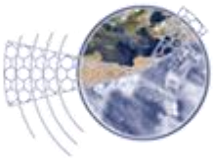
TEI Κρήτης
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ



ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός πληροφοριακού συστήματος για την διαχείριση επαφών/φωτογραφιών

(σε PHP framework με χρήση REST)

ΣΤΑΜΑΤΑΚΗΣ ΝΙΚΟΛΑΟΣ (ΑΜ 2299)

Επιβλέπων καθηγητής : κος Παπαδάκης Νικόλαος

Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους όσους στάθηκαν στο πλευρό μου όλο αυτό το χρονικό διάστημα. Πρώτα απ' όλα τις οικογένειές μας που μας στάθηκαν οικονομικά και ψυχολογικά παρέχοντάς μας την κατάλληλη στήριξη στις σπουδές μου. Θα ήθελα να ευχαριστήσω επίσης όλους μου τους φίλους, συνάδελφους, καθηγητές που με τη βοήθεια τους έφτασα μέχρι εδώ.

Τέλος θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου και επιβλέπων κ. Παπαδάκη Νικόλαο που με τις γνώσεις του και την βοήθειά του υλοποιήθηκε με επιτυχία η παρούσα πτυχιακή.

Εισαγωγή

Η παρούσα πτυχιακή εργασία παρουσιάζει μια εφαρμογή διαχείρησης επαφών όπου ο χρήστης μπορεί να αποθηκεύει βασικές πληροφορίες για άτομα που γνωρίζει, είτε σε προσωπικό, είτε σε επαγγελματικό επίπεδο.

Η εφαρμογή δημιουργήθηκε με τη χρήση ενός framework γραμμένο σε γλώσσα PHP, το CodeIgniter, και με τη χρήση βάσης δεδομένων MySQL. Δύο γλώσσες οι οποίες είναι οι επικρατέστερες στο διαδίκτυο και χαρακτηρίζονται από την ταχύτητα και το πόσο «ελαφριές» είναι.

Επίσης, στην εφαρμογή προστέθηκε ένα RESTful API όπου δίνει τη δυνατότητα στον χρήστη να εξάγει τα δεδομένα της εφαρμογής σε αρχείο XML, δίνοντας του έτσι τη δυνατότητα να μπορεί να τα μεταφέρει εκτός εφαρμογής.

ΠΕΡΙΕΧΟΜΕΝΑ

3.ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

A.Το Internet

Το Διαδίκτυο (αγγλ. Internet) είναι παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών, οι οποίοι χρησιμοποιούν καθιερωμένη ομάδα πρωτοκόλλων, η οποία συχνά αποκαλείται "TCP/IP" (αν και αυτή δεν χρησιμοποιείται από όλες τις υπηρεσίες του Διαδικτύου) για να εξυπηρετεί εκατομμύρια χρηστών καθημερινά σε ολόκληρο τον κόσμο. Οι διασυνδεδεμένοι ηλεκτρονικοί υπολογιστές ανά τον κόσμο, οι οποίοι βρίσκονται σε ένα κοινό δίκτυο επικοινωνίας, ανταλλάσσουν μηνύματα (πακέτα) με τη χρήση διαφόρων πρωτοκόλλων (τυποποιημένοι κανόνες επικοινωνίας), τα οποία υλοποιούνται σε επίπεδο υλικού και λογισμικού. Το κοινό αυτό δίκτυο καλείται Διαδίκτυο.

Το Διαδίκτυο είναι ένα επικοινωνιακό δίκτυο που επιτρέπει την ανταλλαγή δεδομένων μεταξύ οποιουδήποτε διασυνδεδεμένου υπολογιστή. Η τεχνολογία του είναι κυρίως βασισμένη στην διασύνδεση επιμέρους δικτύων ανά τον κόσμο και πολυάριθμα πρωτόκολλα επικοινωνίας. Στην πιο εξειδικευμένη και περισσότερο χρησιμοποιούμενη μορφή του, με τον όρο Διαδίκτυο, περιγράφεται το παγκόσμιο πλέγμα διασυνδεδεμένων υπολογιστών και των υπηρεσιών και πληροφοριών που παρέχει στους χρήστες του. Το Διαδίκτυο χρησιμοποιεί μεταγωγή πακέτων και τη στοίβα πρωτοκόλλων]. Σήμερα, ο όρος διαδίκτυο κατέληξε στο να αναφέρεται στο παγκόσμιο αυτό δίκτυο. Για να ξεχωρίζει, το παγκόσμιο αυτό δίκτυο γράφεται με κεφαλαίο το αρχικό "Δ". Η τεχνική της διασύνδεσης δικτύων μέσω μεταγωγής πακέτων και της στοίβας πρωτοκόλλων ονομάζεται Διαδικτύωση.

Το Διαδίκτυο, σε συνδυασμό με την ολοένα αναπτυσσόμενη ψηφιακή τεχνολογία, έχει δημιουργήσει μία τεράστια αγορά γνώσεων/πληροφοριών. Παραδοσιακές μορφές τέχνης (όπως για παράδειγμα ο κινηματογράφος και η μουσική) μέσω της ψηφιακής τεχνολογίας παίρνουν την ίδια μορφή (αρχείων δεδομένων) με αντικείμενα που εκ πρώτης όψεως είναι εντελώς διαφορετικά (όπως για παράδειγμα η ιατρική επιστήμη ή κάποιο πρόγραμμα λογισμικού). Παρατηρείται λοιπόν μία συγκέντρωση γνώσης ή, αν

είναι δυνατό να λεχτεί, πολιτιστικής κληρονομιάς, που σχετίζεται άμεσα με το Ίντερνετ. Το μεγάλο ερώτημα που προκύπτει πλέον είναι το "ποιος θα διοικήσει, ποιος θα ελέγξει την γνώση αυτή".

Από τη στιγμή που το Διαδίκτυο είναι ένα δίκτυο συνδεδεμένων υπολογιστών, κάθε χρήστης έχει την δυνατότητα να μοιραστεί πληροφορίες με άλλους χρήστες γενόμενος, πολλές φορές, ο ίδιος δημιουργός και πάροχος των πληροφοριών αυτών. Δεν υπάρχει άμεσος έλεγχος των πληροφοριών που "ανεβαίνουν" στο Διαδίκτυο από κάποιον ιεραρχικά ανώτερο χρήστη ή οργανισμό. Το θέμα της μη ιεραρχημένης πληροφορίας, όμως, τίθεται υπό αμφισβήτηση. Ο όγκος της πληροφορίας στο Διαδίκτυο είναι πράγματι μεγάλος. Παρ' όλα αυτά, υπάρχουν πληροφορίες ευκολότερα και δυσκολότερα προσβάσιμες από τον χρήστη.

Το Διαδίκτυο κατέστησε εφικτή τη συγκέντρωση μεγάλου όγκου πληροφοριών και επηρέασε σημαντικά τον τρόπο διάθεσής τους, δεν συμβαίνει όμως στον ίδιο βαθμό το ίδιο και στον τρόπο παραγωγής αυτών. Για παράδειγμα, ο τρόπος παραγωγής μιας κινηματογραφικής ταινίας δεν έχει επηρεαστεί σημαντικά από την ύπαρξη του Διαδικτύου, ανεξάρτητα από το αν έχει επηρεαστεί ή όχι από την ψηφιακή τεχνολογία. Παρ' όλα αυτά, και σύμφωνα με την "ιντερνετοφιλική" προσέγγιση, το Διαδίκτυο ασκεί μεγάλη επίδραση στην διαδικασία παραγωγής δημοσιογραφικών προϊόντων. Η δημιουργία της είδησης παύει να είναι πλέον μονοπώλιο λίγων, αφού ο κάθε χρήστης μπορεί εάν το επιθυμεί να δημιουργήσει πληροφορία ανά πάσα στιγμή. Το πιο τρανταχτό παράδειγμα της επίδρασης αυτής είναι τα ιστολόγια (blogs), όπου μπορεί κανείς να εκφέρει απόψεις και να σχολιάσει γεγονότα πάσης φύσεως (βλ. δημοσιογραφία στον ιστό και δημοσιογραφία των πολιτών). Ως αποτέλεσμα της επιρροής αυτής του Ίντερνετ στη παραγωγή ειδήσεων τα όρια μεταξύ ενός απλού χρήστη του διαδικτύου και ενός επαγγελματία δημοσιογράφου γίνονται περισσότερο δυσδιάκριτα. Αυτό με τη σειρά του οδηγεί στην ανάγκη για επαναπροσδιορισμό της έννοιας της δημοσιογραφίας καθώς και της απαραίτητης εκπαίδευσης των δημοσιογράφων.

Η ανάγκη για τον επαναπροσδιορισμό της δημοσιογραφίας, όμως, δεν είναι τόσο μεγάλη σύμφωνα με τους υποστηριχτές της "αντι-πλουραλιστικής" προσέγγισης, καθώς θεωρούν πως το Ίντερνετ δεν μπορεί να ασκήσει ουσιαστική επίδραση στην επικοινωνία γενικότερα και στην δημοσιογραφία ειδικότερα.

Επίσης, λόγω της μεγάλης συγκέντρωσης γνώσης στο Διαδίκτυο, η έννοια της κοινωνικής ισότητας παίρνει και πάλι μεγάλη σημασία. Το χάσμα ανάμεσα σε πληροφοριακά πλούσιους και πληροφοριακά φτωχούς θα διευρύνεται όσο αυξάνεται η συγκέντρωση της γνώσης αυτής. Το παραπάνω αποτελεί ακόμα έναν λόγο που κάνει πιο επιτακτική την ανάγκη για διερεύνηση του αρχικού ερωτήματος "ποιος θα ελέγξει τη γνώση αυτή".

b. Apache Server

Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται ένα ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X. Κυκλοφόρησε υπό την άδεια λογισμικού Apache και είναι λογισμικό ανοιχτού κώδικα. Συντηρείται από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων π.χ. Oracle, MySQL.

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET. Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους.

Χαρακτηριστικά και λειτουργίες του Apache HTTP

Ο Apache διαθέτει ποικιλία χαρακτηριστικών και μπορεί να υποστηρίξει μια μεγάλη γκάμα εφαρμογών με τις οποίες και συνεργάζεται. Οι δυνατότητες του προγράμματος αυτού καθαυτού και τα χαρακτηριστικά του δεν είναι και τόσο πολλά. Ένα από τα βασικότερα χαρακτηριστικά του όμως, το οποίο και του δίνει μεγάλες δυνατότητες, είναι ότι μπορεί να προσαρμόσει επάνω του πολλές προσθήκες προγραμμάτων (modules), τα οποία με τη σειρά τους παρέχουν διαφορετικές λειτουργίες. Μερικά από τα πιο γνωστά modules του Apache HTTP είναι τα modules πιστοποίησης, όπως για παράδειγμα τα mod_access, mod_auth, mod_digest κ.λπ. Παρέχει επίσης SSL σε TLS μέσω των (mod_ssl), και proxy module (mod_proxy), πραγματοποιεί ανακατευθύνσεις διευθύνσεων (URL rewrites) μέσω του mod_rewrite, καταγραφές συνδέσεων μέσω του mod_log_config, συμπίεση αρχείων μέσω του mod_gzip και πολλά άλλα modules τα οποία διατίθενται είτε από το Apache Software Foundation, είτε από τρίτες εταιρίες λογισμικού.

Ένα άλλο χαρακτηριστικό – δυνατότητα του Apache HTTP, όπως έχω αναφέρω πιο πάνω, είναι ότι μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα. Ο Apache HTTP υποστηρίζει επίσης αρκετές διάσημες εφαρμογές και γλώσσες προγραμματισμού όπως MySQL, PHP, Perl, Python κ.λπ.

