wines-analysis.html)
- Ένας ερευνητής κατάφερε να εκπαιδεύσει ένα μοντέλο βαθιάς μάθησης που βασιζόταν σε σκραπαρισμένες φωτογραφίες από Instagram και Tinder, μαζί με τα likes τους, έτσι ώστε να καταλαβαίνει πότε μια φωτογραφία θα θεωρείται 'ελκυστική'. (http://karpathy.github.io/2015/10/25/selfie/). Οι κατασκευαστές κινητών τηλεφώνων ήδη ενσωματώνουν τέτοιου είδους μοντέλα σε εφαρμογές που έχουν να κάνουν με φωτογραφίες, για να βοηθήσουν το χρήστη να 'φρεσκάρει' το προφίλ του.
- Υπάρχει μια μελέτη όπου το W.S. χρησιμοποιείται για να εξάγει πληροφορίες από ιστοσελίδες εύρεσης εργασίας, για να πάρει μια ιδέα σχετικά με τα εργαλεία της επιστήμης δεδομένων στον εργασιακό χώρο
- Το Lyst, ένα διαδικτυακό μαγαζί μόδας στο Λονδίνο, σκράπαρε τον ιστό για να αποκτήσει πληροφορίες σχετικά με προϊόντα μόδας, και έπειτα εφάρμοσε σε αυτά μηχανική μάθηση για να παρουσιάσει αυτές τις πληροφορίες ξεκάθαρα και κομψά στους καταναλωτές μέσω μιας ιστοσελίδας (http://talks.lystit.com/dsl-scraping-presentation/).
- Τέλος μια ερευνητική ομάδα, με τη χρήση του W.S., παρακολουθούσε τα φόρουμς του διαδικτύου για την κατανόηση της κατάστασης του Bitcoin.

Σύμφωνα με τους έντεκα παραπάνω λόγους, μπορεί κανείς να κατανοήσει το ποιός μπορεί να χρησιμοποιήσει το W.S. και γιατί.

Web Scraping vs Web Crawling

Πριν τη μετάβαση στο επόμενο κεφάλαιο πρέπει να ξεκαθαριστούν οι διαφορές ανάμεσα σε δύο πολύ σημαντικούς παραμετρους. Το W.S. και το web crawling. Αν και έχουν πολλά κοινά μεταξύ τους, υπάρχουν χαρακτηριστικές διαφορές που τα ξεχωρίζουν.

Web Scraping

Το W.S. όπως έχει ήδη προαναφερθεί, μπορεί να γίνει και χειροκίνητα, παρόλο που τις περισσότερες φορές εκτελείται αυτόματα. Η βασική λειτουργία που χαρακτηρίζει το W.S. είναι πώς επικεντρώνεται σε συγκεκριμένα δεδομένα. Αυτο σημαίνει πως ο σκράπερ θα εξάγει μόνο τα δεδομένα που θα θελήσει να χρησιμοποιήσει σε μελλοντικές αναλύσεις.

Web Crawling

Το Web Crawling είναι η διαδικασία με την οποία ρομπότς (bots), ή αράχνες (spiders) οπως αλλιώς αποκαλούνται, διαβάζουν και αποθηκεύουν, όλο το περιεχόμενο μια ιστοσελίδας για σκοπούς αρχειοθέτησης ή ευρετηρίασης. Μηχανές αναζήτησης όπως η Google ή η Bing χρησιμοποιούν το web crawling για να εξάγουν πληροφορίες από ιστοσελίδες και να τις εντάξουν στις δικές τους μηχανές αναζήτησης. Αυτός είναι ο τρόπος με τον οποίο η Google μπορεί να πει, ποιες σελίδες θα έχουν τις πληροφορίες που αναζητάτε.

Διαφορά Web Scraping με Web Crawling

Μέχρι τώρα πιθανόν να έχει ήδη γίνει αντιληπτό τι κάνει ο Scraper και τι ο Crawler. Παρόλο που και οι δυο εξάγουν δεδομένα από ιστοσελίδες. Αν οχι, το παρακάτω διάγραμμα σίγουρα θα ξεκαθαρίσει τα πράγματα.

Figure 2 Web scraping vs Web crawling

Ο Web Crawler θα κάνει ένα πέρασμα σε όλες τις υποσελίδες της ιστοσελίδας και οχι ενα υποσύνολο της σελίδας. Απ' την άλλη, ο Scraper εστιάζει σε συγκεκριμένα σετ δεδομένων σε μια ιστοσελίδα. Για παράδειγμα, πληροφορίες ενός προϊόντος, αθλητικά δεδομένα, πληροφορίες για μετοχές κ.α.

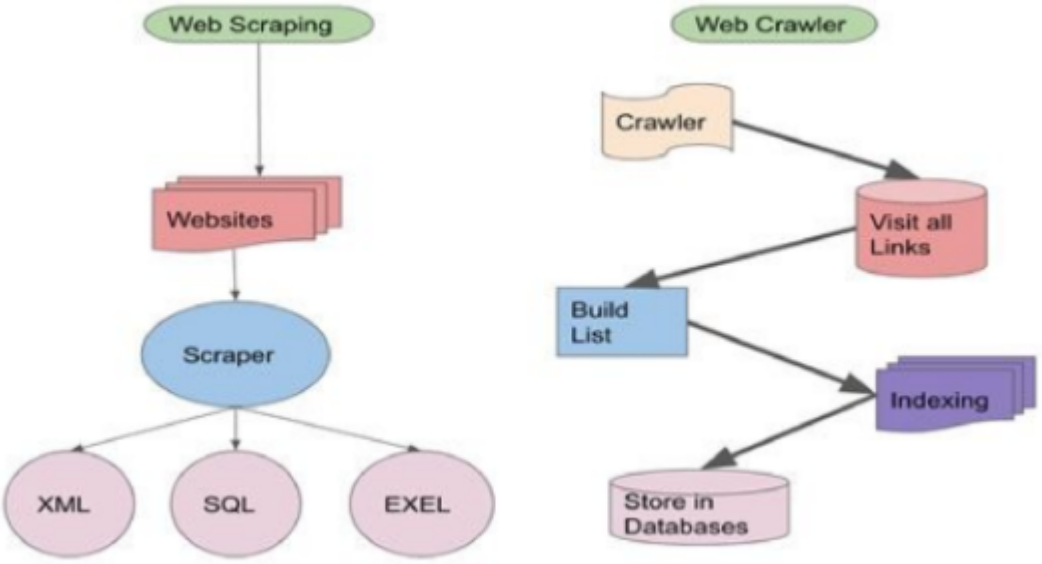

Figure 3 Scraper vs Crawler

Γιατί να χρησιμοποιήσει κανείς την Python στο Web Scraping;

Αφού καλύφτηκε το θεωρητικό κομμάτι του W.S., τώρα, έφτασε το σημείο να παρουσιαστούν οι λόγοι που κάποιος αξίζει να χρησιμοποιήσει την προγραμματιστική γλώσσα Python στο Project του. Όπως είναι φανερό, και όπως έχει προαναφερθεί στην περίληψη του κειμένου, η Δ.Ε. έχει επιλυθεί με χρήση της Python. Υπάρχουν αρκετοί λόγοι που η Python είναι ιδανική γλώσσα για projects που εχουν να κάνουν με το W.S.. Ο κύριος λόγος που κάποιος επιλέγει να χρησιμοποιήσει την Python για W.S. είναι και ο λόγος που επιλέγει την Python για οποιοδήποτε άλλου είδους project. Είναι απ' τις ευκολότερες γλώσσες προγραμματισμού για να μάθει κανείς, ακόμα και αν δεν υπάρχει η παραμικρή εμπειρία σε προγραμματισμό. Αλλος ένας λόγος είναι πώς πλέον η Python είναι μία από τις πιό δημοφιλής γλώσσες προγραμματισμού. Αυτο σημαίνει πως στο διαδίκτυο υπάρχει μια τεράστιο κοινότητα απο προγραμματιστές, οι οποίοι μπορούν να σε βοηθήσουν με τις απαντήσεις που ποστάρουν. Γενικά η Python έχει δημιουργήσει μια από τις μεγαλύτερες κοινότητες χρηστών στις

μέρες μας, χάρη στην προσβασιμότητα και την μακροχρόνια δημοτικότητά της. Ως αποτέλεσμα των προηγουμένων χαρακτηριστικών, είναι τα ποικίλα διαθέσιμα εργαλεία που προσφέρει, με τα περισσότερα απο αυτά να είναι δωρεάν. Πολλοί χρήστες επιλέγουν να γράψουν κώδικα σε Python διότι είναι πολύ εύκολος στην γραφή. Για παράδειγμα στην απλούστερη περίπτωση του "Hello World" δείτε παρακάτω την διαφορά ανάμεσα σε Java και C++.

Java:
```
class Example{
public static void main(String[]args){
System.out.println("Hello World");
}}
```

C++:
```
#include <iostream>
int main(){
std::cout << "Hello World";
}
```

Python:
```
print("Hello World");
```

Μπορεί να κατανοηθεί με ευκολία γιατί κάποιος αξίζει να επιλέξει την Python ως γλώσσα προγραμματισμού για W.S. και όχι μόνο. Είναι πολύ σημαντικό ο κώδικάς σε μια εφαρμογή να είναι καθαρογραμένος και εύκολα κατανοητός. Αυτό θα βοηθήσει και τους τυχόν αναγνώστες να καταλάβουν εύκολα τη λύση σου. Επίσης θα βοηθήσει και τον ίδιο τον προγραμματιστή να θυμηθεί τον κώδικα του σε περίπτωση που έχει καιρό να επανέλθει στο συγκεκριμένο project. Για αυτόν το λόγο, ο κώδικας της Python, αν είναι καλά γραμμένος είναι εξαιρετικά εύκολος στη συντήρηση και στην επαναχρησιμοποίηση, κάτι που είναι πολύ σημαντικό στο W.S.. Τέλος, η Python είναι εξαιρετική για project με συνεχή εξέλιξη. Το W.S. πολλές φορές είναι απαραίτητο έτσι ώστε να διευρύνουμε το project μας. Παρακάτω στην Δ.Ε. θα παρουσιαστεί πόσο εύκολα μπορεί να συνδυαστεί ο βασικός κώδικας του W.S. με τυχόν υπάρχοντα κώδικα που έχουμε.

Γιατί Python 3 και οχι 2;

Σύμφωνα με τους ίδιους τους δημιουργούς της Python, η Python 2 είναι κληρονομιά, ενώ η Python 3 είναι το παρόν και το μέλλον της γλώσσας. η Δ.Ε. σαφώς και έχει υλοποιηθεί με Python 3. Παρακάτω θα αναλυθούν μερικά χαρακτηριστικά της δεύτερης έκδοσης της Python, έπειτα της τρίτης, και τέλος θα γίνει μια μικρή σύγκριση.

Python 2

Η Python 2 για πρώτη φορά κυκλοφόρησε από την ομάδα BeOpen Python Labs το 2000. Αλλά πρίν τη δημιουργία της ομάδας, ο Rossum ήταν υπεύθυνος για τις περισσότερες λειτουργίες και την αποσφαλμάτωση της γλώσσας. Αλλά ήθελε ο ρόλος της Python να είναι πιο περιεκτικός και βασικός ως προς τη γραφή. Στόχος της γλώσσας ήταν να είναι εύκολη στην εκμάθηση για έναν "μέσο άνθρωπο". Με αυτήν την ιδέα, η ομάδα Python Labs κυκλοφόρησε την έκδοση Python 2.X , με την πρόθεση, η γλώσσα να ανοιχτεί σε "βελτιώσεις από την κοινότητα" αντί να είναι αρμοδιότητα του Rossum. Η Python 2.7 είναι η τελευταία έκδοση της Python 2, καθώς και το έτος 2020 είναι η τελευταία χρονιά της γλώσσας.

Python 3

Η Python 3 κυκλοφόρησε το 2008, αλλά δεν ήταν απλώς μια νέα έκδοση της Python 2 μετά την αποσφαλμάτωση. Άλλαξε ολοκληρωτικά σε μια γλώσσα, που ήταν συμβατή προς τα μπροστά. Δηλαδή θα υποστήριζε μόνο εκδόσεις που θα έβγαιναν μετά από αυτήν. Ο σκοπός της σύνταξης της Python 3, ήταν να αποτρέψει τον περιττό ή επαναλαμβανόμενο κώδικα (δηλαδή τον κώδικα που ουσιαστικά εκτελεί τις ίδιες λειτουργίες με διαφορετικούς τρόπους). Η Python 3.X στόχευε στο να αποκτήσει, μόνο έναν, σαφή τρόπο να κάνεις πράγματα. Για τους αρχάριους προγραμματιστές, αυτό απέρριπτε αμέσως τα κύρια προβλήματα που παρουσιαζόταν κατά την εκμάθηση μιας γλώσσας προγραμματισμού.

Python 2 vs Python 3 : Σύγκριση

Η Python 3 είναι μια γλώσσα με πολλαπλά παραδείγματα. Αυτό σημαίνει πως έχει μια ποικιλία από ταξινομήσεις για τους σκοπούς της. ¨Εχοντας αυτήν τη λειτουργικότητά, η Python 3 είναι ιδανική για πολλα πράγματα όπως η επιστήμη των δεδομένων, το web development και την ανάλυση scripts. Με την ιδεολογία του "ένας τρόπος να κάνεις κάτι" οι απεριόριστες δυνατότητες γίνονται απλούστερες για τους προγραμματιστές.
Η πλήρης αναμόρφωση τής Python 2 δεν έγινε επειδή η γλώσσα ήταν κακή. Η αλλαγή της γλώσσας προς την Python 3 έγινε για υποστήριξη στο γράψιμο. Δημιουργήθηκε ένα σύστημα που καθιερώνει ένα σύνολο μεταβλητών ή χαρακτηριστικών στην ιδιότητα του τύπου (type). Η Python 2 και η Python 3 έχουν μερικές βασικές διακρίσεις. Η σύνταξη είναι η κύρια, όπως και η εκτύπωση (print). Στην Python 2 η εντολή είναι Print "hello world" ενώ στην Python 3 print("hello world").
Αλλη μια έντονη διαφορά είναι ο default τύπος των συμβολοσειρών κειμένου (strings). Η Python 2 χρησιμοποιεί ASCII, που σημαίνει ότι δεν παρέχει μεγάλη ποικιλία σε χαρακτήρες. Από την άλλη, η Python 3, χρησιμοποιεί Unicode, το οποίο είναι το χρήσιμο UTF-8 (Unicode Transformation Format – 8-bit). Αυτό έχει ως αποτέλεσμα, να δώσει την δυνατότητα στη γλώσσα να αναπαριστά ξένες γλώσσες και άλλα ευρέως χρησιμοποιούμενα σύμβολα.

Ακόμα, οι βιβλιοθήκες της Python 2 δεν είναι "συμβατές προς τα μπροστά" όπως στην Python 3. Αυτό σημαίνει πώς δεν μπορεί να υποστηρίξει νέες εκδόσεις και επίσης δυσκολεύει την Python 3 στο να εντάξει τις βιβλιοθήκες της.
Τέλος, υπάρχει άλλη μια σημαντική διαφορά όσον αφορά τη διαίρεση. Στην Python 2 η διαίρεση ακέραιου αριθμού επιστρέφει τον κοντινότερο στρογγυλοποιημένο αριθμό προς τα κάτω (π.χ. 5/2 θα επιστρέψει 2). Στην Python 3 τα πράγματα είναι όπως θα έπρεπε. Στην ίδια διαίρεση ακέραιου αριθμού θα επιστρέψει ακριβώς 2.5.

