ΤΕΙ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ **ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ**

ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΔΙΑΧΕΙΡΙΣΗ ΣΥΝΑΝΤΗΣΕΩΝ ΜΕ ΧΡΗΣΗ Jquery Mobile.

ΞΕΡΟΥΤΣΙΚΟΣ ΑΝΔΡΕΑΣ

ΑΝΤΙΡΡΙΟ ΣΕΠΤΕΜΒΡΙΟΣ 2014

ΕΠΟΠΤΗΣ ΚΑΘΗΓΗΤΗΣ

ΧΡΙΣΤΟΔΟΛΟΥ ΣΩΤΗΡΙΟΣ

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΔΙΑΧΕΙΡΙΣΗ ΣΥΝΑΤΝΗΣΕΩΝ ΜΕ ΧΡΗΣΗ Jquery Mobile.

Πτυχιακή εργασία που υποβλήθηκε στο Α.Τ.Ε.Ι.

ΠΑΤΡΑΣ για την απόκτηση του πτυχίου

Υπό

ΤΡΙΑΝΤΑΦΥΛΛΟΥ ΒΑΣΙΛΕΙΟΣ

Εργασία η οποία έλαβε μέρος στο Τμήμα

Μηχανικών Πληροφορικής με την επίβλεψη

του ΧΡΙΣΤΟΔΟΥΛΟΥ ΣΩΤΗΡΙΟΥ

Τμήμα Μηχανικών Πληροφορικής

ΤΕΙ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

Αντίρριο

15 ΣΕΠΤΕΜΒΡΙΟΣ 2014

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία εκπονήθηκε από τον φοιτητή Ξερούτσικο Ανδρέα του τμήματος Μηχανικών Πληροφορικής του Ανώτατου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Πάτρας κατά το ακαδημαϊκό έτος 2014-2015 υπό την επίβλεψη του καθηγητή του τμήματος Χριστοδούλου Σωτηρίου.

Στο σημείο αυτό αισθάνομαι την ανάγκη να εκφράσω τις θερμές ευχαριστίες μου σε όσους συνέβαλαν στην ολοκλήρωση αυτής της προσπάθειας :

Πρώτα απ' όλα, στον επιβλέποντα καθηγητή μου Κ. Χριστοδούλου Σωτήρη για τη καθοδήγηση, την υποστήριξη, τις ουσιώδεις συμβουλές, καθώς και την ενθάρρυνση που μου παρείχε σε όλο αυτό το διάστημα.

Επίσης θα ήθελα να ευχαριστήσω το Γιώργο, το Χρήστο, το Βλάση και την Ευδοκία για τη πολύτιμη συμβολή και συμπαράσταση τους για τη περαίωση της εργασίας αυτης.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου για την ανυπολόγιστη ηθική υποστήριξη, την συμπαράσταση και την κατανόηση που έδειξαν όλον αυτόν τον καιρό.

ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας πτυχιακής εργασίας είναι να σχεδιαστεί και να υλοποιηθεί μια εφαρμογή για διαχείριση και υπενθύμιση συναντήσεων που θα εκτελείται σε όλες τις συσκευές smartphones ανεξαρτήτως λειτουργικού συστήματος. Η εφαρμογή έχει την μορφή web site και αποτελεί ουσιαστικά ένα περιβάλλον όπου υπάρχει η δυνατότητα εγγραφής χρηστών όπως και η πλήρη διαχείρηση των λογαριασμών και προφίλ αυτών. Στη συνέχεια υπάρχει η διαχείρηση των συναντήσεων που πραγματοποιείται ως εξής : ο χρήστης-συντονιστής οργανώνει συνάντηση στέλνοντας προσκλήσεις σε όσους θα συμμετέχουν καθοριζοντας ημερομηνία, ώρα και τόπο συνάντησης. Κάθε χρήστης που προσκαλείται στο ραντεβού, καθορίζει με τι μέσο θα μετακινηθεί, για να μπορεί το σύστημα να εκτιμήσει τον χρόνο που απαιτείται ανάλογα με τη θέση του.

Επιπλέον θα παρακολουθείται τη θέση όλων των χρηστών με χρήση GPS , θα υπολογίζει πότε πρέπει να ξεκινήσουν για μια συνάντηση ανάλογα με τη τοποθεσία, την τρέχουσα θέση τους και το μέσο που θα έχει επιλεχτεί (πόδια, αυτοκίνητο, μέσα μαζικής μεταφοράς), όπως και την εκτίμηση σε πόση ώρα θα φτάσουν.

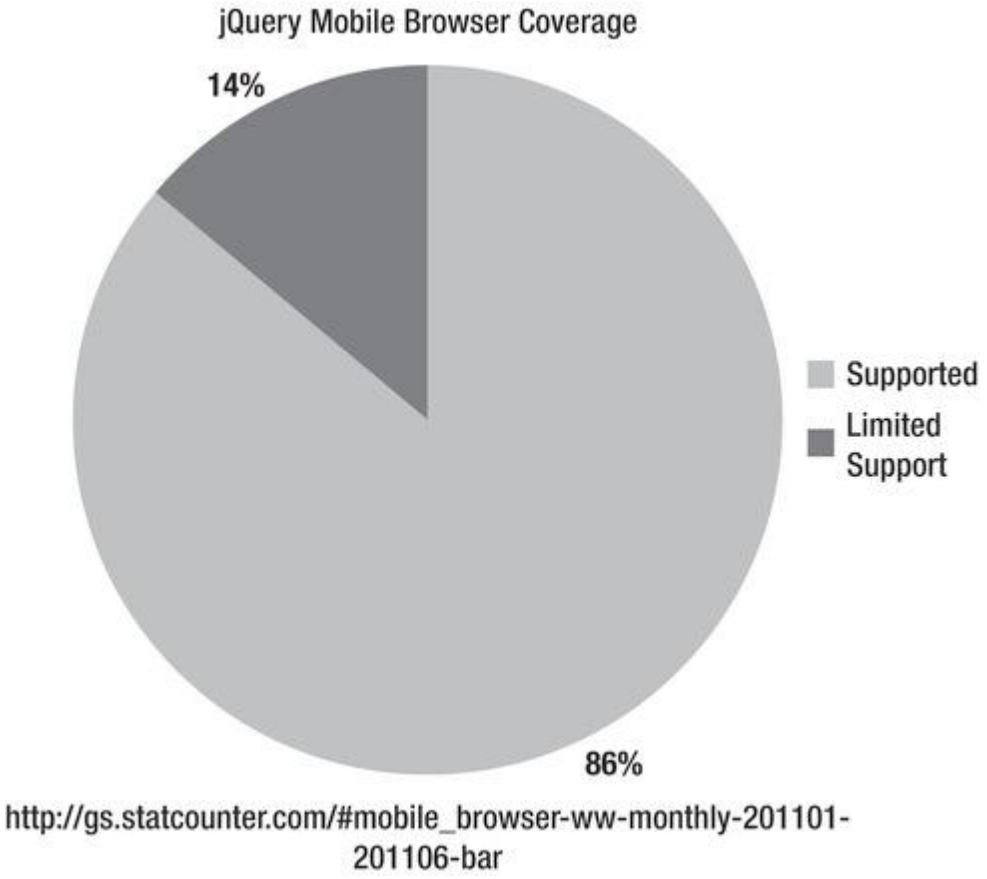
ΚΕΦΑΛΑΙΟ 1:

Γιατί Jquery Mobile ?

Τα στατιστικά στοιχεία δείχνουν ένα άνευ προηγουμένου ποσοστό έγκρισης, καθιστώντας το mobile web ένα πολύ καυτό θέμα και απαιτεί ένα νέο σύνολο δεξιοτήτων από web developers και designers. Η χρήση κινητών συσκευών έχει εκτοξευθεί στα ύψη σύμφωνα με την έκθεση της Nielsen Mobile Media, "44 τοις εκατό των συνδρομητών κινητής τηλεφωνίας των ΗΠΑ έχει τώρα smartphone συσκευή, σε σύγκριση με 18 τοις εκατό πριν από δύο χρόνια. «Ο αριθμός των συνδρομητών smartphone χρησιμοποιώντας το Internet μέσω κινητού έχει αυξηθεί 45 τοις εκατό από το 2010. Καταλαβαίνουμε λοιπόν ότι η κατοχή ενός Smartphone τη σημερινή μέρα αποτελεί ένα σύνηθες γεγονός. Αυτό εν συνεχεία προυποθέτει την εύκολη χρήση, κατανόηση και ανάπτυξη όλων των εφαρμογών αλλά και του δικτυακού χώρου που αποτελούν αναπόσπαστο μέρος μιας έξυπνης κινητής συσκευής. Φανταστείτε όλες τις συσκευές που θα μπορούσαν να έχουν πρόσβαση στον ιστότοπό σας. Στη συνέχεια, φανταστείτε ότι χρειάζεται να δοκιμάσετε όλες αυτές τις πλατφόρμες κάθε φορά που χτίζετε μια mobile ιστοσελίδα. Αυτό θα είναι επίπονο και απίστευτα χρονοβόρο. Το jQuery Mobile είναι ενα νέο , απλό στη χρήση , framework UI για τη δημιουργία cross-platform Mobile Web εφαρμογών . Είναι μια ισχυρή αλλά ελαφριά βιβλιοθήκη JavaScript που απλοποιεί τη JavaScript κωδικοποίηση και επεκτείνει τις δυνατότητες του Cascading Style Sheets (CSS). Επιπλέον, εξαλείφει τα θέματα συμβατότητας περί cross-browser και είναι συμβατό με CSS3. Αυτό σημαίνει ταχύτερο scripting, λιγότερες δοκιμές, και λιγότερος κώδικας για τους διαφορετικούς τρόπους που οι browsers λειτουργούν.Σε λίγα λεπτά , μπορείτε να δημιουργήσετε κινητές εφαρμογές (apps) που έχουν βελτιστοποιηθεί για να τρέξει σχεδόν σε κάθε τηλέφωνο , tablet , desktop , και η συσκευή e –reader διατίθενται σήμερα.Αυτό συμβαίνει επειδή το framework jQuery mobile παίρνει μια προοδευτική προσέγγιση, η οποία δεν φέρνει μόνο πλούσιες διαδραστικές εμπειρίες με τις συσκευές, αλλά παρέχει επίσης υποστήριξη για παλαιότερα προγράμματα περιήγησης και τηλέφωνα.Οi Jquery Mobile εφαρμογές είναι καθολικά προσβάσιμες σε όλες τις συσκευές με ένα πρόγραμμα περιήγησης.Χρηματοδοτείται σήμερα από εταιρείες όπως η Mozilla, Palm, Adobe, Nokia, BlackBerry, και πολλά άλλα. Πλέον, λειτουργεί αδιάλειπτα σε όλες τις δημοφιλείς πλατφόρμες κινητών συσκευών.

Καθολική Πρόσβαση

Οι Jquery Mobile εφαρμογές είναι καθολικά προσβάσιμες σε όλες τις συσκευές με ένα πρόγραμμα περιήγησης. Αυτό είναι ένα μεγάλο πλεονέκτημα. (βλ. Σχήμα 1-1). Σχεδόν κάθε κινητή συσκευή έχει ένα πρόγραμμα περιήγησης. Εάν η εφαρμογή σας είναι καθολικά προσβάσιμη σε αυτό το ευρύ φάσμα είναι ένα σημαντικό ανταγωνιστικό πλεονέκτημα. Η παρακάτω είναι μια πλήρη λίστα των υποστηριζόμενων συσκευών σε jQuery Mobile 1.0, το οποίο περιλαμβάνει τα περισσότερα τηλέφωνα, tablets, desktop browsers, και ακόμη και e-readers.

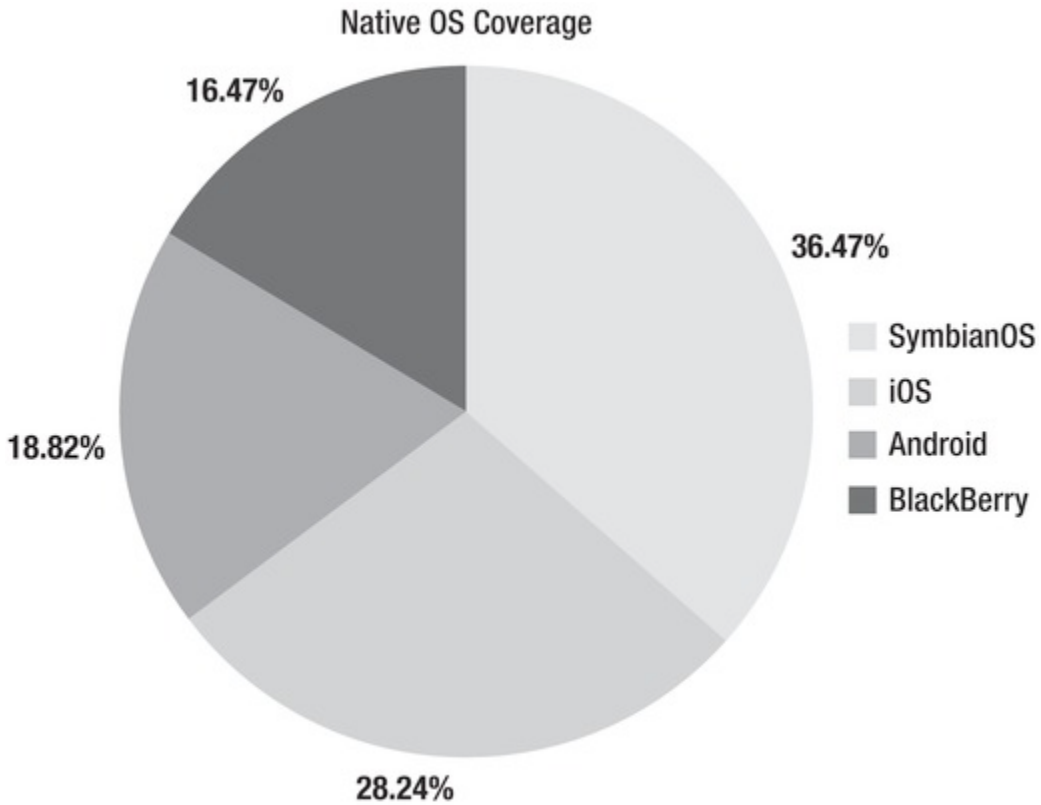


Σχήμα 1-1

Supported Devices:

- Phones/Tablets , Android 1.6+ , BlackBerry 5+, iOS 3+,
- Windows Phone 7
- WebOS 1.4+
- Symbian (Nokia S60)
- Firefox Mobile Opera Mobile 11+
- Opera Mini 5+
- Desktop browsers
- Chrome 11+
- Firefox 3.6+
- Internet Explorer 7+
- Safari
- e-readers
- Kindle
- Nook

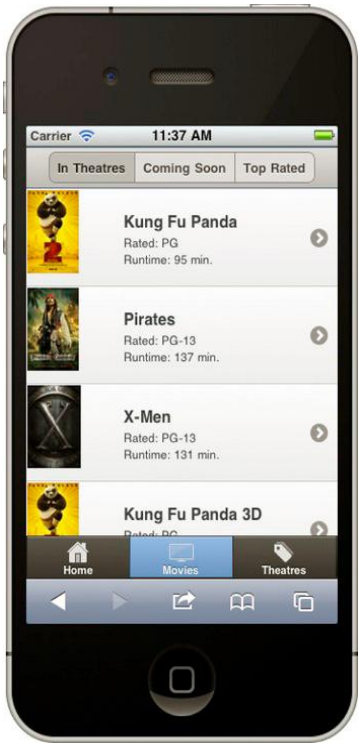
Συγκριτικά, η ανάπτυξη native εφαρμογών έχει ένα πολύ περιοριστικό μοντέλο διανομής (βλ. Σχήμα 1-2). Οι native εφαρμογές είναι διαθέσιμες μόνο στο native λειτουργικό τους σύστημα. Για παράδειγμα, μια iPhone εφαρμογή είναι προσβάσιμη μόνο από μια συσκευή iOS. Ευτυχώς, οι jQuery Mobile εφαρμογές που δεν περιορίζονται από αυτό το κανόνα..



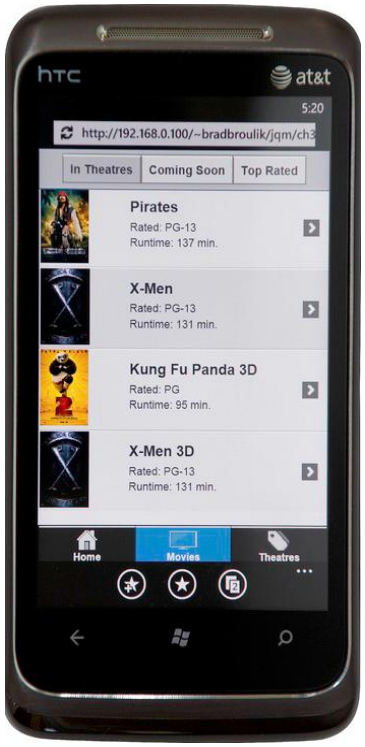
http://gs.statcounter.com/#mobile_os-ww-monthly-201101-201106-bar

Ενιαίο UI σε όλες τις πλατφόρμες κινητών

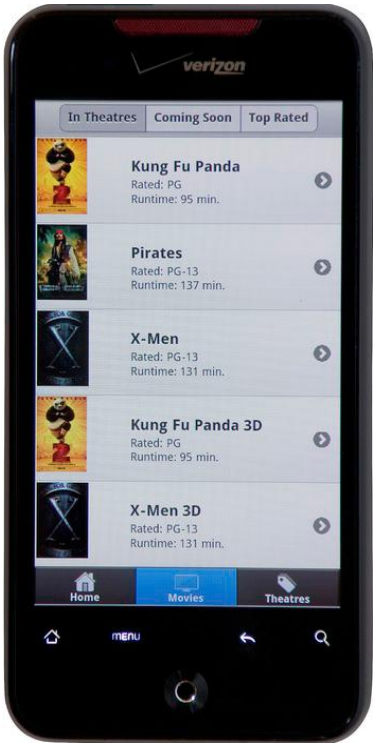
Το jQuery Mobile προσφέρει ένα ενιαίο περιβάλλον εργασίας για το χρήστη σχεδιάζοντας με την βοήθεια των HTML5 και CSS3 προτύπων. Οι χρήστες κινητών τηλεφώνων περιμένουν την εμπειρία στις συσκευές τους να είναι συνεπής σε όλες τις πλατφόρμες (βλέπε Εικόνα 1-3, Εικόνα 1-4, Εικόνα 1-5). Αντίθετα, παρατηρώντας τις native Twitter εφαρμογές στις δύο συσκευές iPhone και Android παρατηρούμε ότι η εμπειρία δεν είναι η ίδια. Οι εφαρμογές jQuery Mobile έχουν διορθώσει αυτό το πρόβλημα, παρέχοντας μια εμπειρία για το χρήστη που είναι οικεία και εύχρηστη, ανεξάρτητα από την πλατφόρμα. Επιπλέον, ένα ενιαίο περιβάλλον εργασίας χρήστη θα εξασφαλίσει συνεπή τεκμηρίωση, στιγμιότυπα οθόνης ανεξάρτητα από την πλατφόρμα του τελικού χρήστη.



Εικόνα 1-3

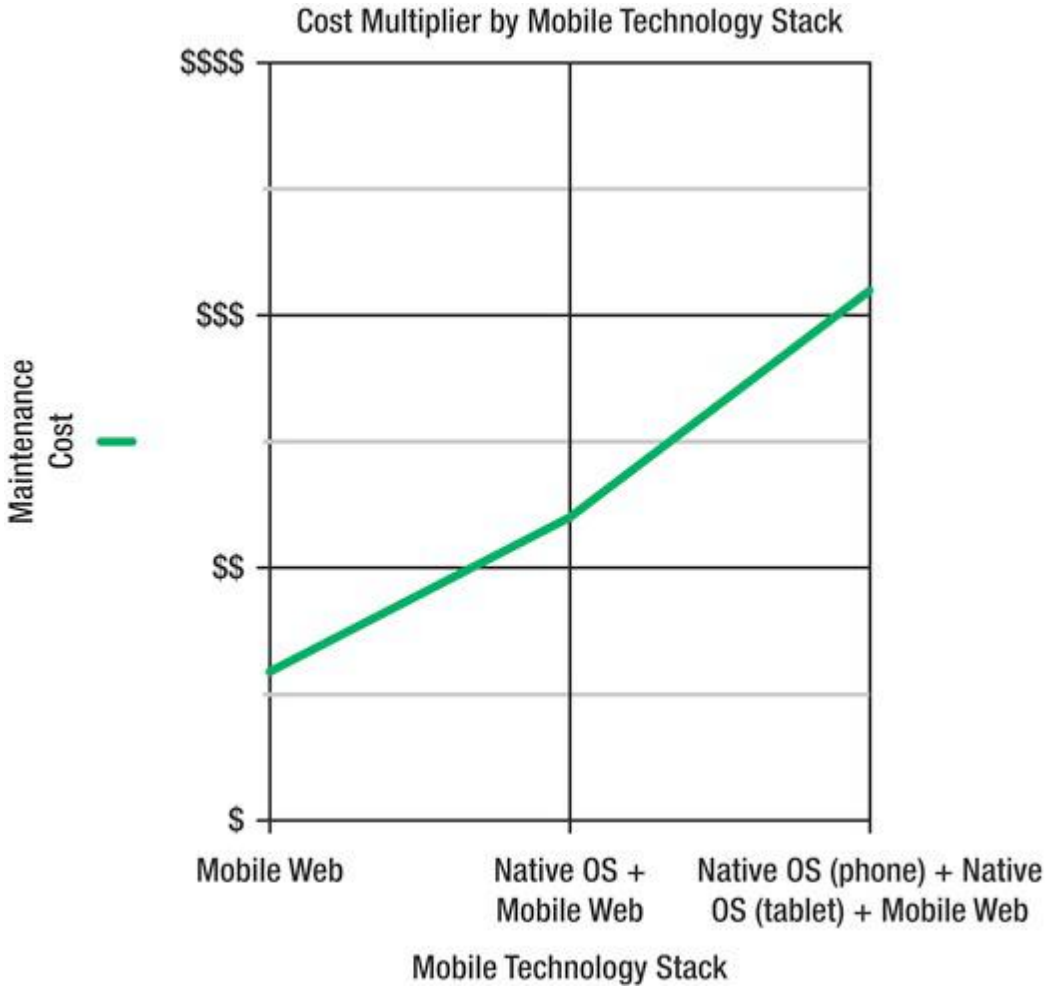


Εικόνα 1-4



Εικόνα 1-5

Το Jquery Mobile βοηθά επίσης να εξαλείψει την ανάγκη για συγκεκριμένες προσαρμογές UI στη συσκευή. Ένας απλός κώδικας jQuery Mobile θα αποδίδει σταθερά σε όλες τις υποστηριζόμενες πλατφόρμες χωρίς προσαρμογές, έτσι μακροπρόθεσμα γίνεται πιο αποδοτικός όσον αφορά στην στήριξη και το κόστος συντήρισης. (βλέπε Σχήμα 1-6).



Σχήμα 1-6

Απλοποιημένη Markup-Driven Ανάπτυξη

Οι jQuery Mobile σελίδες δομούνται με HTML5 σήμανση (βλ. Καταχώρηση 1-1). Εκτός από τα νέα χαρακτηριστικά της, η μετάβαση σε HTML5 jQuery Mobile θα πρέπει να είναι μια σχετικά ομαλή μετάβαση. Σε ότι αφορά την JavaScript και CSS, το jQuery Mobile κάνει όλη τη δουλειά από προεπιλογή. Ωστόσο, υπάρχουν περιπτώσεις όπου μπορεί να χρειαστεί να επικαλεστεί

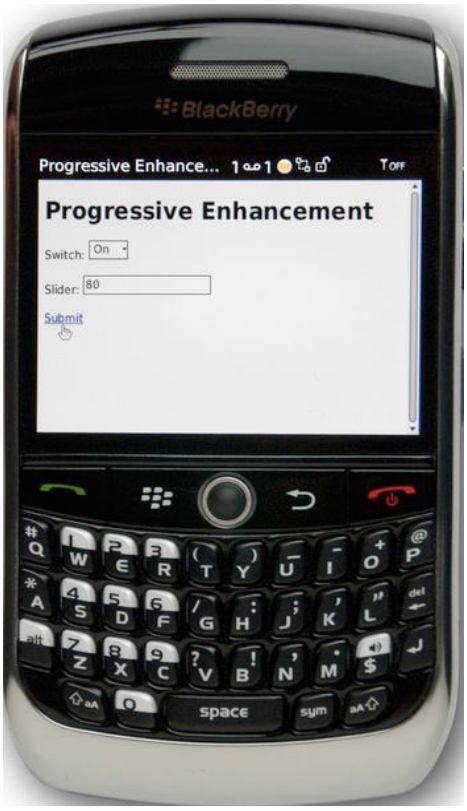
κανείς κώδικα JavaScript για τη δημιουργία μιας πιο δυναμικής ή ενισχυμένης σελίδας. Εκτός από την απλότητα της σήμανσης που απαιτείται για το σχεδιασμό των σελίδων επιτρέπει επιπλέον την ταχεία προτυποποίηση των διεπαφών χρήστη. Πολύ γρήγορα μπορούμε να δημιουργήσουμε μια στατική ροή απο λειτουργικές σελίδες, μεταβάσεις, και widgets

Καταχώρηση 1–1.

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Title</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="jquery.mobile-1.0.min.css" />
  <script type="text/javascript" src="jquery-1.6.2.min.js"></script>
  <script type="text/javascript" src="jquery.mobile-1.0.min.js"></script>
</head>

<body>

<div data-role="page">
    <div data-role="header">
        <h1>Page Header</h1>
</div>

