Τμήμα Ηλεκτρολόγων Μηχανικών
& Μηχανικών Υπολογιστών
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Τεχνολογίες και Υπηρεσίες Ευφυών Συστημάτων Πληροφορικής και Επικοινωνιών
Πρόγραμμα Μεταπτυχιακών Σπουδών

Διπλωματική Εργασία

Αποδοτική διερεύνηση και εξόρυξη γνώσης από το Web

Υπεύθυνος Καθηγητής: Ταμπακάς Βασίλειος

Φοιτητής: Αρβανιτάκης Παναγιώτης (2001)

09 - 09 - 2022

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, __-__-2022

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ
1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

Υπεύθυνη Δήλωση Φοιτητή Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη εργασία. Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος. Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Αρβανιτάκη Παναγιώτη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο Πανεπιστήμιο Πελοποννήσου, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.
Οι μορφή των εισαγωγικών σελίδων είναι με μικρούς χαρακτήρες ρωμαϊκής γραφής (i, ii, iii, iv, κλπ)

Περίληψη

Ο σκοπός της διπλωματικής εργασίας είναι να αναζητηθούν εργαλεία εξαγωγής δεδομένων και στην συνέχεια την ανάπτυξη εφαρμογής που θα παρουσιάζει τουριστικά δεδομένα με γραφικό τρόπο. Σε περίπτωση που δεν μπορούν να χρησιμοποιηθούν ήδη υπάρχον εργαλεία θα δημιουργηθεί από την αρχή ένα νέο.

Η μεθοδολογία που θα χρησιμοποιηθεί θα είναι τεχνικές Web scraping. Το πρόγραμμα θα γραφτεί σε γλώσσα nodejs και τα στοιχεία θα αποθηκεύονται στο elasticSearch. Όλα τα δεδομένα θα χρησιμοποιούνται από την εφαρμογή προκειμένου να εμφανίζονται στους χρήστες.

Η εφαρμογή θα είναι ένα πλήρες γραφικό περιβάλλον όπου ο χρηστής θα έχει την δυνατότητα να αναζητήσει ξενοδοχεία.

Abstract

The scope of this study is to search for scraping tools and then to develop an application that will present tourist data. In case existing tools cannot be used, a new one will be created from scratch.

In order to succeed this, web-scraping techniques will be implemented. The code will be written in nodejs language and the data will be stored in elasticSearch. All data will be used by the application to display to users.

The application will be a complete graphical environment where the user will be able to search for hotels.

Περιεχόμενα

Περιεχόμενα Εικόνων

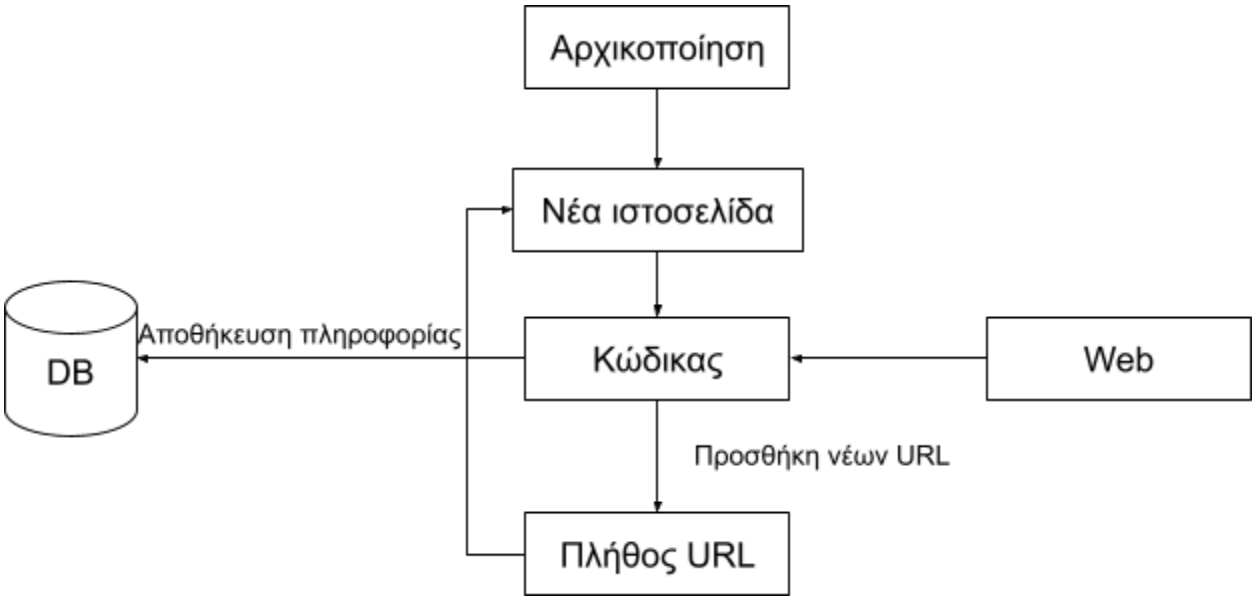
1. Εισαγωγή

1.1 Τι είναι ένας Crawler;

Ένας ανιχνευτής είναι ένα πρόγραμμα ή κάποιο script το οποίο έχει την δυνατότητα να επισκέπτεται οποιαδήποτε σελίδα στον παγκόσμιο ιστό με συστηματικό, αυτοματοποιημένο τρόπο. Το διαδίκτυο είναι σχεδιασμένο με τέτοιο τρόπο έτσι ώστε να θυμήζει κάποιον κατευθυνόμενο γράφο. Αυτό σημαίνει ότι κάθε σελίδα έχει κάποιες υποσελίδες που συνδέεται έχοντας σαν ακμές κάποιους συνδέσμους προς αυτές. Ένας ανιχνευτής χρησιμοποιεί αυτή την δομή προκειμένου να μετακινείται από σελίδα σε υποσέλιδα έχοντας σαν κύρια πληροφορία τον σύνδεσμό της. Για αυτόν τον λόγο εκτός από την ονομασία του ανιχνευτή μπορεί να ειπωθεί και σαν ρομπότ ή αράχνη. Η κύρια λειτουργία τους είναι να συλλέγουν όλον τον κώδικα μιας ιστοσελίδας και στην συνέχεια να τον αποθηκεύουν τοπικά σε κάποια βάση δεδομένων ή σε οποιοδήποτε άλλο αποθετήριο. Οι ανιχνευτές αρχικά αποθηκεύουν τοπικά την συνολική πληροφορία και στην συνέχεια χρησιμοποιούν κάποιο εργαλείο ανάλυσης και την απλοποιούν έτσι ώστε να είναι κατανοητή από κάποιον απλό χρήστη. [2]

1.2 Πως δουλεύει ένας Crawler;

Ο ανιχνευτής ξεκινάει γνωρίζοντας από την αρχή μια λίστα από διευθύνσεις τις οποίες πρέπει να επισκεφθεί και να αντλήσει την πληροφορία, συνήθως αυτές οι διευθύνσεις αποθηκεύονται σε ένα αρχείο με το όνομα seed. Για κάθε url καλείται να εξάγει τον κώδικα της ιστοσελίδας και στην συνέχεια τα αποθηκεύει σε κάποια δομή. Επίσης, συλλέγει ένα πλήθος από διάφορα url που αποτελούν τις υποσελίδες της σελίδας κόμβου. Για αυτό γίνεται ο απαραίτητος έλεγχος προκειμένου να διαπιστωθεί ότι δεν έχει γίνει εξαγωγή πληροφορίας για αυτά. Αν δεν έχει γίνει τότε αποθηκεύονται μαζί με τις άλλες διευθύνσεις που είναι προς επεξεργασία. Αυτή η διαδικασία κάνει κύκλους μέχρι να μην υπάρχει κάποιο url που να μην έχει γίνει εξαγωγή πληροφορίας. Οι νέες σελίδες που εντοπίζονται συνήθως είναι εκατομμύριες, αυτό σημαίνει ότι ο ανιχνευτής συνεχίζει να λειτουργεί για αρκετό καιρό μέχρι να τις επισκεφτεί όλες. Παρακάτω παρουσιάζεται ολόκληρη η διαδικασία σχηματικά και επιγραμματικά.



Εικόνα 1.1: Αρχιτεκτονική ενός crawler

Τα βήματα επιγραμματικά είναι:

- Επιλογή ενός ή πολλών URL
- Φέρνει το url στο σώμα του
- Επιλογή ενός ή πολλών URL από αυτά που έχει στο σώμα του
- Κατέβασμα του κώδικα της σελίδας
- Επεξεργασία του κώδικα για εξαγωγή νέων συνδέσμων
- Προσθήκη των νέων URL στα συνολικά
- Επαναφορά στο βήμα 2 [2]

1.3 Τεχνικές Crawling

Για την σωστή λειτουργία ενός ανιχνευτή εκτελούνται διάφορες τεχνικές με βάση την πληροφορία που καλείται να εξάγει. Η πληροφορία αυτή καθορίζεται από τον προγραμματιστή ή από τον χρήστη σε περίπτωση που του έχει δοθεί το δικαίωμα από τον διαχειριστή της εφαρμογής. Συγκεκριμένα οι τεχνικές που υπάρχουν είναι:

1. **Ανίχνευση γενικού σκοπού**
   Η συγκεκριμένη τεχνική έχει ως σκοπό την εξαγωγή πληροφορίας από ένα πλήθος διευθύνσεων καθώς και των υποδιευθύνσεων που θα βρεθούν σε αυτές. Σε αυτή την τεχνική ο ανιχνευτής καλείται να κατεβάσει μεγάλο όγκο πληροφορίας από πολλές και διαφορετικές ιστοσελίδες. Αυτό μπορεί να προκαλέσει καθυστέρηση σε όλη την διαδικασία καθώς και μείωση ταχύτητας στο δίκτυο αφού επισκέπτεται αρκετές σελίδες.
2. **Εστιασμένη Ανίχνευση**
   Ο σκοπός αυτής της τεχνικής είναι να μειώσει την κίνηση στο δίκτυο μειώνοντας τις λήψεις. Αυτό το πετυχαίνει κάνοντας ανίχνευση πάνω σε κάποιο συγκεκριμένο θέμα που έχει οριστεί από την αρχή. Ο σκοπός της είναι από ένα πλήθος ιστοσελίδων και με βάσει το θέμα να επιλεχθούν κάποια URL και από αυτά συγκεκριμένη πληροφορία. Ανιχνεύει επιλεκτικά και αυτό βοηθά στην εξοικονόμηση πόρων σε όλα τα επίπεδα.

3. **Κατανεμημένη Ανίχνευση**
   Σε αυτή την τεχνική σχεδιάζεται ένα ολόκληρο σύστημα από ανιχνευτές οι οποίοι θα εκτελούνται μαζικά σε ένα σύνολο σελίδων με απώτερο σκοπό την μαζική αποθήκευση δεδομένων. Επίσης, θα πρέπει να σχεδιαστεί και η επικοινωνία μεταξύ αυτών έτσι ώστε να αποφύγετε το πολλαπλό crawl και στην συνέχεια τις διπλοεγγραφές στην δομή αποθηκευσης, όποιες και αν είναι αυτές. [2]

1.4 Αρχιτεκτονική των web Crawler

Ένας ανιχνευτής αποτελεί σήμερα ένα πολύ σημαντικό κομμάτι των μηχανών αναζήτησης. Αυτό συμβαίνει καθώς ένα πρόγραμμα ανίχνευσης αναπτύσσεται ακριβώς με τον ίδιο τρόπο όπως αναπτύσσεται ο ιστός. Αρχίζει γνωρίζοντας μια λίστα από διευθύνσεις από τις οποίες η κάθε μία ονομάζεται “σπόρος”. Ο όρος “σπόρος” στις διευθύνσεις έχει δοθεί διότι η κάθε μια μπορεί να συνδέεται με πολλά διαφορετικά url και όσο η ανίχνευση εμβαθύνεται τόσο περισσότερα url εμφανίζονται δημιουργώντας ένα δέντρο. Ένας ανιχνευτής καλείται να επισκεφτεί όλες τις διευθύνσεις που έχουν οριστεί από την αρχή καθώς και τις επόμενες που πρόκειται να εντοπίσει. Οι υπερσυνδεσμοι που εντοπίζονται στην συνέχεια ονομάζονται σύνορα ανίχνευσης και αυτό διότι από αυτό το στάδιο και μετά ο ανιχνευτής δεν γνωρίζει τι υπάρχει παρά μόνο αν επισκεφθεί και τις διευθύνσεις αυτές. Κάθε διεύθυνση που επισκέπτεται την επεξεργάζεται κατάλληλα και αποθηκεύει την πληροφορία σε κάποια δομή καθώς επίσης το ίδιο το url το επαναπρογραμματίζει προσθέτοντας το στην ουρά εκ νέου. Τέλος, οι βάσεις δεδομένων θα πρέπει να είναι κατάλληλες προκειμένου να μπορούν να αποθηκεύσουν έναν τεράστιο όγκο από δεδομένα, για αυτό συνήθως χρησιμοποιούνται μηχανήματα όπως είναι το DB2 που έχει κατασκευαστεί για αποθήκευση μεγάλου όγκου δεδομένων. [2]

1.5 Τι είναι το Web Mining

Το Web Mining είναι η διαδικασία κατά την οποία χρησιμοποιούνται τεχνικές και αλγόριθμοι Data Mining με σκοπό την εξαγωγή πληροφορίας απευθείας από τον Ιστό. Αυτή η πληροφορία μπορεί να υπάρχει σε έγγραφα, υπηρεσίες ιστού, περιεχόμενο, υπερσυνδέσμους κλπ. Ο στόχος εξόρυξης δεδομένων από τον ιστό είναι να εντοπιστούν μοτίβα τα οποία καταλλήγουν σε καθαρή πληροφορία. Αυτή η πληροφορία χρησιμοποιείται για ανάλυση έτσι ώστε να υπάρχει εικόνα για τάσεις, βιομηχανία και γενικά να βγαίνει ένα τελικό συμπέρασμα για τον τελικό χρήστη. [2]

1.6 Συνεισφορά

Ο σκοπός της παρούσας διπλωματικής είναι να δημιουργηθεί εφαρμογή η οποία θα παρουσιάζει τουριστικά δεδομένα από διάφορες τουριστικές πλατφόρμες σε μια. Ο χρήστης θα πλέον να επισκέπτεται μια σελίδα στο διαδίκτυο και να βρίσκει πληροφορίες τουριστικών καταλυμάτων από διάφορα κανάλια όπως είναι οι βαθμολογίες.

2. Εντοπισμός εργαλείων

2.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο θα αναζητηθούν ήδη υπάρχον εργαλεία ανίχνευσης προκειμένου να αντιληφθούμε τις δυνατότητες που μπορεί να έχεις ένας crawler. Για κάθε ένα εργαλείο θα παρουσιάζονται τα θετικά και τα αρνητικά του στοιχεία καθώς και σε κάποιες περιπτώσεις κάποια από τα βασικά χαρακτηριστικά του.

2.2 PYSPIDER

Το PYSpider αποτελεί ένα αρκετά ισχυρό πρόγραμμα ανίχνευσης ιστού ανοικτού κώδικα το οποίο έχει αναπτύχθηκε σε python. Έχει μια κατανεμημένη αρχιτεκτονική με ενότητες όπως fetcher, χρονοπρογραμματισμού και επεξεργασίας.

Το Pyspider περιέχει πολλά χαρακτηριστικά παρόλα αυτά παρακάτω περιγράφονται τα βασικά:

- Διαθέτει μια ισχυρή διεπαφή χρήστη που βασίζεται στον ιστό με την οποία μπορούν να επεξεργαστούν σενάρια και μια πληθώρα εργαλείων που παρέχει λειτουργίες όπως παρακολούθηση εργασιών, διαχείριση έργου και προβολή αποτελεσμάτων.
- Συμβατό με JavaScript
- Μπορεί να αποθηκεύσει δεδομένα στις πιο γνωστές βάσεις δεδομένων όπως MYSQL, MongoDB, Redis, SQLite και Elasticsearch.
- Εκτός από τις βάσεις δεδομένων μπορεί να γίνουν χρήση άμεσα σε ουρές δεδομένων όπως είναι το RabbitMQ, Beanstalk και Redis.

Πλεονεκτήματα:

- Παρέχει ισχυρό έλεγχο προγραμματισμού.
- Υποστηρίζει ανίχνευση ιστοτόπων που βασίζεται σε JavaScript.
- Άμεση σύνδεση με RabbitMQ, Beanstalk και Redis
- Παρέχει υποστήριξη σε βάσεις δεδομένων όπως SQLite, MongoDB και MYSQL.

Μειονεκτήματα:

- Η ανάπτυξη και η εγκατάσταση του είναι λίγο δύσκολη και χρονοβόρα.

2.3 Portia

Το Portia είναι το καλύτερο εργαλείο ανίχνευσης με οπτικό περιβάλλον που αναπτύχθηκε από το scrapinghub και δεν απαιτεί γνώσεις προγραμματισμού. Έτσι, έχει αναπτυχθεί για μη προγραμματιστές. Μπορεί να δοκιμαστεί δωρεάν χωρίς να εγκαταστήσετε τίποτα δημιουργώντας έναν λογαριασμό στο scrapinghub. Κατά την εγγραφή, μπορεί κανείς να χρησιμοποιήσει την φιλοξενούμενη έκδοση που βασίζεται στο web.

Μερικά από τα βασικά χαρακτηριστικά του περιλαμβάνουν:

- Με το Portia, ακολουθούμε μια απλή διαδικασία όπως είναι κάποια κλίκ πάνω στην σελίδα προκειμένου να πάρει τα δεδομένα που ο προγραμματιστής χρειάζεται. Οι

ενέργειες του χρήστη όπως κύλιση, κλικ, αναμονή καθορίζονται από τον χρήστη κατά την εγγραφή του προκειμένου να μπορεί να συλλέξει περισσότερα δεδομένα.
- Δημιουργεί τη δομή των σελίδων μετά την επίσκεψη τους.
- Είναι αποτελεσματικό στην ανίχνευση ιστότοπων που λειτουργούν με AJAX.
- Είναι συμβατό με τεχνολογίες JavaScript όπως το Backbone, το Angular και το Ember.

Πλεονεκτήματα:
- Είναι συμβατό με CSS και XPath.
- Υποστηρίζει μορφές αποθήκευσης δεδομένων όπως CSV, JSON, XML.
- Φιλτράρει τη σελίδα που επισκέπτεται.

Μειονεκτήματα:
- Η διαδικασία ανίχνευσης είναι χρονοβόρα σε σύγκριση με άλλα εργαλεία ανοιχτού κώδικα.
- Μπορεί να περιηγηθεί σε περιττές σελίδες και να αποφέρει ανεπιθύμητα αποτελέσματα εάν δεν έχουν καθοριστεί οι ενδιαφερόμενες σελίδες.

2.4 Web-Harvest

Το Web-Harvest είναι ένα εργαλείο φτιαγμένο σε Java το οποίο μπορεί να εξάγει πληροφορία από το web. Έχει την δυνατότητα επιλογής χρήσιμης πληροφορίας χρησιμοποιώντας τεχνολογίες διαχείρισης XML αρχείου όπως είναι XSLT , XQuery και Regular Expressions. Βασίζεται κυρίως σε ιστοσελίδες που είναι γραμμένες σε Html/XML παρόλα αυτά μπορεί να υποστηρίξει κάθε είδους σελίδα προσθέτοντας κάποιες βιβλιοθήκες της Java.

Πλεονεκτήματα
- Υπάρχει δυνατότητα εντοπισμού χρήσιμης πληροφορίας
- Δυνατότητα εξαγωγής πληροφορίας από οποιαδήποτε ιστοσελίδα

Μειονεκτήματα
- Δεν υποστηρίζει κατανεμημένο Περιβάλλον
- Η ομάδα υποστήριξης είναι μικρή
- Χρησιμοποιείται κυρίως με γλώσσα προγραμματισμού Java

2.5 Puppeteer

Όπως ένας Webdriver έτσι και ο Puppeteer προσφέρει ένα API υψηλού επιπέδου για τον έλεγχο οτιδήποτε υπάρχει στον ιστό. Τα περισσότερα πράγματα που μπορούν να γίνουν χειροκίνητα στο πρόγραμμα περιήγησης από κάποιον χρήστη, μπορούν να γίνουν χρησιμοποιώντας το Puppeteer API. Το Puppeteer API είναι ιεραρχικό και αντικατοπτρίζει τη δομή του προγράμματος περιήγησης. Η επικοινωνία με το πρόγραμμα περιήγησης είναι δυνατή χρησιμοποιώντας το Πρωτόκολλο DevTools. Όλη η επικοινωνία γίνεται δυνατή με τη χρήση σειριακών αντικειμένων JSON μιας σταθερής δομής [12].

Πλεονεκτήματα:
- Η επικοινωνία γίνεται με χρήση JSON
- Περιέχει το puppeteer API
- Μπορεί να αναπαράξει ακριβώς ότι κάνει ο χρήστης

Μειονεκτήματα:

- Υποστηρίζει μόνο Javascript

2.6 Jaunt

Το Jaunt βασίζεται στην JAVA και έχει σχεδιαστεί για αναζήτηση ιστού, αυτοματοποίηση ιστού και ερωτήματα JSON. Προσφέρει ένα γρήγορο, εξαιρετικά ελαφρύ και χωρίς κεφαλές πρόγραμμα περιήγησης που παρέχει λειτουργικότητα web scraping, πρόσβαση στο DOM και έλεγχο κάθε Αίτησης/Απόκρισης HTTP, αλλά δεν υποστηρίζει JavaScript.

Πλεονεκτήματα:

- Επεξεργάζεται μεμονωμένες αιτήσεις/απαντήσεις HTTP
- Εύκολη διασύνδεση με οποιοδήποτε REST API
- Υποστήριξη για HTTP, HTTPS & βασικό έλεγχο ταυτότητας
- Ερώτημα με δυνατότητα RegEx σε DOM & JSON
- Εντάσσεται στα πολύ ελαφριά προγράμματα περιήγησης

Μειονεκτήματα:

- Δεν υποστηρίζει Javascript

2.7 Heritrix

Το Heritrix αποτελεί ένα εργαλείο της The Internet Archive του οποίου η αρχιτεκτονική βασίζεται στο έργο Mercator. Προσφέρει την δυνατότητα εκτέλεσης του σε κατανεμημένο περιβάλλον. Υποστηρίζει την επεκτασιμότητα όμως όχι την δυναμική επεκτασιμότητα. Αυτό σημαίνει ότι ο προγραμματιστης θα πρέπει να έχει καθορίσει από την αρχή της ανίχνευση το πλήθος των μηχανών.

Το αποτέλεσμα του Heritrix είναι αρχεία τύπου WARC καθώς δεν υποστηρίζει αποθήκευση δεδομένων σε άλλες μορφές αποθήκευσης. Η συνεχείς ανίχνευση δεδομένων δεν υποστηρίζεται, παρόλο αυτά υπάρχει δυνατότητα επεξεργασίας αυτών. Αποτελεί ένα ώριμο εργαλείο καθώς έχει κυκλοφορήσει από το 2004

Πλεονεκτήματα

- Ώριμο εργαλείο
- Υποστήριξη κατανεμημένου περιβάλλοντος

Μειονεκτήματα

- Δεν προσφέρει αποθήκευση δεδομένων μέσω των πιο γνωστών μορφών
- Δεν υποστηρίζει δυναμική επεκτασιμότητα

3. Εγκατάσταση και επιλογή εργαλείων

3.1 Εισαγωγή

Στο παρόν κεφάλαιο θα αναληθούν κάποια από τα βασικά εργαλεία που χρειάζονται προκειμένου να γίνει η εγκατάσταση του Sparkler και του Nutch (Παράρτημα 1 & 2). Επίσης θα αναζητηθούν πληροφορίες σχετικά με αυτά τα δύο εργαλεία όπως είναι η αρχιτεκτονική τους και στην συνέχεια θα εγκατασταθούν προκειμένου να ελεγχθούν ως προς την χρήση τους.

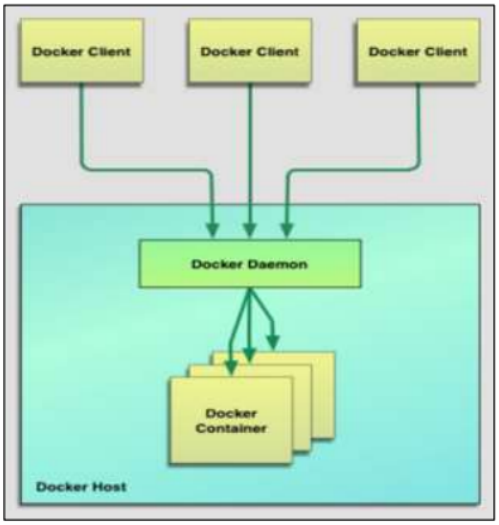
3.2 Εργαλεία

3.2.1 Docker

Ο Docker είναι μια πλατφορμα λογισμικού ανοιχτού κώδικα η οποία οπτικοποιεί σε επίπεδο λογισμικού. Αυτό που πραγματικά προσφέρει είναι την δυνατότητα εγκατάστασης συγκεκριμένης εφαρμογής / βιβλιοθήκης / υπηρεσίας χωρίς την χρήση λειτουργικού συστήματος. Σε ένα περιβάλλον που το ονομάζει container δίνει την δυνατότητα προσθήκης πολλών εικονικοποιημένων εφαρμογών οι οποίες μπορούν να επικοινωνήσουν μεταξύ τους. Ο Russell το 2015 επιβεβαίωσε ότι το docker δίνει την δυνατότητα δοκιμής του κώδικα ο οποίος μπορεί κατευθείαν να μεταβεί σε περιβάλλον παραγωγής. [5]

3.2.1.1 Docker Πελάτη - Διακομιστή

Το Docker μπορεί να θεωρηθεί ότι χρησιμοποιεί ένα μοντέλο πελάτη - διακομιστή. Ο διακομιστής λαμβάνει το αίτημα από τον client και το επεξεργάζεται ανάλογα. Το ολοκληρωμένο RESTful API καθώς και κάποιες εντολές αποστέλλονται από το Docker. Ο client και ο server μπορούν να συνδεθούν στον ιδιο τοπικό υπολογιστή ή κάποιος τοπικός docker client μπορεί να συνδεθεί σε κάποιον απομακρυσμένο server - ο οποίος εκτελείται σε άλλο μηχάνημα. [5]



Εικόνα 3.1: Λειτουργία Docker

3.2.1.2 Εικόνες Docker

Το εργαλείο Docker προσφέρει δύο μεθόδους δημιουργίας μιας εικόνας. Αρχικά, ο πρώτος αφορά την δημιουργία εικόνας χρησιμοποιώντας την ιδιότητα read-only. Την αρχή κάθε εικόνας αποτελεί μια εικόνα βασική όπως είναι αυτή με το λογισμικό του Ubuntu. Η κάθε εικόνα δημιουργεί ένα container το οποίο έχει όλες τις ιδιότητες ενός πραγματικού λειτουργικού συστήματος. Η βασική εικόνα μπορεί να δημιουργηθεί από την αρχή. Σε κάθε αρχική, η νέα εικόνα μπορεί να γίνουν τροποποιήσεις προκειμένου να λειτουργεί σύμφωνα με τα χαρακτηριστικά που έχουν δοθεί στον προγραμματιστή. Η δεύτερη μέθοδος είναι να δημιουργηθεί ένα αρχείο docker. Κάθε αρχείο docker περιλαμβάνει μια σειρά από εντολές που θα πρέπει να εκτελεστούν σειριακά προκειμένου να διαμορφωθεί το περιβάλλον και να ξεκινήσει κάποιο έργο. Κατα την εκτέλεση αυτού του αρχείου δημιουργείται μια εικόνα docker. [5]

3.2.1.3 Αποθετήρια Docker

Οι εικόνες docker αποθηκεύονται σε αποθετήρια docker. Αυτά έχουν ακριβώς τις ίδιες ιδιότητες με αυτά που αποθηκεύουμε τον κώδικα. Συγκεκριμένα μπορούμε να ανεβάσουμε κάποιες αλλαγές (push), να ενημερώσουμε την εικόνα με κάποιες αλλαγές που έχει κάνει κάποιος άλλος (pull) κλπ. Επίσης για τα αποθετήρια διαθέτει το Docker Hub το οποίο επιτρέπει σε κάποιον απομακρυσμένα ή οχι να τραβήξει την εικόνα ή τις αλλαγές. [5]

3.2.1.4 Docker Container

Η κάθε εικόνα docker δημιουργεί ένα docker container. Το κάθε container περιέχει οτιδήποτε χρειάζεται μια εφαρμογή προκειμένου να αρχίσει να λειτουργεί. Για παράδειγμα αν έχουμε μια εικόνα με Ubuntu που περιέχει SQL Server, όταν εκτελεστεί η εντολή docker run τοτε θα δημιουργηθεί ένα container το οποίο εκκινεί έναν SQL Server στα Ubuntu.