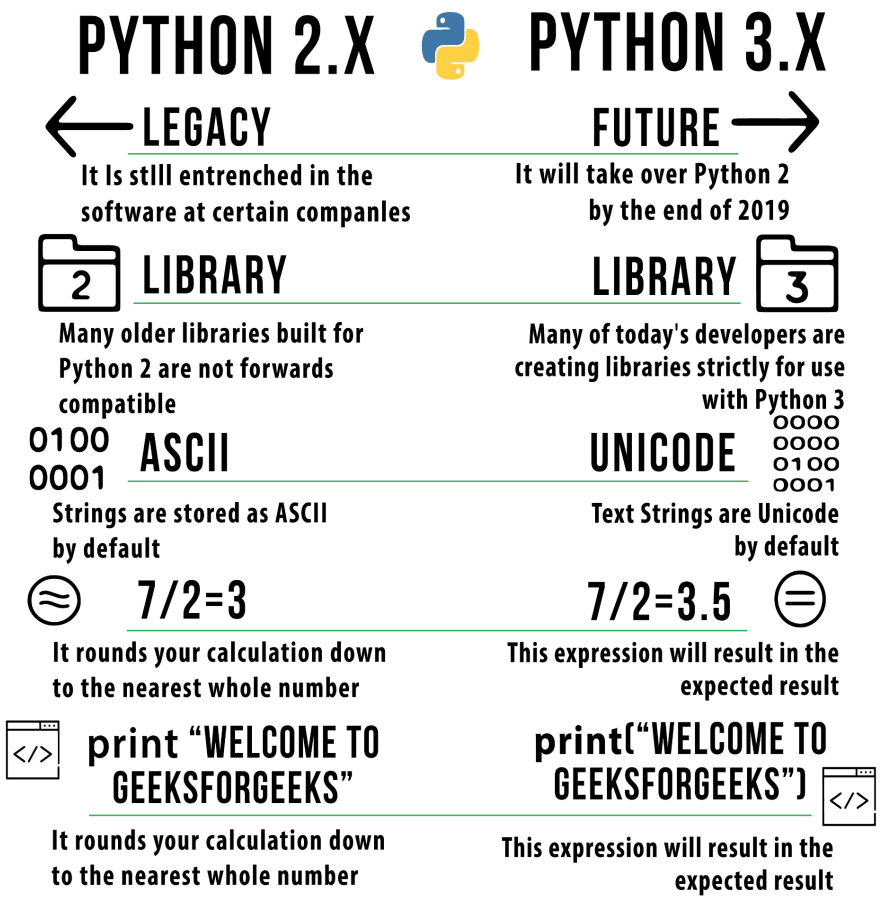

Figure 4 Python 2 vs Python 3

Συμπέρασμα

Ολοκληρώνοντας τη σύγκριση, φτάνει επιτέλους το σημείο του συμπεράσματος. Ποιά έκδοση λοιπόν της Python είναι καλύτερη για να μάθει ένας χρήστης; Η απάντηση είναι ξεκάθαρη. Σίγουρα η Python 3. Ειδικά για τους αρχάριους προγραμματιστές, η Python 3 είναι η καλύτερη επιλογή. Η αναγνωσιμότητα, η λειτουργικότητα και η δημοτικότητα της Python 3, της δίνουν το πάνω χέρι. Επίσης είναι λογικό κάποιος να προτιμήσει την τελευταία έκδοση της Python, ειδικά όταν η Python 2 κινδυνεύει να "λήξει". Τέλος, είναι πολύ σπάνιο να βρεθεί μια σύγχρονη εταιρεία που χρησιμοποιεί Python 2, καθώς σχεδόν όλες χρησιμοποιούν την τρίτη έκδοση.

Κεφάλαιο 2ο: MongoDB

Εισαγωγή

Η MongoDB είναι μια σχετικά καινούργια γενιά βάσης δεδομένων η οποία δεν έχει καμία σχέση με πίνακες, σχήματα, SQL ή γραμμές. Δεν έχει συναλλαγές, συμμόρφωση ACID (acid compliance), ενώσεις (joins), ξένα κλειδιά και άλλες τέτοιες λειτουργίες που πολλές φορές προκαλούν πονοκέφαλο στο χρήστη. Με λίγα λόγια, η MongoDB είναι μια τελείως διαφορετική βάση δεδομένων που πολύ πιθανόν, ο χρήστης να μην είναι συνηθισμένος, ειδικά αν η προτίμηση του είναι οι σχεσιακές βάσεις δεδομένων. Σε αυτό το κεφάλαιο θα γίνει μια ανασκόπηση της φιλοσοφίας της MongoDB και θα ειπωθούν λίγα λόγια, για τους λόγους που αξίζει κανείς να επιλέξει στη συγκεκριμένη βάση δεδομένων.

Ανασκόπηση της φιλοσοφίας της MongoDB

Όπως όλα τα προγράμματα, ετσι και η MongoDB έχει τις αρχιτεκτονικές της φιλοσοφίες που βοηθούν στην ανάπτυξή της. Μια απο τις σημαντικότερες είναι πως η θεωρία που λέει πως "ένα μέγεθος δεν χωράει τα πάντα". Για πολλά χρόνια, χρησιμοποιούσαν τις παραδοσιακές σχεσιακές (SQL) βάσεις δεδομένων (όπως η mySQL, η PostgresSQL, η Oracle κ.α.) για να αποθηκεύουν περιεχόμενα από όλους τους τύπους. Δεν είχε σημασία αν τα δεδομένα "ταίριαζαν καλα" στο σχεσιακό μοντέλο, θα έμπαιναν στη βάση έτσι κι αλλιώς.
Εδώ εμφανίζεται η ομάδα της MongoDB που δεν θα έφτιαχνε άλλη μια βάση δεδομένων που θα προσπαθούσε να κάνει τα πάντα για όλους. Η ιδέα ήταν να φτιάξουν μια βάση που θα δούλευε με καταχωρήσεις (documents) και όχι με γραμμές (rows) οπου θα ηταν γρήγορη, μαζικά επεκτάσιμη και εύκολη στη χρήση. Για να γίνει αυτό όμως, η ομάδα της MongoDB έπρεπε να αφήσει κάποιες λειτουργίες στην άκρη, που σημαίνει πως η βάση δεν είναι ιδανική για όλες τις απαιτήσεις του χρήστη. Για παράδειγμα η MongoDB δεν περιέχει υποστήριξη συναλλαγών. Αυτό σημαίνει πως δεν κάνει για εφαρμογές λογιστικής. Παρόλα αυτά, η δομή της MongoDB που είναι document-oriented () είναι βελτιστοποιημένη για ταχύτητα και επεκτασιμότητα. Δεν χρειάζεται να ανησυχεί κανείς για το πώς θα βάλει τα δεδομένα στη βάση. Δηλαδή δεν είναι απαραίτητο να ταξινομούνται τα δεδομένα σε πίνακες πριν ανέβουν στη βάση, αρκεί να είναι όλα μαζεμένα και μετα η MongoDB θα κάνει την υπόλοιπη δουλειά. Η γενική ιδέα είναι ο χρήστης να επιλέξει το σωστό εργαλείο για τη σωστή δουλειά. Αν δηλαδή αποφασίσει να χρησιμοποιήσει την MongoDB ξέροντας για τι είναι ικανή, σίγουρα θα μείνει ικανοποιημένος.

Γιατι να χρησιμοποιήσει κανείς τη MongoDB;

Παρακάτω αναφέρονται οι πέντε πιο σημαντικοί λόγοι για να χρησιμοποιήσει κανείς την MongoDB.

1. Προσανατολισμένη σε καταχωρήσεις (document-oriented). Αφού ο τύπος της βάσης MongoDB είναι NoSQL, αντί να έχει τα δεδομένα σε σχεσιακή μορφή, τα αποθηκεύει σε

μορφή καταχωρήσεων. Αυτό την κάνει να είναι ευέλικτη και προσαρμόσιμη στις απαιτήσεις και στις επιχειρηματικές καταστάσεις.

2. Ευρετηρίαση (Indexing). Μπορούν να δημιουργηθούν ευρετήρια (indexes) για να βελτιώσουν την απόδοση της αναζήτησης μέσα στη MongoDB. Αυτό σημαίνει ότι μπορεί να γίνει αναζήτηση για οποιοδήποτε πεδίο μέσα στη βάση.
3. Ad hoc ερωτήματα (Ad hoc queries. δηλ. ερωτήματα για έναν σκοπό). Η MongoDB υποστηρίζει αναζήτηση με πεδίο (field), ερωτήματα εύρους, καθώς και regular expression queries (ερωτήματα που συμπεριλαμβάνουν ειδικούς χαρακτήρες π.χ `{}[]()^$.|*+?)`. Μπορούν να γίνουν ερωτήματα στη βάση για να επιστραφούν συγκεκριμένα πεδία μέσα από μια καταχώρηση (document).
4. Αντιγραφή (Replication). Η MongoDB μπορεί να παρέχει υψηλή διαθεσιμότητα σε σετ αντιγράφων. Ενα σετ αντιγράφων αποτελείται από δύο ή περισσότερα αντίγραφα της MongoDB. Κάθε μέλος ενός σετ αντιγράφων, μπορεί να ενεργεί ως πρωτεύον ή δευτερεύον αντίγραφο ανά πάσα στιγμή. Το πρωτεύον αντίγραφο είναι ο main server που αλληλεπιδρά με τον πελάτη και εκτελεί όλες τις εντολές read/write. Το δευτερεύον, κατέχει ένα αντίγραφο των δεδομένων του πρωτεύον, χρησιμοποιώντας μια ενσωματωμένη αντιγραφή.
5. Κατανομή φόρτου (load balancing). Η MongoDB χρησιμοποιεί την έννοια του διαμοιρασμού για να κλιμακωθεί οριζόντια, διαχωρίζοντας δεδομένα σε πολλαπλά "αντίγραφα" της MongoDB. Η MongoDB μπορεί να τρέξει σε πολλαπλούς διακομιστές (servers), κατανοώντας το φόρτο και/ή αντιγράφοντας τα δεδομένα για να κρατήσει το σύστημα ασφαλή σε περίπτωση αστοχίας του hardware.

Κεφάλαιο 3ο: GitHub

Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιαστούν ξεχωριστά, το Git και το GitHub. Τι είναι, ποιά η διαφορά τους και γιατί κανείς να τα χρησιμοποιήσει. Στη συνέχεια θα περιγραφούν οι βασικές εντολές του Git και τέλος θα αναλυθεί τι είναι τα GitHub actions καθώς και ήταν ένα από τα βασικά εργαλεία που χρησιμοποιήθηκαν στην παρούσα Δ.Ε.

Τι είναι το git;

Το Git είναι ένα σύστημα ελέγχου έκδοσης. Στη μηχανική λογισμικού, ο έλεγχος έκδοσης είναι μια κατηγορία συστημάτων που είναι υπεύθυνα για τη διαχείριση αλλαγών σε προγράμματα υπολογιστών, έγγραφα, μεγάλους ιστότοπους ή άλλες συλλογές πληροφοριών. Πιό συγκεκριμένα το Git είναι ένα κατανεμημένο σύστημα ελέγχου έκδοσης, που σημαίνει ότι όσοι δουλεύουν στο ίδιο project του Git, έχουν ένα αντίγραφο ολόκληρου του ιστορικού του project, και όχι απλά την τρέχων κατάστασή του.

Τι είναι το GitHub;

Το GitHub είναι μια διαδικτυακή πλατφόρμα, στην οποία μπορείς να ανεβάσεις ένα αντίγραφο από το Git repository σου. Ένα Git repository είναι ο . git/ φάκελος μέσα σε ένα project. Παρακολουθεί όλες τις αλλαγές που έγιναν σε αρχεία στο project σας, δημιουργώντας ένα ιστορικό με την πάροδο του χρόνου. Το Github λοιπόν, πέρα απτο να ανεβάζεις το git repository σου, σου επιτρέπει να συνεργαστείς πολύ πιο εύκολα με άλλα άτομα στο ίδιο project. Αυτό γίνεται παρέχοντας μια κεντρική τοποθεσία για να μοιραστείς το repository και να εκτελείς και άλλες σημαντικές λειτουργίες όπως *το forking, τα Pull requests, τα Issues,* καθώς και να συζητήσεις και να αξιολογήσεις αλλαγές με την ομάδα σου πιο αποτελεσματικά.

Γιατί να χρησιμοποιήσει κανείς το GitHub

Πέρα απ το να ανεβάζεις το git repository σου, το GitHub έχει και άλλες οφέλιες. Παρακάτω θα αναφέρω τους λόγους για τους οποίους αξίζει κάποιος να χρησιμοποιήσει το GitHub.

- **Καταγραφή απαιτήσεων**
  Με χρήση αιτημάτων, μπορείς να καταγράψεις τα σφάλματα ή να προσδιορίσεις καινούργιες λειτουργιες που θα ήθελες να χρησιμοποιήσει η ομάδα σου.

- **Συνεργασία σε ανεξάρτητα ρεύματα του ιστορικού**

Χρησιμοποιώντας *branches και pull requests,* μπορείς να συνεργαστείς με διαφορετικά branches ή λειτουργίες.

- **Αξιολόγηση της προόδου των εργασιών**
  Παρακολουθώντας τη λίστα των pull requests, μπορείς να δεις όλες τις διαφορετικές λειτουργίες που εκτελούνται εκείνη τη στιγμή. Επίσης κάνοντας κλικ στα pull requests μπορείς δεις τις τελευταίες αλλαγές που έχουν γίνει, τις συζητήσεις που έχουν γίνει πάνω σε αυτές τις αλλαγές, καθώς και μπορείς να αφήσεις και τη δικιά σου αξιολόγηση για το αν είσαι σύμφωνος, πριν αποδεχτούν οι αλλαγές.

- **Παρακολούθηση της ομαδικής προόδου**
  Μέσω των commits μπορεί ο καθένας να παρακολουθήσει και να δει τη δουλειά που έχει κάνει ο καθένας μέσα σε ένα project,

Βασικές εντολές του Git

Υπάρχουν πολλαπλές εντολές του Git που χρησιμοποιούν προγραμματιστές, για να εκτελέσουν ορισμένες λειτουργίες όπως η δημιουργία, η αντιγραφή, η αλλαγή κ.α. Αυτές οι εντολές εκτελούνται μέσω εφαρμογών όπως το GitHub Desktop, το Git Kraken ή και απευθείας από τη γραμμή εντολών. Παρακάτω θα αναφέρω και θα περιγράψω τις βασικές εντολές του Git.

git init
Δημιουργεί ένα ολοκαίνουργιο Git repository και αρχίζει να παρακολουθεί έναν υπάρχοντα κατάλογο. Προσθέτει έναν κρυφό υποφάκελο στον υπάρχοντα κατάλογο που φιλοξενεί την εσωτερική δομή δεδομένων που απαιτείται για τον έλεγχο έκδοσης.

git clone your_repo_url
Αντιγράφει το repository τοπικά στον υπολογιστή σας.

git diff
Μας δείχνει αναλυτικά τι άλλαξε σε κάθε αρχείο.

git branch
Μας δείχνει τα branches που δουλεύουν τοπικά.

git add
Η χρήση του repository έχει ως εξης:
Όταν αλλάξουμε κάτι, θέλουμε να το προσθέσουμε στο repository έτσι ώστε να υπάρχει σαν history και να μπορούμε να το επαναφέρουμε. Η διαδικασία ξεκινά από αυτή την εντολή. Όταν γράψουμε αυτή την εντολή προστίθενται σε ένα ενδιάμεσο state τα αρχεία που αλλάξαμε που λέγεται 'stage'. Εδώ μπορούμε να διαλέξουμε ποια αρχεία αφορούν το συγκεκριμένο feature που θέλουμε να βάλουμε στο repository. Μπορούμε να βάλουμε μεμονωμένα αρχεία με την εντολή git add filename, να τα προσθέσουμε όλα με την εντολή git add -A ή να αφαιρέσουμε αρχεία από το stage (για να τα βάλουμε σε κάποιο άλλο 'commit').

git commit -m "commit message"
Με αυτή την εντολή φτιάχνουμε ένα σημείο στο repository όπου μπορούμε να το επαναφέρουμε ανά πάσα στιγμή. Καλό είναι τα commits να είναι συχνά και κομμένα ανάλογα το feature που αντιπροσωπεύουν κάθε φορά.

git status
Η εντολή αυτή εμφανίζει την κατάσταση του καταλόγου εργασίας και τα στάδια ολοκλήρωσης ενός project. Μας επιτρέπει να δούμε ποιές αλλαγές έχουν πραγματοποιηθεί, ποιές όχι και ποιά αρχεία δεν παρακολουθούνται από το Git.

git merge <branch>
Η εντολή αυτή ενοποιεί το ιστορικό του branch που περνάς ώς παράμετρο με το τωρινό branch. Προσπαθεί να ενώσει τις αλλαγές στα αρχεία και από τα δύο branches. Ετσι δημιουργεί ένα commit με 2 γονείς. Το τρέχον branch και το branch που δίνεται ως παράμετρος.

git pull origin master
Με αυτή την εντολή τραβάμε τοπικά ότι αλλαγή έχει γίνει από άλλους και έχει πάει στο remote repository.

git push origin master
Ότι αλλαγές κάνουμε με τις εντολές git add και git commit, βρίσκονται τοπικά μέχρι να τις 'στείλουμε' στο remote repository και να το πάρουν και οι άλλοι. Με την εντολή git push origin master οι αλλαγές στέλνονται στο remote repository.