  <div data-role="content">
        <p>Hello jQuery Mobile!</p>
  </div>

<div data-role="footer">
        <h4>Page Footer</h4>
  </div>

</div>

</body>
</html>
```

Προοδευτική Ενίσχυση

Το Jquery Mobile θα αποδώσει τη πιο κομψή δυνατή εμπειρία χρήστη για μια συσκευή. Παραδείγματος χάρη κοιτάξτε το διακόπτη ελέγχου jQuery Mobile στο Σχήμα 1-7



Σχήμα 1-7

Σχήμα 1-8

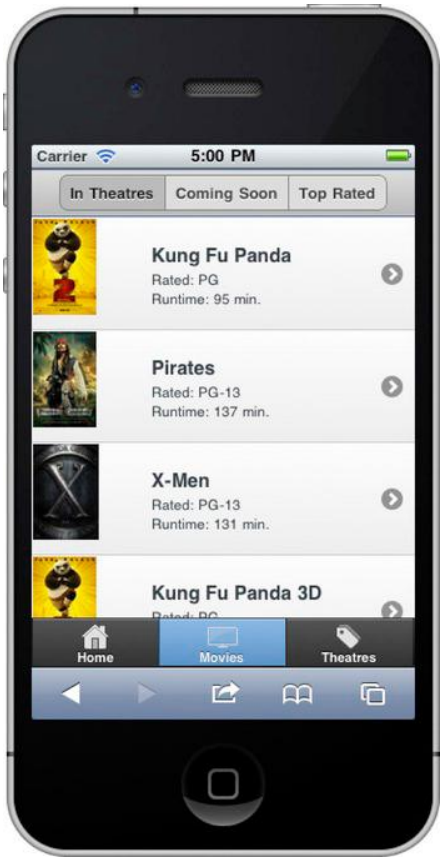
Το jQuery Mobile αποδίδει το switch έλεγχο εφαρμάζοντας style CSS3. Εναλλακτικά, στο Σχήμα 1-8 είναι το ίδιο switch ελέγχου σε παλιότερη έκδοση προγράμματος περιήγησης, όπου και δεν αποδίδει το πλήρες styling CSS3.

Στις περισσότερες περιπτώσεις, αν η συσκευή σας δεν υποστηρίζει ένα χαρακτηριστικό μιας native εφαρμογης που δεν θα υπάρχει η δυνατότητα να

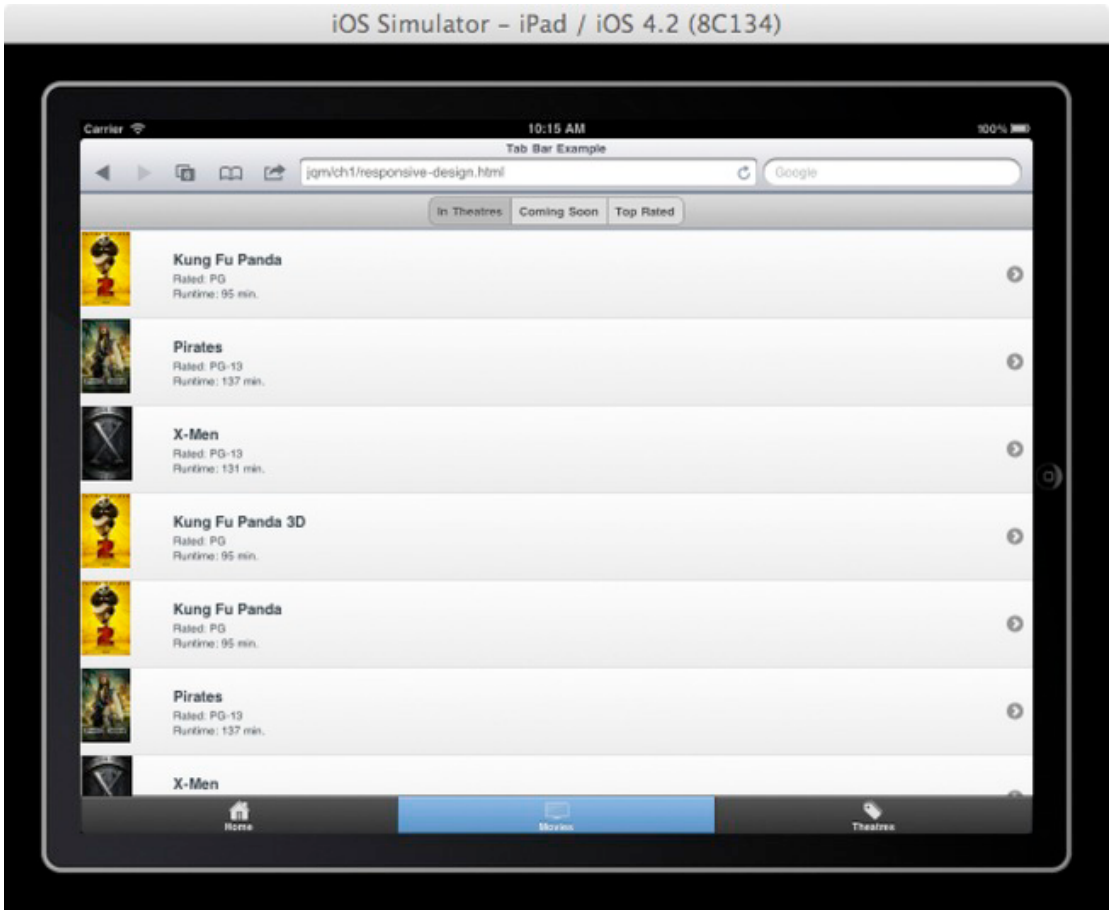
κατεβάσετε την εφαρμογή. Για παράδειγμα, ένα νέο χαρακτηριστικό του iOS 5 είναι το iCloud persistence. Αυτό το νέο χαρακτηριστικό απλοποιεί το συγχρονισμό των δεδομένων σε πολλαπλές συσκευές. Αν δημιουργήσουμε μια νέα iOS εφαρμογή που ενσωματώνει αυτό το νέο χαρακτηριστικό, θα πρέπει να ρυθμίσουμε το "ελάχιστο επιτρεπτό SDK "για την εφαρμογή μας έκδοσης 5.0. Οι εφαρμογές jQuery Mobile είναι πιο ευέλικτες από την άποψη αυτή.

Αποδοτικός Σχεδιασμος

Ένα jQuery Mobile UI θα αποδόσει υπεύθυνα σε διάφορα μεγέθη οθονών. Για παράδειγμα, το ίδιο UI θα εμφανίσει κατάλληλα για κινητά τηλέφωνα (βλ. Εικόνα 1-9) ή μεγαλύτερες συσκευές, όπως όπως tablets, επιτραπέζιοι υπολογιστές, τηλεοράσεις ή (βλέπε Εικόνα 1-10).



Εικόνα 1-9



Εικόνα 1-10

«Build once, run anywhere»

Είναι δυνατόν να οικοδομήσουμε μια εφαρμογή η οποία είναι καθολικά διαθέσιμη για όλους τους καταναλωτές (κινητά τηλέφωνα, υπολογιστές, ταμπλέτες); Ναι, αυτό είναι δυνατό. Το Jquery Mobile παρέχει υποστήριξη cross-browser, δηλαδή τη δυνατότητα προσαρμογής της λειτουργικότητας μιας ιστοσελίδας, εφαρμογής ή client-side κώδικα σε περιβάλλοντα που παρέχουν όλα τα αναγκαία χαρακτηριστικά ή ακόμα και οταν υπάρχει έλλειψη αυτών. Για παράδειγμα, μικρές συσκευές μπορούν να εξυπηρετήσουν μικρές εικόνες με σύντομη περιεχόμενο, ενώ μεγαλύτερες συσκευές μπορούν να εξυπηρετήσουν μεγαλύτερες εικόνες με πιο αναλυτικό περιεχόμενο. Σήμερα, οι περισσότερες οργανώσεις με παρουσία στο κόσμο του mobile τυπικά υποστηρίζουν εξίσου ένα desktop Web και μια ιστοσελίδα κινητού. Είναι πλεον

χάσιμο χρόνου η υποστήριξη πολλαπλών διανομών μιας εφαρμογής. Ο ρυθμός με τον οποίο οι οργανισμοί αγκαλιάζουν την εξέλιξη του mobile, σε συνδυασμό με την ανάγκη τους για αποφυγή σπατάλης χρόνου, θα οδηγήσει στο μύθο «build once, run anywhere»

Αποδοτικές Φόρμες

Σε ορισμένες περιπτώσεις, το jQuery Mobile θα δημιουργήσει αποδοτικά σχέδια για σας. Οι ακόλουθες εικόνες δείχνουν πόσο καλά ανταποκρίνεται ο σχεδιασμός jQuery Mobile για οριζόντια και κάθετη τοποθέτηση. Για παράδειγμα, κατά την κάθετη απεικόνηση (Βλ. Εικόνα 1-11), οι ετικέτες τοποθετούνται πάνω από τα πεδία της φόρμας. Εναλλακτικά, στην οριζόντια απεικόνηση (βλ. Εικόνα 1-12) τα πεδία της φόρμας και ετικέτες εμφανίζονται δίπλα το ένα στο άλλο. Αυτή η αποδοτική-πρακτική σχεδίαση παρέχει μια ευχάριστη εμπειρία για το χρήστη. Το Jquery Mobile προσφέρει πολλά από αυτα τα χαρακτηριστικα για σας χωρίς καμία προσπάθεια από μέρους σας!

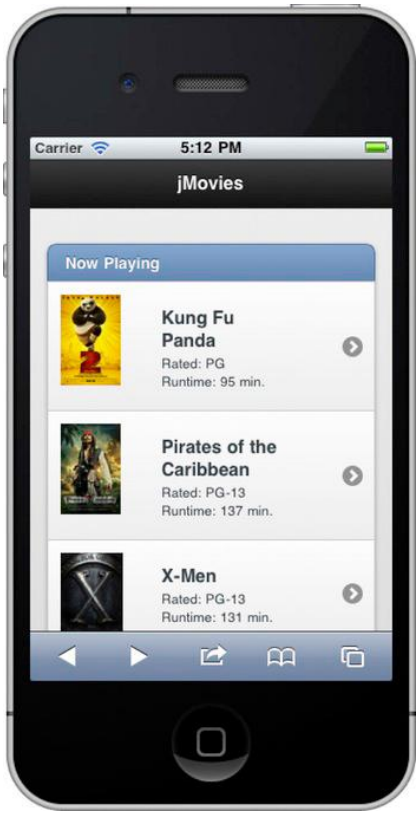
Εικόνα 1-11.Κάθετη απεικόνηση

Εικόνα 1-12.Οριζόντια απεικόνηση

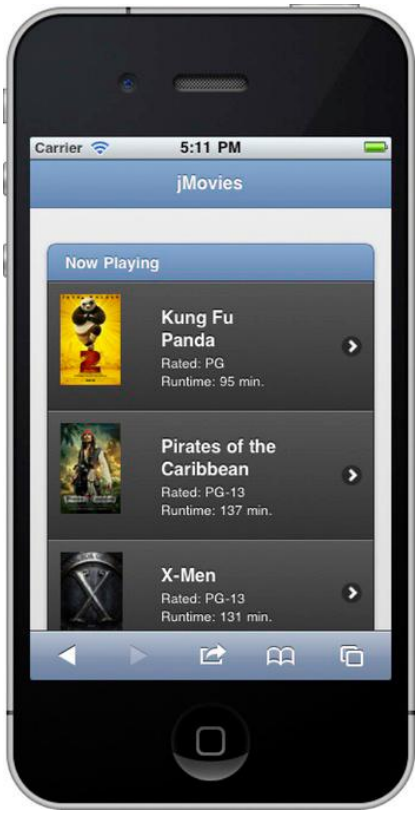
Θεματικός Σχεδιασμός

Το Jquery Mobile υποστηρίζει ενα θεματικό σχεδιασμό που επιτρέπει στους σχεδιαστές να δομήσουν γρήγορα το στυλ του UI τους. Το framework παρέχει αυτόματα πέντε θεματικά σχέδια με την δυνατότητα εναλλαγής θεμάτων για όλα τα συστατικά, συμπεριλαμβανομένης της σελίδας, κεφαλίδα, το περιεχόμενο, και το υποσέλιδο. Το πιο χρήσιμο εργαλείο για τη δημιουργία προσαρμοσμένων θεμάτων είναι το ThemeRoller3. Ο επανασχεδιασμός ενός UI χρειάζεται την ελάχιστη δυνατή προσπάθεια. Για παράδειγμα, μπορώ να πάρω γρήγορα ένα προεπιλεγμένο θέμα jQuery Mobile (βλ. Εικόνα 1-13) και να την ξαναδομίσω με ένα νέο θέμα σε δευτερόλεπτα. Στην περίπτωση των τροποποιημένων μου θέμα (βλ. Σχήμα 1-14), επέλεξα μια εναλλακτική θέμααπό τη λίστα. Η μόνη σήμανση που χρειαζόταν ήταν μία προσθήκη δεδομένων τύπου : θέμα χαρακτηριστικό=”**a**”.

<!-- Set the lists background to black -->
<ul data-role="listview" data-inset="true" **data-theme="a"**>



Εικόνα 1-13.Θέμα προεπιλογής



Εικόνα 1-14 .Εναλλακτικό θέμα

Προσβασιμότητα

Οι Jquery Mobile εφαρμογές είναι συμβατό με το 508 από προεπιλογή, ένα χαρακτηριστικό το οποίο είναι πολύτιμο για όλους. Ειδικότερα το 508 είναι ένας ομοσπονδιακός νόμος που απαιτεί οι εφαρμογες να είναι προσβάσιμες από χρήστες με αναπηρίες. Οι πιο συχνά χρησιμοποιούμενες βοηθητικές τεχνολογίες για το κινητό Web είναι αναγνώστες οθόνης.

Ειδικότερα, η κυβέρνηση ή κρατικές υπηρεσίες απαιτούν από τις εφαρμογές τους για να είναι 100% προσβάσιμες. Επιπλέον, η χρήση αναγνώστη οθόνη του κινητού αυξάνεται. Σύμφωνα με την WebAIM, 66,7% των χρηστών ανάγνωσης οθόνης χρησιμοποιούν τον αναγνώστη οθόνης για την κινητή συσκευή τους.

Συνοψίζοντας ας δούμε τα σημαντικότερα χαρακτηριστικά που κάνουν το jQuery Mobile μοναδικό :

1. Οι Jquery Mobile εφαρμογές είναι καθολικά διαθέσιμες σε όλες τις συσκευές με πρόγραμμα περιήγησης και έχουν βελτιστοποιηθεί για να "τρέξουν" σχεδόν σε κάθε τηλέφωνο , tablet ,desktop , και e -reader συσκευή.

2. Μπορούν να επωφεληθούν από την άμεση ανάπτυξη των δυνατοτήτων στο Web .

3. Ένας απλός κώδικας jQuery Mobile μπορεί να αποδίδει σταθερά σε όλες υποστηριζόμενες πλατφόρμες χωρίς προσαρμογές .

4. Το Jquery Mobile είναι ενα απλοποιημένο framework σήμανσης . Μπορεί να χτίσετε jQuery Mobile εφαρμογές με 100 % σήμανση.

5. Το Jquery Mobile χρησιμοποιεί προηγμένες τεχνικές βελτίωσης για να καταστήσει μια πλούσια εμπειρία για όλες τις συσκευές και παρέχει εμπειρία για τους ηλικιωμένους περιηγητές.

6. 'Ενα jQuery Mobile UI αποδίδει αποτελεσματικά μεταξύ των συσκευών διαφόρων μεγεθών, συμπεριλαμβανομένων τηλεφώνων , ταμπλετών , επιτραπέζιων υπολογιστών ή και τηλεοράσεων .

7. Το jQuery Mobile υποστηρίζει ένα θεματικό σχεδιασμό που επιτρέπει στους σχεδιαστές να ξανασχεδιάσουν γρήγορα το UI σε παγκόσμιο επίπεδο .

ΚΕΦΑΛΑΙΟ 2:

Γνωρίζοντας το Jquery Mobile

Στο κεφάλαιο 1, εξετάσαμε τα χαρακτηριστικά που κάνουν το jQuery Mobile μοναδικό. τώρα θα εξετάσουμε τα βασικά στοιχεία του jQuery Mobile. Θα ξεκινήσουμε με μια γενική επισκόπηση του προτύπου σελίδας jQuery Mobile. Στη συνέχεια, θα δούμε πώς το Jquery Mobile βελτιώνει τη σήμανση στη κινητή τηλεφωνία. Επίσης, θα εξετάσουμε πώς στο Jquery λειτουργεί η πλοήγηση.

Πρότυπη Σελίδα Jquery Mobile

Ένα πρότυπο σελίδας Jquery Mobile φαίνεται στη καταχώρηση 2-1. Μπορείτε να αντιγράψτε το πρότυπο HTML και να το επικολλήσετε σε ένα editor και στη συνέχεια να ανοίξετε το από το αγαπημένο σας πρόγραμμα περιήγησης. Το πρότυπο είναι σημασιολογική HTML5 και περιέχει χαρακτηριστικα jQuery Mobile όπως και αρχεία του CSS, JAVASCRIPT.

Καταχώρηση 2–1. *jQuery Mobile Page Template*

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Title</title>
1
 <meta name="viewport" content="width=device-width, initial-scale=1"> 2
<link rel="stylesheet" type="text/css" href="jquery.mobile.css" />
3  <script type="text/javascript" src="jquery.js"></script>
4  <!--<script src="custom-scripts-here.js"></script>-->
5  <script type="text/javascript" src="jquery.mobile.js"></script>
  </head>
  <body>
6  <div data-role="page">
7  <div data-role="header">
  <h1>Page Header</h1>
  </div>
8   <div data-role="content">
  <p>Hello jQuery Mobile!</p>
   </div>
9   <div data-role="footer">
```

```
  <h4>Page Footer</h4>
  </div>
  </div>
</body>
```

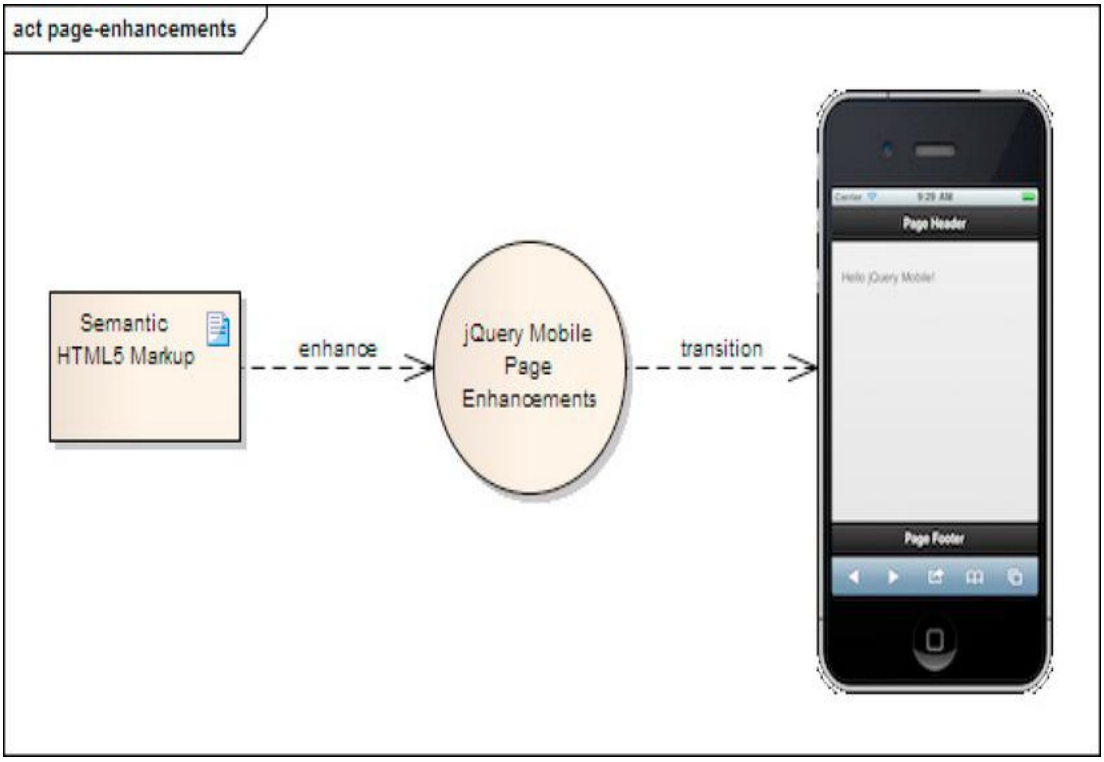
1 . Αυτή είναι η συνιστώμενη διαμόρφωση για jQuery Mobile . Η αξία «device-width» υποδεικνύει ότι το περιεχόμενο κατανεμέιται σε όλο το πλάτος της συσκευής
2 . Το CSS του Jquery Mobile θα εφαρμόσει στυλιστικές βελτιώσεις για όλους τους περιηγητές.
3 . Η βιβλιοθήκη jQuery είναι μια βασική εξάρτηση της jQuery Mobile και συνιστάται η χρήση του πυρήνα API jQuery εντός των Mobile σελίδων σας, αν η εφαρμογή σας απαιτεί πιο δυναμική συμπεριφορά .
4 . Αν χρειαστεί να παρακάμψετε την προεπιλεγμένη ρύθμιση jQuery Mobile, μπορείτε να εφαρμόσετε σας προσαρμογές εδώ.
5 . Η βιβλιοθήκη jQuery Mobile JavaScript πρέπει να δηλώνονται μετά το jQuery. Η βιβλιοθήκη jQuery Mobile είναι η καρδιά που ενισχύει ολόκληρη η κινητή εμπειρία .
6 . Το data- role = "page" προσδιορίζει το «περιεχόμενο» σελίδας για μια σελίδα jQuery Mobile. Αυτό το στοιχείο απαιτείται μόνο κατά την κατασκευή πολλών σελίδων σχέδια (βλέπε Λίστα 2-3) .
7 . Το data - role = " header " είναι η μπάρα κεφαλίδας ή τίτλου , όπως φαίνεται στην Εικόνα 2-1 . Αυτό το χαρακτηριστικό είναι προαιρετικό.
8 . Το data-role = "content" είναι το «δοχείο» περιτύλιγμα για το σώμα του περιεχομένου. Αυτό το χαρακτηριστικό είναι προαιρετικό.
9 . Το data-role = "footer" περιέχει τη γραμμή υποσέλιδου, όπως φαίνεται στο Σχήμα 2-1. Αυτό το χαρακτηριστικό είναι προαιρετικό

Εικόνα 2-1. Hello Jquery Mobile

Jquery Mobile Βελτιώσεις

Πώς το Jquery Mobile ενισχύει τη σήμανσης για μια βελτιστοποιημένη εμπειρία κινητής τηλεφωνίας;

1 . Πρώτον, το Jquery Mobile θα φορτώσει τη σημασιολογική σήμανση HTML (βλ. Σχήμα 2-1).
2 . Στη συνέχεια, θα επαναλάβει κάθε συστατικό της σελίδας, που ορίζεται από το ρόλο που αποδίδουν. Θα εφαρμόσει βελτιώσεις CSS3 για κάθε συστατικό. Τελικά θα ενισχύει τη σήμανση σε μια σελίδα που θα τη καθιστά καθολική σε όλες τις κινητές πλατφόρμες.
3 . Τέλος, αφού οι βελτιώσεις της σελίδας είναι πλήρεις, το Jquery Mobile θα εμφανίσει τη βελτιστοποιημένη σελίδα.



Σχήμα 2-1. ***jQuery Mobile Page Enhancements Diagram***

Multi-Page Περίγραμμα

Το Jquery Mobile υποστηρίζει τη δυνατότητα να ενσωματώσετε πολλές σελίδες σε ένα ενιαίο HTML έγγραφο, όπως φαίνεται στη καταχώρηση 2-3. Αυτή η στρατηγική μπορεί να χρησιμοποιηθεί για να επιτύχει μικρότερους χρόνους απόκρισης κατά τη φόρτωση υπο-σελίδων. Όπως μπορείτε να δείτε στο το παρακάτω παράδειγμα, το πολυσέλιδο έγγραφο είναι πανομοιότυπο με το ενιαίο πρότυπο σελίδας που είδαμε νωρίτερα.

Καταχώρηση 2–3. *Multi-Page Template*

```
<head>
<meta charset="utf-8">
<title>Multi Page Example</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="jquery.mobile-min.css" />
<script type="text/javascript" src="jquery-min.js"></script>
<script type="text/javascript">
/* Shared scripts for all internal and ajax-loaded pages */
```

```
</script>
<script type="text/javascript" src="jquery.mobile-min.js"></script>
</head>
<body>
<!-- First Page -->
<div data-role="page" id="home" data-title="Home"> 1
<div data-role="header">
<h1>Welcome Home</h1>
</div>
<div data-role="content">
<a href="#contact" data-role="button">Contact Us</a> 2
</div>
</div>
<!-- Second Page -->
<div data-role="page" id="contact" data-title="Contact">
<div data-role="header">
<h1>Contact Us</h1>
</div>
<div data-role="content">
Contact information...
</div>
<script type="text/javascript">
/* Page specific scripts here */ 3
</script>
</div>
</body>
```

1 . Κάθε σελίδα σε ένα πολυσέλιδο έγγραφο πρέπει να περιέχει ένα μοναδικό αναγνωριστικό . Μια σελίδα μπορεί να έχει ένα ρόλο, είτε σελίδας ή διαλόγου. Κατά τη φόρτωση ενός πολυσέλιδου εγγράφου μόνο η αρχική σελίδα θα να ενισχυθεί και θα εμφανιστεί. Αυτή η συμπεριφορά είναι ιδανικό για γρήγορη απόκριση φορές.

2 . Κατά τη σύνδεση με μια εσωτερική σελίδα θα πρέπει να αναφέρονται σε αυτήν με το αναγνωριστικό της (id) . Για παράδειγμα, το href για να συνδέσει με τη σελίδα επαφής πρέπει να ρυθμιστεί ως href = " # επαφής» .

3 . Αν θέλετε να δείτε scripts μιας συγκεκριμένης σελίδας πρέπει να είναι τοποθετειμένα εντός της σελίδας του δοχείου . Ο κανόνας αυτός ισχύει επίσης και για τις σελίδες που φορτώνονται μέσω Ajax. Για παράδειγμα , κάθε JavaScript που δηλώνεται εσωτερικά σε multi-page.html # επαφής δεν θα είναι προσβάσιμα σε πολυσέλιδο.html#home. Μόνο τα scripts της ενεργής σελίδας θα είναι προσβάσιμα. Ωστόσο ,όλα τα scripts, συμπεριλαμβανομένης των jQuery , jQuery Mobile , και των δικών σας scripts που δηλώνονται μέσα στην ετικέτα της κεφαλής θα είναι διαθέσιμη σε όλες τις εσωτερικές και Ajaxloaded σελίδες.

Ajax-Driven Πλοήγηση

Στο παράδειγμα παραπάνω (βλ. Σχήμα 2-3) είδαμε πώς το Jquery Mobile πλοηγεί από μία εσωτερική σελίδα στην άλλη. Κατά την πλοήγηση από σελίδα σε σελίδα , μπορούμε να ρυθμίσουμε το είδος της μετάβασης. Από προεπιλογή , το πλαίσιο θα εφαρμόσει μια « διαφάνεια » κατά τη μετάβαση, για όλες τις μεταβάσεις.

< - ! Πλοηγηθείτε σε μια εσωτερική σελίδα - >

<a href="#contact" data-role="button"> Επικοινωνήστε μαζί μας < / a>
< / div >

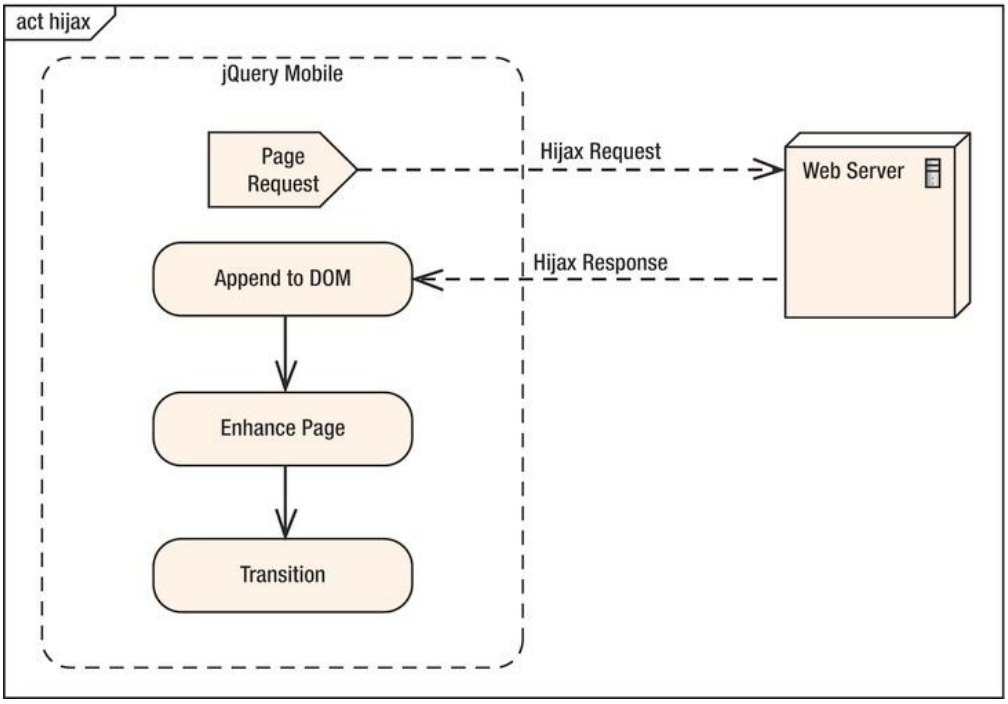
Το μοντέλο πλοήγησης είναι διαφορετικό , όταν μια σελίδα μεταβαίνει σε μια άλλη. Για παράδειγμα :

Λίστα 2-4 . Ajax πλοήγηση

<a href="contact.html" data-role="button"> Επικοινωνήστε μαζί μας < / a>
< / div >

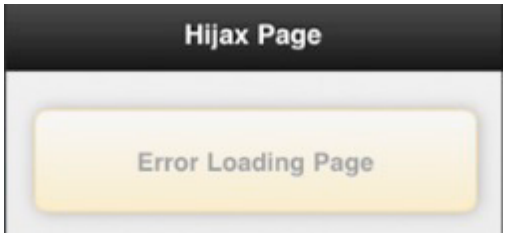
Όταν κάνετε κλικ στο σύνδεσμο "Επικοινωνήστε μαζί μας" παραπάνω, το Jquery Mobile θα επεξεργαστεί το αίτημα αυτό ως ακολούθως:

1 . Θα αναλύσει το href και θα φόρτωσει τη σελίδας μέσω αίτημα Ajax (Hijax). Εάν η σελίδα έχει φορτωθεί με επιτυχία, θα πρέπει να προστεθεί στο DOM της τρέχουσας σελίδας.



Σχήμα 2-2. ***jQuery Mobile Hijax Request***

Έχοντας προστεθεί η σελίδα με επιτυχία στο DOM, το Jquery Mobile θα ενισχύσει την σελίδα, θα ενημέρωσει τα στοιχεία @ href, και θα ορίσει τα data-url στοιχεία αν δεν ορίστηκαν.
2 . Στη συνέχεια θα μεταβεί στη νέα σελίδα με το χαρακτηριστικό: "διαφάνεια" μετάβασης. Το framework είναι σε θέση να επιτύχει μια ομαλή μετάβαση CSS διότι τόσο το "από" και "έως" σελίδα συνυπάρχουν στο DOM. Αφού έχει επιτευχθεί η μετάβαση, της σελίδας θα ανατεθεί το «ui-σελίδα-ενεργό» CSS class.
3 . Η διεύθυνση URL που προκύπτει είναι επίσης bookmarkable.
4 . Εάν οποιαδήποτε σελίδα αποτυγχάνει να φορτωθεί, θα εμφανιστεί ένα μήνυμα "Σφάλμα φόρτωσης σελίδας" (βλ. Σχήμα 2-3).



Σχήμα 2-3. ***Error Loading Screen***

Μεταβάσεις

Το Jquery Mobile έχει έξι CSS-based εφέ μετάβασης για επιλογή κατά τη μετάβαση μεταξύ των σελίδων. Από προεπιλογή, το πλαίσιο θα εφαρμόσει μια «διαφάνεια» που ισχύει για όλα τα μεταβάσεις. Μπορούμε να ορίσουμε μια εναλλακτική μετάβαση ένα data-transition χαρακτηριστικό :

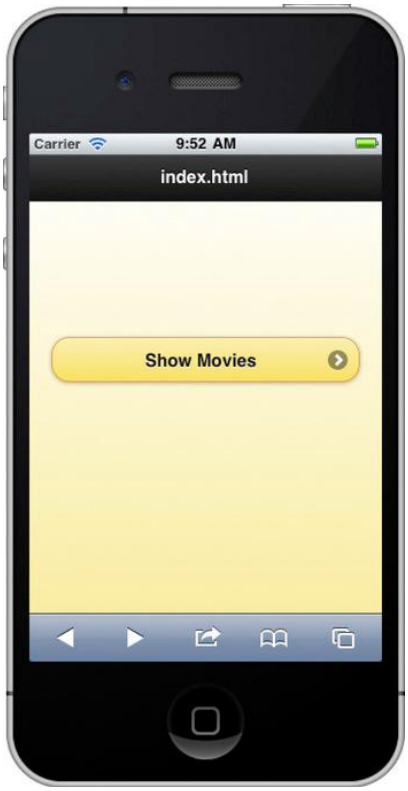
<a href="dialog.html" data-transition="slideup"> Εμφάνιση διαλόγου </ a>
Η πλήρης λίστα των εφέ μετάβασης περιγράφονται παρακάτω:

Μετάβαση και χρήση

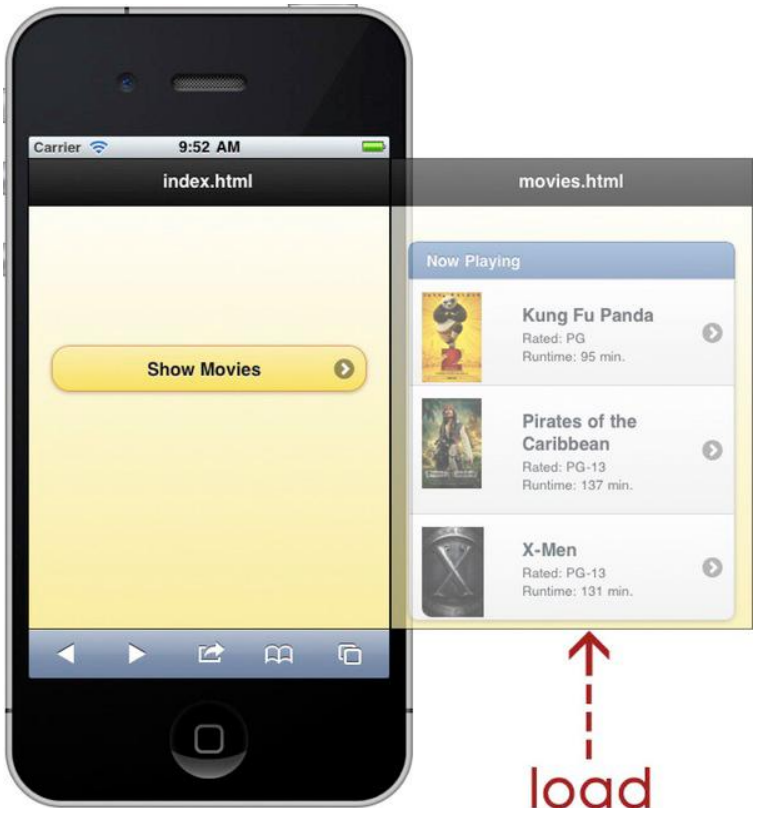
Slide:
Η πλέον κοινή μετάβαση για μετακίνηση μεταξύ των σελίδων. Η μετάβαση αυτή δίνει το εμφάνιση ότι κινείται προς τα εμπρός ή προς τα πίσω σε μια ροή των σελίδων. Είναι η προεπιλεγμένη μετάβαση για όλους τους συνδέσμους.
Slideup:
Μια κοινή μετάβαση για το άνοιγμα των διαλόγων ή εμφανίζομενων πρόσθετων πληροφοριών. Αυτό το είδος μετάβασης δίνει την εντύπωση ότι θα χρειαστεί να συλλέξουμε επιπρόσθετα στοιχεία για τη σελίδα.
Slidedown:
Αυτή η μετάβαση είναι το αντίστροφο του slideup αλλά μπορεί να χρησιμοποιηθεί για ένα παρόμοιο αποτέλεσμα.
Pop:
Ακόμα ένα είδος μετάβασης για το άνοιγμα διαλόγων ή επιπρόσθετων πληροφοριών.
Fade:
Ένα κοινό εφέ μετάβασης για τις σελίδες εισόδου ή εξόδου**.**
Flip:
Μια κοινή μετάβαση για την προβολή επιπρόσθετων πληροφοριών. Τυπικά, το πίσω μέρος της οθόνης έχει εικονίδια που δεν χρειάζεται να είναι στην κύρια UI.
None:
Καμία μετάβαση δεν εφαρμόζεται.