3.2.2 ElasticSearch

Το Elasticsearch είναι μια κατανεμημένη βάση δεδομένων που αποθηκεύει έγγραφα JSON που έχουν σχεδιαστεί ειδικά για αναζήτηση και ανάλυση ημιδομημένων δεδομένων. Σε αντίθεση με ένα σύστημα διαχείρισης σχεσιακής βάσης δεδομένων, το Elasticsearch αν και δεν είναι χωρίς σχήματα (όπως το MongoDB για παράδειγμα) που σημαίνει ότι δεν απαιτείται να οριστούν οι τύποι (συμβολοσειρά, αριθμός, κ.λπ.) των δεδομένων πριν από την εισαγωγή τους, επιτρέπει τον ορισμό των τύπων. Η υποκείμενη τεχνολογία είναι η μηχανή αναζήτησης κειμένου Apache Lucene. [6]

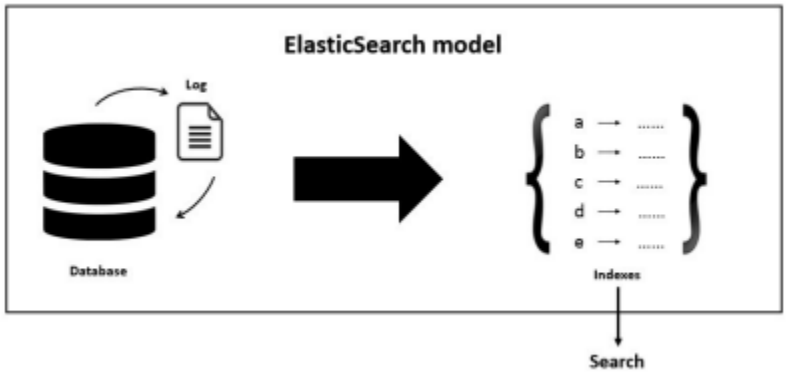
Συνεπώς το Elasticsearch αποτελεί μια μηχανή αναζήτησης πραγματικού χρόνου και ανάλυσης δεδομένων που βασίζεται στο Apache Lucene. Τα οφέλη της elasticSearch είναι:

- **Υψηλή απόδοση:** Η κατανεμημένη φύση του Elasticsearch μας επιτρέπει την επεξεργασία μεγάλου όγκου δεδομένων παράλληλα, κάτι που κάνει την απάντηση του ερωτήματος πολύ γρήγορη.
- **Δωρεάν εργαλεία και πρόσθετα:** δυνατότητα εύκολης διασύνδεσης με διάφορα εργαλεία
- **Εύκολη ανάπτυξη εφαρμογών**: η elasticsearch υποστηρίζεται από τις πιο γνωστές γλώσσες όπως JAVA, Python, Nodejs κλπ

- **Λειτουργίες σε πραγματικό χρόνο:** καθώς χρειάζεται λιγότερο από 1 δευτερόλεπτο για να διαβάσουμε ή να γράψουμε στην Elasticsearch

3.2.2.1 Το μοντέλο του ElasticSearch

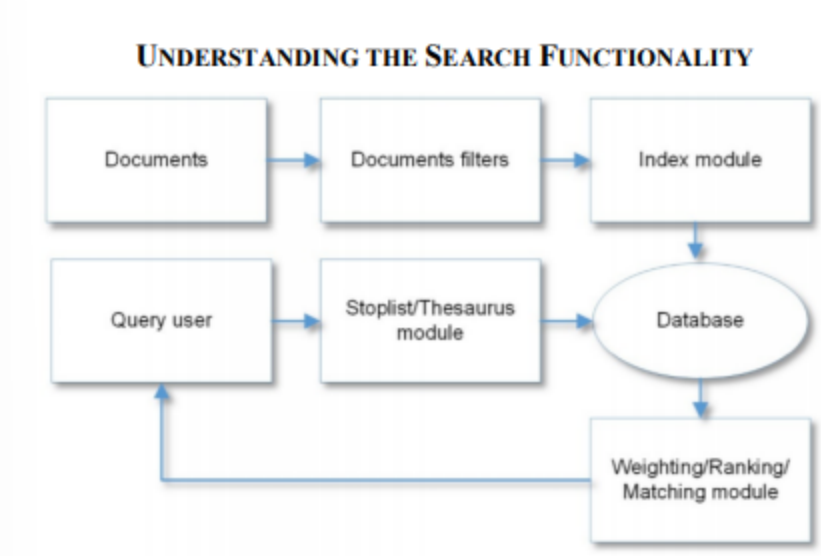
Τα ακατέργαστα δεδομένα ρέουν στην elasticsearch από διάφορες πηγές, συμπεριλαμβανομένων αρχείων καταγραφής, μετρήσεων συστήματος και εφαρμογών ιστού. Η απορρόφηση δεδομένων είναι η διαδικασία με την οποία τα δεδομένα που δεν έχουν επεξεργαστεί αναλύονται, κανονικοποιούνται και εμπλουτίζονται πριν από την ευρετηρίαση στο elasticsearch. Μετά την ευρετηρίαση, οι χρήστες μπορούν να εκτελέσουν σύνθετα ερωτήματα με τα δεδομένα τους και να χρησιμοποιήσουν όλες τις κλήσεις που προσφέρει προκειμένου να εκτελέσει απλά ή σύνθετα ερωτήματα ώστε να επιστραφούν τα δεδομένα. Από το Kibana, οι χρήστες μπορούν να δημιουργήσουν απεικονίσεις των δεδομένων τους, να μοιραστούν και να δημιουργήσουν πίνακες εργαλείων και να διαχειριστούν τα δεδομένα του Elasticsearch. [7]



Εικόνα 3.2: Μοντέλο Elastic search

3.2.2.2 Search Functionality

Το Elasticsearch χρησιμοποιεί την έννοια της ανεστραμμένης ευρετηρίασης για την αναζήτηση των ερωτημάτων. Δηλαδή οι όροι απάντησης ταξινομούνται με αύξουσα σειρά παρόλα αυτά μπορούν να ταξινομηθούν ανάλογα με το τι χρειάζεται ο προγραμματιστής. Ο κύριος λόγος που το elasticsearch επιστρέφει αποτελέσματα σε πραγματικό χρόνο είναι επειδή αναζητά τα ευρετήρια αντί να αναζητά απευθείας το κείμενο. Το Elasticsearch παρέχει τη δυνατότητα περαιτέρω υποδιαίρεσης του ευρετηρίου σε πολλά κομμάτια που ονομάζονται θραύσματα. Όταν ένα νέο έγγραφο αποθηκεύεται και καταχωρείται στο ευρετήριο, το elasticsearch ορίζει ένα θραύσμα υπεύθυνο για αυτό το έγγραφο. Όταν δημιουργείται ένα ευρετήριο, ο χρήστης μπορεί να καθορίσει τον αριθμό των θραυσμάτων που θα δημιουργηθούν. Κάθε θραύσμα είναι από μόνο του ένα πλήρως λειτουργικό ευρετήριο. Οποιοσδήποτε αριθμός εγγράφων μπορεί να μεταφορτωθεί στο elasticsearch ανεξάρτητα από τον τύπο του. [7]



Εικόνα 3.3: Elastic search Functionality

3.2.3 Maven

Το apache Maven είναι ένα πολύ δημοφιλή εργαλείο ανοιχτού κώδικα το οποίο μπορεί να κατασκευάσει πολύ γρήγορα οποιαδήποτε εφαρμογή. Με αυτόν τον τρόπο αφαιρεί μεγάλο μέρος σκληρής δουλειάς από εταιρείες που δημιουργούν έργα βασισμένα στην Java. Το maven με την βοήθεια ενός αρχείου παραμέτρων POM, χρησιμοποιώντας μια δηλωτική προσέγγιση περιγράφει το project και ότι αυτό χρειάζεται για να λειτουργήσει. Συνεπώς, προσφέρει μια εύκολη διαχείριση των πακέτων - βιβλιοθηκών που θα χρειαστεί το project. Επίσης, περιγράφεται η δομή και τα περιεχόμενα του έργου [8].

3.2.3.1 Μοντέλο

Η καρδιά ενός έργου Maven 2 είναι το μοντέλο αντικειμένου έργου (ή POM για συντομία). Περιέχει μια λεπτομερή περιγραφή του έργου σας, συμπεριλαμβανομένων πληροφοριών σχετικά με τη διαχείριση εκδόσεων και διαμόρφωσης, τις εξαρτήσεις, τους πόρους εφαρμογών και δοκιμών, τα μέλη και τη δομή της ομάδας και πολλά άλλα που χρειάζεται ένα έργο για να λειτουργήσει. Το POM έχει τη μορφή ενός αρχείου XML και σε οποιοδήποτε εφαρμογή εμφανίζεται με την μορφή pom.xml, το οποίο τοποθετείται στον αρχικό κατάλογο του έργου σας. Ένα παράδειγμα αρχείου pom.xml παρουσιάζεται παρακάτω [8]:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.javaworld.hotels</groupId>
  <artifactId>HotelDatabase</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>Maven Quick Start Archetype</name>
  <url>http://maven.apache.org</url>
  <dependencies>
     <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>3.8.1</version>
        <scope>test</scope>
     </dependency>
  </dependencies>
</project>
```

Εικόνα 3.4: Παράδειγμα pom.xml αρχείου

3.2.3.2 Ενέργειες

Σε κάθε έργο που δημιουργείται υπάρχουν περιπτώσεις που κάποια κομμάτια ή αρχεία ή εξαρτήσεις κλπ χρειάζονται μόνο για παράδειγμα κατά την εκτέλεση δοκιμής (test). Συνεπώς, το maven έχει προβλέψει αυτές τις περιπτώσεις και έχει δημιουργήσει παραμέτρους που δηλώνουν την αντίστοιχη ενέργεια. Έτσι για κάθε αρχείο και μόνο όταν δεν χρειάζεται το αφαιρεί από το classpath της εφαρμογής και ετσι δεν συμπεριλαμβάνεται στην κεντρική εφαρμογή.

Συγκεκριμένα το maven παρέχει 4 ρυθμίσεις προκειμένου να εφαρμοστούν όλα τα παραπάνω:

- compile: είναι η προεπιλεγμένη τιμή και χρησιμοποιείται για την μεταγλώττιση της εφαρμογής.
- provided: Χρησιμοποιείται για την μεταγλώττιση αλλά δεν αναπτύσει την εφαρμογή.
- runtime: Δεν χρειάζεται για την μεταγλώττιση παρά μόνο για την εκτέλεση.
- test: ο σκοπός αυτής της επιλογής είναι να τρέξει όλες τις δοκιμές που υπάρχουν στο υπάρχον έργο.[8]

3.2.4 Scala

Η scala είναι μια γλώσσα προγραμματισμού γενικής χρήσης καθώς υποστηρίζει αντικειμενοστραφή και συναρτησιακό προγραμματισμό. Σχεδιάστηκε προκειμένου να λύσει κάποια προβλήματα της java όπως είναι ο εξαναγκασμός υλοποίησης με αντικειμενοστραφή προγραμματισμό.

3.2.4.1 Επισκόποιση

Η επισκόπηση της γλώσσας Scala επεκτείνεται στα ακόλουθα βασικά χαρακτηριστικά [11]:

- Τα προγράμματα γραμμένα με scala μοιάζουν αρκετά σε αυτά της Java σε πολλά σημεία και μπορούν να επικοινωνήσουν εύκολα με κώδικα γραμμένο σε Java.
- Η Scala έχει ένα ομοιόμορφο μοντέλο αντικειμένου, με την έννοια ότι κάθε τιμή είναι ένα αντικείμενο και κάθε πράξη είναι μια κλήση μεθόδου
- Η Scala θεωρείται μια λειτουργική γλώσσα διότι προσφέρει συναρτήσεις στις οποίες οι τιμές εντάσσονται στην πρώτη κατηγορία.
- Έχει ομοιόμορφες και ισχυρές αφαιρετικές έννοιες τόσο για τύπους όσο και για τιμές.
- Επιτρέπει την αποσύνθεση αντικειμένων με αντιστοίχιση σχεδίων
- Τα μοτίβα και οι εκφράσεις γενικευονται για να υποστηρίξουν τη φυσική επεξεργασία των εγγράφων XML
- Συνολικά, αυτές οι δομές διευκολύνουν την έκφραση αυτόνομων στοιχείων χρησιμοποιώντας βιβλιοθήκες Scala
- Η Scala εφαρμόζεται προς το παρόν στην Java και σε πλατφόρμες .NET

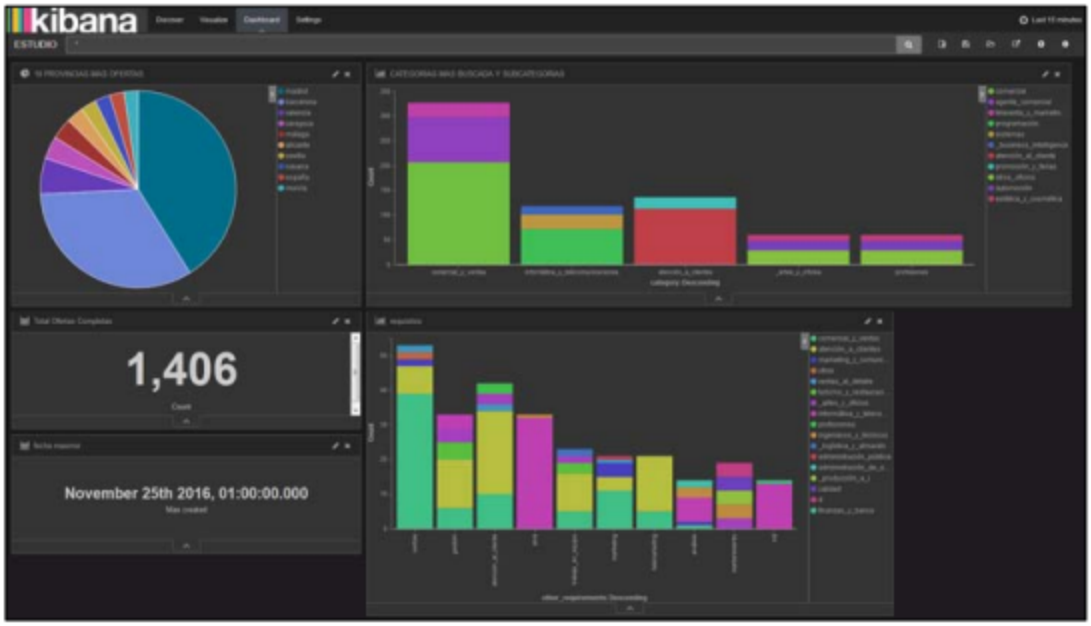
```
// Java
class PrintOptions {
  public static void main(String[] args) {
    System.out.println("Options␣selected:");
    for (int i = 0; i < args.length; i++)
      if (args[i].startsWith("-"))
        System.out.println("␣"+args[i].substring(1));
  }
}

// Scala
object PrintOptions {
  def main(args: Array[String]): unit = {
    System.out.println("Options␣selected:");
    for (val arg <- args)
      if (arg.startsWith("-"))
        System.out.println("␣"+arg.substring(1));
  }
}
```

Εικόνα 3.5: Σύγκριση Java & Scala

3.2.5 Kibana

Το Kibana είναι ένα δωρεάν και ανοιχτού κώδικα περιβάλλον το οποίο οπτικοποιεί τα δεδομένα που είναι αποθηκευμένα στο elasticSearch καθώς και βοηθάει στην πλοήγηση του Elastic αφου παρέχει μια πλήρη λίστα από συνηθισμένα query. Πολύ σημαντική λειτουργία του θεωρείται η παρακολούθηση του φορτίου και η κατανόηση της κίνησης των αιτημάτων καθώς προσφέρει διαγράμματα και πίνακες που αναλύουν την πληροφορία. Τέλος, μπορούν να ενεργοποιηθούν ειδοποιήσεις που αφορούν καθημερινή/εβδομαδιαία/μηνιαία ενημέρωση σχετικά με κάποια κομμάτια που ο ιδιος ο προγραμματιστής επιλέγει.

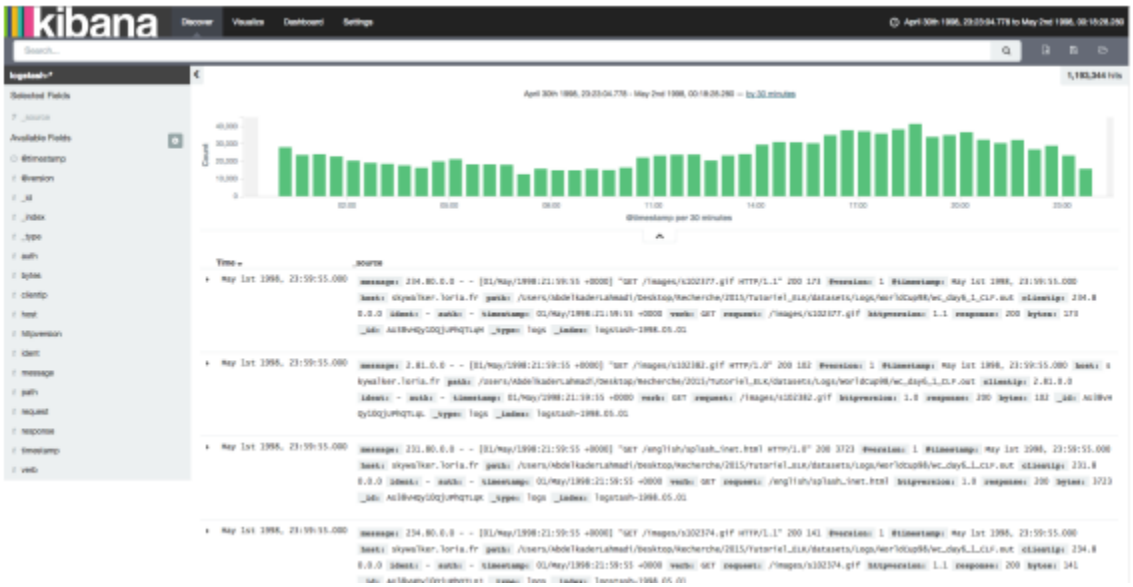


Εικόνα 3.6: Το περιβάλλον του Kibana

3.2.5.1 Οπτικοποίηση

Τα δεδομένα στο περιβάλλον του kibana μπορούν να οπτικοποιηθούν με διάφορους τρόπους όπως περιγράφονται παρακάτω [10]:

- Περιβάλλον που μπορεί να ανοίξει από κάποιον Browser
- αναζήτηση, προβολή και αλληλεπίδραση με δεδομένα που είναι αποθηκευμένα στον δείκτη Elasticsearch
- διαγράμματα, πίνακες και χάρτες
- πίνακες εργαλείων για την εμφάνιση αλλαγών στα ερωτήματα του Elasticsearch σε πραγματικό χρόνο



Εικόνα 3.7: Εμφάνιση των δεδομένων στο Kibana

3.2.6 Ant

Το ant χρησιμοποιείται κυρίως για το build εφαρμογών που έχουν υλοποιηθεί με Java. Παρόλο αυτά μπορεί να χρησιμοποιηθεί για έργα που έχουν χτιστεί σε C και C++. Η κύρια λειτουργία του είναι να εγκαθιστά όλες τις βιβλιοθήκες που απαιτούνται σύμφωνα με το build αρχείο και στο τέλος να τις προσθέτει όλες μαζί με τα άλλα αρχεία στον build φακελο. Η εμφάνιση του Ant φάνηκε τολμηρή. Πολλοί θεωρούν πως το Ant αποτελεί μια εναλλακτική λύση του make για εφαρμογές Java. Στην ουσία όμως θεωρείται ένα πολύ ισχυρό εργαλείο το οποίο στην πραγματικότητα το μοναδικό κοινό που έχουν είναι ο σκοπός τους - να χτίζουν έργα. [9]

3.2.6.1 Χρήση

Το ant κυρίως χρησιμοποιείται για την μεταγλώττιση αρχείων παρόλα αυτά μπορεί εξίσου να χρησιμοποιηθεί για την δημιουργία και ανάπτυξη εφαρμογών. Στην πραγματικότητα όλα αυτά που χρησιμοποιούνται συνήθως στα εργα (tomcat, log4j, compiere, mondrian κλπ) χτίστηκαν με το Ant. Για να μπορεί να χρησιμοποιηθεί θα πρέπει τα έργα να περιλαμβάνουν ένα αρχείο build.xml στον αρχικό τους φάκελο το οποίο είναι και αυτό που καθορίζει για το πως θα χτιστεί το έργο. [9]

3.2.6.2 Απλότητα

Το Ant μπορεί να χρησιμοποιηθεί πολύ εύκολα σε κάποιο έργο. Το μόνο που χρειάζεται είναι ένα αρχείο XML με μια απλή δομή. Ο προγραμματιστής ορίζει τις εργασίες που πρέπει να ολοκληρωθούν (Javac, αντιγραφή, zip κλπ) και τις ομαδοποιεί σε στόχους. Η συγκεκριμένη δομή μπορεί να επεξεργαστεί με οποιοδήποτε πρόγραμμα επεξεργασίας κειμένου και αποτελεί την μόνη δομή που χρησιμοποιεί το Ant. Παρακάτω παρουσιάζεται ένα απλό παράδειγμα της δομής αυτού του αρχείου: [9]

