

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: Τεχνολογίες και προγραμματισμός στον Ιστό 2021

ΔΙΔΑΣΚΟΝΤΑΣ: Β. Ζαφείρης

ΕΡΓΑΣΙΑ 2

Ημ/νια παράδοσης: 31/05/2021

(ατομική)

Δίνεται το Web API του ηλεκτρονικού βιβλιοπωλείου [Penguin Random House](http://www.penguinrandomhouse.biz/webservices/rest/) με τεκμηρίωση διαθέσιμη στη διεύθυνση <http://www.penguinrandomhouse.biz/webservices/rest/>. Το API δίνει τη δυνατότητα σε εφαρμογές ιστού να αναζητήσουν πληροφοριακούς πόρους που διαχειρίζεται το βιβλιοπωλείο, όπως βιβλία (works), εκδόσεις βιβλίων με συγκεκριμένο ISBN (titles) καθώς και στοιχεία συγγραφέων (authors). Για κάθε πληροφοριακό πόρο, το API παρέχει δυνατότητα αναζήτησης ή εντοπισμού συγκεκριμένης πληροφοριακής οντότητας βάσει μοναδικού αναγνωριστικού. Για παράδειγμα, η αναζήτηση μεταξύ των βιβλίων που εκδίδει ή εμπορεύεται το βιβλιοπωλείο γίνεται μέσω του URI `/resources/works` συνοδευόμενο από κατάλληλες παραμέτρους ερωτημάτων. Αντίστοιχα, ο εντοπισμός συγκεκριμένου έργου βάσει του μοναδικού κωδικού του μπορεί να γίνει μέσω του URI `/resources/works/{workId}`, όπου το `{workId}` είναι μεταβλητό και λαμβάνει τιμές μεταξύ των έγκυρων αναγνωριστικών. Το API είναι διαθέσιμο στο διεύθυνση <https://reststop.randomhouse.com/>, που αποτελεί το πρώτο συνθετικό των URIs στα οποία μπορούν να απευθύνονται HTTP αιτήματα (πχ. GET <https://reststop.randomhouse.com/resources/works>).

Ζητείται η υλοποίηση μιας εφαρμογής ιστού η οποία θα παρέχει τις εξής περιπτώσεις χρήσης:

PX1. Αναζήτηση βιβλίων

Η εφαρμογή θα υλοποιεί υπηρεσία αναζήτησης βιβλίων ή συγγραφέων με χρήση του παραπάνω Web API. Συγκεκριμένα, οι χρήστες θα μπορούν να εισάγουν λέξεις κλειδιά του τίτλου ενός βιβλίου, εντοπίζοντας βιβλία που ταιριάζουν στα κριτήρια αναζήτησης.

Η υπηρεσία αναζήτησης θα είναι υλοποιημένη εξ' ολοκλήρου στον Web Browser αξιοποιώντας το Fetch API για κλήση του Web API. Η εμφάνιση των αποτελεσμάτων αναζήτησης θα μπορεί να γίνει παράγοντας δυναμικά HTML κώδικα και προσαρτώντας τον σε κατάλληλα σημεία της σελίδας αναζήτησης, με χρήση ιδιοτήτων του DOM API. Η παραγωγή HTML κώδικα μπορεί να γίνει, επίσης, με αξιοποίηση της βιβλιοθήκης [Handlebars](#) και κατάλληλων HTML templates.

ΠΧ2. Αποθήκευση αγαπημένων βιβλίων

Ο χρήστης θα έχει τη δυνατότητα αποθήκευσης, ενός ή περισσότερων βιβλίων από τα αποτελέσματα αναζήτησης, στη λίστα αγαπημένων βιβλίων του. Η αποθήκευση θα γίνεται με επιλογή κατάλληλου κουμπιού/εικονιδίου, το οποίο θα είναι ξεχωριστό για κάθε βιβλίο και θα εμφανίζεται μαζί με τα στοιχεία του στα αποτελέσματα αναζήτησης. Η επιτυχής αποθήκευση ενός βιβλίου στη λίστα αγαπημένων βιβλίων θα ενεργοποιεί κουμπί/εικονίδιο για αναίρεση της ενέργειας (διαγραφή από τη λίστα αγαπημένων βιβλίων). Για κάθε βιβλίο θα αποθηκεύονται τουλάχιστον ο τίτλος του, ο συγγραφέας καθώς και το μοναδικό id του που παρέχεται από την υπηρεσία Penguin Random House. Δεν θα επιτρέπεται η διπλή καταχώριση ενός βιβλίου στη λίστα αγαπημένων βιβλίων. Σε περίπτωση που υπάρχει ήδη, θα εμφανίζεται κατάλληλο μήνυμα στο χρήστη.

Η διαχείριση της λίστας αγαπημένων βιβλίων του χρήστη θα γίνεται από κατάλληλη υπηρεσία ιστού, την οποία καλείστε να υλοποιήσετε στην πλατφόρμα Node.js. Η προσθήκη/διαγραφή βιβλίων θα γίνεται μέσω κατάλληλων HTTP αιτημάτων, τα οποία θα υποβάλλονται από τον Web Browser μέσω του Fetch API, κατά το πάτημα των αντίστοιχων κουμπιών. Η υλοποίηση του διακομιστή ιστού και των υπηρεσιών αποθήκευσης/διαγραφής βιβλίων θα γίνει με χρήση του Node.js API και του [Express framework](#). Τα δεδομένα θα υποβάλλονται στο διακομιστή σε μορφή JSON. Ο διακομιστής θα τα μετατρέπει σε αντικείμενα και θα τα αποθηκεύει στην τοπική βάση δεδομένων.