Η διαδικασία της μετάβασης από τη σελίδα προς σελίδα παρουσιάζεται στα ακόλουθα βήματα:

1 . Ένας χρήστης ακουμπά το κουμπί για να μεταβεί στην επόμενη σελίδα (βλ. Εικόνα 2-5).
2 . Το πλαίσιο θα φορτώσει την επόμενη σελίδα με ένα αίτημα Hijax και θα προστεθεί το στο DOM της τρέχουσας σελίδας. (βλ. Εικόνα 2–6).
3 . (βλ. Εικόνα 2-7). Αυτό το παράδειγμα χρησιμοποιεί η προεπιλεγμένη "slide" μετάβαση.
4 . Η επόμενη σελίδα εμφανίζεται και η μετάβαση είναι πλήρης (βλ. Εικόνα 2-8).

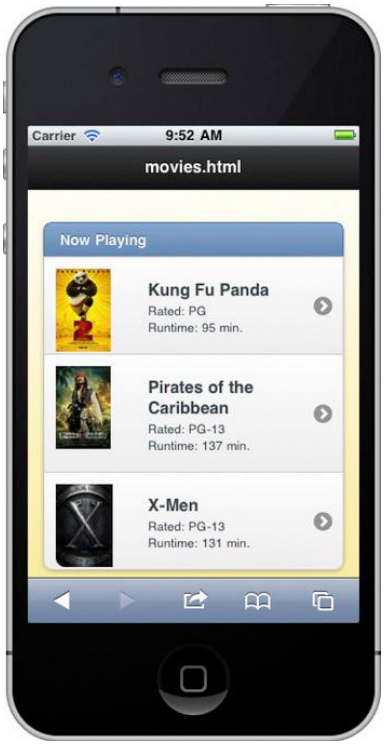


Εικόνα 2-5 . ***Tap the button to navigate to another page***



Εικόνα 2-6. ***Step #2: Framework loads the next page side-by-side***

Εικόνα 2-7. ***Step #3: Framework transitions to next page***



Εικόνα 2-8.Step #4: Transition is complete

Διάλογοι

Οι διάλογοι είναι παρόμοιοι με τις σελίδες εκτός των συνόρων τους, που είναι σε εσοχή για να τους δώσει την εμφάνιση ενός παραθύρου. Το Jquery Mobile είναι αρκετά ευέλικτο σε σχέση με το πώς μπορούμε να το θεματίσουμε τους διαλόγους. Μπορούμε να δημιουργήσουμε διαλόγους επιβεβαίωσης (βλ. Εικόνα 2-9), διάλογοι συναγερμού (βλ. Εικόνα 2 - 10), και ακόμα φύλλο στυλ διαλόγους (βλέπε Εικόνα 2-11, Εικόνα 2-12).

Εικονα 2-9. ***Confirmation Dialog***

Εικόνα 2-10. Alert Dialog

Μπορούμε να μετατρέψουμε μια σελίδα σε ένα διάλογο είτε τη σύνδεση ή τη σελίδα συνιστώσα. Σε ένα σύνδεσμο, προσθέσετε τα data-rel = "διαλόγου" χαρακτηριστικό, όπως φαίνεται στη καταχώρηση 2-5. Η προσθήκη αυτού του χαρακτηριστικό θα φορτώσει αυτόματα τη σελίδα-στόχο και θα βελτιώσει το παράθυρο διαλόγου.

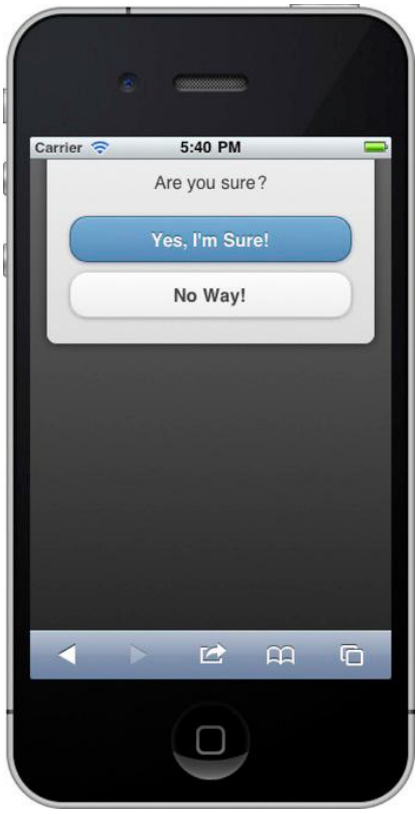
Καταχώρηση 2–5. *Link-level transformation*

```
<!-- Open a page as a dialog -->
<a href="#terms" data-rel="dialog" data-transition="slidedown">Terms</a>
<!-- The page remains unchanged. -->
<div data-role="page" id="terms">
<div data-role="header">
<h1>Terms and Conditions</h1>
</div>
<div data-role="content">
Do you agree to these terms?
<a href="#" data-role="button" data-inline="true"
data-rel="back" data-theme="a">Disagree</a>
<a href="#" data-role="button" data-inline="true">Agree</a>
</div>
</div>
```

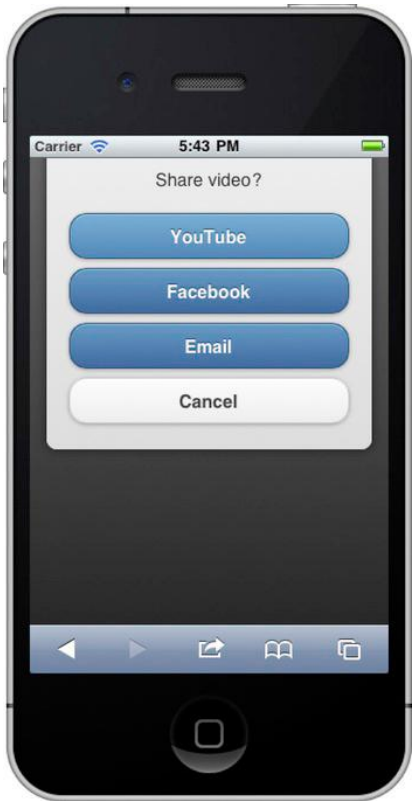
We can also configure dialogs at the page container. Add the data-role="dialog"
attribute to the page container and when the component loads it will be enhanced as a
modal dialog

Καταχώρηση 2–6*. Page-level transformation*

```
<!-- Link without data-rel="dialog" attribute -->
<a href="#terms" data-transition="slidedown">Terms and Conditions</a>
<!-- Configure this page to appear as a dialog -->
<div data-role="dialog" id="terms">
<div data-role="header">
<h1>Terms and Conditions</h1>
</div>
<div data-role="content" data-theme=”c”>
Do you agree to these terms?
<a href="#" data-role="button" data-inline="true"
data-rel="back" data-theme="a">Disagree</a>
<a href="#" data-role="button" data-inline="true">Agree</a>
</div>
```



Εικόνα 2-11. ***Action Sheet #1***



Εικόνα 2-12. Action Sheet #2

Καταχώρηση 2–7. *Action Sheet*

```
<!-- Logout link -->
<a href="#logout" data-transition="slidedown">Logout</a>
<!-- Create an action sheet by simply removing the header! -->
<div data-role="dialog" id="logout">
<div data-role="content">
<span class="title">Are you sure?</span>
<a href="#home" data-role="button" data-theme="b">Yes, I'm Sure!</a>
<a href="#" data-role="button" data-theme="c" data-rel="back">No Way!</a>
</div>
<style>
span.title { display:block; text-align:center;
margin-top:10px; margin-bottom:20px; }
</style>
</div>
```

Αυτό είναι επίσης η πρώτη επαφή μας με τα data-themeχαρακτηριστικά. Μπορούμε απλά να προσθέσουμε αντίθεση και το στυλ σε όλες τις συνιστώσες Jquery Mobile με αυτή την ιδιότητα. Στο παραδείγματα μας, μπορούμε να ορίσουμε το θέμα του υπόβαθρου και τα κουμπιά μας.

Dialog UX Guidelines

Συνέπεια είναι ο πιο σημαντικός στόχος του σχεδιασμού όταν styling στοιχεία UI σας. σε αφορά, στο διάλογο, συγκεκριμένες κατευθυντήριες γραμμές μερικές συμβουλές από το κινητό Interface της Apple Guidelines4 περιλαμβάνουν:

Alerts:

 Προτιμήστε ειδοποιήσεις για να εμφανίσετε σημαντικές πληροφορίες που αφορούν τη χρήση τη χρήση της εφαρμογής (βλ. Εικόνα 2-10).

 Τα κουμπιά ειδοποίησης είναι χρωματισμένα είτε φωτεινά ή σκοτινά. Για μια προειδοποίηση ενός πλήκτρου το κουμπί είναι πάντα ανοιχτόχρωμο. Για ένα παράθυρο διαλόγου με δύο κουμπιά, το αριστερό κουμπί είναι πάντα σκοτεινό και το δεξί πλήκτρο του είναι πάντα το φωτινό (βλέπε Εικόνα 2 - 9).

 Σε ένα παράθυρο διαλόγου με δύο κουμπιά που προτείνει την ευνοϊκή δράση που οι χρήστες είναι πιθανό να επιλέξουν, το κουμπί που ακυρώνει τη δράση θα πρέπει να είναι η αριστερά και σκουρόχρωμο (βλ.Εικόνα 2-9).

 Σε ένα παράθυρο διαλόγου με δύο κουμπιά που προτείνει μια δυνητικά επικίνδυνη ενέργεια (διαγραφή), το κουμπί που ακυρώνει τη δράση θα πρέπει να είναι στα δεξιά και αωοικτόχρωμο. Συχνά τα κουμπιά που εκτελούν επικίνδυνες ενέργειες είναι κόκκινα.

Action Sheets:

 Προτιμήστε τα φύλλα δράσης για να συγκεντρώσετε την επιβεβαίωση των καθηκόντων που ενεργοποιούνται από το χρήστη (βλ. Εικόνα 2-11). Φύλλα δράσης μπορούν επίσης να χρησιμοποιηθούν για να παρέχουν στο χρήστη ένα εύρος επιλογών για την τρέχουσα εργασία τους (βλ. Εικόνα 2-12).

 Ένα φύλλο δράσης περιλαμβάνει πάντα τουλάχιστον δύο κουμπιά που επιτρέπουν στο χρήστη να επιλέξει πώς να ολοκληρώσουν το έργο τους.

 Συμπεριλάβετε ένα κουμπί ακύρωσης για να επιτρέψετε στους χρήστες να εγκαταλείψουν την εργασία. Το κουμπί της ακύρωσης βρίσκεται στο κάτω μέρος του φύλλου δράσης για την ενθάρρυνση των χρηστών να διαβάζουν όλες τις επιλογές πριν από την επιλογή της ακύρωσης. Το χρώμα πρέπει να αντιστοιχεί με το χρώμα του φόντου.

ΚΕΦΑΛΑΙΟ 3:

ΠΛΟΗΓΗΣΗ ΜΕ ΚΕΦΑΛΙΔΕΣ, ΓΡΑΜΜΕΣ ΕΡΓΑΛΕΙΩΝ, ΚΑΙ ΜΠΑΡΕΣ

Όλες οι εφαρμογές κινητών χρειάζονται γραμμές εργαλείων για να βοηθήσουν την πλοήγηση ή τη διαχείριση δεδομένων στην οθόνη. Σε αυτό το κεφάλαιο θα εξετάσουμε τα στοιχεία jQuery Mobile που παρέχουν αυτά τα χαρακτηριστικά. Τα κύρια συστατικά είναι οι κεφαλίδες και τα υποσέλιδα. Οι κεφαλίδες χρησιμοποιούνται συνήθως για την εμφάνιση του τίτλου της σελίδας και μπορεί προαιρετικά να περιλαμβάνει ελέγχους για να βοηθήσει την πλοήγηση αντικείμενων σχετικά με την οθόνη. Τα υποσέλιδα έχουν σχεδιαστεί παρόμοια με κεφαλίδες, αλλά προσδιορίζουν συνήθως μια γραμμή εργαλείων ή μπάρα.

ΚΕΦΑΛΙΔΑ

Η μπάρα εμφανίζει τον τίτλο της τρέχουσας οθόνης. Επιπλέον, μπορείτε να προσθέσετε κουμπιά για την πλοήγηση ή να προσθέσετε στοιχεία ελέγχου που διαχειρίζονται δεδομένα στη σελίδα. Αν και η κεφαλίδα είναι προαιρετική, χρησιμοποιείται συνήθως για να παρέχει έναν τίτλο για την ενεργή σελίδα. Ας ξεκινήσουμε με την εξέταση της δομή της κεφαλίδας και να δούμε πώς μπορούμε να προσθέσουμε επιπλέον στοιχεία ελέγχου για να βοηθήσουμε στη καλύτερη διαχείριση των στοιχείων της σελίδας.

Υπάρχουν μερικά σημεία που έχουν σημασία για την κεφαλίδα:

- Η κεφαλίδα ορίζεται με το data-role = "header" χαρακτηριστικό.
- Η κεφαλίδα είναι ένα προαιρετικό συστατικό.
- Μπορούμε να ρυθμίσουμε το θέμα της επικεφαλίδας με το data-theme χαρακτηριστικό. Αν το θέμα δεν έχει οριστεί για την κεφαλίδα θα κληρονομήσει το προεπιλεγμένο θέμα (data-theme = "a").
- Όλα τα επίπεδα επικεφαλίδας (H1-H6) έχουν θεματιστεί ομοιόμορφα για να διατηρηθεί η συνοχή.
- Μπορούμε να κάνουμε την κεφαλίδα σταθερή με την προσθήκη του data-position = "fixed "χαρακτηριστικό.

Δομή

Η βασική χρήση του header είναι να εμφανίσει απλά τον τίτλο του την ενεργή σελίδα. Μια κεφαλίδα στην απλούστερη μορφή της δείχνεται παρακάτω.

```
<div data-role="header">
<h1>Header Title</h1>
</div>
```

Τοποθέτηση

Υπάρχουν τρεις μορφές διαθέσιμες για την τοποθέτηση του header:

 Προεπιλεγμένη θέση: Η κεφαλίδα θα εμφανίζεται στο επάνω άκρο της οθόνης και θα βγαίνουν εκτός οπτικού πεδίου κατά την κύλιση.

```
<div data-role="header">
<h1>Default Header</h1>
</div>
```

 Fixed: Μία σταθερή κεφαλίδα θα παραμένει πάντα τοποθετημένη στο ορατό πάνω άκρο της οθόνης. Ωστόσο, κατά τη διάρκεια εκδήλωσης κύλισης η επικεφαλίδα θα εξαφανιστεί μέχρι το τέλος της κύλισης. Μπορούμε να δημιουργήσουμε μία σταθερή κεφαλίδα με τη προσθήκη των data-position = "fixed" χαρακτηριστικό.

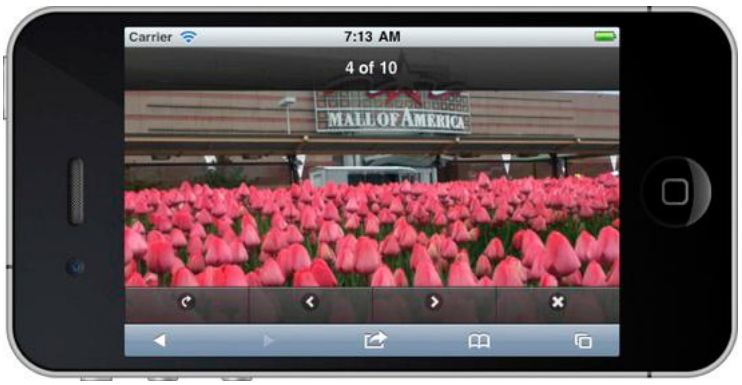
```
<div data-role="header" data-position="fixed">
<h1>Fixed Header</h1>
</div>
```

 Responsive: Όταν δημιουργούμε μια σελίδα fullscreen τα περιεχόμενα θα φαίνονται από άκρη σε άκρη και η κεφαλίδα και το υποσέλιδο θα εμφανίζονται και εξαφανίζονται βάση του αγγίγματος. Λειτουργία πλήρους οθόνης είναι ένα χρήσιμο σενάριο για την έκθεση φωτογραφιών ή βίντεο. Για να δημιουργήσουμε μία fullscreen σελίδα προσθέτουμε το χαρακτηριστικό data-fullscreen="true" και συμπεριλαμβάνουμε το data-position="fixed" χαρακτηριστικό για την κεφαλίδα και το υποσέλιδο (βλέπε Καταχώρηση 3-1). Για παράδειγμα, στη καταχώρηση 3-1 έχουμε μία fullscreen σελίδα που εμφανίζει μια φωτογραφία. Εάν ο χρήστης ακουμπήσει την οθόνη, η κεφαλίδα και το υποσέλιδο θα εμφανιστούν και εξαφανιστουν (βλ. Σχήμα 3-2). Σε αυτό το παράδειγμα, έχουμε ένα πρόγραμμα προβολής φωτογραφιών.

Καταχώρηση 3–1. *Fullscreen*

```
<div data-role="page" data-fullscreen="true">
<div data-role="header" data-position="fixed">
<h3>Header</h3>
</div>
<div data-role="content">
<!-- Fullscreen content -->
</div>
<div data-role="footer" data-position="fixed">
<h3>Footer</h3>
```

Εικόνα 3–1. ***Fullscreen***

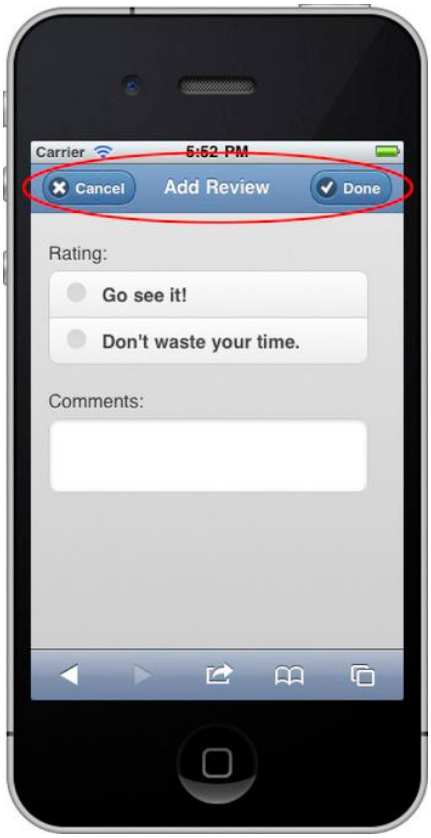


Εικόνα 3–2. ***Fullscreen with responsive header and footer***

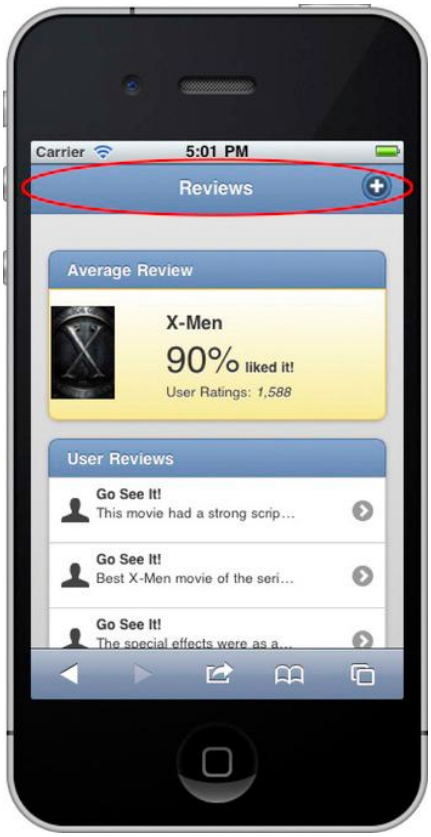
ΚΟΥΜΠΙΑ

Υπάρχουν περιπτώσεις, όπου θα χρειαστεί να προσθέσουμε στοιχεία ελέγχου στην κεφαλίδα για να βοηθήσουμε στηδιαχείρηση των περιεχομένων της οθόνης. Υπάρχουν τρεις μορφές κουμπιών που μπορούμε να προσθέσουμε στη κεφαλίδα. Αυτά περιλαμβάνουν:

- Ένα κουμπί μόνο με κείμενο.
- Ένα κουμπί με μόνο ένα εικονίδιο (βλ. Εικόνα 3-4). Ένα κουμπί με εικονίδιο απαιτεί την προσθήκη δύο χαρακτηριστικών: data-icon και dataiconpos= "notext ".
- Ένα κουμπί με κείμενο και εικονίδιο (βλ. Εικόνα 3-3). Αυτό το κουμπί απαιτεί το data-icon χαρακτηριστικό.



Εικόνα 3-3. ***Header with buttons***



Εικόνα 3-4. Header with icon

Το Jquery Mobile θα περικόψει τις κεφαλίδες και τα υποσέλιδα με μεγάλους τίτλους (βλ. Εικόνα 3-6). Όταν το κείμενο είναι πολύ μεγάλο ,θα περικόψει το κείμενο και θα προσθέσει αποσιωπητικά στο τέλος. Αν αντιμετωπίσουμε αυτή τη κατάσταση και θέλουμε να δείξουμε το πλήρες κείμενο (βλ. Εικόνα 3-7), μπορούμε να ρυθμίσουμε τον επιλογέα CSS για την αντιμετώπιση του θέματος, όπως φαίνεται στη καταχώρηση 3-5.

Καταχώρηση 3–5. *Truncation Fix*

```
.ui-header .ui-title, .ui-footer .ui-title {
margin-right: 0 !important; margin-left: 0 !important;
}
```

Εικόνα 3–6. ***Truncation Issue***

Εικόνα 3–7. ***Truncation Fix***

ΠΙΣΩ ΚΟΥΜΠΙ

Τα κουμπιά επιστροφής (βλ. Σχήμα 3-8) μπορούν να δημιουργήσουν μεγάλες συζητήσεις μεταξύ των UX σχεδιαστών. Πρέπει προσθέτουμε τα δικά μας τα κουμπιά επιστροφής ή θα πρέπει να αξιοποιούμε το υλικό / λογισμικό των κουμπιών επιστροφής, που είναι διαθέσιμα σε ορισμένες συσκευές και όλους τους περιηγητές; Ευτυχώς, το Jquery Mobile μας προσφέρει την επιλογή , της αυτόματης ενεργοποίησης ή απενεργοποίησης τους σε παγκόσμιο επίπεδο.

Εικόνα 3–8. ***Back button must be explicitly enabled***.

Το κουμπί επιστροφής είναι απενεργοποιημένο από προεπιλογή εντός Jquery Mobile. Εάν χρειαζόμαστε το πίσω κουμπί να εμφανιστεί μέσα στην κεφαλίδα έχουμε πολλές επιλογές για την προσθήκη του:

- Μπορούμε να προσθέσουμε το κουμπί επιστροφής σε μια συγκεκριμένη σελίδα με την προσθήκη δεδομένων autoback-btn = "true" στη σελίδα.
- Μπορούμε να ενεργοποιήσουμε σε παγκόσμιο επίπεδο το πίσω κουμπί ρυθμίζοντας την επιλογή addBackBtn, όταν συνδέεται με την επιλογή mobileinit. Μετά τη ρύθμιση αυτής της επιλογής, το κουμπί επιστροφής θα εμφανιστεί αυτόματα αν υπάρχει μια σελίδα στην μνήμη.

```
<div data-role="page" data-add-back-btn="true"
data-back-btn-text="Previous">
// Globally enable the back button, set the default back button text,
// and set back button theme
$(document).bind('mobileinit',function(){
$.mobile.page.prototype.options.addBackBtn = true;
$.mobile.page.prototype.options.backBtnText = "Previous";
$.mobile.page.prototype.options.backBtnTheme = "b";
});
```

Επιπλέον, αν έχουμε ενεργοποιήσει το κουμπί επιστροφής σε παγκόσμιο επίπεδο, μπορούμε να επιλέξουμε να το απενεργοποιήσουμε σε συγκεκριμένες σελίδες, προσθέτοντας τα data-add-back-btn="false" στην σελίδα. Αυτό θα αφαιρέσει το κουμπί επιστροφής από την επικεφαλίδα των συγκεκριμένων σελίδων.

<div data-role="header" **data-add-back-btn="false"**>

ΣΥΝΔΕΣΗ ΕΠΙΣΤΡΟΦΗΣ

Εάν θέλουμε να δημιουργήσουμε ένα κουμπί που συμπεριφέρεται παρόμοια με ένα πίσω κουμπί, μπορούμε να προσθέσουμε το data-rel = "back".

Go Back

Με το data-rel = "back", ο σύνδεσμος θα μιμιθεί το πίσω κουμπί.

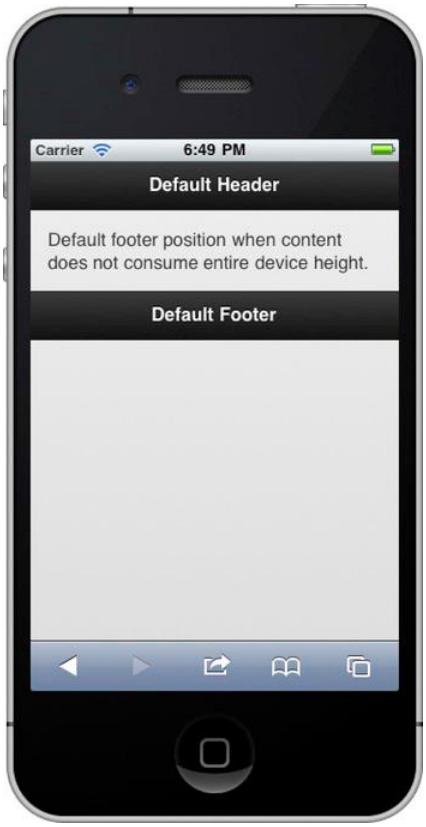
ΥΠΟΣΕΛΙΔΟ

Το συστατικό footer είναι σχεδόν ταυτόσημο με την κεφαλίδα με μικρές μόνο διαφορές. Η κύρια διαφορά είναι ότι το υποσέλιδο είναι πιο ευέλικτο σε σχέση με την τοποθέτηση των κουμπιών του. Το υποσέλιδο τοποθετεί τα κουμπιά διαδοχικά από αριστερά προς τα δεξιά. Αυτή η ευελιξία μας επιτρέπει να θεματίσουμε το υποσέλιδο μας ως μια γραμμή εργαλείων ή μπάρα.

Υπάρχουν μερικά σημεία που έχουν σημασία για το υποσέλιδο:
.
 Το υποσέλιδο ορίζεται με το data-role = "footer" χαρακτηριστικό.
Το υποσέλιδο τοποθετεί κουμπιά inline και σε διαδοχική σειρά από αριστερά προς τα δεξιά. Αυτό επιτρέπει την ευελιξία για τη δημιουργία εργαλείων.
 Το υποσέλιδο είναι προαιρετικό.
 Μπορούμε να ρυθμίσουμε το θέμα του υποσέλιδου με το data-theme χαρακτηριστικό. Αν δεν έχει οριστεί θέμα για το υποσέλιδο θα κληρονομήσει το θέμα από τη σελίδα συστατικό. Το προεπιλεγμένο θέμα είναι μαύρο (data-theme="a").
 Μπορούμε να κάνουμε το υποσέλιδο σταθερό με το data-position = "fixed " χαρακτηριστικό.

Η υποσημείωση στην απλούστερη μορφή της, εμφανίζεται στον παρακάτω κώδικα. Το data-role = "footer" είναι το μόνο απαιτούμενο χαρακτηριστικό. Μέσα στο υποσέλιδο, μπορούμε να συμπεριλάβουμε οποιαδήποτε σημασιολογική HTML. Υποσέλιδα συνήθως χρησιμοποιούνται για να περιέχουν γραμμή εργαλείων και στην καρτέλα ελέγχου. Η γραμμή εργαλείων παρέχει ένα σύνολο ενεργειών, όπου μπορούν να αξιοποιήσουν οι χρήστες. Και μια μπάρα δίνει στους χρήστες τη δυνατότητα την εναλλαγή μεταξύ των διαφορετικών απόψεων μέσα από την εφαρμογή.

```
<div data-role="footer">
<!-- Add footer text or buttons here -->
</div>
```



Εικόνα 3–9. *Default footer position*

ΤΟΠΟΘΕΤΗΣΗ

Οι τρεις μορφές της θέσης για την κεφαλίδα ισχύουν και για το υποσέλιδο. Αυτές περιλαμβάνουν:

 Default: Ένα προεπιλεγμένο υποσέλιδο τοποθετείται μετά το τμήμα περιεχομένου. Για παράδειγμα, εάν το περιεχόμενό μας εκτείνεται πέρα από το ύψος του παραθύρου προβολής μας, το υποσέλιδο δεν θα εμφανίζεται μέχρι να μεταβούμε στο τέλος του περιεχομένου.

```
<div data-role="footer">
<!-- Default footer -->
</div>
```

 Fixed: Ένα σταθερό υποσέλιδο, θα παραμείνει πάντα τοποθετημένο στο ορατό κάτω άκρο της οθόνης. Ωστόσο, κατά τη διάρκεια εκδήλωσης κύλισης το υποσέλιδο θα εξαφανιστεί μέχρι το κύλισης να τελειώσει. Μπορούμε να δημιουργήσουμε ένα σταθερό υποσέλιδο με την προσθήκη του data-position = "fixed" χαρακτηριστικού.

```
<div data-role="footer" data-position="fixed">
<h3>Fixed Footer</h3>
</div>
```

 Responsive: Όταν δημιουργούμε μια σελίδα fullscreen τα περιεχόμενα θα φαίνονται από άκρη σε άκρη και η κεφαλίδα και το υποσέλιδο θα εμφανίζονται και εξαφανίζονται βασιζόμενα σε μια απόκριση στο άγγιγμα. Λειτουργία πλήρους οθόνης είναι ένα χρήσιμο σενάριο για προβολές φωτογραφιών ή βίντεο. Για να δημιουργήσουμε μία fullscreen σελίδα προσθέτουμε τα data fullscreen = "true" στη σελίδα και data-position="φιχεδ" χαρακτηριστικά για την κεφαλίδα και το υποσέλιδο στοιχεία.