Αυτά είναι μερικά από τα χαρακτηριστικά και τις λειτουργίες του που κάνουν τον Apache τον πιο δημοφιλή Web Server από το 1996 έως τις μέρες μας. Περισσότερο από το 50% των ιστοχώρων του παγκόσμιου ιστού, χρησιμοποιεί τον Apache ως εξυπηρετητή. Το υπόλοιπο ποσοστό καλύπτουν αντίστοιχα προγράμματα, όπως το Microsoft Internet Information Services (IIS), ο Sun Java System Web Server, ο Zeus Web Server κα.

XAMPP

Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl.

Το XAMPP είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

- X (αναφέρεται στο «cross-platform» που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας)
- Apache HTTP εξυπηρετητής
- MySQL
- PHP
- Perl

Το XAMPP είναι ένα ελεύθερο λογισμικό το οποίο περιέχει ένα εξυπηρετητή ιστοσελίδων το οποίο μπορεί να εξυπηρετεί και δυναμικές ιστοσελίδες τεχνολογίας PHP/MySQL. Είναι ανεξάρτητο πλατφόρμας και τρέχει σε Microsoft Windows, Linux, Solaris, and Mac OS X και χρησιμοποιείται ως πλατφόρμα για την σχεδίαση και ανάπτυξη ιστοσελίδων με την τεχνολογίες όπως PHP, JSP και Servlets.

Το πρόγραμμα εκδίδεται σύμφωνα με τους όρους της GNU General Public License και λειτουργεί ως ένας ελεύθερος web server ικανός να εξυπηρετεί δυναμικές σελίδες.

Αυτή τη στιγμή υπάρχουν τέσσερις XAMPP διανομές:

- XAMPP για Linux
- XAMPP για τα Windows
- XAMPP για MacOSX
- XAMPP για το Solaris

Αυτό το λογισμικό είναι χρήσιμο ενώ δημιουργείτε δυναμικές ιστοσελίδες που χρησιμοποιούν γλώσσες προγραμματισμού όπως η PHP, JSP, Servlets.

Το XAMPP είναι μια εύκολη στην εγκατάσταση διανομή Apacheπου περιέχει MySQL, PHP και Perl. Το XAMPP είναι πραγματικά πολύ εύκολο στην εγκατάσταση και στη χρήση.

Φιλοσοφία

Η φιλοσοφία πίσω από XAMPP είναι η οικοδομήση μιας εύκολης στην εγκατάσταση διανομής για τους προγραμματιστές ώστε να μπουν στον κόσμο του Apache. Για να είναι βολικό για τους προγραμματιστές το XAMPP έχει ρυθμιστεί με όλα τα χαρακτηριστικά ενεργοποιημένα.

Η προεπιλεγμένη ρύθμιση δεν είναι καλό από άποψη ασφαλείας και δεν είναι αρκετά ασφαλές για ένα περιβάλλον παραγωγής

Προδιαγραφές

Το XAMPP απαιτεί μόνο ένα zip, tar, 7z, ή ένα αρχείο exe για να το κατεβάσετε και να τρέξει, και λίγο ή καθόλου διαμόρφωση των διαφόρων στοιχείων που συνθέτουν το webserver είναι απαραίτητη. Το XAMPP ενημερώνεται τακτικά για να ενσωματώσει τις τελευταίες εκδόσεις του Apache / MySQL / PHP και Perl. Επίσης, έρχεται με μια σειρά άλλων ενοτήτων, συμπεριλαμβανομένων OpenSSL και το phpMyAdmin.

Η εγκατάσταση του XAMPP χρειάζεται λιγότερο χρόνο από την εγκατάσταση κάθε στοιχείου του ξεχωριστά. Αυτοτελείς, πολλές εμφανίσεις του XAMPP μπορεί να υπάρχει σε έναν υπολογιστή, και κάθε δεδομένη περίσταση μπορούν να αντιγραφούν από έναν υπολογιστή σε άλλο.

Προσφέρεται σε δύο πλήρη, κανονική έκδοση και μια μικρότερη έκδοση.

Χρήση

Επισήμως, οι σχεδιαστές του XAMPP το προορίζαν αρχικά μόνο για χρήση μόνο ως εργαλείο ανάπτυξης, για να επιτρέπει στους σχεδιαστές και τους προγραμματιστές ιστοσελίδων να ελέγχουν την εργασία τους στους δικούς τους υπολογιστές χωρίς πρόσβαση στο Internet. Για να γίνει αυτό όσο το δυνατόν ευκολότερο, πολλά και σημαντικά χαρακτηριστικά ασφαλείας είναι απενεργοποιημένα από προεπιλογή. Στην πράξη, ωστόσο, το XAMPP μερικές φορές χρησιμοποιείται για να εξυπηρετήσει ιστοσελίδες στο World Wide Web. Ένα ειδικό εργαλείο παρέχεται ώστε να προστατέψετε με κωδικό πρόσβασης τα πιο σημαντικά μέρη του πακέτου.

Το XAMPP παρέχει επίσης υποστήριξη για τη δημιουργία και χειρισμό βάσεων δεδομένων MySQL και SQLite, μεταξύ άλλων.

Μόλις το XAMPP εγκατασταθεί, μπορείτε να τη χρησιμοποιήτε τον localhost σας σαν έναν απομακρυσμένο υπολογιστή με σύνδεση χρησιμοποιώντας ένα FTP client. Χρησιμοποιώντας ένα πρόγραμμα όπως το FileZilla έχει πολλά πλεονεκτήματα κατά την εγκατάσταση ενός συστήματος περιεχομένου διαχείρισης (CMS) όπως το Joomla. Μπορείτε επίσης να συνδεθείτε στον localhost μέσω του FTP με το πρόγραμμα επεξεργασίας HTML.

Η προεπιλογή χρήστη FTP είναι "newuser" και ο προεπιλεγμένος κωδικός πρόσβασης είναι "wampp".

Η προεπιλογή χρήστηMySQL είναι "root", ενώ δεν υπάρχει καμία προεπιλογή για τον κωδικό πρόσβασης.

c. PHP

i. Η γλώσσα

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Τι είναι η PHP;

Η PHP, όπως αναφέρεται και στην ιστοσελίδα της (php.net) είναι μία πολύ διαδεδομένη γλώσσα προγραμματισμού γενικής χρήσης, που είναι κατάλληλη για προγραμματισμό διαδικτυακών εφαρμογών και μπορεί να εισαχθεί σε HTML.

Είναι μια γλώσσα προγραμματισμού ειδικά για την κατασκευή δυναμικών ιστοσελίδων, σελίδων δηλαδή που μπορούν να τροποποιούνται από το διαχειριστή τους online ή να διαφοροποιούνται, ανάλογα με τα χαρακτηριστικά του χρήστη που τις προβάλει, όπως για παράδειγμα, το λειτουργικό του σύστημα, η διεύθυνση IP του κ.ά.

Χρησιμοποιείται όχι για την αισθητική διαμόρφωση μιας σελίδας, αλλά για τον χειρισμό των λειτουργιών και εργασιών που θα διεκπεραιώνει. Συνεπώς, ο κώδικας που γράφεται για μια ιστοσελίδα σε γλώσσα PHP δεν γίνεται άμεσα αντιληπτός αλλά μετά από την επέμβαση του χρήστη στην ιστοσελίδα.

Για να γίνει αυτό κατανοητό: η PHP χρησιμοποιείται ευρέως για τον χειρισμό ιστοσελίδων με δυνατότητες όπως η εγγραφή χρηστών (user registration), τα fora κ.ά. Λειτουργεί με την βοήθεια της HTML και πλέον και με την XHTML (νέα αναθεωρημένη έκδοση της HTML).

Σε συνδυασμό και με την MySQL μπορεί να χρησιμοποιηθεί κάλλιστα για την διαχείριση δεδομένων μέσα σε βάσεις. Για παράδειγμα, σε μια ιστοσελίδα που είναι απαραίτητη η εγγραφή των χρηστών, η PHP μπορεί να αποθηκεύει τα ονόματα και τους κωδικούς τους σε μια βάση δεδομένων.

Επεκτάσεις αρχείων και διακομιστές

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωτίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS X υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου (mod_php) ή με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία CGI ή FCGI ή με την έλευση της php5.4 υποστηρίζονται η εκτέλεση σε πολυάσχολους ιστοχώρους, FastCGI Process Manager (FPM).

Εναλλακτικός τρόπος εκτέλεσης ιστοσελίδων χωρίς χρονοβόρες διαδικασίες

Ο συνδυασμός Linux/Apache/PHP/MySQL, που είναι η πιο δημοφιλής πλατφόρμα εκτέλεσης ιστοσελίδων είναι γνωστός και με το ακρωνύμιο LAMP. Παρόμοια, ο συνδυασμός */Apache/PHP/MySQL ονομάζεται *AMP, όπου το πρώτο αρχικό αντιστοιχεί στην πλατφόρμα, στην οποία εγκαθίστανται ο Apache, η PHP και η MySQL (π.χ. Windows, Mac OS X).

Ο LAMP συνήθως εγκαθίσταται και ρυθμίζεται στο Linux με τη βοήθεια του διαχειριστή πακέτων της εκάστοτε διανομής. Στην περίπτωση άλλων λειτουργικών συστημάτων, επειδή το κατέβασμα και η ρύθμιση των ξεχωριστών προγραμμάτων μπορεί να είναι πολύπλοκη, υπάρχουν έτοιμα πακέτα προς εγκατάσταση, όπως το XAMPP και το WAMP για τα Windows και το MAMP για το Mac OS X.

Ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

d. MVC Model

Model-View-Controller (MVC) είναι ένα πρότυπο λογισμικού για την υλοποίηση διεπαφών χρήστη. Διαιρεί μια δεδομένη εφαρμογή λογισμικού σε τρία αλληλένδετα μέρη, έτσι ώστε να διαχωρίζονται σε εσωτερικές αναπαραστάσεις των πληροφοριών από τους τρόπους που οι πληροφορίες παρουσιάζονται ή γίνονται αποδεκτές από το χρήστη. Η κεντρική συνιστώσα, το μοντέλο(Model), αποτελείται από τα δεδομένα εφαρμογής, τους κανόνες των επιχειρήσεων, τη λογική και τις λειτουργίες. Μια προβολή (View) μπορεί να είναι οποιαδήποτε παράσταση εξόδου των πληροφοριών, όπως ένα γράφημα ή διάγραμμα. Πολλαπλές προβολές των ίδιων πληροφοριών είναι δυνατή, όπως για παράδειγμα ένα γράφημα ράβδων για τη διαχείριση και την προβολή πινάκων για τους λογιστές. Το τρίτο μέρος, ο ελεγκτής (Controller), δέχεται είσοδο και το μετατρέπει σε εντολές για το μοντέλο ή την προβολή.

Αλληλεπιδράσεις στοιχείων

Εκτός από τη διαίρεση της εφαρμογής σε τρία είδη των συστατικών, ο σχεδιασμός Model-View-Controller (MVC) καθορίζει τις μεταξύ τους αλληλεπιδράσεις.

Πιο αναλυτικά:

- Ένα μοντέλο (Model) ειδοποιεί τις συνδεδεμένες προβολές(Views) και των ελεγκτών της (Controllers), όταν έχει υπάρξει μια αλλαγή στην κατάσταση τους. Η κοινοποίηση αυτή επιτρέπει στα Views να ενημερώνουν την παρουσίασή τους, καθώς και τους ελεγκτές να αλλάξουν το διαθέσιμο σύνολο των εντολών. Σε ορισμένες περιπτώσεις, μια εφαρμογή MVC μπορεί να είναι "παθητική", έτσι ώστε άλλα στοιχεία να ενεργοποιήσουν το μοντέλο για ενημερώσεις αντί να κοινοποιηθεί.

- Μια προβολή, ενημερώνεται από τον Controller για όλες τις πληροφορίες που χρειάζεται για την παραγωγή μιας παράστασης εξόδου στο χρήστη. Μπορεί επίσης να παρέχει γενικούς μηχανισμούς για να ενημερώσει τον controller για τις εισόδους (inputs) του χρήστη.