```
<project name="example"
        default="run">
  <target name="compile">
    <javac srcdir="."
      destdir="."/>
  </target>
  <target name="run"
        depends="compile">
     <java classname=
      "Hello"/>
  </target>
</project>
```

Εικόνα 3.8: Η δομή του αρχείου xml της ant

3.3 Sparkler

3.3.1 Εισαγωγή

Το sparkler είναι ένα εργαλείο το οποίο έχει την δυνατότητα να συλλέξει διαφόρων ειδών δεδομένα βασιζόμενο στην Java επεκτείνοντας τη λειτουργικότητα του Apache Spark και χρησιμοποιώντας διάφορα άλλα εργαλεία της Apache όπως είναι το kafka, Lucene/Sorl, Tika και pf4j. Αποτελεί εξέλιξη του Apache Nutch καθώς τρέχει στο Apache Spark Cluster. Ο αρχικός του σχεδιασμός αφορούσε την χρήση στο DARPA και MEMEX παρόλα αυτά έχει χρησιμοποιηθεί και σε άλλα έργα. Το μεγαλύτερο χαρακτηριστικό του αποτελεί η υψηλή αποδοτικότητα που προσφέρει αφου επιτρέπει ελεγχόμενες ανιχνεύσεις μεγάλης κλίμακας. Τα δεδομένα που εντοπίζει τα παρουσιάζει με γραφικό τρόπο σε πραγματικό χρόνο. Η τελική σχεδίαση του επιτρέπει στους χειριστές να προσθέτουν ή να αφαιρούν μια μεγάλη ποσότητα προσθέτων. Το περιβάλλον ανίχνευσης του Sparkler από την άλλη μεριά είναι ένα πλήθος από εργαλεία που έχουν δημιουργηθεί πάνω από αυτό. Το συγκεκριμένο περιβάλλον προσφέρει εντολές που βασίζονται στον Docker για την εύκολη δημιουργία και εκτέλεση διεργασιών Crawl. [3]

3.3.2 Χρήση

Το Sparkler χρησιμεύει κυρίως για συλλογή αρχείων όπως είναι ο τύπος PDF. Έχοντας σαν κριτήριο να μην γνωρίζει την τοποθεσία που πρόκειται να επισκεφθεί, το καθιστά πολύ λιγότερο αποτελεσματικό σε σχέση από ανιχνευτές που έχουν δημιουργηθεί για συγκεκριμένες ιστοσελίδες και έτσι δεν μπορεί να συλέξει συγκεκριμένες πληροφορίες. Σε μια έρευνα που έχει πραγματοποιηθεί το sparkler δεν μπόρεσε να επιστρέψει όλα τα PDF που υπήρχαν στην Υπηρεσία Εσωτερικών Εσόδων (IRS) παρόλα αυτά μετά απο ανίχνευση σε 71508 σελίδες βρήκε 311 PDF [3].

3.3.3 Τεχνολογίες

Το Sparkler προκειμένου να ανταγωνιστεί το ήδη υπάρχον εργαλείο ανίχνευσης Nutch έχει προσθέσει μια πληθώρα από τεχνολογίες οι οποίες τελικά συνδυάζονται μεταξύ τους και βγάζουν το τελικό αποτέλεσμα. Συγκεκριμένα [4]:

- Ολοκληρωμένο πακέτο ανίχνευσης - παρόμοιο σαν αυτό του Nutch
- Τα apache Sorl ή elasticSearch χρησιμοποιείται σαν βάση προκειμένου να αποθηκεύονται τα δεδομένα
- Πολλαπλά τμήματα από έργα της Maven με πακέτα OSGi
- Συνεχόμενη Ροή δεδομένων χρησιμοποιώντας το Apache Kafka
- Ανάλυση αρχείων με την χρήση του Apache Tika
- Οπτικοποίηση ανιχνευσιμων δεδομένων με το πρόγραμμα Banana

3.3.4 Αρχιτεκτονική

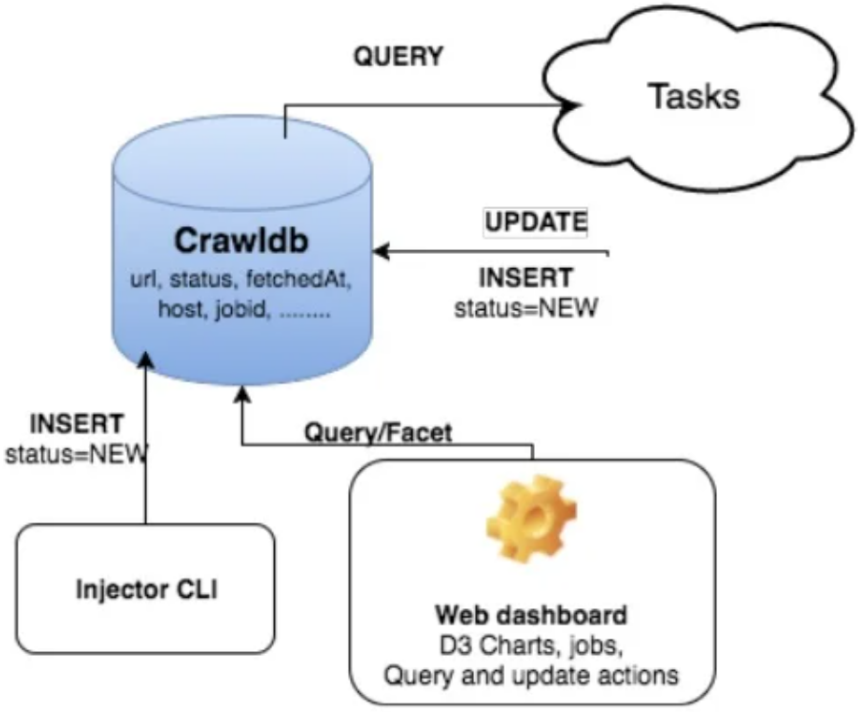
Το Apache Spark έχει μια καλά σχεδιασμένη πολυεπίπεδη αρχιτεκτονική όπου όλα τα κομμάτια και τα επίπεδα του συνδέονται με ομαλό τρόπο. Αυτή η αρχιτεκτονική επεκτείνεται

περαιτέρω με διάφορες βιβλιοθήκες ή πρόσθετα με τις οποίες το sparkler μπορεί να επικοινωνήσει. Η αρχιτεκτονική του Apache Spark βασίζεται σε:

- κατανεμημένο σύνολο δεδομένων (RDD)
- Directed Acyclic Graph (DAG)

3.3.4.1 Crawl Βάση

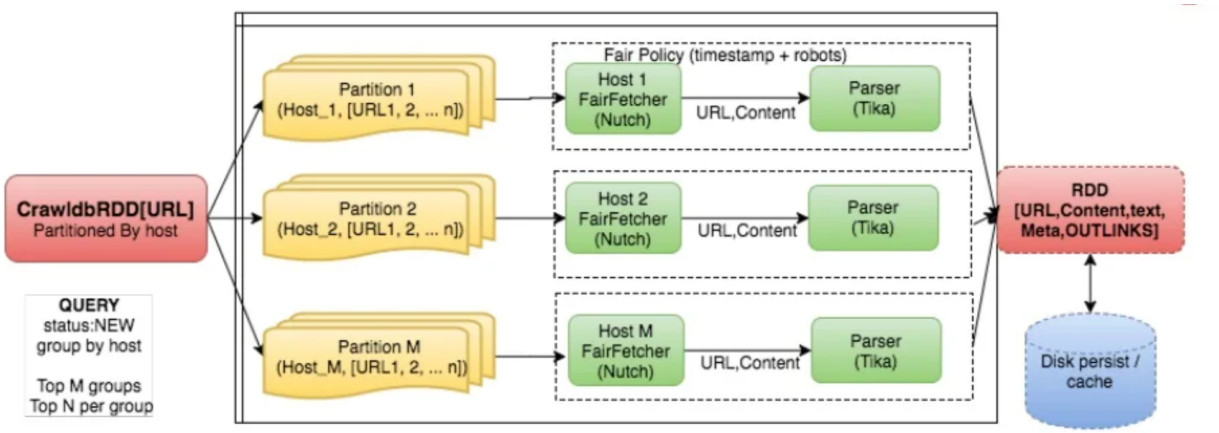
Η αρχιτεκτονική του sparkler περιέχει σαν αρχή μια βάση noSQL η οποία δημιουργείται είτε με Sorl είτε με Elasticsearch. Στην συγκεκριμένη βάση μπορεί να αποθηκεύσει κάποιο υπεύθυνος (admin) είτε το ιδιο το πρόγραμμα. Για την εκκίνηση χρειάζεται ο διαχειριστής να εισάγει τα πρώτα ερωτήματα. Το ιδιο το πρόγραμμα μπορεί να επεξεργαστεί ή να εισάγει και αυτό κάποιο ερώτημα. Εφόσον υπάρχουν ερωτήματα ένα πλήθος από αυτά πηγαίνουν στην διαδικασία ανίχνευσης.



Εικόνα 3.9: Αρχιτεκτονική crawl βάσης

3.3.4.2 RDD

Σε αυτό το σημείο εκτελεί ένα ερώτημα που επιστρέφει κάποιο συγκεκριμένο πλήθος ή όλα τα url που είναι προς ανίχνευση. Εδώ το sparkler σπάει τις διαδικασίες σε Ν πλήθος προκειμένου να εκτελεστούν παράλληλα. Σε κάθε μια διαδικασία γίνεται ανίχνευση πληροφορίων και εκτελούνται κάποια εργαλεία που μπορεί να έχουν προστεθεί όπως είναι το Apache Tika. Στο τέλος όλα τα δεδομένα καταλλήγουν σε ένα RDD.



Εικόνα 3.10: Αρχιτεκτονική RDD

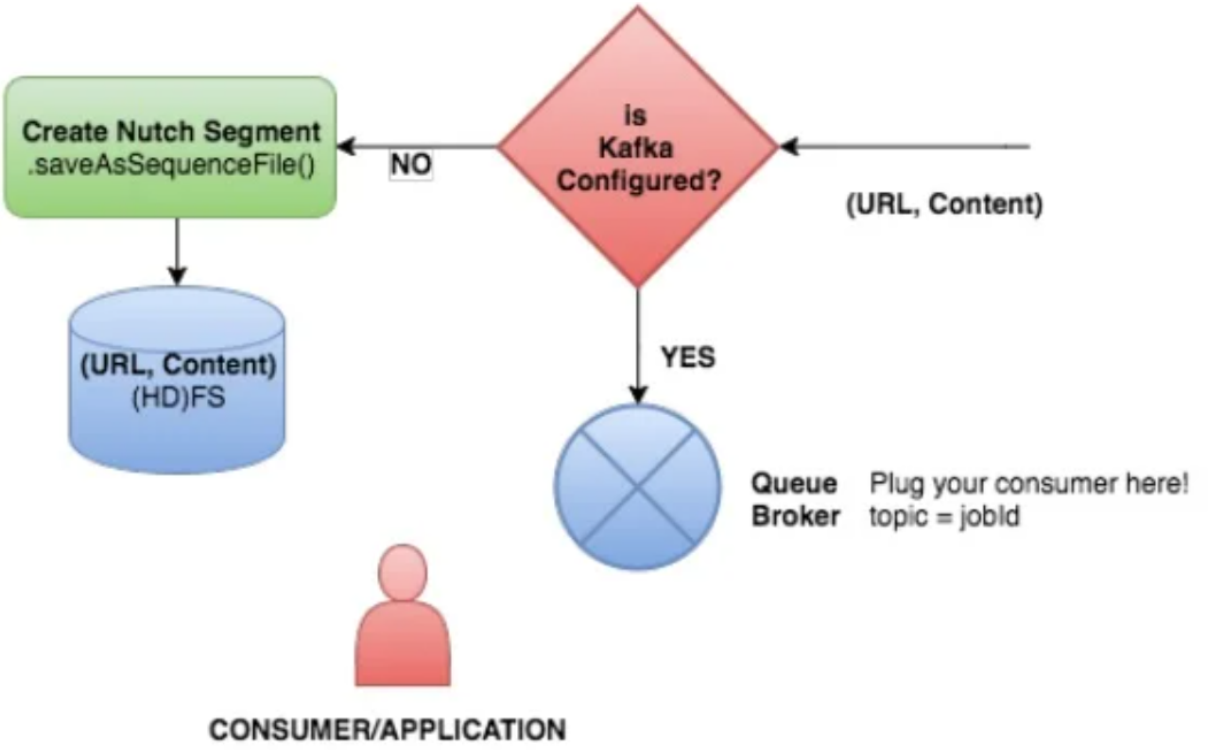
3.3.4.3 Συνδέσεις Pipeline

Το παρακάτω σημείο θεωρείται ένα από τα πιο βασικά και τα πιο δύσκολα σημεία προς υλοποίηση. Συγκεκριμένα σαν πρώτο βήμα καθαρίζονται τα url, στην συνέχεια διαγράφονται όλες οι διπλές διευθύνσεις και τέλος ελέγχεται αν υπάρχουν παλιά url προκειμένου να διαγραφούν. Τελικά όλα αυτά αποθηκεύονται ή ενημερώνονται στην αρχική βάση.

Εικόνα 3.11: Συνδέσεις Pipeline

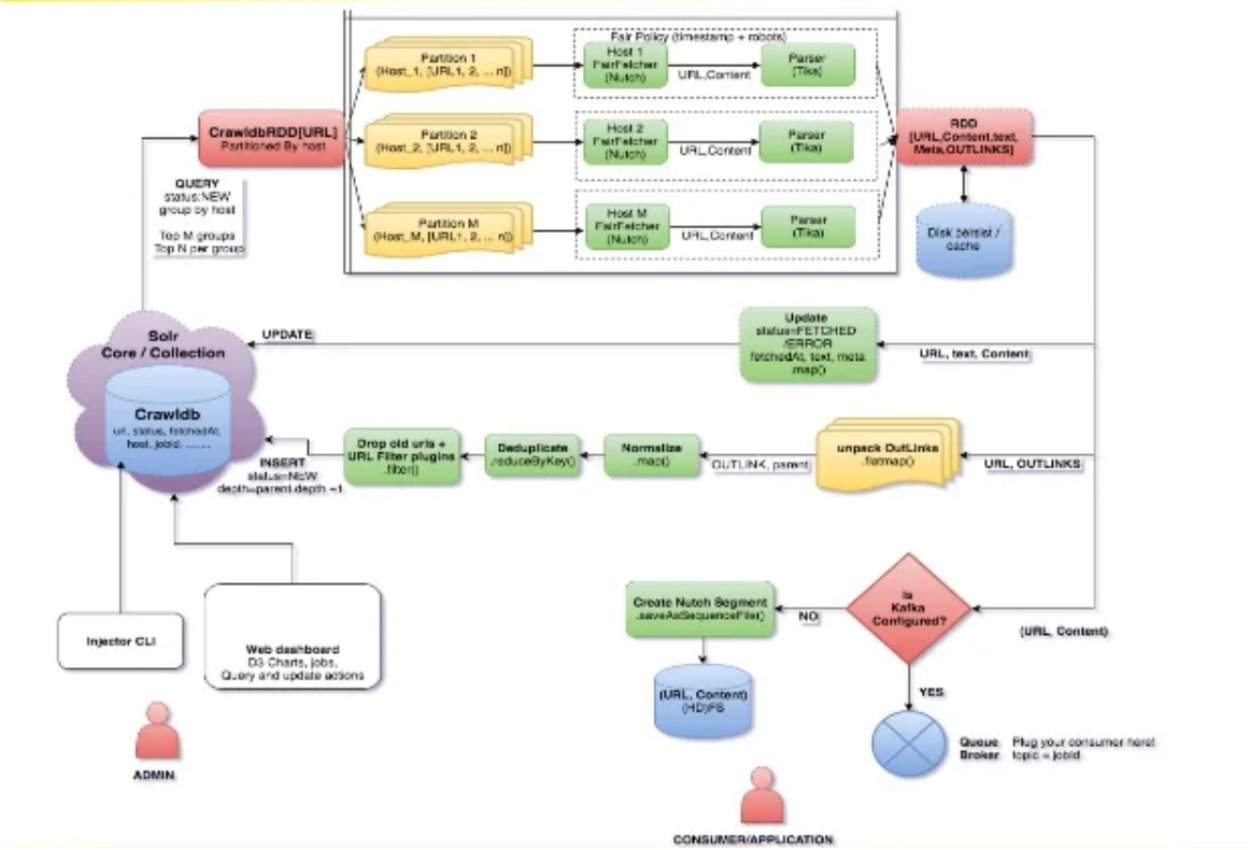
3.3.4.4 Επιλογή μέσου αποθήκευσης

Εδώ καταλλήγουν τα δεδομένα προκειμένου να επιλεχθεί ποιο μέσω θα χρησιμοποιηθεί για την τελική αποθήκευση . Συνεπώς, αν υπάρχει κάποια σύνδεση με το Apache kafka τοτε αποθηκεύεται σε αυτό αλλιώς τα δεδομένα αποθηκεύονται σε αρχεία. Σε μια πιο νέα έκδοση του sparkler δίνει την δυνατότητα να αποθηκευτούν σε Elastic Search ή Sorl ανάλογα τι έχει χρησιμοποιηθεί στην αρχική βάση με απώτερο σκοπό την χρήση κοινής Βάσης. Ετσι στο τέλος οποιαδήποτε εφαρμογή μπορεί να έχει πρόσβαση στα δεδομένα μέσα από το μέσο αποθήκευσης.



Εικόνα 3.12: Επιλογή μέσου αποθήκευσης

3.3.4.5 Πλήρης Αρχιτεκτονική



Εικόνα 3.13: Αρχιτεκτονική Sparkler

3.3.5 Προβλήματα

Κατα την εγκατάσταση του εργαλείου Sparkler (Παράρτημα 2) παρατηρήθηκαν ορισμένες δυσλειτουργίες που δεν του επιτρέπουν να λειτουργήσει με τον σωστό τρόπο. Συγκεκριμένα το βήμα που αφορά το build της εφαρμογής δεν τελειώνει με τον σωστό τρόπο καθώς περιλαμβάνει κάποιο σφάλμα. Παρόλο που εμφανίζεται το σφάλμα στην οθόνη του προγραμματιστή, φαίνεται το SNAPSHOT να έχει δημιουργηθεί. Όμως κατά την εκτέλεση του κάποια από τα δεδομένα αποθηκεύονται κενά. Αυτό σημαίνει ότι κάποιο πρόβλημα υπάρχει σε αυτό το εργαλείο και μπορεί να διαπιστωθεί καθώς στο git υπάρχει ανοιχτό issue που αφορά την μη σωστή λειτουργία του build.

3.4 Nutch

3.4.1 Εισαγωγή

Το nutch είναι ένα εξερετικά επεκτάσιμο έργο ανοιχτού κώδικα το οποίο εστιάζει στην άντληση πληροφορίας από το διαδικτυο, συγκεκριμένα η διαδικασία αυτή ονομάζεται web crawling. Το εργο αυτό χρησιμοποιεί ευρετήρια αποθήκευσης Lucene.

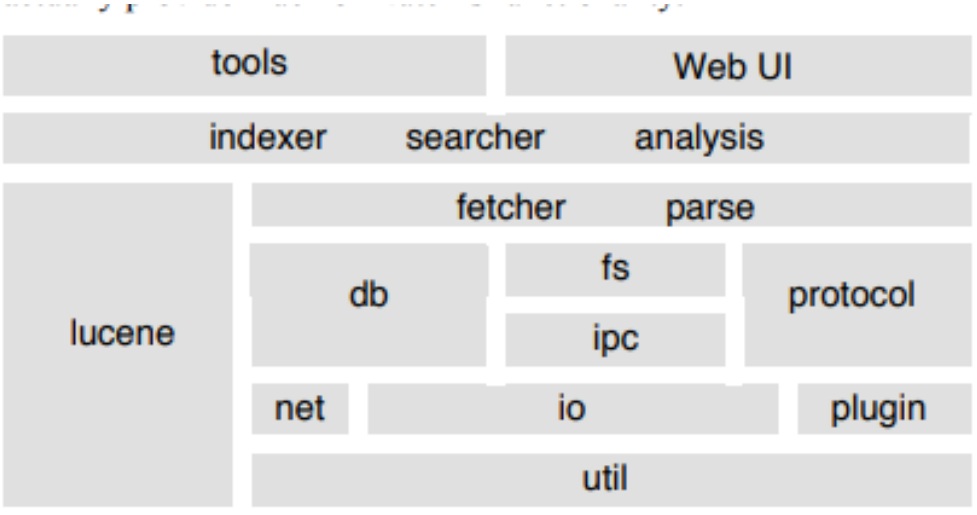
Επίσης, το nutch μπορεί να συνεργαστεί με μεγάλη ευκολία με ένα ακόμα Apache εργαλείο το apache hadoop. Συνήθως χρησιμοποιείται με την βοήθεια ενός framework εκ των apache solr ή elasticsearch.

3.4.2 Αρχιτεκτονική

Η αρχιτεκτονική του nutch είναι σχεδιασμένη με αυτόν τον τρόπο έτσι ώστε να μπορεί να χρησιμοποιεί plugins προκειμένου να κάνει διάφορες αναλύσεις πολυμέσων, ανάλυση HTML, πρωτόκολλα ανάκτησης δεδομένων και ερωτήματα. Ο κεντρικός πυρήνας του εργαλείου αυτού αποτελείται από τέσσερα κύρια κομμάτια:

- **Αναζήτηση:** Έχοντας από την αρχή γνωστό ένα ερώτημα θα πρέπει να βρεθεί ένα σύνολο εγγράφων διαφόρων μορφών και στην συνέχεια να παρουσιαστεί με τον κατάλληλο τρόπο.
- **Indexer:** Δημιουργία ενός ανεστραμμένου ευρετηρίου από το οποίο ο προγραμματιστής μπορεί να εξάγει χρήσιμη πληροφορία. Τα ευρετήρια που χρησιμοποιεί είναι τύπου Lucene.
- **Βάση δεδομένων:** Χρησιμοποιείται κυρίως για να αποθηκεύσει τα περιεχόμενα των εγγράφων που τις περισσότερες φορές αποτελούν κώδικα HTML καθώς και άλλες χρήσιμες πληροφορίες όπως είναι σύνδεσμοι, ημερομηνία κ.λ.π. Ο σκοπός της αποθηκευσης είναι να χρησιμοποιηθούν τα δεδομένα για ευρετηρίαση και για να μπορεί ο προγραμματιστής να έχει όλα τα δεδομένα μαζικά.
- **Featcher:** Σε αυτό το στάδιο το εργαλείο ζητάει ιστοσελίδες, τις αναλύει και στο τέλος εξάγει συνδέσμους από τις παραπάνω διευθύνσεις.

Παρακάτω παρουσιάζονται οι σχέσεις μεταξύ των μερών που αναλύθηκαν παραπάνω [1]:



Εικόνα 3.14: Αρχιτεκτονική Nutch

3.4.3 Indexing Κειμένου

Σύμφωνα με τις απαιτήσεις ευρετηρίου κειμένου που έχει ορίσει η ομάδα του nutch, το Lucene πληρεί τις προϋποθέσεις και για αυτόν τον λόγο χρησιμοποιείται για το συγκεκριμένο εργαλείο. Το nutch εκτός από το ευρετήριο εκμεταλεύεται επίσης την αναζήτηση ή την εύρεση διάφορων δεδομένων που μπορεί να υπάρχουν σε μια ιστοσελίδα όπως είναι η αναζήτηση λέξεων κλειδιών και εκφράσεων πολλαπλών πεδίων καθώς και κάποιων είδη κειμένων. Χρησιμοποιώντας μια τυπική αναζήτηση ιστού το κάνει να μην χρειάζεται κάποιους τύπους ευρετηρίων που δεν περιέχει η Lucene όπως είναι η αντιστοίχηση κοινωνικής έκφρασης.

Για την ευρετηρίαση κειμένου η Lucene περιέχει την δυνατότητα δημιουργίας ευρετηρίου πλήρους κειμένου ανεστραμμένου αρχείου, το οποίο αρκεί για τον σκοπό της ευρετηρίασης κειμένου αλλά δεν καλύπτει άλλες πρόσθετες λειτουργίες που χρειάζεται μια μηχανή αναζήτησης. Εκτός από το κείμενο το nutch έχει προβλέψει την αποθήκευση συνδέσμων για άμεσα στατιστικά στοιχεία καθώς και για ευκολη “δεντροποίηση” του ιστου.

Ακόμη ένα χαρακτηριστικό του Nutch είναι η διαγραφή των διπλότυπων εγγραφών. Αυτό πραγματοποιείται με την εντολή dedup η οποία αφαιρεί τις διπλοεγγραφές από τα τμήματα (Segments) ταυτόχρονα. [1]

3.4.4 Ανάλυση συνδέσμων

Για την ανάλυση συνδέσμων το Nutch χρησιμοποιεί έναν παρόμοιο με αυτόν που χρησιμοποιεί το PageRank. Ακόμα περιλαμβάνει μια πιθανότητα τυχαίου σφάλματος με τιμή 0,15 που το ονομάζει DECAY_VALUE και το οποίο εκτελείται από το Distributed Analysis Tool. Ακόμη, μπορεί με διάφορους μηχανισμούς να εντοπίσει την κατάταξη συνδέσμων και αυτό το έχει αποδείξει κάνοντας το σε 100 εκατομμύρια ιστότοπους στον παγκόσμιο ιστό.

Ακόμη, περιλαμβάνει ανάλυση κατανεμημένων συνδέσμων η οποία αποτελεί μια μαζική σύγχρονη παράλληλη διαδικασία και η οποία περιέχει πολλές φάσεις. Στην αρχή κάθε φάσης θα πρέπει να χωριστεί σε πολλά κομμάτια η λίστα των url ώστε να ενημερωθεί η βαθμολογία τους. Για την εξυπηρέτηση κάθε φάσης χρησιμοποιείται ένα ενδιάμεσο αρχείο επεξεργασίας βαθμολογίας προκειμένου να βρεθούν όλοι οι σύνδεσμοι σε σελίδες στο συγκεκριμένο κομμάτι. Στο τέλος κάθε φάσης ενημερώνεται η βαθμολογία λαμβάνοντας υπόψην τα αρχεία επεξεργασίας.

Η κατανεμημένη ανάλυση δεν χρησιμοποιεί την υπηρεσία IPC της nutch. Η εργασία επικοινωνεί μέσω της δημιουργίας αρχείων σε κοινόχρηστους φακέλους, ακριβώς με τον ίδιο τρόπο όπως γίνεται και η ανάκτηση.

3.4.5 Αναζήτηση

Η διεπαφή χρήστη αναζήτησης του Nutch εκτελείται ως ένα πρόγραμμα Java το οποίο αναλύει ένα ερώτημα κειμένου χρησιμοποιώντας την αναζήτηση ενός NutchBean. Εάν το Nutch εκτελείται μόνο σε έναν διακομιστή, μετατρέπεται το ερώτημα του χρήστη σε ερώτημα Lucene

και επιστρέφει μια λίστα επισκέψεων από το Lucene, την οποία το JSP στη συνέχεια την μετατρέπει σε HTML. Εάν το Nutch διανέμεται σε πολλούς διακομιστές, η μέθοδος αναζήτησης του NutchBean επικαλείται εξ αποστάσεως τις μεθόδους αναζήτησης άλλων NutchBeans σε άλλα μηχανήματα, τα οποία μπορούν να διαμορφωθούν είτε για να εκτελούν την αναζήτηση τοπικά είτε για κομμάτια της εργασίας σε ακόμη άλλους διακομιστές .

Η κατανεμημένη αναζήτηση έχει δημιουργηθεί πάνω από ένα προσαρμοσμένο σύνολο εργαλείων παραλληλισμού συμπλέγματος SIMD στο πακέτο net.nutch.ipc, το οποίο παρέχει μια μέθοδο αποκλεισμού «κλίσης» για την εκτέλεση της ίδιας λειτουργίας παράλληλα σε πολλούς διακομιστές και στη συνέχεια συγκεντρώνει τα αποτελέσματα μαζί. Στην ορολογία του [49] το Nutch χρησιμοποιεί partition-bydocument, όπου όλες οι δημοσιεύσεις για ένα συγκεκριμένο έγγραφο αποθηκεύονται στον ίδιο κόμβο. Κατά συνέπεια, κάθε ερώτημα πρέπει να μεταδίδεται σε όλους τους κόμβους. Στο Nutch, η κλάση net.nutch.searcher.DistributedSear h.Client παρέχει αυτήν τη λειτουργία. Υλοποιεί την ίδια διεπαφή net.nutch.searcher.Searcher που χρησιμοποιεί η Nutch για να καλεί αναζητήσεις σε τοπικά αποθηκευμένα τμήματα (segments), που αντιπροσωπεύονται από αντικείμενα του net.nutch.searcher.FetchedSegmen.[1]

3.4.6 Αλλαγές για την σωστή λειτουργία

Για τον σκοπό της συγκεκριμένης διπλωματικής χρειάστηκε να γίνουν κάποιες αλλαγές προκειμένου το nutch να επιστρέφει καθαρά τον κώδικα σε μορφή html και όχι σαν επεξεργασμένο καθαρό κείμενο. Η αλλαγή αυτή χρειάζεται προκειμένου να γίνει εξαγωγή χρήσιμης πληροφορίας από τον συνολικό κώδικα και αυτό μπορεί να επιτευχθεί μόνο στην περίπτωση που στο κείμενο υπάρχουν και τα ταγκ της κάθε σελίδας. Συγκεκριμένα θα πρέπει να αλλάξουμε το plugin parse-html στο αρχείο HtmlParser.java που υπάρχει στο μονοπάτι /plugin/parse-html/src/<package>/HtmlParser.java. Η αλλαγή που πρέπει να γίνει αφορά την γραμμή 255 η οποία θα πρέπει να μοιάζει ως εξής:

```
ParseResult parseResult = ParseResult.createParseResult(content.getUrl(),
       new ParseImpl(new String(content.getContent()), parseData));
```

3.4.7 Script εκκίνησης του Nutch

Για να γίνει χρήση του παρακάτω κώδικα θα πρέπει να γίνει εγκατάσταση του εργαλείου όπως φαίνεται στο Παράρτημα 1.

```
#!/bin/bash

topN=1000
path="/home/nutch/Desktop/nutch-crawler/nutch/runtime/local/"
depth=1
```

```
echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Initialize JAVA_HOME"
echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Initialize JAVA_HOME" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/

echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Delete Crawldb & Linkdb"
echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Delete Crawldb & Linkdb" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
rm -rf ${path}crawl/crawldb/*
rm -rf ${path}crawl/linkdb/*

for (( i=1; i<=depth; i++ ))
do
    echo ================ depth:$i ================
    echo ================ depth:$i ================ >> "logs/infoLogs$(date
"+%y-%m-%d").txt"
    if [ $i -eq 1 ]
    then
       #Inject
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Inject"
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Inject" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
       ${path}bin/nutch inject ${path}crawl/crawldb ${path}urls >>
"logs/scriptLogs$(date "+%y-%m-%d").txt"
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Inject"
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Inject" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"

       #Generate
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Generate"
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Generate" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
       ${path}bin/nutch generate ${path}crawl/crawldb ${path}crawl/segments
>> "logs/scriptLogs$(date "+%y-%m-%d").txt"
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Generate"
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Generate" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
    else
       #Generate
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Generate"
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Generate" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
```

```
       ${path}bin/nutch generate ${path}crawl/crawldb ${path}crawl/segments
-topN ${topN} >> "logs/scriptLogs$(date "+%y-%m-%d").txt"
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Generate"
       echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Generate" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
    fi

    segment=$(ls -d
/home/nutch/Desktop/nutch-crawler/nutch/runtime/local/crawl/segments/2* |
tail -1)

    #Fetch
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Fetch"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Fetch" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
    ${path}bin/nutch fetch ${segment} >> "logs/scriptLogs$(date
"+%y-%m-%d").txt"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Fetch"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Fetch" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"

    #Parse
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Parse"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Parse" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
    ${path}bin/nutch parse ${segment} >> "logs/scriptLogs$(date
"+%y-%m-%d").txt"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Parse"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Parse" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"

    #Updatedb
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Updatedb"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Updatedb" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
    ${path}bin/nutch updatedb ${path}crawl/crawldb ${segment} >>
"logs/scriptLogs$(date "+%y-%m-%d").txt"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Updatedb"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Updatedb" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
done

#Invertlinks
```

```
echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Invertlinks"
echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Invertlinks" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
${path}bin/nutch invertlinks ${path}crawl/linkdb -dir ${path}crawl/segments
>> "logs/scriptLogs$(date "+%y-%m-%d").txt"
echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Invertlinks"
echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Invertlinks" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"

for y in `ls -d
/home/nutch/Desktop/nutch-crawler/nutch/runtime/local/crawl/segments/2*`
do
    echo ==================== Save Segment ==================
    #Index
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Index"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Start Index" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
    ${path}bin/nutch index ${path}crawl/crawldb/ -linkdb
${path}crawl/linkdb/ ${y} -filter -normalize -deleteGone >>
"logs/scriptLogs$(date "+%y-%m-%d").txt"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Index"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Finish Index" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"

    #Delete Segment
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Delete Segment $y"
    echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] Delete Segment $y" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
    rm -rf $y
done

echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] exit from program"
echo "[INFO] [$(date "+%y-%m-%d %H:%M:%S")] exit from program" >>
"logs/infoLogs$(date "+%y-%m-%d").txt"
```

4. Τελική Εφαρμογή

4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα δημιουργηθεί μια τελική εφαρμογή η οποία θα παρουσιάζει τα δεδομένα, που υπάρχουν στο elasticsearch, στον χρήστη. Αυτή η εφαρμογή θα αποτελείται από το api το οποίο θα είναι ο μεσολαβητής ανάμεσα στο elasticsearch και τους υπόλοιπους κόμβους (Παράρτημα 3 - Τελική Εφαρμογή: API). Τέλος θα περιέχει έναν scraper που θα συλλέγει τα δεδομένα (Παράρτημα 3 - Τελική Εφαρμογή: Scraper) και την τελική εφαρμογή το γραφικό περιβάλλον που θα βλέπει ο χρήστης (Παράρτημα 3 - Τελική Εφαρμογή: Front end).

4.2 Τεχνολογίες

4.1.1 Nodejs

Το Node.js (από εδώ και στο εξής μόνο το Node) είναι ένα περιβάλλον χρόνου εκτέλεσης για τη γλώσσα JavaScript που βασίζεται στη μηχανή JavaScript V8. Το Node παρέχει τη δυνατότητα εγγραφής εφαρμογών JavaScript back end. Η ανάπτυξη ενός προγράμματος βασίζεται σε JavaScript για την αλληλεπίδραση και την τροποποίηση του μοντέλου αντικειμένου εγγράφου στο πρόγραμμα περιήγησης. Τωρα οι προγραμματιστές χρησιμοποιώντας το Nodejs δεν χρειάζεται να αλλάξουν την γλώσσα προγραμματισμού εάν αναπτύσουν το μπροστινό ή το πίσω μέρος μιας εφαρμογής. Τέλος, η δημοφιλία του nodejs βασίστηκε στην δημοφιλία που είχε ήδη η γλώσσα προγραμματισμού Javascript [13].

4.1.1.1 Ασύγχρονη λειτουργία

Η υποκείμενη βελτιστοποίηση απόδοσης από τον V8 engine δεν είναι το μόνο μέρος του Node που κάνει τη JavaScript πολύ επεκτάσιμη και αποτελεσματική. Από πάντα οι διακομιστές χρησιμοποιούν πολλαπλά νήματα για να χειριστούν την εισερχόμενη κίνηση και τις βαριές λειτουργίες I/O. Σε αντίθεση με αυτό, το Node δεν χρησιμοποιεί πολλαπλά νήματα για την αντιμετώπιση των λειτουργιών του. Αντίθετα, βασίζεται σε ένα μόνο νήμα, αλλά χρησιμοποιεί το πρότυπο προγραμματισμού βάση συμβάντων. Αυτό σημαίνει ότι σχεδόν κάθε λειτουργία του Node είναι ασύγχρονη [13].

4.1.2 Expressjs

Το express.js είναι μια βιβλιοθήκη βασισμένη στο nodejs. Προσφέρει ένα αρκετά απλοποιημένο API για ορισμένες από τις βασικές λειτουργίες του Node. Αποτελεί ένα στρώμα αφαίρεσης πάνω από το http που περιέχει το βασικό API του Node. Το express παρέχει μια πληθώρα από λειτουργίες βασισμένες στο Http και οι οποίες μπορούν να εκτελεστούν απευθείας για τον χειρισμό αιτημάτων, τον καθορισμό κάποιον endpoint κλπ [13].

4.1.2.1 Middleware

Το Middleware είναι ένα χαρακτηριστικό του express το οποίο ασχολείται μοναχά με αιτήματα, απαντήσεις και επακόλουθες λειτουργίες ενδιάμεσου λογισμικού. Κάθε διακομιστής που δημιουργείται στο Node ακούει σε κάποια συγκεκριμένη θύρα για εισερχόμενα αιτήματα που θα τοποθετηθούν σε κάποιον βρόχο προκειμένου να γίνει η κατάλληλη διαχείριση τους. Αυτό σημαίνει ότι η διαχείριση αυτών γίνεται μόνο από έναν μονολιθικό χειριστή αιτημάτων. Έτσι το Middleware αποτελεί μια συνάρτηση που διαχωρίζει το μονολιθικό σε πολλά μεμονωμένα βήματα. Αυτά τα βήματα μπορεί να είναι εκτέλεση κώδικα, αλλαγές σε αντικείμενα αιτήματος και απόκρισης κλπ [13].

```
const express = require("express");
const http = require("http");

let app = express();

app.use((req, res, next) => {
    console.log("Incoming request url: "
            + req.url);
    res.end("Hello, World!");
});

http.createServer(app).listen(8000);
```

Εικόνα 4.1: Middleware

4.1.2.2 Routing

Ένα από τα βασικά χαρακτηριστικά της βιβλιοθήκης expressjs είναι το Routing και αυτό γιατί πολύ σπάνια μια εφαρμογή ιστού έχει μόνο μια σελίδα που συνδέεται με μια διεύθυνση URL. Ακόμα κι αν έχουμε μια εφαρμογή με μια σελίδα θα υπάρχουν διευθύνσεις με παραμέτρους για ερωτήματα ή αναγνωριστικά χρήστη. Η Express έχει ενσωματωμένες λειτουργίες για το χειρισμό εισερχόμενων αιτημάτων HTTP με ένα καθορισμένο URL που θα αντιστοιχιστεί σε έναν χειριστή αιτημάτων. Οι πιο σημαντικές μέθοδοι που χρησιμοποιεί για τα αιτήματα αυτά είναι οι get(), post(), put() και delete(). Με αυτές τις μεθόδους μια εφαρμογή express μπορεί να δημιουργήσει ένα πλήρες API (CRUD) [13].

```
app.use( (req, res, next) => {
    console.log("Request URL: " + req.url);
    console.log("Request Date: " + new Date());
    next();
});

app.get('/', (req, res, next) => {
        /* show homepage */
});

app.get('/login', isAuthenticated,
    (req, res, next) => {
        /* proceed to login page */
    });

app.get('/users/:userid',
    (req, res, next) => {
        /* Parse :userid and show user's page*/
});

app.use( (req, res, next) => {
        res.status(404).send(
          "404 Page not found!");
});
```

Εικόνα 4.2: Routing

4.1.2.3 Static Files

Μια εφαρμογή ιστού συνήθως έχει ένα στατικό μέρος στο οποίο αποθηκεύει αρχεία που πρέπει να αποσταλούν στους χρήστες. Αυτά τα στατικά αρχεία μπορεί να είναι αρχεία HTML, CSS ή JavaScript π.χ. που περιέχει ένα πρότυπο που μπορεί να γεμίσει με δεδομένα στην πλευρά του πελάτη. Αλλά ακόμα κι αν είναι συμπληρωμένα στην πλευρά του πελάτη, ο διακομιστής του πίσω άκρου πρέπει να μπορεί να τους παρέχει αυτά τα ίδια αρχεία με κάποιο τρόπο. Είναι δυνατό να γίνει αυτό σε απλό Node, αλλά το Express έχει μια βοηθητική λειτουργία σχετικά με αυτό ακριβώς.

```
const staticPath = path.resolve(__dirname,
        "static");