ΚΟΥΜΠΙΑ

Υπάρχουν τρεις μορφές των κουμπιών που μπορούμε να προσθέσουμε σε ένα υποσέλιδο:

 Κουμπί με κείμενο. Αυτό το στύλ κουμπιού λειτουργεί καλά μέσα σε γραμμή εργαλείων επειδή η εμφάνιση της δεν είναι τόσο μεγάλη όσο ενός μπαρ καρτέλα. Ενας σύνδεσμος του υποσέλιδου θα εμφανιστεί ως κουμπί με κείμενο:

Sync

 Κουμπί μόνο με εικονίδιο. Αυτό το στύλ κουμπιού λειτουργεί επίσης καλά μέσα σε γραμμή εργαλείων. Ένα εικονίδιο-κουμπί απαιτεί την προσθήκη δύο χαρακτηριστικών, data-icon και data-iconpos="notext":

 Ένα κουμπί με κείμενο και εικονίδιο. Αυτό το στύλ κουμπιού λειτουργεί καλά μέσα σε μια μπάρα:

<a href="#" data-icon="home"> σελίδα </ a>

ΚΕΦΑΛΑΙΟ 4:

ΜΟΡΦΟΠΟΙΗΣΗ ΣΤΟΙΧΕΙΩΝ ΚΑΙ ΚΟΥΜΠΙΩΝ

Mobile εφαρμογές πρέπει να υποστηρίζουν μια αποτελεσματική εμπειρία χρήστη. Για το λόγο αυτό, είναι σπάνιο να δείτε εφαρμογές για κινητά με πολλές φορμες. Στην πραγματικότητα, όσο λιγότερη είναι η αλληλεπίδραση των χρηστών με τις εφαρμογές, τόσο πιο αποτελεσματικοί θα γίνουν οι χρήστες και οι εφαρμογέ. Σε αυτό το κεφάλαιο, θα αρχίσουμε με το πιο δημοφιλές συστατικό UI, το κουμπί. Κουμπιά μπορούν να οριστούν και να ρυθμιστούν με πολλούς τρόπους. Στη συνέχεια, θα ρίξουμε μια αναλυτική ματιά στο κάθε φόρμα HTML. Θα εξετάσουμε επίσης τα χαρακτηριστικά των δεδομένων jQuery Mobile που είναι μοναδικά για κάθε στοιχείο της φόρμας.

ΕΙΔΗ ΚΟΥΜΠΙΩΝ

Τα κουμπιά είναι η πιο συχνά χρησιμοποιούμενη μορφή ελέγχου στο πλαίσιο των κινητών εφαρμογών, επειδή παρέχουν μια πολύ αποτελεσματική εμπειρία στο χρήστη. Έχουμε ήδη δει τα κουμπιά που χρησιμοποιούνται σε πολλά παραδείγματα, συμπεριλαμβανομένων διαλόγους μας, τα φύλλα δράσης, κατακερματισμένες ελέγχους, και την κεφαλίδα. Τα Jquery Mobile κουμπιά έρχονται σε πολλές μορφές. Έχουμε κουμπιά συνδέσμου, κουμπιά φόρμας, κουμπιά εικόνας, εικονίδια πλήκτρα και κουμπιά με κείμενο και εικόνες. Είτε έχουμε ένα κουμπί συνδέσμου ή μια φόρμα που βασίζεται σε κουμπί, το framework θα θεματίσει και τα δύο με τον ίδιο τρόπο.

ΣΥΝΔΕΣΜΟΥ

Κουμπιά σύνδεσης αποτελούν το ειδος κουμπιού με τη μεγαλύτερη χρήση. Όποτε χρειάζεται να δομίσουμε ένα συνηθισμένο σύνδεσμο ως κουμπί, προσθέτουμε το data-role="button" χαρακτηριστικό στο σύνδεσμο (βλ. Εικόνα 4-1).

Εικόνα 4–1. ***Link buttons***

Από προεπιλογή, τα κουμπιά στο τμήμα του περιεχομένου μιας σελίδας δομούνται ως block-level στοιχεία ώστε να γεμίσουν όλο το πλάτος του εξωτερικού δοχείου τους. Ωστόσο, εάν θέλουμε ένα πιο συμπαγής κουμπί που είναι τόσο ευρύ όσο το κείμενο και τα εικονίδια στο εσωτερικό, προσθέτουμε το data-inline = "true "χαρακτηριστικό (βλέπε Καταχώρηση 4-1).

Καταχώρηση 4–1. *Link buttons*

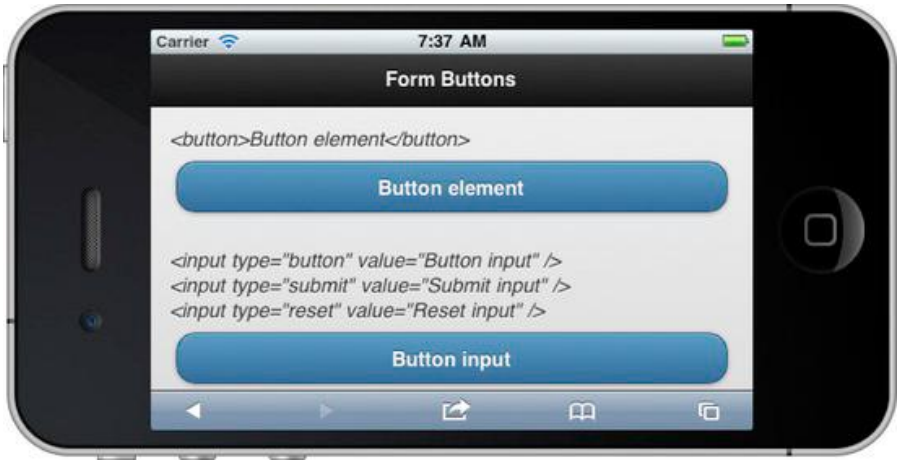
```
<a href="#" data-role="button">Link button</a>
<a href="#" data-role="button" data-inline="true">Disagree</a>
<a href="#" data-role="button" data-inline="true">Agree</a>
```

ΦΟΡΜΑΣ

Κουμπιά φορμώ (βλέπε Καταχώρηση 4-2) είναι στην πραγματικότητα πιο εύκολο να τα δομίσουμε σε σχέση με τα κουμπιά που είναι link-based επειδή δεν απαιτούνται τροποποιήσεις από την πλευρά μας. Για απλότητα, το framework μετατρέπει αυτόματα οποιοδήποτε στοιχείο κουμπί ή εισόδου σε ένα κουμπί για κινητές συσκευές για μας (βλ. Σχήμα 4-2).

Καταχώρηση 4–2. *Form buttons*

```
<button type="submit">Button element</button>
<input type="button" value="button" />
<input type="submit" value="submit" />
<input type="reset" value="reset" />
```



Εικόνα 4–2. *Form buttons*

ΕΙΚΟΝΑΣ

Δομόντας εικόνες σαν κουμπιά απαιτεί ελάχιστη προσπάθεια από μέρους μας. Όταν περιτυλίγουμε μια εικόνα με μια ετικέτα, δεν απαιτούνται τροποποιήσεις (βλ. Σχήμα 4-3). Ωστόσο, κατά την επισύναψη μιας εικόνας σε ένα στοιχείο εισόδου θα πρέπει να προσθέσουμε το data-role = "none" χαρακτηριστικό.

Εικόνα 4–3. ***Image buttons***

Καταχώρηση 4–3. *Image buttons*

```
<!-- Image buttons -->
<input type="image" src="cloud.png" data-role="none" />
                    <a href="#"><img src="cloud.png"></a>
```

ΜΕ ΕΙΚΟΝΙΔΙΑ

Το Jquery Mobile περιλαμβάνει μια σειρά από απλά εικονίδια που χρησιμοποιούνται συνήθως στον τομέα των κινητών εφαρμογών, το οποίο περιλαμβάνει ένα ενιαίο λευκό εικονίδιο sprite που έχει ένα ημι-διαφανές μαύρο κύκλο πίσω από το εικονίδιο για να εξασφαλίσει μια καλή αντίθεση σε οποιοδήποτε χρώμα του φόντου (βλ. Εικόνα 4-4).

Εικόνα 4–4. *Buttons with standard icons*

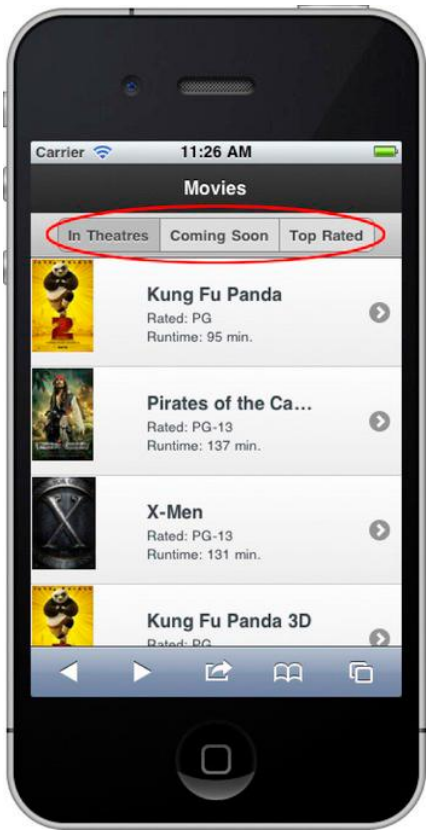
Ο πίνακας 4-1 περιέχει κάθε εικονίδιο αποδίδουν αξία και την αντίστοιχη εικόνα του. Κάθε τιμή χαρακτηριστικού έχει μια σχετική εικόνα, εκτός από τα data-icon = "custom".

Table 4–1. *data-icon listing*

data-icon	Image
plus	
minus	
delete	
arrow-r	
arrow-l	
arrow-u	
arrow-d	
check	
gear	
refresh	
forward	
back	
grid	
star	
alert	
info	
home	
search	
custom	

ΟΜΑΔΟΠΟΙΗΣΗ

Μέχρι στιγμής, κάθε παράδειγμα το κουμπί που φαίνεται είχε κάθε κουμπί διαχωρίζονται από τους άλλους. Ωστόσο, αν θέλετε να ομαδοποιήσετε τα κουμπιά μαζί σας, μπορείτε να τυλίξετε τα κουμπιά σας μέσα μία ομάδα ελέγχου. Για παράδειγμα, στην κατακερματισμένη από τον έλεγχό μας στο κεφάλαιο 3 ήταν ομαδοποιούνται με τον τρόπο αυτό (βλέπε Εικόνα 4-8).



Εικόνα 4–8. ***Grouping Buttons***

Για να πάρουμε αυτό το αποτέλεσμα, μαζεύουμε μια ομάδα πλήκτρων σε ένα δοχείο με το datarole ="control group "χαρακτηριστικό (βλέπε Καταχώρηση 4-8).

Καταχώρηση 4–8. *Grouping buttons (ch3/header-segmented-control.html)*

```
<div data-role="controlgroup" data-type="horizontal">
<a href="#" data-role="button">In Theatres</a>
<a href="#" data-role="button">Coming Soon</a>
<a href="#" data-role="button">Top Rated</a>
                                        </div>
```

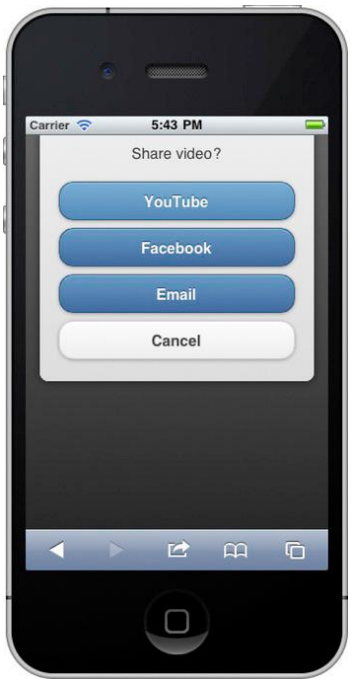
Από προεπιλογή, το framework θα ομαδοποιήσει τα κουμπιά κάθετα, αφαιρώντας όλα τα περιθώρια, και προσθέτοντας σύνορα μεταξύ των πλήκτρων. Επιπλέον, για να ενισχύσει οπτικά την ομάδα, τα πρώτα και τελευταία στοιχεία θα είναι διακοσμημένα με στρογγυλεμένες γωνίες. Επειδή τα κουμπιά είναι τοποθετημένα κάθετα από προεπιλογή, μπορούμε να τα δομίσουμε οριζόντια με τη προσθήκη του data="horizontal" χαρακτηριστικού. Σε αντίθεση με κατακόρυφα κουμπιά που καταναλώνουν ολόκληρο το πλάτος του εξωτερικού δοχείου τους, τα οριζόντια κουμπιά είναι μόνο τόσο ευρύ όσο το περιεχόμενό τους.

ΘΕΜΑΤΙΖΟΝΤΑΣ ΚΟΥΜΠΙΑ

Κουμπιά, όπως και όλες τις συνιστώσες jQuery Mobile, θα κληρονομήσουν το θέμα από τη μητρικό τους δοχείο. Επιπλέον, όταν θα πρέπει να θεματίσουμε κουμπιά με διαφορετικά χρώματα μπορούμε να εφαρμόσουμε το θέμα της επιλογής μας σε οποιοδήποτε κουμπί με την προσθήκη των data-theme χαρακτηριστικών (βλέπε Καταχώρηση 4-9).

Καταχώρηση 4-9. Θέματα κουμπιών

```
<a href="#home" data-role="button" data-theme="b"> YouTube </ a>
<a href="#home" data-role="button" data-theme="b"> Facebook </ a>
<a href="#home" data-role="button" data-theme="b"> Email </ a>
<a href="#home" data-role="button" data-theme="c"> Ακύρωση </ a>
```



Εικόνα 4–9. ***Theming Buttons***

ΦΟΡΜΕΣ

Μέθοδοι για την οικοδόμηση form-based εφαρμογών μέσα σε Jquery Mobile είναι πολύ παρόμοιες με εκείνες που παραδοσιακά έχουν χρησιμοποιηθεί για να δημιουργήσουμε φόρμες στο διαδίκτυο. Παρά το γεγονός ότι η δράση και μέθοδος χαρακτηριστικό θα πρέπει να καθορίζονται για λόγους σαφήνειας, δεν είναι αναγκαίο. Από προεπιλογή, η δράση θα προκαθορίσει στης σελίδας το μονοπάτι, η οποία μπορεί να βρεθεί με $. mobile.path.get () και μία απροσδιόριστη μέθοδος θα προκαθορίσει ως "get". Όταν οι φόρμες υποβάλλονται, θα μεταβούν στην επόμενη σελίδα με την προεπιλεγμένη "Διαφάνεια" μετάβαση. Ωστόσο, μπορούμε να διαμορφώσουμε τη μορφή μετάβασης με το ίδια δεδομένα χαρακτηριστικά που χρησιμοποιήσαμε προηγουμένως για να διαχειριστούμε τους συνδέσμους μας (βλέπε Λίστα 4-13).

Καταχώρηση 4–13. *Submitting forms*

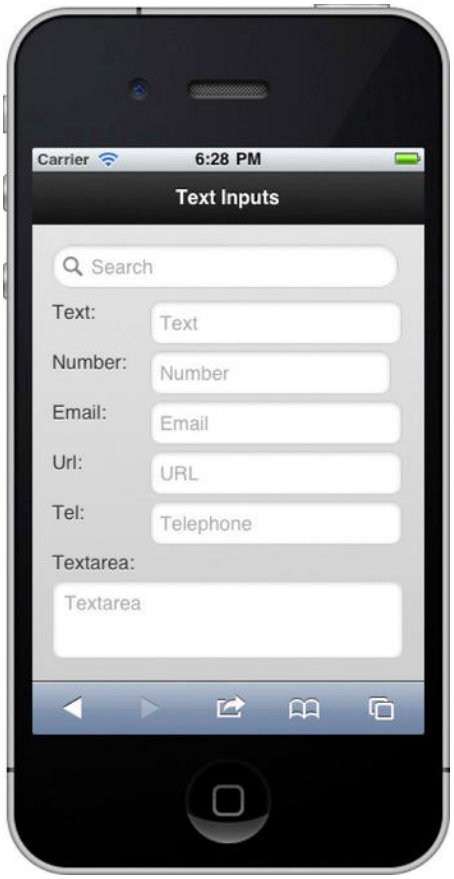
```
<form action="/save.html" method="post" data-transition="pop">
<label for="email">Email:</label>
<input type="email" name="email" id="email" value="" />
<button type="submit" name="submit">Submit</button>
</form>
```

Μπορούμε να προσθέσουμε τα εξής χαρακτηριστικά στο στοιχείο φόρμα μας για τη διαχείριση της μετάβασης ή για την απενεργοποιήση του Ajax:

- data-transition=”pop”
- data-direction=”reverse”
- data-ajax=”false”

ΚΕΙΜΕΝΑ

Τα κείμενα είναι η πιο περίπλοκη μορφή για εργασία σε κινητές συσκευές. Όπως αναφέρθηκε προηγουμένως, APIs συσκευών μπορούν να βοηθήσουν στην απλοποίηση τηε εμπειρίας του χρήστη. Αν και είναι ένας καλός στόχος η ελαχιστοποιήση αυτων των κουραστικών εργασιών, υπάρχουν φορές που θα πρέπει να συλλέξουμε σχόλια χρηστών με κείμενο. Τα πιο κοινά πεδία φόρμας κειμένου δείχνονται στο Εικόνα 4-10.



Εικόνα 4–10. ***Text inputs***

Καταχώρηση 4–14. *Text inputs*

```
<input type="text" name="text" value="" id="text" placeholder="Text"/>
<input type="number" name="number" value="" id="number" />
<input type="email" name="email" value="" id="email" data-theme="d" />
<input type="url" name="url" value="" id="url" />
<input type="tel" name="tel" value="" id="tel" />
<input type="search" name="search" value="" id="search" />
    <textarea cols="40" rows="8" name="textarea" id="textarea"></textarea>
```

Κατά την οικοδόμηση των φορμών, είναι σημαντικό να συνδέουμε το πεδίο εισαγωγής με σημασιολογικό τύπο του. Η ένωση αυτή έχει δύο πλεονεκτήματα. Πρώτον, όταν το πεδίο εισαγωγής δέχεται να επικεντρωθεί, δηλαδή να έχει το κέρσορα πάνω του, ζητά από το χρήστη το πληκτρολόγιο. Για παράδειγμα, ένα πεδίο που ορίζεται ως type = "number" θα προτρέψει αυτόματα ένα αριθμητικό πληκτρολόγιο (βλέπε Εικόνα 4-11). Ομοίως, ένα πεδίο που αντιστοιχίζεται με type = "tel" θα οδηγήσει σε ένα «τηλεφωνικό» πληκτρολόγιο (βλέπε σχήμα 4-12).

Εικόνα 4–11. ***Numeric keyboard***

Εικόνα* 4–12.** ***Telephone keyboard

Επιπλέον, αυτή η προδιαγραφή επιτρέπει το πρόγραμμα περιήγησης να εφαρμόζει τους κανόνες επικύρωσης που εφαρμόζονται για τον τύπο πεδίου. Η υποστήριξη του προγράμματος περιήγησης για την αυτόματη επικύρωση κατά την υποβολή των φορμών εξακολουθεί να είναι ελάχιστη, αλλά θα βελτιωθεί με την πάροδο του χρόνου.

ΜΕΝΟΥ

Το framework θα ενισχύσει αυτόματα όλα τα εγγενή στοιχεία με καμία πρόσθετη σήμανση να απαιτείται (βλέπε Καταχώρηση 4-16).

Καταχώρηση 4–16. *Native select menu*

```
<label for="genre">Genre:</label>
<select name="genre" id="genre">
<option value="action">Action</option>
<option value="comedy">Comedy</option>
<option value="drama">Drama</option>
</select>
```

Ο μετασχηματισμός αυτός θα αντικαταστήσει την αρχική επιλογή με ένα στυλ κουμπιού jQuery Mobile που περιέχει ένα εικονίδιο με βέλος που είναι ευθυγραμμισμένο δεξιά. Από προεπιλογή, πατώντας αυτό το κουμπί επιλογής θα ξεκινήσει τη διαδικασία επιλογής για το λειτουργικό σύστημα (βλ. Εικόνα 4-13).

Εικόνα 4–13. ***Select menu***

Αφού οι χρήστες κάνουν την επιλογή τους, το κουμπί επιλογής θα εμφανίσει την τιμή της συγκεκριμένης επιλογής (δ). Εάν η τιμή του κειμένου είναι πολύ μεγάλη για το κουμπί, το κείμενο θα περικοπεί. Επιπλέον, τα multi-κουμπιά επιλογής θα εμφανίσουν μια «φούσκα» ή σήμα μετά την επιλογή περισσότερων από μία επιλογές (βλ. Εικόνα 4-13). Αυτό είναι ένα οπτικό εφέ που αναδεικνύει τον αριθμό των επιλογών.

RADIO ΚΟΥΜΠΙΑ

Τα συγκεκριμένα κουμπιά επιλογής θα περιορίσουν την επιλογή του χρήστη σε ένα μόνο σημείο (βλ. Εικόνα 4-15). Για παράδειγμα, επιλέγοντας ένα αίτημα, μεταξύ πολλών επιλογών συνήθως επιτυγχάνεται με radio κουμπιά, τα οποία προτιμώνται για την απλότητα και την ευκολία χρήσης τους. Οι χρήστες μπορούν να πατήσουν οπουδήποτε στο κουμπί για να κάνουν την επιλογή τους, και το Jquery Mobile θα ενημερώνει αυτόματα το στοιχείο ελέγχου φόρμας.

Εικόνα 4–15. ***Radio buttons***

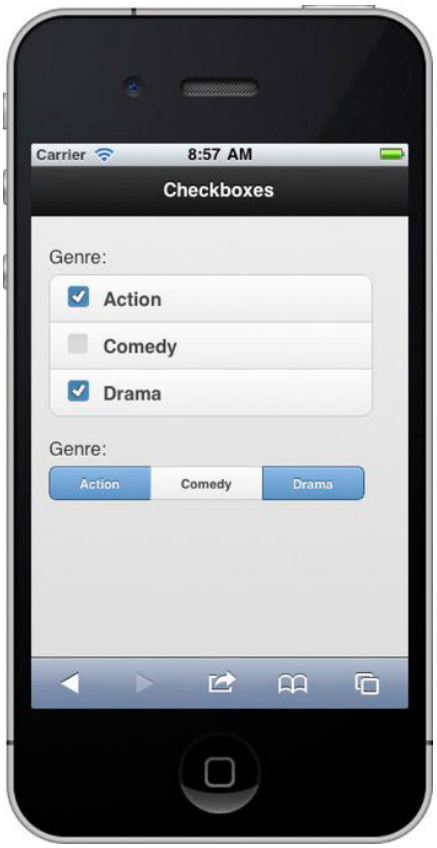
Στη καταχώρηση 4-19 προσθέσαμε τρία επιπλέον χαρακτηριστικά για να βοηθήσει το στυλ και τη θέση μας ραδιόφωνο κουμπιά. Το πρώτο χαρακτηριστικό, data-role = "ομάδα μαρτύρων", κομψά ομάδες τα κουμπιά μαζί με στρογγυλεμένες γωνίες. Το δεύτερο χαρακτηριστικό, data-type = "οριζόντια", υπερισχύει η προεπιλεγμένη τοποθέτηση κάθετη και εμφανίζει τα κουμπιά οριζόντια. Τέλος,έχουμε θέμα τα κουμπιά μας. Από προεπιλογή, τα κουμπιά επιλογής θα κληρονομήσουν το θέμα της μητρικής τους ελέγχου. Ωστόσο, εάν θέλετε να εφαρμόσετε μια εναλλακτική θέμα για τα κουμπιά επιλογής σας, μπορεί να προσθέσετε τα δεδομένα-θέμα χαρακτηριστικό στην ετικέτα του αντίστοιχο κουμπί.

Καταχώρηση 4–19. *Horizontal radio buttons*

```
<fieldset data-role="controlgroup" data-type="horizontal">
<legend>Map view:</legend>
<input type="radio" name="map" id="map1" value="Map" />
<label for="map1" data-theme=”b”>Map</label>
<input type="radio" name="map" id="map2" value="Satellite" />
<label for="map2" data-theme=”b”>Satellite</label>
<input type="radio" name="map" id="map3" value="Hybrid" />
<label for="map3" data-theme=”b”>Hybrid</label>
</fieldset>
```

CHECK ΚΟΥΜΠΙΑ

Checkboxes είναι μια κοινή μορφή ελέγχου που επιτρέπει στους χρήστες να επιλέξουν πολλαπλές τιμές από μια λίστα από τις πολλές επιλογές (βλ. Σχήμα 4-16). Οι χρήστες μπορούν να πατήστε οπουδήποτε στο κουτάκι κουμπί για να κάνει μια επιλογή και jQuery Mobile θα ενημερώνει αυτόματα τις υποκείμενες σχηματίζουν τον έλεγχο.



Εικόνα 4–16 *Checkboxes*

Η σήμανση για το θεματισμό και τη τοποθέτηση των check κουμπιών είναι πανομοιότυπη με αυτή που χρεισιμοποιήσαμε προηγουμένως για radio κουμπιά (βλέπε Καταχώρηση 4-21). Και πάλι, θα προσθέσουμε τρία επιπλέον χαρακτηριστικά για να να βοηθήσουμε το στυλ και τη θέση των κουμπιών. Το πρώτο χαρακτηριστικό, data-role = "control-group", κομψά ομαδοποιεί τα στοιχεία με στρογγυλεμένες γωνίες. Το δεύτερο χαρακτηριστικό, data-type= "horizontal", παρακάμπτει την προεπιλεγμένη κατακόρυφη τοποθέτηση των κουμπιών και τα εμφανίζει οριζόντια. Τέλος, θεματίζουμε τα κουμπιά μας. Από προεπιλογή, τα check κουμπιά θα κληρονομήσουν το θέμα του ελέγχου τους. Ωστόσο, εάν θέλουμε να εφαρμόσουμε ένα εναλλακτικό θέμα μπορούμε να προσθέσουμε το data-theme χαρακτηριστικό στην ετικέτα του αντίστοιχου κουτιού.

Καταχώρηση 4–21. *Horizonal checkboxes*

```
<fieldset data-role="controlgroup" data-type="horizontal">
<legend>Genre:</legend>
<input type="checkbox" name="genre" id="c1" />
<label for="c1" data-theme=”c”>Action</label>
<input type="checkbox" name="genre" id="c2" />
<label for="c2" data-theme=”c”>Comedy</label>
<input type="checkbox" name="genre" id="c3" />
<label for="c3" data-theme=”c”>Drama</label>
</fieldset>
```

ΟΛΙΣΘΗΤΗΣ

Ένας ολισθητής είναι ένα κοινό στοιχείο ελέγχου φόρμας που επιτρέπει στους χρήστες να επιλέξουν μια τιμή ανάμεσα σε ένα ελάχιστο και μέγιστο εύρος (βλ. Εικόνα 4-17).

Εικόνα 4–17. *Slider*

Στο παράδειγμα μας, ρυθμίζουμε την ένταση της φωτεινότητας με ένα ολισθητή ρυθμίζοντας το εύρος μεταξύ χαμηλής και υψηλής έντασης. Μπορούμε να προσαρμόσουμε το ελάχιστο και μέγιστο όριο του ολισθητή και

επίσης να ορίσουμε την προεπιλεγμένη του τιμή. Ο χρήστης μπορεί να ρυθμίσει τον ολισθητή είτε σύροντας τον ολισθητή είτε εισάγοντας μια τιμή στο αντίστοιχο πεδίο κειμένου του. Όπως φαίνεται στη καταχώρηση 4-23, καμία πρόσθετη σήμανση δεν είναι απαραίτητη για το Jquery Mobile ώστε να βελτιώσει τον ολισθητή. Κάθε στοιχείο εισόδου με type = "range" θα βελτιωθεί αυτόματα.

Καταχώρηση 4–23. *Slider*

```
<label for="volume">Volume:</label>
<input type="range" name="volume" id="volume" value="5" min="0" max="9"/>
```

Ένας ολισθητής αποτελείται από δύο θεματικά μέρη. Υπάρχει το μέρος γνωστό ως ολισθητής και το μέρος γνωστό ως τροχιά. Κάθε ένα από αυτά τα μέρη μπορεί να θεματιστεί ξεχωριστά. Για να θεματίσουμε τον ολισθητή, προσθέτουμε το data-theme="a" χαρακτηριστικό στο στοιχείο εισόδου. Επιπλέον, για να θεματίσουμε τη τροχιά, προσθέτουμε το data-track-theme="a" χαρακτηριστικό στο στοιχείο εισόδου:

```
<input type="range" name="brightness" id="brightness" min="0" max="10"
data-theme="b"
data-track-theme="a" />
```

ΔΙΑΚΟΠΤΗΣ ΕΛΕΓΧΟΥ

Ένας διακόπτης ελέγχου (βλέπε εικόνα 4-18) χρησιμοποιείται συνήθως για τη διαχείριση boolean on / off σημαίες.

Εικόνα 4–18. ***Switch control***

Διακόπτες ελέγχου είναι συχνά τα προτιμώμενα μέσα που επιτρέπουν στο χρήστη να χειραγωγήσει τις ρυθμίσεις εφαρμογών, λόγω της απλότητας και της ευκολίας χρήσης τους. Για να αντιστρέψει το διακόπτη, ο χρήστης μπορεί να αξιοποιήσει είτε τον έλεγχο ή να σύρει το διακόπτη. Για να δημιουργήσουμε ένα στοιχείο ελέγχου διακόπτη, προσθέτουμε ένα στοιχείο me το data-role= "slider" και δύο επιλογές για τη διαχείριση των on / off μελών (βλ. Καταχώρηση 4-25).

Καταχώρηση 4–25. *Switch control*

```
<label for="alerts">Alerts:</label>
<select name="slider" id="alerts" data-role="slider">
<option value="off">Off</option>
<option value="on">On</option>
</select>
```

Ένας διακόπτης ελέγχου αποτελείται από δύο θεματικά μέρη. Υπάρχει το μέρος γνωστό ως ολισθητής και το μέρος γνωστό ως τροχιά. Κάθε ένα από αυτά τα μέρη μπορεί να θεματιστεί ξεχωριστά. Για να θεματίσουμε τον ολισθητή, προσθέτουμε το data-theme="a" χαρακτηριστικό στο στοιχείο εισόδου. Επιπλέον, για να θεματίσουμε τη τροχιά, προσθέτουμε το data-track-theme="a" χαρακτηριστικό στο στοιχείο εισόδου:

<select name="slider" **data-theme="b" data-track-theme="c"** data-role="slider">
<option value="off">Off</option>
<option value="on">On</option>
</select>

ΗΜΕΡΟΜΗΝΙΑ

Το Mobiscroll6 είναι ένας βελτιστοποιημένος επιλογέας ημερομηνίας για συσκευές με οθόνη αφής. Το API είναι Mobiscroll διαμορφώσιμο, το οποίο επιτρέπει την απεικόνιση πολλών συνδυασμών ημερομηνίας και ώρας (βλέπε Εικόνα 4-21). Επιπλέον, το Mobiscroll μπορεί να θεματιστεί και μπορεί επίσης να προσαρμοστεί για να εμφανίσει όλα τα δεδομένα που απαιτούνται (βλ. Εικόνα 4-22).

Εικόνα 4–21. *Date Picker*

Εικόνα 4–22. W*ith custom lists*

Μπορούμε να ενημερώσουμε τις επιλογές MobiScroll για να δημιουργήσουμε μια προσαρμοσμένη ταινία αναζήτησης (βλέπε Καταχώρηση 4-29).