GitHub Actions

Σε αυτήν την υποενότητα θα εξηγήσω τι είναι τα GitHub Actions. Στη συνέχεια θα αναφέρω τα δύο είδη των GitHub Actions και τέλος, πιο συγκεκριμένα, θα περιγράψω τι είναι ένα workflow στα GitHub Actions και πώς λειτουργεί.

Τι είναι τα GitHub Actions;

Τα GitHub Actions είναι scripts (μία δέσμη ενεργειών) τα οποία είναι υπεύθυνα για την αυτοματοποίηση εργασιών σε μια ροή εργασιών ανάπτυξης λογισμικού στο GitHub.

Τα δύο είδη των GitHub Actions

Υπάρχουν δύο είδη GitHub Actions. Τα container actions και τα javascript actions. Με τα **container actions**, το περιβάλλον είναι είναι μέρος του κώδικα του action. Αυτά τα actions μπορούν να τρέξουν

μονο σε περιβάλλον Linux το οποίο φιλοξενεί το GitHub. Επισης τα container actions υποστηρίζουν πολλές διαφορετικές γλώσσες προγραμματισμού.

Τα **javascript actions** δε συμπεριλαμβάνουν το περιβάλλον στον κώδικα. Θα πρέπει να προσδιοριστεί το περιβάλλον για να εκτελεστούν τα actions. Μπορείς να τρέξεις τα actions σε κάποια εικονική μηχανή στο cloud ή επί τόπου. Τέλος, τα javascript actions υποστηρίζουν Linux, macOS και Windows περιβάλλον.

Τι είναι ένα GitHub Action workflow;

Το GitHub Action workflow είναι μια διαδικασία που εγκαθιστάς στο repository σου, για την αυτοματοποίηση εργασιών του κύκλου ζωής ανάπτυξης λογισμικού, συμπεριλαμβάνοντας τα GitHub Actions. Με το workflow μπορείς να δημιουργήσεις, να τεστάρεις, να κατασκευάσεις και να αναπτύξεις οποιοδήποτε project στο GitHub. Για να δημιουργήσεις ένα workflow, προσθέτεις τα actions σε ενα .yml αρχείο στο φάκελο .github/workflows μέσα στο GitHub repository σου. Παρακάτω θα δειξω ενα απλο παράδειγμα για να γίνει πιο εύκολα κατανοητό τι είναι ενα workflow και πως λειτουργεί.

```yml
name: A workflow for my Hello World file
on: push
jobs:
  build:
    name: Hello world action
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v1
    - uses: ./action-a
      with:
        MY_NAME: "Mona"
```

Figure 5 Παράδειγμα yml αρχείου

Παρατηρούμε στην εικόνα την ιδιότητα on:. Είναι μια ενέργεια (trigger) που καθορίζει πότε θα τρέξει το workflow. Στο συγκεκριμένο παράδειγμα, παρατηρούμε πως το workflow θα τρέξει όταν ολοκληρωθεί ενα push στο repository. Μπορούμε να θέσουμε στο trigger μια μεταβλητή όπως το on: push ή και πολλές μεταβλητές σε μορφή πίνακα π.χ. on: [push, pull_request] . Ακόμα μπορούμε να προγραμματίσουμε το περιορίσουμε την εκτέλεση ενός workflow σε συγκεκριμένα αρχεία, tags, ή αλλαγές στο branch. Όπως για παράδειγμα στην παρακάτω εικόνα, οπου το το workflow εκτελείται στα push και pull requests αλλά μόνο για το master branch.

```yml
on:
  # Trigger the workflow on push or pull request,
  # but only for the master branch
  push:
    branches:
      - master
  pull_request:
    branches:
      - master
  # Also trigger on page_build, as well as release created events
  page_build:
  release:
    types: # This configuration does not affect the page_build event above
      - created
```

Figure 6 Εκτέλεση push/pull requests μόνο για το master branch.

Τέλος, μπορούμε να προγραμματίσουμε το καθε πότε θα τρέχει το workflow. Αυτό γίνεται με τη λειτουργία schedule, όπου δηλώνεται η σύνταξη της cron για τον ακριβή προσδιορισμό του χρόνου. Για άλλη μια φορά τα πράγματα θα γίνουν πιο ξεκάθαρα με ένα παράδειγμα.

```yml
on:
  schedule:
    - cron: '30 5 * * 1,3'
```

Figure 7 Προγραμματισμένο workflow

Στο παραπάνω παράδειγμα το workflow θα εκτελεστεί στις 05:30 τη δευτέρα και την τετάρτη. Η cron σύνταξη έχει πέντε πεδία που διαχωρίζονται με κενό (space), και το κάθε πεδίο αναπαριστά μια μονάδα του χρόνου.

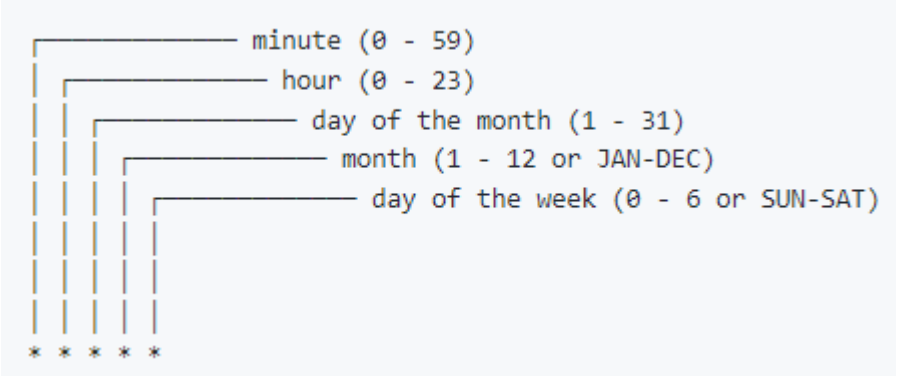

Figure 8 Σύνταξη cron

Ιδανικό εργαλείο για τη σύνταξη cron είναι το https://crontab.guru/ .

Ένα workflow θα πρέπει να έχει τουλάχιστον μια δουλεία (job). Το job είναι ένα τμήμα του workflow που συσχετίζεται με έναν δρομέα (runner). Ο δρομέας μπορεί να είναι φιλοξενούμενος στο GitHub ή αυτοφιλοξενούμενος (self-hosted). Εσείς διευκρινίζετε τον δρομέα μέσω της ιδιότητας runs-on: . Στην εικόνα 5 το workflow θα τρέξει το job στο ubuntu-latest.
Κάθε job του workflow έχει βήματα (steps) προς ολοκλήρωση. Στο παράδειγμά μου, το βήμα χρησιμοποιεί το action actions/checkout@v1 για να ελέγξει το repository.

Κεφάλαιο 4ο: Υλοποίηση του Project

Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλυθούν τα βήματα της υλοποίησης της Δ.Ε. Η Δ.Ε. υλοποιείται στα Windows και σε γλώσσα Python. Άρα η Python θεωρείται δεδομένη στις εγκαταστάσεις του υπολογιστή. Παρακάτω ακολουθούν τα βήματα για τις απαραίτητες εγκαταστάσεις καθώς και οι επαληθεύσεις τους. Συνεχίζοντας, λίγα λόγια για τις βιβλιοθήκες των requests και BeautifulSoup και τέλος, όλη η διαδικασία εκτέλεσης του project από την αρχή μέχρι το τέλος. Δηλαδή η σύνδεση με τη βάση δεδομένων και η αλληλεξάρτηση με το πρόγραμμα, το W.S. μαζί με όλες τις λειτουργίες του καθώς και η χρήση του GitHub Action workflow για την αυτοματοποιημένη εκτέλεση του project.

Επαλήθευση της διαδρομής της Python στα Windows

Πολλές φορές μετά την εγκατάσταση της Python στον υπολογιστή μας, η διαδρομή για το python.exe δεν προστίθεται από προεπιλογή στα environment variables των windows. Αυτό μπορούμε να το επαληθεύσουμε μέσω του cmd με την εντολή *python* .
Αν η διαδρομή για την python έχει προστεθεί κανονικά στα environment variables, τοτε η εντολή θα τρέξει κανονικά χωρίς κανένα error. Απ΄την άλλη, αν δεν υπάρχει, τότε θα πρέπει να προστεθεί χειροκίνητα, με τα ακόλουθα βήματα.

1. Δεξί κλικ στον υπολογιστή μου, και ιδιότητες.
2. Επιλογή Ρυθμίσεις συστήματος για προχωρημένους.
3. Επιλογή Μεταβλητές περιβάλλοντος
4. Στο μενού που θα ανοιξει, μεταβλητές συστήματος, επιλογή του Path και επεξεργασία. Η επόμενη οθόνη θα εμφανίσει όλους τους καταλόγους που αποτελούν μέρος της μεταβλητής PATH.
5. Κλικ στο Νέο και εισαγωγή του καταλόγου εγκατάστασης της Python.

Η Python μπορεί τώρα να χρησιμοποιηθεί απευθείας από τη γραμμή εντολών χωρίς να χρειάζεται να γράψετε την τοποθεσία της. Η επιβεβαίωση γίνεται με την εντολή python –version, όπου θα επιστρέψει την έκδοση της Python που είναι εγκατεστημένη στο σύστημά σας.

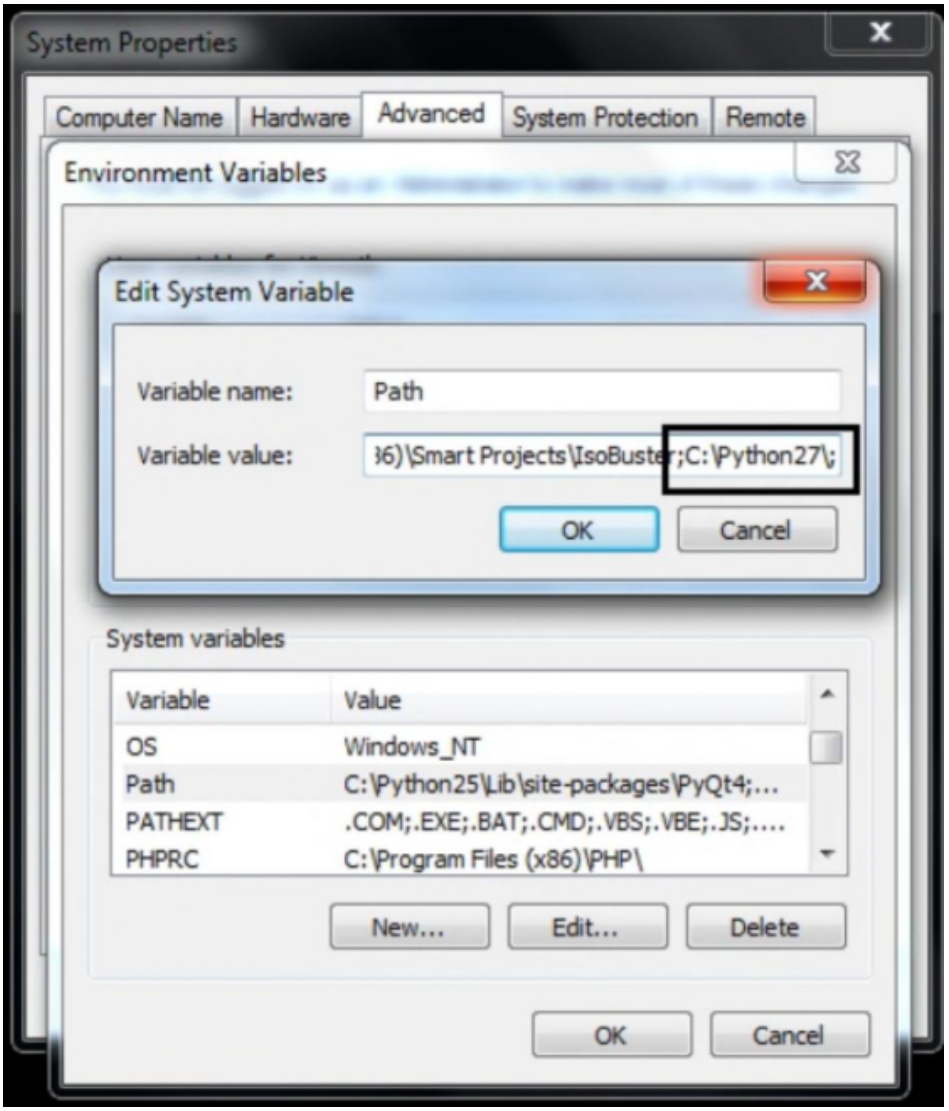

Figure 9 Προσθήκη της Python στα environment variables

Αφού ολοκληρωθεί η διαδικασία της διαδρομής της Python, σειρά έχει η εγκατάσταση του pip.

Εγκατάσταση του pip

Το pip με την εντολή install είναι από τις βασικές λειτουργίες που χρειάστηκαν για την εκκίνηση του project. Pip install και ακολουθεί το όνομα του πακέτου που θέλουμε να εγκατασταθεί. Το pip ψάχνει για το πακέτο στην PyPi (Python Package Index), υπολογίζει τις εξαρτήσεις του και τις εγκαθιστά για να εξασφαλίσει ότι τα αιτήματα θα λειτουργήσουν σωστά.

Αρχικά, επειδή πολλές φορές το pip εγκαθίσταται αυτόματα μαζί με την εγκατάσταση της Python, πρέπει να ελέγξουμε αν υπάρχει ήδη στον υπολογιστή μας. pip help είναι η εντολή που χρειάζεται στην παρούσα φάση. Αν τρέξει κανονικά, σημαίνει πως όλα είναι σωστά εγκατεστημένα. Αν όχι, θα εμφανιστεί ένα error με την παρακάτω μορφή.

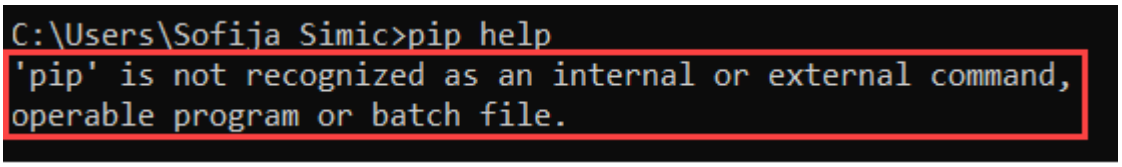

Figure 10 pip is not recognized error

Για την εγκατάσταση του pip λοιπόν, χρησιμοποιούμε την εντολή python get-pip.py .

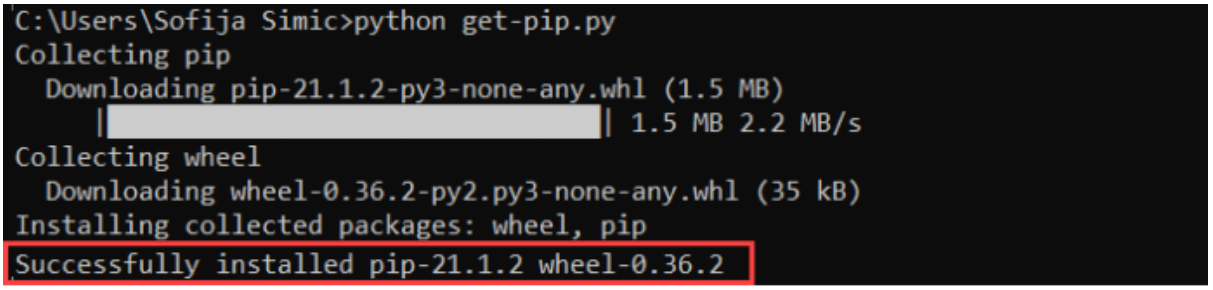

Figure 11 pip installed successfully

Εάν το αρχείο δεν βρεθεί, θα πρέπει να ξαναγίνει έλεγχος για τη διαδρομή προς το φάκελο όπου έγινε η αποθήκευση του αρχείο. Με την εντολή dir γίνεται η προβολή των περιεχομένων του τρέχοντος καταλόγου.
Αφού ολοκληρωθεί η εγκατάσταση, με την επαναχρησιμοποίηση της εντολής pip help πρέπει να ξαναγίνει έλεγχος για επιβεβαίωση. Αν υπάρξει κάποιο error θα πρέπει να επαναληφθεί η διαδικασία της εγκατάστασης. Τέλος, όπως και στην Python, έτσι και εδώ, θα πρέπει να υπάρχει το pip στις μεταβλητές περιβάλλοντος των windows. Οπότε με ακριβώς την ίδια διαδικασία θα πρέπει να γίνει η πρόσθεση του pip στις διαδρομές.