- Ένας ελεγκτής μπορεί να στείλει εντολές στο μοντέλο για να ενημερώνει την κατάσταση του μοντέλου (π.χ., την επεξεργασία ενός εγγράφου). Μπορεί επίσης να στέλνει εντολές στη σχετιζόμενη με αυτό προβολή του (View) να αλλάξει την παρουσίαση της προβολής του μοντέλου (π.χ., με κύλιση σε ένα έγγραφο).

Χρήση σε εφαρμογές web

Αν και αρχικά αναπτύχθηκε για desktop computing, το Model-View-Controller έχει υιοθετηθεί ευρέως ως μια αρχιτεκτονική για τις World Wide Web εφαρμογές σε όλες τις μεγάλες γλώσσες προγραμματισμού. Αρκετά εμπορεύσιμα και μη εμπορεύσιμα frameworks έχουν δημιουργηθεί που επιβάλλουν το συγκεκριμένο μοτίβο. Τα frameworks αυτά διαφέρουν ως προς τις ερμηνείες τους, κυρίως με τον τρόπο που οι ευθύνες του MVC κατανέμονται μεταξύ του πελάτη και του διακομιστή.

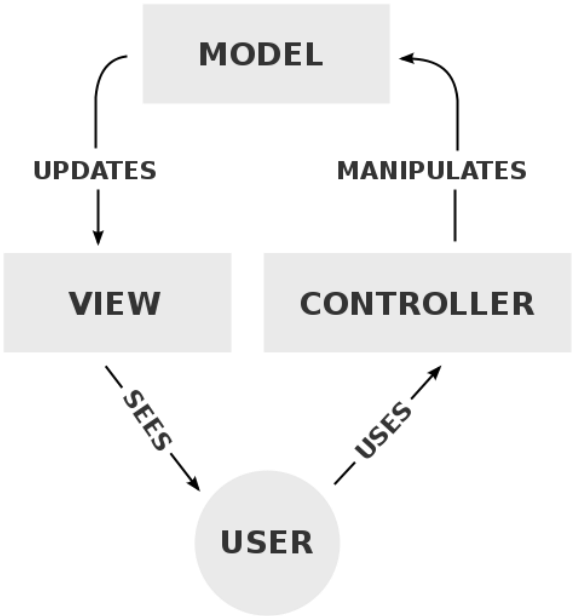
Τα πρώτα MVC frameworks είχαν μια λεπτή προσέγγιση προς τον πελάτη, τοποθετόντας σχεδόν εξ ολοκλήρου το μοντέλο, την προβολή και τον ελεγκτή στο διακομιστή. Σε αυτή την προσέγγιση, ο πελάτης στέλνει είτε hyperlink requests ή εισαγωγή φόρμας στο controller και στη συνέχεια λαμβάνει μια πλήρης και ενημερωμένη web σελίδα (ή άλλο έγγραφο) από την προβολή (View). Το μοντέλο υπάρχει εξ ολοκλήρου στο διακομιστή. Καθώς οι client-side τεχνολογίες έχουν ωριμάσει, frameworks, όπως το JavaScriptMVC και το Backbone έχουν δημιουργηθεί που επιτρέπουν στοιχεία του MVC να εκτελούνται εν μέρει από τον πελάτη (βλέπε επίσης AJAX).

Ιστορία

Το MVC ήταν μια από τις πλέον πρωτοπόρες ιδέες στην πρόωρη ανάπτυξη των γραφικών διεπαφών χρήστη, και μία από τις πρώτες προσεγγίσεις για να περιγράψουν και να εφαρμόσουν δομές λογισμικού από την άποψη των αρμοδιοτήτων τους.

Ο Trygve Reenskaug εισήγαγε MVC στο Smalltalk-76 κατά την επίσκεψή του στη Xerox Parc στη δεκαετία του 1970. Στη δεκαετία του 1980, ο Jim Althoff και άλλοι έθεσαν σε εφαρμογή μια έκδοση του MVC για τη βιβλιοθήκη Smalltalk-80 class. Κι όμως, δεν ήταν παρά μόνο αργότερα, σε ένα άρθρο του 1988 στο The Journal of Object Technology, που το MVC εκφράστηκε ως γενική έννοια.

Το μοντέλο MVC εξελίχθηκε στη συνέχεια, προκαλώντας παραλλαγές, όπως HMVC, MVA, MVP, MVVM, και άλλοι που προσαρμόζονται MVC σε διαφορετικά πλαίσια.



Δομή Μοντέλου

e. **REST (**Representational state transfer**)**

Representational state transfer (REST) είναι ένα αρχιτεκτονικό στυλ που αποτελείται από ένα συντονισμένο σύνολο αρχιτεκτονικών περιορισμών που εφαρμόζονται σε κατασκευαστικά στοιχεία, συνδέσμους και στοιχεία δεδομένων, μέσα σε ένα κατανεμημένο σύστημα υπερμέσων. Το REST αγνοεί τις λεπτομέρειες της εφαρμογής στοιχείου και της σύνταξης του πρωτοκόλλου, προκειμένου να επικεντρωθεί σχετικά με τους ρόλους των στοιχείων, οι περιορισμοί την αλληλεπίδρασή τους με άλλα συστατικά, και την ερμηνεία τους των σημαντικών στοιχείων δεδομένων.

Ο όρος REST εισήχθη και καθορίστηκε το 2000 από τον Roy Fielding στη διδακτορική του διατριβή στο Πανεπιστήμιο της Καλιφόρνιας Irvine.

Το REST έχει εφαρμοστεί για να περιγράψει την επιθυμητή αρχιτεκτονική web, να εντοπίσει τα υφιστάμενα προβλήματα, να συγκρίνει τις εναλλακτικές λύσεις, και να εξασφαλιστεί ότι οι επεκτάσεις του πρωτοκόλλου δεν θα παραβιάζουν τις βασικές συνιστόσες που κάνουν το Web τόσο επιτυχημένο. Η παράταξη που χρησιμοποιείται για το σχεδιασμό REST είναι HTTP 1.1 και Ενιαία αναγνωριστικά πόρων (URI).

Το αρχιτεκτονικό στυλ REST εφαρμόζεται επίσης για την ανάπτυξη των υπηρεσιών Web ως εναλλακτική λύση σε άλλες προδιαγραφές κατανεμημένων υπολογιστικών όπως το SOAP.

Ιστορία

Το αρχιτεκτονικό ύφος REST αναπτύχθηκε από την W3C Ομάδα Τεχνικής Αρχιτεκτονικής (TAG) παράλληλα με το HTTP 1.1, με βάση το υπάρχον σχέδιο του HTTP 1.0 . Το World Wide Web αποτελεί τη μεγαλύτερη εφαρμογή ενός συστήματος σύμφωνα με το REST αρχιτεκτονικό στυλ.

Αρχιτεκτονική Λογισμικού

Μια αρχιτεκτονική λογισμικού ορίζεται από μια διαμόρφωση των εξαρτημάτων, συνδετήρες, και τα δεδομένα περιορίζονται στις σχέσεις τους, προκειμένου να επιτευχθεί ένα επιθυμητό σύνολο αρχιτεκτονικών ιδιοτήτων.

- Στοιχεία(Compoments):
  Ένα στοιχείο είναι μια αφηρημένη μονάδα οδηγίες του λογισμικού και εσωτερική κατάσταση που παρέχει μια μετατροπή των δεδομένων μέσω της διασύνδεσης.
- Συνδέσεις (Connectors):
  είναι ένας αφηρημένος μηχανισμός που μεσολαβεί την επικοινωνία, το συντονισμό ή τη συνεργασία μεταξύ των συνιστωσών.
- Δεδομένα(Data):
  Ένα σημείο αναφοράς είναι ένα στοιχείο των πληροφοριών που μεταφέρεται από ένα συστατικό, ή που λαμβάνεται από ένα συστατικό, μέσω ενός συνδέσμου.

Αρχιτεκτονικές ιδιότητες

Οι αρχιτεκτονικές ιδιότητες που προκαλείται από τα αρχιτεκτονικά περιορισμούς του REST αρχιτεκτονικό στυλ είναι:

- Επίδοση
- Επεκτασιμότητα των αλληλεπιδράσεων στοιχείων
  Η επεκτασιμότητα του REST μπορεί να περιγραφεί όπως παρακάτω:
  Ο διαχωρισμός client-server του REST για τα προβλήματα απλοποιεί την υλοποίηση στοιχείων, μειώνει την πολυπλοκότητα της σημασιολογίας συνδετών, βελτιώνει την αποτελεσματικότητα της ρύθμισης απόδοσης, και αυξάνει την επεκτασιμότητα των καθαρών συστατικών του διακομιστή. Πολυεπίπεδοι περιορισμοί του συστήματος επιτρέπουν στους μεσάζοντες-proxies, πύλες και τα τείχη προστασίας- να εισαχθούν σε διάφορα σημεία της λειτουργίας της εφαρμογής, χωρίς αλλαγή των διεπαφών μεταξύ των στοιχείων, επιτρέποντας τους έτσι να βοηθήσουν στη μετάφραση της επικοινωνίας ή τη βελτίωση των επιδόσεων μέσω μεγάλης κλίμακας, από κοινού caching. Το REST επιτρέπει ενδιάμεση επεξεργασία περιορίζοντας τα μηνύματα να είναι αυτο-περιγραφικά: η αλληλεπίδραση είναι χωρίς κατάσταση μεταξύ των αιτημάτων, πρότυπες μεθόδους και τους τύπους μέσων που χρησιμοποιούνται για να δηλώσουν τη σημασιολογία και την ανταλλαγή πληροφοριών, καθώς και οι απαντήσεις δείχνουν σαφώς δυνατότητα cache.
- Απλότητα των διεπαφών
- Διαμόρφωση των στοιχείων για την ικανοποίηση των μεταβαλλόμενων αναγκών (ακόμα και όταν η εφαρμογή εκτελείται)
- Ορατότητα της επικοινωνίας μεταξύ των εξαρτημάτων από πράκτορες των υπηρεσιών
- Φορητότητα εγκατάστασης στοιχείων
- Αξιοπιστία

Αρχιτεκτονικοί περιορισμοί

Οι αρχιτεκτονικές ιδιότητες του REST πραγματοποιούνται με την εφαρμογή ειδικών περιορισμών αλληλεπιδράσεων με τα συστατικά, συνδετήρες, και στοιχεία δεδομένων.

Οι τυπικοί περιορισμοί του REST είναι:

Client–server

Μια ενιαία διεπαφή χωρίζει τους πελάτες από τους διακομιστές. Αυτός ο διαχωρισμός των προβλημάτων που σημαίνει ότι, για παράδειγμα, οι πελάτες δεν ασχολούνται με την αποθήκευση δεδομένων, η οποία παραμένει στο εσωτερικό κάθε διακομιστή, έτσι ώστε η δυνατότητα μεταφοράς του κωδικού πελάτη να είναι βελτιωμένη. Οι διακομιστές δεν ασχολούνται με το περιβάλλον εργασίας χρήστη ή της κατάστασης του χρήστη, έτσι ώστε οι διακομιστές μπορεί να είναι απλούστεροι και πιο επεκτάσιμοι. Διακομιστές και πελάτες μπορούν επίσης να αντικατασταθούν και να αναπτύσσεται ανεξάρτητα, εφ 'όσον η διασύνδεση μεταξύ τους δεν μεταβάλλεται.