Καταχώρηση 4–29. *Mobiscroll*

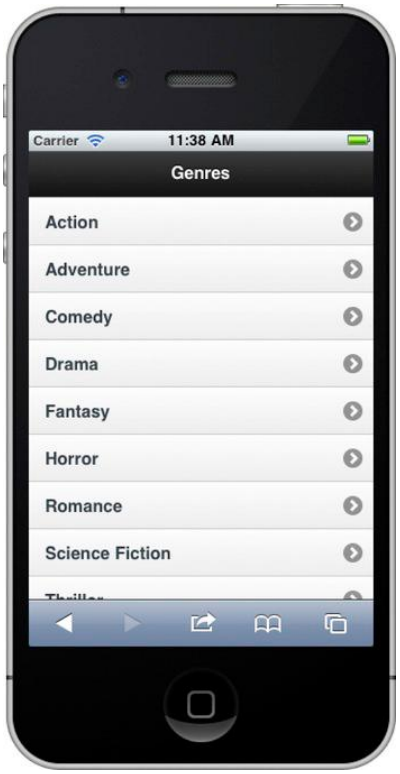
```
// Import the Mobiscroll resources
<script type="text/javascript" src="jquery.scroller-1.0.2.js"></script>
<link type="text/css" rel="stylesheet" href="jquery.scroller-1.0.2.css"/>
// Display the default date picker (see Figure 4–21).
$( “#element1” ).scroller();
// Display a custom filter for a movie search (see Figure 4–22).
$( "#element2" ).scroller({
setText: 'Search',
theme: 'sense-ui',
wheels: [{
'Rating': { '5-star': '*****', '4–star': '****' ... },
'Genre': { 'action': 'Action', 'comedy': 'Comedy', ...},
'Screen': { '3d': '3D', 'imax': 'IMAX', 'wide': 'Wide' }
}]
});
```

ΚΕΦΑΛΑΙΟ 5:

ΛΙΣΤΕΣ

Οι λίστες είναι ένα δημοφιλές στοιχείο της διεπαφής χρήστη, επειδή κάνουν την περιήγηση πολύ απλή και αποτελεσματική. Οι λίστες αποτελούν ένα πολύ ευέλικτο στοιχείο που μπορεί να διακοσμηθεί με πολλούς τρόπους και να προσαρμοστεί πολύ καλά σε διαφορετικά μεγέθη οθόνης. Είτε ψάχνουμε ταχυδρομείο, επαφές, μουσική, ή ρυθμίσεις, κάθε μία από αυτές τις εφαρμογές εμφανίζει λίστες πληροφορίες σε ελαφρώς διαφορετικές μορφές. Από βασικές λίστες που περιλαμβάνουν μόνο κείμενο μέχρι και πολύπλοκες λίστες με γραφικά και λεπτομερή μεταδεδομένα, θα πρέπει να είναι αρκετά ευέλικες για να στηρίξουν πολλές διαμορφώσεις. Ευτυχώς, το Jquery Mobile υποστηρίζει όλες αυτές τις διαμορφώσεις και ακόμα περισσότερα. Σε αυτό το κεφάλαιο θα εξετάσουμε τις λεπτομέρειες του στυλ και τη διαμόρφωση καταλόγων μέσω Jquery Mobile. Θα δούμε επίσης πώς μπορούμε να προσθέσουμε φίλτρα αναζήτησης στους καταλόγους μας.

Το Jquery Mobile θα εξελίξει αυτόματα κάθε λίστα HTML (<ul> ή <ol>) σε μία βελτιστοποιημένη προβολή όταν προσθέσουμε το data-role="list" χαρακτηριστικό στη λίστα μας. Η ενισχυμένη λίστα θα εμφανιστεί από άκρη σε άκρη, από προεπιλογή, και εάν τα στοιχεία της λίστας μας περιέχουν συνδέσμους, θα εμφανίζονατι ως touch-friendly κουμπιά ένα δεξιό εικονίδιο βέλους (βλ. Εικόνα 5-1). Από προεπιλογή, οι λίστες θα είναι διακοσμημένες με το "c" δείγμα (γκρι) χρώμα. Για να εφαρμόσουμε ένα εναλλακτικό θέμα, προσθέτουμε το data-theme χαρακτηριστικό στα στοιχεία της λίστας ή μια λίστα στοιχείων (<li>).

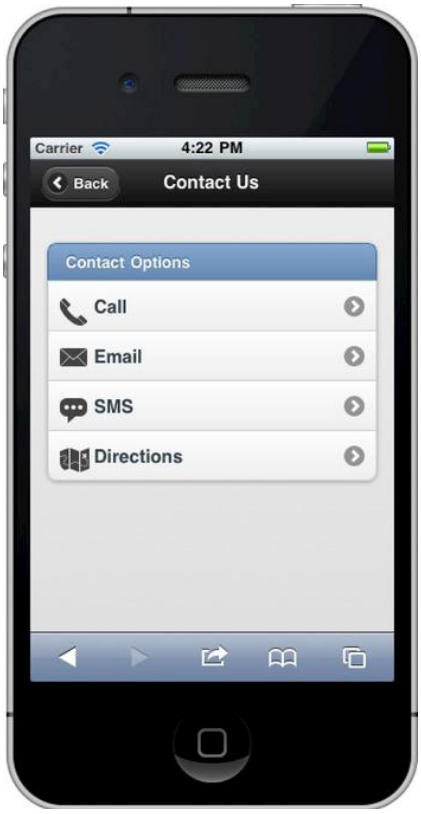


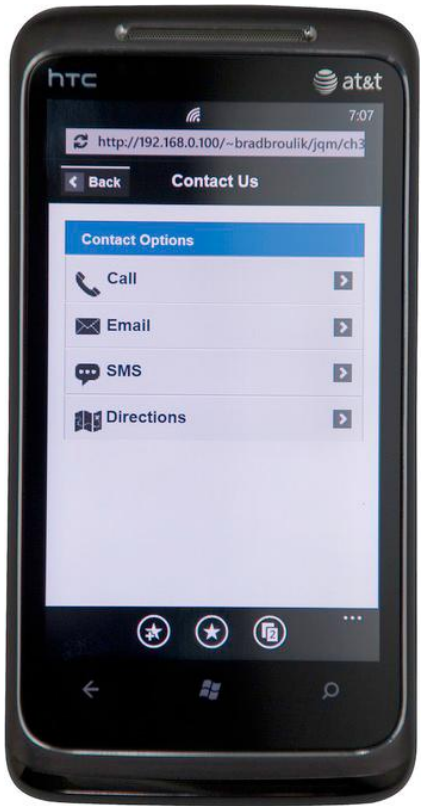
Εικόνα 5–1. ***Basic list***

```
<ul data-role="listview" data-theme="c">
<li><a href="#">Action</a></li>
<li><a href="#">Adventure</a></li>
<li><a href="#">Comedy</a></li>
</ul>
```

ΕΙΣΡΟΗΣ

Μια λίστα εισροής δεν θα εμφανίζεται από άκρη σε άκρη. Αντ 'αυτού, θα τυλίγεται αυτόματα μέσα σε ένα μπλοκ με στρογγυλεμένες γωνίες και με συγκεκριμένα περιθώρια για επιπλέον απόσταση. Για να δημιουργήσουμε μία λίστα ειροής, προσθέτουμε το data-inset="true" χαρακτηριστικό στο στοιχείο της λίστας (βλ. Εικόνα 5-2 και 5-3, και ο σχετικός κώδικας στη Καταχώρηση 5-2).





Εικόνα 5–2. ***Inset list (iOS)*** **Εικόνα 5–3.** ***Inset list (Windows Phone 7)***

Κατα΄χωρηση 5–2. *Inset list*

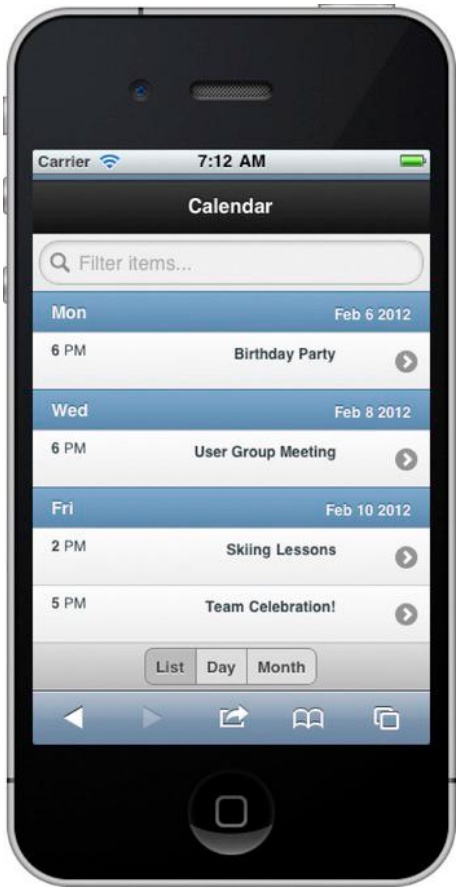
```
<ul data-role="listview" data-inset="true">
<li data-role="list-divider">Contact Options</li>
<li><a href="#"><img src="phone.png" class="ui-li-icon">Call</a></li>
<li>...</li>
</ul>
```

ΔΙΑΧΩΡΙΣΤΙΚΑ

Ένα διαχωριστικό λίστας μπορεί να χρησιμοποιηθεί ως τίτλος για μια ομάδα στοιχείων της λίστας. Για παράδειγμα, εάν η εγαρμογή μας έχει μια λίστα ημερολογίου, μπορούμε να ομαδοποιήσουμε τα συμβάντα του ημερολογίου βάση μέρας (βλ. Εικόνα 5-4). Διαχωριστικά λιστών μπορούν επίσης να χρησιμοποιηθούν ως κεφαλίδες για τις λίστες. Στο προηγούμενο μας παράδειγμα, θέτουμε επίσης την κεφαλίδα της λίστας μας με ένα διαχωριστικό λίστας (βλ. Εικόνα 5-2 και Καταχώρηση 5-2). Για να δημιουργήσουμε μια λίστα με διαχωριστικό, προσθέτουμε το data-role ="list-divider" χαρακτηριστικό σε

οποιοδήποτε στοιχείο της λίστας. Το κείμενο του καταλόγου διαιρέτη θα εμφανιστεί αριστερά στοιχισμένο.

Εξ ορισμού, τα διαχωριστικά λιστών θα πρέπει να είναι θεματισμένα με το "b" δείγμα (γαλάζιο) χρώμα. Για να εφαρμόσουμε ένα εναλλακτικό θέμα, προσθέτουμε το data-divider-theme="a" χαρακτηριστικό για το στοιχείο λίστας.



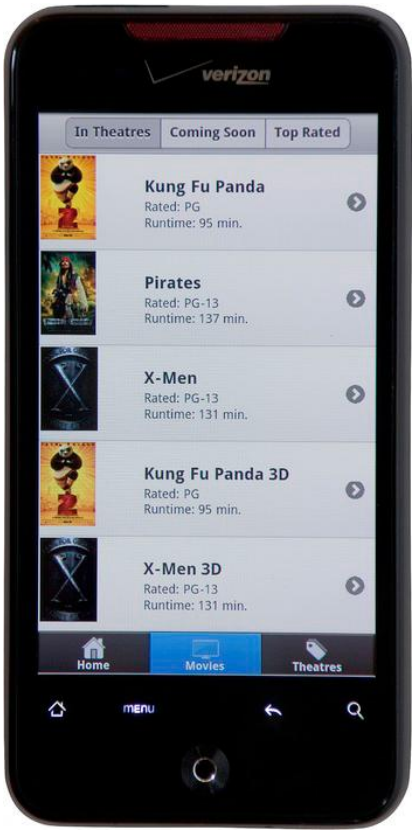
Εικόνα 5–4. ***List dividers***

Καταχώρηση 5–3. *List dividers*

```
<ul data-role="listview">
<li data-role="list-divider" data-divider-theme="a">
Mon <p class="ui-li-aside">Feb 6 2012</p>
</li>
<li>
<a href="#">6 PM <span class="ui-li-aside">Birthday Party</span></a>
</li>
</ul>
```

ΜΕ ΕΙΚΟΝΙΔΙΑ ΚΑΙ ΜΙΚΡΟΓΡΑΦΙΕΣ

Μπορούμε να προσθέσουμε εικονίδια αριστερά της λίστας μας με την προσθήκη μιας εικόνας μέσα σε ένα στοιχείο λίστας (βλ. Εικόνα 5-5). Το framework θα αναβαθμίσει την εικόνα έως 80 pixels.

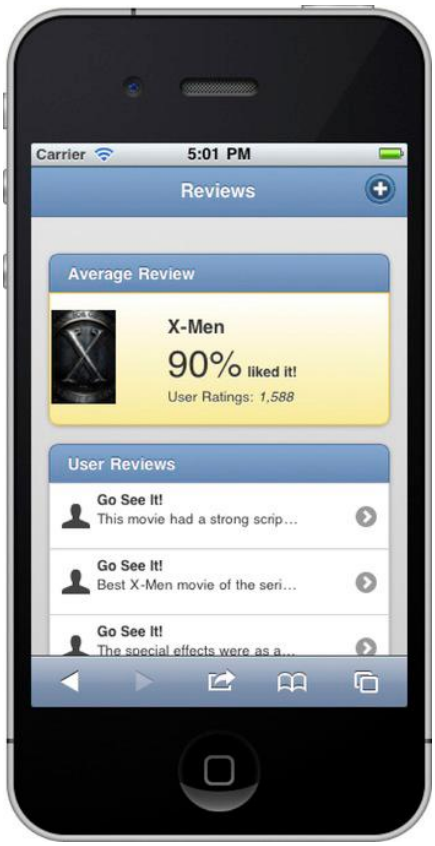


Εικόνα 5–5. ***List with thumbnails***

Καταχώρηση 5–4. *List with thumbnails*

```
<ul data-role="listview">
<li>
<a href="movies/kung-fu-panda.html">
<img src="images/kungfupanda2.jpg" />
<h3>Kung Fu Panda</h3>
<p>Rated: PG</p>
<p>Runtime: 95 min.</p>
</a>
</li>
...
</ul>
```

Μπορούμε επίσης να χρησιμοποιήσουμε μικρότερα εικονίδια αντί των μικρογραφιών. Για να χρησιμοποιήσουμε τα πρότυπα εικονίδια 16x16 pixel σε στοιχεία λίστας, προσθέτουμε την κατηγορία ui-li-icon στο στοιχείο της εικόνας (βλ. Εικόνα 5-6).



Εικόνα 5–6. ***List with icons***

Καταχώρηση 5–5. *List with icons*

```
<ul data-role="listview" data-inset="true" data-theme="d">
<li data-role="list-divider">User Reviews</li>
<li>
<a href="reviews/xmen/404.html">
<img src="images/111-user.png" class="ui-li-icon">
<p><strong>Go See It!</strong></p>
<p>This movie had a strong script and ... </p>
</a>
</li>
...
</ul>
```

ΑΡΙΘΜΗΜΕΝΕΣ

Αριθμημένες λίστες θα δημιουργηθούν κατά τη χρήση μιας διατεταγμένης λίστας <ol> (βλ. Εικόνα 5-8).

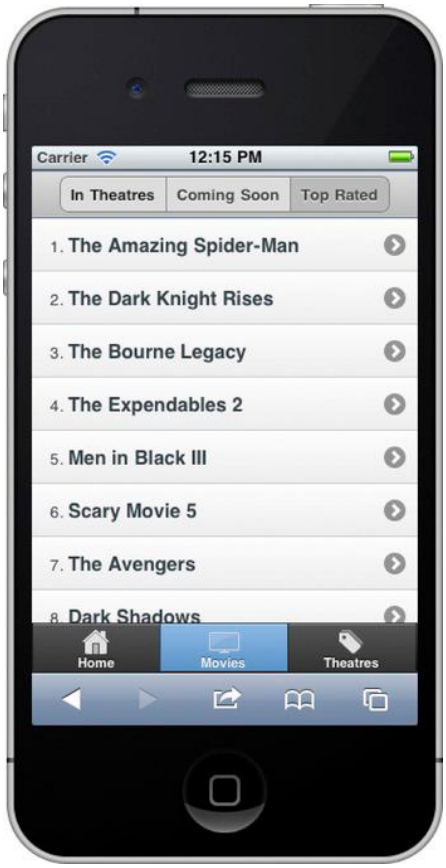


Figure 5–8. ***List with numbers***

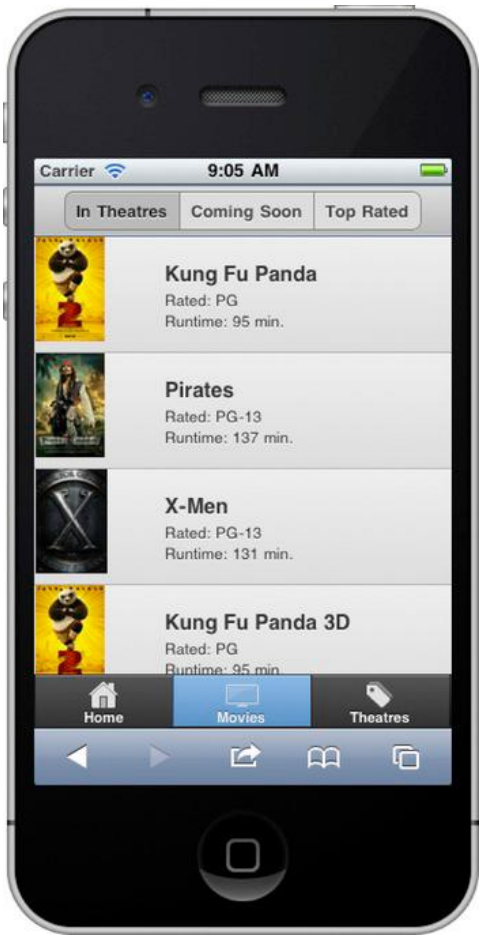
Καταχώρηση 5–7. *List with numbers*

```
<ol data-role="listview">
<li><a href="spider-man.html">The Amazing Spider-Man</a></li>
<li><a href="dark-knight.html">The Dark Knight Rises</a></li>
...
</ol>
```

Εξ ορισμού, το framework θα προσθέσει το αριθμητικό δείκτη προς τα αριστερά κάθε στοιχείου της λίστας. Αυτές οι λίστες είναι χρήσιμες όταν υπάρχει προβολή μιας λίστας από αντικείμενα που μπορούν να κατατάσσονται διαδοχικά.

ΓΙΑ ΔΙΑΒΑΣΜΑ

Λίστες μπορούν επίσης να προβάλλουν δεδομένα μόνο για διάβασμα. Για να δημιουργήσουμε μια λίστα για ανάγνωση, απλά αφαιρούμε τις ετικέτες που χρησιμοποιούνται στα προηγούμενα παραδείγματα μας (βλ. Εικόνα 5-9 και σχετικός κώδικας του στην Καταχώρηση 5-8).



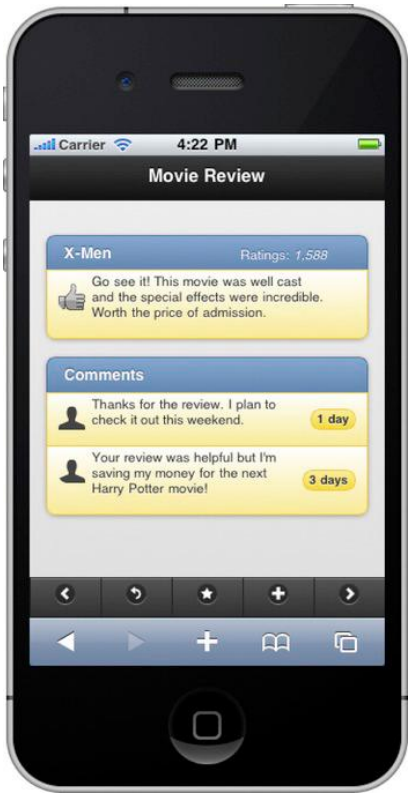
Εικόνα 5–9. ***List with read-only items***

Καταχώρηση 5–8. *List with read-only items*

```
<ul data-role="listview">
<li>
<img src="images/kungfupanda2.jpg" />
<h3>Kung Fu Panda</h3>
<p>Rated: PG</p>
<p>Runtime: 95 min.</p>
</li>
...
</ul>
```

ΣΗΜΑΤΑ

Ένα σήμα λίστας είναι ένα τονισμένο οβάλ που συνήθως δείχνει τον αριθμό των νέων στοιχείων που είναι διαθέσιμα για προβολή. Για παράδειγμα, τα σήματα λιστών χρησιμοποιούνται συνήθως σε εφαρμογές ηλεκτρονικού ταχυδρομείου για να δείξουν πόσα μη αναγνωσμένα γράμματα ηλεκτρονικού ταχυδρομείου έχουμε. Στο παράδειγμά μας, τα σήματα χρησιμοποιούνται για να υποδείξουν πότε ένα σχόλιο προστέθηκε για την αναθεώρηση μιας ταινίας (βλέπε Εικόνα 5-10). Ένα σήμα μπορεί να χρησιμοποιηθεί για να εκφράσει κάθε είδους μετα-δεδομένα.



Εικόνα 5–10. ***List with badges or count bubbles***

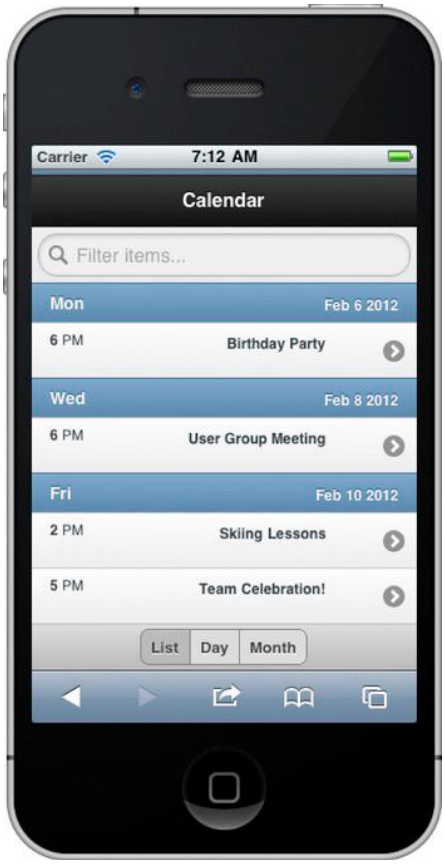
Για να δημιουργήσετε ένα σήμα, τυλίξτε το κείμενο του σήματος με ένα στοιχείο που περιέχει μια κατηγορία ui-li-count. Από προεπιλογή, τα σήματα θα πρέπει να είναι διακοσμημένα με το "c" δείγμα (γκρι) χρώμα. Για να εφαρμόσουμε ένα εναλλακτικό θέμα, προσθέτουμε το data-count-theme χαρακτηριστικό στο στοιχείο της λίστας (βλέπε Καταχώρηση 5-9).

Καταχώρηση 5–9. *List with badges or count bubbles*

```
<ul data-role="listview" data-inset="true" data-count-theme="e">
<li data-role="list-divider">Comments</p></li>
<li>
<img src="../images/111-user.png" class="ui-li-icon">
<p>Thanks for the review. I'll check it out this weekend.</p>
<span class="ui-li-count">1 day ago</span>
</li>
</ul>
```

ΦΙΛΤΡΑΡΙΣΜΕΝΕΣ ΜΕ ΜΠΑΡΑ ΑΝΑΖΗΤΗΣΗΣ

Το Jquery Mobile έχει μια πολύ βολική client-side δυνατότητα αναζήτησης για λίστες φιλτραρίσματος. Για να δημιουργήσουμε μια γραμμή αναζήτησης, προσθέτουμε το data-filter="true" χαρακτηριστικό στη λίστα. Το framework θα προσθέσει ένα φίλτρο αναζήτησης πάνω από τη λίστα και το αρχικό κείμενο θα εμφανίσει τις λέξεις, "Filter items ..." (βλ. Εικόνα 5-11 και τα συναφή κώδικα στην Καταχώρηση 5-10).



Εικόνα 5–11. ***List filtering (unfiltered)***

Καταχώρηση 5–10. *List filtering*

```
<ul data-role="listview" data-filter="true" data-filterplaceholder="
Search..."
<li data-role="list-divider">
Mon <p class="ui-li-aside">Feb 6 2012</p>
</li>
<li>
<a href="b-day.html">
<p>6 PM <span class="ui-li-aside">Birthday Party</span></p>
</a>
</li>
</ul>
```

ΚΕΦΑΛΑΙΟ 6:

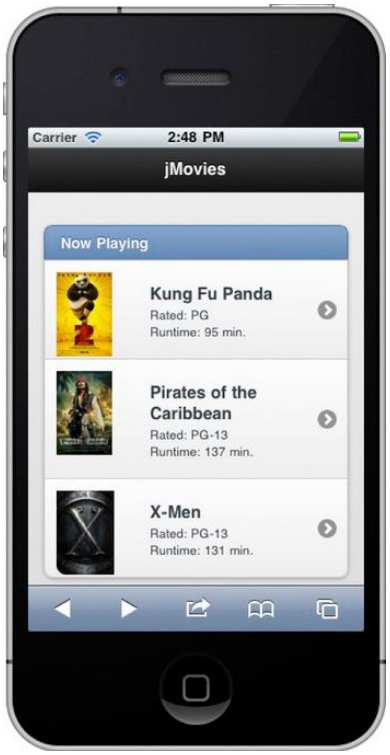
ΘΕΜΑΤΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ

Το Jquery Mobile έχει ένα ενσωματωμένο framework που επιτρέπει τη γρήγορη αλλαγή θέματος στη διεπαφή χρήστη. Αυτό πραγματοποιείται μέσω πολλών χαρακτηριστικών CSS3, τα οποία βοηθούν στην οικοδόμηση πιο κομψών και ευαίσθητων σχεδίων. Είναι σε θέση να εφαρμόσει στρογγυλεμένες γωνίες, σκιές, και κλίσεις, χωρίς να χρειάζεται να επικαλεστεί τις εικόνες. Ουσιαστικά, έχουμε μία «ελαφριά» θεματοποίηση που καθιστά ένα ενιαίο σχεδιασμό σε όλα τα προγράμματα περιήγησης.

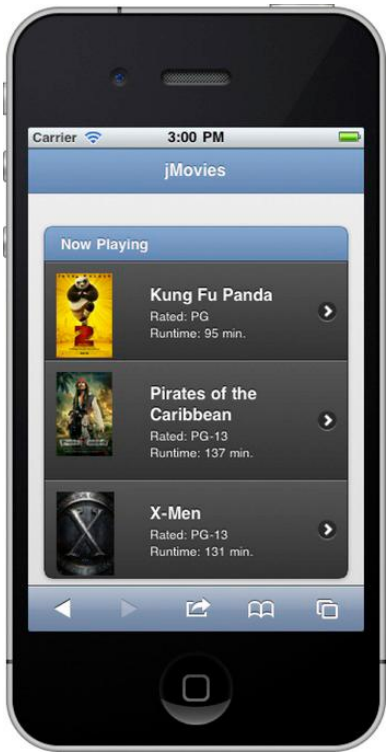
Σε αυτό το κεφάλαιο, θα συζητήσουμε τα βασικά στοιχεία του πλαισίου θεματοποίησης και θα επανεξετάσουμε τα προεπιλεγμένα θέματα που περιλαμβάνονται στο Jquery Mobile. Θα διερευνήσουμε επίσης τους τρεις τρόπους θεματισμού που μπορεί να αποδοθεί σε στοιχεία. Ενώ όλα τα στοιχεία μπορούν να θεματιστούν με το data-theme χαρακτηριστικό, τα περισσότερα έχουν προεπιλογμένα θέματα και μπορούν επίσης να κληρονομήσουν το θέμα τους από έναν γονέα περιέκτη.

Τέλος, θα δούμε πώς μπορούμε να δημιουργήσουμε το δικό μας θεματισμό. Δημιουργία προσαρμοσμένων θεμάτων θα είναι απαραίτητο αν θέλουμε να δημιουργήσουμε πιο πλούσια σχέδια. Υπάρχουν δύο διαθέσιμες επιλογές για δημιουργία προσαρμοσμένων θεμάτων και θα δούμε βήμα-βήμα τη καθεμία. Η πρώτη είναι η χειροκίνητη προσέγγιση, η οποία δίνει στους σχεδιαστές πλήρη έλεγχο της διάταξης τους. Η δεύτερη είναι η χρήση του ThemeRoller1, ενός web-based εργαλείου που αυτοματοποιεί τη διαδικασία της δημιουργίας νέων θεμάτων.

Σε πολλά παραδείγματα που έχουμε ήδη δει πώς μπορούμε να χρησιμοποιήσουμε το data-theme χαρακτηριστικό για εφαρμογή εναλλακτικών θεμάτων σελιδών (σελίδα, κεφαλίδα, το περιεχόμενο, υποσέλιδο) και τη μορφοποίηση στοιχείων. Για παράδειγμα, μπορούμε να πάρουμε μία μη θεματισμένη σελίδα (βλ. Εικόνα 7-1) και να την θεματίσουμε εκ νέου με μια διαφορετική κεφαλίδα και κατάλογο (βλ. Εικόνα 7-2) με την απλή προσθήκη του data-theme χαρακτηριστικού (βλ. Καταχώρηση 7-1).



Εικόνα 7–1. ***Default theme***



Εικόνα 7–2. ***Alternate theme***

Καταχώρηση 7–1. *data-theme attribute*

```
<div data-role="page">
<div data-role="header" data-theme="b">
<h1>jMovies</h1>
</div>
<div data-role="content">
<ul data-role="listview" data-inset="true" data-theme="a">
<li data-role="list-divider">Now Playing</li>
</ul>
</div>
</div>
```

ΘΕΜΑΤΑ ΚΑΙ ΔΕΙΓΜΑΤΑ

Το αρχείο CSS jQuery Mobile είναι πάντα το πρώτο στοιχείο που εισάγουμε στο στοιχείο κεφαλής (βλέπε Λίστα 7-2). Αυτό το αρχείο περιέχει την προκαθορισμένη δομή και θεματοποίηση για Jquery Mobile εφαρμογές.

Καταχώρηση 7-2. *jQuery Mobile CSS import*

```
<head>
```

<link rel="stylesheet" type="text/css" href="jquery.mobile-min.css" />
<script type="text/javascript" src="jquery-min.js"></script>
<script type="text/javascript" src="jquery.mobile-min.js"></script>
</head>

Το έγγραφο jQuery Mobile CSS χωρίζεται σε δύο τμήματα: ένα θεματισμένο τμήμα και ένα δομικό τμήμα.

 θέμα – Το επάνω μισό του εγγράφου περιέχει τις προεπιλεγμένες ρυθμίσεις θέματος. Οι ρυθμίσεις θέματος διαχειρίζονται την οπτική styling (υπόβαθρο, σύνορα, χρώμα, γραμματοσειρά, σκιές) για όλα τα στοιχεία. Όταν θέτουμε το χαρακτηριστικό data-theme, είμαστε σε θέση να επιλέξουμε από πέντε διαφορετικές επιλογές (α, β, γ, δ, ε). Αυτά τα γράμματα (ae) τεχνικά αναφέρονται ως δείγματα. Εξετάζοντας το αρχείο jQuery Mobile CSS παρατηρούμε ότι το πρώτο δείγμα που εμφανίζεται στο αρχείο CSS είναι το δείγμα "α" (βλέπε Καταχώρηση 7-3).

Καταχώρηση 7–3. *jQuery Mobile CSS swatch “a”*

```
/* A----------------------------------------------------------------------*/
.ui-bar-a {
border: 1px solid #2A2A2A;
background: #111111;
color: #ffffff;
font-weight: bold;
text-shadow: 0 -1px 1px #000000;
...
background-image: linear-gradient(top, #3c3c3c, #111);
}
.ui-body-a {
border: 1px solid #2A2A2A;
background: #222222;
color: #fff;
text-shadow: 0 1px 0 #000;
font-weight: normal;
background-image: linear-gradient(top, #666, #222);
}
```

Το θεματισμένο τμήμα αναλύεται στις ακόλουθες υποενότητες:

 Δείγματα - Από προεπιλογή, το Jquery Mobile έχει πέντε δείγματα για να επιλέξουμε από (α, β, γ, δ, ε) και μπορούμε να προσθέσουμε πολλά μοναδικά δείγματα. Τα δείγματα μας επιτρέπουν να διαμορφώσουμε μοναδικά υπόβαθρα, γραμμές, χρώματα, γραμματοσειρές, και τις σκιές για τα στοιχεία μας. Για απλότητα, η ονομασία για τα νέα δείγματα είναι γράμματα του εύρους (az).