app.use(express.static(staticPath));
```

Εικόνα 4.3: Static Files

4.1.3 React

Το React είναι μια βιβλιοθήκη διεπαφής χρήστη που αναπτύχθηκε στο Facebook για να διευκολύνει τη δημιουργία διαδραστικών και επαναχρησιμοποιήσιμων στοιχείων διεπαφής χρήστη. Χρησιμοποιείται ήδη από το ίδιο το Facebook στις πραγματικές εφαρμογές που δημιουργεί μιας και είναι ο δημιουργός. Το ReactJS είναι το καλύτερο για την απόδοση

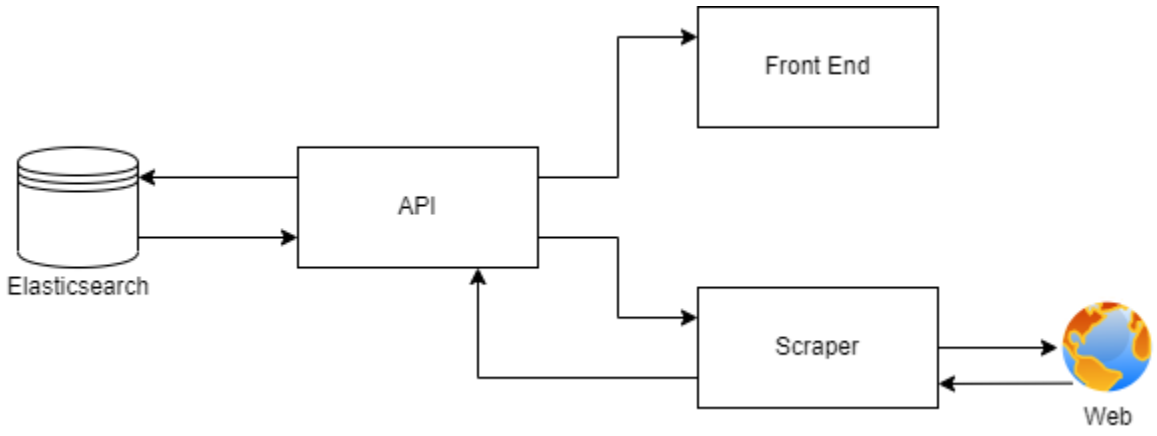
σύνθετων διεπαφών χρήστη με υψηλή απόδοση. Το βασικό θεμελιώδες πίσω από το React είναι η έννοια του εικονικού DOM. Το ReactJS χρησιμοποιεί αποτελεσματικά το εικονικό DOM, το οποίο μπορεί να αποδοθεί είτε από την πλευρά του πελάτη είτε από την πλευρά του διακομιστή και να επικοινωνεί εμπρός και πίσω. Το Virtual DOM αποδίδει υποδέντρα κόμβων με βάση τις αλλαγές κατάστασης. Κάνει τον ελάχιστο δυνατό χειρισμό DOM για να διατηρεί ενημερωμένα τα στοιχεία σας. Το React είναι πιο ελαφρύ από το Angular, είναι γεμάτο με τις λιγότερες συνθήκες και εξαλείφει την ανάγκη χρήσης επιπλέον στοιχείων όπως πρόσθετα [14].

4.1.3.1 Virtual DOM

Όπως το πραγματικό DOM, το Virtual DOM είναι ένα δέντρο κόμβων που παραθέτει στοιχεία, τα χαρακτηριστικά και το περιεχόμενό τους ως αντικείμενα και ιδιότητες. Η react περιέχει την μέθοδο render() η οποία δημιουργεί ένα δέντρο κόμβου από τα στοιχεία React και ενημερώνει αυτό το δέντρο με νέες τιμές που αλλάζουν από ενέργειες κατά την διάρκεια που ένας χρήστης αλληλεπιδρά. Κάθε φορά που αλλάζουν τα δεδομένα σε μια εφαρμογή React, δημιουργείται μια νέα αναπαράσταση Virtual DOM της διεπαφής χρήστη [14].

4.3 Αρχιτεκτονική

Στην παρούσα εφαρμογή για την αποθήκευση των δεδομένων έχει χρησιμοποιηθεί το Elasticsearch. Σε αυτό έχει πρόσβαση μόνο το API το οποίο είναι υπεύθυνο στο να γράφει και να τραβάει δεδομένα. Ο κόμβος scraper παρέχει δύο λειτουργίες. Αρχικά ζητάει από τον κόμβο API όλα τα url που πρόκειται να αναζητήσει, δεδομένα και ότι πληροφορία εντοπίζει ξανα επικοινωνεί με αυτό προκειμένου να αποθηκευτούν. Η δεύτερη αφορά την επίσκεψη του σε ιστοσελίδες που μας ενδιαφέρει να τραβήξουμε πληροφορίες. Και ο τελευταίος κόμβος αφορά την εφαρμογή η οποία δημιουργήθηκε και η οποία το μόνο που κάνει είναι τραβάει δεδομένα από τον κόμβο του API



Εικόνα 4.4: Αρχιτεκτονική Εφαρμογής

4.4 API

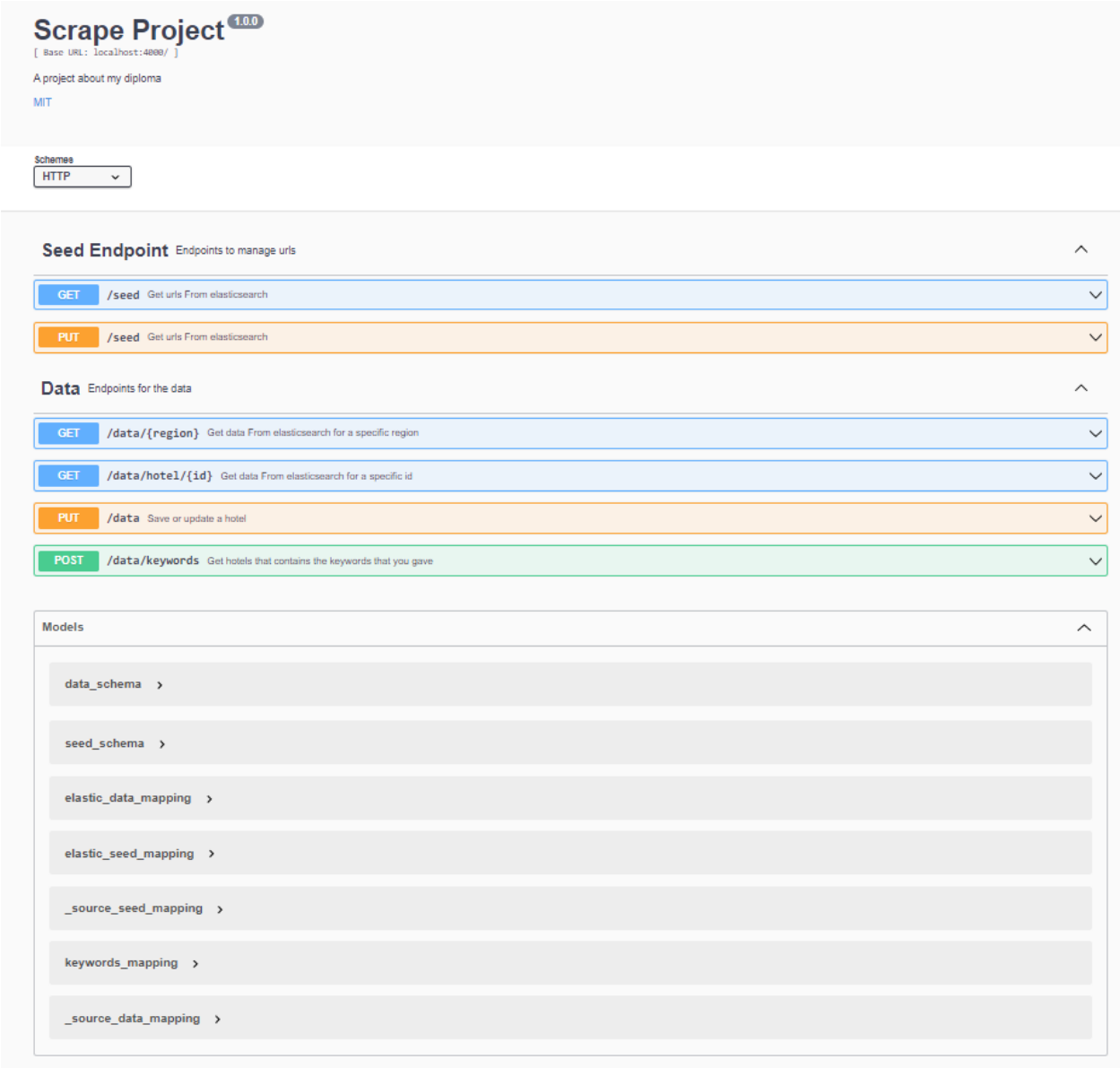
Για τον σκοπό της παρούσας εφαρμογής δημιουργήθηκε ένα API με σκοπό κάθε εξωτερικό πρόγραμμα που θέλει δεδομένα να τα ζητάει από το ιδιο και να μην εχει απευθείας πρόσβαση στο Elasticsearch. Για την εύκολη εύρεση των endpoint έχει δημιουργηθεί και swagger το οποίο παρουσιάζει τις κλήσεις με γραφικό τρόπο. Συνεπώς, για την επικοινωνία δημιουργήθηκαν διάφορες κλήσεις όπως φαίνεται παρακάτω:

- GET /seed: επιστρέφει όλες τις διευθύνσεις για τις οποίες πρέπει να γίνει συλλογή δεδομένων
- PUT /seed: με αυτή την κλήση μπορούμε να αποθηκεύσουμε ή να τροποποιήσουμε μια διεύθυνση. Για την σωστή λειτουργία του χρειάζεται να προστεθεί ένα body:

```
{
  "url": "string",
  "channel": "string",
  "region": "string"
}
```

- GET /data/hotel/{id}: επιστρέφει τα δεδομένα από ένα συγκεκριμένο ξενοδοχείο.
- POST /data: προσθέτει ένα νέο ξενοδοχείο στα δεδομένα.
- GET /data/{region}: φέρνει όλα τα δεδομένα για μια συγκεκριμένη περιοχή
- POST /data/keywords: βρίσκει ξενοδοχεία τα οποία περιέχουν κάποιες λέξεις κλειδιά

Swagger



Εικόνα 4.5: Η κλήσεις του Api μέσω του Swagger

Ολόκληρος ο κώδικας που αφορά το API της τελικής εφαρμογής υπάρχει στο Παράρτημα 3 - Τελική εφαρμογή: API.

4.5 Scraper

Τον Scraper αποτελεί ένα πρόγραμμα που επισκέπτεται κάποιες σελίδες και μαζεύει δεδομένα. Συγκεκριμένα κατά την έναρξη του προγράμματος τραβάει όλα τα url από το API προκειμένου να ξεκινήσει να τα επισκέπτεται. Για κάθε ένα url εντοπίζει όλες τις διευθύνσεις που αφορούν προβολή επιχείρησης και τις αποθηκεύει σε έναν τοπικό πίνακα. Στην συνέχεια για κάθε ιστοσελίδα του παραπάνω πίνακα επισκέπτεται και εντοπίζει όλες τις πληροφορίες που εμείς του έχουμε καθορίσει. Οι πληροφορίες εξαρτώνται από το JSON με όνομα tags που έχει αρχικοποιηθεί για την Booking.com και το οποίο περιλαμβάνει όλα τα tags για κάθε

συγκεκριμένη πληροφορία. Στο τέλος της διαδικασίας καλεί εκ νέου το API προκειμένου να κάνει την αποθήκευση των δεδομένων.

JSON Tags

```
const tags = {
   hotelName: "string",
   hotelName_sec: "string",
   hotelName_tag: "string",
   address: "string",
   descr: "string",
   stars: "string",
   headerOfDescription: "string",
   facilities: "string",
   facilitiesPolicy: "string",
   facilitiesPer: "string",
   facilitiesTitle: "string",
   globalscoreText: "string",
   globalscoreScore: "string",
   globalscore: "string",
   globalscoreTitle: "string",
   globalscoreScPrSc: "string",
   surroundings: "string",
   surroundingsTitle: "string",
   surroundingsPr: "string",
   photos: "string"
}
```

JSON Data

```
let data = {
   hotelName: “string”,
   address: “string”,
   descr: “string”,
   stars: “number”,
   headerOfDescription: “string”,
   facilities: “array”,
   globalscoreText: “string”,
   globalscoreScore: “number”,
   globalscore: “array”,
   surroundings: “array”,
   photos: “array”,
   region: “string”,
   id: “string”
}
```

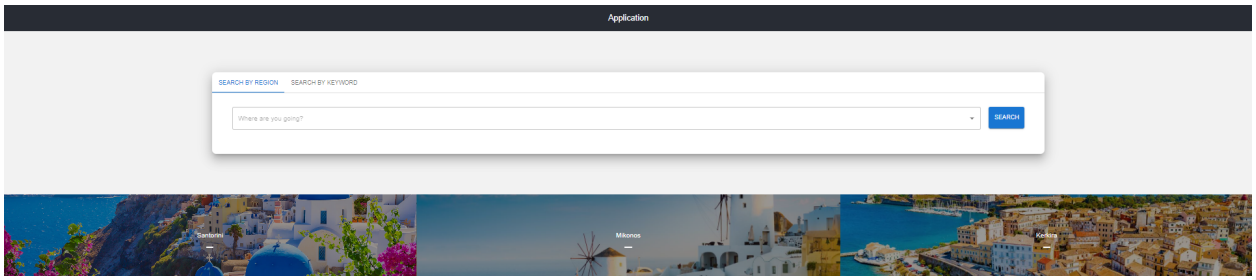
Για το κομμάτι που αφορά την υλοποίηση του Scraper έχει προστεθεί ολόκληρος ο κώδικας στο κεφάλαιο Παράρτημα 3 - Τελική εφαρμογή: Scraper.

4.6 Front end

Η τελική εφαρμογή έχει και το κομμάτι του frontend το οποίο περιέχει 3 βασικες οθόνες:

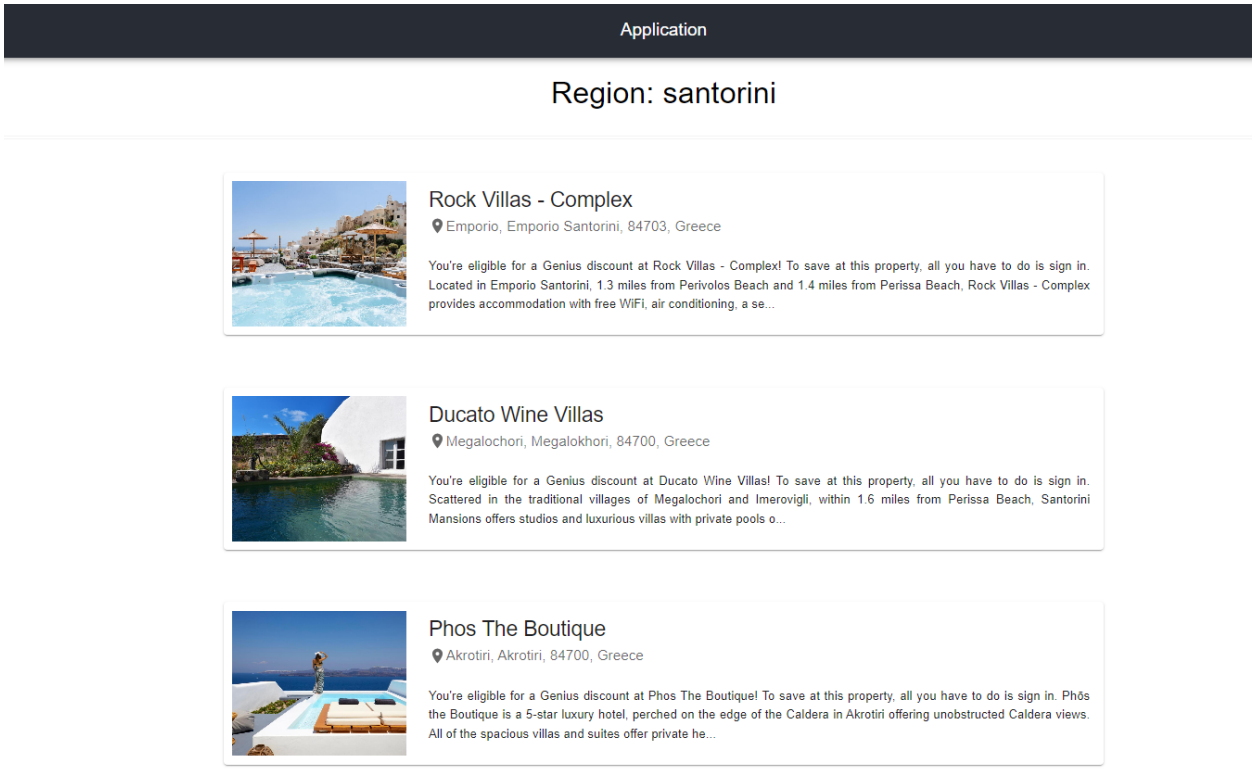
- Η πρώτη επιτρέπει στον χρήστη να αναζητήσει με βάση την περιοχή που θέλει να επισκεφτεί ή με βάση κάποια λέξη κλειδί π.χ “Parking”. Επίσης μπορεί να επιλέξει απευθείας μια από τις προτεινόμενες περιοχές για να δει τα ξενοδοχεία κατευθείαν.
- Η δεύτερη αφορά τα αποτελέσματα της αναζήτησης τα οποία εμφανίζονται με σειρά και με φιλική διάταξη ως προς τον χρήστη. Η συγκεκριμένη σελίδα περιέχει και σελιδοποίηση προκείμενου να εμφανίζονται συγκεκριμένα αποτελέσματα την στιγμή που η περιοχή έχει πολλά.
- Την τρίτη σελίδα αποτελεί η προβολή της επιχείρησης στην οποία ο χρήστης μπορεί να δει διάφορα στοιχεία αυτής, όπως:
    - Φωτογραφίες
    - Περιγραφή
    - Χαρακτηριστικά
    - Βαθμολογία
    - Διεύθυνση
    - Όνομα

Αρχική Οθόνη



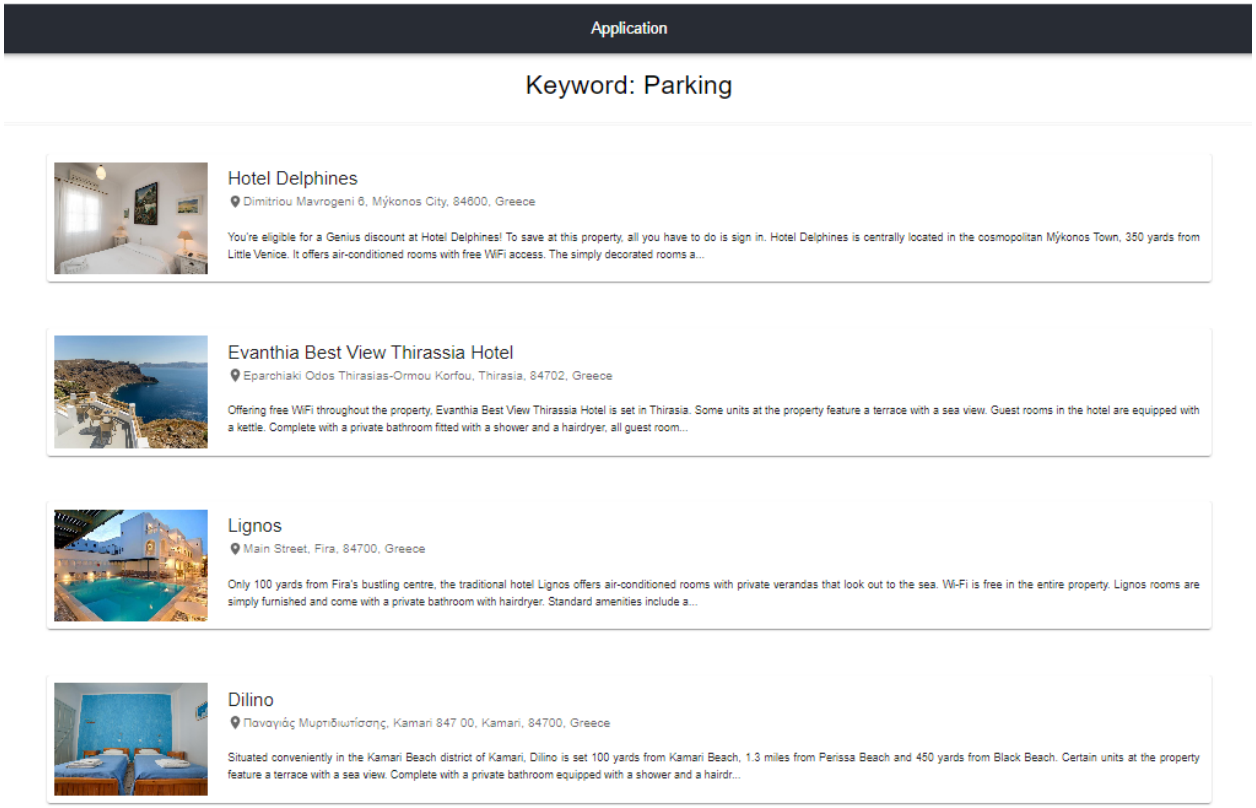
Εικόνα 4.6: Αρχική Οθόνη

Αναζήτηση με βάση την περιοχή



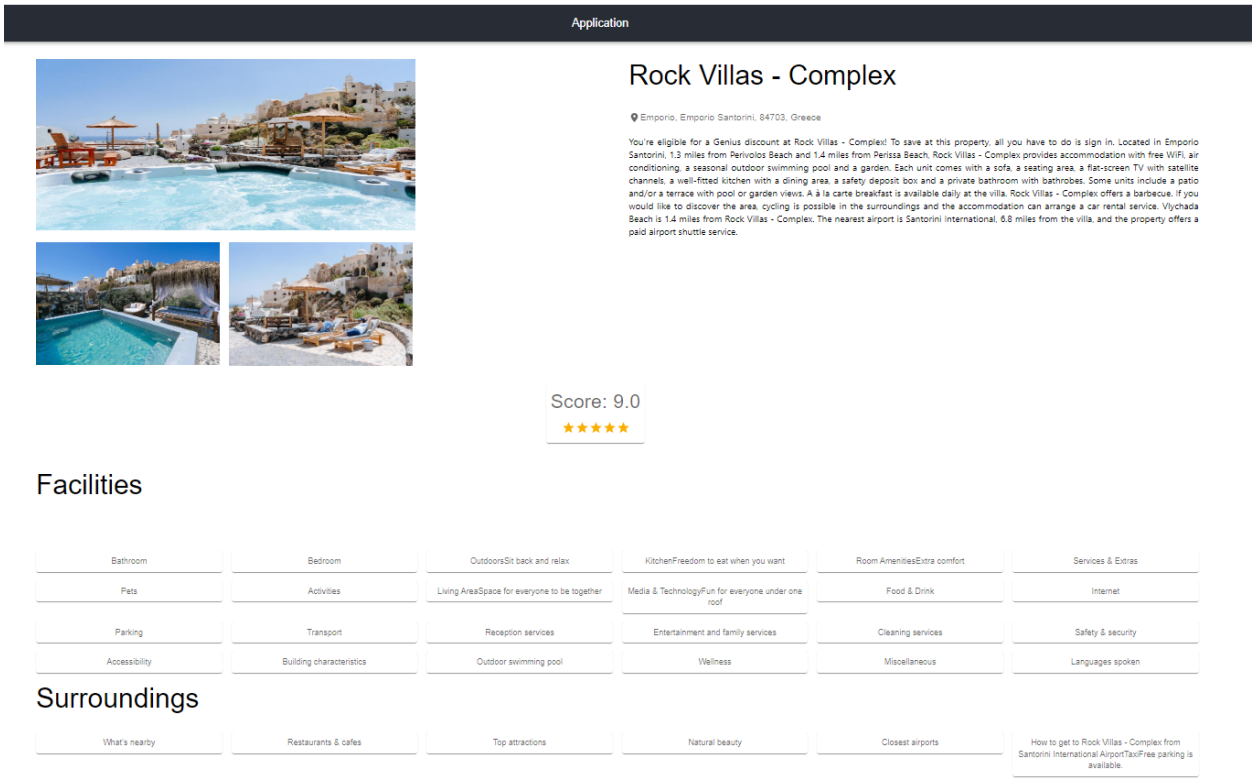
Εικόνα 4.7: Οθόνη αναζήτησης περιοχής

Αναζήτηση με βάση λέξη κλειδί



Εικόνα 4.8: Οθόνη αναζήτησης με λέξη κλειδί

Προβολή Επιχείρησης



Εικόνα 4.9: Οθόνη προβολής επιχείρησης

Ολόκληρος ο κώδικας που αφορά το γραφικό κομμάτι της τελικής εφαρμογής υπάρχει στο Παράρτημα 3 - Τελική εφαρμογή: Front End.

5. Βελτιώσεις & Συμπεράσματα

5.1 Εισαγωγή

Το τελευταίο κεφάλαιο έχει ως σκοπό να αναλυθούν κάποιες βελτιώσεις που έχουν εντοπιστεί και θα μπορούσαν να γίνουν προκειμένου το σύστημα να γίνει καλύτερο και πιο αποτελεσματικό ως προς τον χρήστη. Επίσης θα παρουσιαστούν τα τελικά συμπεράσματα από την παρούσα διπλωματική σε σχέση με την χρήση των εργαλείων που χρησιμοποιήθηκαν.

5.2 Χρήση πολλαπλών καναλιών και εντοπισμό αλλαγών

Στην υλοποίηση που έχει γίνει για αυτή την εφαρμογή τα δεδομένα προέρχονται από μόνο ένα κανάλι την Booking.com. Συνεπως θα γινόταν να προστεθούν κι άλλα με τα παρακάτω βήματα:

1. Προσθήκη πεδίου booking για να γνωρίζει ο scraper αλλά και ολη η εφαρμογή από που προέρχεται το αίτημα

2. Εύρεση tags για όλα τα άλλα κανάλια που θέλουμε να προστεθούν
3. Και δημιουργία νέων ενεργειών αν χρειάζονται πχ πάτημα κάποιου κουμπιού για να εμφανιστεί μια πληροφορία.

Επίσης θα πρέπει να προστεθεί ένας μηχανισμός ο οποίος θα εντοπίζει περιπτώσεις αλλαγών των tag κάποιου καναλιού προκειμένου ο προγραμματιστής να τα αλλάξει. Για τον προγραμματιστή θα ήταν εύκολο να αλλάζει ή να προσθέτει τα tag από κάποιο γραφικό περιβάλλον για μεγαλύτερη ταχύτητα (δημιουργία κάποιου διαχειριστικού).

5.3 Δημιουργία Βιβλιοθήκης

Ο scraper είναι ένα πρόγραμμα το οποίο ξεκινάει να συλλέγει πληροφορίες. Εφόσον αυτό αφορά πρόγραμμα δεν είναι εφικτό να χρησιμοποιηθεί από κάποια άλλη εφαρμογή εκτός της εφαρμογής που έχει ήδη δημιουργηθεί. Για αυτό θα ήταν προτιμότερο να γίνει όλο το module του scraping σαν μια βιβλιοθήκη η οποία θα γίνεται install από τους χρήστες και θα μπορούν να το χρησιμοποιούν . Σαν είσοδο αυτή η βιβλιοθήκη θα πρέπει να δέχεται το url της περιοχής στο συγκεκριμένο κανάλι και το πρόγραμμα θα πρέπει να καταλαβαίνει τι ενέργειες πρέπει να κάνει προκειμένου να συλλέξει τις πληροφορίες. Για να γίνει αυτό θα πρέπει να ακολουθηθούν κάποια βήματα. Συγκεκριμένα:

1. Χρησιμοποιούμε την εντολή export στην κεντρική συνάρτηση.
2. Εκτελούμε `npm publish` και `npm publish --access public`
3. Δημιουργούμε έναν φάκελο έξω από τον φάκελο του project `mkdir test-directory`
4. Μπαίνουμε στον φάκελο που μόλις δημιουργήθηκε `cd /test-directory`
5. Στον τεστ φάκελο μπορούμε να εκτελέσουμε την εντολή `npm install <your-module-name>` (το module-name είναι αυτό που υπάρχει στο package.json αρχείο) και θα δούμε ότι η βιβλιοθήκη έχει προστεθεί.

5.4 Χρήση Μηχανικής Μάθησης

Η χρήση μηχανικής μάθησης θα βοηθούσε αρκετά την παρούσα εφαρμογή προκειμένου να κάνει το περιβάλλον πιο φιλικό προς τον χρήστη. Αρχικά, θα πρότεινε περιοχές όχι με βάση τον προγραμματιστή αλλά με βάση την επισκεψιμότητα, την βαθμολογία, κλπ, σε συνεργασία με τα ενδιαφέροντα που εμφανίζονται στο προφίλ του χρήστη. Επίσης, στα αποτελέσματα θα μπορούσε να βγάζει σαν πρώτο ένα ξενοδοχείο που μπορεί να έρχεται πιο κοντά στα ενδιαφέροντα του χρήστη. Κατα την είσοδο του στην προβολή της επιχείρησης θα ήταν αρκετά ελκυστικό να του πρότεινε κι αλλα με καλύτερα ή ιδια χαρακτηριστικά. Τελος, θα μπορούσε να τον προτρέψει να ξανά επισκεφτεί ξενοδοχεία για τα οποία είχε αφιερώσει τον περισσότερο χρόνο προκειμένου να τον πείσει να κάνει κάποια κράτηση.

5.5 Συμπεράσματα

Στο πλαίσιο αυτής της διπλωματικής έγινε εγκατάσταση δύο υπάρχον εργαλείων που αυτοματοποιούν το ανίχνευση δεδομένων σε ιστοσελίδες, το Sparkler και το Nutch. Κατά την χρήση αυτών εντοπίστηκαν δυσλειτουργίες που αναλύθηκαν και επιβλήθηκαν προκειμένου να

εξεταστεί το εργαλείο και να καταλλήξουμε πως αυτό μπορεί να εκπληρώσει την συγκεκριμένη διαδικασία που πρέπει να γίνει.

Sparkler

Για το συγκεκριμένο εργαλείο δεν έγινε σωστά το build διότι γίνεται με την εντολή ant και στο αποθετήριο του εργαλείου υπάρχουν άλυτα προβλήματα με την χρήστη ant από το 2020. Έγινε προσπάθεια να χρησιμοποιηθεί μια παλαιότερη έκδοση προκειμένου να χρησιμοποιηθεί το εργαλείο παρόλα αυτά και πάλι δεν λειτουργούσε και αυτό οφείλεται στο ότι στο build αρχείο υπάρχουν βιβλιοθήκες των οποίων η έκδοση δεν υπάρχει πλέον.

Nutch

Διαπιστώθηκε ότι υπάρχει η δυνατότητα να επισκέπτεται σελίδες και να επιστρέφει όλο τον πηγαίο κώδικα. Παρόλα αυτά δεν επιτρέπει την πρόσβαση σε σελίδες που είναι ανενεργές στο αρχείο robot.txt κάθε σελίδας. Για την παρούσα εφαρμογή χρειαζόταν να χρησιμοποιηθούν σελίδες αναζήτησης που συνήθως είναι ανενεργές.

Για τους παραπάνω λόγους, σχεδιάστηκε και αναπτύχθηκε από την αρχή ένα εργαλείο το οποίο επισκέπτεται σελίδες και εξάγει την πληροφορία που μας ενδιαφέρει.

Πηγές

Άρθρα

[1]Rohit Khare, Doug Cutting, Kragen Sitaker, and Adam Rifkin, “Nutch: A Flexible and Scalable Open-Source Web Search Engine,” Nov. 2004.

[2]Md. AbuKausar, V. S. Dhaka, and S. Kumar Singh, “Web crawler: A review,” *International Journal of Computer Applications*, vol. 63, no. 2, pp. 31–36, Feb. 2013, doi: 10.5120/10440-5125.

[3]T. Allison *et al.*, “Research Report: Building a Wide Reach Corpus for Secure Parser Development,” in *2020 IEEE Security and Privacy Workshops (SPW)*, May 2020. Accessed: Dec. 26, 2022. [Online]. Available: http://dx.doi.org/10.1109/spw50608.2020.00066

[4]T. Gowda, K. Singh, and C. Mattmann, “Sparkler—Crawler on Apache Spark: Spark Summit East talk by Karanjeet Singh and Thamme Gowda Narayanaswamy,” 2017.

[5]Babak Bashari Rad, Harrison John Bhatti, and Mohammad Ahmadi, “An Introduction to Docker and Analysis of its Performance,” Mar. 2017.

[6]James Hamilton, Brad Schofield, Manuel Gonzalez Berges, and Jean-Charles Tournier,

"SCADA STATISTICS MONITORING USING THE Elastic Stack (Elasticsearch, Logstash, Kibana)", doi: :10.18429/JACoW-ICALEPCS2017-TUPHA034.

[7]Sumukh Bhandarkar and Nandita B N, "A Full-Text-Based Search Algorithm vs Elasticsearch ," Mar. 2020.

[8]John Ferguson Smart, "An introduction to Maven 2," May 2005.

[9]Nicolás Serrano and Ismael Ciordia, "Ant: Automating the Process of Building Applications," 2004.

[10]Abdelkader Lahmadi and Frédéric Beck, "Powering Monitoring Analytics with ELK stack," 2015.

[11]Martin Odersky *et al.*, "An Overview of the Scala Programming Language," 2004.

[12]van der Kuijl, M.G., "Remote and parallel test automation at the GUI level using a generic adapter," Aug. 2021.

[13]Christian Peters, "Rich Internet Applications w/HTML and Javascript," Feb. 2017.

[14]A. Kumar and R. Kumar, "Comparative analysis of angularjs and reactjs," *International Journal of Latest Trends in Engineering and Technology*, vol. 7, no. 4, 2016, doi: 10.21172/1.74.030.

Ιστοσελίδες

https://www.edureka.co/blog/spark-architecture/
https://aws.amazon.com/opensearch-service/the-elk-stack/what-is-elasticsearch/
https://wiki.linuxfoundation.org/networking/net-tools
https://kyivenergo.com/page-53/maven/
https://linux-user.gr/t/docker-ti-einai-kai-pws-to-egkathistoyme-ston-ypologisth-mas/2010
https://en.wikipedia.org/wiki/Scala_(programming_language)
https://github.com/USCDataScience/sparkler
https://outsourceit.today/comparison-open-source-web-crawlers/#10ef
https://mechanicalsoup.readthedocs.io/en/stable/index.html
http://stormcrawler.net/
https://polynoe.lib.uniwa.gr/xmlui/bitstream/handle/11400/1430/Mladenova_161116.pdf?sequence=1&isAllowed=y
https://github.com/USCDataScience/sparkler/issues/157
https://securityforeveryone.com/blog/top-8-open-source-web-crawlers

Παράρτημα 1

Εγκατάσταση Nutch

Εγκατάσταση της Java

```
sudo apt-get install openjdk-8-jdk
```

```
panaarva@ubuntu:~$ sudo apt-get install openjdk-8-jdk
[sudo] password for panaarva:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
java -version
```

```
panaarva@ubuntu:~$ java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1~20.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
panaarva@ubuntu:~$
```

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/
```

```
nutch@ubuntu:~/Desktop/nutch-crawler/elasticsearch-7.4.2$ export JAVA_HOME=/usr/
lib/jvm/java-11-openjdk-amd64/
```

Εγκατάσταση του Git

```
sudo apt install git
```

```
panaarva@ubuntu:~$ sudo apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
git --version
```

```
panaarva@ubuntu:~$ git --version
git version 2.25.1
panaarva@ubuntu:~$
```

Δημιουργία κεντρικού φακέλου

Θα χρειαστεί να δημιουργηθεί ένας φάκελος στον οποίο θα εγκατασταθεί το project καθώς και κάθε εργαλείο που θα χρειαστεί σε αυτό

```
mkdir nutch-crawler
```

```
panaarva@ubuntu:~/Desktop$ ls
nutch-crawler
panaarva@ubuntu:~/Desktop$
```

```
cd nutch-crawler
```

```
panaarva@ubuntu:~/Desktop$ cd nutch-crawler
panaarva@ubuntu:~/Desktop/nutch-crawler$
```

Εγκατάσταση του ant

Το ant χρησιμοποιείται κυρίως για το build εφαρμογών που έχουν υλοποιηθεί με Java. Παρόλο αυτά μπορεί να χρησιμοποιηθεί για έργα που έχουν χτιστεί σε C και C++.

```
sudo apt install ant
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler$ sudo apt install ant
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Εγκατάσταση του Project