Stateless

Η επικοινωνία client-server περιορίζεται περαιτέρω από κανένα πλαίσιο πελάτη που αποθηκεύονται στο διακομιστή μεταξύ των αιτημάτων. Κάθε αίτηση από οποιονδήποτε πελάτη περιέχει όλες τις απαραίτητες πληροφορίες για να εξυπηρετήσει το αίτημα, και η κατάσταση σύνοδος διεξάγεται στον πελάτη. Σημαντικό να σημειωθεί είναι ότι η κατάσταση περιόδου λειτουργίας μπορεί να μεταφερθεί από τον server σε άλλη υπηρεσία, όπως μια βάση δεδομένων για να διατηρήσει μια επίμονη κατάσταση για κάποιο χρονικό διάστημα και να επιτρέψει τον έλεγχο ταυτότητας. Ο πελάτης ξεκινά την αποστολή των αιτήσεων όταν είναι έτοιμοι να κάνουν τη μετάβαση σε μια νέα κατάσταση. Ενώ το ένα ή περισσότερα αιτήματα είναι εξαιρετική, ο πελάτης θεωρείται ότι είναι σε μεταβατικό στάδιο. Η αναπαράσταση κάθε κατάστασης αίτησης περιέχει συνδέσμους που μπορούν να χρησιμοποιηθούν την επόμενη φορά που ο πελάτης επιλέγει να ξεκινήσει μια νέα κατάσταση μετάβασης.

Cacheable

Όπως και για το World Wide Web, οι πελάτες μπορούν να αποθηκεύουν προσωρινά τις απαντήσεις. Οι απαντήσεις θα πρέπει, ως εκ τούτου, σιωπηρά ή ρητά, αυτοπροσδιορίζονται ως cacheable, ή όχι, να αποτρέψει τους πελάτες επαναχρησιμοποίηση παλιών ή ακατάλληλων δεδομένων σε απάντηση της σε περαιτέρω αιτήματα. Σωστά διαχειριζόμενο caching εν μέρει ή πλήρως, καταργεί ορισμένες αλληλεπιδράσεις client-server, βελτιώνοντας περαιτέρω την επεκτασιμότητα και την απόδοση.

Layered system

Ένας πελάτης δεν μπορεί κανονικά να ξεχωρίσει αν είναι συνδεδεμένος απευθείας με το διακομιστή ή σε έναν ενδιάμεσο διακομιστή στη σύνδεση του. Ενδιάμεσοι διακομιστές μπορεί να βελτιώσει την επεκτασιμότητα του συστήματος, επιτρέποντας εξισορρόπηση φορτίου και με την παροχή κοινών κρύπτες. Μπορούν επίσης να επιβάλουν τις πολιτικές ασφάλειας.

Code on demand (optional)

Οι servers μπορούν να παρατείνουν προσωρινά ή να προσαρμόσουν τη λειτουργικότητα ενός πελάτη με τη μεταφορά εκτελέσιμο κώδικα. Παραδείγματα αυτό μπορεί να περιλαμβάνει καταρτίζονται συστατικά όπως βοηθητικές εφαρμογές Java και client-side scripts όπως η JavaScript. "Κώδικας on demand" είναι η μόνη προαιρετική περιορισμός της αρχιτεκτονικής REST.

Έννοια

Το REST προορίζεται να προκαλέσει μια εικόνα για το πώς μια καλά σχεδιασμένη εφαρμογή Web συμπεριφέρεται: παρουσιάζονται με ένα δίκτυο ιστοσελίδων (μια εικονική κατάσταση-μηχανής), ο χρήστης προχωρά μέσω μιας εφαρμογής επιλέγοντας συνδέσμους (μεταβάσεις), με αποτέλεσμα η επόμενη σελίδα (που αντιπροσωπεύει η επόμενη κατάσταση της αίτησης) που μεταβιβάζεται στο χρήστη και να καταστεί για τη χρήση τους.

Το REST περιγράφηκε αρχικά στα πλαίσια της HTTP, αλλά δεν περιορίζεται σε αυτό το πρωτόκολλο. Ξεκούραστη αρχιτεκτονικές μπορούν να βασίζονται σε άλλα πρωτόκολλα επιπέδου εφαρμογής, εφόσον παρέχουν ήδη ένα πλούσιο και ομοιόμορφο λεξιλόγιο για τις εφαρμογές που βασίζονται στη μεταβίβαση των ουσιαστικών παραστατικής κατάσταση. Ξεκούραστη εφαρμογές μεγιστοποιούν τη χρήση της υφιστάμενης, καλά καθορισμένης διεπαφής και άλλες ενσωματωμένες δυνατότητες που παρέχονται από την επιλεγμένη πρωτοκόλλου του δικτύου, και να ελαχιστοποιήσει την προσθήκη νέων χαρακτηριστικών για συγκεκριμένες εφαρμογές πάνω του.

Επαναχρησιμοποίηση λεξιλόγιο εναντίον αυθαίρετη επέκταση του: HTTP και SOAP

Εκτός από URIs, τύπους μέσων Internet, αιτήματα και κωδικούς απόκρισης, κλπ., το HTTP έχει ένα λεξιλόγιο των πράξεων αποκαλούνται μέθοδοι αιτημάτων, και κυρίως τα :

GET

POST

PUT

DELETE

Το REST χρησιμοποιεί αυτές τις πράξεις και τα άλλα υπάρχοντα χαρακτηριστικά του πρωτοκόλλου HTTP. Για παράδειγμα, πολυεπίπεδη μεσολάβησης και πύλη συστατικά εκτελούν πρόσθετες λειτουργίες για το δίκτυο, όπως HTTP caching και την επιβολή της ασφάλειας.

Αγνοεί πολλές από τις υπάρχουσες δυνατότητες HTTP, όπως η πιστοποίηση, caching, και το περιεχόμενο τύπου διαπραγμάτευσης Αυτό λεξιλόγιο πρόσθετο. Το SOAP πλεονέκτημα έχει πάνω REST είναι ότι διαθέτει μια άμεση εκπροσώπηση των εν λόγω εννοιών του λογισμικού που είναι σύγχρονα ευρέως δέχονται, όπως π.χ. "OOP αντικειμένων" με τα στοιχεία μέλους και τα καθήκοντα του μέλους και τις σχέσεις μεταξύ τους, και ότι οι εφαρμογές SOAP απρόσκοπτα νοιάζονται για unmarshalling των δεδομένων που προέρχονται από μια ανταλλαγή δεδομένων HTTP (ανεξάρτητα της ύπαρξης αίτηση ή απόκριση δεδομένων) και διαλογής των αντικειμένων σε μια αναπαράσταση που είναι κατάλληλο για μετάδοση σε ένα πρωτόκολλο δικτύου.

Το SOAP μπορεί να χρησιμοποιηθεί χωρίς HTTP, αλλά στην πράξη χρησιμοποιείται κυρίως ως απλά ένα στρώμα RPC πάνω από HTTP.

Κεντρική αρχή

Μια σημαντική έννοια στο REST είναι η ύπαρξη των πόρων (πηγές ειδικών πληροφοριών), καθένα από τα οποία γίνεται αναφορά με ένα παγκόσμιο αναγνωριστικό (π.χ., ένα URI στο HTTP). Για να χειραγωγήσουν τους πόρους αυτούς, τα συστατικά του δικτύου (πράκτορες χρήστη και διακομιστές προέλευσης) επικοινωνούν μέσω τυποποιημένης διασύνδεσης (π.χ., HTTP) και την ανταλλαγή αναπαραστάσεις αυτών των πόρων (τα ίδια τα έγγραφα μεταφοράς των πληροφοριών). Για παράδειγμα, ένας πόρος που αντιπροσωπεύει κύκλο (ως λογική αντικείμενο) μπορεί να δεχθεί και να επιστρέψει μια παράσταση που καθορίζει ένα σημείο κέντρο και ακτίνα, διαμορφωμένη σε SVG, αλλά μπορεί επίσης να δεχθεί

και να επιστρέψει μια παράσταση που καθορίζει κάθε τρία διακριτά σημεία κατά μήκος της καμπύλης (δεδομένου ότι αυτό επίσης ταυτοποιεί έναν κύκλο) ως λίστα διαχωρισμένη με κόμματα.

Οποιοσδήποτε αριθμός των συνδετήρων (π.χ. πελάτες, τους διακομιστές, κρύπτες, σήραγγες, κ.α.) μπορεί να μεσολαβήσει το αίτημα, αλλά ο καθένας κάνει αυτό χωρίς "να δει το παρελθόν" δικό της αίτημα (που αναφέρεται ως "layering", ένα άλλο περιορισμό της REST και μια κοινή αρχή και σε πολλά άλλα μέρη της πληροφόρησης και αρχιτεκτονική δικτύωσης). Έτσι, μια εφαρμογή μπορεί να αλληλεπιδράσει με έναν πόρο, γνωρίζοντας δύο πράγματα: το αναγνωριστικό του πόρου και η δράση που απαιτείται, δεν χρειάζεται να γνωρίζει εάν υπάρχουν κρύπτες, πληρεξούσια, πύλες, firewalls, σήραγγες, ή οτιδήποτε άλλο μεταξύ αυτής και η διακομιστή που κατέχουν πραγματικά τις πληροφορίες. Η εφαρμογή, ωστόσο, πρέπει να κατανοήσουν τη μορφή των πληροφοριών (αναπαράσταση) επέστρεψε, το οποίο είναι συνήθως ένα HTML, XML, ή έγγραφο JSON κάποιου είδους, αν και μπορεί να είναι μια εικόνα, απλό κείμενο, ή οποιοδήποτε άλλο περιεχόμενο.

Γιατί να χρησιμοποιήσετε την REST Αρχιτεκτονική για τη δημιουργία Web Services;

Αναπαράστασης Μεταφορά κράτος (REST) είναι μια νέα αρχιτεκτονική για τις υπηρεσίες web που έχει σημαντικό αντίκτυπο στη βιομηχανία . Οι περισσότερες από τις νέες δημόσιες υπηρεσίες web από τις μεγάλες πωλητές (Google , Yahoo , Amazon , Microsoft) βασίζονται σε REST , όπως η τεχνολογία για την ανταλλαγή και τη συγχώνευση πληροφοριών από πολλαπλές πηγές .

Το REST δεν είναι ένα επίσημο πρότυπο , αλλά μάλλον ένα αρχιτεκτονικό ύφος των δικτυωμένων συστημάτων που αποτελούνται από πελάτες και διακομιστές . Πελάτες κίνηση των αιτήσεων servers? Servers επεξεργασία των αιτημάτων και να επιστρέψει τις κατάλληλες απαντήσεις . Οι αιτήσεις και οι απαντήσεις είναι χτισμένα γύρω από τη μεταφορά της « αναπαραστάσεις» της « πόρων» . Ένας πόρος μπορεί να είναι οποιαδήποτε συνεκτική και ουσιαστική έννοια που απευθύνεται . Μια αναπαράσταση ενός πόρου είναι συνήθως ένα έγγραφο που καταγράφει την τρέχουσα ή προβλεπόμενη κατάσταση του πόρου .

Το κίνητρο πίσω από REST υπηρεσίες web είναι αυτό που οδηγεί την πρόοδο της τεχνολογίας : να κάνουν περίπλοκα πράγματα απλούστερα . Οι υπηρεσίες web πρώτης γενιάς στηρίχθηκε στην ανταλλαγή πακέτων XML σύμφωνα με το SOAP (απλό πρωτόκολλο πρόσβασης αντικειμένου) τις προδιαγραφές που χρησιμοποιούν το πρωτόκολλο HTTP . Στην πραγματικότητα προδιαγραφών υπηρεσίες web δεν λέει ότι τα μηνύματα SOAP πρέπει να ανταλλάσσονται μέσω HTTP . Μμε είναι η κατεύθυνση ότι η Microsoft πήγε με την αρχιτεκτονική WCF τους . Αυτό αποσυνδέει μηνυμάτων SOAP από το υποκείμενο πρωτόκολλο επικοινωνίας που επιτρέπει για το TCP / IP και άλλους τρόπους ανταλλαγής SOAP . Αλλά ... χρειαζόμαστε SOAP σε όλα;

Υποστηρικτές της αρχιτεκτονικής REST θεωρούν SOAP και XML να είναι πολύ βαρύ . HTTP ίδια έχει αρκετό δυνατότητες για εφαρμογές να επικοινωνούν μέσω του δικτύου . Στην πραγματικότητα , HTTP είναι ποιες αρμοδιότητες Web και έχει ένα πολύ πλούσιο λεξιλόγιο από την άποψη των ρημάτων , URIs , αίτησης και απάντησης κεφαλίδες και Internet τύπους μέσων . Είναι επίσης κοινό στις υπηρεσίες REST να χρησιμοποιήσετε JSON (JavaScript Object Notation) ως ένα βολικό , "χωρίς λιπαρά " εναλλακτική λύση για την XML . JSON , είναι ένα υποσύνολο της JavaScript , είναι μια ιδανική μορφή των δεδομένων για τη δημιουργία καθαρών πελάτες HTML τρέχει στο web browsers με ενσωματωμένη λογική JavaScript και πρόσβαση REST πόρων σε μια ασύγχρονη τρόπο χρησιμοποιώντας AJAX για υψηλά ανταποκρινόμενες διεπαφές χρήστη .