 Παγκόσμιες ρυθμίσεις θέματος - Οι Παγκόσμιες ρυθμίσεις θέματος διαμορφώνονται μετά τα δείγματα. Αυτές οι ρυθμίσεις προσθέσετουν

βελτιώσεις οπτικού styling στα κουμπιά, όπως στρογγυλεμένες γωνίες, εικονίδια, επικαλύψεις, και σκιές. Δεδομένου ότι αυτές οι ρυθμίσεις είναι καθολικές, θα κληρονομηθούν από όλες τις διαμορφώσεις δειγμάτων (βλ. Καταχώρηση 7-4).

Καταχώρηση 7–4. *jQuery Mobile global theme styling*

```
/* Active class used as the "on" state across all themes
------------------------------------------------------------------------*/
.ui-btn-active {
border: 1px solid #155678;
background: #4596ce;
font-weight: bold;
color: #fff;
cursor: pointer;
text-shadow: 0 -1px 1px #145072;
text-decoration: none;
}
```

 Δομή - Το τελευταίο μισό του αρχείου Jquery Mobile CSS περιέχει δομή styling που περιλαμβάνει κατά κύριο λόγο την τοποθέτηση, βάτες, περιθώρια, ρυθμίσεις ύψους και πλάτους (βλ. Καταχώρηση 7-5).

Καταχώρηση 7–5. *jQuery Mobile structure styling*

```
/* some unsets - more probably needed */
.ui-mobile, .ui-mobile body { height: 100%; }
.ui-mobile fieldset, .ui-page { padding: 0; margin: 0; }
.ui-mobile a img, .ui-mobile fieldset { border: 0; }
...
.ui-checkbox, .ui-radio {
position:relative; margin: .2em 0 .5em; z-index: 1;
}
.ui-checkbox .ui-btn, .ui-radio .ui-btn {
margin: 0; text-align: left; z-index: 2;
}
```

Τώρα που έχουμε δει το κύριο αρχείο CSS για Jquery Mobile, ας ρίξουμε μια πιο προσεκτική ματιά στα πέντε δείγματα που περιλαμβάνονται στο Jquery Mobile και να δούμε πώς εμφανίζονται σε πολλές διαφορετικές συνιστώσες (βλέπε εικόνες 7-3-7-6).

Εικόνα 7–3. *Grid swatches*

Εικόνα 7–4. *List swatches*

Εικόνα 7–5. ***Button swatches***

Εικόνα* 7–6.** ***Form field swatches

Για να διατηρήσουμε το στυλ των δειγμάτων σταθερά σε όλες τις συνιστώσες, εδώ είναι τα θέματα που χρησιμοποιούνται για κάθε δείγμα:

- “a” - (black) υψηλότερο επίπεδο οπτικής προτεραιότητας.
- “b” - (blue) δευτερογενές επίπεδο.
- “c” - (gray) βάση.

 “d” - (white/gray) εναλλακτικό δευτερογενές επίπεδο.
 “e” - (yellow) χρώμα έμφασης.

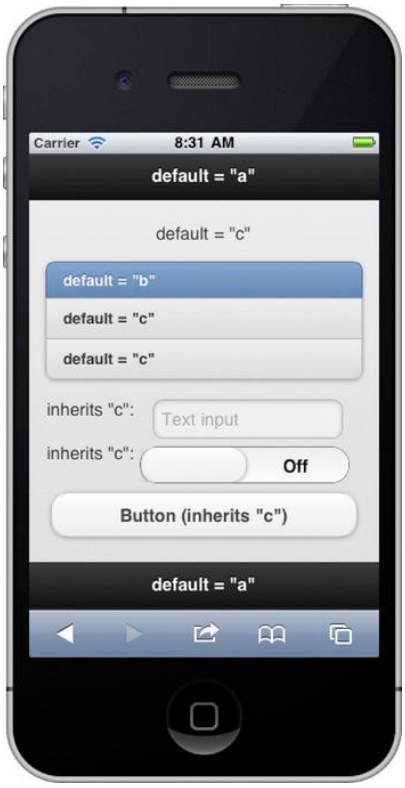
ΠΡΟΕΠΙΛΕΓΜΕΝΑ ΘΕΜΑΤΑ

Εάν δεν προσθέσουμε χαρακτηριστικά data-theme σε μια σελίδατο framework θα εφαρμόσει τα προεπιλεγμένα θέματα για όλες τις σελίδες και τα στοιχεία (βλ. Πίνακα 7-1).

Πίνακας 7-1. Θέματα ανά στοιχείο

Component	Default Theme	Inherit's Parent Theme?	Example
Button	Inherited from parent	Yes	Listing 4-8
Checkbox	Inherited from parent	Yes	Listing 4-20
Content	data-theme=“c”	Yes	Listing 7–6
Dialog	data-theme=”a”	Yes	Listing 2-6
Grid	None	Yes	Listing 6-3
Footer	data-theme=”a”	No	Listing 7–6
Header	data-theme=”a”	No	Listing 7–6
List view	data-theme=”c”	No	Listing 5-1
List badge	data-theme=”c”	No	Listing 5-9
List divider	data-theme=”b”	No	Listing 5-3
List item	data-theme=”c”	Yes (from list only)	Listing 7–6
List split button	data-theme=”b”	No	Listing 5-6
Page	data-theme=”c”	No	Listing 7–5
Radio button	Inherited from parent	Yes	Listing 4-18
Select	Inherited from parent	Yes	Listing 4-16
Slider	Inherited from parent	Yes	Listing 4-22
Switch	Inherited from parent	Yes	Listing 4-24
Text input	Inherited from parent	Yes	Listing 4-13

Για παράδειγμα, εάν δημιουργήσουμε μια βασική Jquery Mobile σελίδα, χωρίς να ορίσουμε τα θέματά της, τα στοιχεία μας θα αποκτήσουν το προεπιλεγμένο θέμα τους ή θα κληρονομούν το θέμα τους. Στην Εικόνα 7-7, προεπιλεγμένα θέματα εφαρμόστηκαν στη σελίδα, κεφαλίδα, υποσέλιδο, περιεχόμενο και στοιχεία λίστας, ενώ τα στοιχεία της φόρμας κληρονομούν τα θέματά τους.



Εικόνα 7–7. ***Page with default and inherited themes***

Με την αναφορά "Προεπιλεγμένα θέματα ανά στοιχείο" πίνακα μας (βλ. Πίνακα 7-1), μπορούμε να καθορίσουμε ποιες προεπιλογές θα πρέπει να εφαρμόζονται για κάθε συστατικό. Ας ρίξουμε μια πιο προσεκτική ματιά στο περιεχόμενο και τα στοιχεία των κουμπιών. Από προεπιλογή, το περιεχόμενο θα έχει data-theme="c". Ωστόσο, το στοιχείο κουμπί δεν έχει προεπιλεγμένο θέμα, έτσι θα κληρονομήσει το προεπιλεγμένο θέμα του. Στη Καταχώρηση 7-6,ο «πατέρας» του κουμπιού είναι το περιεχόμενο. Ως αποτέλεσμα, το κουμπί θα κληρονομήσει θέμα "c".

Καταχώρηση 7–6. *Page with default themes*

```
<div data-role="page">
<div data-role="header">
<h1>default = "a"</h1>
</div>
<div data-role="content">
default = "c"
<ul data-role="listview" data-inset="true">
<li data-role="list-divider">default = "b"</li>
<li>default = "c"</li>
<li>default = "c"</li>
```

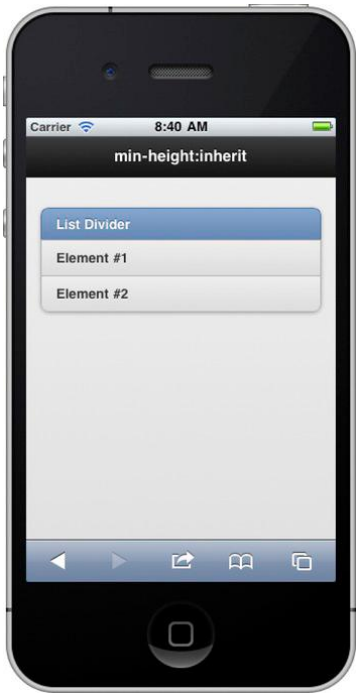
```
</ul>
<form id="test" id="test" action="#" method="post">
<p>
<label for="text">inherits "c":</label>
<input type="text" name="text" id="text" value="" />
</p>
<p>
<label for="sound">inherits "c":</label>
<select name="slider" id="sound" data-role="slider">
<option value="off">Off</option>
<option value="on">On</option>
</select>
</p>
<a href="#" data-role="button">Button (inherits "c")</a>
</form>
</div>
<div data-role="footer" data-position="fixed">
<h3>default = "a"</h3>
</div>
</div>
```

ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΘΕΜΑΤΩΝ

Τα θέματα εφαρμόζονται στα στοιχεία με την ακόλουθη σειρά προτεραιότητας:

1. Σαφή θέματα - Αν ορίσουμε ρητά τα data-theme χαρακτηριστικά για κάθε στοιχείο, αυτό το θέμα θα υπερισχύσει κάθε κληρονομιμένου ή προεπιλεγμένου θέματος.
2. Κληρονομιμένα θέματα - Κληρονομιμένα θέματα θα αντικαταστήσουν όλα τα προεπιλεγμένα θέματα.
3. Προεπιλεγμένα θέματα - Προεπιλεγμένα θέματα εφαρμόζονται όταν δεν εφαρμόζονται τα προηγούμενα δύο. Για λίστα με προεπιλεγμένα θέματα ανά συνιστώσα, ανατρέξτε στον Πίνακα 7-1.

Εικόνα 7–10.
Content height not 100%



Εικόνα 7–11. ***Content height 100%***

CUSTOM ΘΕΜΑΤΑ

Το jQuery Mobile επιτρέπει στους σχεδιαστές να προσαρμόσουν γρήγορα ή να ξαναδομήσουν τη διεπαφή χρήστη. Σε αυτή την ενότητα, θα δούμε πώς μπορούμε να δημιουργήσουμε το δικά μας custom δείγματα. Όπως αξιολόγησσαμε προηγουμένως, το προεπιλεγμένο έγγραφο Jquery Mobile CSS χωρίζεται σε δύο τμήματα: ένα θεματικό και ένα δομικό. Εν συνεχεία θα δημιουργήσουμε ένα custom δείγμα, στο οποίο μπορούμε να βασιστούμε σε περίπτωση επικίνδυνων ενεργειών. Στο Jquery Mobile, μπορούμε δημιουργήσουμε ένα custom δείγμα για να διαχειριστεί το θεματισμό (υπόβαθρο, τα σύνορα, χρώμα, γραμματοσειρά, σκιές) των εικόνων και / ή τα κουμπιά.

Για να δημιουργήσουμε ένα τέτοιο δείγμα τα ακόλουθα βήματα είναι απαραίτητα:

1. Πρώτον, πρέπει να δημιουργήσουμε ένα ξεχωριστό αρχείο CSS για το θέμα (css / θέμα / customtheme. css). Αυτό κρατά τις προσθήκες διαχωρισμένες από την κεντρική Jquery Mobile CSS σελίδα και θα διευκολύνει τις μελλοντικές αναβαθμίσεις.

Καταχώρηση 7-9. *jQuery Mobile's structure file without default themes*

```
<head>
<meta charset="utf-8">
<title>Custom Theme</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel=stylesheet href="css/theme/custom-theme.css" />
<link rel=stylesheet href="css/structure/jquery.mobile.structure.css"/>
<script type="text/javascript" src="jquery-min.js"></script>
<script type="text/javascript" src="jquery.mobile-min.js"></script>
</head>
```

2. Επιλέγουμε ένα υπάρχον δείγμα ως σημείο αναφοράς. Βλέποντας τα δείγματα, αντιγράφουμε ένα που θα μοιάζει πολύ με το ύφος του νέου δείγματος μας. Αυτό θα συμβάλει στην ελαχιστοποίηση του αριθμού των τροποποιήσεων που θα πρέπει να γίνουν προκειμένου να δημιουργήσουμε το νέο δείγμα σας.

3. Στη συνέχεια, αντιγράφουμε το δείγμα βάσης και το επικολλούμε στο αρχείο custom-theme.css. Αργότερα, μετονομάζουμε το δείγμα έτσι ώστε να συνδέεται με ένα μοναδικό γράμμα (fz). Για παράδειγμα, αντικαταστούμε όλα τα επιθέματα με CSS "-e" σε "-v" (βλέπε Καταχώρηση 7-10). Το νέο δείγμα μπορεί τώρα να αναφερθεί με το data-theme = "v".

Καταχώρηση 7-10. *Custom “v” swatch modeled after swatch “e”*

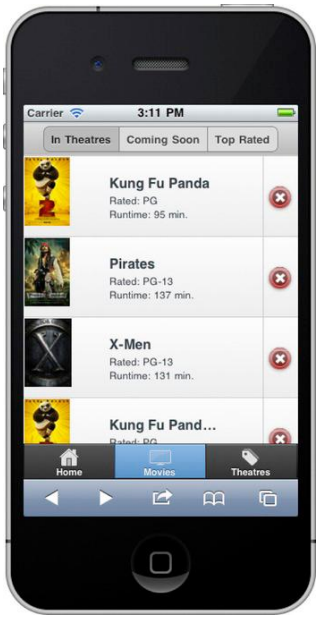
```
/* V------------------------------------------------------------*/
.ui-bar-v {
font-weight: bold;
border: 1px solid #999;
background: #dedede;
color: #000;
text-shadow: 0 1px 0px #fff;
…
}
.ui-btn-up-v {
border: 1px solid #999;
background: #e79696;
color: #fff;
text-shadow: 0 1px 0px #fff;
...
}
```

4. Τώρα το συναρπαστικό έργο της αναβάθμισης των ρυθμίσεων CSS (υπόβαθρο, τα σύνορα, χρώμα, γραμματοσειρά, και σκιές) για το νέο μας δείγμα. Για το "v" δείγμα, αναβαθμίστηκαν όλα τα κουμπιά για να έχουν ένα κόκκινο φόντο με λευκό κείμενο (βλ. Καταχώρηση 7-11).

Καταχώρηση 7-11. *Update “v” swatch buttons with red background gradient and white text*

```
/* V------------------------------------------------------------*/
.ui-btn-up-v {
border: 1px solid #999;
background: #e79696;
color: #fff;
text-shadow: 0 1px 0px #fff;
background-image: -webkit-gradient(
linear, 0% 0%, 0% 100%, from(#E79696), to(#ce2021),
color-stop(.4,#E79696)
);
background-image: -webkit-linear-gradient(
0% 56% 90deg,#CE2021, #E79696, #E79696 100%
);
background-image: -moz-linear-gradient(
0% 56% 90deg,#CE2021, #E79696, #E79696 100%
);
}
```

5. Στη συνέχεια, πρέπει να ενσωματώσουμε το νέο μας "v" δείγμα με μια πραγματική σελίδα για δοκιμή. Δημιούργησαμε δύο σελίδες για να βοηθήσουμε τη δοκιμή του νέου "v" δείγματος. Στην πρώτη σελίδα, θέλαμε να δούμε πώς το νέο δείγμα εμφανίστηκε σε ένα εικονίδιο κουμπί. Για τη δοκιμή αυτή, δημιουργήσαμε μια διαχωρισμένη λίστα με κουμπιά με το δεξί κουμπί ως εικονίδιο διαγραφής και θεματισμένο το δευτεροβάθμιο κουμπί με το νέο μας "v" δείγμα (βλ. Εικόνα 7-12 και σχετικός κώδικας του στην Καταχώρηση 7-12).

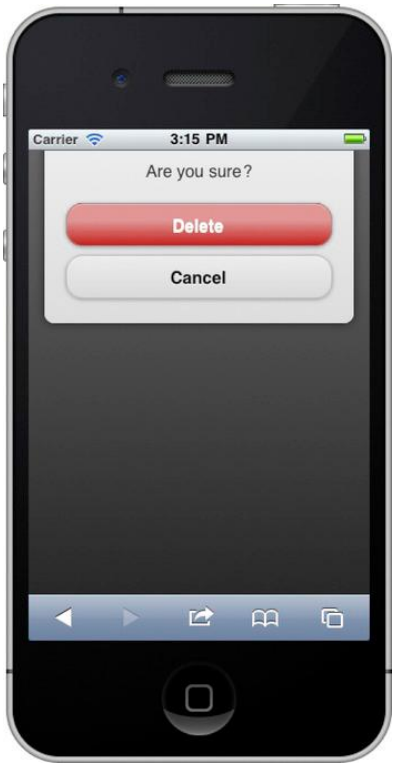


Εικόνα 7-12. ***Red icon background for potentially dangerous actions***

Καταχώρηση 7-12. *Split buttons with “v” swatch*

```
<head>
<link rel=stylesheet href="css/theme/custom-theme1.css" />
<link rel=stylesheet href="css/structure/jquery.mobile-min.css"/>
...
</head>
<ul data-role="listview" data-split-icon="delete" data-split-theme="v">
<li>
<a href="#">
<img src="../images/kungfupanda2.jpg" />
<h3>Kung Fu Panda</h3>
<p>Rated: PG</p>
<p>Runtime: 95 min.</p>
</a
<a href="#delete" data-transition="slidedown">Delete</a>
</li>
```

Επίσης, εισάγαμε το νέο μας αρχείο theme.css πριν από το κύριο αρχείο Jquery Mobile CSS. Για τη δεύτερη δοκιμή μας, θα εφαρμόσαμε το νέο "v" δείγμα σε ένα κουμπί διαγραφής. Για αυτή τη δοκιμή, δημιουργήσαμε ένα παράθυρο διαλόγου για να επιβεβαιώσουμε τη δυνητικά επικίνδυνη δράση και θεματίσαμε το κουμπί διαφραφής με το νέο κόκκινο θέμα μας (βλ. Εικόνα 7-13).



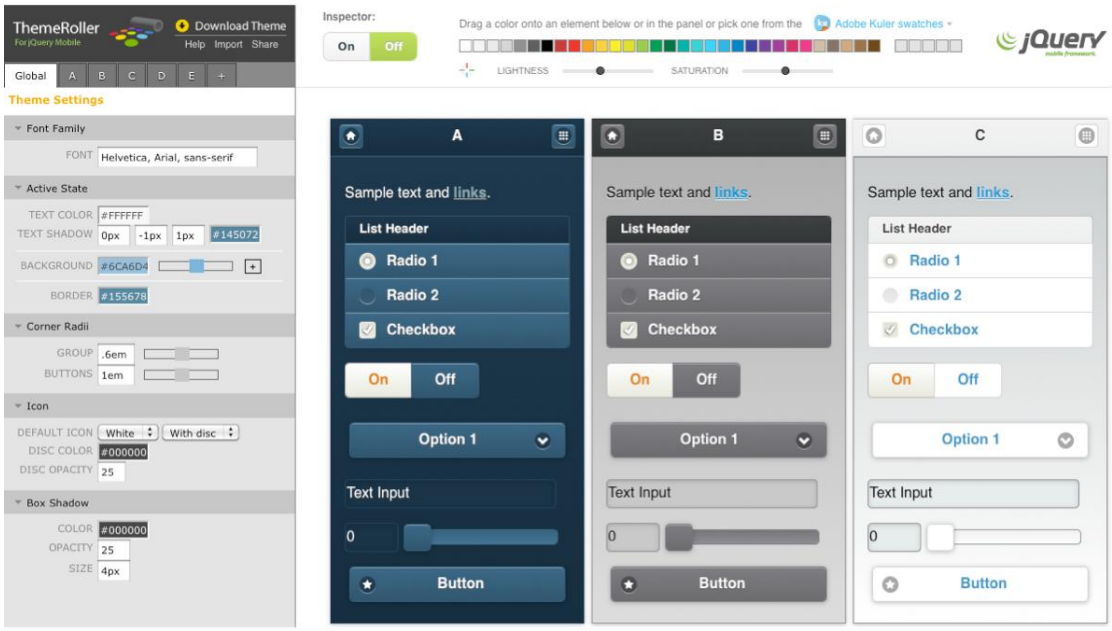
Εικόνα 7-13. ***Red delete button for delete action***

Καταχώρηση 7-13. *Delete button with “v” swatch*

```
<div data-role="dialog" id="delete">
<div data-role="content" data-theme="c">
<span class="title">Are you sure?</span>
<a href="#home" data-role="button" data-theme="v">Delete</a>
<a href="#home" data-role="button" data-rel="back">Cancel</a>
</div>
</div>
```

ThemeRoller

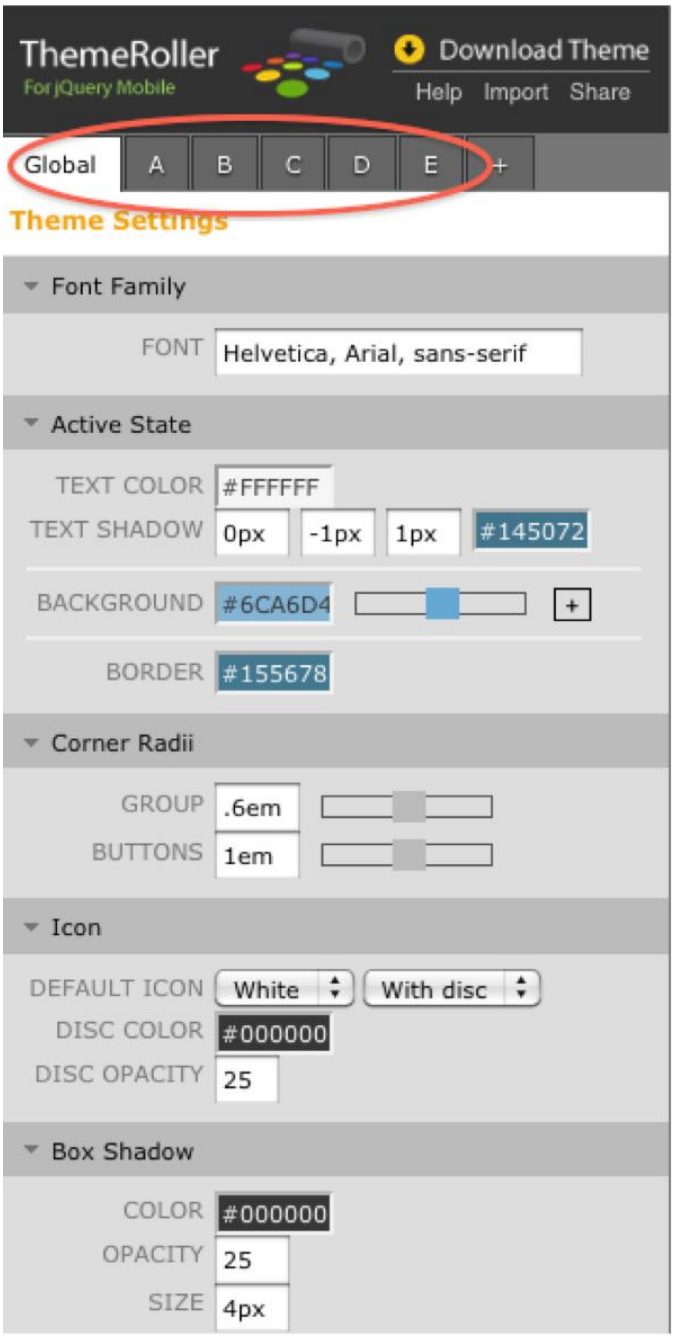
Το ThemeRoller είναι ένα web-based εργαλείο που μας βοηθά να αυτοματοποιήσουμε τη διαδικασία της δημιουργίας νέων CSS θεμάτων για Jquery Mobile. Αυτό είναι ένα πολύ χρήσιμο εργαλείο, διότι μας επιτρέπει να κάνουμε ενημερώσεις των χρωμάτων στο αριστερό τμήμα του παραθύρου και τη δυνατότητα προεπισκόπησης των αποτελεσμάτων στο δεξί τμήμα του παραθύρου μέσα σε μια πραγματική διάταξη jQuery Mobile (βλέπε εικόνα 7-14).



Εικόνα 7-14. *ThemeRoller*

ΔΕΙΓΜΑΤΑ ΚΑΙ ΡΥΘΜΙΣΕΙΣ

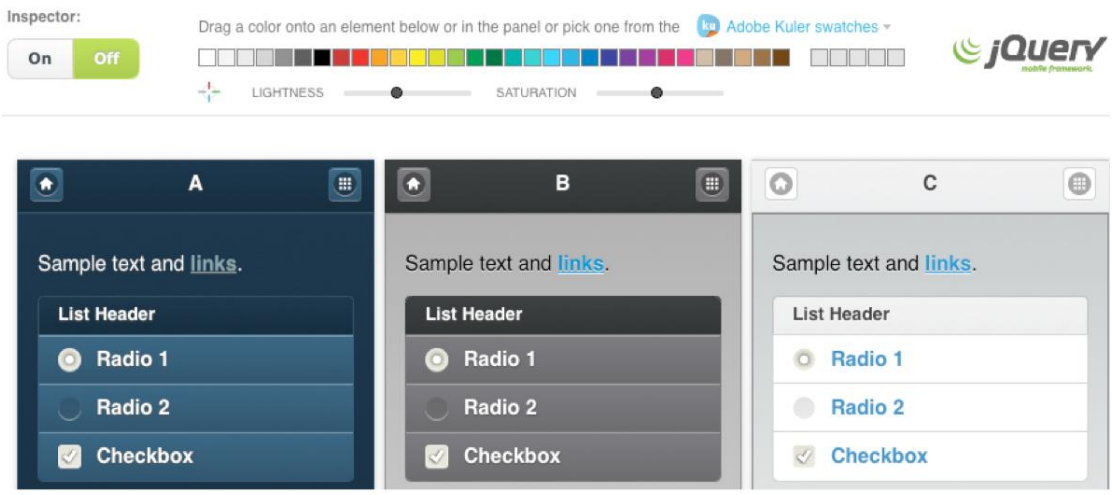
Μπορούμε να προσαρμόσουμε γρήγορα τις ιδιότητες CSS που ισχύουν σε παγκόσμιο επίπεδο σε όλα τα δείγματα κάτω απο τη καρτέλα "Global" που εμφανίζεται στο αριστερό παράθυρο. Εδώ μπορούμε να ρυθμίσουμε τη γραμματοσειρά, χρώματα, γωνία ακτίνων, εικόνες, και τις σκιές (βλ. Εικόνα 7-15).



Εικόνα 7-15. *Global Theme Settings*

Preview Inspector and QuickSwatch Bar

Για να γίνει ακόμα πιο εύκολη η οικοδομήση των θέματα υπάρχουν δύο μοναδικά εργαλεία: Preview Inspector και η QuickSwatch Bar. Το Preview Inspector είναι μια εναλλαγή που μπορεί να είναι είτε ενεργή είτε ανενεργή (βλ. Εικόνα 7-16). Στην ενεργή κατάσταση κάνοντας κλικ σε ένα στοιχείο στο παράθυρο προεπισκόπησης θα εμφανιστεί αυτόματα τα επεξεργάσιμα χαρακτηριστικά στο αριστερό παράθυρο.



Εικόνα 7-16. ***Preview inspector and QuickSwatch bar***

Η QuickSwatch Bar είναι ένα φάσμα χρωμάτων που εμφανίζεται στα δεξιά του Inspector (βλ. Εικόνα 7-16). Αυτό είναι ένα ισχυρό εργαλείο που μας επιτρέπει να επιλέξουμε και να «σύρουμε» οποιοδήποτε χρώμα πάνω ένα στοιχείο στη σελίδα προεπισκόπησης ή σε ένα χαρακτηριστικό χρώματος στο αριστερό παράθυρο. Κάτω από το Bar QuickSwatch είναι δύο ρυθμιστικά για να προσαρμόσουμε τη φωτεινότητα και τον κορεσμό των χρωμάτων. Επιπλέον, τα πιο πρόσφατα επιλεγμένα χρώματα θα εμφανίζονται στα δεξιά του χρωματικού φάσματος για γρήγορη επαναχρησιμοποίηση.

Adobe Kuler Integration

Μπορεί να είναι απιτητικό όταν πρέπει να δημιουργήσουμε μια παλέτα χρωμάτων από το μηδέν. Για να απλουστευτεί αυτή η διαδικασία, το Themeroller έχει το Kuler της Adobe ενσωματομένο (βλ. Εικόνα 7-17).

Εικόνα 7-17. ***Adobe's Kuler App***

Το Kuler είναι μία ιστοσελίδα που επιτρέπει στους ανθρώπους να δημιουργούν, να μοιράζονται και να αξιολογούν παλέτες χρωμάτων. Για να δούμε τις παλέτες που είναι διαθέσιμες στο Kuler κάνουμε κλικ στο σύνδεσμο "Adobe Kuler" που εμφανίζεται πάνω από την QuickSwatch Bar. Όταν η εφαρμογή Kuler ανοίγει, ένα φίλτρο αναζήτησης εμφανίζεται στο αριστερό τμήμα του παραθύρου που μας επιτρέπει να φιλτράρουμε από την πιό τελευταία, δημοφιλής, αξιολόγηση, ή προσαρμοσμένη αναζήτηση. Όταν βρούμε ένα χρώμα ενδιαφέρον, απλά το επιλέγουμε και το «σέρνουμε» πάνω σε ένα στοιχείο στο παράθυρο προεπισκόπησης μας.

ΞΕΚΙΝΩΝΤΑΣ

Για σκοπούς σύγκρισης, πρόκειται να δημιουργήσουμε ένα κόκκινο δείγμα στην Themeroller να να δούμε τις διαφορές με το δείγμα που δημιουργήσαμε στη προηγούμενη ενότητα. Πρόκειται να παρακάμψουμε το προεπιλεγμένο Jquery Mobile, "e" δείγμα με το νέο κόκκινο δείγμα. Στο Themeroller, για να ενημερώσουμε ένα θέμα τα εξής βήματα είναι απαραίτητα:

1. Στο Themeroller, εισαγάγουμε ένα υπάρχον θέμα κάνοντας κλικ στο σύνδεσμο "Εισαγωγή" στην πάνω αριστερή γωνία (βλ. Εικόνα 7-18). Για την άσκηση αυτή θα εισάγουμε και θα τροποποιήσουμε το προεπιλεγμένο θέμα Jquery Mobile.

Εικόνα 7-18. ***Import Existing Theme***

2. Αφού εισαχθεί το θέμα, προσδιορίζουμε το δείγμα ώστε να το τροποποιήσουμε. Για αυτό το βήμα, πρόκειται να τροποποιήσουμε το προεπιλεγμένο "e" δείγμα.

3. Στη συνέχεια, θα βρούμε ένα κατάλληλο χρώμα βάσης για το κόκκινο δείγμα μας. Μπορούμε να βρούμε κατάλληλο κόκκινο χρώμα είτε στο Bar QuickSwatch ή στο εργαλείο ένταξης Kuler.

4. Μετά την εύρεση του κατάλληλου χρώματος βάσης μπορούμε να ενημερώσουμε τα στοιχεία μας στο παράθυρο προεπισκόπησης με το επιλεγμένο χρώμα.