```
git clone https://github.com/apache/nutch.git
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler$ git clone https://github.com/apache/nut
ch.git
Cloning into 'nutch'...
remote: Enumerating objects: 67498, done.
remote: Counting objects: 100% (1331/1331), done.
remote: Compressing objects: 100% (536/536), done.
remote: Total 67498 (delta 386), reused 1192 (delta 330), pack-reused 66167
Receiving objects: 100% (67498/67498), 133.27 MiB | 5.29 MiB/s, done.
Resolving deltas: 100% (32310/32310), done.
panaarva@ubuntu:~/Desktop/nutch-crawler$
```

```
cd nutch
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler$ cd nutch
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch$
```

Επεξεργασία του αρχείου nutch-site.xml προκειμένου να συμπίπτει με το παρακάτω παράδειγμα:

```
vi conf/nutch-site.xml
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
   <name>plugin.folders</name>
   <value>/home/panaarva/Desktop/nutch-crawler/nutch/build/plugins</value>
```

```
 </property>
</configuration>
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
   <name>plugin.folders</name>
   <value>/home/panaarva/Desktop/nutch-crawler/nutch/build/plugins</value>
 </property>
</configuration>
~
~
~
~
~
~
~
~
~
~
~
"conf/nutch-site.xml" [New File]
```

Build Εφαρμογής

```
ant clean runtime
```

```
runtime:
    [mkdir] Created dir: /home/panaarva/Desktop/nutch-crawler/nutch/runtime
    [mkdir] Created dir: /home/panaarva/Desktop/nutch-crawler/nutch/runtime/local
    [mkdir] Created dir: /home/panaarva/Desktop/nutch-crawler/nutch/runtime/deploy
     [copy] Copying 1 file to /home/panaarva/Desktop/nutch-crawler/nutch/runtime/deploy
     [copy] Copying 2 files to /home/panaarva/Desktop/nutch-crawler/nutch/runtime/deploy/bin
     [copy] Copying 1 file to /home/panaarva/Desktop/nutch-crawler/nutch/runtime/local/lib
     [copy] Copying 1 file to /home/panaarva/Desktop/nutch-crawler/nutch/runtime/local/lib/native
     [copy] Copying 36 files to /home/panaarva/Desktop/nutch-crawler/nutch/runtime/local/conf
     [copy] Copying 2 files to /home/panaarva/Desktop/nutch-crawler/nutch/runtime/local/bin
     [copy] Copying 165 files to /home/panaarva/Desktop/nutch-crawler/nutch/runtime/local/lib
     [copy] Copying 655 files to /home/panaarva/Desktop/nutch-crawler/nutch/runtime/local/plugins
     [copy] Copied 3 empty directories to 3 empty directories under /home/panaarva/Desktop/nutch-crawler/nutch/runtime/local/test

BUILD SUCCESSFUL
Total time: 19 minutes 2 seconds
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch$
```

Επαλήθευση στο ότι το build έχει γίνει σωστά:

```
runtime/local/bin/nutch
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch$ runtime/local/bin/nutch
nutch 1.19-SNAPSHOT
Usage: nutch COMMAND [-Dproperty=value]... [command-specific args]...
where COMMAND is one of:
  readdb            read / dump crawl db
  mergedb           merge crawldb-s, with optional filtering
  readlinkdb        read / dump link db
  inject            inject new urls into the database
  generate          generate new segments to fetch from crawl db
  freegen           generate new segments to fetch from text files
  fetch             fetch a segment's pages
  parse             parse a segment's pages
  readseg           read / dump segment data
  mergesegs         merge several segments, with optional filtering and slicing
  updatedb          update crawl db from segments after fetching
```

Αλλαγή εκ νέου του αρχείου conf/nutch-site.xml

```
vi conf/nutch-site.xml
```

```
<property>
<name>http.agent.name</name>
<value>SICrawler</value>
<description>HTTP 'User-Agent' request header. MUST NOT be empty -
please set this to a single word uniquely related to your organization.
NOTE: You should also check other related properties:
http.robots.agents
http.agent.description
http.agent.url
http.agent.email
http.agent.version
and set their values appropriately.
</description>
</property>
<property>
<name>plugin.includes</name>
<value>protocol-http|urlfilter-regex|parse-(html|tika)|index-(basic|anchor)
|urlnormalizer-(pass|regex|basic)|scoring-opic|indexer-elastic</value>
</property>
<property>
<name>db.ignore.external.links</name>
<value>false</value>
<description>If true, outlinks leading from a page to external hosts or
domain
will be ignored. This is an effective way to limit the crawl to include
only initially injected hosts or domains, without creating complex
URLFilters.
```

```
See 'db.ignore.external.links.mode'.
</description>
</property>
<property>
<name>elastic.host</name>
<value>localhost</value>
<description>The hostname to send documents to using TransportClient.
Either host and port must be defined or cluster.
</description>
</property>
<property>
<name>elastic.port</name>
<value>9300</value>
<description>
The port to connect to using TransportClient.
</description>
</property>
<property>
<name>elastic.cluster</name>
<value>elasticsearch</value>
<description>The cluster name to discover. Either host and port must
be defined.
</description>
</property>
<property>
<name>elastic.index</name>
<value>nutch</value>
<description>
The name of the elasticsearch index. Will normally be autocreated if it
doesn't exist.
</description>
</property>
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
   <name>plugin.folders</name>
   <value>/home/panaarva/Desktop/nutch-crawler/nutch/build/plugins</value>
 </property>
<property>
<name>http.agent.name</name>
<value>SICrawler</value>
<description>HTTP 'User-Agent' request header. MUST NOT be empty -
please set this to a single word uniquely related to your organization.
NOTE: You should also check other related properties:
http.robots.agents
http.agent.description
http.agent.url
http.agent.email
http.agent.version
and set their values appropriately.
</description>
```

Δημιουργία φακέλου urls

Δημιουργείται φάκελος με την ονομασία urls ο οποίος θα περιέχει αρχεία που θα δηλώνουν τα url από τα οποία θέλουμε να αντλήσουμε πληροφορία.

```
mkdir runtime/local/urls
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch$ ls runtime/local/
bin  conf  lib  plugins  test  urls
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch$
```

Εγκατάσταση ElasticSearch

```
wget -c
https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.4.2-da
rwin-x86_64.tar.gz -O - | tar -xz
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch$ wget -c https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.4.2-dar
win-x86_64.tar.gz -O - | tar -xz
--2022-05-22 01:59:02--  https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.4.2-darwin-x86_64.tar.gz
Resolving artifacts.elastic.co (artifacts.elastic.co)... 34.120.127.130, 2600:1901:0:1d7::
Connecting to artifacts.elastic.co (artifacts.elastic.co)|34.120.127.130|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 279956874 (267M) [application/x-gzip]
Saving to: 'STDOUT'

-                           100%[=====================================================>] 266.99M  2.81MB/s    in 55s

2022-05-22 01:59:57 (4.87 MB/s) - written to stdout [279956874/279956874]

panaarva@ubuntu:~/Desktop/nutch-crawler/nutch$
```

Αλλάζουμε το config/elasticsearch.yml

```
vi config/elasticsearch.yml
```

```
network.host: localhost
xpack.ml.enabled: false
```

```
#path.logs: /path/to/logs
#
# ---------------------------------- Memory -----------------------------------
#
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network -----------------------------------
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: localhost
xpack.ml.enabled: false
#
# Set a custom port for HTTP:
#
```

Με την παρακάτω εντολή εκκινείται το elasticSearch

```
bin/elasticsearch
```

```
[2022-05-22T02:07:22,118][WARN ][o.e.b.BootstrapChecks    ] [ubuntu] the default discovery settings are unsuitable for production u
se; at least one of [discovery.seed_hosts, discovery.seed_providers, cluster.initial_master_nodes] must be configured
[2022-05-22T02:07:22,136][INFO ][o.e.c.c.ClusterBootstrapService] [ubuntu] no discovery configuration found, will perform best-effo
rt cluster bootstrapping after [3s] unless existing master is discovered
[2022-05-22T02:07:25,146][INFO ][o.e.c.c.Coordinator      ] [ubuntu] setting initial configuration to VotingConfiguration{EcuEIa6jQ
pGTzjU-MOycaw}
[2022-05-22T02:07:25,251][INFO ][o.e.c.s.MasterService    ] [ubuntu] elected-as-master ([1] nodes joined)[{ubuntu}{EcuEIa6jQpGTzjU-
MOycaw}{PSSZxSSuRS6798tY_q3dEQ}{localhost}{127.0.0.1:9300}{dim}{xpack.installed=true} elect leader, _BECOME_MASTER_TASK_, _FINISH_E
LECTION_], term: 1, version: 1, reason: master node changed {previous [], current [{ubuntu}{EcuEIa6jQpGTzjU-MOycaw}{PSSZxSSuRS6798t
Y_q3dEQ}{localhost}{127.0.0.1:9300}{dim}{xpack.installed=true}]}
[2022-05-22T02:07:25,306][INFO ][o.e.c.c.CoordinationState] [ubuntu] cluster UUID set to [MzwuSt9FS-CYl4W2GWtXUw]
[2022-05-22T02:07:25,332][INFO ][o.e.c.s.ClusterApplierService] [ubuntu] master node changed {previous [], current [{ubuntu}{EcuEIa
6jQpGTzjU-MOycaw}{PSSZxSSuRS6798tY_q3dEQ}{localhost}{127.0.0.1:9300}{dim}{xpack.installed=true}]}, term: 1, version: 1, reason: Pub
lication{term=1, version=1}
[2022-05-22T02:07:25,411][INFO ][o.e.h.AbstractHttpServerTransport] [ubuntu] publish_address {localhost/127.0.0.1:9200}, bound_addr
esses {127.0.0.1:9200}
[2022-05-22T02:07:25,412][INFO ][o.e.n.Node               ] [ubuntu] started
[2022-05-22T02:07:25,521][INFO ][o.e.g.GatewayService     ] [ubuntu] recovered [0] indices into cluster_state
[2022-05-22T02:07:25,709][INFO ][o.e.c.m.MetaDataIndexTemplateService] [ubuntu] adding template [.watches] for index patterns [.wat
ches*]
[2022-05-22T02:07:25,763][INFO ][o.e.c.m.MetaDataIndexTemplateService] [ubuntu] adding template [.triggered_watches] for index patt
erns [.triggered_watches*]
[2022-05-22T02:07:25,837][INFO ][o.e.c.m.MetaDataIndexTemplateService] [ubuntu] adding template [.watch-history-10] for index patte
rns [.watcher-history-10*]
```

Εγκατάσταση του Curl

```
sudo apt install curl
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/elasticsearch-7.4.2$ sudo apt install curl
[sudo] password for panaarva:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Ελέγχεται η λειτουργία του ElasticSearch

```
curl http://localhost:9200
```

```
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
panaarva@ubuntu:~/Desktop/nutch-crawler/elasticsearch-7.4.2$ curl http://localhost:9200
{
  "name" : "ubuntu",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "MzwuSt9FS-CYl4W2GWtXUw",
  "version" : {
    "number" : "7.4.2",
    "build_flavor" : "default",
    "build_type" : "tar",
    "build_hash" : "2f90bbf7b93631e52bafb59b3b049cb44ec25e96",
    "build_date" : "2019-10-28T20:40:44.881551Z",
    "build_snapshot" : false,
    "lucene_version" : "8.2.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Εγκατάσταση του Kibana

```
wget -c https://artifacts.elastic.co/downloads/kibana/kibana-7.4.2-linux-x86_64.tar.gz -O - | tar -xz
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler$ wget -c https://artifacts.elastic.co/downloads/kibana/kibana-7.4.2-linux-x86_64.tar.gz -O
- | tar -xz
--2022-05-22 02:13:34--  https://artifacts.elastic.co/downloads/kibana/kibana-7.4.2-linux-x86_64.tar.gz
Resolving artifacts.elastic.co (artifacts.elastic.co)... 34.120.127.130, 2600:1901:0:1d7::
Connecting to artifacts.elastic.co (artifacts.elastic.co)|34.120.127.130|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 252554263 (241M) [application/x-gzip]
Saving to: 'STDOUT'

-                           100%[=====================================================>] 240.85M  8.63MB/s    in 40s

2022-05-22 02:14:15 (5.98 MB/s) - written to stdout [252554263/252554263]

panaarva@ubuntu:~/Desktop/nutch-crawler$
```

```
cd kibana-7.4.2-linux-x86_64/
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/kibana-7.4.2-linux-x86_64$ ls
bin           config  LICENSE.txt  node_modules  optimize      plugins     src           x-pack
built_assets  data    node         NOTICE.txt    package.json  README.txt  webpackShims
```

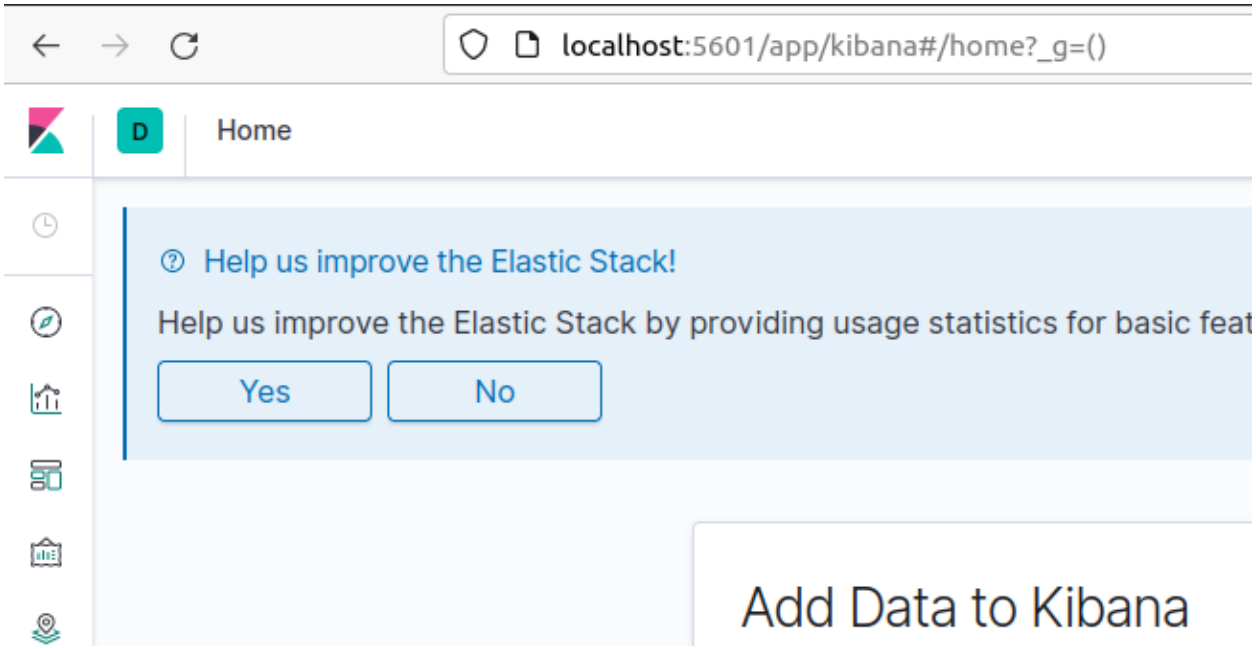
Εκκίνηση του Kibana

```
bin/kibana
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/kibana-7.4.2-linux-x86_64$ bin/kibana
  log   [09:16:33.752] [info][plugins-system] Setting up [4] plugins: [security,translations,inspector,data]
  log   [09:16:33.765] [info][plugins][security] Setting up plugin
  log   [09:16:33.767] [warning][config][plugins][security] Generating a random key for xpack.security.encryptionKey. To prevent se
ssions from being invalidated on restart, please set xpack.security.encryptionKey in kibana.yml
  log   [09:16:33.768] [warning][config][plugins][security] Session cookies will be transmitted over insecure connections. This is
not recommended.
  log   [09:16:33.875] [info][plugins][translations] Setting up plugin
  log   [09:16:33.877] [info][data][plugins] Setting up plugin
  log   [09:16:33.883] [info][plugins-system] Starting [3] plugins: [security,translations,data]
```

Ελέγχουμε την λειτουργικότητα του Kibana

Από τον Browser θα πρέπει να μεταβείτε στο http://localhost:5601/

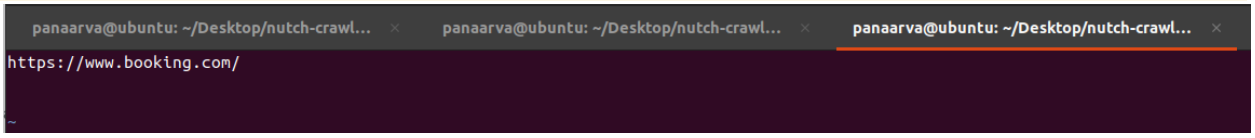


Αλλαγές στο Project του Nutch

Προστίθενται τα url που θέλουμε να αναλύσουμε την πληροφορία στο αρχείο seed.txt αποθηκεύοντας το στο μονοπάτι /runtime/local/urls

```
vi seed.txt
```

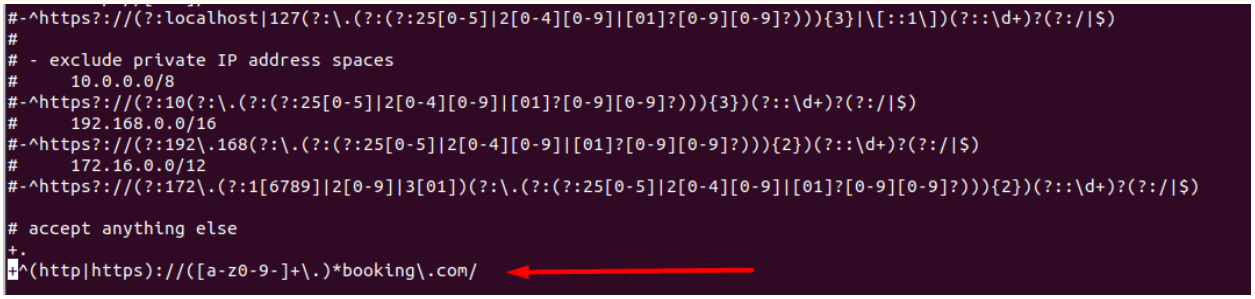
```
https://www.booking.com/
```



Στην συνέχεια γίνεται επεξεργασία στο αρχείο regex-urlfilter.txt που βρίσκεται στο μονοπάτι runtime/local/conf προστίθοντας την παρακάτω γραμμή:

```
vi regex-urlfilter.txt
```

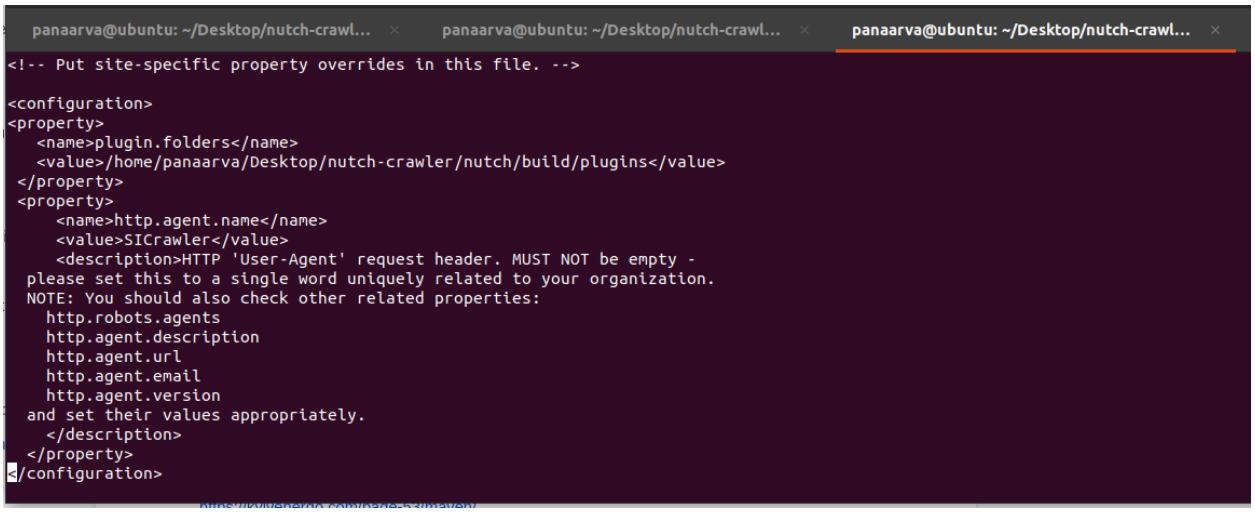
```
+^(http|https)://([a-z0-9-]+\.)*booking\.com/
```



Αλλάζετε το αρχειό nutch-site.xml που βρίσκεται στο /runtime/local/conf

```
vi runtime/local/conf/nutch-site.xml
```

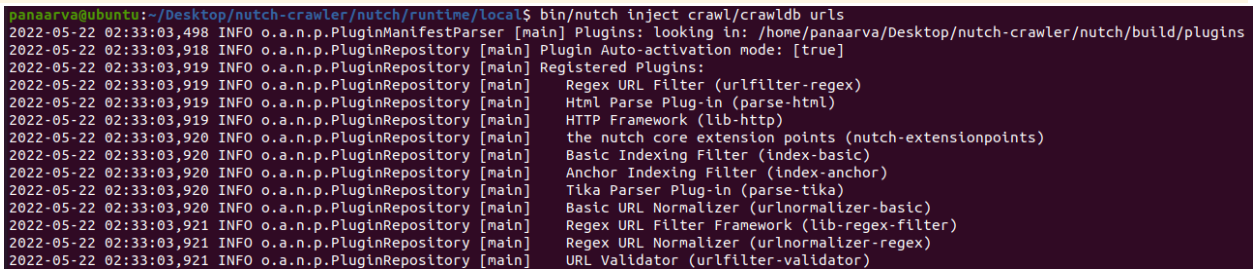
```
<property>
      <name>http.agent.name</name>
      <value>SICrawler</value>
      <description>HTTP 'User-Agent' request header. MUST NOT be empty -
  please set this to a single word uniquely related to your organization.
  NOTE: You should also check other related properties:
      http.robots.agents
      http.agent.description
      http.agent.url
      http.agent.email
      http.agent.version
  and set their values appropriately.
      </description>
  </property>
```



Διαβάζετε το seed.txt προκειμένου να δημιουργηθεί ο φάκελος crawldb

```
cd runtime/local
```

```
bin/nutch inject crawl/crawldb urls
```



Μετά λαμβάνονται όλα τα url από τον φάκελο crawldb για να δημιουργηθεί ένα segment για να ξεκινήσει η διαδικασία

```
bin/nutch generate crawl/crawldb crawl/segments
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch generate crawl/crawldb crawl/segments
2022-05-22 02:34:47,031 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 02:34:47,357 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 02:34:47,358 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 02:34:47,358 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 02:34:47,359 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 02:34:47,359 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 02:34:47,359 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
```

Προστίθεται μια μεταβλητη, η τιμή της οποίας θα είναι το id του segment. Αυτό γίνεται προκειμένου να μπορεί να χρησιμοποιείται εύκολα στις παρακάτω εντολές.

```
segment=`ls -d crawl/segments/2* | tail -1`
```

```
echo $segment
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ segment=`ls -d crawl/segments/2* | tail -1`
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ echo $segment
crawl/segments/20220522023451
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$
```

Αντλούμε όλο το περιεχόμενο από το url που προστέθηκε στο αρχείο seed.txt του φακέλου urls που προστέθηκε παραπάνω.

```
bin/nutch fetch $segment
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch fetch $segment
2022-05-22 02:37:18,348 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 02:37:18,780 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 02:37:18,781 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 02:37:18,781 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 02:37:18,782 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 02:37:18,782 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 02:37:18,785 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
2022-05-22 02:37:18,786 INFO o.a.n.p.PluginRepository [main]    Basic Indexing Filter (index-basic)
2022-05-22 02:37:18,786 INFO o.a.n.p.PluginRepository [main]    Anchor Indexing Filter (index-anchor)
2022-05-22 02:37:18,786 INFO o.a.n.p.PluginRepository [main]    Tika Parser Plug-in (parse-tika)
2022-05-22 02:37:18,786 INFO o.a.n.p.PluginRepository [main]    Basic URL Normalizer (urlnormalizer-basic)
2022-05-22 02:37:18,787 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter Framework (lib-regex-filter)
```

Μετατρέπεται σε μορφή JSON το περιεχόμενο που αντλήθηκε παραπάνω

```
bin/nutch parse $segment
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch parse $segment
2022-05-22 02:38:02,718 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 02:38:03,065 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 02:38:03,066 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 02:38:03,066 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 02:38:03,067 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 02:38:03,067 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 02:38:03,067 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
2022-05-22 02:38:03,067 INFO o.a.n.p.PluginRepository [main]    Basic Indexing Filter (index-basic)
2022-05-22 02:38:03,068 INFO o.a.n.p.PluginRepository [main]    Anchor Indexing Filter (index-anchor)
2022-05-22 02:38:03,068 INFO o.a.n.p.PluginRepository [main]    Tika Parser Plug-in (parse-tika)
2022-05-22 02:38:03,068 INFO o.a.n.p.PluginRepository [main]    Basic URL Normalizer (urlnormalizer-basic)
2022-05-22 02:38:03,068 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter Framework (lib-regex-filter)
2022-05-22 02:38:03,068 INFO o.a.n.p.PluginRepository [main]    Regex URL Normalizer (urlnormalizer-regex)
2022-05-22 02:38:03,068 INFO o.a.n.p.PluginRepository [main]    URL Validator (urlfilter-validator)
2022-05-22 02:38:03,069 INFO o.a.n.p.PluginRepository [main]    CyberNeko HTML Parser (lib-nekohtml)
2022-05-22 02:38:03,069 INFO o.a.n.p.PluginRepository [main]    OPIC Scoring Plug-in (scoring-opic)
2022-05-22 02:38:03,069 INFO o.a.n.p.PluginRepository [main]    Pass-through URL Normalizer (urlnormalizer-pass)
2022-05-22 02:38:03,069 INFO o.a.n.p.PluginRepository [main]    Http Protocol Plug-in (protocol-http)
2022-05-22 02:38:03,069 INFO o.a.n.p.PluginRepository [main]    SolrIndexWriter (indexer-solr)
2022-05-22 02:38:03,070 INFO o.a.n.p.PluginRepository [main] Registered Extension-Points:
2022-05-22 02:38:03,070 INFO o.a.n.p.PluginRepository [main]     (Nutch Content Parser)
2022-05-22 02:38:03,070 INFO o.a.n.p.PluginRepository [main]     (Nutch URL Filter)
2022-05-22 02:38:03,070 INFO o.a.n.p.PluginRepository [main]     (HTML Parse Filter)
```

Ενημερώνεται ο clawldb φάκελος με την παραπάνω μορφοποίηση.

```
bin/nutch updatedb crawl/crawldb $segment
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch updatedb crawl/crawldb $segment
2022-05-22 02:39:10,301 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 02:39:10,718 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 02:39:10,719 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 02:39:10,719 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 02:39:10,720 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 02:39:10,720 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 02:39:10,720 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
2022-05-22 02:39:10,720 INFO o.a.n.p.PluginRepository [main]    Basic Indexing Filter (index-basic)
```

Επαναλαμβάνεται όλη η διαδικασία για τα πρώτα 1000 url που εντοπίστιστηκαν στο url που έχει δηλωθεί.
Δημιουργείται νέο segment και προστιθεται σε νέα μεταβλητή

```
bin/nutch generate crawl/crawldb crawl/segments -topN 1000
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch generate crawl/crawldb crawl/segments -topN 1000
2022-05-22 02:40:06,156 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 02:40:06,927 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 02:40:06,932 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 02:40:06,933 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 02:40:06,934 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 02:40:06,934 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 02:40:06,935 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
2022-05-22 02:40:06,935 INFO o.a.n.p.PluginRepository [main]    Basic Indexing Filter (index-basic)
2022-05-22 02:40:06,935 INFO o.a.n.p.PluginRepository [main]    Anchor Indexing Filter (index-anchor)
2022-05-22 02:40:06,936 INFO o.a.n.p.PluginRepository [main]    Tika Parser Plug-in (parse-tika)
2022-05-22 02:40:06,937 INFO o.a.n.p.PluginRepository [main]    Basic URL Normalizer (urlnormalizer-basic)
```

```
segment2=`ls -d crawl/segments/2* | tail -1`
echo $segment2
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ segment2=`ls -d crawl/segments/2* | tail -1`
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ echo $segment2
crawl/segments/20220522024013
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$
```

```
bin/nutch fetch $segment2
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch fetch $segment2
2022-05-22 02:41:55,392 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 02:41:56,412 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 02:41:56,415 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 02:41:56,421 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 02:41:56,422 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 02:41:56,423 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 02:41:56,435 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
```

```
2022-05-22 02:42:01,939 INFO o.a.n.f.FetcherThread [FetcherThread] Denied by robots.txt: https://secure.booking.com/help.html
2022-05-22 02:42:02,156 INFO o.a.n.f.Fetcher [LocalJobRunner Map Task Executor #0] -activeThreads=10, spinWaiting=6, fetchQueues.totalSize=45, fetchQueues.getQueueCount=4
2022-05-22 02:42:02,348 INFO o.a.n.n.u.r.RegexURLNormalizer [FetcherThread] can't find rules for scope 'fetcher', using default
2022-05-22 02:42:03,157 INFO o.a.n.f.Fetcher [LocalJobRunner Map Task Executor #0] -activeThreads=10, spinWaiting=9, fetchQueues.totalSize=45, fetchQueues.getQueueCount=2
2022-05-22 02:42:04,158 INFO o.a.n.f.Fetcher [LocalJobRunner Map Task Executor #0] -activeThreads=10, spinWaiting=10, fetchQueues.totalSize=45, fetchQueues.getQueueCount=2
2022-05-22 02:42:05,159 INFO o.a.n.f.Fetcher [LocalJobRunner Map Task Executor #0] -activeThreads=10, spinWaiting=10, fetchQueues.totalSize=45, fetchQueues.getQueueCount=2
2022-05-22 02:42:06,160 INFO o.a.n.f.Fetcher [LocalJobRunner Map Task Executor #0] -activeThreads=10, spinWaiting=10, fetchQueues.totalSize=45, fetchQueues.getQueueCount=2
2022-05-22 02:42:07,162 INFO o.a.n.f.Fetcher [LocalJobRunner Map Task Executor #0] -activeThreads=10, spinWaiting=10, fetchQueues.totalSize=45, fetchQueues.getQueueCount=2
2022-05-22 02:42:07,517 INFO o.a.n.f.FetcherThread [FetcherThread] FetcherThread 53 fetching https://cf.bstatic.com/ (queue crawl delay=5000ms)
2022-05-22 02:42:07,714 INFO o.a.n.n.u.r.RegexURLNormalizer [FetcherThread] can't find rules for scope 'fetcher', using default
2022-05-22 02:42:08,163 INFO o.a.n.f.Fetcher [LocalJobRunner Map Task Executor #0] -activeThreads=10, spinWaiting=10, fetchQueues.totalSize=44, fetchQueues.getQueueCount=1
2022-05-22 02:42:08,535 INFO o.a.n.f.FetcherThread [FetcherThread] FetcherThread 57 fetching https://www.booking.com/index.ar.html (queue crawl delay=5000ms)
```

```
bin/nutch parse $segment2
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch parse $segment2
2022-05-22 02:47:09,592 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 02:47:10,207 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 02:47:10,210 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 02:47:10,211 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 02:47:10,212 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 02:47:10,212 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 02:47:10,212 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
2022-05-22 02:47:10,213 INFO o.a.n.p.PluginRepository [main]    Basic Indexing Filter (index-basic)
2022-05-22 02:47:10,213 INFO o.a.n.p.PluginRepository [main]    Anchor Indexing Filter (index-anchor)
2022-05-22 02:47:10,213 INFO o.a.n.p.PluginRepository [main]    Tika Parser Plug-in (parse-tika)
```

```
bin/nutch updatedb crawl/crawldb $segment2
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch updatedb crawl/crawldb $segment2
2022-05-22 02:48:07,858 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 02:48:08,322 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 02:48:08,324 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 02:48:08,324 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 02:48:08,324 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 02:48:08,325 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 02:48:08,325 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
```

```
bin/nutch generate crawl/crawldb crawl/segments -topN 1000
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch generate crawl/crawldb crawl/segments -topN 1000
2022-05-22 02:48:52,153 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 02:48:52,729 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 02:48:52,731 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 02:48:52,732 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 02:48:52,732 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 02:48:52,733 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 02:48:52,733 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
```

```
segment3=`ls -d crawl/segments/2* | tail -1`
echo $segment3
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ segment3=`ls -d crawl/segments/2* | tail -1`
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ echo $segment3
crawl/segments/20220522024858
```

```
bin/nutch fetch $segment3
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch fetch $segment3
2022-05-22 02:51:04,729 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 02:51:05,221 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 02:51:05,223 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 02:51:05,223 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 02:51:05,224 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 02:51:05,225 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 02:51:05,225 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
2022-05-22 02:51:05,226 INFO o.a.n.p.PluginRepository [main]    Basic Indexing Filter (index-basic)
2022-05-22 02:51:05,226 INFO o.a.n.p.PluginRepository [main]    Anchor Indexing Filter (index-anchor)
```

```
bin/nutch parse $segment3
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch parse $segment3
2022-05-22 03:00:33,173 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 03:00:33,625 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 03:00:33,627 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 03:00:33,627 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 03:00:33,627 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 03:00:33,628 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 03:00:33,628 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
2022-05-22 03:00:33,628 INFO o.a.n.p.PluginRepository [main]    Basic Indexing Filter (index-basic)
2022-05-22 03:00:33,628 INFO o.a.n.p.PluginRepository [main]    Anchor Indexing Filter (index-anchor)
```

```
bin/nutch updatedb crawl/crawldb $segment3
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch updatedb crawl/crawldb $segment3
2022-05-22 03:01:04,561 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 03:01:05,107 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 03:01:05,109 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 03:01:05,110 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 03:01:05,110 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 03:01:05,111 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 03:01:05,111 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
```

Ενημερώνεται το linkdb

```
bin/nutch invertlinks crawl/linkdb -dir crawl/segments
```

```
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$ bin/nutch invertlinks crawl/linkdb -dir crawl/segments
2022-05-22 03:02:18,396 INFO o.a.n.p.PluginManifestParser [main] Plugins: looking in: /home/panaarva/Desktop/nutch-crawler/nutch/build/plugins
2022-05-22 03:02:18,919 INFO o.a.n.p.PluginRepository [main] Plugin Auto-activation mode: [true]
2022-05-22 03:02:18,921 INFO o.a.n.p.PluginRepository [main] Registered Plugins:
2022-05-22 03:02:18,922 INFO o.a.n.p.PluginRepository [main]    Regex URL Filter (urlfilter-regex)
2022-05-22 03:02:18,922 INFO o.a.n.p.PluginRepository [main]    Html Parse Plug-in (parse-html)
2022-05-22 03:02:18,923 INFO o.a.n.p.PluginRepository [main]    HTTP Framework (lib-http)
2022-05-22 03:02:18,923 INFO o.a.n.p.PluginRepository [main]    the nutch core extension points (nutch-extensionpoints)
```

Τροποποιείται το nutch-site.xml που βρίσκεται στον /runtime/conf προκειμένου τα δεδομένα να αποθηκεύονται στο elasticSearch.