ΠΧ3. Διαχείριση λίστας αγαπημένων βιβλίων

Ο χρήστης θα έχει δυνατότητα πρόσβασης στη λίστα των αγαπημένων του βιβλίων, μέσω κατάλληλου συνδέσμου από την σελίδα αναζήτησης. Η λίστα αγαπημένων βιβλίων θα εμφανίζεται σε ξεχωριστή σελίδα, όπου θα παρέχεται δυνατότητα επεξεργασίας ή διαγραφής των στοιχείων ενός βιβλίου μέσω κατάλληλων κουμπιών. Επιπλέον, θα υπάρχει πεδίο αναζήτησης για φιλτράρισμα της λίστας των αγαπημένων βιβλίων βάσει λέξεων κλειδιών. Το φιλτράρισμα θα ενεργοποιείται αυτόματα όσο πληκτρολογεί ο χρήστης, μετά από μια σύντομη περίοδο αδράνειας (πχ. 0.5 sec).

Η επιλογή της επεξεργασίας ενός βιβλίου θα εμφανίζει σε ξεχωριστή σελίδα, μια φόρμα με τα στοιχεία του βιβλίου. Ο χρήστης θα έχει δυνατότητα τροποποίησης των πληροφοριών τίτλου και συγγραφέων του βιβλίου (όχι το αναγνωριστικό του). Επίσης, θα μπορεί να προσθέσει μια σύντομη κριτική. Τα παραπάνω στοιχεία θα εμφανίζονται με κατάλληλο τρόπο στην σελίδα που παρουσιάζει τη λίστα αγαπημένων βιβλίων.

Η δημιουργία της σελίδας με τη λίστα αγαπημένων βιβλίων μπορεί να γίνει δυναμικά είτε από την πλευρά του Browser ή του Διακομιστή με χρήση Handlebars templates. Η υποβολή της φόρμας θα γίνεται με κατάλληλο κουμπί και η επεξεργασία των στοιχείων της θα γίνεται από το τμήμα διακομιστή της εφαρμογής.

Περιορισμοί/λεπτομέρειες υλοποίησης

- **Αρχιτεκτονική:** η εφαρμογή θα βασίζεται στην αρχιτεκτονική Πελάτη-Διακομιστή. Το τμήμα Πελάτη θα υλοποιηθεί ως μια σειρά από HTML σελίδες που θα ενσωματώνουν κώδικα JavaScript. Το τμήμα διακομιστή θα υλοποιηθεί ως μια Node.js εφαρμογή, η οποία θα καλείται από τον Πελάτη μέσω κατάλληλων υπερσυνδέσμων ή HTTP αιτημάτων με χρήση του Fetch API.
- **Βιβλιοθήκες/frameworks:** Επιτρέπεται η χρήση του Fetch API και της βιβλιοθήκης Handlebars για την υλοποίηση του Πελάτη. Η υλοποίηση του διακομιστή μπορεί να βασιστεί στο Node.js API, στο Express framework και στην βιβλιοθήκη Handlebars. Δεν επιτρέπεται η χρήση άλλων εξωτερικών βιβλιοθηκών/frameworks.
- **Βάση δεδομένων:** δεν απαιτείται η αποθήκευση των δεδομένων σε πραγματική βάση δεδομένων. Τα δεδομένα θα διατηρούνται μόνο κατά τη διάρκεια λειτουργίας του εξυπηρετητή και θα αποθηκεύονται σε κατάλληλες λίστες, τις οποίες θα διαχειρίζονται αντικείμενα DAO (Data Access Objects). Τα DAO αντικείμενα θα υποστηρίζουν μεθόδους αποθήκευσης, αναζήτησης και διαγραφής βιβλίων. Σε περίπτωση που ενσωματωθεί πραγματική βάση δεδομένων και γίνει κατάλληλη υλοποίηση των DAO αντικειμένων θα υπάρξει bonus 10%.
- **Debugging:** Για πιο αποδοτική δοκιμή και εκσφαλμάτωση της εφαρμογής, προτείνεται η χρήση του εργαλείου [nodemon](#). Το nodemon ξεκινά τον διακομιστή της εφαρμογής και φορτώνει τη νέα έκδοση του κώδικά του, μετά από κάθε αλλαγή.
- **Web API:** η λειτουργικότητα του τμήματος διακομιστή της εφαρμογής σας θα παρέχεται με τη μορφή υπηρεσιών ιστού. Η σχεδίαση των υπηρεσιών ιστού θα πρέπει να γίνει με κατάλληλη επιλογή URIs και σωστή χρήση των HTTP methods και HTTP κωδικών απόκρισης.
- **Παρουσίαση:** Η παρουσίαση της εφαρμογής θα βασιστεί σε σημασιολογικά στοιχεία HTML και κατάλληλους κανόνες CSS.

Βαθμολόγηση

Η κατανομή της βαθμολογίας, αναλύεται ως εξής:

- ΠΧ1 (35%)
- ΠΧ2 (25%)
- ΠΧ3 (20%)
- Χρήση σημασιολογικών στοιχείων HTML και κανόνων CSS (10%)
- Σχεδίαση υπηρεσιών με κατάλληλα URIs και σωστή χρήση των HTTP methods και κωδικών απόκρισης (5%)
- Καλή σχεδίαση κώδικα με χρήση συναρτήσεων και κλάσεων (5%)
- Ενσωμάτωση πραγματικής βάσης δεδομένων (10%)

Καλή δουλειά!