5. Στο παράθυρο προεπισκόπησης, κάνουμε τις απαραίτητες προσαρμογές. Μπορούμενα ρυθμίσουμε τα χρώματα ελαφρώς ή να προσθέσουμε ανεπαίσθητες επιδράσεις με στοιχεία φόντου. Όπως ήταν αναμενόμενο, Themeroller καθιστά τη διαδικασία της δόμισης και της προεπισκόπησης πολύ περισσότερο αποτελεσματική σε σύγκριση με «χειρονακτική» διαδικασία δόμισης χρωμάτων.

6. Αφού έχουμε επιλέξει το νέο θέμα, μπορούμε να «κατεβάσουμε» το CSS του θέματος, κάνοντας κλικ στο σύνδεσμο "Λήψη Theme" στην επάνω αριστερή γωνία του Themeroller (βλ. Εικόνα 7-19).

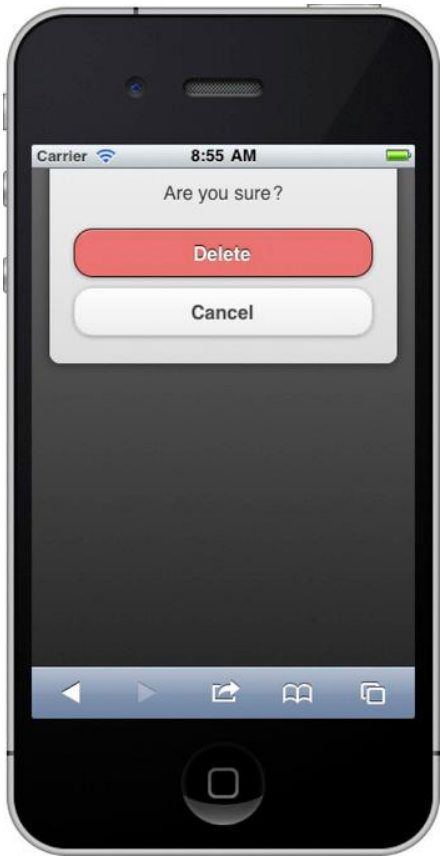
Εικόνα 7-19. ***Download Theme***

7. Μπορούμε τώρα να ενσωματώνουμε το νέο θέμα μας στην αίτησή μας (βλ. Καταχώρηση 7-14 και των συναφών στιγμιότυπων οθόνης στην Εικόνα 7-20).

Καταχώρηση 7-14. *ThemeRoller custom theme import*

```
<head>
<meta charset="utf-8">
<title>Custom Theme</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel=stylesheet href="css/theme/custom-theme2.css" />
<link rel=stylesheet href="css/structure/jquery.mobile.structure.css"/>
<script type="text/javascript" src="jquery-min.js"></script>
<script type="text/javascript" src="jquery.mobile-min.js"></script>
</head>
<div data-role="dialog" id="delete">
```

```
<div data-role="content" data-theme="c">
<span class="title">Are you sure?</span>
<a href="#home" data-role="button" data-theme="e">Delete</a>
<a href="#home" data-role="button" data-rel="back">Cancel</a>
</div>
</div>
```



Εικόνα 7-20. ThemeRoller's red delete button

ΚΕΦΑΛΑΙΟ 7:

Jquery Mobile API

Όλα τα καλά δομημένα frameworks επιτρέπουν στους προγραμματιστές να επεκτείνουν και να αντικαθιστούν τις προεπιλεγμένες ρυθμίσεις διαμόρφωσης. Επιπλέον, παρέχουν μεθόδους ευκολία να συμβάλει στην απλούστευση τον κωδικό σας. Το Jquery Mobile περιλαμβάνει ένα αρκετά εκτεταμένο API που εκθέτει κάθε ένα από αυτά τα χαρακτηριστικά. Κατ 'αρχάς, θα εξετάσουμε το πώς να ρυθμίζουμε το Jquery Mobile. Εμείς θα επανεξετάσουμε κάθε χαρακτηριστικό ρύθμισης στο Jquery Mobile, και θα δούμε πώς το API μας επιτρέπει να ρυθμίσουμε κάθε επιλογή. Στη συνέχεια, θα εξετάσουμε τις πιο δημοφιλείς μεθόδους, σελίδα γεγονότα, και τις ιδιότητες. Αυτά τα χαρακτηριστικά API είναι χρήσιμα όταν θέλουμε να ενημερώσουμε Mobile εφαρμογές Web.Τέλος, θα επανεξετάσουμε μια ταξινομημένη λίστα όλων των χαρακτηριστικών στοιχείων Jquery Mobile.

Διαμορφώνοντας το Jquery Mobile

Όταν το Jquery Mobile προετοιμάζει, πυροδοτεί ένα event mobileinit σχετικά με το αντικείμενο του εγγράφου. Μπορούμε να συνδεθούμε με το mobileinit και να εφαρμόσουμε παρακάμψεις στο Jquery Mobile, ($. mobile) προεπιλεγμένες ρυθμίσεις διαμόρφωσης. Επιπλέον, μπορούμε να επεκτείνουμε το Jquery Mobile με τα πρόσθετα συμπεριφοράς και των ιδιοτήτων. Για παράδειγμα, υπάρχουν δύο τρόποι για να ρυθμίσουμε το Jquery Mobile ως φαίνεται στα παρακάτω παραδείγματα.

Παραδείγματα:

// Configure properties via jQuery's extend method

```
$( document ).bind( "mobileinit", function(){
$.extend( $.mobile, {
// Override loading message
loadingMessage: "Loading...",
// Override default transition from “slide” to “pop”
defaultTransition: "pop"
});
});
```

// Configure properties individually

```
$( document ).bind( "mobileinit", function(){
$.mobile.loadingMessage = "Initializing";
$.mobile.defaultTransition = "slideup";
});
```

Εφόσων το mobileinit ενεργοποιείται αμέσως μετά την εκτέλεση του Jquery Mobile, θα πρέπει να έχουμε τοποθετήσει προσαρμοσμένη δέσμη ενεργειών πριν από το αρχείο Jquery Mobile JavaScript.

Παράδειγμα:

```
<head>
<script type="text/javascript" src="jquery-min.js"></script>
<script type=”text/javascript” src="custom-scripts-here.js"></script>
<script type="text/javascript" src="jquery.mobile-min.js"></script>
</head>
```

ΔΙΑΜΟΡΦΩΝΟΝΤΑΣ JQUERY ΕΠΙΛΟΓΕΣ

Οι παρακάτω είναι παραμετροποιήσιμες $.Mobile επιλογές που μπορούμε να παρακάμψουμε στο JavaScript.

- **activeBtnClass**(string, default: "ui-btnactive")

Η κλάση CSS χρησιμοποιείται για την αναγνώριση και το στυλ του «ενεργού» κουμπιού. Αυτό το χαρακτηριστικό CSS χρησιμοποιείται συνήθως για να προσδιορίσει το ύφος του ενεργού κουμπιού σε ένα μπαρ καρτέλας.

- **activePageClass**(string, default: "ui-page-active")

Η κλάση CSS αποδίδεται στην σελίδα ή στο παράθυρο διαλόγου που είναι τη στιγμή εκείνη ορατή και ενεργή. Για παράδειγμα, όταν πολλές σελίδες που έχουν τοποθετηθεί στο DOM η ενεργή σελίδα θα έχει εφαρμόσει αυτό το χαρακτηριστικό CSS.

- **ajaxEnabled**(boolean, default: true)

Δυναμική φόρτωση σελίδων μέσω Ajax, όταν είναι δυνατόν. Ajax φόρτωσης είναι ενεργοποιημένο από προεπιλογή για όλες τις σελίδες εκτός από εξωτερικές διευθύνσεις URL, ή συνδέσμους που είναι σημειωμένοι με ένα rel ="external'' ή target="_blank" χαρακτηριστικό. Αν το Ajax είναι απενεργοποιημένο, οι σύνδεσμοι της σελίδας θα φορτωθεί με HTTP αιτήσεις, χωρίς CSS μεταβάσεις.

 allowCrossDomainPages(boolean, default: false)

Κατά την ανάπτυξη με PhoneGap, συνιστάται να θέσουμε αυτή τη δυνατότητα διαμόρφωσης σε ενεργή κατάσταση. Αυτό επιτρέπει στο Jquery Mobile τη διαχείριση της λογικής φόρτωση της σελίδας των cross-domain αιτήσεων στο PhoneGap.

 defaultDialogTransition(string, default: "pop")

Η προεπιλεγμένη μετάβαση για χρήση κατά τη μετάβαση σε ένα παράθυρο διαλόγου. Μπορούτε να ορίσουμε τη μετάβαση σε "none" για να μην υπάρξει εφέ μετάβασης.

 hashListeningEnabled(boolean, default: true)

Αυτόματα φορτώνει και δείχνει τις σελίδες που βασίζονται σε location.hash. Το Jquery Mobile «ακούει» τις αλλαγές location.hash ώστε να φορτώσει τις εσωτερικές σελίδες εντός του DOM. Μπορούμε να απενεργοποιήσουμε αυτή την επιλογή και να χειριστούμε τις αλλαγές hash manually ή να απενεργοποιήσουμε αυτή την επιλογή για να έχουμε πρόσβαση στο σελιδοδείκτη ως σύνδεση.

 loadingMessage(string, default: "loading")

Ορίζει το μήνυμα φόρτωσης, ώστε να εμφανιστεί κατά τη διάρκεια αιτήσεων Ajax. Επιπροσθέτως, μπορούμε να ορίσουμε μια ψεύτικη (boolean) για να απενεργοποιήσουμε το μήνυμα. Επίσης, αν θέλουμε να ενημερώσουμε το μήνυμα φόρτωσης κατά το χρόνο εκτέλεσης σε κάθε σελίδα μπορούμε να το ενημερώσουμε μέσα στη σελίδα.

Παράδειγμα:

```
// Update loading message
$.mobile.loadingMessage = "My custom message!";
// Show loading message
$.mobile.showPageLoadingMsg();
```

 minScrollBack(string, default: 250)

Ορίζει την ελάχιστη απόσταση κύλισης που θα θυμάται, όταν επιστρέφει στη σελίδα. Κατά την επιστροφή σε μια σελίδα, το πλαίσιο θα μεταβεί αυτόματα στη θέση ή τη σύνδεσης που ξεκίνησε τη μετάβαση, όταν η θέση κύλισης του εν λόγω συνδέσμου είναι πέρα από τη ρύθμιση minScrollBack. Από προεπιλογή, το όριο κύλισης είναι 250 pixels. Αν θέλουμε να εξαλείψουμε την ελάχιστη ρύθμιση, ώστε το framework πάντα να κυλάει ανεξάρτητα από τη θέση κύλισης, ορίζουμε αυτή την τιμή σε "0". Αν θέλουμε να απενεργοποιήσουμε αυτό το χαρακτηριστικό, ορίζουμε την τιμή σε "άπειρο".

 ns(string, default: "")

Τα δεδομένα χαρακτηριστικά είναι ένα νέο χαρακτηριστικό στην HTML5. Για παράδειγμα, το «data-role» είναι το προεπιλεγμένο namespace για το χαρακτηριστικό ρόλο. Αν θέλουμε να παρακάμψουμε την προεπιλεγμένη ρύθμιση ονομάτων σε παγκόσμιο επίπεδο θα παρακάμψουμε το $. mobile.ns.option

Παράδειγμα:

// Set a custom namespace
$.mobile.ns = "jqm-";
As a result, all of your jQuery Mobile data-* attributes will require the prefix "data-jqm-". For instance, the "data-role" attribute now becomes "data-jqmrole".

ΜΕΘΟΔΟΙ

Το Jquery Mobile εκθέτει μια σειρά από μεθόδους που είναι χρήσιμες όταν θέλουμε να ενημέρωσουμε τις Mobile Web εφαρμογές μας.

 $.mobile.changePage()

Η λειτουργία changePage χειρίζεται όλες τις λεπτομέρειες της μετάβασης από τη μία σελίδα στην άλλη.

Χρήση:

$.mobile.changePage(toPage, [options])

 toPage (sting or jQuery collection). Η σελίδα στην οποία θα μεταφερθεί.

 toPage (string). Ένας φάκελος URL ("contact.html") ή η id εσωτερικών στοιχείων

("#contact").

 toPage (object). Μία συλλογή Jquery εμπεριέχοντας ένα στοιχείο σελίδας: $(“#contactPage”)

 allowSamePageTransition (boolean, default: false).

Η μέθοδος changePage θα αγνοήσει τα αιτήματα που μεταβαίνουν στην ίδια σελίδα. Ορίζουμε αυτή την επιλογή true για να επιτρέψουμε ίδιες μεταβάσεις.

 changeHash (boolean, default: true).

Ενημέρωση του κατακερματισμού της URL toPage όταν η αλλαγή σελίδας είναι πλήρης.

 data (string or object, default: undefined).

Τα δεδομένα για να στείλουμε σε ένα Ajax αίτημα.

 dataUrl (string, default: toPage URL).

Ορίζει τη διεύθυνση URL για να δείξει στον πεδίο τοποθεσίας του προγράμματος περιήγησης.

 reloadPage (boolean, default: false).

Ανανέωση της σελίδας ακόμα κι αν είναι ήδη στο DOM της σελίδας.

 transition (string, default: $.mobile.defaultTransition).

Η μετάβαση στην αίτηση για την αλλαγή σελίδας. Η προεπιλεγμένη μετάβαση είναι η slide.

 type (string, default: “get”).

Καθορίζει τη μέθοδο ("Get" ή "Post") για χρήση όταν κάνουμε μια αίτηση σελίδας.

Παράδειγμα #1:

```
//Transition to the "contact.html" page.
$.mobile.changePage( "contact.html" );
<!-- Markup equivalent when link clicked -->
<a href="contact.html">Contact Us</a>
```

Παράδειγμα #2:

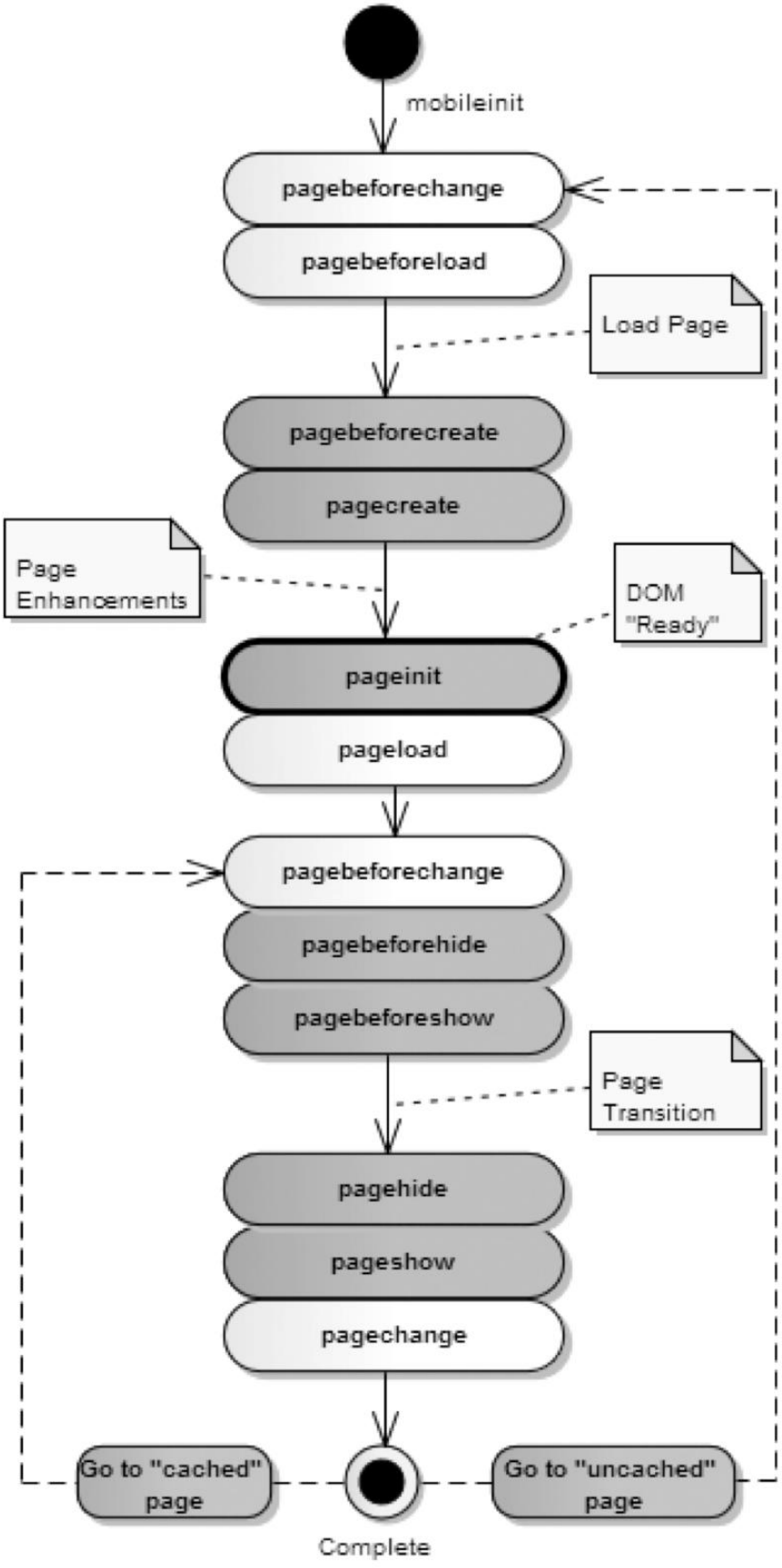
```
// Transition to the internal "#contact" page with a reverse "pop" transition.
$.mobile.changePage( ”#contact”, { transition: "pop", reverse: true } );
<!-- Markup equivalent when link clicked -->
<a href="contact.html" data-transition="pop" datadirection="
reverse">Contact</a>
```

Παράδειγμα #3:

```
/* Dynamically create a new page and open it */
// Create page markup
var newPage = $("<div data-role=page data-url=hi><div data-role=header>
<h1>Hi</h1></div><div data-role=content>Hello Again!</div></div>");
// Add page to page container
newPage.appendTo( $.mobile.pageContainer );
// Enhance and open new page
$.mobile.changePage( newPage );
```

EVENTS

Το Jquery Mobile εκθέτει επίσης διάφορα events που είναι χρήσιμα όταν θέλουμε να εφαρμόσουμε πριν ή μετά την επεξεργασία κατά τη διάρκεια των αλλαγών της σελίδας στη Mobile Web εφαρμογή. Σε αυτή την ενότητα θα επανεξετάσουμε ένα μέρος των Jquery Mobile events. Ας ξεκινήσουμε με ένα διάγραμμα (Σχήμα 8-1). Αυτό το διάγραμμα δείχνει τα κύρια γεγονότα της σελίδας που θα συμβούν μέσα στο Jquery Mobile και βοηθά στην απεικόνηση της αλληλουχίας της κάθε εκδήλωσης στην σελίδα.



Εικόνα 8-1. *jQuery Mobile Page Events*

ΑΛΛΑΓΗ ΣΕΛΙΔΑΣ

Οι εκδηλώσεις αλλαγών σελίδας ενεργοποιειούνται αυτόματα κατά την πλοήγηση σε άλλη σελίδα. Εσωτερικά, αυτά τα γεγονότα είναι σκανδάλη όταν η $.mobile.changePage μέθοδος καλεστεί. Κατά τη διάρκεια αυτής της διαδικασίας, τα δύο γεγονότα θα εκτελεστούν. Το πρώτο συμβάν που ενεργοποιείται είναι το pagebeforechange. Το επόμενο συμβάν που θα ενεργοποιηθεί εξαρτάται από την κατάσταση της αλλαγής της σελίδας. Όταν η αλλαγή σελίδας είναι επιτυχής, το συμβάν pagechange θα ενεργοποιειθεί και αν αποτύχει η αλλαγή σελίδας, τα pagechangefailed events ενεργοποιούνται.

 pagebeforechange

Αυτό είναι το πρώτο συμβάν που ενεργοποιείται κατά τη διάρκεια μιας αλλαγής σελίδας. Callbacks για αυτό το γεγονός έχουν περάσει δύο επιχειρήματα. Το πρώτο είναι event και το δεύτερο επιχείρημα είναι ένα αντικείμενο δεδομένων.

 toPage (string). Ένας φάκελος URL ή ένα Jquery Collection αντικείμενο. Αυτό είναι το ίδιο επιχείρρημα το οποίο έχει περαστεί στο $.mobile.changePage().

 options (object). Αυτές είναι οι ίδιες επιλογές οι οποίες έχουν περαστεί στο $.mobile.changePage.

Example:

```
$( document ).bind( "pagebeforechange", function( e, data ) {
console.log(“Change page starting...”);
// Get the page
var toPage = data.toPage;
// Get the page options
var options = data.options;
// Inspect toPage or override options (redirect)…
// Prevent a page change
e.preventDefault();
});
```

Παράδειγμα:

```
$( document ).bind( "pagebeforechange", function( e, data ) {
console.log(“Change page starting...”);
// Get the page
var toPage = data.toPage;
// Get the page options
```

```
var options = data.options;
// Inspect toPage or override options (redirect)…
// Prevent a page change
e.preventDefault();
});
```

 pagechange

Αυτό είναι το τελευταίο συμβάν για να ενεργοποιηθεί μετά από μια επιτυχημένη αλλαγή σελίδας. Callbacks γι 'αυτό έχουν περαστεί δύο επιχειρήματα. Το πρώτο είναι event και το δεύτερο επιχείρημα είναι ένα αντικείμενο δεδομένων. Το αντικείμενο δεδομένων, πέρασε ως το δεύτερο επιχείρημα, περιέχει το ακόλουθες ιδιότητες:

 toPage (string). Ένας φάκελος URL ή ένα jQuery Collection αντικείμενο. Αυτί είναι το ίδιο επιχείρρημα που είχε περαστεί στο $.mobile.changePage().

 options (object). Αυτές είναι οι ίδιες επιλογές οι οποίες έχουν περαστεί στο $.mobile.changePage.

Example:

```
$( document ).bind( "pagechange", function( e, data ){
console.log(“Change page successfully completed...”);
var toPage = data.toPage;
var options = data.options;
});
```

ΑΡΧΙΚΟΠΟΙΗΣΗ ΣΕΛΙΔΑΣ

Συμβάντα αρχικοποίησης σελίδας ενεργοποιούνται στη σελίδα πριν και μετά το Jquery Mobile βελτιστοποιήσει τη σελίδα. Αυτά τα συμβάντα ενεργοποιούνται μόνο μία φορά κατά τη διάρκεια του κύκλου ζωής μιας σελίδας.

 pagebeforecreate

Ενεργοποιείται κατά τη διάρκεια μιας αλλαγής σελίδας. Αυτό το συμβάν επέρχεται αφού το δοχείο σελίδας έχει εισαχθεί στο DOM, αλλά πριν την σελίδα να ενισχυθεί. Αυτή είναι η προτιμώμενη θέση να προ-αναλύσει τη σήμανση πριν το πλαίσιο ενισχύει τη σελίδα.

Παράδειγμα:

```
$( "#to-page-id" ).live( "pagebeforecreate", function(){
console.log( "Pre-parse the markup before the framework enhances the
widgets" );
});
```

 pagecreate

Ενεργοποιείται κατά τη διάρκεια μιας αλλαγής σελίδας. Αυτή είναι το γεγονός που ενεργοποιείται από το framework για την προετοιμασία όλων των plugins της σελίδας.

Παράδειγμα:

```
$( "#to-page-id" ).live( "pagecreate", function(){
console.log(“Page plugins are being initialized...”);
// Initialize custom plugins
( “:jqmData(role='my-plugin')” ).myPlugin();
});
```

 pageinit

Ενεργοποιείται στη σελίδα που αρχικοποιείται αφού οι βελτιώσεις είναι πλήρεις. Η σελίδα είναι τώρα σε ένα DOM έτοιμο στάδιο.

Παράδειγμα:

```
$( "#to-page-id" ).live( "pageinit", function(){
console.log(“The page has been enhanced...”);
// Attach event handlers or run other jQuery code...
});
```

ΜΕΤΑΒΑΣΗΣ ΣΕΛΙΔΑΣ

Συμβάντα μετάβασης σελίδας ενεργοποιούνται για τις "από" και "έως" σελίδες κατά τη διάρκεια μιας μετάβασης σελίδας.

- **pagebeforehide**

Ενεργοποιείται στην "από" σελίδα στην έναρξη της μετάβασης. Το συμβάν λαμβάνει χώρα πριν από το pagebeforeshow συμβάν. Θα ενεργοποιηθεί μόνο αν η αίτηση αλλαγής σελίδας έχει μια συνδεδεμένη "από" σελίδα. Οι Callbacks για αυτό το συμβάν έχουν δύο επιχειρήματα. Το πρώτο είναι το συμβάν και το δεύτερο επιχείρημα είναι ένα αντικείμενο δεδομένων. Το αντικείμενο δεδομένων, που περιέχει τις ακόλουθες ιδιότητες:

- nextPage (object). Έια Jquery συλλογής αντικείμενο εμπεριέχοντας το στοιχείο της σελίδας στο οποίο μεταφερόμαστε.

Παράδειγμα:

```
$( "#from-page-id" ).live( "pagebeforehide", function( e, data ){
console.log( "The page transition is just starting..." );
});
```

 pagebeforeshow

Ενεργοποιείται στη "να" σελίδα αφού η σελίδα έχει ενισχυθεί και λίγο πριν η μετάβαση σελίδας αρχίσει. Οι Callbacks για την εκδήλωση αυτή έχουν δύο επιχειρήματα. Το πρώτο είναι το συμβάν και το δεύτερο επιχείρημα είναι ένα αντικείμενο δεδομένων, που περιέχει τις ακόλουθες ιδιότητες:

 prevPage (object). Ένα Jquery συλλογής αντικείμενο εμπεριέχοντας το στοιχείο της σελίδας απο το οποίο μεταφερόμαστε.

Example:
```
$( "#to-page-id" ).live( "pagebeforeshow", function( e, data ){
console.log( "The page transition is just starting..." );
});
```

 pagehide

Ενεργοποιείται στην "από" σελίδα αφού η μετάβαση είναι πλήρης και πριν από το pageshow συμβάν. Θα ενεργοποιηθεί μόνο αν η αίτηση αλλαγής σελίδας έχει μία συνδεδεμένη "από" σελίδα. Οι Callbacks για την εκδήλωση αυτή έχουν δύο επιχειρήματα. Το πρώτο είναι το συμβάν και το δεύτερο επιχείρημα είναι ένα αντικείμενο δεδομένων, που περιέχει τις ακόλουθες ιδιότητες:

 nextPage (object). Έια Jquery συλλογής αντικείμενο εμπεριέχοντας το στοιχείο της σελίδας στο οποίο μεταφερόμαστε.

 pageshow

Ενεργοποιείται στη "να" σελίδα αφού η μετάβαση είναι πλήρης και αφού η "από" σελίδα έχει κρυφτεί. Οι Callbacks για την εκδήλωση αυτή έχουν δύο επιχειρήματα. Το πρώτο είναι το γεγονός και το δεύτερο επιχείρημα είναι ένα αντικείμενο δεδομένων. Το πρώτο είναι το συμβάν και το δεύτερο επιχείρημα είναι ένα αντικείμενο δεδομένων, που περιέχει τις ακόλουθες ιδιότητες:

 prevPage (object). Έια Jquery συλλογής αντικείμενο εμπεριέχοντας το στοιχείο της σελίδας απο το οποίο μεταφερόμαστε.

Example:

```
$( "#to-page-id" ).live( "pageshow", function( e, data ){
console.log( "The page transition is complete!" );
});
```

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΔΕΔΟΜΕΝΩΝ

Τα χαρακτηριστικά των δεδομένων Jquery Mobile παρέχουν τη δυνατότητα βελτίωσης και ρύθμισης εφαρμογών με απλή σήμανση HTML. Ο πλήρης κατάλογος όλων των δεδομένων χαρακτηριστικών, σε αλφαβητική σειρά με περιγραφές και παραδείγματα παρατίθενται παρακάτω.

data-ajax: Αυτό το χαρακτηριστικό μπορεί να συνδέεται με συνδέσμους, κουμπιά ή φόρμες. Όταν ρυθμιστεί σε ψευδής κατάσταση, θα αναγκάσει την επαναφόρτωση της σελίδας (παρακάμπτοντας Ajax και μεταβάσεις).

Παράδειγμα:

```
 <a href="multi-page.html" datarole="
button" data-ajax="false">
multi-page</a>
```

data-add-back-btn: Αυτό το χαρακτηριστικό μπορεί να συνδέεται σε μια σελίδα περιέκτη. Όταν οριστεί σε αληθής κατάσταση, ένα πίσω κουμπί θα εμφανιστεί αυτόματα στο τίτλο της ελίδας.

Παράδειγμα:


```
<div data-role="page" data-addback-
btn="true">
```

data-back-btn-text: Αυτό το χαρακτηριστικό συνδέεται στη σελίδα περιέκτη. Το προεπιλεγμένο κείμενο για το κουμπί πίσω είναι η λέξη "Back".

Παράδειγμα:

<div data-role="page" data-addback-btn="true" **data-back-btntext=" Previous"**>

data-collapsed: Μπορούμε να ρυθμίσουμε ένα πτυσσόμενο δοχείο για να συμπτυχθεί (datacollapsed = "true ") ή να επεκταθεί (στοιχεία καταρρεύσει = "false") με την προσθήκη αυτού του χαρακτηριστικού.

Παράδειγμα:

<h3>Wireless</h3>

data-corners: Αυτό το χαρακτηριστικό μπορεί να συνδεθεί με συνδέσεις ή κουμπιά. Όταν είναι σε ψευδής κατάσταση, το framework θα αφαιρέσει στρογγυλεμένες γωνίες από το κουμπί. Τα κουμπιά θα έχουν στρογγυλεμένες γωνίες από προεπιλογή.

Παράδειγμα:

Disagree

data-count-theme: Αυτό το χαρακτηριστικό καθορίζει ένα εναλλακτικό θέμα για το σήμα ή την καταμέτρηση φούσκα.

Παράδειγμα:

<ul data-role="listview" **datacount-theme="e"**>

data-direction: Αυτό το χαρακτηριστικό συνδέεται με δεσμούς, κουμπιά ή φόρμες. Όταν έχει ρυθμιστεί σε αντιστραφή θα εφαρμόσει μια αντίστροφη μετάβασης. Για παράδειγμα, μία προς τα εμπρός "Διαφάνεια" μετάβαση θα γλιστρήσει αριστερά.

Παράδειγμα:

<a href="" data-icon="home"
data-iconpos="notext" **datadirection="reverse"** class="uibtn-right">Home

data-divider-theme:Ορίζει το θέμα του διαχωριστή λίστας.

Παράδειγμα:

<li data-role="list-divider" **datadivider-theme=”a”**>

data-dom-cache: Αυτό το χαρακτηριστικό σας επιτρέπει να cache σελίδες εντός του DOM. Από προεπιλογή, αυτό το χαρακτηριστικό έχει οριστεί σε false και ως Συνεπώς, το πλαίσιο θα κρατήσει μόνο το "από" και "έως" σελίδες στην DOM αποθηκευμένο ενεργά. είναι συνιστάται να αφήσετε την τιμή αμετάβλητες, έτσι το DOM παραμένει ελαφρύ.