Με την έκρηξη του web , και τώρα Web 2.0 , είναι μόλις τώρα αρχίζουμε να βλέπουμε httthe αυτόματη αλληλεπίδραση μεταξύ των διαφόρων ιστοσελίδων, καθώς και μεταξύ των web sites και εφαρμογές client , web sites και των βάσεων δεδομένων των επιχειρήσεων , ως μέρος μιας παγκόσμιας περισσότερα διασύνδεσης .

Στο διαδίκτυο , τα δεδομένα κινείται ταχύτερα από ό, τι μπορούμε να περιηγηθείτε . Ως εκ τούτου , υπάρχει ισχυρή ζήτηση για τα προγράμματα που μπορεί να βρει , παρακολουθεί και να ελέγχει τις πληροφορίες που προέρχονται από διάφορες πηγές, συμπεριλαμβανομένων των δεδομένων πωλήσεων , οικονομικές πληροφορίες , online κοινότητες , εκστρατείες μάρκετινγκ , για να αναφέρουμε μερικές .

Η ραγδαία αναδυόμενη τεχνολογία των υπηρεσιών web έχει τη δυνατότητα να αλλάξει τον τρόπο που λειτουργεί το Ίντερνετ για τις επιχειρήσεις . Ενώ σήμερα οι υπηρεσίες web λειτουργεί καλά για τις επιχειρήσεις προς τους καταναλωτές εφαρμογές , δεν το κάνουν για τις επιχειρήσεις - to-business εφαρμογές . Για παράδειγμα , ένας μεμονωμένος πελάτης μπορεί εύκολα να αγοράσει ένα βιβλίο από ένα online πωλητή , αλλά για ένα βιβλιοπωλείο που θέλει να κάνει μια αγορά όγκο, είναι πολύ πιο περίπλοκη . Το βιβλιοπωλείο ενδεχομένως θα χρειαστεί να χρησιμοποιήσετε πολλαπλές εφαρμογές για την παρακολούθηση των πωλήσεων , να καθορίσουν εκ νέου εντολές και να παρακολουθείτε την αποστολή . Συχνά , τα δεδομένα από τη μία εφαρμογή πρέπει να τεθεί εκ νέου σε άλλες εφαρμογές , κάνοντας την όλη διαδικασία αναποτελεσματική . Οι υπηρεσίες Web θα πρέπει να είναι σε θέση να αντιμετωπίσει αυτό το ζήτημα .

Το REST μπορεί επίσης να αλλάξει τον τρόπο με τον τρόπο με τον οποίο είναι σύνθετα συστήματα architected . Παραδοσιακή Service Oriented Architecture (SOA) κινείται αργά προς Web Oriented Architecture (WOA) - ο όρος επινοήθηκε από Dion Hinchcliff - όπου οι εφαρμογές μόχλευσης μια πλούσια ιστοσελίδα της REST πόρων . Αντί για μερικές υπηρεσίες σημείο SOA , τα στοιχεία των επιχειρήσεων θα εκτεθεί μέσω εκατομμύρια κόκκους πόρων REST , όπως και το ίδιο το διαδίκτυο .

Το REST γίνεται γρήγορα η προτιμώμενη τεχνολογία για την κατασκευή αυθαίρετων εφαρμογές που επικοινωνούν μέσω του δικτύου . Το REST αξιοποιεί πλήρως τα πρωτόκολλα και τα πρότυπα που τροφοδοτούν το World Wide Web και είναι απλούστερη από τις παραδοσιακές υπηρεσίες web SOAP -based . Με την εμφάνιση του cloud computing , και το αυξανόμενο ενδιαφέρον για το web εφαρμογές που φιλοξενούνται , οι τεχνολογίες που βασίζονται σε REST μπορούν να βοηθήσουν τόσο στην ανάπτυξη των πελατών πλούσια διεπαφή χρήστη καλώντας σε απομακρυσμένους διακομιστές ? Και στην ανάπτυξη της πραγματικής servers για το χειρισμό των δομών δεδομένων σε έναν πελάτη αίτηση (γραμμένο σε οποιαδήποτε γλώσσα) ή απευθείας στο πρόγραμμα περιήγησης .

Πηγή άρθρου:

embarcadero.com

Tim DelChiaro

4. CodeIngiter και κώδικας

Το CodeIgniter είναι μια ταχείου πλαισίου web εφαρμογή ανάπτυξης λογισμικού ανοικτού κώδικα , για χρήση στην κατασκευή δυναμικών ιστοσελίδων με την PHP . Η πρώτη δημόσια έκδοση του CodeIgniter κυκλοφόρησε στις 28 του Φεβρουαρίου του 2006 , και η τελευταία σταθερή έκδοση 2.1.4 κυκλοφόρησε 8 του Ιούλη του 2013 .

Το CodeIgniter είναι αόριστα βασισμένο στο δημοφιλές Model-View - Controller πρότυπο ανάπτυξης . Ενώ τα views και οι1 controllers είναι ένα απαραίτητο μέρος της ανάπτυξης στο πλαίσιο CodeIgniter , τα models είναι προαιρετικά.

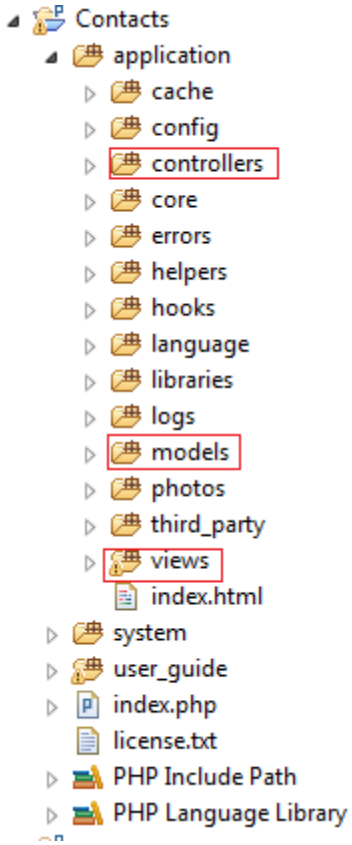
Το CodeIgniter πιο συχνά σημειώνεται για την ταχύτητά του σε σύγκριση με άλλα PHP frameworks. Σε μια σημαντική κρίση PHP frameworks σε γενικές γραμμές , ο PHP δημιουργός Rasmus Lerdorf μίλησε στο Froscon τον Αύγουστο του 2008 , σημειώνοντας ότι του άρεσε CodeIgniter "επειδή είναι πιο γρήγορα, πιο ελαφρύ και το λιγότερο σαν ένα framework . "

Ο πηγαίος κώδικας του CodeIgniter διατηρείται στο GitHub , και από την preview έκδοση 3.0 - dev , είναι πιστοποιημένο λογισμικό ανοιχτού κώδικα σύμφωνα με την άδεια License Λογισμικό Ανοικτού (" OSL ") κατά 3,0 . Οι εκδόσεις του CodeIgniter πριν από 3,0 είναι αδειοδοτημένο με ένα ιδιόκτητο Apache / BSD - style άδεια ανοιχτού κώδικα .

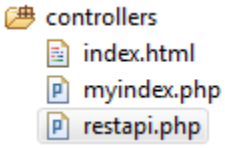
Η απόφαση για τη μετάβαση σε μια άδεια OSL προκάλεσε κάποια διαμάχη στην κοινότητα , ιδιαίτερα σχετικά με την GPL ασυμβίβαστο της νέας άδειας , για την οποία EllisLab ανταποκρίθηκε με μια σειρά άρθρων με τίτλο Software License Εβδομάδα Ευαισθητοποίησης .

Στις 9 Ιουλίου 2013, EllisLab ανακοίνωσε ότι αναζητά ένα νέο ιδιοκτήτη για CodeIgniter του , δηλώνοντας την έλλειψη συμμετοχής ως λόγο . Δεν είναι σαφές εάν CodeIgniter θα παραμείνει η ραχοκοκαλιά του λογισμικού Expression Engine της.

Παρακάτω βλέπετε τη δομή του συστήματος αρχείων του CI (CodeIngiter):



Σημειωμένα είναι οι φάκελοι τους οποίους χρησιμοποιούμε για να φτιάξουμε την εφαρμογή μας (Contacts), όπου στον καθένα τοποθετούνται οι αντίστοιχες κλάσεις (Controllers, Modes & Views). Ας αρχίσουμε να βλέπουμε κάποια κομμάτια από τον κώδικα της εφαρμογής. Ξεκινάμε με τους Controllers που είναι και το κέντρο της λογικής της εφαμοργφής.

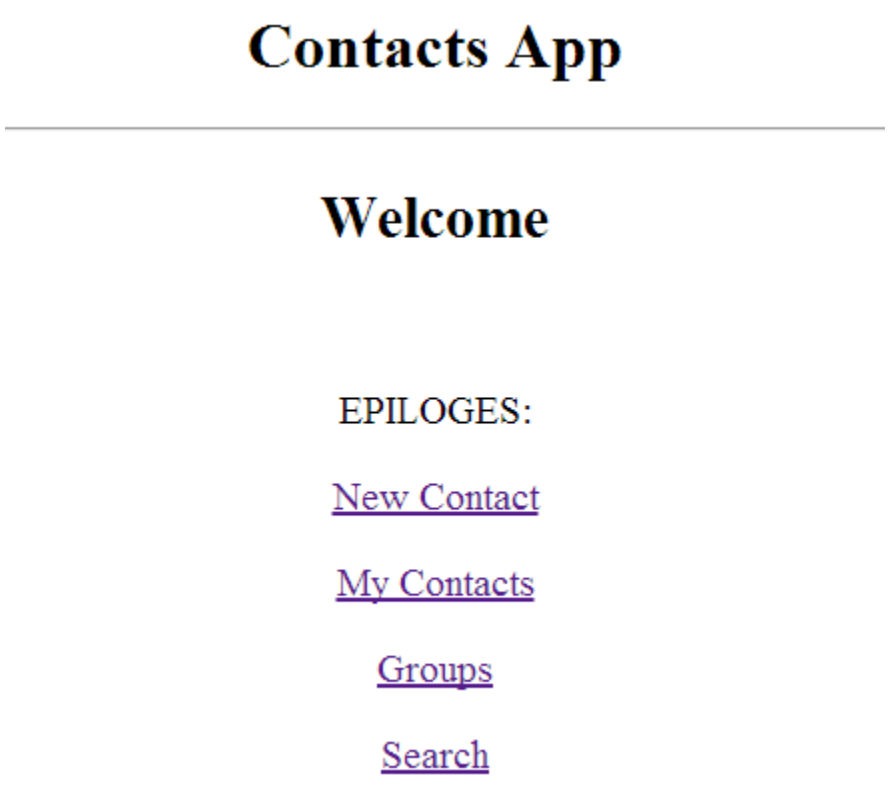


Μέσα στο φάκελο έχουμε δημιουργήσει τον κεντρικό controller myindex.php όπου είναι όλες οι κλάσεις που χρησιμοποιούμε για τις λειτουργίες του κώδικα. Επίσης υπάρχει ένα αρχείο .html το οποίο χρησιμεύει για λόγους ασφαλείας στην εφαρμογή μας. Τέλος, υπάρχει και το αρχείο restapi.php μέσα στο οποίο είναι οι κλάσεις που ορίζουν το REST API μας.

