Ανάλυση και Σχεδιασμός Πληροφοριακών Συστημάτων

Θέμα Β

Υποσύστημα Διασύνδεσης με τρίτα συστήματα

Δι@ύγεια - API Ανοιχτών Δεδομένων

Άντληση και Αποθήκευση Δεδομένων

Φοίβος-Ευστράτιος Καλεμκερής 03116010

Θωμάς Δούκας 03116081

Επαμεινώνδας Ορέστης Μπάτσης 03116647

1 Εισαγωγή

Στο πλαίσιο της εργασίας αυτής, επιλέξαμε να υλοποιήσουμε έναν μηχανισμό άντλησης και αποθήκευσης δεδομένων από το ΔΙΑΥΓΕΙΑ. Επιπλέον, υλοποιήσαμε ένα απλό Command Line Interface (CLI), μέσω του οποίου ο χρήστης έχει τη δυνατότητα άντλησης δεδομένων από τη διεπαφή της Διαύγειας με βάση την ημερομηνία έκδοσης, αναζήτησης στα τοπικά αποθηκευμένα δεδομένα με χρήση διαφορετικών φίλτρων, καθώς και εξαγωγής των τοπικά αποθηκευμένων αρχείων σε μορφή JSON ή AKN4EU για χρήση και εκτός της εφαρμογής.

2 Άντληση Δεδομένων από το Diavgeia OpenData API

Η βασική λειτουργία της εφαρμογής συνίσταται στην άντληση του συνόλου των μεταδεδομένων από το Diavgeia OpenData API και την αποθήκευσή τους σε μορφή AKN4EU. Προκειμένου να συγκεντρώσουμε το σύνολο των μεταδεδομένων για όλες τις αναρτημένες πράξεις εντός του χρονικού διαστήματος που μας ενδιαφέρει, πραγματοποιούμε, αρχικά, μια απλή αναζήτηση πράξεων (GET /search). Η κλήση αυτή επιστρέφει μεταδεδομένα για όλες τις πράξεις που βρίσκονται σε ισχύ. Μας ενδιαφέρει να αποθηκεύσουμε μόνο τις πράξεις που δε βρίσκονται ήδη στην τοπική βάση δεδομένων (αποφυγή διπλοτύπων). Για το λόγο αυτό, για κάθε πράξη που επιστρέφεται από την αρχική κλήση, πραγματοποιούμε ερώτημα στη βάση χρησιμοποιώντας το αναγνωριστικό αυτής (Αριθμός Διαδικτυακής Ανάρτησης, 'ada'). Αν η πράξη δε βρεθεί, πραγματοποιούμε μια δεύτερη κλήση στη διεπαφή της Διαύγειας για την ανάκτηση του ιστορικού της (GET /decisions/:ada/versionlog) βάσει και πάλι του αναγνωριστικού 'ada'. Το ιστορικό περιλαμβάνει μια λίστα από προηγούμενες εκδόσεις της πράξης και θα χρησιμοποιηθεί για τον εμπλουτισμό των μεταδεδομένων που αντλήθηκαν προηγουμένως.

Κατά την διαδικασία άντλησης, επιτρέπεται η επιλογή του πλήθους των δεδομένων προς άντληση, καθώς και του εύρους ημερομηνιών έκδοσης (from-to issueDate). Σε περίπτωση που ο χρήστης αφήσει κενό το πεδίο πλήθους δεδομένων, αυτό ορίζεται αυτόματα σε 10. Αντίστοιχα, αν αφεθεί κενό κάποιο από τα πεδία FromIssueDate και ToIssueDate, τότε:

- Αν οριστούν τιμές για τα FromIssueDate και ToIssueDate και το διάστημα που ορίζουν ξεπερνά τις 180 ημέρες, το σύστημα αυτομάτως εισάγει ως τιμή του ToIssueDate την τιμή του FromIssueDate "συν" 180 ημέρες.
- Αν οριστεί τιμή για το ToIssueDate, αλλά όχι για το FromIssueDate , τότε το σύστημα αυτομάτως εισάγει ως τιμή το ToIssueDate "μείον" 180 ημέρες.
- Αν οριστεί τιμή για το FromIssueDate, αλλά όχι για το ToIssueDate, τότε το σύστημα αυτομάτως εισάγει ως τιμή το FromIssueDate "συν" 180 ημέρες.
- Αν δεν οριστεί τιμή για κανένα από τα FromIssueDate και ToIssueDate, τότε το σύστημα αυτομάτως εισάγει ως τιμή για το ToIssueDate την τρέχουσα ημερομηνία και για το FromIssueDate την τιμή του ToIssueDate "μείον" 180 ημέρες.

Προτού αποθηκεύσουμε τα δεδομένα στη Βάση, καλούμαστε να τα μετασχηματίσουμε σε μορφή AKN4EU. Προς το σκοπό αυτό, αντιστοιχούμε τα πεδία των μεταδεδομένων που αντλήθηκαν από τη διεπαφή της Διαύγειας σε νέα, ακολουθώντας τις συμβάσεις μορφής και ονοματοδοσίας του προτύπου AKN4EU. Επισημαίνουμε στο σημείο αυτό ότι τα έγγραφα που αποθηκεύονται στη Βάση, αν και διατηρούν τη δομή Akoma-Ntoso, είναι της μορφής JSON και όχι XML. Το παραπάνω αποτελεί σχεδιαστική επιλογή, αφού το DBMS που χρησιμοποιήθηκε (MongoDB) δεν επιτρέπει την εισαγωγή "καθαρής" XML. Ωστόσο, η μετατροπή του JSON σε XML πραγματοποιείται κανονικά για τα δεδομένα που επιστρέφονται στο χρήστη μέσω του Command Line Interface, αλλά και κατά την εξαγωγή τους σε .akn αρχεία.

Αρχικά, δημιουργούμε τον σκελετό του αρχείου που αποτελείται από το πεδίο 'doc' μέσα στο οποίο βρίσκεται το πεδίο 'meta', όπου επιλέξαμε να εισάγουμε τα εξής πεδία:

- **Identification** Το συγκεκριμένο πεδίο περιέχει όλες τις απαραίτητες πληροφορίες σχετικά με έγγραφα κειμένων, ημερομηνίες και συντάκτες. Ακολουθώντας το μοντέλο Functional Requirements for Bibliographic Records (FRBR) διαχωρίζουμε το πεδίο σε συγκεκριμένες υποενότητες.
  - **Work**: Στο πεδίο αυτό παρέχονται πληροφορίες σχετικά με το πρωτότυπο νομοθετικό κείμενο. Συγκεκριμένα, επιλέξαμε να αποθηκεύσουμε τα στοιχεία της παλαιότερης έκδοσης της εκάστοτε πράξης όπως αυτή παρέχεται από το ιστορικό (versionlog). Για την αρχική έκδοση παρέχονται δεδομένα σχετικά με την κατάσταση της πράξης ('status') και την ημερομηνία τελευταίας τροποποίησης ('versionTimestamp'). Ταυτόχρονα, παρατίθενται πληροφορίες για τον υπεύθυνο δημιουργίας της συγκεκριμένης έκδοσης ('creator'), την χώρα έκδοσης της, αλλά και τον τύπο του εγγράφου ('documentType'). Τέλος, για τη δημιουργία μοναδικού uri χρησιμοποιούμε πληροφορίες από τη χώρα έκδοσης, τον τύπο εγγράφου και την ημερομηνία (ΥΥΥΥ-ΜΜ) τελευταίας τροποποίησης.
  - **Expression**: Ακολούθως, το πεδίο expression παρέχει μεταδεδομένα για την τελευταία έκδοση του εκάστοτε εγγράφου. Όπως και στην προηγούμενη περίπτωση, αποθηκεύονται στοιχεία σχετικά με την ημερομηνία έκδοσης ('issueDate'), την κατάσταση της πράξης, το αναγνωριστικό του συντάκτη και τη χώρα-γλώσσα σύνταξης. Επιπλέον, δημιουργούμε μοναδικό uri αξιοποιώντας τη χώρα έκδοσης, τον τύπο εγγράφου και την ημερομηνία (ΥΥΥΥ-ΜΜ-DD) τελευταίας τροποποίησης.
  - **Manifestation**: Στο παρόν πεδίο αποθηκεύονται πληροφορίες σχετικά με τη μορφή αρχείου που χρησιμοποιείται στη φυσική έκδοση του εγγράφου. Στα πλαίσια της συγκεκριμένης εφαρμογής, αποθηκεύουμε ως ημερομηνία έκδοσης την ημερομηνία άντλησης των δεδομένων, θεωρώντας ως συντάκτη τον χρήστη ο οποίος πραγματοποιεί την διαδικασία μετατροπής σε akn μορφή (AKNConversion).
  - **Item**: Τέλος, επιλέγουμε στο πεδίο Item να αποθηκεύσουμε πληροφορίες για την πρόσβαση σε μία από τις παραπάνω φυσικές εκδόσεις. Συγκεκριμένα χρησιμοποιούμε πληροφορίες μεταδεδομένων όπως παρέχονται από το Diavgeia OpenData API ('documentUrl'), παρέχοντας πρόσβαση σε αρχείο μορφής pdf της εκάστοτε νομοθετικής πράξης (Expression).
- **Lifecycle** Σε αυτό το πεδίο επιλέξαμε να αποθηκεύουμε το ιστορικό (versionlog) των πράξεων με κάθε εγγραφή να περιέχει πληροφορίες όπως το Version Id ('versionId'), το status του εκάστοτε version ('status'), το timestamp που είναι η ημερομηνία καταχώρησης ('versionTimestamp'), τα αναγνωριστικό του υπεύθυνου για τη δημιουργία της συγκεκριμένης έκδοσης χρήστη (′creator′), το description ('description') που ειναι μια σύντομη περιγραφή για το περιεχόμενο των αλλαγών στο εκάστοτε version αλλά και τον αριθμό της πρωτότυπης έκδοσης πράξης ('correctedVersionId').
- **Publication** Σε αυτό το πεδίο διακρίνουμε πληροφορίες σχετικά με την δημοσίευση της κάθε πράξης. Πιο συγκεκριμένα, κάθε πράξη διακρίνεται από ένα μοναδικό αναγνωριστικό 'ada' (Αριθμός Διαδικτυακής Ανάρτησης), με βάση το οποίο γίνεται ο έλεγχος για διπλότυπα, καθώς και ένας τύπος αναζήτησης. Ακολουθούν πεδία όπως ο αριθμός πρωτοκόλλου ('protocolNumber'), το version Id ('versionId') (δηλαδή η πιο πρόσφατη έκδοση της πράξης), το αναγνωριστικό του φορέα έκδοσης της πράξης ('organizationId'), τα αναγνωριστικά όσων υπογράφουν ('signerIds'), η λίστα κωδικών που αντιστοιχούν στις μονάδες οι οποίες εμπλέκονται στην έκδοση της συγκεκριμένης πράξης ('unitIds'), καθώς και η ημερομηνία και ώρα έκδοσης ('publishTimestamp').
- **Proprietary** Στο πεδίο αυτό περιλαμβάνονται πληροφορίες σχετικά με το θέμα ('subject') και την θεματική κατηγορία ('thematicCategoryIds') της κάθε πράξης. Ακόμα, περιέχεται η ημερομηνία και ώρα της τελευταίας τροποποίησης ('submissionTimestamp'), μια λίστα περιγραφών των συνημμένων εγγράφων ('attachments'), ο κωδικός του τύπου πράξης ('decisionTypeId'), ο αριθμός πρωτότυπης έκδοσης ('correctedVersionId'), η ένδειξη για το αν η πράξη περιλαμβάνει προσωπικά δεδομένα ('privateData') και, τέλος, η ενότητα ειδικών πεδίων ('extraFieldValues'). Το περιεχόμενο αυτής της ενότητας διαφέρει, ανάλογα με τον τύπο της πράξης.