```
vi runtime/conf/nutch-site.xml
```

```
<property>
      <name>plugin.includes</name>

<value>protocol-http|urlfilter-regex|parse-(html|tika)|index-(basic|anchor)
|urlnormalizer-(pass|regex|basic)|scoring-opic|indexer-elastic</value>
  </property>
  <property>
      <name>db.ignore.external.links</name>
      <value>false</value>
      <description>If true, outlinks leading from a page to external hosts
or domain
      will be ignored. This is an effective way to limit the crawl to
include
      only initially injected hosts or domains, without creating complex
URLFilters.
      See 'db.ignore.external.links.mode'.
      </description>
  </property>
  <property>
      <name>elastic.host</name>
      <value>localhost</value>
      <description>The hostname to send documents to using TransportClient.
      Either host and port must be defined or cluster.
      </description>
  </property>
  <property>
      <name>elastic.port</name>
      <value>9300</value>
      <description>
      The port to connect to using TransportClient.
      </description>
  </property>
  <property>
      <name>elastic.cluster</name>
      <value>elasticsearch</value>
      <description>The cluster name to discover. Either host and port must
      be defined.
      </description>
  </property>
  <property>
```

```
      <name>elastic.index</name>
      <value>nutch</value>
      <description>
      The name of the elasticsearch index. Will normally be autocreated if
it
      doesn't exist.
      </description>
  </property>
```

```
<property>
   <name>plugin.includes</name>
   <value>protocol-http|urlfilter-regex|parse-(html|tika)|index-(basic|anchor)|urlnormalizer
</property>
<property>
    <name>db.ignore.external.links</name>
    <value>false</value>
    <description>If true, outlinks leading from a page to external hosts or domain
    will be ignored. This is an effective way to limit the crawl to include
    only initially injected hosts or domains, without creating complex URLFilters.
    See 'db.ignore.external.links.mode'.
    </description>
</property>
<property>
    <name>elastic.host</name>
    <value>localhost</value>
    <description>The hostname to send documents to using TransportClient.
    Either host and port must be defined or cluster.
    </description>
</property>
<property>
    <name>elastic.port</name>
    <value>9300</value>
    <description>
    The port to connect to using TransportClient.
    </description>
</property>
```

Αποθήκευση δεδομένων στο elasticSearch

```
bin/nutch index crawl/crawldb/ -linkdb crawl/linkdb/ $segment -filter
-normalize -deleteGone
```

ElasticIndexWriter:

host	Comma-separated list of hostnames	localhost
port	The port to connect to elastic server.	9200
scheme	The scheme (http or https) to connect to elastic server.	http
index	Default index to send documents to.	nutch
username	Username for auth credentials	elastic
password	Password for auth credentials	
max.bulk.docs	Maximum size of the bulk in number of documents.	250
max.bulk.size	Maximum size of the bulk in bytes.	2500500
exponential.backoff.millis	Initial delay for the BulkProcessor exponential backoff policy.	100
exponential.backoff.retries	Number of times the BulkProcessor exponential backoff policy should retry bulk operations.	10
bulk.close.timeout	Number of seconds allowed for the BulkProcessor to complete its last operation.	600

```
2022-05-22 03:10:08,769 INFO o.a.n.i.a.AnchorIndexingFilter [pool-5-thread-1] Anchor deduplication is: off
2022-05-22 03:10:09,741 INFO o.a.n.i.IndexingJob [main] Indexer: number of documents indexed, deleted, or skipped:
2022-05-22 03:10:09,771 INFO o.a.n.i.IndexingJob [main] Indexer:      1  indexed (add/update)
2022-05-22 03:10:09,774 INFO o.a.n.i.IndexingJob [main] Indexer: finished at 2022-05-22 03:10:09, elapsed: 00:00:07
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$
```

```
bin/nutch index crawl/crawldb/ -linkdb crawl/linkdb/ $segment2 -filter
-normalize -deleteGone
```

ElasticIndexWriter:

host	Comma-separated list of hostnames	localhost
port	The port to connect to elastic server.	9200
scheme	The scheme (http or https) to connect to elastic server.	http
index	Default index to send documents to.	nutch
username	Username for auth credentials	elastic
password	Password for auth credentials	
max.bulk.docs	Maximum size of the bulk in number of documents.	250
max.bulk.size	Maximum size of the bulk in bytes.	2500500
exponential.backoff.millis	Initial delay for the BulkProcessor exponential backoff policy.	100
exponential.backoff.retries	Number of times the BulkProcessor exponential backoff policy should retry bulk operations.	10
bulk.close.timeout	Number of seconds allowed for the BulkProcessor to complete its last operation.	600

```
2022-05-22 03:12:01,301 INFO o.a.n.i.a.AnchorIndexingFilter [pool-5-thread-1] Anchor deduplication is: off
2022-05-22 03:12:02,843 INFO o.a.n.i.IndexingJob [main] Indexer: number of documents indexed, deleted, or skipped:
2022-05-22 03:12:02,849 INFO o.a.n.i.IndexingJob [main] Indexer:      2  deleted (gone)
2022-05-22 03:12:02,849 INFO o.a.n.i.IndexingJob [main] Indexer:      4  deleted (redirects)
2022-05-22 03:12:02,850 INFO o.a.n.i.IndexingJob [main] Indexer:     44  indexed (add/update)
2022-05-22 03:12:02,852 INFO o.a.n.i.IndexingJob [main] Indexer: finished at 2022-05-22 03:12:02, elapsed: 00:00:06
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$
```

```
bin/nutch index crawl/crawldb/ -linkdb crawl/linkdb/ $segment3 -filter
-normalize -deleteGone
```

ElasticIndexWriter:

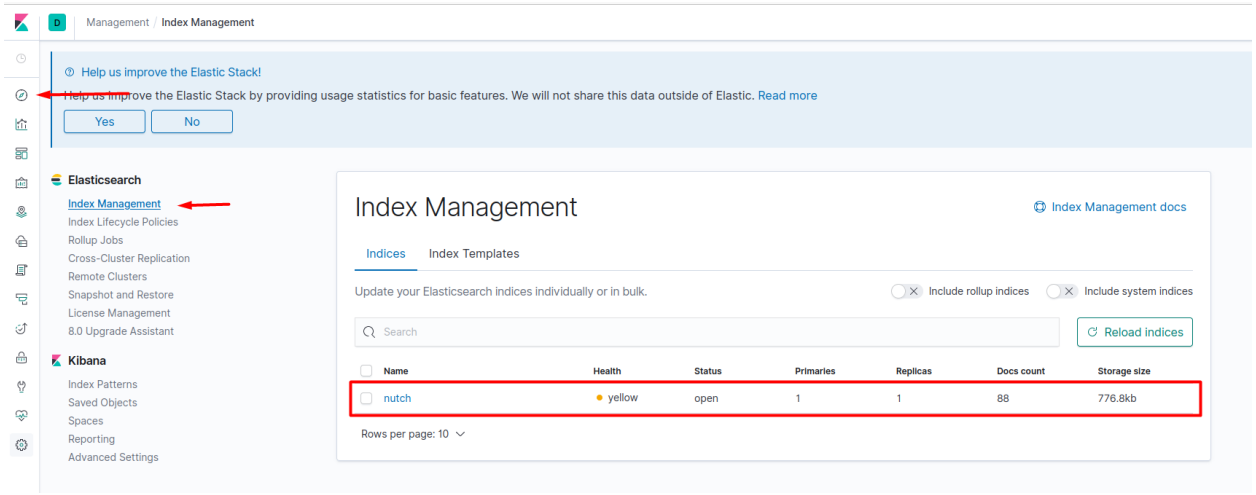
host	Comma-separated list of hostnames	localhost
port	The port to connect to elastic server.	9200
scheme	The scheme (http or https) to connect to elastic server.	http
index	Default index to send documents to.	nutch
username	Username for auth credentials	elastic
password	Password for auth credentials	
max.bulk.docs	Maximum size of the bulk in number of documents.	250
max.bulk.size	Maximum size of the bulk in bytes.	2500500
exponential.backoff.millis	Initial delay for the BulkProcessor exponential backoff policy.	100
exponential.backoff.retries	Number of times the BulkProcessor exponential backoff policy should retry bulk operations.	10
bulk.close.timeout	Number of seconds allowed for the BulkProcessor to complete its last operation.	600

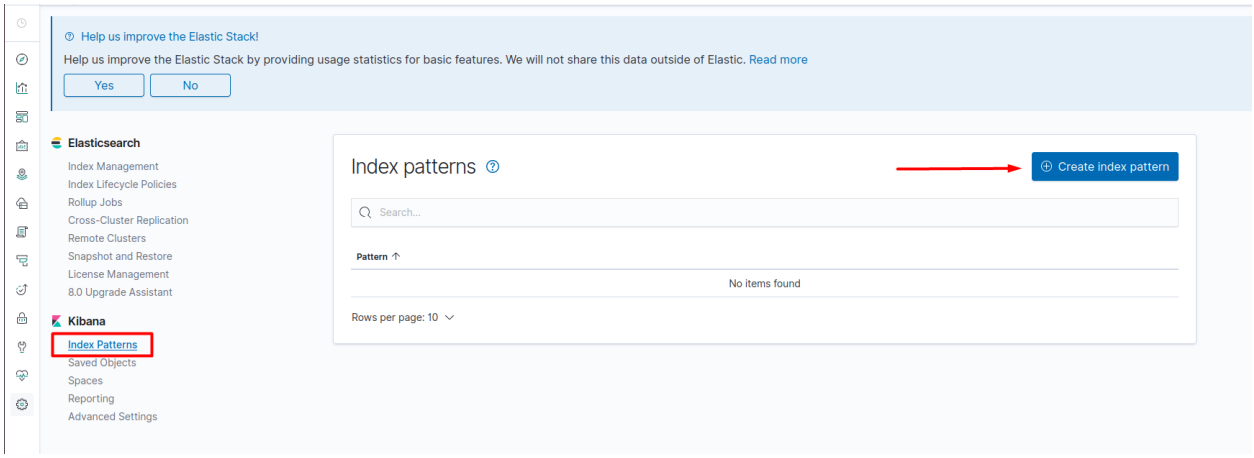
```
2022-05-22 03:12:42,488 INFO o.a.n.i.a.AnchorIndexingFilter [pool-5-thread-1] Anchor deduplication is: off
2022-05-22 03:12:42,577 INFO o.a.n.n.u.r.RegexURLNormalizer [pool-5-thread-1] can't find rules for scope 'indexer', using default
2022-05-22 03:12:43,042 INFO o.a.n.i.IndexingJob [main] Indexer: number of documents indexed, deleted, or skipped:
2022-05-22 03:12:43,055 INFO o.a.n.i.IndexingJob [main] Indexer:     44  deleted (gone)
2022-05-22 03:12:43,056 INFO o.a.n.i.IndexingJob [main] Indexer:     87  deleted (redirects)
2022-05-22 03:12:43,056 INFO o.a.n.i.IndexingJob [main] Indexer:     43  indexed (add/update)
2022-05-22 03:12:43,059 INFO o.a.n.i.IndexingJob [main] Indexer: finished at 2022-05-22 03:12:43, elapsed: 00:00:04
panaarva@ubuntu:~/Desktop/nutch-crawler/nutch/runtime/local$
```

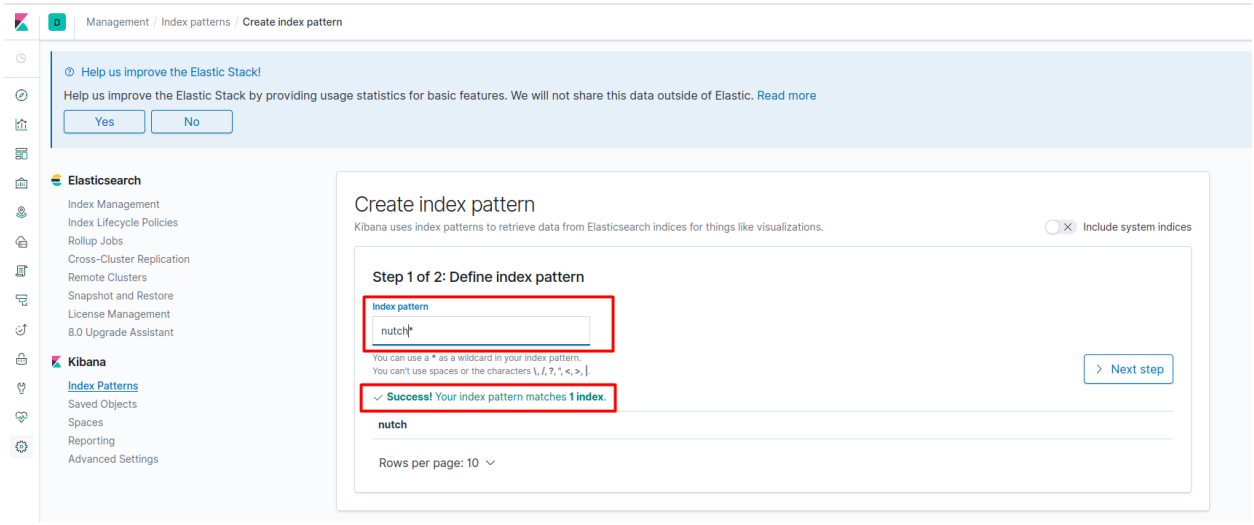
Ελέγχεται αν έχουν αποθηκευτεί σωστά τα δεδομένα πηγαίνοντας από browser στο παρακάτω link:

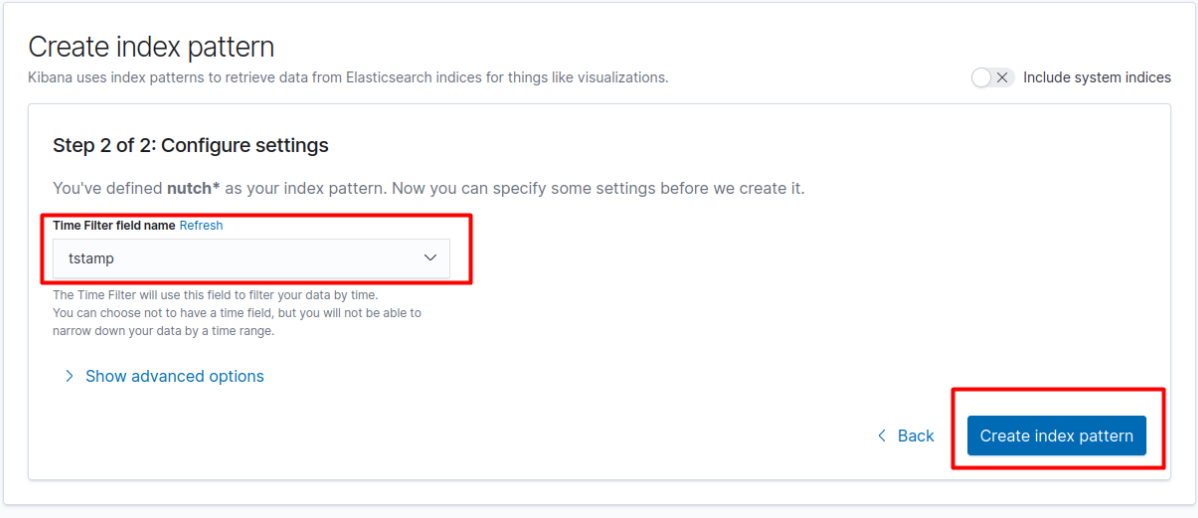
http://localhost:9200/nutch/_search

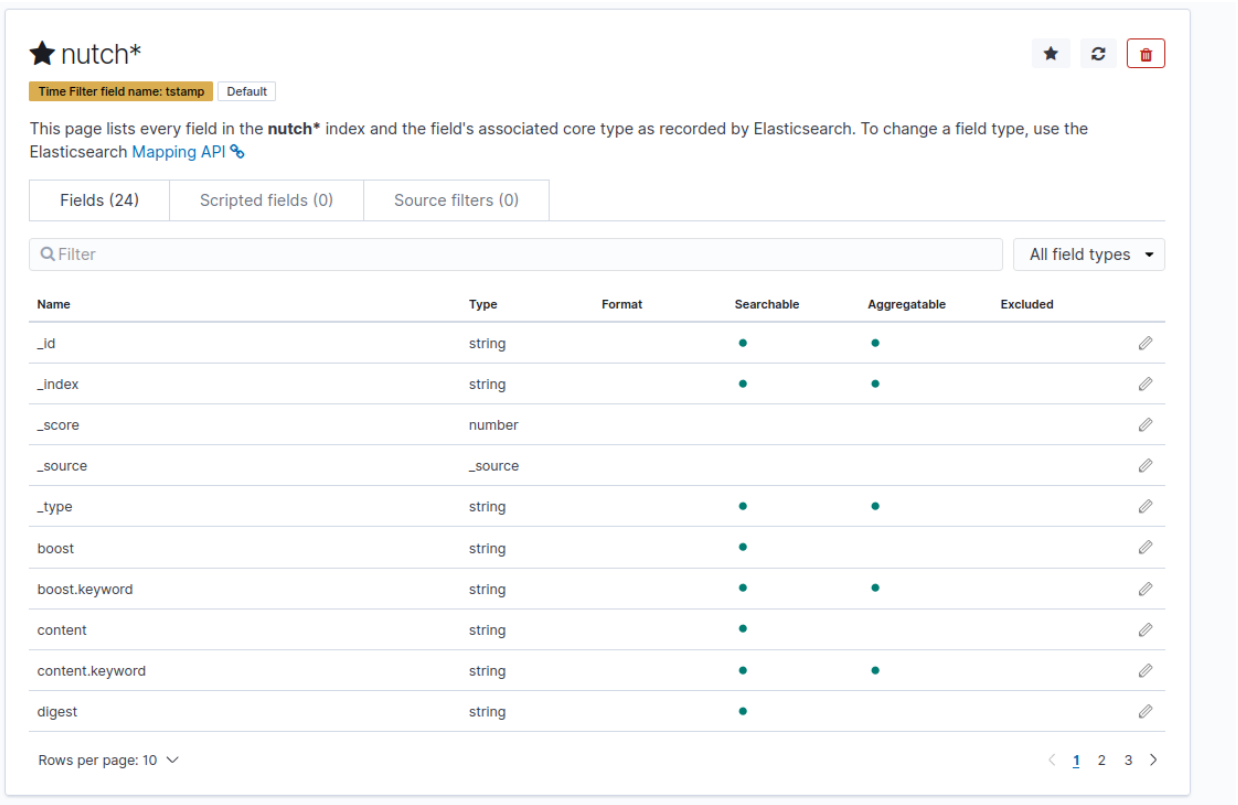
Σύνδεση Kibana με τον index του elasticSearch



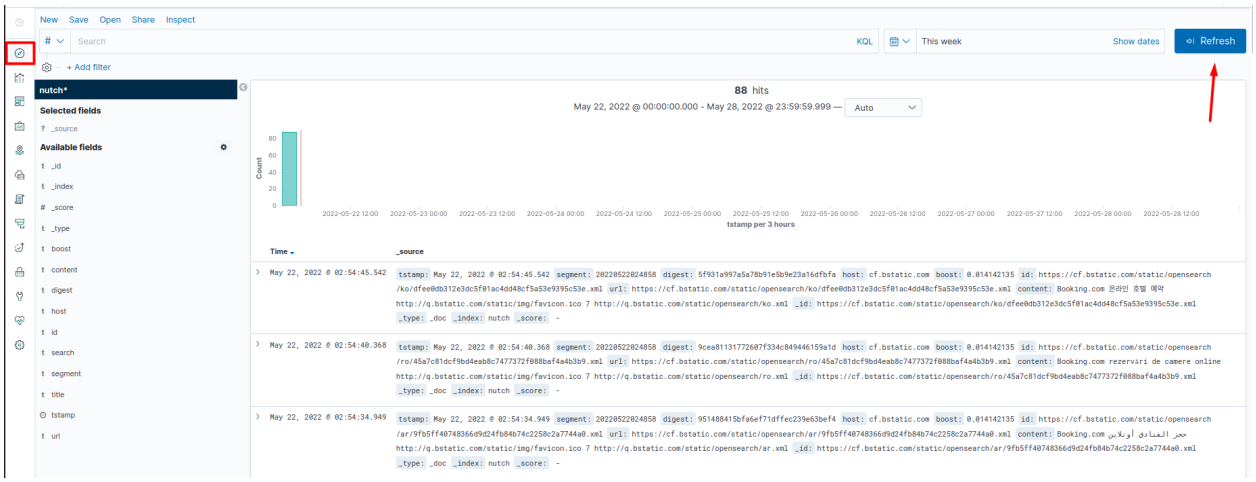

Help us improve the Elastic Stack!
Help us improve the Elastic Stack by providing usage statistics for basic features. We will not share this data outside of Elastic. Read more
Yes
No
Elasticsearch
Index Management
Index Lifecycle Policies
Rollup Jobs
Cross-Cluster Replication
Remote Clusters
Snapshot and Restore
License Management
8.0 Upgrade Assistant
Kibana
Index Patterns
Saved Objects
Spaces
Reporting
Advanced Settings
Index patterns
Create index pattern
Search...
Pattern
No items found
Rows per page: 10


Management / Index patterns / Create index pattern
Help us improve the Elastic Stack!
Help us improve the Elastic Stack by providing usage statistics for basic features. We will not share this data outside of Elastic. Read more
Yes
No
Elasticsearch
Index Management
Index Lifecycle Policies
Rollup Jobs
Cross-Cluster Replication
Remote Clusters
Snapshot and Restore
License Management
8.0 Upgrade Assistant
Kibana
Index Patterns
Saved Objects
Spaces
Reporting
Advanced Settings
Create index pattern
Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.
Include system indices
Step 1 of 2: Define index pattern
Index pattern
nutch*
You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.
Next step
Success! Your index pattern matches 1 index.
nutch
Rows per page: 10


Create index pattern
Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.
Include system indices
Step 2 of 2: Configure settings
You've defined nutch* as your index pattern. Now you can specify some settings before we create it.
Time Filter field name Refresh
tstamp
The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.
Show advanced options
Back
Create index pattern



Τελική οπτικοποίηση των δεδομένων



Παράρτημα 2

Εγκατάσταση Sparkler

Εγκατάσταση net-tools βιβλιοθήκης που περιέχει βασικές εντολές όπως ifconfig κλπ:

```
sudo apt install net-tools
```

```
panaarva@ubuntu:~$ sudo apt install net-tools
[sudo] password for panaarva:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 36 not upgraded.
Need to get 196 kB of archives.
After this operation, 864 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+
git20180626.aebd88e-1ubuntu1 [196 kB]
Fetched 196 kB in 1s (165 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 181482 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ..
.
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...


Progress: [ 80%] [##############################################.....]
```

Εγκατάσταση JAVA 8

```
sudo apt-get update
```

```
panaarva@ubuntu:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
```

```
sudo apt-get install openjdk-8-jdk
```

```
panaarva@ubuntu:~/Desktop$ sudo apt-get install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni
  libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev
  libxt-dev openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless x11proto-core-dev
  x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-8-demo openjdk-8-source
  visualvm icedtea-8-plugin fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni
  libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev
  libxt-dev openjdk-8-jdk openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless
  x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 21 newly installed, 0 to remove and 137 not upgraded.
Need to get 43.5 MB of archives.
After this operation, 162 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
java -version
```

```
panaarva@ubuntu:~$ java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1~20.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
```

Εγκατάσταση Docker

```
sudo apt-get install ca-certificates curl gnupg lsb-release
```

```
panaarva@ubuntu:~/Desktop$ sudo apt-get install ca-certificates curl gnupg lsb-release
Reading package lists... Done
Building dependency tree
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu2).
lsb-release set to manually installed.
ca-certificates is already the newest version (20210119~20.04.2).
ca-certificates set to manually installed.
gnupg is already the newest version (2.2.19-3ubuntu2.1).
gnupg set to manually installed.
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  libcurl4
1 upgraded, 1 newly installed, 0 to remove and 136 not upgraded.
Need to get 161 kB/396 kB of archives.
After this operation, 416 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg
--dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

```
panaarva@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sud
o gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
panaarva@ubuntu:~$
```

```
echo                    "deb       [arch=$(dpkg       --print-architecture)
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list
> /dev/null
```

```
panaarva@ubuntu:~$ echo   "deb [arch=$(dpkg --print-architecture) signed-by=/usr
/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ub
untu \
>   $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null
panaarva@ubuntu:~$
```

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

```
panaarva@ubuntu:~/Desktop$ sudo apt-get install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-el
  git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-scan-plugin git git-man
  liberror-perl pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 136 not upgraded.
Need to get 102 MB of archives.
After this operation, 444 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
sudo docker run hello-world
```

```
panaarva@ubuntu:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:10d7d58d5ebd2a652f4d93fdd86da8f265f5318c6a73cc5b6a9798ff6d2b2e67
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

panaarva@ubuntu:~$
```

Εγκατάσταση Scala

```
wget www.scala-lang.org/files/archive/scala-2.13.0.deb
```

```
panaarva@ubuntu:~$ wget www.scala-lang.org/files/archive/scala-2.13.0.deb
--2022-05-01 07:07:54--  http://www.scala-lang.org/files/archive/scala-2.13.0.de
b
Resolving www.scala-lang.org (www.scala-lang.org)... 128.178.218.78
Connecting to www.scala-lang.org (www.scala-lang.org)|128.178.218.78|:80... conn
ected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.scala-lang.org/files/archive/scala-2.13.0.deb [following]
--2022-05-01 07:07:55--  https://www.scala-lang.org/files/archive/scala-2.13.0.d
eb
Connecting to www.scala-lang.org (www.scala-lang.org)|128.178.218.78|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 609688786 (581M) [application/x-debian-package]
Saving to: ‘scala-2.13.0.deb’

scala-2.13.0.deb    100%[===================>] 581.44M  3.01MB/s    in 4m 56s

2022-05-01 07:12:51 (1.97 MB/s) - ‘scala-2.13.0.deb’ saved [609688786/609688786]
```

```
sudo dpkg -i scala*.deb
```

```
panaarva@ubuntu:~/Desktop$ sudo dpkg -i scala*.deb
Selecting previously unselected package scala.
(Reading database ... 180044 files and directories currently installed.)
Preparing to unpack scala-2.13.0.deb ...
Unpacking scala (2.13.0-400) ...
Setting up scala (2.13.0-400) ...
Creating system group: scala
Creating system user: scala in scala with scala daemon-user and shell /bin/false
Processing triggers for man-db (2.9.1-1) ...
```

```
scala -version
```

```
panaarva@ubuntu:~/Desktop$ scala -version
Scala code runner version 2.13.0 -- Copyright 2002-2019, LAMP/EPFL and Lightbend, Inc.
panaarva@ubuntu:~/Desktop$
```