Ως παράδειγμα, έχουμε παρακάτω την κεντρική μέθοδο με την οποία ξεκινάει η εφαρμογή μας:

```
public function index()
{
    $this->load->view('header');
    $this->load->view('app_menu');
    $this->load->view('footer');
}
```

Η κλάση καλεί 3 Views, τα οποία είναι ξεκάθαρο το τι εμφανίζουν από το όνομά τους. Με αυτές τις ελάχιστες γραμμές κώδικα, καλούνται αντίστοιχα αρχεία html για το front-end της εφαρμογής και μας παρουσιάζουν την παρακάτω κεντρική σελίδα:



Στη συνέχεια θα δούμε μια άλλη μέθοδο της κλάσης, την contact, την οποία βλέπουμε παρακάτω:

```
public function contact($id) {
    $data = array();
    $data['contact_id'] = $id;
    $data['contact'] = $this->contacts_model->get_contact($id);
    $this->load->model('group_model');
    $data['groups'] = $this->group_model->get_group($id);
    $this->load->model('photo_model');
    $data['photo'] = $this->photo_model->get_photo($id);
    $this->load->view('header');
    $this->load->view('contact', $data);
    $this->load->view('footer');
}
```

Σκοπός της μεθόδου είναι να μας εμφανίζει μια συγκεκριμένη επαφή που έχουμε επιλέξει. Αρχικά δημιουργούμε μια μεταβλητή πίνακα (array) για να μεταφέρουμε την πληροφορία που θα τραβήξουμε από τη βάση δεδομένων (μέσω κάποιων models) στην προβολή (View) που θα δούμε παρακάτω. Στη συνέχεια παίρνουμε το id της επαφής που έχει ζητηθεί από το όρισμα που δέχεται η μέθοδος και το αποθηκεύουμε στον πίνακα με όνομα “contact_id”. Στη συνέχεια καλούμε από το model contacts_model τη μέθοδο get_contact() η οποία παίρνει ως όρισμα το id που είδαμε προηγουμένως και επιστρέφει της πληροφορίες της επαφής, οι οποίες αποθηκεύονται στον πίνακα, με την ονομασία “contact”. Να σημειώσουμε ότι επειδή το συγκεκριμένο model μας χρειάζεται συνέχεια στην εφαρμογή, το έχουμε ορίσει να είναι φορτωμένο συνέχεια, μέσω της δυνατότητας autoload που παρέχει το CI. Σε αντίθετη περίπτωση, θα έπρεπε πρώτα να το φορτώσουμε για να μπορέσουμε να το χρησιμοποιήσουμε, όπως γίνεται αμέσως μετα με το model “photo_model”. Χρησιμοποιόντας την βιβλιοθήκη “load” φορτώνουμε το model που θέλουμε και στη συνέχεια χρησιμοποιούμε τη μέθοδό του, “get_photo()” η οποία μας επιστρέφει τις πληροφορίες της φωτογραφίας της επαφής και τις αποθηκεύει στον πίνακα $data με την ονομασία “photo”.

Τέλος, αφού έχουμε συγκεντρώσει όλη την πληροφορία που χρειαζόμαστε στον $data, καλούμε τα Views που χρειαζόμαστε και την περνάμε στο View που θα τις εμφανίζει (“contact”).

Πριν πάμε να δούμε πώς θα εμφανίσει τη σελίδα Contact με τις πληροφορίες που έχουμε ζητήσει, θα δούμε το βασικό model της εφαρμογής, το “contacts_model”.

```
class contacts_model extends CI_Model
{
    function get_contacts() {
        $query = $this->db->get('contacts');
        return $query->result();
    }

    function get_contact($id) {
        $query = $this->db->get_where('contacts', array('id' => $id));
        return $query->result();
    }

    function add_contact($table,$data) {
        $this->db->insert($table, $data);
        return ;
    }

    function delete_contact($id) {
        $this->db->where('id',$id);
        $this->db->delete('contacts');
    }

    function update_contact($id,$data) {
        $this->db->where('id',$id);
        $this->db->update('contacts',$data);
    }

    function get_search($token,$column) {
        //$query = $this->db->get_where('contacts', $column => $token);
        $query = $this->db->query('SELECT * FROM contacts WHERE '.$column.' like "%'.$token.'%"');
        return $query->result();
    }
}
```

Από το συγκεκριμένο, θα δούμε τη μέθοδο που χρησιμοποιήθηκε προηγουμένως, τη “get_contact()”.

```
function get_contact($id) {
    $query = $this->db->get_where('contacts', array('id' => $id));
    return $query->result();
}
```

Η μέθοδος δέχεται σαν όρισμα το id του χρήστη. Στη συνέχεια χρησιμοποιεί τη βιβλιοθήκη “db” και τη μέθοδο αυτής get_where() όπου δίνοντας ως ορίσματα των πίνακα που θέλουμε από τη βάση (contacts) και το πεδίο του πίνακα με την αντίστοιχη τιμή (δίνεται πάντα σε array το δεύτερο όρισμα), αυτόματα παράγει το παρακάτω query: *select * from contacts where id = $id*

Στη συνέχεια, αποθηκεύει τα αποτελέσματα στη μεταβλητή $query και με τη χρήση της μεθόδου result(), επιστρέφει το αποτελέσματα από όπου ζητήθηκαν.

Σειρά έχει το View που παρουσιάζει την επαφή που ζητήθηκε. Παρακάτω θα δούμε το κομμάτι του front-end κώδικα που παρουσιάζει μέρος των στοιχείων της επαφής.

```
<table border=1 cellpadding=10 width=60%>
<?php if(isset($contact)) : foreach($contact as $row) : ?>
    <tr>
        <td width=20%>Name:</td>
        <td><?php echo $row->name; ?></td>
    </tr>
    <tr>
        <td>Surname:</td>
        <td><?php echo $row->surname; ?></td>
    </tr>
    <tr>
        <td>E-mail:</td>
        <td><?php echo $row->email; ?></td>
    </tr>

<?php endforeach; ?>
</table>
<?php else : ?>
<p>No contacts yet</p>
<?php endif;?>
```

Αρχικά δημιουργείται ένα <table> και μια συνθήκη που ελέγχει αν υπάρχουν δεδομένα στη μεταβλητή $contact. Εφόσον υπάρχουν δεδομένα, χρησιμοποιούμε μια foreach για να εμφανίσουμε τα δεδομένα σύμφωνα με το στήσιμο του View. Να σημειωθεί εδώ ότι πλέον μπορούμε να χρησιμοποιούμε ως μεταβλητές/πίνακες τις ονομασίες που δώσαμε στον controller όταν αποθηκεύσαμε πληροφορία στον $data. Αν δεν υπάρχουν, μας εμφανίζεται το μήνυμα “No contacts yet”.

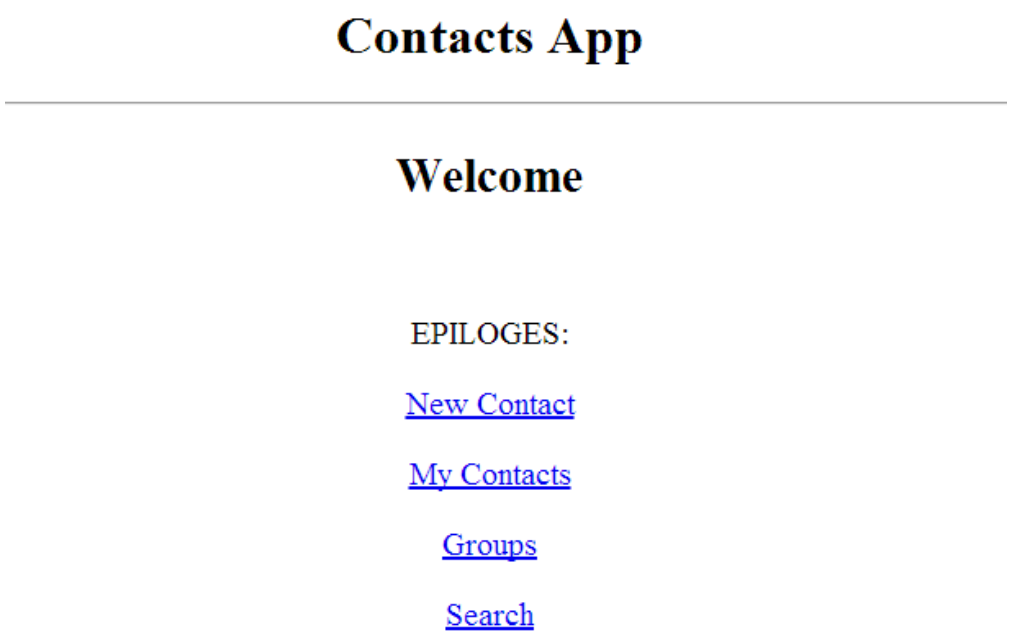
Το παραπάνω, εφόσον υπάρχουν επαφές αποθηκευμένες στην εφαρμογή, θα μας τις εμφανίζει όπως παρακάτω:

Your contacts

Name	Surname	Email	
test	test	test	More
nikos	enas	akomh	More

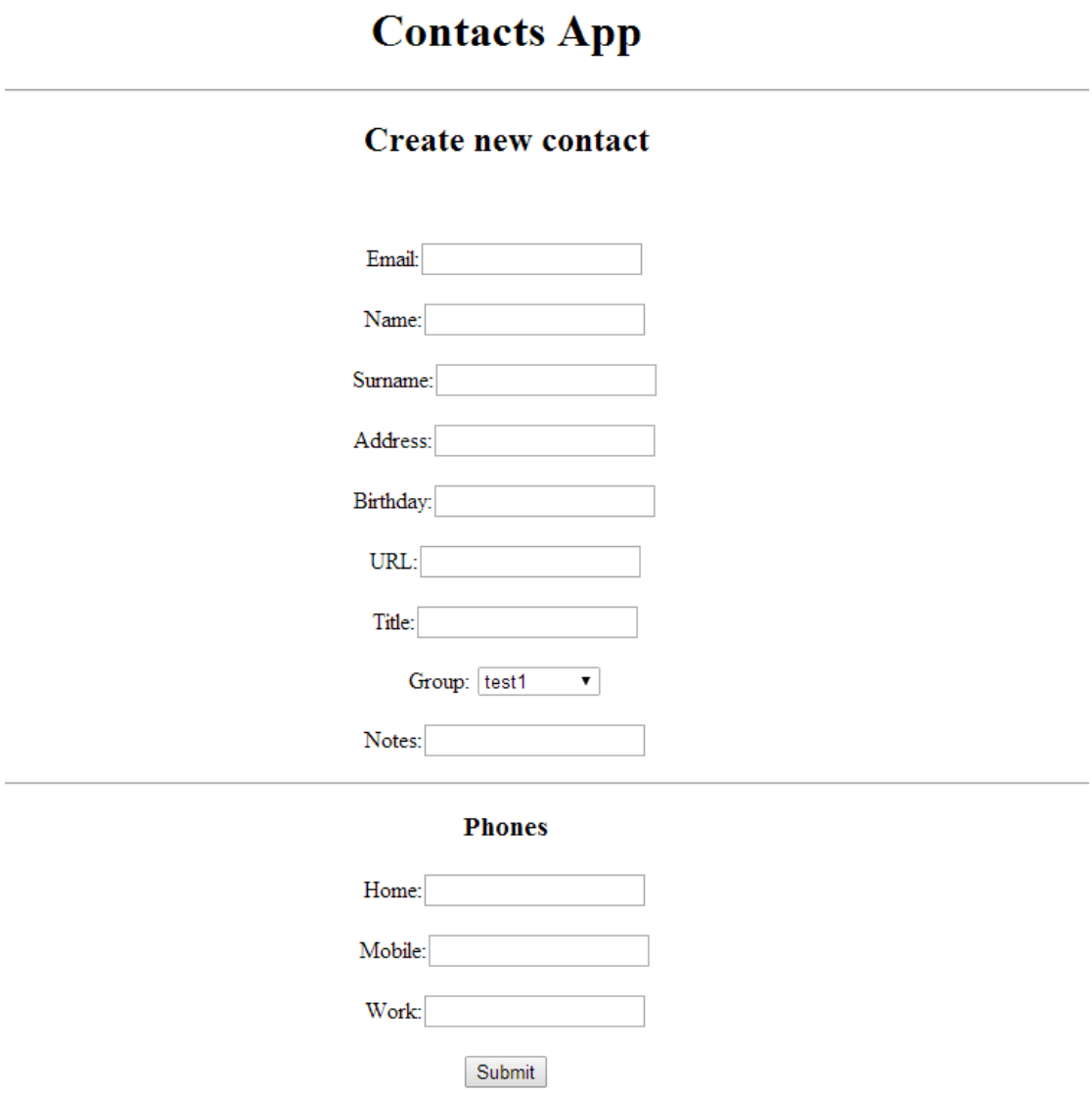
5. ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

Παρακάτω θα δούμε τις λειτουργίες της εφαρμογής μας. Αρχίζουμε με την κεντρική σελίδα της εφαρμογής.