3 Λειτουργίες CLI

Προκειμένου να διευκολυνθεί η διαδικασία άντλησης και αναζήτησης δεδομένων υλοποιήσαμε διεπαφή (CLI) που επιτρέπει στο χρήστη να εκτελέσει τις βασικές λειτουργίες της εφαρμογής.

- **Άντληση δεδομένων από Diavgeia OpenData API.** Προσδιορίζοντας συγκεκριμένο διάστημα για την ημερομηνία έκδοσης του νομοθετικού κειμένου, αλλά και το πλήθος των αποτελεσμάτων που επιθυμεί -με τους περιορισμούς που παρουσιάστηκαν σε προηγούμενη υποενότητα- ο χρήστης αποθηκεύει τα αρχεία μεταδεδομένων στην βάση δεδομένων, σε μορφή AKN4EU.
- **Αναζήτηση δεδομένων σε τοπική Βάση Δεδομένων.** Παρέχεται στον χρήστη η δυνατότητα αναζήτησης των μεταδεδομένων συγκεκριμένης νομοθετικής πράξης με πολλαπλά κριτήρια:
  - Αριθμός Διαδικτυακής Ανάρτησης (ADA)
  - Εύρος ημερομηνιών έκδοσης (FromIssueDate - ToIssueDate)
  - Αναζήτηση κειμένου σε πολλαπλά πεδία μεταδεδομένων (Θέμα πράξης, κατάσταση πράξης, κλπ)

  Η διαδικασία αναζήτησης επιστρέφει κάθε φορά όλες τις εγγραφές που ικανοποιούν τα κριτήρια ταξινομημένα κατά φθίνουσα σειρά συνάφειας.
- **Αποθήκευση εγγραφής.** Ο χρήστης έχει τη δυνατότητα να αποθηκεύσει τοπικά τα μεταδεδομένα που επιστρέφονται, σε αρχεία μορφής .akn και .json.
- **Εκκαθάριση βάσης δεδομένων.** Δίνεται, τέλος, η δυνατότητα εκκαθάρισης της τοπικής βάσης δεδομένων.

4 Τεχνικά Χαρακτηριστικά

Για την υλοποίηση χρησιμοποποιήθηκαν οι ακόλουθες τεχνολογίες:

1. **Python3**
2. **MongoDB v4.4** για την αποθήκευση των μεταδεδομένων
3. **pymongo** για τη διασύνδεση με τη βάση
4. **Flask** για τη διαχείριση του routing
5. **requests** για την εκτέλεση κλήσεων στο API
6. **PyInquirer** για την υλοποίηση διαδραστικού CLI

Στον φάκελο του project, εκτός από την αναφορά και την αντίστοιχη παρουσίαση, περιλαμβάνεται ο κώδικας που υλοποιεί τις λειτουργίες που περιγράψαμε. Ειδικότερα, εντός του φακέλου inyanga βρίσκονται:

1. *server.py*: εκκινεί τον server της εφαρμογής (εκτελείται 1ο)
2. *cli.py*: εκκινεί το CLI (python cli.py)
3. *endpoints/indlovu.py*: περιλαμβάνει τα endpoints του API μας
4. *collect.py*: συνάρτηση για την άντληση δεδομένων από το API της Διαύγειας
5. *dict2akn.py*: συνάρτηση που υλοποιεί τη μετατροπή σε AkomaNtoso μορφή
6. *docs/*: φάκελος με παραδείγματα εξαγόμενων από την εφαρμογή .akn και .json εγγράφων

Για την εγκατάσταση του συνόλου των εξαρτήσεων από pip modules μπορεί να εκτελεστεί το script *dependencies.sh*.