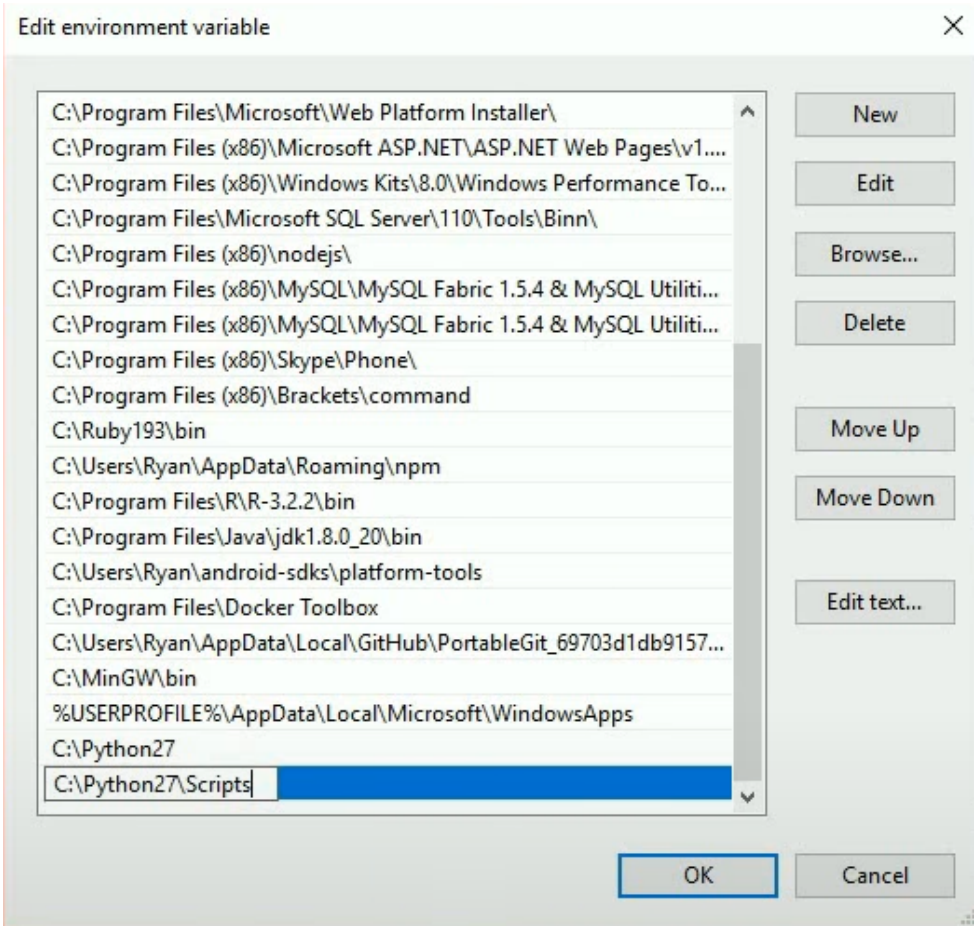

Figure 12 Προσθήκη του pip στα environment variables

Βημα 1 : Beautifulsoup και Requests

Η Beautifulsoup είναι μία βιβλιοθήκη της Python η οποία 'τραβάει' δεδομένα μέσα από σελίδες HTML ή XML αρχεία. Είναι από τις πιο διαδεδομένες βιβλιοθήκες που χρησιμοποιούνται στο W.S. και γενικά εξοικονομεί ώρες ή ημέρες εργασίας στους προγραμματιστές. Το W.S. της παρούσας Δ.Ε. έχει υλοποιηθεί με την beautifulSoup και με *requests*. Η βιβλιοθήκη requests της Python επιστρέφει στον προγραμματιστή μια ιστοσελίδα με χρήση της μεθόδου get() στο URL της. Η απόκριση response περιλαμβάνει πολλά πράγματα, αλλά η response.content επιστρέφει τον HTML κώδικα. Αφού υπάρχει πρόσβαση στην HTML τότε ξεκινάει η ανάλυση στα ενδιαφέροντα δεδομένα. Πρώτο βήμα λοιπόν, η εγκατάσταση του beautifulSoup και των requests.

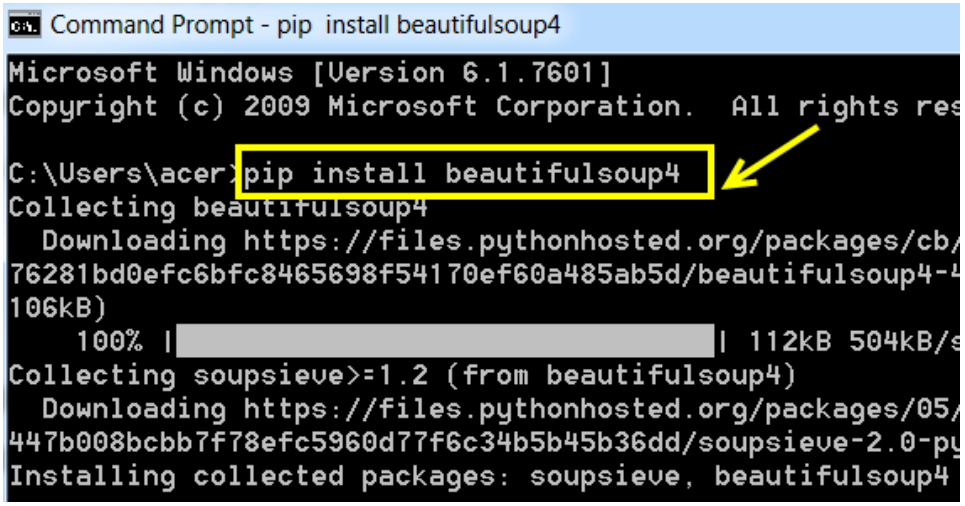

Figure 13 pip install bs4

Με χρήση του pip ολοκληρώνεται η εγκατάσταση των απαιτούμενων βιβλιοθηκών. Στην παραπάνω εικόνα, απεικονίζεται ο τρόπος εγκατάστασης. Παρομοίως, και για τις υπόλοιπες βιβλιοθήκες που θα χρειαστούν.

Βήμα 2 : Σύνδεση με τη βάση δεδομένων

Αρχικά για να υπάρχει πρόσβαση στη MongoDB θα πρέπει να γίνει και η σύνδεση με τη βάση. Για να συμβεί αυτό, πρώτο βήμα είναι να εγκατασταθεί το πακέτο της pymongo, με την εντολή `pip install pymongo` . Στη συνέχεια πρέπει να γίνει import της βιβλιοθήκης χρησιμοποιώντας την εντολή `from pymongo import MongoClient` . Αφού ολοκληρωθεί η δήλωση του cluster, ακολουθεί και η δήλωση της βάσης και της συλλογής (collection), οπου περιλαμβάνει το document `basic_datas` . Η όλη διαδικασία αναπαριστάται στην εικόνα παρακάτω.

```
1  from pymongo import MongoClient
2
3  #db
4  cluster = MongoClient("mongodb+srv://Vasiloudis:Vasiloudis@myCluster.bjuk6.mongodb.net/booking?ssl=true&ssl_cert_reqs=CERT_NONE")
5  my_db = cluster["booking"]
6  basic_datas = my_db["basic_datas"]
```

Figure 14 Σύνδεση με MongoDB

Βήμα 3 : Δημιουργία του βασικού link

Στη βάση δεδομένων, υπάρχει ένα document με τα βασικά δεδομένα τα οποία έχει δώσει ο χρήστης στο front-end της εφαρμογής, με σκοπό τη δημιουργία του link στο booking.com, απ' το οποίο θα αντληθούν τα δεδομένα. Παρακάτω απεικονίζεται αυτό το document με όνομα `basic_datas` .

Figure 15 basic_datas document

Τα παραπάνω δεδομένα, θα χρησιμοποιηθούν ως παράμετροι για να εισέλθουν στο αρχικό link του booking.com.

```
link = "https://www.booking.com/searchresults.html?checkin_month={in_month}&checkin_monthday={in_day}" \
    "&checkin_year={in_year}&checkout_month={out_month}&checkout_monthday={out_day}&checkout_year={out_year}" \
    "&group_adults={people}&group_children=0&order=price&ss={city}%2C%20{country}" \
    ";changed_currency=1;selected_currency=EUR;top_currency=1" \
        .format(in_month=my_db.basic_datas.find_one()['in_month'],
                in_day=str(today),#my_db.basic_datas.find_one()['in_day'],
                in_year=my_db.basic_datas.find_one()['in_year'],
                out_month=my_db.basic_datas.find_one()['out_month'],
                out_day=str(tomorrow),#my_db.basic_datas.find_one()['out_day'],
                out_year=my_db.basic_datas.find_one()['out_year'],
                people=my_db.basic_datas.find_one()['people'],
                city=my_db.basic_datas.find_one()['city'],
                country=my_db.basic_datas.find_one()['country'])
```

Figure 16 Δημιουργία του βασικού link

Στην παραπάνω εικόνα, γίνεται χρήση της μεθόδου format() , έτσι ώστε όπου υπάρχουν παρενθέσεις {temp}στο string του link, να αντικαθιστούνται με το temp που δηλώνεται μέσα στη μέθοδο. Όλες οι μεταβλητές λαμβάνονται από το document basic_datas, της βάσης, με τη μέθοδο find_one() , εκτός από τις in_day και out_day . Επειδή το πρόγραμμα τρέχει καθημερινά για να σκραπάρει δεδομένα μέρα με τη μέρα, οι δύο παραπάνω μεταβλητές αντιστοιχούν στη σημερινή μέρα και την επομένη (check-in/check-out). Η μεταβλητή today δημιουργείται με τον ακόλουθο τρόπο:

```
today = int(str(date.today()).split('-')[2])
```

Αρχικά, πρέπει να γίνει import της βιβλιοθήκης για το date. Αρα:

```
from datetime import date
```

Με τη μέθοδο date.today() επιστρέφεται η σημερινή ημερομηνία (π.χ. 2022-1-21). Γίνεται μετατροπή της μεταβλητής σε string έτσι ώστε να γίνει χρήση της split('-') μεθόδου, η οποία 'σπάει' το αρχικό string σε κομμάτια string, διαχωρίζοντας τα απο παύλε, και εντάσσοντας σε μια λίστα. Με το [2] γίνεται αναφορά στην τρίτη μεταβλητή της λίστας, άρα το 21 στο παράδειγμα που αναφέρθηκε. Έτσι λοιπόν λαμβάνεται η τιμή της ημέρας. Αφού λοιπόν 'τραβηχτούν' οι τιμές απο τη MongoDB θα παραχθεί το ακόλουθο link:
https://www.booking.com/searchresults.html?checkin_month=2&checkin_monthday=21&checkin_year=2022&checkout_month=2&checkout_monthday=22&checkout_year=2022&group_adults=2&group_children=0&order=price&ss=Thessaloniki%2C%20Greece;changed_currency=1;selected_currency=EUR;top_currency=1

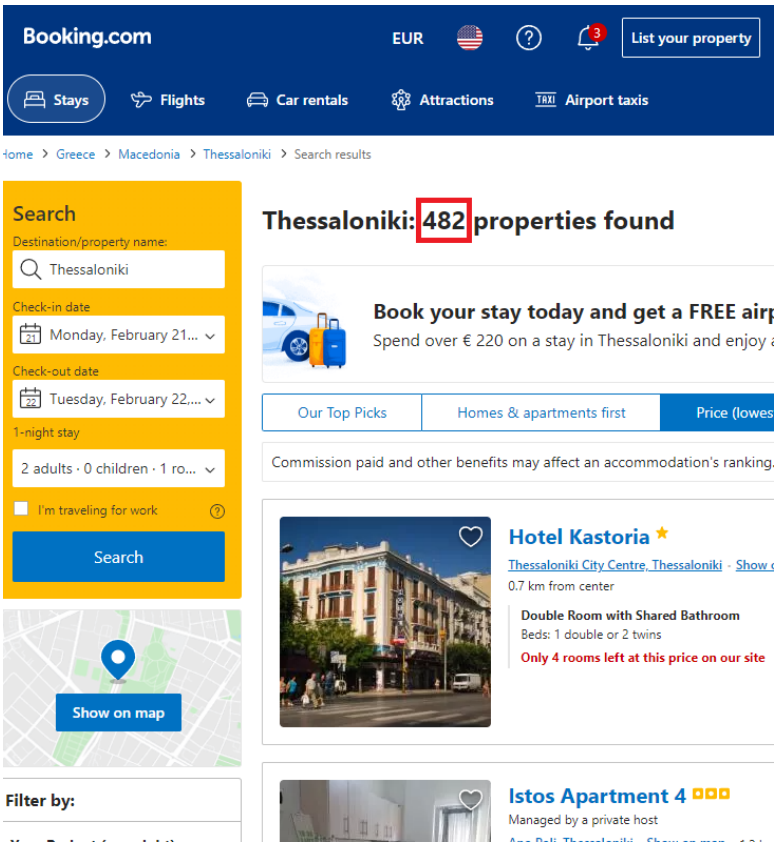

Figure 17 Στιγμιότυπο του βασικού link στο booking.com

Αρχικός στόχος λοιπόν, είναι μέσω του παραπάνω link, να συλλεχθούν όλα τα link των καταλυμάτων και μέσω αυτών, οι πληροφορίες που απαιτούνται.