Σε κάθε σελίδα της εφαρμογής υπάρχει στα δεξιά η επιλογή “Back” η οποία μας δίνει τη δυνατότητα να πάμε στην προηγούμενη σελίδα.

Αρχικά θα δημιουργήσουμε μια νέα επαφή, επιλέγοντας “New Contact”. Μας παρουσιάζεται η παρακάτω φόρμα την οποίακ και συμπληρώνουμε με τα στοιχεία που επιθυμούμε.



Σημείωση: Το πεδίο group είναι ενδεικτικό και ορίζεται δυναμικά. Τον τρόπο θα τον δούμε παρακάτω.

Αφού συμπληρώσουμε όσα πεδία επιθυμούμε (ή έχουμε τη σχετική πληροφορία) μεταφερόμαστε στην σελίδα των επαφών.

Contacts App

Your contacts

Name	Surname	Email	
test	test	test	More
nikos	enas	akomh	More
test	test	test@test.gr	More

Εδώ βλέπουμε ότι η επαφή μας δημιουργήθηκε και μας παρουσιάζονται τα απολύτως βασικά (όνομα, επίθετο, email). Εφόσον θέλουμε να δούμε όλες τις πληροφορίες της επαφής πατάμε “More” και μεταφερόμαστε στη σελίδα επαφής.

Contacts App

Your contact

Update	Delete	Change photo

Name:	test
Surname:	test
E-mail:	test@test.gr
Address:	test

Birthday:	
URL:	
Title:	test
Notes:	
Phone(home):	9999999
Phone(mobile):	0
Phone(work):	0
Group:	test1

Στην αρχή της σελίδας βρίσκεται η φωτογραφία της επαφής (αν δεν έχουμε βάλει, εμφανίζεται η παραπάνω default εικόνα) και οι εξής επιλογές: Update, Delete, Change Photo. Και παρακάτω, εμφανίζεται η καρτέλα με όλες τις πληροφορίες που περάσαμε στην επαφή μας.

Ας δούμε την πρώτη από τις τρεις επιλογές, το Update, το οποίο μας παραπέμπει στην παρακάτω καρτέλα.

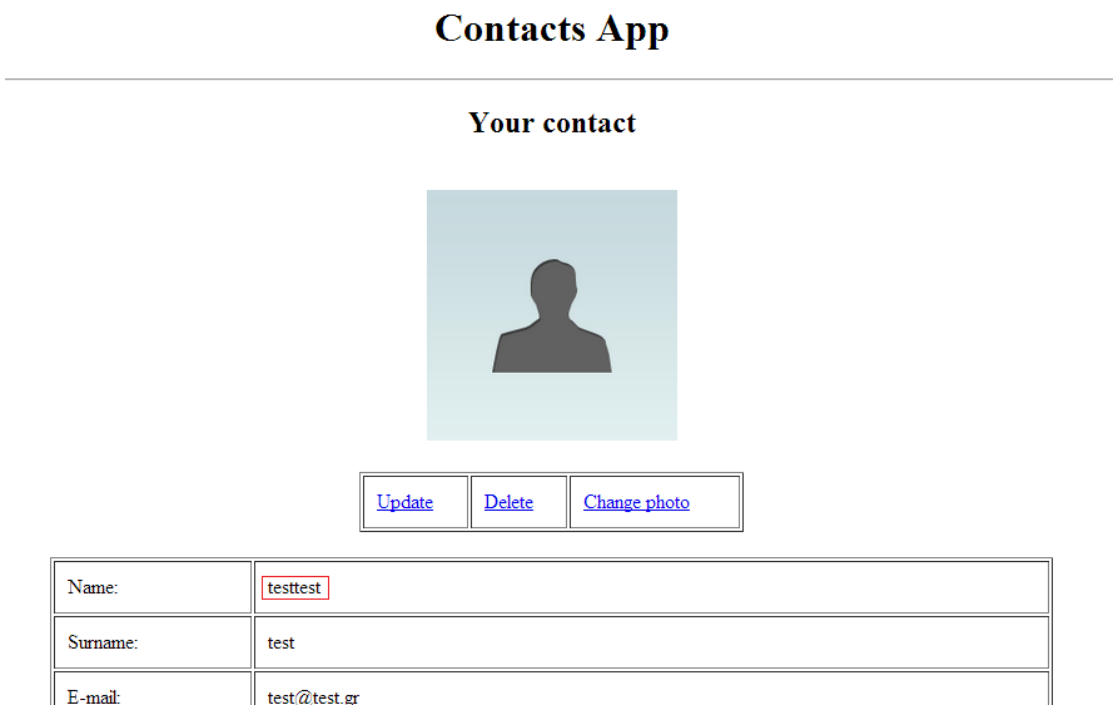
Your contact

Update

Name:	test
Surname:	test
E-mail:	test@test.gr
Address:	test
Birthday:	
URL:	
Title:	test
Notes:	
Phone(home):	9999999
Phone(mobile):	0
Phone(work):	0
Group:	test1

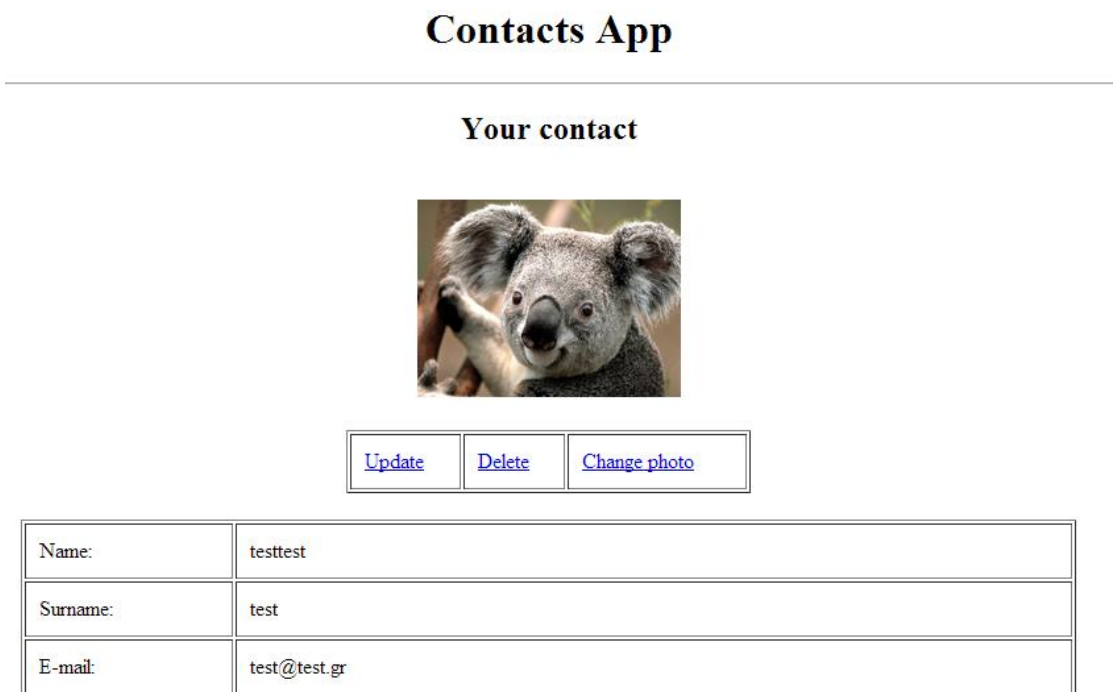
Εδώ βλέπουμε την ίδια καρτέλα με πριν, αλλά με τη δυνατότητα να μπορούμε να συμπληρώσουμε τα πεδία με ό,τι θέλουμε, όπως όταν πραγματοποιήσαμε την αρχική εγγραφή. Χάρην ευκολίας, φαίνονται στα πεδία που έχουν ήδη κάποια τιμή, οι τιμές τους ώστε να διευκολύνεται ο χρήστης στην διόρθωση, αντικατάσταση ή εκ νέου εισαγωγή πληροφοριών στην καρτέλα της επαφής. Όταν ολοκληρώσουμε τις αλλαγές μας, πατάμε το κουμπί Update.

Βλέπουμε ότι μας επιστρέφει στην προηγούμενη σελίδα, έχοντας πραγματοποιήσει την αλλαγή που κάναμε.

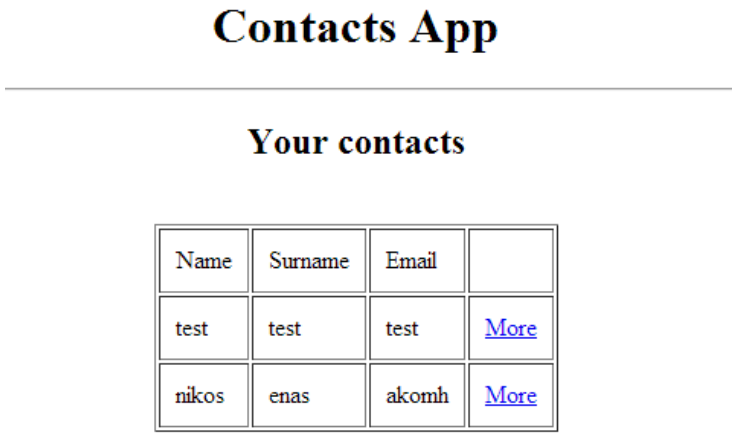


Στη συνέχεια θα δούμε την επιλογή Change Photo. Επιλέγοντας το μας πηγαίνει στην παρακάτω οθόνη όπου μπορούμε να επιλέξουμε μια εικόνα και να την «ανεβάσουμε» στην εφαρμογή.

Πατόντας το κουμπί Upload μας επιστρέφει στην ίδια σελίδα και πρέπει να πατήσουμε το “Back” για να μας επιστρέψει στην καρτέλα της επαφής. Εκεί βλέπουμε ότι η default εικόνα έχει αντικατασταθεί από την εικόνα που επιλέξαμε. Εδώ να σημειώσουμε ότι η εικόνα προσαρμόζεται σε συγκεκριμένο μέγεθος και διαστάσεις, διατηρόντας τον λόγο αναλογιών της.



Τελευταία επιλογή το Delete, το οποίο διαγράφει την επαφή από την εφαρμογή. Πατόντας το link μας μεταφέρει στην λίστα με τις επαφές μας, όπου και βλέπουμε ότι έχει διαγραφεί.