Εγκατάσταση Sbt

```
sudo apt-get update
sudo apt-get install apt-transport-https curl gnupg -yqq
```

```
panaarva@ubuntu:~/Desktop$ sudo apt-get install apt-transport-https curl gnupg -yqq
Selecting previously unselected package apt-transport-https.
(Reading database ... 183891 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.0.6_all.deb ...
Unpacking apt-transport-https (2.0.6) ...
Setting up apt-transport-https (2.0.6) ...
panaarva@ubuntu:~/Desktop$
```

```
echo "deb https://repo.scala-sbt.org/scalasbt/debian all main" | sudo tee /etc/apt/sources.list.d/sbt.list
```

```
panaarva@ubuntu:~$ echo "deb https://repo.scala-sbt.org/scalasbt/debian all main
" | sudo tee /etc/apt/sources.list.d/sbt.list
deb https://repo.scala-sbt.org/scalasbt/debian all main
panaarva@ubuntu:~$
```

```
echo "deb https://repo.scala-sbt.org/scalasbt/debian /" | sudo tee /etc/apt/sources.list.d/sbt_old.list
```

```
panaarva@ubuntu:~$ echo "deb https://repo.scala-sbt.org/scalasbt/debian /" | sud
o tee /etc/apt/sources.list.d/sbt_old.list
deb https://repo.scala-sbt.org/scalasbt/debian /
panaarva@ubuntu:~$
```

```
curl -sL "https://keyserver.ubuntu.com/pks/lookup?op=get&search=0x2EE0EA64E40A89B84B2DF73499E82A75642AC823" | sudo -H gpg --no-default-keyring --keyring gnupg-ring:/etc/apt/trusted.gpg.d/scalasbt-release.gpg --import
```

```
panaarva@ubuntu:~/Desktop$ curl -sL "https://keyserver.ubuntu.com/pks/lookup?op=get&search=0x2EE0EA64
E40A89B84B2DF73499E82A75642AC823" | sudo -H gpg --no-default-keyring --keyring gnupg-ring:/etc/apt/tr
usted.gpg.d/scalasbt-release.gpg --import
gpg: keyring '/etc/apt/trusted.gpg.d/scalasbt-release.gpg' created
gpg: directory '/root/.gnupg' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 99E82A75642AC823: public key "sbt build tool <scalasbt@gmail.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
panaarva@ubuntu:~/Desktop$
```

```
sudo chmod 644 /etc/apt/trusted.gpg.d/scalasbt-release.gpg
```

```
panaarva@ubuntu:~$ sudo chmod 644 /etc/apt/trusted.gpg.d/scalasbt-release.gpg
panaarva@ubuntu:~$
```

```
sudo apt-get update
sudo apt-get install sbt
```

```
panaarva@ubuntu:~/Desktop$ sudo apt-get install sbt
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sbt
0 upgraded, 1 newly installed, 0 to remove and 136 not upgraded.
Need to get 19.9 kB of archives.
After this operation, 49.2 kB of additional disk space will be used.
Get:1 https://scala.jfrog.io/artifactory/debian all/main amd64 sbt all 1.6.2 [19.9 kB]
Fetched 19.9 kB in 12s (1,689 B/s)
Selecting previously unselected package sbt.
(Reading database ... 183895 files and directories currently installed.)
Preparing to unpack .../apt/archives/sbt_1.6.2_all.deb ...
Unpacking sbt (1.6.2) ...
Setting up sbt (1.6.2) ...
Creating system group: sbt
Creating system user: sbt in sbt with sbt daemon-user and shell /bin/false
Processing triggers for man-db (2.9.1-1) ...
panaarva@ubuntu:~/Desktop$
```

Εγκατάσταση Maven

```
sudo apt install maven
```

```
panaarva@ubuntu:~/Desktop$ sudo apt install maven
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaopalliance-java libapache-pom-java libatinject-jsr330-api-java libcdi-api-java
  libcommons-cli-java libcommons-io-java libcommons-lang3-java libcommons-parent-java
  libgeronimo-annotation-1.3-spec-java libgeronimo-interceptor-3.0-spec-java libguava-java
  libguice-java libhawtjni-runtime-java libjansi-java libjansi-native-java libjsr305-java
  libmaven-parent-java libmaven-resolver-java libmaven-shared-utils-java libmaven3-core-java
  libplexus-cipher-java libplexus-classworlds-java libplexus-component-annotations-java
  libplexus-interpolation-java libplexus-sec-dispatcher-java libplexus-utils2-java
  libsisu-inject-java libsisu-plexus-java libslf4j-java libwagon-file-java
  libwagon-http-shaded-java libwagon-provider-api-java
```

```
Setting up libwagon-file-java (3.3.4-1) ...
Setting up libcommons-io-java (2.6-2ubuntu0.20.04.1) ...
Setting up libguice-java (4.2.1-1) ...
Setting up libjansi-java (1.18-1) ...
Setting up libmaven-shared-utils-java (3.3.0-1) ...
Setting up libsisu-inject-java (0.3.3-1) ...
Setting up libsisu-plexus-java (0.3.3-3) ...
Setting up libmaven3-core-java (3.6.3-1) ...
Setting up maven (3.6.3-1) ...
update-alternatives: using /usr/share/maven/bin/mvn to provide /usr/bin/mvn (mvn) in auto mode
```

Εγκατάσταση και επεξεργασία του Repository

```
sudo apt install git
```

```
panaarva@ubuntu:~/Desktop$ sudo apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.25.1-1ubuntu3.4).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 136 not upgraded.
panaarva@ubuntu:~/Desktop$
```

```
cd Desktop
git clone https://github.com/USCDataScience/sparkler
```

```
panaarva@ubuntu:~/Desktop$ git clone https://github.com/USCDataScience/sparkler
Cloning into 'sparkler'...
remote: Enumerating objects: 13840, done.
remote: Counting objects: 100% (4866/4866), done.
remote: Compressing objects: 100% (1279/1279), done.
remote: Total 13840 (delta 2109), reused 4778 (delta 2063), pack-reused 8974
Receiving objects: 100% (13840/13840), 22.45 MiB | 1.31 MiB/s, done.
Resolving deltas: 100% (5509/5509), done.
panaarva@ubuntu:~/Desktop$
```

```
cd sparkler/
sbt package
```

```
panaarva@ubuntu:~/Desktop/sparkler$ sbt package
[info] welcome to sbt 1.5.0 (Private Build Java 1.8.0_312)
[info] loading settings for project sparkler-build-build from metals.sbt ...
[info] loading project definition from /home/panaarva/Desktop/sparkler/project/p
roject

  | => sparkler-build-build / dependencyPositions 0s
  | => sparkler-build-build / otherResolvers 0s
```

Αφαιρούμε από το αρχείο ‘sparkler/release.sh’ την λέξη “test”

```
sudo nano release.sh
```

```
  GNU nano 4.8                                  release.sh
#!/bin/bash
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Script      : release.sh
# Usage       : ./release.sh
# Description: Release Sparkler Silently - Create tag with version in version.sbt and bump it

sbt clean package test && sbt releaseSilent
```

Σχολιάζεται ή αφαιρείται στο παρακάτω αρχείο η γραμμή 56 προκειμένου να μην εκτελεστεί:

```
sudo nano ./sparkler-app/src/main/scala/edu/usc/irds/sparkler/storage/solr/StatusUpdateSolrTransformer.scala
```

```
panaarva@ubuntu:~/Desktop/sparkler$ nano ./sparkler-app/src/main/scala/edu/usc/irds/sparkler/storage/solr/StatusUpdateSolrTransformer.scala
```

```
    val hashFunction: HashFunction = Hashing.sha256()
    var toUpdate : Map[String, Object] = Map(
      Constants.storage.ID -> data.fetchedData.getResource.getId,
      Constants.storage.STATUS -> Map("set" -> data.fetchedData.getResource.get>
      Constants.storage.FETCH_TIMESTAMP -> Map("set" -> data.fetchedData.getFet>
      Constants.storage.LAST_UPDATED_AT -> Map("set" -> new Date()).asJava,
      Constants.storage.RETRIES_SINCE_FETCH -> Map("inc" -> 1).asJava,
      Constants.storage.EXTRACTED_TEXT -> data.parsedData.extractedText,
      Constants.storage.CONTENT_TYPE -> data.fetchedData.getContentType.split(">
      Constants.storage.FETCH_STATUS_CODE -> data.fetchedData.getResponseCode.t>
      Constants.storage.SIGNATURE -> hashFunction.hashBytes(data.fetchedData.ge>
      Constants.storage.RELATIVE_PATH -> URLUtil.reverseUrl(data.fetchedData.ge>
      Constants.storage.OUTLINKS -> data.parsedData.outlinks.toArray,
      Constants.storage.SEGMENT -> data.fetchedData.getSegment,
      //Constants.storage.CONTENTHASH -> ContentHash.fetchHash(data.fetchedData>
    )
```

Προσθέτουμε στο αρχείο Crawler.scala στην γραμμή 171 την εντολή conf.set("spark.io.compression.codec", "snappy")

```
sudo nano ./sparkler-app/src/main/scala/edu/usc/irds/sparkler/pipeline/Crawler.scala
```

```
def init(): Unit = {
  jobId = if(!jobIdFile.isEmpty){
    Source.fromFile(jobIdFile).getLines.mkString
  } else{
    jobId
  }
  setConfig()
  if (this.outputPath.isEmpty) {
    this.outputPath = jobId
  }
  val conf = new SparkConf().setAppName(jobId)
  conf.set("spark.io.compression.codec", "snappy")
  if (sparkMaster != null && sparkMaster.nonEmpty) {
    conf.setMaster(sparkMaster)
  }
```

Δηλώνουμε το λογαριασμό μας στο git προκειμένου να πραγματοποιήσουμε ένα commit

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
git commit -a -m "removed test"
```

```
panaarva@ubuntu:~/Desktop/sparkler$ git commit -a -m "removed test"
[main 9bb6a4b] removed test
 3 files changed, 3 insertions(+), 2 deletions(-)
 mode change 100644 => 100755 release.sh
```

Δίνουμε τα κατάλληλα δικαιώματα στο αρχείο release.sh έτσι ώστε να μπορούμε να το τρέξουμε

```
chmod 754 release.sh
```

```
panaarva@ubuntu:~/Desktop/sparkler$ chmod 754 release.sh
panaarva@ubuntu:~/Desktop/sparkler$
```

Τρέχουμε το αρχείο releash.sh

```
./release.sh
```

```
panaarva@ubuntu:~/Desktop/sparkler$ ./release.sh
[info] welcome to sbt 1.5.0 (Private Build Java 1.8.0_312)
[info] loading settings for project sparkler-build-build from metals.sbt ...
[info] loading project definition from /home/panaarva/Desktop/sparkler/project/p
roject

  | => sparkler-build-build / dependencyPositions 0s
  | => sparkler-build-build / otherResolvers 0s
```

Ακόμα και η παραπάνω εκτέλεση εμφανίσει κάποιο σφάλμα, αυτό που θα πρέπει να κοιτάξουμε αν δημιουργήθηκε το SNAPSHOT εκτελώντας την παρακάτω εντολή:

```
ls
./build/sparkler-app-0.5.26-SNAPSHOT/lib/com.kythera.sparkler-app-0.5.26-SN
```

```
APSHOT.jar
```

```
panaarva@ubuntu:~/Desktop/sparkler$ ls ./build/sparkler-app-0.5.26-SNAPSHOT/lib/com.kythera.sparkler-app-0.5.26-SNAPSHOT.jar
./build/sparkler-app-0.5.26-SNAPSHOT/lib/com.kythera.sparkler-app-0.5.26-SNAPSHOT.jar
panaarva@ubuntu:~/Desktop/sparkler$
```

Εγκατάσταση Spark

```
cd
wget
https://downloads.apache.org/spark/spark-3.0.3/spark-3.0.3-bin-hadoop2.7.tg
z
```

```
panaarva@ubuntu:~$ wget https://downloads.apache.org/spark/spark-3.0.3/spark-3.
0.3-bin-hadoop2.7.tgz
--2022-05-01 09:22:28--  https://downloads.apache.org/spark/spark-3.0.3/spark-3
.0.3-bin-hadoop2.7.tgz
Resolving downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.99
.95.219, 2a01:4f9:3a:2c57::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|135.181.214.104|:443.
.. connected.
HTTP request sent, awaiting response... 200 OK
Length: 220400553 (210M) [application/x-gzip]
Saving to: ‘spark-3.0.3-bin-hadoop2.7.tgz’

      spark-3.0.3-b    0%[                    ] 248.00K   119KB/s
```

```
tar xvf spark-*
```

```
panaarva@ubuntu:~$ tar xvf spark-*
spark-3.0.3-bin-hadoop2.7/
spark-3.0.3-bin-hadoop2.7/NOTICE
spark-3.0.3-bin-hadoop2.7/kubernetes/
spark-3.0.3-bin-hadoop2.7/kubernetes/tests/
spark-3.0.3-bin-hadoop2.7/kubernetes/tests/worker_memory_check.py
spark-3.0.3-bin-hadoop2.7/kubernetes/tests/py_container_checks.py
spark-3.0.3-bin-hadoop2.7/kubernetes/tests/pyfiles.py
spark-3.0.3-bin-hadoop2.7/kubernetes/dockerfiles/
spark-3.0.3-bin-hadoop2.7/kubernetes/dockerfiles/spark/
spark-3.0.3-bin-hadoop2.7/kubernetes/dockerfiles/spark/entrypoint.sh
spark-3.0.3-bin-hadoop2.7/kubernetes/dockerfiles/spark/bindings/
spark-3.0.3-bin-hadoop2.7/kubernetes/dockerfiles/spark/bindings/R/
```

```
sudo mv spark-3.0.3-bin-hadoop2.7 /opt/spark
```

```
panaarva@ubuntu:~$ sudo mv spark-3.0.3-bin-hadoop2.7 /opt/spark
[sudo] password for panaarva:
panaarva@ubuntu:~$
```

Εκτελούμε τα 3 παρακάτω export:

```
export SPARK_HOME=/opt/spark
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
export PYSPARK_PYTHON=/usr/bin/python3
```

Θα πρέπει να εκτελεστούν τα 2 παρακάτω script:

```
start-master.sh
```

```
panaarva@ubuntu:~$ start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spar
k-panaarva-org.apache.spark.deploy.master.Master-1-ubuntu.out
panaarva@ubuntu:~$
```

```
start-slave.sh spark://localhost:7077
```

```
panaarva@ubuntu:~$ start-slave.sh spark://localhost:7077
starting org.apache.spark.deploy.worker.Worker, logging to /opt/spark/logs/spar
k-panaarva-org.apache.spark.deploy.worker.Worker-1-ubuntu.out
panaarva@ubuntu:~$
```

```
spark-shell
```

```
panaarva@ubuntu:~$ spark-shell
22/05/01 09:32:51 WARN Utils: Your hostname, ubuntu resolves to a loopback addr
ess: 127.0.1.1; using 192.168.230.130 instead (on interface ens33)
22/05/01 09:32:51 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
 address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/o
pt/spark/jars/spark-unsafe_2.12-3.0.3.jar) to constructor java.nio.DirectByteBu
ffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.
unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflec
tive access operations
WARNING: All illegal access operations will be denied in a future release
22/05/01 09:32:51 WARN NativeCodeLoader: Unable to load native-hadoop library f
or your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLev
el(newLevel).
Spark context Web UI available at http://192.168.230.130:4040
Spark context available as 'sc' (master = local[*], app id = local-1651422780845).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.0.3
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.15)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

```
val df = spark.sql("select 1 i, 2 j, 3 k")
```

```
scala> val df = spark.sql("select 1 i, 2 j, 3 k")
df: org.apache.spark.sql.DataFrame = [i: int, j: int ... 1 more field]
```

```
df.show()
```

```
scala> df.show()
+---+---+---+
|  i|  j|  k|
+---+---+---+
|  1|  2|  3|
+---+---+---+
```

Εγκατάσταση ElasticSearch

```
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

```
panaarva@ubuntu:~$ curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
OK
panaarva@ubuntu:~$
```

```
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
```

```
panaarva@ubuntu:~$ echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
panaarva@ubuntu:~$
```

```
sudo apt update
sudo apt install elasticsearch
```

```
panaarva@ubuntu:~$ sudo apt install elasticsearch
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils libjline2-java scala-library scala-parser-combinators scala-xml ubuntu-fan
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 36 not upgraded.
Need to get 312 MB of archives.
After this operation, 518 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 elasticsearch amd64 7.17.3 [312 MB]
0% [1 elasticsearch 1,475 kB/312 MB 0%]
```

Πραγματοποιείται μια αλλαγή στο αρχείο elasticsearch.yml

```
sudo nano /etc/elasticsearch/elasticsearch.yml
network.host: localhost
```

```
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
#network.host: 192.168.0.1
network.host: localhost
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
```

```
sudo systemctl start elasticsearch
sudo systemctl enable elasticsearch
```

```
panaarva@ubuntu:/opt$ sudo systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service → /lib/systemd/system/elasticsearch.service.
panaarva@ubuntu:/opt$
```

```
sudo systemctl status elasticsearch
```

```
panaarva@ubuntu:/opt$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2022-05-02 00:00:07 PDT; 2min 18s ago
       Docs: https://www.elastic.co
   Main PID: 22195 (java)
      Tasks: 73 (limit: 7075)
     Memory: 3.1G
     CGroup: /system.slice/elasticsearch.service
             ├─22195 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=60
             └─22389 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

May 01 23:59:39 ubuntu systemd[1]: Starting Elasticsearch...
May 02 00:00:07 ubuntu systemd[1]: Started Elasticsearch.
```

```
sudo ufw allow from 198.51.100.0 to any port 9200
```

```
panaarva@ubuntu:/opt$ sudo ufw allow from 198.51.100.0 to any port 9200
Rules updated
```

Ενεργοποίηση του firewall

```
sudo ufw enable
```

```
panaarva@ubuntu:/opt$ sudo ufw enable
Firewall is active and enabled on system startup
panaarva@ubuntu:/opt$
```

Ελέγχουμε το elasticsearch αν έχει ξεκινήσει σωστά

```
curl -X GET 'http://localhost:9200'
```

```
panaarva@ubuntu:/opt$ curl -X GET 'http://localhost:9200'
{
  "name" : "ubuntu",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "araDYEp9Rpa3nQuw2JTWaA",
  "version" : {
    "number" : "7.17.3",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "5ad023604c8d7416c9eb6c0eadb62b14e766caff",
    "build_date" : "2022-04-19T08:11:19.070913226Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Θα πρέπει να αλλάξουμε τις ρυθμίσεις στο αρχείο sparkler-default.yaml έτσι ώστε να υπάρχει σύνδεση μεταξύ elasticsearch και sparkler

```
sudo nano /home/panaarva/Desktop/sparkler/build/conf/sparkler-default.yaml
```

```
crawldb.backend: elasticsearch  # "solr" is default until "elasticsearch" becomes usable.

# Type: String. Default: http://localhost:8983/solr/crawldb
# for standalone server
# For quick test crawls using embedded solr
# solr.uri: file://conf/solr/crawldb
# For cloudmode with zookeepers; Format = collectionName::zkhost1:port1,zkhost2:port2,zkhost3:port3
# solr.uri: crawldb::localhost:9983
solr.uri: http://ec2-35-174-200-133.compute-1.amazonaws.com:8983/solr/crawldb

# elasticsearch settings
elasticsearch.uri: http://localhost:9200
```

Δημιουργία index στο elasticsearch με όνομα crawldb

```
curl -X PUT 'http://localhost:9200/crawldb'
```

```
panaarva@ubuntu:~/Desktop/sparkler$ curl -X PUT 'http://localhost:9200/crawldb'
{"acknowledged":true,"shards_acknowledged":true,"index":"crawldb"}panaarva@ubuntu:~/Desktop/sparkler$
```

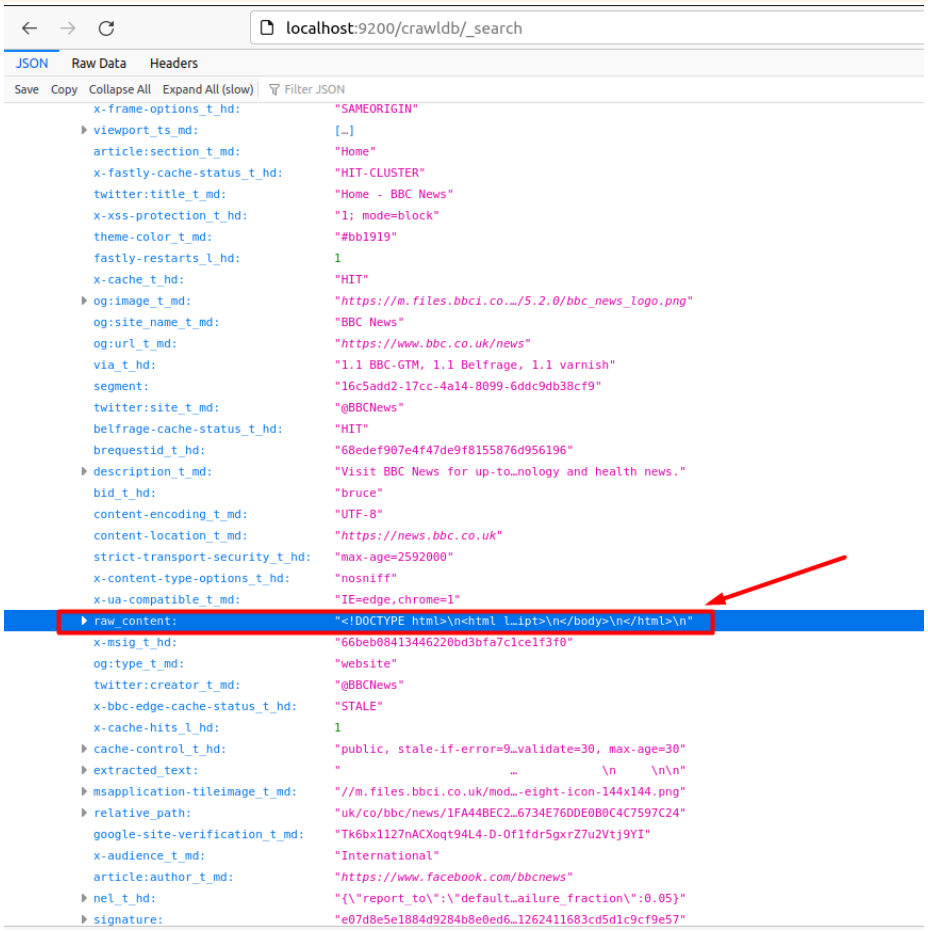
```
spark-submit --class edu.usc.irds.sparkler.Main --master spark://localhost:7077 --driver-java-options '-Dpf4j.pluginsDir=/home/panaarva/Desktop/sparkler/build/plugins' --jars $(echo /home/panaarva/Desktop/sparkler/build/sparkler-app-0.5.26-SNAPSHOT/lib/*.jar | tr ' ' ',') /home/panaarva/Desktop/sparkler/build/sparkler-app-0.5.26-SNAPSHOT/lib/com.kythera.sparkler-app-0.5.26-SNAPSHOT.jar inject -su https://news.bbc.co.uk
```

```
panaarva@ubuntu:~/Desktop/sparkler$ spark-submit --class edu.usc.irds.sparkler.Main --master spark://localhost:7077 --driver-java-options '-Dpf4j.pluginsDir=/home/panaarva/Desktop/sparkler/build/plugins'
 --jars $(echo /home/panaarva/Desktop/sparkler/build/sparkler-app-0.5.26-SNAPSHOT/lib/*.jar | tr ' ' ',') /home/panaarva/Desktop/sparkler/build/sparkler-app-0.5.26-SNAPSHOT/lib/com.kythera.sparkler-app-0
.5.26-SNAPSHOT.jar inject -su https://news.bbc.co.uk
22/05/02 00:15:42 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.230.131 instead (on interface ens33)
22/05/02 00:15:42 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
22/05/02 00:15:43 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
log4j:WARN No appenders could be found for logger (org.pf4j.DefaultPluginStatusProvider).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
>>jobId = sjob-1651475744519
```

```
java -Xms1g -cp /home/panaarva/Desktop/sparkler/build/conf:$(echo /home/panaarva/Desktop/sparkler/build/sparkler-app-0.5.26-SNAPSHOT/lib/*.jar | tr ' ' ':') -Dpf4j.pluginsDir=/home/panaarva/Desktop/sparkler/build/plugins edu.usc.irds.sparkler.Main inject -id sjob-1 -su https://news.bbc.co.uk
```

```
panaarva@ubuntu:~/Desktop/sparkler$ java -Xms1g -cp /home/panaarva/Desktop/sparkler/build/conf:$(echo /home/panaarva/Desktop/sparkler/build/sparkler-app-0.5.26-SNAPSHOT/lib/*.jar | tr ' ' ':') -Dpf4j.plu
ginsDir=/home/panaarva/Desktop/sparkler/build/plugins edu.usc.irds.sparkler.Main inject -id sjob-1 -su https://news.bbc.co.uk
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/panaarva/Desktop/sparkler/build/sparkler-app-0.5.26-SNAPSHOT/lib/ch.qos.logback.logback-classic-1.2.6.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/panaarva/Desktop/sparkler/build/sparkler-app-0.5.26-SNAPSHOT/lib/org.apache.logging.log4j.log4j-slf4j-impl-2.11.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/panaarva/Desktop/sparkler/build/sparkler-app-0.5.26-SNAPSHOT/lib/org.slf4j.slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
```

```
java -Xms1g -cp /home/panaarva/Desktop/sparkler/build/conf:$(echo /home/panaarva/Desktop/sparkler/build/sparkler-app-0.5.26-SNAPSHOT/lib/*.jar | tr ' ' ':') -Dpf4j.pluginsDir=/home/panaarva/Desktop/sparkler/build/plugins edu.usc.irds.sparkler.Main crawl -id sjob-1 -tn 10 -i 1
```



sudo apt install cmdtest

Παράρτημα 3

Τελική Εφαρμογή: API

saveOrUpdate.js

```
const express = require('express');
const router = express.Router();
const createError = require('http-errors');
const axios = require('axios');
// Add Data.
router.post('/', async (req, res, next) => {
   let hits = []
   const data = req.body
   try {
       const el_response = await axios.put("http://localhost:9200/data/_doc/" +
data.id, data)
       res.status(200).json({status: 200, statusText: el_response.statusText});
   } catch (e) {
       console.error(e);
       res.status(200).json({status: 404, statusText: "Error",});
   }
})
// Get Data.
router.get('/:region', async (req, res, next) => {
   const region = req.params.region
   const body = {
       "size": 10000,
       "query":
           {
               "bool":
                   {
                       "must": [
                           {"term": {"region": region}}
                       ]

                   }
           }
   }
   try {
       const {data} = await axios.post("http://localhost:9200/data/_search", body)
       res.status(200).json({status: 200, statusText: data.statusText, data:
data.hits.hits});
   } catch (e) {
       console.error(e);
       res.status(200).json({status: 404, statusText: "Error", e, data: []});
```

```
    }
})
// Get by keywords
router.post('/keywords', async (req, res, next) => {
    const {keywords} = req.body
    const body = {
        "size": 10000,
        "query":
            {
                "bool": {
                    "must": {
                        "multi_match": {
                            "query": keywords.toString(),
                            "fields": ["facilities.title"]
                        }
                    }
                }
            }
    }
    try {
        const {data} = await axios.post("http://localhost:9200/data/_search", body)
        res.status(200).json({status: 200, statusText: data.statusText, data:
data.hits.hits});
    } catch (e) {
        console.error(e);
        res.status(200).json({status: 404, statusText: "Error", e, data: []});
    }
})
// Get Data from a specific id.
router.get('/hotel/:id', async (req, res, next) => {
    const id = req.params.id
    try {
        const body = {
            "size": 10000,
            "query":
                {
                    "bool":
                        {
                            "must": [
                                {"term": {"_id": id}}
                            ]

                        }
                }
        }
        const {data} = await axios.post("http://localhost:9200/data/_search", body)
        res.status(200).json({status: 200, statusText: data.statusText, data:
```

```
data.hits.hits});
   } catch (e) {
       console.error(e);
       res.status(200).json({status: 404, statusText: "Error", e, data: []});
   }
})
module.exports = router;
```

seed.js

```
const express = require('express');
const router = express.Router();
const createError = require('http-errors');
const axios = require('axios');
// GET urls.
router.get('/', async (req, res, next) => {
   try {
       const {data} = await
axios.get("http://localhost:9200/seed/_search?size=10000")
       res.status(200).json({status: 200, statusText: data.statusText, data:
data.hits.hits});
   } catch (e) {
       console.error(e);
       res.status(200).json({status: 404, statusText: "Error", e, data: []});
   }
})
router.put('/', async (req, res, next) => {
   const body = req.body
   try {
       const {data} = await axios.put("http://localhost:9200/seed/_doc/" +
(body.channel + "_" + body.region), body)
       console.log(data)
       res.status(200).json({status: 200, statusText: data.statusText, resp:
data});
   } catch (e) {
       console.error(e);
       res.status(200).json({status: 404, statusText: "Error", resp: e});
   }
})
module.exports = router;
```

app.js

```
const express = require('express')
const saveOrUpdateRoutes = require('./routes/saveOrUpdate');
const seedRoutes = require('./routes/seed');
const swaggerUi = require('swagger-ui-express');
swaggerDocument = require('./swagger.json');
var cors = require('cors')
```

```
const app = express()
app.use(express.json());
app.use(cors())
const port = 4000

app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument));
app.use("/data", saveOrUpdateRoutes);
app.use("/seed", seedRoutes);

app.listen(port, () => {
   console.log(`
================================================================================
===
        _____   ____                __        ____
       |\\  ||   |   |  /      |    |      /   \\     |    \\  |
       | \\ ||   |   |    |      |----| ---  /_____\\    |___/  |
       |  \\|\\___/   | \\____  |    |    /       \\   |       |
================================================================================
===
`);
   console.log(`[INFO] API: http://localhost:${port}`);
   console.log(`[INFO] SWAGGER: http://localhost:${port}/api-docs`);
})
```

swagger.json

```
{
 "swagger": "2.0",
 "info": {
   "version": "1.0.0",
   "title": "Scrape Project",
   "description": "A project about my diploma",
   "license": {
     "name": "MIT",
     "url": "https://opensource.org/licenses/MIT"
   }
 },
 "host": "localhost:4000",
 "basePath": "/",
 "tags": [
   {
     "name": "Seed Endpoint",
     "description": "Endpoints to manage urls"
   },
   {
     "name": "Data",
     "description": "Endpoints for the data"
   }
```

```
],
"schemes": [
  "http"
],
"consumes": [
  "application/json"
],
"produces": [
  "application/json"
],
"paths": {
  "/seed": {
    "get": {
      "tags": [
        "Seed Endpoint"
      ],
      "summary": "Get urls From elasticsearch",
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "$ref": "#/definitions/seed_schema"
          }
        }
      }
    },
    "put": {
      "tags": [
        "Seed Endpoint"
      ],
      "summary": "Get urls From elasticsearch",
      "parameters": [
        {
          "name": "body",
          "in": "body",
          "description": "body",
          "required": true,
          "schema": {
            "$ref": "#/definitions/_source_seed_mapping"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "$ref": "#/definitions/data_schema"
```

```
        }
      }
    }
  }
},
"/data/{region}": {
  "get": {
    "tags": [
      "Data"
    ],
    "parameters": [
      {
        "name": "region",
        "in": "path",
        "description": "Region",
        "required": true,
        "type": "string"
      }
    ],
    "summary": "Get data From elasticsearch for a specific region",
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "$ref": "#/definitions/data_schema"
        }
      }
    }
  }
},
"/data/hotel/{id}": {
  "get": {
    "tags": [
      "Data"
    ],
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "description": "Hotel ID",
        "required": true,
        "type": "string"
      }
    ],
    "summary": "Get data From elasticsearch for a specific id",
    "responses": {
      "200": {
```

```
          "description": "OK",
          "schema": {
            "$ref": "#/definitions/data_schema"
          }
        }
      }
    }
  },
  "/data": {
    "put": {
      "tags": [
        "Data"
      ],
      "parameters": [
        {
          "name": "put data",
          "in": "body",
          "description": "Body of new hotel",
          "schema": {
            "$ref": "#/definitions/_source_data_mapping"
          }
        }
      ],
      "summary": "Save or update a hotel",
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "$ref": "#/definitions/data_schema"
          }
        }
      }
    }
  },
  "/data/keywords": {
    "post": {
      "tags": [
        "Data"
      ],
      "parameters": [
        {
          "name": "keywords_body",
          "in": "body",
          "description": "All the keywords",
          "schema": {
            "$ref": "#/definitions/keywords_mapping"
          }
```

```
        }
      ],
      "summary": "Get hotels that contains the keywords that you gave",
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "$ref": "#/definitions/data_schema"
          }
        }
      }
    }
  }
},
"definitions": {
  "data_schema": {
    "properties": {
      "status": {
        "type": "integer"
      },
      "data": {
        "type": "array",
        "$ref": "#/definitions/elastic_data_mapping"
      }
    }
  },
  "seed_schema": {
    "properties": {
      "status": {
        "type": "integer"
      },
      "data": {
        "type": "array",
        "$ref": "#/definitions/elastic_seed_mapping"
      }
    }
  },
  "elastic_data_mapping": {
    "required": [
      "_id"
    ],
    "properties": {
      "_index": {
        "type": "string"
      },
      "_type": {
        "type": "string"
```

```
      },
      "_id": {
        "type": "string"
      },
      "_source": {
        "type": "object",
        "$ref": "#/definitions/_source_data_mapping"
      },
      "_score": {
        "type": "string"
      }
    }
  },
  "elastic_seed_mapping": {
    "required": [
      "_id"
    ],
    "properties": {
      "_index": {
        "type": "string"
      },
      "_type": {
        "type": "string"
      },
      "_id": {
        "type": "string"
      },
      "_source": {
        "type": "object",
        "$ref": "#/definitions/_source_seed_mapping"
      },
      "_score": {
        "type": "string"
      }
    }
  },
  "_source_seed_mapping": {
    "required": [
      "url",
      "channel",
      "region"
    ],
    "properties": {
      "url": {
        "type": "string"
      },
      "channel": {
```

```
        "type": "string"
      },
      "region": {
        "type": "string"
      }
    }
  },
  "keywords_mapping": {
    "required": [
      "keywords"
    ],
    "properties": {
      "keywords": {
        "type": "array",
        "items": {
          "type": "string"
        }
      }
    }
  },
  "_source_data_mapping": {
    "required": [
      "id"
    ],
    "properties": {
      "hotelName": {
        "type": "string"
      },
      "address": {
        "type": "string"
      },
      "descr": {
        "type": "string"
      },
      "stars": {
        "type": "integer"
      },
      "headerOfDescription": {
        "type": "string"
      },
      "facilities": {
        "type": "array"
      },
      "globalscoreText": {
        "type": "string"
      },
      "globalscoreScore": {
```

```
        "type": "string"
      },
      "globalscore": {
        "type": "array"
      },
      "surroundings": {
        "type": "array"
      },
      "photos": {
        "type": "array"
      },
      "region": {
        "type": "string"
      },
      "id": {
        "type": "string"
      }
    }
  }
 }
}
```