Βήμα 4 : Συλλογή συνδέσμου του κάθε καταλύματος

Σε αυτό το βήμα θα συλλεχθούν 482 links, που αντιστοιχούν στα καταλύματα της εικόνας 17 και θα προστεθούν σε μια λίστα, για τη συνέχεια της υλοποίησης. Για να γίνει όμως αυτό, πρέπει πρώτα να παρατηρηθεί, ότι τα καταλύματα μοιράζονται , εικοσιπέντε ανα σελίδα, στο booking. Αυτό σημαίνει πως ο σκράπερ θα πρέπει να περάσει από όλες τις σελίδες έτσι ώστε να συλλέγξει όλα τα links. Σημαντική διαπίστωση που πρέπει να γίνει είναι ο τρόπος που αλλάζει το link από σελίδα σε σελίδα. Στο booking.com παρατηρήθηκε πως στη δεύτερη σελίδα, απλά προστίθεται στο link το &offset=25 , στην τρίτη έγινε &offset=50 και ούτω καθεξής. Αρα ανά σελίδα το offset γίνεται +25.
Ξεκινώντας με τη διαδικασία του W.S. λοιπον, λαμβάνεται το string της εικόνας 17 *"Thessaloniki: 482 properties found"*. Με δεξί κλικ και πατώντας inspect elements στην ιστοσελίδα του βασικού link, ανοίγει το περιεχόμενο της ιστοσελίδας. Παρατηρείται πως το ζητούμενο string εντάσσεται μέσα σε ένα h1 tag :
<h1 class="_30227359d _0db903e42">Θεσσαλονίκη: Βρέθηκαν 482 καταλύματα</h1>
Παρακάτω παρουσιάζεται ο κώδικας της διαδικασίας και στη συνέχεια η ανάλυση και η επεξήγησή του.

```
import requests
from bs4 import BeautifulSoup as bs
import numpy as np
import re
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.101
Safari/537.36"} # Windows 10 with Google Chrome
main_page = requests.get(link, headers = headers)
#Convert to bs object
soup = bs(main_page.content, 'lxml') #html.parser
properties = soup.find('h1', class_='_30227359d').text.strip()
number = re.sub("[^0-9]", "", properties)
loopnumber = round(np.ceil(int(float(number)/25))) #vriskw ton arithmo
twn selidwn gia na kanw loop oles tis selides
pages = np.arange(0, loopnumber+1, 1)
```

Αρχικά γίνονται τα κατάλληλα imports για της βιβλιοθήκες που θα χρησιμοποιηθούν. Έπειτα δηλώνονται τα headers, που συμπεριλαμβάνουν τον user-agent, και θα σταλούν σαν request σε εναν web server. Ο web server χρησιμοποιεί τις πληροφορίες του user-agent για να προσδιορίσει τον τύπο της συσκευής, την έκδοση του λειτουργικού συστήματος και το πρόγραμμα περιήγησης που χρησιμοποιείται. Για παράδειγμα, για Google Chrome σε Windows 10 ο user-agent είναι ο εξής: *"Mozilla/5.0 (Windows NT 10.0; Win64; x64)"*.
Στη συνέχεια, γίνεται το αίτημα get με χρήση των requests, με παραμέτρους το βασικό link και τα headers, Το αίτημα αποθηκεύεται στη μεταβλητή main_page. Για να ξεκινήσει το scraping πρέπει η μεταβλητή main_page να μετατραπεί σε ένα αντικείμενο beautifulSoup. Με χρήση του bs() και περνώντας παραμετρικά το main_page.content που είναι το περιεχόμενο HTML του main_page,

καθώς και το ‘lxml’ όπου είναι ο html-parser. Η lxml είναι μια βιβλιοθήκη της Python που επιτρέπει τον εύκολο χειρισμό αρχείων XML και HTML.

```
properties = soup.find('h1', class_='_30227359d').text.strip()
number = re.sub("[^0-9]", "", properties)
```

Στις παραπάνω δύο γραμμές του κώδικα χρησιμοποιώντας το soup.find() , αποκτάται πρόσβαση, όπως προαναφέρθηκε, στο σημείο *“Thessaloniki: 482 properties found”* της ιστοσελίδας, και έτσι αποθηκεύεται το περιεχόμενο, στη μεταβλητή properties. Ότι σκραπάρεται από μια ιστοσελίδα επιστρέφεται σε μορφή string. Με τη μέθοδο της sub(), από τη βιβλιοθήκη re της python, χωρίζεται το string και αποθηκεύεται στην μεταβλητή number μόνο ο αριθμός, αρα το 482. Ακολουθεί η διαίρεση με τον αριθμό των καταλυμάτων ανα σελίδα, αρα 25, και έτσι βγαίνει ο αριθμός τον σελίδων. Τέλος, από τη βιβλιοθήκη numpy της Python χρησιμοποιείται η μέθοδος arange() με σκοπό να δημιουργηθεί μια λίστα με πρώτο στοιχεία το 0, τελευταίο τον αριθμό των σελίδων, και βήμα 1 (π.χ. [0,1,2,3,4,5] αν ο αριθμός των σελίδων είναι 5).

```
for page in pages:
    temp_link = link+"&rows=25&offset="+str(page*25)
    #print(temp_link)
    offset_page = requests.get(temp_link, headers = headers)
    soup = bs(offset_page.content, 'lxml')
    #ftiaxnw lista me ola ta urls twn ksenodoxeiwn
    for a in soup.find_all('a', {'class':'fb01724e5b'}):
        links.insert(0, a['href'].strip())
```

Μέσα σε ένα for loop που κάνει ένα “πέρασμα” σε όλες τις σελίδες του βασικού link, δημιουργείται μια λίστα, στην οποία θα προστεθούν όλα τα links των καταλυμάτων. Παρακάτω, επισυνάπτεται μια εικόνα που δείχνει το inspect element της σελίδας, και πιο συγκεκριμένα τα tags που περιλαμβάνουν το link του κάθε καταλύματος.

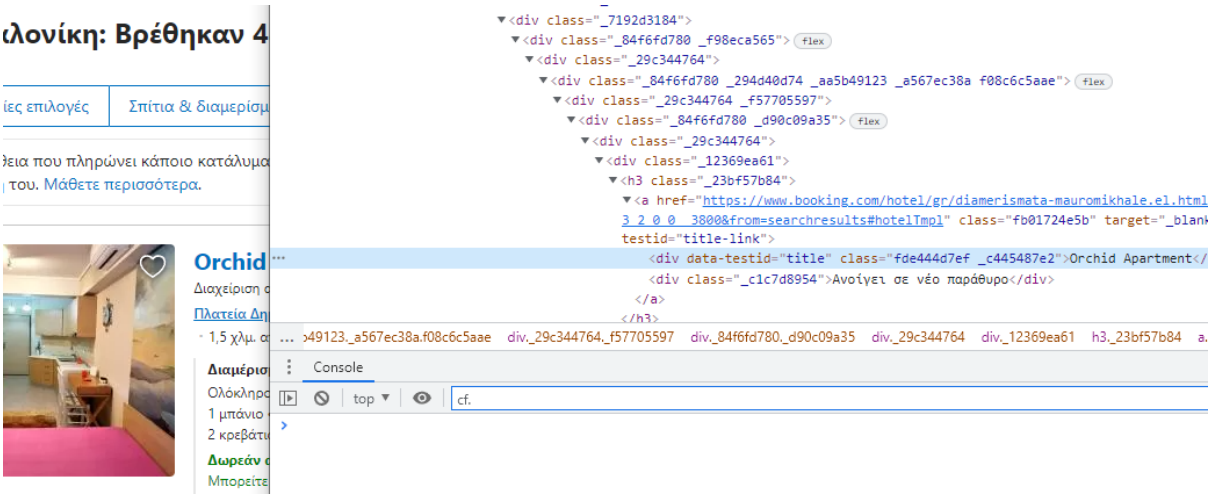

Figure 18 Στιγμιότυπο του inspect element του βασικού link

```
for a in soup.find_all('a', {'class':'fb01724e5b'}):
        links.insert(0, a['href'].strip())
```

Στις παραπάνω γραμμές κώδικα λοιπόν, χρησιμοποιείται η μέθοδος find_all() για να βρεθούν όλα τα *a tags* τα οποία ανήκουν στην κλάση *'fb01724e5b'* και κατέχουν τα ζητούμενα links. Στη συνέχεια με τη μέθοδο insert() και χρησιμοποιώντας το a['href'] εισάγονται σε μια λίστα με όνομα links τα urls του κάθε καταλύματος. Οπότε αφού το πρόγραμμα τρέξει για όλες τις σελίδες, η λίστα links θα έχει πλέον και τα 482 ζητούμενα links, άρα μέσα σε ένα while loop στη συνέχεια θα μπορεί να γίνει το web scraping για κάθε κατάλυμα.

Βήμα 5 : Scraping των δεδομένων

Σε αυτό το βήμα, το οποίο είναι και το πιό βασικό καθώς αποτελεί και το μεγαλύτερο κομμάτι της Δ.Ε. θα εκτελεστεί το W.S. για να ληφθούν τα απαραίτητα δεδομένα για τη σύγκριση των καταλυμάτων, τόσο ως προς την τιμή, όσο και ως προς το τι παρέχει το κάθε κατάλυμα. Τα δεδομένα τα οποία θα αντληθούν από την κάθε ιστοσελίδα είναι τα εξής:

- Το id του καταλύματος
- Το όνομα του καταλύματος
- Η εικόνα του καταλύματος
- Ο τύπος του καταλύματος (π.χ. ξενοδοχείο, διαμέρισμα κ.τ.λ.π.)
- Η διεύθυνση του καταλύματος
- Η βαθμολογία του καταλύματος
- Οι αξιολογήσεις του καταλύματος
- Τι παρέχει το κάθε κατάλυμα
- Το id του κάθε δωματίου του καταλύματος
- Τον τύπο του δωματίου
- Την τιμή του δωματίου
- Τον αριθμό των ατόμων που χωράει
- Τις επιλογές που παρέχει το δωμάτιο (π.χ. ακυρώσεις, πρωινά κ.τ.λ.π)
- Επιπλέον χαρακτηριστικά (π.χ. μπαλκόνι, ηλεκτρικές συσκευές, θέρμανση κ.α.)
- Βαθμολογίες πελατών (π.χ. στην τοποθεσία, στην καθαριότητα κ.α.)

Πιο συγκεκριμένα, παρακάτω κατατάσσονται τρεις εικόνες οι οποίες είναι είναι στιγμιότυπα από το ξενοδοχείο Stay Hybrid στο booking.com, και αναδεικνύουν τα σημεία που θα εκτελεστεί το W.S.

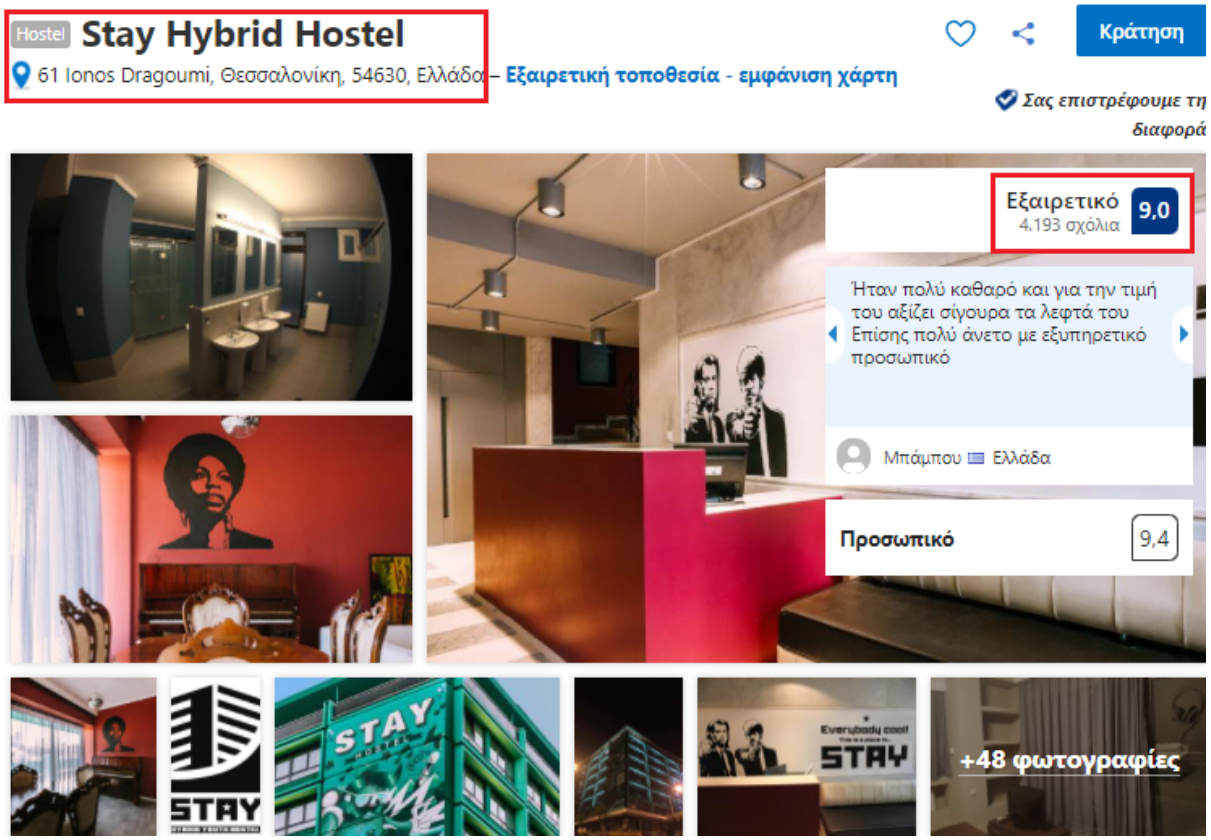

Figure 19 δεδομένα για scraping

Απο την εικόνα 19 τα δεδομένα που θα αντληθούν είναι ο τύπος του καταλύματος, το όνομά του, η διεύθυνση καθώς και η βαθμολογία μαζί με τις κριτικές.

Figure 20 πίνακας ενός καταλύματος στο booking.com

Από την εικόνα 20 τα δεδομένα που θα αντληθούν είναι όλα όσα απεικονίζονται, τα οποία αναρτώνται σε μορφή πίνακα.

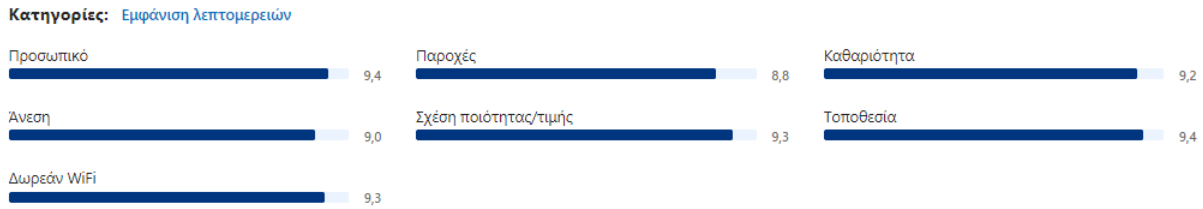

Figure 21 δεδομένα για scraping 2

Και τέλος, στην εικόνα 21 θα αντληθούν τα δεδομένα από τις μπάρες με τις αξιολογήσεις για γενικές πληροφορίες του καταλύματος. Επίσης θα εκτελεστεί η διαδικασία του W.S. σε δεδομένα τα οποία δεν είναι φανερά στο χρήστη.
Παρακάτω θα προβληθούν λίγα κομμάτια του κώδικα μαζί με την επεξήγηση τους, καθώς η διαδικασία είναι ίδια, για την άντληση των δεδομένων.

Ξεκινώντας χρησιμοποιείται ενα while loop με σκοπό να εκτελεστεί το W.S. για όλα τα links που αντιστοιχούν σε καταλύματα. Μέσα σε αυτό το loop, γράφονται οι παρακάτω εντολές για την άντληση των απαιτούμενων δεδομένων.

```
while i < len(links):
    page = requests.get(links[i], headers = headers)
    soup = bs(page.content, 'lxml')
```

Figure 22 while loop

Στη συνέχεια χρησιμοποιείται η split() συνάρτηση για να αποκοπεί το ασήμαντο μέρος του link για κάθε κατάλυμα. Για το hotel_id, επαναλαμβάνεται η ίδια διαδικασία. Πρέπει πρώτα να παρατηρηθεί ο κώδικας στο inspect element της σελίδας (εικόνα 24). Αφού λοιπόν εντοπιστεί το σημείο που αναζητείται, χρησιμοποιείται και η εντολή find() μαζί με το attrs[] για να γίνει σωστά η λήψη. Παρομοίως και για το link του εικονιδίου του καταλύματος θα ληφθεί το src.

```
hotel_link = links[i].split('?', 1)[0]
#print("Hotel link :"+ hotel_link)
hotel_id = soup.find('p',class_='hp-lists-counter').attrs['data-hotel-id'].strip()
#print("Hotel ID :"+ hotel_id)
img_link = soup.find('img', class_='hide').attrs['src']
#print("Image link :"+img_link)
```

Figure 23 Κώδικας για scraping δεδομένων

```
expand-list="2" data-show-text="1" data-placement="hp_title" data-position="bottom r
disable-tooltip data-title="Αποθήκευση" data-single-selection class="wl-entrypoint b
i-button--secondary bui-button--wide small " aria-label="Προσθέστε κατάλυμα στις λίσ
role="button" aria-live="assertive" data-loadingtext="Φόρτωση">…</button> flex
<p class="hp-lists-counter wishlist-social-count" data-component="wishlist/savers"
hotel-saved="0" data-hotel-id="2601480" data-hotel-count="2744">…</p> == $0
</div>
</div>
</div>
```

Figure 24 Inspect element της ιστοσελίδας για το hotel_id

```
if soup.find('div', class_='_4abc4c3d5'):
    reviews =soup.find('div',class_='_4abc4c3d5').text.strip().split()[0]
    if ',' in reviews:
        reviews = reviews.replace(",", "")
else:
    reviews = -1
#print(reviews)
```

Figure 25 Στιγμιότυπο κώδικα για το scraping των reviews

Ο παραπάνω κώδικας αναφέρεται στις αξιολογήσεις που κατέχει το κατάλυμα. Υπάρχει βέβαια και η περίπτωση, ενα κατάλυμα να μην έχει καμία αξιολόγηση. Σε αυτήν την περίπτωση επιστρέφεται η τιμή -1. Αφού η μεταβλητή των reviews επιστρέφεται σε string μετά το scraping, θα χρειαστεί να μετατραπεί σε int. ‘Ομως, οταν ο αριθμός των αξιολογήσεων είναι πάνω από 1,000 τότε υπάρχει και ένα κόμμα στη μεταβλητή. Γι΄ αυτόν το λόγο υπάρχει και το εσωτερικό if στην εικόνα 25, που το αφαιρεί.