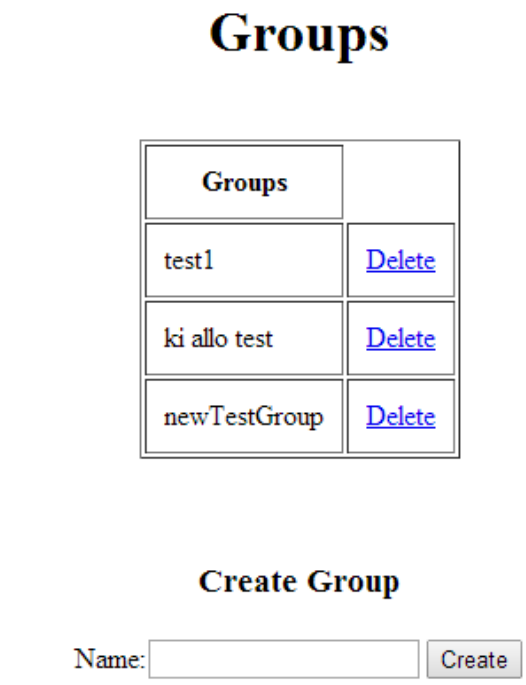


Επιστρέφοντας στην κεντρική σελίδα της εφαρμογής, επόμενη επιλογή είναι το “My Contacts”, το οποίο όμως είναι η σελίδα που είδαμε μόλις, στη δημιουργία της επαφής, όπως και στη διαγραφή της.

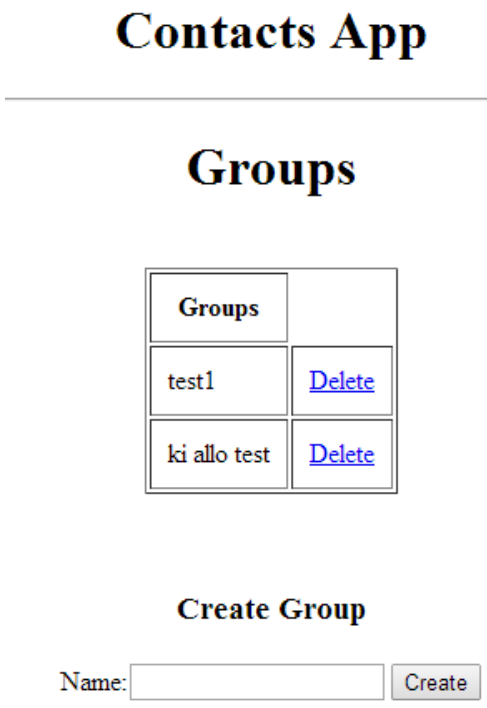
Επόμενη επιλογή είναι το “Groups”, το οποίο μας μεταφέρει στην παρακάτω σελίδα. Εδώ έχουμε τη δυνατότητα να δούμε ποια groups επαφών υπάρχουν ήδη, να δημιουργήσουμε καινούρια, καθώς και να σβήσουμε παλαιότερα.

Χάρην παραδείγματος, θα δημιουργήσουμε ένα νέο group με την ονομασία newTestGroup, χρησιμοποιόντας την φόρμα “Create Group”.

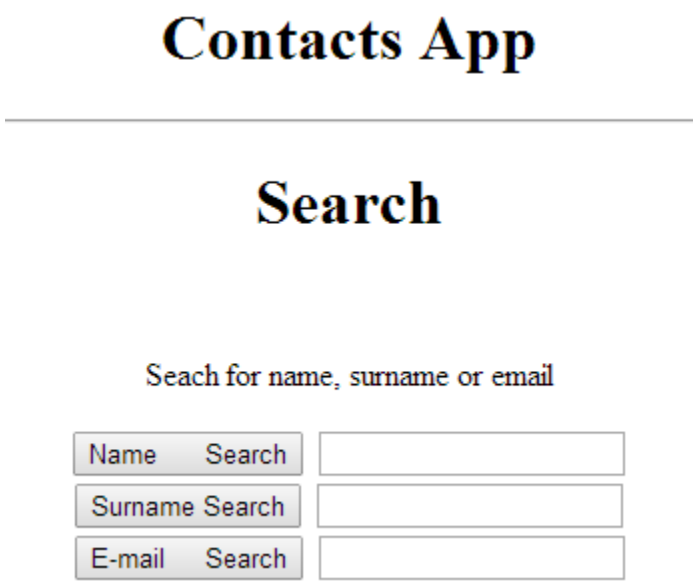
Παρακάτω βλέπουμε τα αποτελέσματα αυτής μας της ενέργειας.



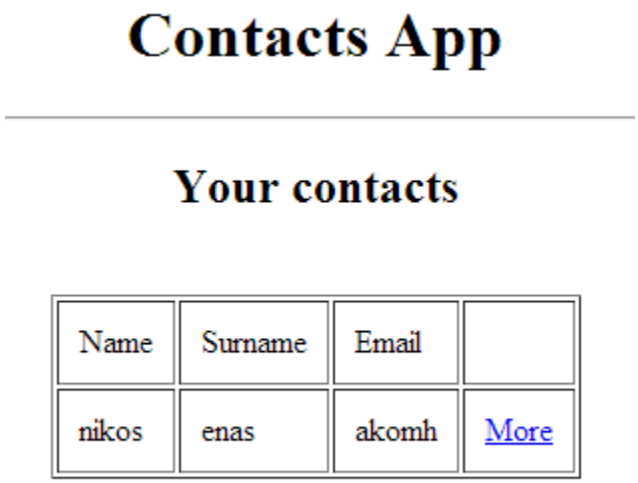
Στη συνέχεια θα διαγράψουμε αυτό το group, πατόντας το link “Delete” δίπλα από το όνομα του. Το αποτέλεσμα το βλέπουμε παρακάτω.



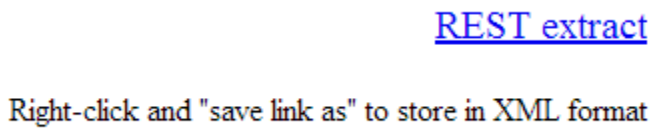
Έχοντας ολοκληρώσει και την περιγραφή αυτής της δυνατότητας της εφαρμογής, επιστρέφουμε στην κεντρική σελίδα ώστε να δούμε την τελευταία από τις βασικές δυνατότητες της, η οποία είναι η αναζήτηση επαφών (Search).



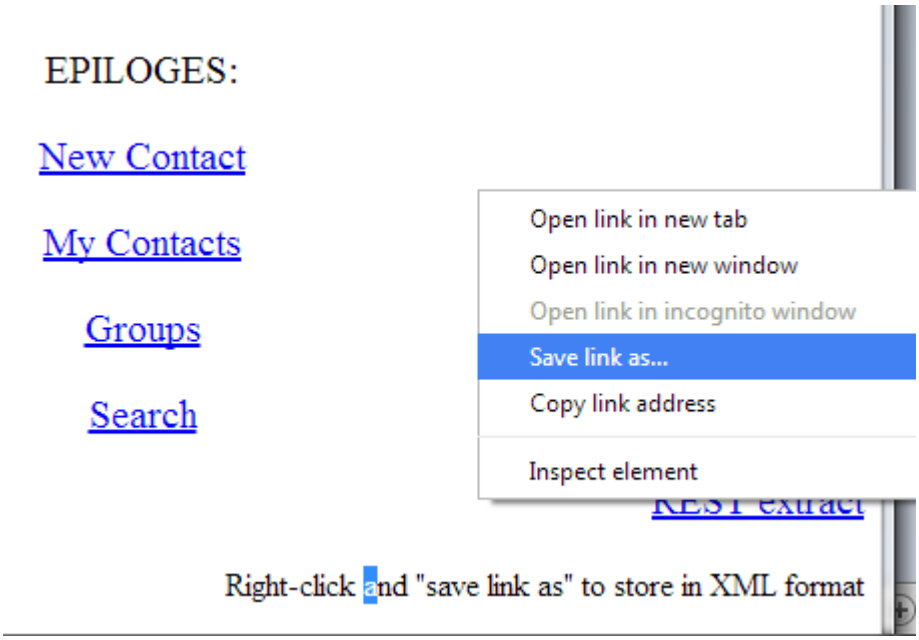
Εδώ μπορούμε να κάνουμε αναζήτηση στις επαφές μας αναλόγως με την πληροφορία που διαθέτουμε ή ψάχνουμε. Εμείς θα κάνουμε μια δοκιμή με το όνομα “Nikos”. Σαν αποτέλεσμα θα μας εμφανίζει μια λίστα με τις επαφές που ταιριάζουν σ’αυτή μας την αναζήτηση. Πλέον σ’αυτή τη λίστα, έχουμε τις ίδιες δυνατότητες με την επιλογή “My Contacts” που είδαμε προηγουμένως στην κεντρική σελίδα της εφαρμογής.



Τέλος, στην κεντρική σελίδα της εφαρμογής, κάτω δεξιά, υπάρχει η επιλογή να μας εμφανιστεί και να αποθηκεύσουμε τα δεδομένα της εφαρμογής μας, κατευθείαν από τη βάση δεδομένων. Η επιλογή είναι το “REST Extract” και χρησιμοποιεί ένα RESTful web service το οποίο τραβάει τα δεδομένα από τη βάση και τα επιστρέφει σε μορφή XML, μορφή που του επιτρέπει να μεταφέρονται τα δεδομένα σε ένα οποιοδήποτε άλλο πληροφοριακό σύστημα ή εφαρμογή.



Όπως αναφέρει και η υποσημείωση κάτω από το link μπορούμε να κάνουμε δεξί κλικ και αποθήκευση ώστε να «κατέβει» και να αποθηκευτεί σε ένα αρχείο XML.



Εναλλακτικά, αν πατήσουμε το link θα μας παρουασιαστούν τα δεδομένα της εφαρμογή σε μορφή XML, όπως παρακάτω:

This XML file does not appear to have any style information associated with it. The document tree is shown

```
▼<xml>
  ▼<item>
     <id>9</id>
     <email>test</email>
     <name>test</name>
     <surname>test</surname>
     <address>test</address>
     <bday/>
     <url/>
     <title/>
     <belong_group>5</belong_group>
     <notes/>
     <user_id>0</user_id>
     <time_stamp>2014-03-06 22:34:34</time_stamp>
     <ph_home>0</ph_home>
     <ph_mobile>0</ph_mobile>
     <ph_work>0</ph_work>
     <photo_id>0</photo_id>
   </item>
  ▼<item>
     <id>11</id>
     <email>akomh</email>
     <name>nikos</name>
     <surname>enas</surname>
     <address/>
     <bday/>
     <url/>
     <title/>
     <belong_group>5</belong_group>
     <notes/>
     <user_id>0</user_id>
     <time_stamp>2014-03-09 23:02:14</time_stamp>
     <ph_home>0</ph_home>
     <ph_mobile>0</ph_mobile>
     <ph_work>0</ph_work>
     <photo_id>0</photo_id>
   </item>
 </xml>
```

Με το παραπάνω ολοκληρώνεται η παρουσίαση της εφαρμογής.

6. Βιβλιογραφία

The Internet

- ✓ http://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CE%B4%CE%AF%CE%BA%CF%84%CF%85%CE%BF

Apache Server

- ✓ http://el.wikipedia.org/wiki/Apache_HTTP_%CE%B5%CE%BE%CF%85%CF%80%CE%B7%CF%81%CE%B5%CF%84%CE%B7%CF%84%CE%AE%CF%82

XAMPP

- ✓ http://el.wikipedia.org/wiki/XAMPP
- ✓ http://www.apachefriends.org/en/xampp.html

PHP

- ✓ http://el.wikipedia.org/wiki/Mod_php
- ✓ http://www.ip.gr/General/%CE%A4%CE%B9_%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9_%CE%B7_PHP-63.html

mySQL

- ✓ http://el.wikipedia.org/wiki/MySQL
- ✓ http://deltahacker.gr/2008/02/01/php-mysql-tutorial-the-fellowship-of-the-ring/

PHPMyAdmin

- ✓ http://www.phpmyadmin.net/home_page/index.php
- ✓ http://en.wikipedia.org/wiki/PhpMyAdmin

MVC

- ✓ http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

REST

- ✓ http://en.wikipedia.org/wiki/Representational_state_transfer
- ✓ http://edn.embarcadero.com/article/40467

CodeIgniter

- ✓ http://en.wikipedia.org/wiki/CodeIgniter