Τελική Εφαρμογή: Scraper

getHotelsUrls.js

```
const puppeteer = require('puppeteer');

module.exports = async (url) => {
   //Variables
   let finalUrls = []
   let numOfpages = 0
   let removeDuplicates = []
   //initialize
   numOfpages = 0
   const tags = {
       numOfPages: "li[class=\"f32a99c8d1\"]",
       nextButton: "button[aria-label=\"Next page\"]",
       loader: ".bb52d7d9f0"
   }

   //Launch browser
   const browserOptions = {headless: false}
   const browser = await puppeteer.launch(browserOptions);
   const page = await browser.newPage();
   await page.goto(url, {
```

```
        waitUntil: 'networkidle2',
    });
    if (numOfpages === 0)
        numOfpages = await page.$$eval(tags.numOfPages, (elements) =>
elements[elements.length - 1].textContent);
    for (let y = 0; y < numOfpages; y++) {
        removeDuplicates = []
        let hrefs = await page.$$eval('a', as => as.map(a => a.href));
        hrefs = hrefs.map(a => {
            if (a.indexOf('/hotel/gr/') !== -1) {
                removeDuplicates.push(a.replace("#hotelTmpl", "").replace("&map=1",
""))
            }
        })
        removeDuplicates = [...new Set(removeDuplicates)]
        finalUrls = [...finalUrls, ...removeDuplicates]
        await page.waitForSelector(tags.loader, {hidden: true});
        await page.$eval(tags.nextButton, form => form.click());
    }
    finalUrls = [...new Set(finalUrls)]
    await browser.close();
    return finalUrls
}
```

getInformationPerHotel.js

```
const puppeteer = require('puppeteer');

module.exports = async (hotelUrl) => {
    const tags = {
        hotelName: "#hp_hotel_name > h2",
        hotelName_sec: "#hp_hotel_name",
        hotelName_tag: ".e2f34d59b1",
        address: ".hp_address_subtitle.js-hp_address_subtitle.jq_tooltip",
        descr: "#property_description_content",
        stars: "span[data-testid=\"rating-stars\"] > span",
        headerOfDescription: ".hp_header_compact.hp-hotel-description-header",
        facilities: ".hotel-facilities-group",
        facilitiesPolicy: ".hotel-facilities-group__policy",
        facilitiesPer: ".bui-list > .bui-list__item",
        facilitiesTitle: ".bui-title",
        globalscoreText: ".b1e6dd8416.b48795b3df",
        globalscoreScore: ".b5cd09854e.d10a6220b4",
        globalscore: ".hp-social_proof>.review_list_score_container > div > div > ul
> li",
        globalscoreTitle: ".c-score-bar__title",
        globalscoreScPrSc: ".c-score-bar__score",
        surroundings: ".hp_location_block__section_container",
```

```
        surroundingsTitle: ".bui-title",
        surroundingsPr: ".bui-list__body",
        photos: ".bh-photo-grid > div[aria-hidden=\"true\"] > a > img"
    }
    let data = {
        hotelName: null,
        address: null,
        descr: null,
        stars: null,
        headerOfDescription: null,
        facilities: [],
        globalscoreText: null,
        globalscoreScore: null,
        globalscore: [],
        surroundings: [],
        photos: [],
        region: null,
        id: null
    }

    //Launch browser
    const browserOptions = {headless: true}
    const browser = await puppeteer.launch(browserOptions);
    try {
        const page = await browser.newPage();
        await page.goto(hotelUrl);
        //get HotelName
        data.hotelName = await page.evaluate((tags) => {
            let hotelName = document.querySelector(tags.hotelName)
            let replaceTag = null
            if (!hotelName) {
                hotelName = document.querySelector(tags.hotelName_sec)
                replaceTag = document.querySelector(tags.hotelName_tag).textContent
            }
            if (hotelName)
                hotelName = hotelName.textContent
            if (replaceTag)
                hotelName = hotelName.replace(replaceTag, "")

            return hotelName
        }, tags)
        // data.hotelName = await page.$eval(tags.hotelName, elem =>
elem.textContent);
        data.hotelName = data.hotelName.replace(/\n/g, "").trim()
        //get Address
        data.address = await page.$eval(tags.address, elem => elem.textContent);
        data.address = data.address.replace(/\n/g, "")
```

```
        //get Descriptions
        data.descr = await page.$eval(tags.descr, elem => elem.textContent);
        //get Stars
        data.stars = await page.$$eval(tags.stars, (elements) => elements);
        data.stars = data.stars.length
        //get Header of Descriptions
        try {
            data.headerOfDescription = await page.$eval(tags.headerOfDescription,
elem => elem.textContent);
            data.headerOfDescription = data.headerOfDescription.replace(/\n/g,
"").trim()
        } catch (e) {
        }
        // get facilities
        data.facilities = await page.evaluate((tags) => {
            let returnArr = []
            const elements = document.querySelectorAll(tags.facilities)
            for (let i = 0; i < elements.length; i++) {
                let title = null
                let results = []
                title =
elements[i].querySelector(tags.facilitiesTitle).textContent.replace(/\n/g, "");
                let policy = elements[i].querySelector(tags.facilitiesPolicy)
                if (policy)
                    policy = policy.textContent.replace(/\n/g, "")
                const dt = elements[i].querySelectorAll(tags.facilitiesPer);
                results = Array.prototype.map.call(dt, function (t) {
                    return t.textContent.replace(/\n/g, "");
                });
                returnArr.push({title, facilities: results, policy})
            }
            return returnArr
        }, tags);
        // Get Globalscore
        data.globalscoreText = await page.$eval(tags.globalscoreText, elem =>
elem.textContent);
        data.globalscoreText.replace(/\n/g, "")
        data.globalscoreScore = await page.$eval(tags.globalscoreScore, elem =>
elem.textContent);
        data.globalscore = await page.evaluate((tags) => {
            let returnArr = []
            const elements = document.querySelectorAll(tags.globalscore)
            for (let i = 0; i < elements.length; i++) {
                let title = null
                let score = null
                title =
elements[i].querySelector(tags.globalscoreTitle).textContent.replace(/\n/g, "");
```

```
                score =
elements[i].querySelector(tags.globalscoreScPrSc).textContent.replace(/\n/g, "");
                returnArr.push({title, score})
            }
            return returnArr
        }, tags);
        // Get surroundings
        data.surroundings = await page.evaluate((tags) => {
            let returnArr = []
            const elements = document.querySelectorAll(tags.surroundings)
            for (let i = 0; i < elements.length; i++) {
                let title = null
                let results = []
                title =
elements[i].querySelector(tags.surroundingsTitle).textContent.replace(/\n/g, "");
                const dt = elements[i].querySelectorAll(tags.surroundingsPr);
                results = Array.prototype.map.call(dt, function (t) {
                    return t.textContent.replace(/\n/g, "");
                });
                returnArr.push({title, surroundings: results})
            }
            return returnArr
        }, tags);
        data.photos = await page.evaluate((tags) => {
            return Array.from(document.querySelectorAll(tags.photos), e => e.src)
        }, tags);
        await browser.close();
        return data
    } catch (e) {
        await browser.close();
        return "notSave"
    }
}
```

saveElastic.js

```
const axios = require('axios');

module.exports = async (data) => {
    return await axios.post("http://localhost:4000/data", data, {})
}
```

app.js

```
const getHotelUrls = require('./src/getHotelUrls');
const getInformationPerHotel = require('./src/getInformationPerHotel');
const saveElastic = require('./src/saveElastic');
const axios = require('axios');
```

```
(async () => {
   let urls = await axios.get("http://localhost:4000/seed")
   urls = urls.data && urls.data.data ? urls.data.data : []
   for (let url of urls) {
       const finalUrl = url._source.url
       const hotelUrls = await getHotelUrls(finalUrl)
       for (let hotelUrl of hotelUrls) {
           try {
               let data = await getInformationPerHotel(hotelUrl);
               if (data !== "notSave") {
                   data.region = finalUrl.substr(finalUrl.indexOf("ss=") + 3,
finalUrl.length)
                   const suburl = hotelUrl.substr(hotelUrl.indexOf("hotel/gr/") +
("hotel/gr/").length, hotelUrl.length)
                   data.id = suburl.substr(0, suburl.indexOf("."))
                   //Save or Update data (ElasticSearch)
                   const resp = await saveElastic(data)
               }
           } catch (e) {
               console.log("Error")
               console.log(e)
           }
       }
   }
})();
```

Τελική Εφαρμογή: Front end

App.js

```
import React from 'react';
import './App.css';
import Header from './components/header'
import Main from "./pages/main"
import Hotels from "./pages/hotels"
import Hotel from "./pages/hotel"
import {BrowserRouter as Router, Routes, Route} from 'react-router-dom';

function App() {
   return (
       <div className="App">
           <Header/>
           <Router>
               <Routes>
                   <Route exact path='/' element={< Main/>}/>
                   <Route path='/hotels/:region' element={< Hotels/>}/>
                   <Route path='/hotel/:id' element={< Hotel/>}/>
```

```
                </Routes>
            </Router>
        </div>
    );
}

export default App;
```

hotel.js

```
import * as React from 'react';
import {useEffect, useState} from "react";
import {useLocation} from "react-router-dom";
import axios from "axios";
import StandardImageList from '../components/imageList'
import Grid from "@mui/material/Grid";
import Typography from "@mui/material/Typography";
import PlaceIcon from "@mui/icons-material/Place";
import {styled} from "@mui/material/styles";
import Paper from "@mui/material/Paper";
import RatingComp from '../components/rating'
import Loader from "../components/loader"

const Item = styled(Paper)(({theme}) => ({
    backgroundColor: theme.palette.mode === 'dark' ? '#1A2027' : '#fff',
    ...theme.typography.body2,
    padding: theme.spacing(1),
    textAlign: 'center',
    color: theme.palette.text.secondary,
}));
export default function Hotel() {
    const location = useLocation();
    const [data, setData] = useState([])
    const [loading, setLoading] = useState(false)
    useEffect(() => {
        const id = location.pathname.replace("/hotel/", "")
        if (id) {
            getData(id)
        }
    }, [location])
    const getData = async (id) => {
        setLoading(true)
        const {data} = await axios.get("http://localhost:4000/data/hotel/" + id)
        console.log(data.data[0]._source)
        setData(data.data)
        setLoading(false)
    }
    return (
```

```
       <>
           {loading ?
               <Loader/> :
               <>{data.map((dt) => {
                   return <Grid container spacing={2}>
                       <Grid container rowSpacing={1} columnSpacing={{xs: 1, sm: 2,
md: 3}}>
                           <Grid item xs={2}>
                           </Grid>
                           <Grid item xs={4}>
                               <StandardImageList imageData={dt._source.photos}/>
                           </Grid>
                           <Grid item xs={4}>
                               <Grid container spacing={2} width={"100%"}
height={"40%"}
                                     style={{margin: "30px", marginRight: "0px",
textAlign: "justify"}}>
                                   <Grid item xs={12}>
                                       <Typography variant="h3" component="div"
gutterBottom>
                                           {dt._source.hotelName}
                                       </Typography>
                                   </Grid>
                                   <Grid item xs={12}>
                                       <Typography variant="subtitle1"
color="text.secondary" component="div"
                                                   style={{display: "flex",
alignItems: "center"}}>
                                           <PlaceIcon
fontSize={"small"}/>{dt._source.address}
                                       </Typography>
                                   </Grid>
                                   <Grid item xs={12}>
                                       {dt._source.descr}
                                   </Grid>
                               </Grid>
                           </Grid>
                           <Grid item xs={2}>
                           </Grid>
                       </Grid>
                       <Grid container rowSpacing={1} columnSpacing={{xs: 1, sm: 2,
md: 3}} style={{
                           display: "flex", flexDirection: "column",
                           alignItems: "center"
                       }}>
                           <Grid item xs={12}>
                               <Item><Typography variant="h4" component="div"
```

```
gutterBottom>
                                    Score: {dt._source.globalscoreScore}
                                </Typography>
                                    <RatingComp value={dt._source.globalscoreScore /
2}/></Item>
                            </Grid>
                        </Grid>
                        <Grid container rowSpacing={1} columnSpacing={{xs: 1, sm: 2,
md: 3}}>
                            <Grid item xs={2}>
                            </Grid>
                            <Grid item xs={8}>
                                <Grid container spacing={2} width={"100%"}
height={"40%"}
                                      style={{
                                          margin: "30px",
                                          marginRight: "0px",
                                          marginBottom: "-35px",
                                          textAlign: "justify"
                                      }}>
                                    <Grid item xs={12}>
                                        <Typography variant="h3" component="div"
gutterBottom>
                                            Facilities
                                        </Typography>
                                    </Grid>
                                </Grid>
                                <Grid container spacing={2} width={"100%"}
height={"40%"}
                                      style={{margin: "30px", marginRight: "0px",
textAlign: "justify"}}>
                                    {dt._source.facilities.map((fc) => {
                                        return (<Grid item xs={2}>
                                            <Item>{fc.title}</Item>
                                        </Grid>)
                                    })}
                                </Grid>
                            </Grid>
                            <Grid item xs={2}>
                            </Grid>

                        </Grid>
                        <Grid container rowSpacing={1} columnSpacing={{xs: 1, sm: 2,
md: 3}}>
                            <Grid item xs={2}>
                            </Grid>
                            <Grid item xs={8}>
```

```
                                <Grid container spacing={2} width={"100%"}
height={"40%"}
                                      style={{
                                          margin: "30px",
                                          marginRight: "0px",
                                          marginBottom: "-35px",
                                          textAlign: "justify"
                                      }}>
                                    <Grid item xs={12}>
                                        <Typography variant="h3" component="div"
gutterBottom>
                                            Surroundings
                                        </Typography>
                                    </Grid>
                                </Grid>
                                <Grid container spacing={2} width={"100%"}
height={"40%"}
                                      style={{margin: "30px", marginRight: "0px",
textAlign: "justify"}}>
                                    {dt._source.surroundings.map((sr) => {
                                        return (<Grid item xs={2}>
                                            <Item>{sr.title}</Item>
                                        </Grid>)
                                    })}
                                </Grid>
                            </Grid>
                            <Grid item xs={2}>
                            </Grid>

                        </Grid>
                    </Grid>
                })}</>
            }
        </>
    )
}
```

hotels.js

```
import * as React from 'react';
import {useEffect, useState} from "react";
import {useLocation} from "react-router-dom";
import axios from "axios";
import HotelView from "../components/hotelView"
import Pagination from "@mui/material/Pagination";
import Stack from "@mui/material/Stack";
import Loader from "../components/loader"
import Typography from "@mui/material/Typography";
```

```
export default function Hotels(props) {
   const location = useLocation();
   const [data, setData] = useState([])
   const [allData, setAllData] = useState([])
   const [region, setRegion] = useState("")
   const [loading, setLoading] = useState(false)
   const [totalPages, setTotalPages] = useState(0)
   const [page, setPage] = React.useState(1);

   useEffect(() => {
       const reg = location.pathname.replace("/hotels/", "")
       if (reg && reg.indexOf('hotels') === -1) {
           if(location.search){
               setRegion(reg)
               getData(reg)
           }else {
               setRegion(reg)
               getData(reg)
           }
       }
   }, [location])
   const handleChange = (event, value) => {
       setPage(value);
       setData(allData.slice((value - 1) * 10, (value * 10) - 1))
   };
   const getData = async (reg) => {
       setLoading(true)
       let dt;
       if(location.search) {
           const body = {
               "keywords": [reg]
           }
           const {data} = await axios.post("http://localhost:4000/data/keywords" ,
body)
           console.log(data)
           dt = data.data
       }else{
           const {data} = await axios.get("http://localhost:4000/data/" + reg)
           dt = data.data
       }
       setData(dt.slice(0, 9))
       setAllData(dt)
       setTotalPages(Math.ceil(dt.length / 10))
       setLoading(false)
   }
   return (
```

```
        <>
            {loading ?
                <Loader/> :
                <div style={{
                    display: "flex",
                    alignItems: "center",
                    flexDirection: "column"
                }}>
                    <Typography variant="h4" component="div" gutterBottom
style={{margin: "20px"}}>
                        {(location.search)?"Keyword":"Region"}: {region}
                    </Typography>
                    <hr style={{
                        borderColor: "whitesmoke",
                        borderStyle: "solid",
                        borderWidth: "2px",
                        width: "100%"
                    }}/>
                    <HotelView data={data}/>
                    <Stack spacing={2} style={{margin: "20px"}}>
                        <Pagination count={totalPages} page={page}
onChange={handleChange}/>
                    </Stack>
                </div>
            }
        </>
    )
}
```

main.js

```
import * as React from 'react';
import Search from "../components/search";
import Collage from "../components/collage";

export default function Main() {
    return (
        <>
            <Search/>
            <Collage/>
        </>
    )
}
```

collage.js

```
import * as React from 'react';
import {styled} from '@mui/material/styles';
import Box from '@mui/material/Box';
```

```
import ButtonBase from '@mui/material/ButtonBase';
import Typography from '@mui/material/Typography';
import {Link} from "react-router-dom";

const images = [
   {
       url: '/images/collage/santorini.png',
       title: 'Santorini',
       width: '33%',
       search: "santorini"
   },
   {
       url: '/images/collage/mikonos.png',
       title: 'Mikonos',
       width: '34%',
       search: "mikonos"
   },
   {
       url: '/images/collage/kerkira.png',
       title: 'Kerkira',
       width: '33%',
       search: "kerkira"
   },
];

const ImageButton = styled(ButtonBase)(({theme}) => ({
   position: 'relative',
   height: 200,
   [theme.breakpoints.down('sm')]: {
       width: '100% !important', // Overrides inline-style
       height: 100,
   },
   '&:hover, &.Mui-focusVisible': {
       zIndex: 1,
       '& .MuiImageBackdrop-root': {
           opacity: 0.15,
       },
       '& .MuiImageMarked-root': {
           opacity: 0,
       },
       '& .MuiTypography-root': {
           border: '4px solid currentColor',
       },
   },
}));

const ImageSrc = styled('span')({
```

```
  position: 'absolute',
  left: 0,
  right: 0,
  top: 0,
  bottom: 0,
  backgroundSize: 'cover',
  backgroundPosition: 'center 40%',
});

const Image = styled('span')(({theme}) => ({
  position: 'absolute',
  left: 0,
  right: 0,
  top: 0,
  bottom: 0,
  display: 'flex',
  alignItems: 'center',
  justifyContent: 'center',
  color: theme.palette.common.white,
}));

const ImageBackdrop = styled('span')(({theme}) => ({
  position: 'absolute',
  left: 0,
  right: 0,
  top: 0,
  bottom: 0,
  backgroundColor: theme.palette.common.black,
  opacity: 0.4,
  transition: theme.transitions.create('opacity'),
}));

const ImageMarked = styled('span')(({theme}) => ({
  height: 3,
  width: 18,
  backgroundColor: theme.palette.common.white,
  position: 'absolute',
  bottom: -2,
  left: 'calc(50% - 9px)',
  transition: theme.transitions.create('opacity'),
}));

export default function ButtonBases() {
  return (
      <Box sx={{display: 'flex', flexWrap: 'wrap', minWidth: 300, width: '100%'}}>
          {images.map((image) => (
              <ImageButton
```

```
                    focusRipple
                    key={image.title}
                    style={{
                        width: image.width,
                    }}
                    component={Link} to={{pathname: "/hotels/" + image.search}}
                >
                    <ImageSrc style={{backgroundImage: `url(${image.url})`}}/>
                    <ImageBackdrop className="MuiImageBackdrop-root"/>
                    <Image>
                        <Typography
                            component="span"
                            variant="subtitle1"
                            color="inherit"
                            sx={{
                                position: 'relative',
                                p: 4,
                                pt: 2,
                                pb: (theme) => `calc(${theme.spacing(1)} + 6px)`,
                            }}
                        >
                            {image.title}
                            <ImageMarked className="MuiImageMarked-root"/>
                        </Typography>
                    </Image>
                </ImageButton>
            ))}
        </Box>
    );
}
```

header.js

```
import * as React from 'react';
import AppBar from '@mui/material/AppBar';
import Box from '@mui/material/Box';
import Toolbar from '@mui/material/Toolbar';
import Typography from '@mui/material/Typography';

export default function Header() {
   return (
       <Box sx={{flexGrow: 1}}>
           <AppBar position="static" style={{backgroundColor: "#282c34"}}>
               <Toolbar>
                   <Typography variant="h6" component="div" sx={{flexGrow: 1}}>
                       Application
                   </Typography>
               </Toolbar>
```

```
            </AppBar>
        </Box>
    );
}
```

hotelView.js

```
import * as React from 'react';
import Box from '@mui/material/Box';
import Card from '@mui/material/Card';
import CardContent from '@mui/material/CardContent';
import CardMedia from '@mui/material/CardMedia';
import Typography from '@mui/material/Typography';
import PlaceIcon from '@mui/icons-material/Place';
import {Link} from "react-router-dom"

export default function HotelView({data}) {
    return (
        <>
            {data.map((dt) => {

                return (
                    <Card sx={{display: 'flex'}} style={{margin: "30px", width:
"50%", textDecoration: 'none'}}
                          component={Link} to={{pathname: "/hotel/" + dt._id}}>
                        <CardMedia
                            style={{margin: "10px"}}
                            component="img"
                            sx={{width: 200}}
                            image={dt._source.photos[0]}
                            alt="Live from space album cover"
                        />
                        <Box sx={{display: 'flex', flexDirection: 'column'}}>
                            <CardContent sx={{flex: '1 0 auto'}}>
                                <Typography component="div" variant="h5"
style={{display: "flex"}}>
                                    {dt._source.hotelName}
                                </Typography>
                                <Typography variant="subtitle1"
color="text.secondary" component="div"
                                            style={{display: "flex", alignItems:
"center"}}>
                                    <PlaceIcon
fontSize={"small"}/>{dt._source.address}
                                </Typography><br/>
                                <Typography variant="subtitle2" color="text."
component="div"
                                            style={{display: "flex", alignItems:
```

```
"center", textAlign: "justify"}}>
                                    {dt._source.descr.substr(0, 300)}{"..."}
                                </Typography>
                            </CardContent>
                        </Box>
                    </Card>)
            })}
        </>
    );
}
```

imageList.js

```
import * as React from 'react';
import Grid from "@mui/material/Grid";

export default function StandardImageList({imageData}) {
   return (
       <Grid container spacing={2} width={"100%"} height={"40%"} style={{margin:
"30px"}}>
           <Grid item xs={12}>
               <Grid item xs={8}>
                   <img src={imageData[0]} width={"100%"} height={"300px"}/>
               </Grid>
           </Grid>
           <Grid item sm={4} xl={4}>
               <img src={imageData[1]} width={"100%"}/>
           </Grid>
           <Grid item sm={4} xl={4}>
               <img src={imageData[2]} width={"100%"}/>
           </Grid>

       </Grid>
   );
}
```

listFeutures.js

```
import * as React from 'react';
import List from '@mui/material/List';
import ListItem from '@mui/material/ListItem';
import ListItemText from '@mui/material/ListItemText';
import ListItemAvatar from '@mui/material/ListItemAvatar';
import Avatar from '@mui/material/Avatar';
import ImageIcon from '@mui/icons-material/Image';
import WorkIcon from '@mui/icons-material/Work';
import BeachAccessIcon from '@mui/icons-material/BeachAccess';

export default function FolderList() {
```

```
    return (
        <List sx={{width: '100%', maxWidth: 360, bgcolor: 'background.paper'}}>
            <ListItem style={{display: 'list-item'}}>
                <ListItemAvatar>
                    <Avatar>
                        <ImageIcon/>
                    </Avatar>
                </ListItemAvatar>
                <ListItemText primary="Photos" secondary="Jan 9, 2014"/>
            </ListItem>
            <ListItem>
                <ListItemAvatar>
                    <Avatar>
                        <WorkIcon/>
                    </Avatar>
                </ListItemAvatar>
                <ListItemText primary="Work" secondary="Jan 7, 2014"/>
            </ListItem>
            <ListItem>
                <ListItemAvatar>
                    <Avatar>
                        <BeachAccessIcon/>
                    </Avatar>
                </ListItemAvatar>
                <ListItemText primary="Vacation" secondary="July 20, 2014"/>
            </ListItem>
        </List>
    );
}
```

loader.js

```
import * as React from 'react';
import CircularProgress from '@mui/material/CircularProgress';
import Box from '@mui/material/Box';

export default function CircularIndeterminate() {
    return (
        <Box sx={{display: 'flex'}} style={{
            height: "200px",
            display: "flex",
            justifyContent: "center",
            alignItems: "center"
        }}>
            <CircularProgress style={{
                width: "70px",
                height: "70px"
            }}/>
```

```
        </Box>
    );
}
```

rating.js

```
import * as React from 'react';
import {Rating} from "@mui/material";

export default function RatingComp({value}) {
    return (
        <Rating name="read-only" value={value} readOnly/>
    );
}
```

search.js

```
import * as React from 'react';
import Grid from '@mui/material/Grid';
import Paper from '@mui/material/Paper';
import {styled} from '@mui/material/styles';
import {TextField, Button} from '@mui/material';
import Autocomplete from '@mui/material/Autocomplete';
import {Link} from 'react-router-dom'
import Tabs from '@mui/material/Tabs';
import Tab from '@mui/material/Tab';
import Typography from '@mui/material/Typography';
import Box from '@mui/material/Box';
import PropTypes from 'prop-types';

function TabPanel(props) {
    const {children, value, index, ...other} = props;

    return (
        <div
            role="tabpanel"
            hidden={value !== index}
            id={`simple-tabpanel-${index}`}
            aria-labelledby={`simple-tab-${index}`}
            {...other}
        >
            {value === index && (
                <Box sx={{p: 3}}>
                    <Typography>{children}</Typography>
                </Box>
            )}
        </div>
    );
}
```

```
TabPanel.propTypes = {
   children: PropTypes.node,
   index: PropTypes.number.isRequired,
   value: PropTypes.number.isRequired,
};

function a11yProps(index) {
   return {
       id: `simple-tab-${index}`,
       'aria-controls': `simple-tabpanel-${index}`,
   };
}

const options = ['santorini', 'mikonos'];


const Item = styled(Paper)(({theme}) => ({
   ...theme.typography.body2,
   textAlign: 'center',
   color: theme.palette.text.secondary,
   height: 60,
   lineHeight: '60px',
}));

export default function Search() {
   const [value, setValue] = React.useState();
   const [inputValue, setInputValue] = React.useState('');
   const [inputValueKey, setInputValueKey] = React.useState('');
   const [valueTab, setValueTab] = React.useState(0);

   const handleChange = (event, newValue) => {
       setValueTab(newValue);
   };
   return (
       <Grid container style={{
           backgroundColor: "#f2f2f2",
           height: "400px",
           display: "flex",
           alignContent: "center",
           backgroundImage:
"url(https://wallpapersafari.com/santorini-greece-4k-wallpapers/)"
       }}>
           <Grid item xs={2}></Grid>
           <Grid item xs={8}>
               <Item key="16" elevation="16"
                     style={{height: "200px", borderRadius: "10px", display:
```

```
"flex", alignItems: "center"}}>
                    <Box sx={{width: '100%', height: "100%", marginTop: "5px"}}>
                        <Box sx={{borderBottom: 1, borderColor: 'divider'}}>
                            <Tabs value={valueTab} onChange={handleChange}
aria-label="basic tabs example">
                                <Tab label="Search By region" {...a11yProps(0)} />
                                <Tab label="Search by keyword" {...a11yProps(1)} />
                            </Tabs>
                        </Box>
                        <TabPanel value={valueTab} index={0}>
                            <div style={{"display": "flex"}}>
                                <Autocomplete
                                    value={value}
                                    onChange={(event, newValue) => {
                                        setValue(newValue);
                                    }}
                                    inputValue={inputValue}
                                    onInputChange={(event, newInputValue) => {
                                        setInputValue(newInputValue);
                                    }}
                                    id="controllable-states-demo"
                                    options={options}
                                    style={{margin: "10px", marginLeft: "25px"}}
                                    fullWidth
                                    renderInput={(params) => <TextField
                                        fullWidth
                                        id="standard-bare"
                                        variant="outlined"
                                        placeholder="Where are you going?"
                                        {...params}
                                    />}
                                />
                                <Button variant="contained" component={Link}
                                        to={{pathname: "/hotels/" + value, state:
{value}}}

                                        style={{height: "53px", margin: "10px",
marginRight: "25px"}}>Search</Button>
                            </div>
                        </TabPanel>
                        <TabPanel value={valueTab} index={1}>
                            <div style={{"display": "flex"}}>
                                <Autocomplete
                                    freeSolo={true}
                                    options={[]}
                                    inputValue={inputValueKey}
                                    onInputChange={(event, newInputValue) => {
                                        console.log(newInputValue)
```

```
                                    setInputValueKey(newInputValue);
                                }}
                                id="controllable-states-demo"
                                style={{margin: "10px", marginLeft: "25px"}}
                                fullWidth
                                renderInput={(params) => <TextField
                                    fullWidth
                                    id="standard-bare"
                                    variant="outlined"
                                    placeholder="Search by keywords"
                                    {...params}
                                />}
                            />
                            <Button variant="contained" component={Link}
                                    to={{pathname: "/hotels/" +
inputValueKey+"?keywords=true", state: {inputValueKey}}}
                                    style={{height: "53px", margin: "10px",
marginRight: "25px"}}>Search</Button>
                        </div>
                    </TabPanel>
                </Box>
            </Item>
        </Grid>
        <Grid item xs={2}></Grid>
    </Grid>
  );
}
```