Για τα δεδομένα της εικόνας 20, που εντάσσονται σε πίνακα στην ιστοσελίδα, ακολουθεί ο παρακάτω κώδικας.

```
table = soup.find('table', class_="hprt-table")
for row in table.find_all('tr', class_="js-rt-block-row"):
      roomId = row.attrs['data-block-id'].rsplit('_', 4)[0]
      #print("Room Id:",roomId)
```

Αρχικά, εκτελείται το W.S. σε ολόκληρο τον πίνακα, έτσι ώστε να μπορέσουν να αντληθούν τα δεδομένα του, ανά σειρά. Συνεχίζοντας μέσω ενός for loop διαβάζεται ο πίνακας σειρά-σειρά. Στον παραπάνω κώδικα παρατηρείται πως το roomId αντιστοιχεί σε ένα tag με attrs[‘data-block-id’]. Σε αυτό προσαρμόζεται και η μέθοδος rsplit(‘_’, 4) η οποία χωρίζει το string σε μια λίστα από elements και με τη χρήση του [0] αποθηκεύεται το πρώτο στοιχείο της λίστας στη μεταβλητή roomId. Παρακάτω παρουσιάζεται και ο κώδικας του inspect element του συγκεκριμένου element.

```
▶ <tr data-block-id="260148001_231800025_2_0_0" class="js-rt-block-row e2e-hprt-table-row ">…
</tr>
▶ <tr data-block-id="260148001_104764658_1_0_0" class="js-rt-block-row e2e-hprt-table-row ">…
</tr>
```

Στη συνέχεια θα υπάρξει μια τελευταία αναφορά στον κώδικα, όσον αφορά το κομμάτι του W.S. Το απόσπασμα του παρακάτω κώδικα εκτελείται επίσης μέσα στο for loop του πίνακα, και αναφέρεται στην εικόνα 26.

```
ul2 = row.find('td', class_="hprt-table-cell-conditions")
```

```
choices = []
for li2 in ul2.find_all('li', class_="bui-list__item"):
   choice = li2.find('div',class_="bui-list__description").text.strip()
   choices.append(choice)
```

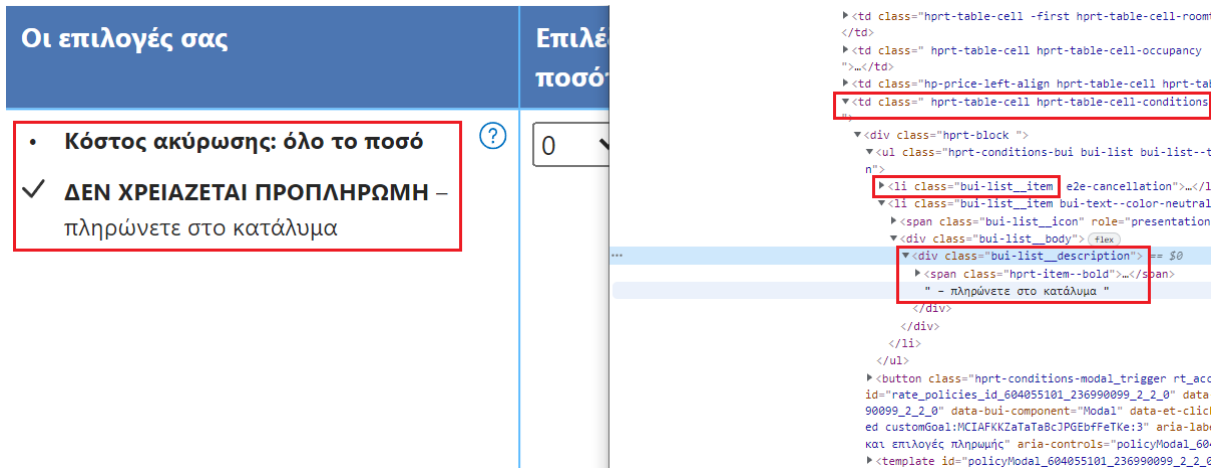

Figure 26 στιγμιότυπο booking.com/inspect element σελίδας

Αυτήν τη φορά, τα δεδομένα που πρέπει να ληφθούν, είναι το κελί του πίνακα "Οι επιλογές σας" της εικόνας 26 . Παρατηρείται ότι η δομή των δεδομένων είναι μέσα σε ένα ul, μοιρασμένα σε li. Και μέσα σε κάθε li είναι το κείμενο που πρέπει να αντληθεί. Συνεπώς. πρώτο βήμα είναι να γίνει το scraping στο ανώτερο στοιχείο και έπειτα μέσα σε ένα for loop να βρεθούν και τα εσωτερικά στοιχεία, με τελικό στόχο, το div που περιλαμβάνει το κείμενο. Έτσι και ολοκληρώνεται η διαδικασία του scraping με χρήση της find() και find_all() , καθώς και η διαδικασία της αποθήκευσης σε πίνακα με τη μέθοδο append().

Βήμα 6 : Κατανομή των δεδομένων στη βάση

Αφού λοιπόν ολοκληρωθεί η διαδικασία του W.S., σειρά έχει η κατανομή τους στη βάση δεδομένων. Θα πρέπει να έχει εφαρμοστεί η διαδικασία του βήματος 2, η οποία είναι η σύνδεση με τη MongoDB. Συνεπώς τα δεδομένα είναι πλέον έτοιμα να ανέβουν στη βάση. Πρίν όμως γίνει αυτό, θα πρέπει τα δεδομένα να συνταχθούν σε πίνακες για να ενταχθούν σωστά στη MongoDB σε μορφή document, όπως έχει προαναφερθεί. Παρακάτω επισυνάπτεται το JSON μοντέλο της βάσης, που χρησιμοποιήθηκε ως οδηγός για την ορθή ομαδοποίηση των δεδομένων.

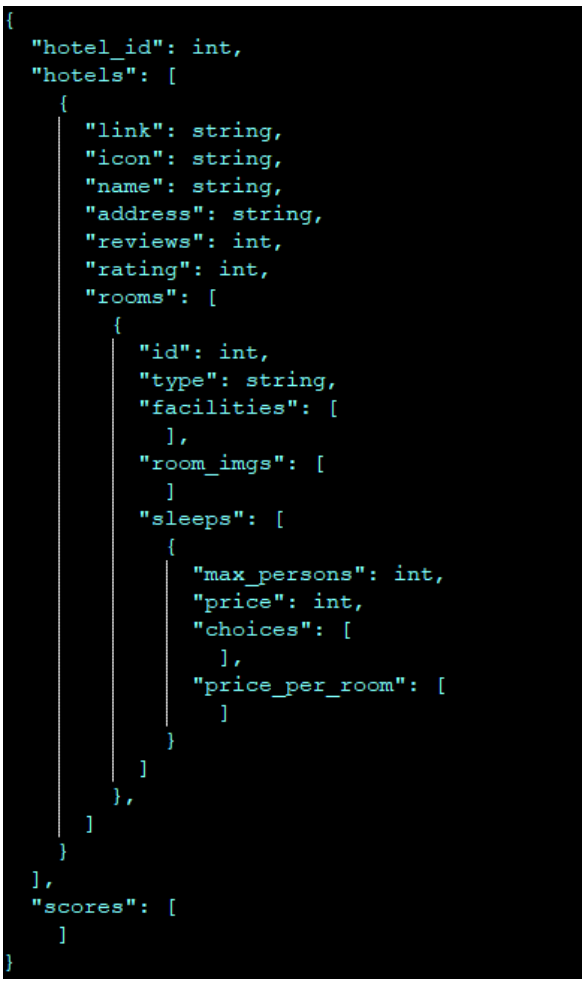

Figure 27 JSON μοντέλο βάσης

Με μία γρήγορη ματιά στην εικόνα 27, βγαίνει το συμπέρασμα πως, πολλά από τα δεδομένα είναι εμφωλευμένα σε μεταβλητές, οι οποίες αποτελούν έναν πίνακα αντικειμένων. Για παράδειγμα, η μεταβλητή rooms αποτελεί έναν πίνακα αντικειμένων, που περιέχει τα πεδία id, type, facilities, room_imgs και sleeps. Κάποια από τα προαναφερόμενα πεδία αποτελούν επίσης πίνακες αντικειμένων. Προχωρώντας λοιπόν, θα γίνει μια αναφορά στον κώδικα που αφορά την ομαδοποίηση των δεδομένων για την σωστή ένταξη στη βάση δεδομένων.

```
hotel = {"days": int(str(total_days).split(' ')[0])+1,"link":
hotel_link, "icon": img_link, "name": name, "type": type, "address":
address, "reviews": int(reviews), "rating": float(rating), "rooms":
rooms}
```

Στο παραπάνω κομμάτι του κώδικα, παρουσιάζεται η τελική μορφή της μεταβλητής hotel, που θα εισαχθεί στη βάση με όλα τα δεδομένα που συμπεριλαμβάνει. Όλες είναι απλές μεταβλητές, πέρα από τη rooms, η οποία αποτελεί έναν πίνακα αντικειμένων.

```
rooms += [ {"id": roomId, "type": roomType,"facilities":
facilities,"room_imgs":imgs, "sleeps": sleeps}]
```

Η rooms εντάσσεται μέσα στο for loop το οποίο διαβάζει τις σειρές του πίνακα για όλα τα δωμάτια του κάθε καταλύματος. Με τον ίδιο τρόπο, όπως και στην μεταβλητή hotel, εισάγονται όλες οι μεταβλητές, μαζί και η μεταβλητή sleeps, η οποία είναι ξεχωριστός πίνακας αντικειμένων.

```
sleeps = [ {"max_persons": int(sleep), "price": int(price), "choices":
choices," price_per_room": price_per_room} ]
```

Αφού λοιπόν ολοκληρωθεί η διαδικασία της ομαδοποίησης των δεδομένων, σειρά έχει η εισαγωγή τους στη MongoDB. Με απλές εντολές της βιβλιοθήκης pymongo ακολουθεί ο κώδικας της εισαγωγής των δεδομένων. Αρχικά πρέπει να γίνει η αναφορά στο collection της βάσης:

```
cluster =
MongoClient("mongodb+srv://Vasiloudis:Vasiloudis@myCluster.bjuk6.mongod
b.net/booking?ssl=true&ssl_cert_reqs=CERT_NONE")
my_db = cluster["booking"]
my_collection = my_db["hotels"]
```

Όταν ολοκληρωθούν τα παραπάνω βήματα, πλέον μπορεί να γίνει το update στην MongoDB.

```
 my_collection.update_one({'hotel_id':int(hotel_id)},{"$set":
{"scores": scores, "hotel": hotel}}, True)
```

Η παραπάνω εντολή είναι ο τρόπος με τον οποίο τα δεδομένα εισάγονται στη βάση. Η τρίτη παράμετρος της μεθόδου, η οποία έχει τιμή *True*, αναφέρεται στο upsert. Άμα δηλαδή τα δεδομένα δεν υπάρχουν στη βάση σε μορφή document, τότε γίνονται insert. Αλλιώς αν το document υπάρχει ήδη στη βάση, τότε το κάνει update και ενημερώνονται οι μεταβλητές οι οποίες έχουν διαφορετική τιμή. Η πρώτη παράμετρος της μεθόδου update_one() χρησιμοποιείται σαν λέξη κλειδί. Έτσι, το πρόγραμμα ψάχνει στη βάση αν υπάρχει το document με το συγκεκριμένο κλειδί. Αν το βρεί, τότε εκτελείται το upsert. Αν όχι, τότε θα γίνει ένα απλό insert. Σε αυτό το project χρησιμοποιείται το hotel_id σαν κλειδί, αφού άλλωστε είναι και μοναδικό για κάθε κατάλυμα.
Παρακάτω παρουσιάζεται το στιγμιότυπο του collection *hotels* της MongoDB με το σύνολο των documents που αντιστοιχούν στα καταλύματα.

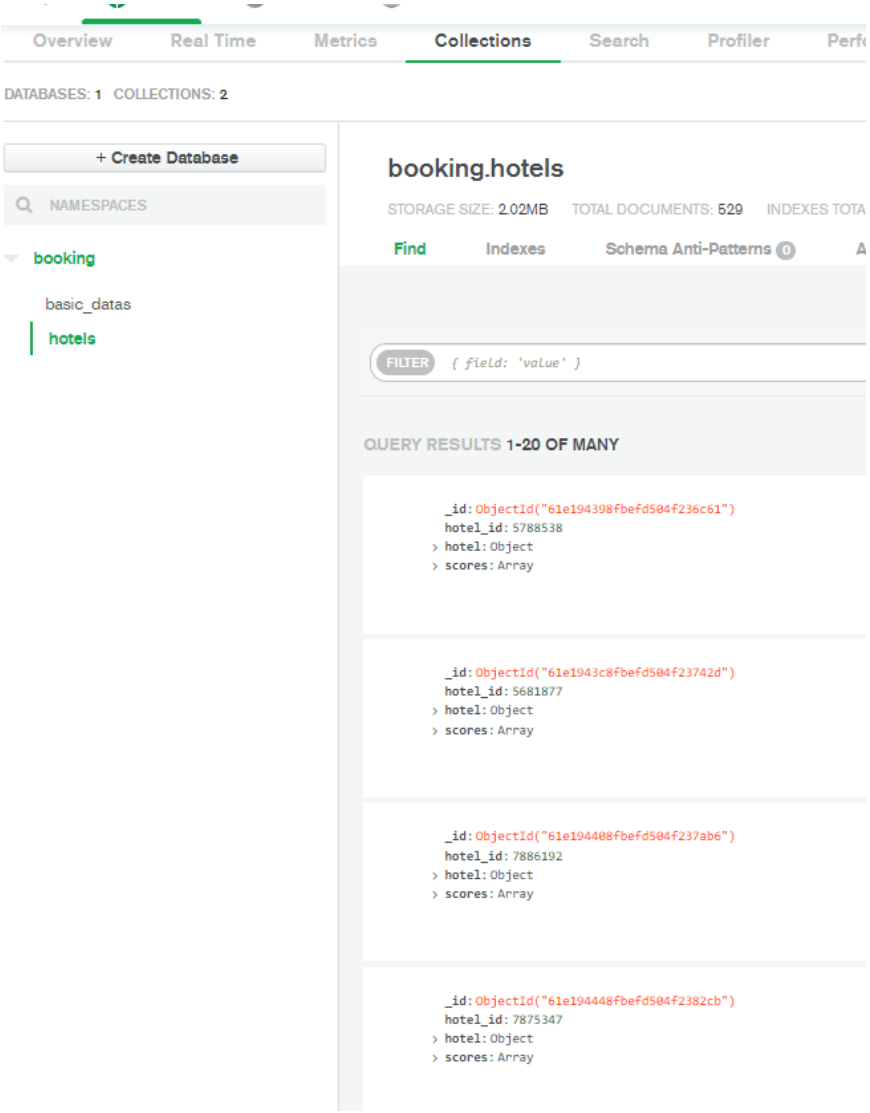

Figure 28 Στιγμιότυπο του hotels collection στη MongoDB

Και τέλος ένα στιγμιότυπο ενός document, αναλυτικά με όλα τα πεδία του.

