

Εργασία

Περιγραφή:

Περιγραφή

Υποθέστε το εξής σενάριο: Σας ζητούν να υλοποιήσετε μία δικτυακή εφαρμογή η οποία να επιτρέπει στους χρήστες της να αναζητούν βιβλία. Συγκεκριμένα θέλουν 2 λειτουργίες

- **Αναζήτηση βιβλίων με τίτλο:** Ο χρήστης μπορεί να επιλέξει ένα τίτλο βιβλίου και το σύστημα θα πρέπει να αναζητά τον τίτλο αυτό και να επιστρέφει το αντίστοιχο βιβλίο. Η υλοποίηση αυτής της λειτουργικότητας θα γίνεται μέσω του resource /getBooksByTitle στο server το οποίο αναζητά μία παράμετρο “title” και επιστρέφει ένα xml με τα στοιχεία ενός βιβλίου με βάση το xml: <https://msdn.microsoft.com/en-us/library/ms762271%28v=vs.85%29.aspx> ή το ισοδύναμο json
- **Αναζήτηση βιβλίων με λέξη κλειδί:** Ο χρήστης μπορεί να καταχωρήσει σε ένα textbox οποιαδήποτε λέξη θέλει και το σύστημα θα πρέπει να αναζητά τη λέξη αυτή στα πεδία: Author, Title, Genre, Description, των στοιχείων των βιβλίων. Στη συνέχεια θα επιτρέπει όλα τα βιβλία που περιέχουν αυτή τη λέξη κλειδί. Η υλοποίηση αυτής της λειτουργικότητας θα γίνεται μέσω του resource /getBooksByKeyword στο server το οποίο αναζητά μία παράμετρο “keyword” και επιστρέφει ένα xml με μία λίστα βιβλίων όπως αυτή: <https://msdn.microsoft.com/en-us/library/ms762271%28v=vs.85%29.aspx> ή το ισοδύναμο json

Τα στοιχεία θα βρίσκονται σε μία MySQL βάση δεδομένων σε ένα server στην οποία τα 2 παραπάνω resources θα έχουν πρόσβαση. Για τη δημιουργία της βάσης χρησιμοποιήστε την παρακάτω SQL:

```
CREATE TABLE `books` (  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `author` VARCHAR( 25 ) NOT NULL ,  
  `title` VARCHAR( 40 ) NOT NULL ,  
  `genre` VARCHAR( 20 ) NOT NULL ,  
  `price` FLOAT NOT NULL ,  
  `publish_date` DATE NOT NULL ,  
  `description` TEXT NOT NULL  
) ENGINE = MYISAM ;
```

Η παρουσίαση και διαχείριση όλων των λειτουργιών θα γίνεται με μία μόνο “frontend.html” σελίδα η οποία θα παίρνει δεδομένα με **AJAX calls** προς τα 2 server resources.

Αρχικά η σελίδα θα έχει ένα dropdown list με δύο επιλογές: “Find Book By Title” και “Find Book By Keyword”. Κάθε επιλογή θα εμφανίζει από κάτω το κατάλληλο πεδίο (input) ώστε ο χρήστης να εκτελέσει την αντίστοιχη λειτουργία. Στην πρώτη περίπτωση το input θα είναι μία νέα dropdown list με όλους τους διαθέσιμους τίτλους από τη βάση. Στη δεύτερη θα είναι ένα textbox όπου ο χρήστης θα μπορεί να βάλει ένα keyword.

Τα αποτελέσματα θα εμφανίζονται δομημένα και για κάθε βιβλίο θα αναφέρονται όλα τα στοιχεία που περιέχονται στο xml.

Αν και στην πραγματικότητα τα δεδομένα με τα οποία θα δουλέψετε θα είναι πολύ μικρά σε όγκο, θα πρέπει να υποθέσετε ότι η βάση δεδομένων μπορεί να μεγαλώσει απεριόριστα, επομένως θα πρέπει να είστε προσεκτικοί στα calls που κάνετε αλλά και στο πώς χρησιμοποιείτε το **API του server**.

Θα πρέπει η υλοποίηση να βασίζεται στο **MVC architectural pattern** ενώ στο server μπορείτε να χρησιμοποιήσετε όποια τεχνολογία προτιμάτε από τις php, node.js, angular.

Για τη μετατροπή των δεδομένων του MySQL server σε XML μπορείτε να βρείτε αρκετές επιλογές ανάλομα με τη server-side τεχνολογία που θα χρησιμοποιήσετε.

Οδηγίες

- Ο κώδικας θα πρέπει να τρέχει τουλάχιστον σε Firefox και Chrome.
- Θα πρέπει να «στήσετε» ένα web και έναν MySQL server
- Η εργασία είναι ατομική.
- Η βάση δεδομένων πρέπει να είναι ακριβώς όπως αναφέρεται πιο πάνω (SQL Query)
- Τα παραδοτέα της εργασίας είναι:
 - ο κώδικας
 - Μία αναφορά που θα εξηγεί την αρχιτεκτονική της υλοποίησής σας με έμφαση **στο MVC pattern**
- ΠΡΟΣΟΧΗ! Οι οδηγίες και όροι που αναφέρονται στον Οδηγό Υποβολής Εργασιών θα τηρηθούν αυστηρά. Μη συμμόρφωση ακόμα και στη μικρότερη λεπτομέρεια θα συνεπάγεται βαθμούς ποινή ή μη αποδοχή της εργασίας.
- Η παράδοση μετά την προθεσμία επιτρέπεται με κόστος -2 βαθμούς για κάθε 24ωρο καθυστέρησης.

Μέγιστη βαθμολογία:

10

Ημερομηνία έναρξης:

17-05-2016 06:06:21