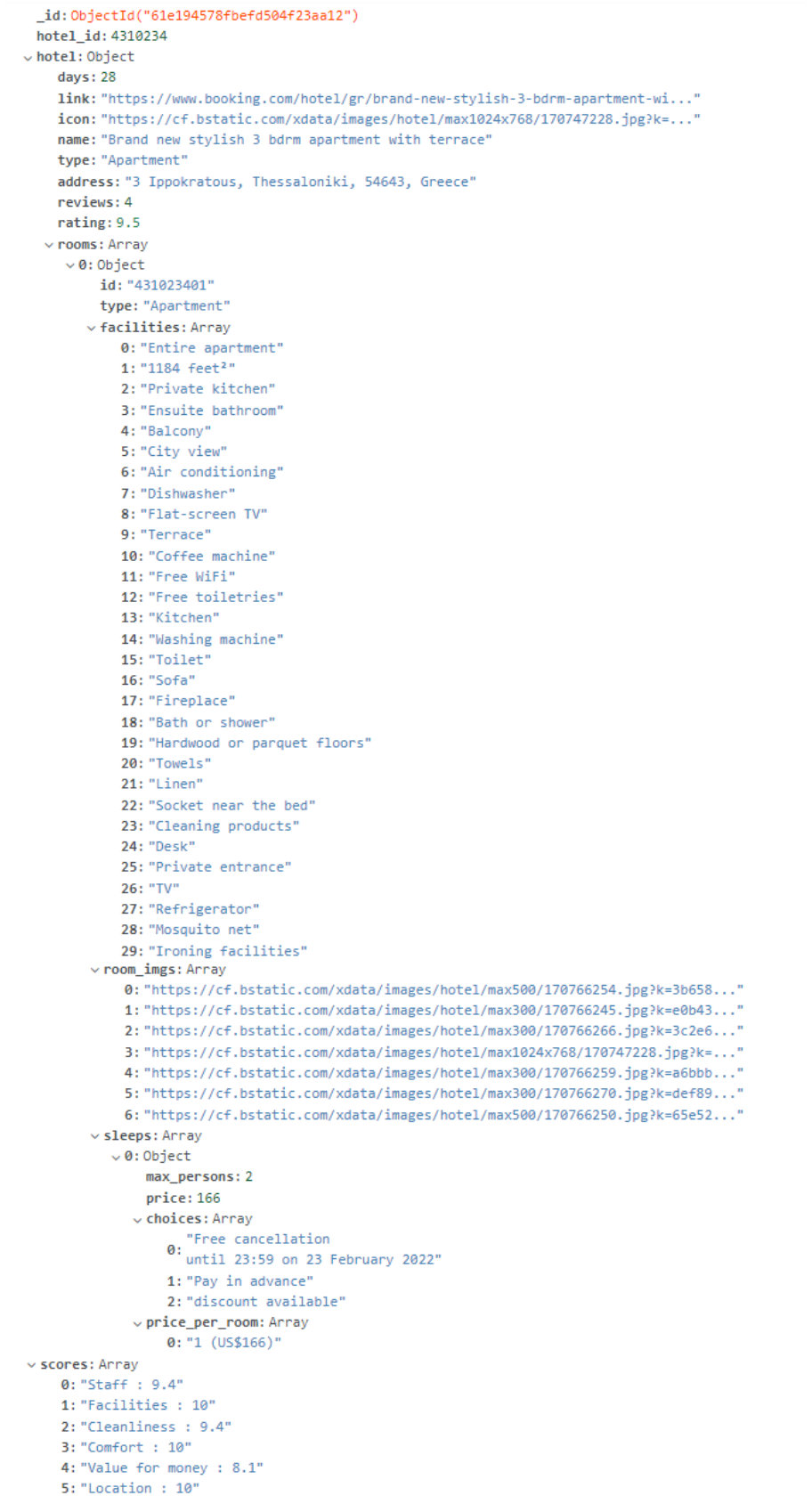

Figure 29 document ενός καταλύματος

Βήμα 7 : Εξαγωγή των δεδομένων

Πέρα από την εισαγωγή των δεδομένων στη βάση, στη συγκεκριμένη εργασία, είναι σημαντικό τα δεδομένα να αποθηκεύονται και εξωτερικά σε κάποιο αρχείο. Ο λόγος είναι κυρίως επειδή οι τιμές γίνονται update κάθε μέρα και έτσι υπάρχουν αλλαγές στη βάση. Επομένως, είναι υποχρεωτικό τα δεδομένα να εξάγονται σε κάποιο αρχείο, για να υπάρχουν όλες οι πληροφορίες του παρελθόντος, και να μη χάνονται. Στην παρούσα Δ.Ε. χρησιμοποιήθηκε το csv ως αρχείο για αποθήκευση, διότι είναι κατάλληλο για μελλοντικές χρήσεις του. Παρακάτω θα γίνει μια αναφορά σε αποσπάσματα και στιγμιότυπα του κώδικα καθώς και του αρχείου csv.

```
import csv
data = [todate,name,hotel_id,roomType,roomId,price]
with open('dpr/'+todate+'-Rooms.csv', 'a', encoding='UTF8', newline='')
as f:
    writer = csv.writer(f)
    writer.writerow(data)
```

Figure 30 Κώδικας για εξαγωγή σε csv

Τα βήματα για την εξαγωγή των δεδομένων σε csv είναι πολύ απλά. Ακολουθούν παρακάτω:

1. Import της βιβλιοθήκης csv
2. open() για τη δημιουργία του αρχείου
3. Δημιουργία του writer για την εγγραφή στο αρχείο
4. Χρήση της writerow() για την προσθήκη των δεδομένων

Στην εικόνα 30 δηλώνονται τα δεδομένα, που θα γραφτούν στο csv αρχείο, ως data. Το πρόγραμμα τρέχει καθημερινά. Αρα ένα νέο αρχείο δημιουργείται ανά ημέρα, με όνομα την τωρινή ημερομηνία (π.χ. 1/27/2022-Rooms.csv), για να αποθηκεύσει τις τιμές της ίδιας ημέρας (δηλ. 1/27/2022). Επίσης στην open() μέθοδο, δίνεται και το ‘a’ σαν παράμετρος που αναφέρεται στα δικαιώματα του αρχείου, και αντιστοιχεί σε όλα τα δικαιώματα. Το encoding=’UTF8’ χρησιμοποιείται για τους non_ASCII χαρακτήρες, και το newline=’’ χρησιμοποιείται για να μην υπάρχει κενό ανάμεσα από κάθε γραμμή.

Παρακάτω εντάσσεται και η εικόνα με ένα αρχείο csv, ως παράδειγμα.

	A	B	C	D	E	F
1	Date	Hotel Name	Hotel ID	Room Name	Room ID	Price
2	2022-01-14	Apartment with one be	5788538	One-Bedroom A	578853802	556
3	2022-01-14	#Icarus Penthouse by	5681877	Penthouse Apar	568187701	245
4	2022-01-14	#Icarus Penthouse by	5681877	Penthouse Apar	568187701	254
5	2022-01-14	#Icarus Penthouse by	5681877	Penthouse Apar	568187701	264
6	2022-01-14	#Icarus Penthouse by	5681877	Penthouse Apar	568187701	273
7	2022-01-14	#Magnolia Apt by halu	7886192	Three-Bedroom	788619201	241
8	2022-01-14	#Magnolia Apt by halu	7886192	Three-Bedroom	788619201	229
9	2022-01-14	#Magnolia Apt by halu	7886192	Three-Bedroom	788619201	251
10	2022-01-14	#Magnolia Apt by halu	7886192	Three-Bedroom	788619201	261
11	2022-01-14	Porto Sea View Apart	7875347	Deluxe Suite wit	787534702	238
12	2022-01-14	Porto Sea View Apart	7875347	Junior Suite with	787534701	399
13	2022-01-14	#Luxlikehome - Aristo	5607919	Two-Bedroom A	560791901	227
14	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	212
15	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	233
16	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	258
17	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	279
18	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	300
19	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	325
20	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	347
21	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	368
22	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	393
23	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	414
24	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	435
25	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	460
26	2022-01-14	Traditional Fully Deta	7255954	Villa	725595401	481
27	2022-01-14	Saint Dimitrios Centra	3780532	Deluxe Apartme	378053201	202
28	2022-01-14	Saint Dimitrios Centra	3780532	Deluxe Apartme	378053201	216
29	2022-01-14	Saint Dimitrios Centra	3780532	Deluxe Apartme	378053201	230
30	2022-01-14	Saint Dimitrios Centra	3780532	Deluxe Apartme	378053201	244
31	2022-01-14	Saint Dimitrios Centra	3780532	Deluxe Apartme	378053201	257
32	2022-01-14	Saint Dimitrios Centra	3780532	Deluxe Apartme	378053201	276
33	2022-01-14	Saint Dimitrios Centra	3780532	Deluxe Apartme	378053201	294
34	2022-01-14	Saint Dimitrios Centra	3780532	Deluxe Apartme	378053201	313
35	2022-01-14	Saint Dimitrios Centra	3780532	Deluxe Apartme	378053201	331
36	2022-01-14	Saint Dimitrios Centra	3780532	Deluxe Apartme	378053201	349
37	2022-01-14	#Bel Air Penthouse by	7604901	Penthouse Apar	760490101	199
38	2022-01-14	Brand new stylish 3 b	4310234	Apartment	431023401	186
39	2022-01-14	Hyatt Regency Thess	17799	1 King Bed	1779903	179
40	2022-01-14	Hyatt Regency Thess	17799	1 King Bed	1779903	199
41	2022-01-14	Hyatt Regency Thess	17799	1 King Bed	1779903	209
42	2022-01-14	Hyatt Regency Thess	17799	1 King Bed	1779903	229
43	2022-01-14	Hyatt Regency Thess	17799	1 King Bed	1779903	166
44	2022-01-14	Hyatt Regency Thess	17799	1 King Bed	1779903	176

Figure 31 Εγγραφο csv μιας ημέρας

Βήμα 8 : Προσθήκη του GitHub Action

Αφού ολοκληρωθεί και η διαδικασία του upload στη βάση δεδομένων, τελευταίο βήμα είναι η προσθήκη του GitHub Action για την αυτοματοποιημένη λειτουργία του project. Παρακάτω θα αναφερθούν τα βήματα δημιουργίας του github action, καθώς και θα προβληθούν στιγμιότυπα του workflow που χρησιμοποιήθηκε στην παρούσα Δ.Ε.

Δημιουργία του workflow

1. Δημιουργία του .github/workflows φακέλου στο repository του GitHub αν δεν υπάρχει ήδη.
2. Δημιουργία του yml αρχείου μέσα στον παραπάνω φάκελο.
3. Συμπλήρωση του κώδικα του yml αρχείου.
4. Επιλογή του **Create a new branch for this commit and start a pull request.** Τέλος, η διαδικασία ολοκληρώνεται με κλικ στο **Propose new file.**

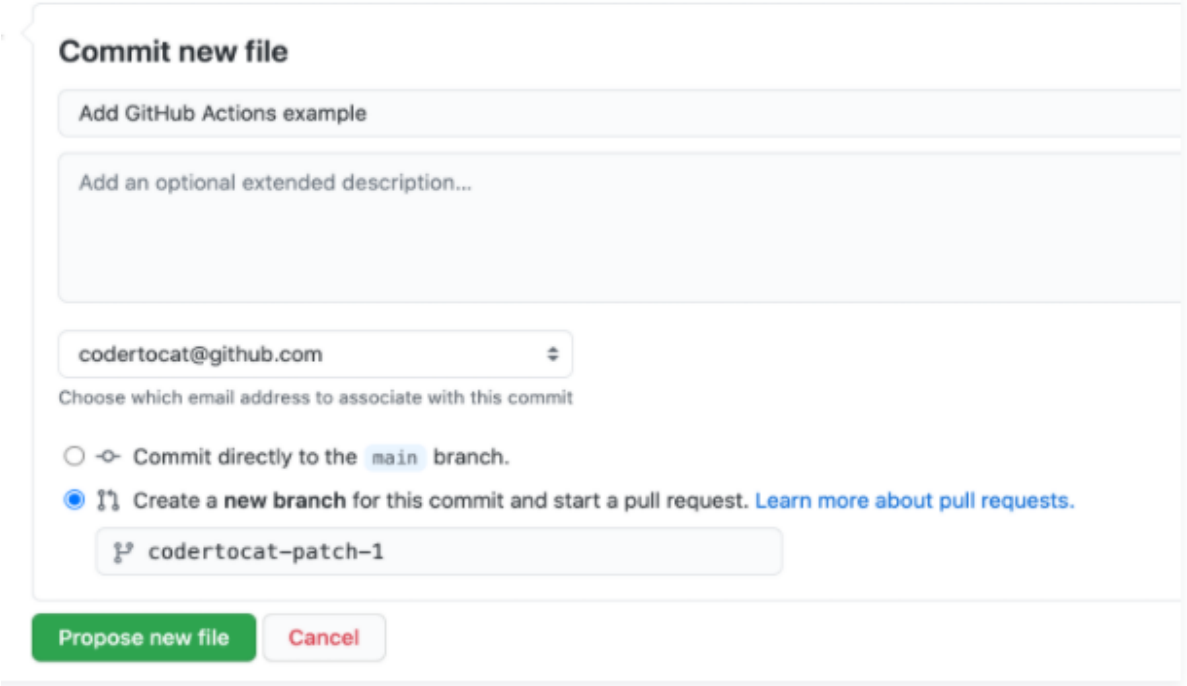

Figure 32 Δημιουργία του workflow

Κάνοντας commit το αρχείο του workflow σε κάποιο branch στο repository, εκτελείται το push και έτσι τρέχει το workflow.

Προβολή των αποτελεσμάτων του workflow

1. Μετάβαση στη κεντρική σελίδα του repository στο GitHub.com.
2. Επιλογή του Actions κάτω από το όνομα του repository.

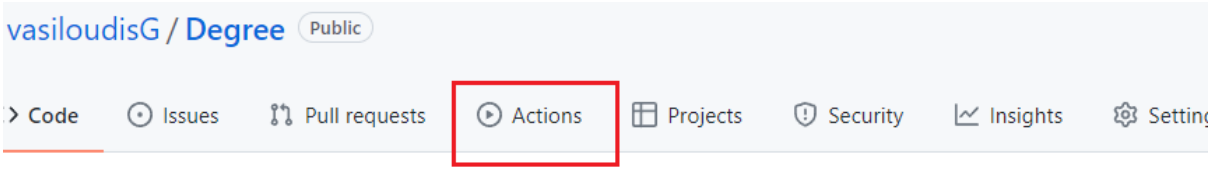

Figure 33 Επιλογή του Actions

3. Στα αριστερά, κλικ στο επιθυμητό workflow.

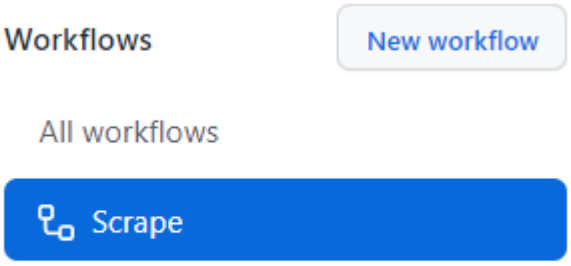

Figure 34 Επιλογή του επιθυμητού workflow

4. Επιλογή του επιθυμητού run, από τη λίστα των runs του workflow.

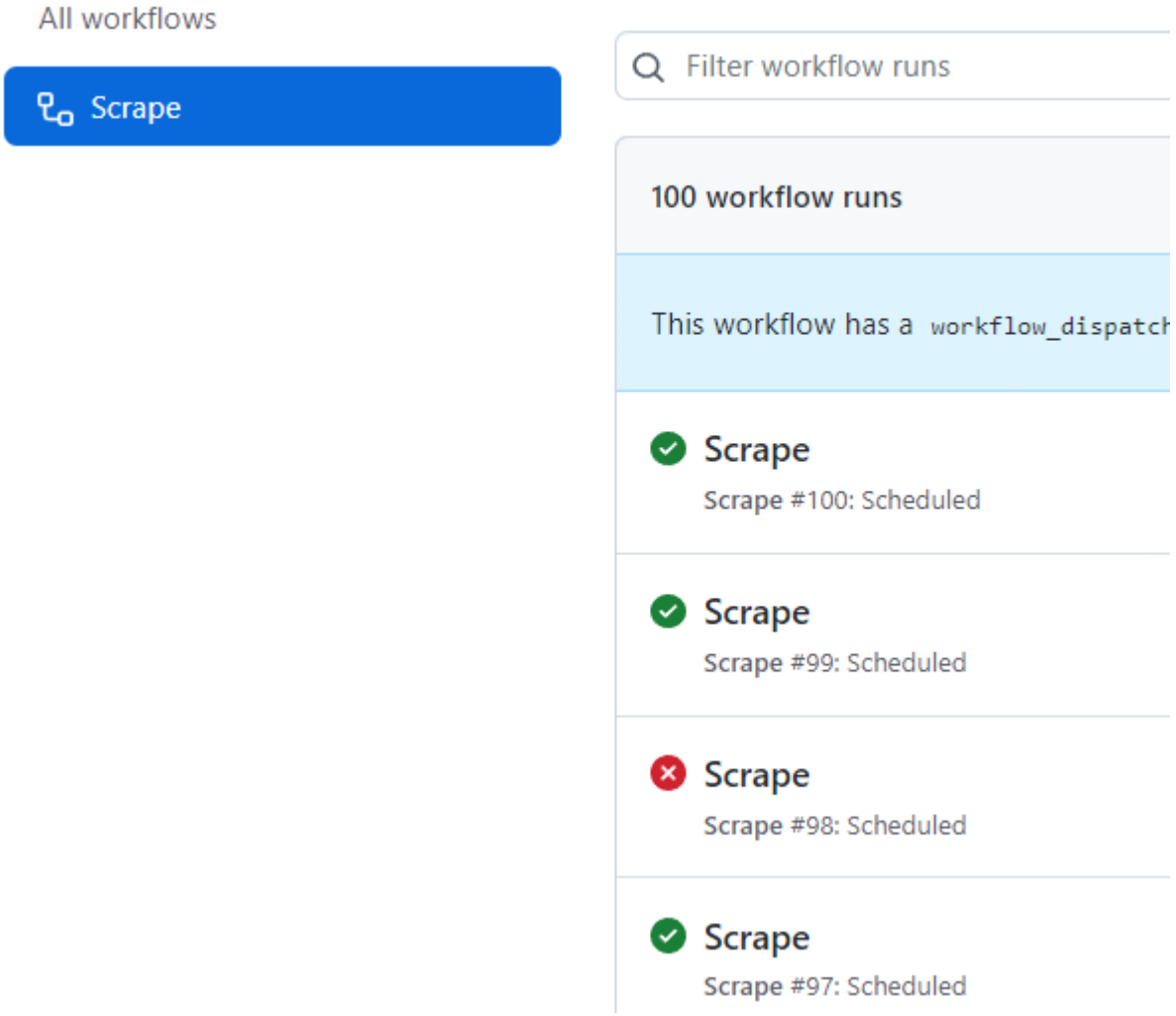

Figure 35 Λίστα με όλα τα workflows

5. Κλικ στο scrape-latest

Figure 36 Κλικ του workflow στα jobs

6. Ο πίνακας που εμφανίζεται, δείχνει όλα τα βήματα που ακολούθησε το workflow. Πατώντας πάνω σε κάποιο, θα προβληθούν οι λεπτομέρειες.

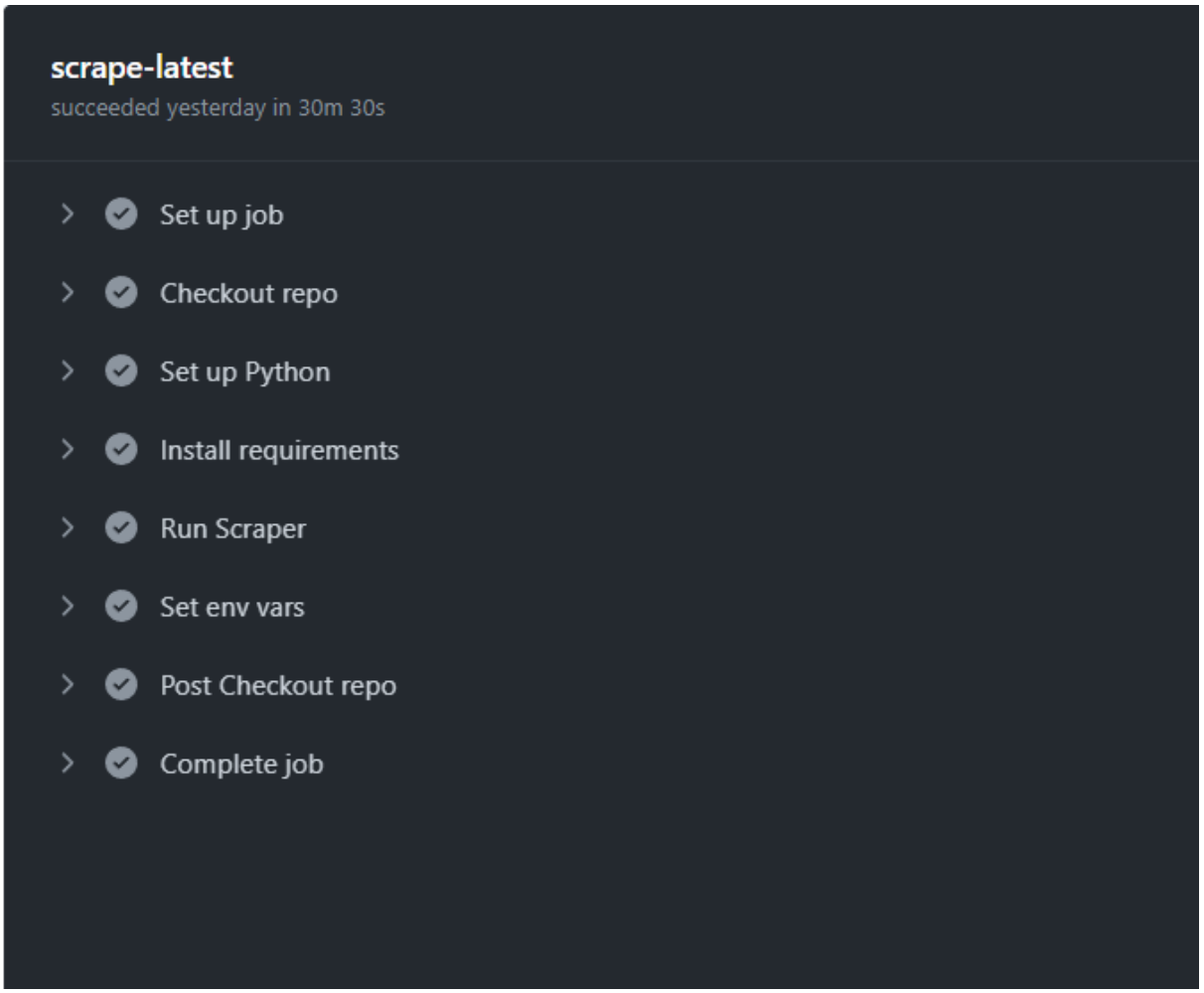

Figure 37 Βηματα του workflow

Το στιγμιότυπο 37 απεικονίζει τα βήματα του workflow που χρησιμοποιήθηκε στην εργασία. Παρατηρείται πως όλα τα βήματα ολοκληρώθηκαν με επιτυχία. Στην περίπτωση που υπάρχει κάποιο σφάλμα τότε απεικονίζεται κάπως έτσι:

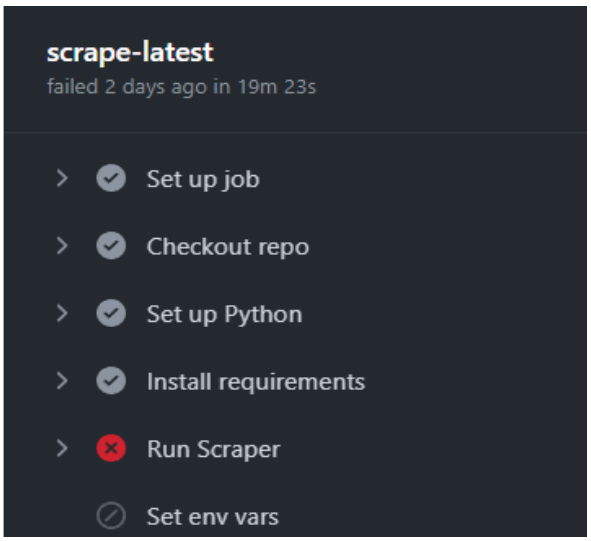

Figure 38 Σφάλμα στην εκτέλεση του workflow

Παρακάτω παρουσιάζεται και το στιγμιότυπο του yml αρχείου για καλύτερη κατανόηση.

```
9   name: Scrape
10
11  on:
12    schedule:
13      - cron: "0 12 * * *"
14    workflow_dispatch:
15
16  env:
17    ACTIONS_ALLOW_UNSECURE_COMMANDS: true
18
19  jobs:
20    scrape-latest:
21      runs-on: windows-latest
22
23      steps:
24        - name: Checkout repo
25          uses: actions/checkout@v2
26        - name: Set up Python
27          uses: actions/setup-python@v2.0.0
28          with:
29            python-version: '3.8'
30        - name: Install requirements
31          run: pip install -r requirements.txt
32        - name: Run Scraper
33          run: python booking_no_sel.py
34        - name: Set env vars
35          run: |
36            echo "DATE=$(python -c 'import datetime as dt; print(dt.datetime.now().date())')" >> $GITHUB_ENV
```

Figure 39 yml αρχείο

Το αρχείο είναι προγραμματισμένο να τρέχει κάθε μέρα στις 12:00 Μ.Μ. Στα steps βρίσκονται όλα τα βήματα που θα εκτελέσει, όπως διακρίνεται και στην εικόνα 39. Το όνομα του αρχείου που θα τρέχει αυτόματα μέσω του workflow είναι το booking_no_sel.py, και αναφέρεται στη γραμμή 33 της

παραπάνω εικόνας. Τέλος στη γραμμή 31, γίνονται install όλες οι απαραίτητες βιβλιοθήκες που θα χρειαστούν, και διαβάζονται μέσω του αρχείου requirements. Παρακάτω παρουσιάζεται το αρχείο requirements.txt.

```
26 lines (26 sloc)  |  874 Bytes
 1  appdirs==1.4.4
 2  beautifulsoup4==4.9.3
 3  bs4==0.0.1
 4  certifi==2020.12.5
 5  chardet == 4.0.0
 6  cssselect==1.1.0
 7  dnspython == 1.16.0
 8  fake-useragent==0.1.11
 9  idna==2.10
10  lxml==4.6.2
11  numpy==1.20.3
```

Figure 40 requirements.txt

Κεφάλαιο 5ο: Συμπέρασμα και προτάσεις βελτίωσης

Στο κεφάλαιο αυτό, παρουσιάζονται τα συμπεράσματα που προέκυψαν απο την υλοποίηση της διπλωματικής εργασίας. Τόσο στο θεωρητικό, όσο και στο πρακτικό κομμάτι της. Τέλος προτείνονται κάποιες πιθανές μελλοντικές επεκτάσεις για τη βελτίωση και την εξέλιξη του παρόντος συστήματος.

Συμπεράσματα

Ο κύριος στόχος της παρούσας Δ.Ε. ήταν να εξηγήσει πώς να χρησιμοποιηθούν οι τεχνικές του W.S., για τη συλλογή δεδομένων από τον ιστό και την εμφάνιση τους με ουσιαστικό τρόπο. Η εκπλήρωση αυτού του στόχου ολοκληρώθηκε χρησιμοποιώντας επιτυχώς τα δεδομένα από το booking.com. Ο χρόνος που χρειάστηκε για την ολοκλήρωση της εργασίας, είναι μικρός σε σχέση με τον χρόνο που θα ‘κερδηθεί’ με τη χρήση της. Με λίγα λόγια, η σωστή χρήση του W.S. είναι πολύ κερδοφόρα, καθώς θα γλιτώσει στο χρήστη αρκετό χρόνο.
Το Beautiful Soup είναι ένα εξαιρετικό εργαλείο για την εξαγωγή πολύ συγκεκριμένων πληροφοριών από μεγάλα αδόμητα ακατέργαστα δεδομένα, και επίσης είναι πολύ γρήγορο και εύχρηστο. Θα συνιστούσα εγγυημένα τη χρήση του Beautiful Soup για παρόμοια project όπως η παρούσα διπλωματική εργασία.
Η εφαρμογή της εργασίας είναι προγραμματισμένη έτσι ώστε όλα να γίνονται
αυτόματα , χωρίς την παρέμβαση του χρήστη. Η προβολή των δεδομένων από το χρήστη μπορεί να γίνει μέσω της βάσης δεδομένων ή μέσω csv αρχείων.
Τέλος από την ολοκλήρωση της εργασίας, βγαίνει το συμπέρασμα πως το W.S. αποτελεί μια τεχνική, όπου μπορεί να προσφέρει πολλές δυνατότητες στο χρήστη. Είναι εύκολη στη μάθηση, οπότε δε θα έπρεπε να λείπει από τις γνώσεις, από κανένα προγραμματιστή.

Προτάσεις βελτίωσης

Η εφαρμογή είναι ολοκληρωμένη όσον αφορά τις βασικές λειτουργίες. Σίγουρα όμως μπορούν να προστεθούν παραπάνω μηχανισμοί για την εμπλούτιση του project. Μια πρόταση είναι, ο χρήστης να μπορεί να δηλώσει το εύρος χιλιομέτρων στο χάρτη, για την ανάλογη εμφάνιση καταλυμάτων, ανάλογα με την περιοχή που τον ενδιαφέρει.
Το πρόγραμμα τρέχει κάθε μέρα αυτόματα. Μια ακόμα πρόταση βελτίωσης θα ήταν ο χρήστης να μπορεί να επιλέξει την ημερομηνία έναρξης και λήξης του προγράμματος, έτσι ώστε να λαμβάνει τα δεδομένα για τις ημέρες της θέλησης του.
Τέλος, τα δεδομένα στη βάση ενημερώνονται καθημερινά. Αυτό σημαίνει πως οι τιμές (π.χ. δωματίου) αντικαθίστανται κάθε φορά με καινούργιες. Η δυσκολία που παρουσιάστηκε σε αυτό το σημείο, είναι πως δε γινόταν οι τιμές να αποθηκεύονται σε πίνακα, και οι νέες τιμές να προστίθενται σε ένα νέο κελί. Έτσι χρειάστηκε να χρησιμοποιηθούν εξωτερικά αρχεία για την αποθήκευση των τιμών.
Δεν αποκλείεται η εφαρμογή της διπλωματικής εργασίας να χρησιμοποιηθεί αργότερα, για μελλοντικές επεκτάσεις. Απο τη στιγμή που όλα τα απαραίτητα δεδομένα βρίσκονται στη βάση, είναι πιθανό να ξεκινήσει και κάποιο καινούργιο project πάνω σε αυτά.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]Python Web Scraping Cookbook: Over 90 proven recipes to get you scraping with Python, microservices, Docker, and AWS by Michael Heydt.
[2]Hands-On Web Scraping with Python: Perform advanced scraping operations using various Python libraries and tools such as Selenium, Regex, and others by ANish Chapagain.
[3]Python Web Scraping Second Edition: Hands-on data scraping and crawling using PyQT, Selenium, HTML and Python by Katharine Jarmul and Richard Lawson.
[4]Practical Web Scraping for Data Science: Best PRactices and Examples with Python by Seppe vanden Broucke and Bart Baessens.
[5]Web Scraping with Python: Collecting More Data from the Modern Web by Ryan Mitchell.
[6]Web Scraping With Python by Chris Sheridan.
[7]A Python Guide for Web Scraping: Explore Python Tools, Web Scraping Techniques, and How to Automate Data for Industrial Applications by Pradumna Milind Panditrao.
[8]MongoDB Simply In Depth by Ajit Singh and Sultan Ahmad.
[9]MongoDB Basics: A quick introduction to MongoDB by David Hows, Peter Membrey, and Eelco Plugge.
[10]MongoDB Fundamentals: A hands-on guide to using MongoDB and Atlas in the real world by Amit Phaltankar, Juned Ahsan, Michael Harrison, and Liviu Nedov.
[11]Getting Started with Beautiful Soup: Build your own web scraper and learn all about web scraping with Beautiful Soup by Vineeth G. Nair.
[12]Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer by Mariot Tsitoara.
[13]Hands-on GitHub Actions: Implement CI/CD with GitHub Action Workflows for Your Applications by Chaminda Chandrasekara and Pushpa Herath.
[14]Automating Workflows with GitHub Actions: Automate software development workflows and seamlessly deploy your applications using GitHub Actions by Priscila Heller.
[15]Programming in Python 3: A complete Introduction to the Python Language by Mark Summerfield.
[16]Introduction to Python Programming for Business and Social Science Application by Frederick Kaefer and Paul Kaefer.
[17]Python Requests Essentials: Learn how to integrate your applications seamlessly with web services using Python Requests by Rakesh Vidya Chandra and Bala Subrahmanyam Varanasi.
[18]Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners by Al Sweigart.

[19]Beginning Python: From Novice to Professional, Presenting elegant Python techniques and data structures for all platforms, the Web, and the enterprise by Magnus Lie Hetland.