Παράδειγμα:

<div data-role="page" **datadom-**
cache="true">

data-filter: Αυτό το χαρακτηριστικό είναι συνδεδεμένο σε λίστες και προσθέτει μια γραμμή αναζήτησης πάνω από τη λίστα των αποτελέσματων όταν η τιμή έχει οριστεί σε true.

Παράδειγμα:

<ul data-role="listview" **datafilter="**
true">

data-filter-theme:Ορίζει το θέμα για το φίλτρο αναζήτησης.

Παράδειγμα:

<ul data-role="listview" datafilter="
true" **data-filtertheme="**
e">

data-fullscreen: Αυτό το χαρακτηριστικό είναι συνδεμένο με μια σελίδα περιέκτη. Το τμήμα του περιεχομένου θα εμφανίζεται σε λειτουργία πλήρους οθόνης, όταν οριστεί αληθής.

Παράδειγμα:

<div data-role="page" **datafullscreen="**
true">

data-icon: Αυτό το χαρακτηριστικό συνδέεται με συνδέσεις και κουμπιά.

Παράδειγμα:

<a href="../../" data-icon="home"
data-iconpos="notext" datadirection="

reverse" class="ui-btnright
jqm-home">Home
data-iconpos: Αυτό το χαρακτηριστικό μπορεί να συνδεθεί με συνδέσεις ή κουμπιά. Αυτό το χαρακτηριστικό θα τοποθετήσει το εικονίδιο. Μπορεί να ρυθμιστεί σε "Top", "κάτω", "αριστερά", "δεξιά" ή "NOTEXT θα αφαιρέσει το προεπιλεγμένο κείμενο του εικονιδίου για κουμπι εικονίδιο μόνο χωρίς κείμενο. Από προεπιλογή, τα εικονίδια έχουν αριστερή στοίχιση.

Παράδειγμα:

<a href="" *data-icon="home"*
data-iconpos="notext" datadirection="
reverse" class="ui-btnright">
Home

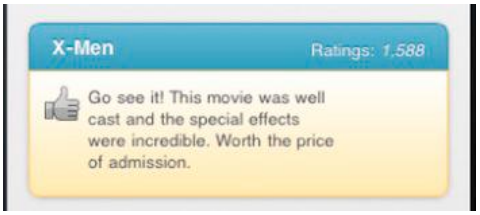
data-inline: Αυτό το χαρακτηριστικό συνδέεται με συνδέσεις ή κουμπιά. Από προεπιλογή, όλα τα κουμπιά στο περιεχόμενο του σώματος είναι διακοσμημένα ως blocklevel στοιχεία και γεμίζουν το πλάτος της οθόνης. Ένα μπλοκ inline τοποθετείται inline (δηλ., στην ίδια γραμμή όπως παρακείμενες περιεχόμενο), αλλά συμπεριφέρεται ως ένα μπλοκ.

Παράδειγμα:

<a href="#agree" datarole="
button" **datainline="
true"**>

data-inset="true": Αυτό το χαρακτηριστικό συνδέεται με λίστες. Όταν οριστεί σε true κατάσταση, θα θεματίσει τα αντικείμενα της λίστας, ώστε να μην εμφανίζονται απο άκρη σε άκρη και αντ 'αυτού θα εμφανιστούν με στρογγυλεμένες γωνίες.

Παράδειγμα:



<ul data-role="listview" **datainset="
true"**>

data-native-menu: Κείμενα επιλογής θα ενεργοποιήσουν τον picker για το λειτουργικό σύστημα από προεπιλογή. Για να καταστεί το μενού επιλογής σε μία προσαρμοσμένη προβολή HTML / CSS, θέτουμε αυτή τη τιμή σε false.

Παράδειγμα:

data-native-menu="false"
<select name="genre" **datanative-**
menu="false">

data-position: Αυτό το χαρακτηριστικό συνδέεται με κεφαλίδες ή υποσέλιδα. Όταν έχει ρυθμιστεί να καθορισθεί θα τοποθετήστε την κεφαλίδα και το υποσέλιδο κατά τη άνω και κάτω μέρος της σελίδας.

Παράδειγμα:

<div data-role="header" **dataposition="**
fixed">

<div data-role="footer" **dataposition="**
fixed">

data-rel="back": Αυτό το χαρακτηριστικό συνδέεται με συνδέσεις ή κουμπιά. Όταν έχει ρυθμιστεί στο "πίσω", ο σύνδεσμος θα μιμιθεί το πίσω κουμπί, πηγαίνοντας πίσω μία καταχώρηση στο ιστορικό και αγνοώντας τους προεπιλεγμένους href συνδέσμους.

Παράδειγμα:

<a href="home.html" datarole="
button" **datarel="**
back">Disagree

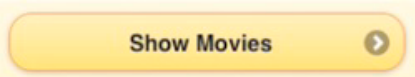
data-rel="dialog": Αυτό το χαρακτηριστικό συνδέεται με συνδέσεις ή κουμπιά. Μπορούμε να ορίσουμε αυτήν τη τιμή σε διάλογο για να επιβεβαιώσουμε πως θέλουμε η σελίδα προορισμού να οριστεί ως ένα παράθυρο διαλόγου.

Παράδειγμα:


```
<a href="dialog.html" datarole="
button" data-rel="dialog"
data-transition="slideup">Show
Dialog</a>
```

data-role="button": Αυτό το χαρακτηριστικό συνδέεται με συνδέσμους.
Η ρύθμιση της τιμής σε κουμπί δομεί το σύνδεσμο ως κουμπί.

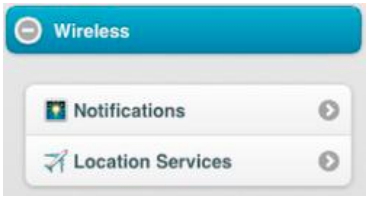
Παράδειγμα:




```
<a href="#movies" datarole="
button">Show Movies</a>
```

data-role="collapsible": Για να δημιουργήσουμε ένα μπλοκ περιεχομένου που μπορεί να αναπτυχθεί και να συμπτυχθεί, το τυλίγουμε ένα στοιχείο που περιέχει αυτό το χαρακτηριστικό.

Παράδειγμα:




```
<div data-role="collapsible">
<h3>Wireless</h3>
</div>
```

data-role="content": Αυτό το χαρακτηριστικό είναι συνδεδεμένο με το div στοιχείο που θα περιέχει το περιεχόμενο. Το στοιχείο αυτό είναι προαιρετικό.

Παράδειγμα:

<div **data-role="content"**>
data-role="dialog": Αυτό το χαρακτηριστικό συνδέεται στη σελίδα περιέκτη. Μπορούμε να εφαρμόσουμε αυτό το ως εναλλακτική λύση στην datarole="page". Ορίζοντας τη τιμή σε διάλογο θα εμφανιστεί η σελίδα ως ένα παράθυρο διαλόγου.

Παράδειγμα:

<div **data-role="dialog"**>

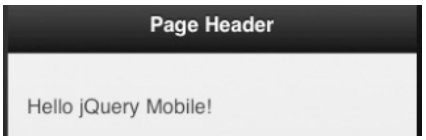
data-role="footer": Αυτή η ιδιότητα δημιουργεί το υποσέλιδο περιέκτη. Το υποσέλιδο είναι προαιρετικό.

Παράδειγμα:

<div **data-role="footer"**>

data-role="header": Αυτό το χαρακτηριστικό δημιουργεί την επικεφαλίδα περιέκτη. Η επικεφαλίδα είναι προαιρετική.

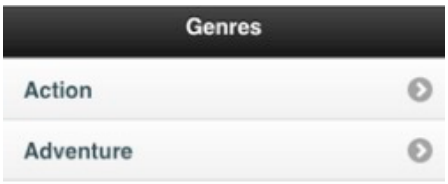
Παράδειγμα:



<div**data-role="header"**>

data-role="listview": Το χαρακτηριστικό αυτό χρησιμοποιείται για να δημιουργήσετε τη λίστα απόψεων.

Παράδειγμα:



<ul **data-role="listview"**>
Action

data-role="navbar":Το χαρακτηριστικό αυτό δημιουργεί μια πλοήγηση ή καρτέλα μπαρ. Μια μπάρα πλοήγησης μπορεί να επισυνάπτεται σε κεφαλίδες ή υποσέλιδα.

Παράδειγμα:

<div **data-role="navbar"**>
...

data-role="page":Αυτό το χαρακτηριστικό καθορίζει το είδος «δοχείου» της σελίδας.

Παράδειγμα:

<!-- header -->
<!-- content -->
<!-- footer -->

data-role="slider":Αυτό το χαρακτηριστικό χρησιμοποιείται για τη δημιουργία ελέγχου μέσου διακόπτη.

Παράδειγμα:

<select name="slider" **datarole="**
slider">
<option
value="off">Off</option>
<option
value="on">On</option>
</select>

data-theme: Αυτό το χαρακτηριστικό μπορεί να προστεθεί σε όλα δοχεία και εξαρτήματα της σελίδας για τη δημιουργία θεματικών σχεδίων.

Παράδειγμα:

<div data-role="header" **datatheme="b"**>

data-title: Αυτό το χαρακτηριστικό συνδέεται στη σελίδα δοχείο και θέτει τον τίτλο για μια σελίδα.

Παράδειγμα:

<div data-role="page" **datatitle="Welcome"**>

data-transition: Αυτό το χαρακτηριστικό μπορεί να συνδεθεί με συνδέσεις, κουμπιά, και τις μορφές. η αξία αυτού του χαρακτηριστικού καθορίζει το CSS-based εφέ μετάβασης προς χρήση όταν μετάβαση μεταξύ των σελίδων. παραπέμπω στον Πίνακα 2-1 για τον πλήρη κατάλογο των διαθέσιμες μεταβάσεις.

Παράδειγμα:

Flip

ΚΕΦΑΛΑΙΟ 8:

Service Integration Strategies

Κατά τη δημιουργία εφαρμογών Web, υπάρχουν στρατηγικές ρωτογενούς πρόσβασης για τη φόρτωση δεδομένων. Εμείς θα εξετάσουμε την client-side τρατηγική για την πρόσβαση. Σε αυτό το κεφάλαιο, πρόκειται να δείξουμε παραδείγματα της ενσωμάτωσης Jquery Mobile με τη στρατηγική πρόσβασης server-side και θα εφαρμόσουμε μία χρήση περίπτωση που θα εκτελεί GET για τα δεδομένα και ένα άλλο που θα κάνει POST αντίστοιχα. Για λόγους σύγκρισης, θα είμαστε εκ νέου εφαρμογή client-side παράδειγμα την εγγραφή μας ως server-side λύση. Η σελίδα σήμανσης μας γίνεται πολύ πιο ευανάγνωστη όταν εκτελείται η λήψη δεδομένων με server-side model-view-controller (MVC) στρατηγική για την πρόσβαση. Τέλος, με τη δημοτικότητα του geolocation και απόψεις χάρτη, θα δούμε πώς να ενσωματωθεί το Jquery Mobile με την HTML5 Geolocation API και το Google Maps.

Server-side Integration with MVC

Σε αυτή την ενότητα, πρόκειται να εστιάσουμε την προσοχή μας σε στρατηγικές πρόσβασης server-side. Στο Web, μια πολύ κοινή στρατηγική ενσωματώνεται με ένα μοντέλο-view-controller (MVC) framework. Θα δούμε δύο παραδείγματα MVC που θα είναι πολύ παρόμοια σε στυλ με clientside μας παραδείγματα. Στο πρώτο παράδειγμα μας, θα μετατρέψουμε την client-side σε μια εφαρμογή server-side. Αυτό το παράδειγμα, θα παρέχει σύγκριση του τρόπου που περιπτώσεις χρήσης μπορούν να εφαρμοστούν σε Jquery Mobile με client-side σε σχέση με τις στρατηγικές πρόσβασης server-side. Στο τελευταίο παράδειγμα μας, θα δούμε πώς θα εφαρμοστεί ένα σενάριο χρήσης που θα λαμβάνει δεδομένα από το διακομιστή.

Server-side Form POST with MVC

Για λόγους σύγκρισης, θα είναι καλό να δούμε μια server-side εφαρμογή του προγράμματός μας περίπτωση χρήσης εγγραφής. Και πάλι, θα έχουμε μια φόρμα εγγραφής που επιτρέπει στους χρήστες να opt-in για να λάβουν έκπτωση ή δωρεάν εισιτήρια κινηματογράφων (βλ. Εικόνα 9-5).

Εικόνα 9–5. ***Registration form for server-side POST with MVC***

Ο κώδικας της σελίδας για τη σελίδα εγγραφής μας είναι πολύ παρόμοιος με αυτόν που φαίνεται στο client-side παράδειγμά μας εκτός από το ότι δεν πρόκειται να παρακάμψουμ τη διαδικασία υποβολής φόρμας Στο server-side παράδειγμα εγγραφής, όταν ο χρήστης κάνει κλικ στο κουμπί εγγραφής, θα αφήσουμε τη φόρμα να υποβάλλει την αίτησή της στο action (βλέπε Καταχώρηση 9-10).

Καταχώρηση 9–10. *Registration form for server-side integration*

(/webapp/ch9/register-server.html)

```
<div data-role="page" id="registrationPage" data-theme="d">
<div data-role="header">
<h1>Register</h1>
</div>
<div data-role="content">
<form id="register" action="/jqm-webapp/mvc/register" method="post">
<label for="email">Email:</label>
<input type="email" name="email" id="email" placeholder="Email"
required />
<input type="submit" value="Register" data-theme="b"/>
</form>
</div>
</div>
```

Μήπως κάτι ξεχωρίζει όταν συγκρίνουμε τη σελίδα εγγραφής από την πλευρά του πελάτη μας (μητρώο-client.html) σε σχέση με αυτό (μητρώο-server.html); Η πιο αξιοσημείωτη διαφορά είναι ότι η σελίδα δεν απαιτεί custom JavaScript. Ως αποτέλεσμα, η σελίδα σήμανσης μας είναι πολύ πιο «καθαρή».

Κατά την υποβολή της φόρμας, μια αίτηση POST θα σταλεί στο μονοπάτι που ορίζεται στο action (/ jqm-webapp/mvc/register). Αυτό το αίτημα θα γίνεται από την πλευρά του διακομιστή από ένα Spring ελεγκτή MVC12* που έχει αναπτυχθεί στον Tomcat. Στο αρχείο web.xml μας, έχουμε διαμορφώσει το servlet-mapping έτσι όλα "/ MVC / *" URL να δρομολογούνται μέσω αποστολέα Spring MVC servlet (βλέπε Καταχώρηση 9-11).

Καταχώρηση 9–11. *Spring MVC servlet-mapping configuration (/WEB-INF/web.xml)*

```
<servlet>
<servlet-name>jqm-webapp</servlet-name>
<servlet-class>
org.springframework.web.servlet.DispatcherServlet
</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>jqm-webapp</servlet-name>
<url-pattern>/mvc/*</url-pattern>
</servlet-mapping>
```

Από την άποψη της διεπαφής χρήστη, η ροή εργασίας είναι ταυτόσημη με το παράδειγμα εγγραφής στην πλευρά του πελάτη. Η φόρμα υποβάλλεται, επεξεργάζεται, και στη συνέχεια φαίνεται η σελίδα με το ευχαριστώ. Ο κωδικός ελέγχου που επεξεργάζεται και ανακατευθύνει την αίτηση στην ευχαριστήρια σελίδα εμφανίζεται Καταχώρηση στο 9-12.

Καταχώρηση 9–12. *Spring MVC registration controller (com.bmb.jqm.controller.RegisterController.java)*

```
@Controller()
public class RegisterController {
@RequestMapping(method = RequestMethod.POST)
public String enroll(@RequestParam("email") String email, HttpSession
session) {
// Save registration...
session.setAttribute("email", email);
return "redirect:/mvc/register/thanks";
}
@RequestMapping(method = RequestMethod.GET)
```

```
public String thanks() {
return "register-thanks";
}
}
```

Ας επανεξετάσουμε τις σημειώσεις Spring MVC όπως προχωράμε μέσω του ελεγκτή :

 Ο σχολιασμός @ Controller καθορίζει τη κλάση ως ελεγκτή που μπορεί να χειριστεί τις αιτήσεις. Ο Spring MVC διαμορφώθηκε με μονοπάτι προς χαρτογράφηση μέσω ονόματος κλάσης. Για παράδειγμα , όλα τα "/εγγραφή/" αιτήσεις θα πρέπει να παραδοθούν στο RegisterController . Αυτή η ρύθμιση είχε οριστεί στο Spring MVC αποστολέα servlet (βλέπε Καταχώρηση 9-13).
 Ο σχολιασμός @ RequestMapping καθορίζει τις μεθόδους που θα χειριστούν τα POST και GET αιτήματα . Όταν η φόρμα έχει υποβληθεί , ένα POST αίτημα θα αποσταλεί με τη μέθοδο εγγραφής . Η μέθοδος «ευχαριστώ» θα χειριστεί όλες τις αιτήσεις GET.
 Ο σχολιασμός @ RequestParam θα δεσμεύσει τη διεύθυνση ηλεκτρονικού ταχυδρομείου που είχε υποβληθεί στη φόρμα, στη παράμετρο εισόδου του ταχυδρομείου. Όταν η μέθοδος εγγραφής καλείται , θα σώσει την εγγραφή , θα βάλει τη διεύθυνση ηλεκτρονικού ταχυδρομείου σε συνεδρία , και θα ανακατευθύνει στη σελίδα με ευχαριστίες (/ jsp / registerthanks .jsp).

Καταχώρηση 9–13. *Spring MVC path to controller mapping configuration (/WEB-INF/jqm-webapp-servlet.xml)*

```
<!-- Enable controller mapping by convention. For example: /foo/* will map to
FooController() -->
<bean
class="org.springframework.web.servlet.mvc.support.ControllerClassNameHandlerMapping" />
```

Η εμφάνηση της σελίδας ευχαριστιών θα δείχνει πανομοιότυπη με το παράδειγμα client-side μας (βλ. Εικόνα 9–6).

Εικόνα 9–6. ***Thank you page after server-side POST with MVC***

Η μόνη διαφορά είναι στο πώς οι σελίδες δημιουργούνται. Αυτή η σελίδα δημιουργείται με την server-side ως JSP και τη διεύθυνση ηλεκτρονικού ταχυδρομείου συνδέεται με τη σύνταξη έκφρασης JSTL (βλέπε Καταχώρηση 9-14). Χωρίς JavaScript, για να δημιουργήσουμε δυναμικά την σελίδα αυτή, η σήμανση είναι πιο ευανάγνωστη σε σύγκριση με το δυναμικά ευχαριστώ που είδαμε προηγουμένως στο client-side παράδειγμα.

Καταχώρηση 9–14. *Thank you page after server-side registration (/jsp/register-thanks.jsp)*

```
<div data-role="page" id="thanksPage" data-theme="d">
<div data-role="header">
<h1>Thank You</h1>
</div>
<div data-role="content" class="thanks">
<p>Thanks for registering. One FREE movie pass was just sent to:
<span class="email">${email}</span></p>
<img src="images/free-movie.jpg">
</div>
</div>
```

Ένα σημαντικό θέμα που πρέπει να γνωρίζεουμε , μετά την υποβολή των μορφών είναι ότι το Jquery Mobile διαχειρίζεται τη διεύθυνση URL που εμφανίζεται στη γραμμή διευθύνσεων του περιηγητή. Για παράδειγμα , αφού ο διακομιστής ανακατευθύνει σε "/MVC/εγγραφή/thanks" το URL του προγράμματος περιήγησης εξακολουθεί να δείχνει το μονοπάτι του action ("/ jqm-webapp/mvc/register"). Εάν δεν έχει εφαρμοστεί μια αίτηση GET χειρισμού για αυτό το μονοπάτι , ένα " φρεσκάρισμα" στην ευχαριστήρια σελίδα, θα οδηγήσει σε ένα 404, χωρίς σφάλμα . Έχουμε δύο επιλογές για το χειρισμό :

 Η απλούστερη λύση είναι να εφαρμόσουμε ένα GET αίτημα χειριστή για το μονοπάτι δράσης. Η RegisterController#thanks μέθοδος χειρίζεται αιτήσεις GET και θα ανανεώσει απλά τη σελίδα ευχαριστιών και θα επανεμφανίσει τη διεύθυνση ηλεκτρονικού ταχυδρομείου, που είναι αποθηκευμένη σε συνεδρία (βλ. Καταχώρηση 9-12). Επίσης, κατά την υποβολή εντύπων στο Web, συνιστάται πρώτα το POST και στη συνέχεια να ανακατευθύνει ώστε να αποφευχθούν τυχόν θέματα διπλών καταχωρήσεων.

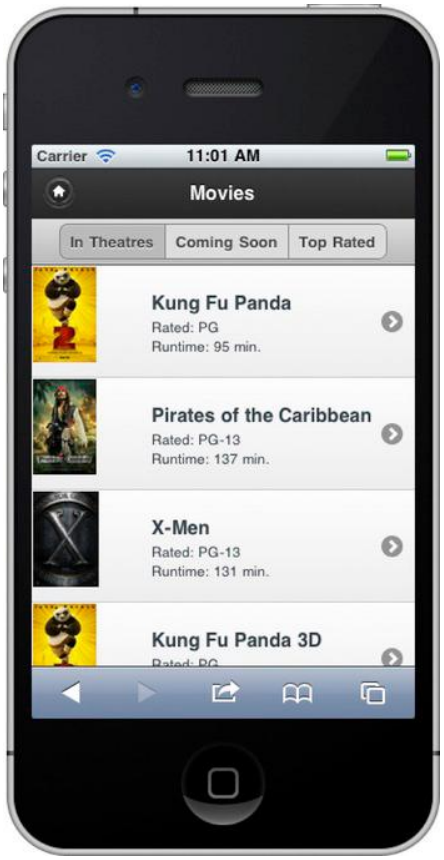
 Εναλλακτικά , μπορούμε να ρυθμίσουμε την ιδιότητα data-url στο περιέκτη. Η τιμή του χαρακτηριστικού data-url θα εμφανίζεται στην γραμμή διευθύνσεων του προγράμματος περιήγησης . Αυτό δίνει επίσης στους προγραμματιστές μεγαλύτερη ευελιξία κατά την κατασκευή σημασιολογικών διαδρομών :

data-url = " / manually / set / url / path / "

Αυτή η στρατηγική μπορεί επίσης να χρησιμοποιηθεί για να κρύψει τα ονόματα των αρχείων . Για παράδειγμα , αν έχετε μεταβούμε στο "/Μου/ταινίες/ index.html", μπορούμε να ενημερώσουμε τα data-url χαρακτηριστικά της σελίδας στο "/Μου/ταινίες/ , το οποίο θα κρύψει το τμήμα index.html από το να εμφανιστεί .

Server-side Data Access with MVC

Στο προηγούμενο παράδειγμα μας, είδαμε πως δημοσιεύουμε δεδομένα φόρμας στο διακομιστή. Σε αυτό το παράδειγμα, θα χρησιμοποιήσουμε μια αίτηση GET για να φέρουμε δεδομένα από το διακομιστή. Αυτό το παράδειγμα θα ανακτήσει μια λίστα ταινιών από το διακομιστή και θα εμφανίσει τα αποτελέσματα σε μια σελίδα Jquery Mobile JSP (βλέπε Εικόνα 9-7).



Εικόνα 9–7. ***Movies fetched from server-side MVC access***

Από την πλευρά του διακομιστή, έχουμε μια εγκατάσταση ελεγκτή Spring MVC για να χειριστεί GET αιτήματα στην ακόλουθη href:

Movies

Όταν κάνουμε κλικ στο κουμπί, μια αίτηση GET θα ενεργοποιηθεί και θα σταλθεί στη MoviesController. Η MoviesController θα ανακτήσει τα δεδομένα μας και θα τα διαβιβάσει στην JSP σελίδα (βλέπε Καταχώρηση 9-15).

Καταχώρηση 9-15. MVC ελεγκτή να GET δεδομένα ταινίας
(com.bmb.jqm.MoviesController.java)

```
@Controller()
public class MoviesController {
@RequestMapping(method = RequestMethod.GET)
public String getMovies(ModelMap model) {
model.addAttribute("movies", getMovieData());
return "movies";
}
}
```

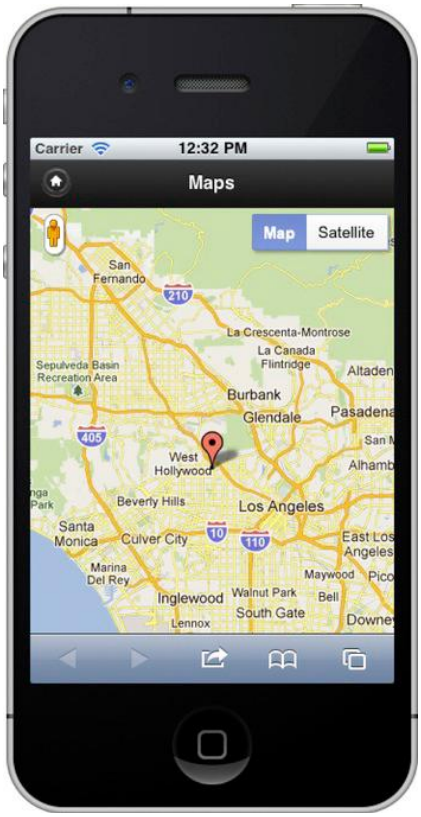
Η απάντηση θα διαβιβαστεί στη σελίδα ταινίες όπου η JSP θα επαναλάβει τη λίστα ταινίες, εμφανίζοντας μια ξεχωριστή λίστα για κάθε αποτέλεσμα (βλ. Καταχώρηση 9-16).

Καταχώρηση 9–16. *JSP to display movie data (/jsp/movies.jsp)*

```
<div data-role="content">
<ul data-role="listview">
<c:forEach var="movie" items="${movies}">
<li>
<a href="#">
<img src="../images/${movie.image}" />
<h3>${movie.title}</h3>
<p>Rated: ${movie.rating}</p>
<p>Runtime: ${movie.runtime} min.</p>
</a>
</li>
</c:forEach>
</ul>
</div>
```

Google Maps

Σε μια πρόσφατη έρευνα του Mobile Web, σχεδόν το 75% των προγραμματιστών Web χρησιμοποιούν το Geolocation καθιστώντας το το πιο δημοφιλές API*13 HTML5. Κατά τη δημιουργία εφαρμογών που έχουν γνώση της γεωγραφικής θέσης, το είναι σύνηθες να έχουν ένα απόσπασμα του χάρτη που να εμφανίζει σημεία ενδιαφέροντος ή διευθύνσεις. Στο διαδίκτυο, το Geolocation*14 σε συνδυασμό με το Google Maps*15 παρέχουν ένα πολύ χρήσιμο API για τη δημιουργία λειτουργικότητα χάρτη. Σε αυτή την ενότητα, πρόκειται να δούμε πόσο καλά το Jquery Mobile ενωποιείται με το geolocation και το Google Maps. Για να ξεκινήσουμε, θα δημιουργήσουμε ένα παράδειγμα που δείχνει τη τρέχουσα θέση μας σε έναν χάρτη (βλ. Εικόνα 9-8).



Εικόνα 9–8. ***Google Maps integration with jQuery Mobile***

Η σήμανση μέσα στη σελίδα μας είναι ελάχιστη, επειδή χρειαζόμαστε μόνο για να δημιουργήσετε το περιεχόμενο δοχείο για χάρτη μας (βλέπε Λίστα 9-17).

Καταχώρηση 9–17. *jQuery Mobile page markup for Google Maps integration (ch9/maps.html)*

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Google Maps</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="jquery.mobile.css" />
<style>
#map-page, #map-canvas { width: 100%; height: 100%; padding: 0; }
</style>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="maps.js"></script>
<script type="text/javascript" src="jquery.mobile.js"></script>
<script src="http://maps.google.com/maps/api/jssensor=false"></script>
</head>
<body>
<div data-role="page" id="map-page">
```

```
<div data-role="header">
<h1>Maps</h1>
</div>
<div data-role="content" id="map-canvas">
<!-- map loads here... -->
</div>
</div>
</body>
</html>
```

Κατά την ενσωμάτωση με το Google Maps, αρκετές προσθήκες είναι απαραίτητες:

1. Κατ 'αρχάς, θα πρέπει να ρυθμίσουμε το στύλ της σελίδας μας και του χάρτη έτσι ώστε να εμφανίζονται σε πλήρη οθόνη:

#map-page, #map-canvas { width: 100%; height: 100%; padding: 0; }

2. Στη συνέχεια, εισάγουμε JavaScript κώδικα για να μας βοηθήσει να προσδιορίσουμε τη θέση του χρήστη και να επιστήσει την οπτική άποψή μας στο χάρτη. Θα εξετάσουμε τις λεπτομέρειες αυτού του αρχείου σύντομα:

<script type="text/javascript" src="maps.js"></script> >

3. Στη συνέχεια, εισάγουμε το Google Maps API:

<script src="http://maps.google.com/maps/api/jssensor=false">

4. Τέλος, έχουμε εντοπίσει το χάρτη μας. Ο χάρτης μας θα συνταχθεί μέσα σε αυτό το στοιχείο:
** **
Το JavaScript θα μας βοηθήσει να προσδιορίσουμε τη γεωγραφικη τοποθεσία του χρήστη και να επιστήσει την οπτική άποψή μας στο χάρτη (βλ. Καταχώρηση 9-18).

Καταχώρηση 9–18. *JavaScript for Google Maps integration (ch9/maps.js)*

```
$( "#map-page" ).live( "pageinit", function() {
// Default to Hollywood, CA when no geolocation support
var defaultLatLng = new google.maps.LatLng(34.0983425, -118.3267434);
if ( navigator.geolocation ) {
function success(pos) {
// Location found, show coordinates on map
drawMap(new google.maps.LatLng(
pos.coords.latitude, pos.coords.longitude));
}
```

```
function fail() {
drawMap(defaultLatLng); // Show default map
}
// Find users current position
navigator.geolocation.getCurrentPosition(success, fail,
{enableHighAccuracy:true, timeout: 6000, maximumAge: 500000});

} else {
drawMap(defaultLatLng); // No geolocation support
}
function drawMap(latlng) {
var myOptions = {
zoom: 10,
center: latlng,
mapTypeId: google.maps.MapTypeId.ROADMAP
};
var map = new google.maps.Map(
document.getElementById("map-canvas"), myOptions);
// Add an overlay to the map of current lat/lng
var marker = new google.maps.Marker({
position: latlng,
map: map,
title: "Greetings!"
});
}
});
```

Όταν η σελίδα χάρτης είναι σε ένα "έτοιμη" κατάσταση θα εκτελέσουμε τα παρακάτω βήματα για να σχεδιάσουμε την όψη του χάρτη:
1 . Κατ 'αρχάς , θα καθορίσουμε αν ο περιηγητής υποστηρίζει το geolocation :
```
if ( navigator.geolocation ) {
```

2 . Εάν το πρόγραμμα περιήγησης υποστηρίζει geolocation θα επιχειρήσουμε να ανακτήσουμε την τρέχουσα θέση των χρηστών:

```
navigator.geolocation.getCurrentPosition(success, fail,
{enableHighAccuracy:true, timeout: 6000, maximumAge: 500000});
```

Η getCurrentPosition API μπορεί να πάρει μέχρι και τρεις παραμέτρους. Η πρώτη παράμετρος είναι η επιστροφή κλήσης επιτυχία . Αυτή είναι η μόνη απαιτούμενη παράμετρος . Η επόμενη παράμετρος είναι η επιστροφή κλήσης σφάλματος και η τελευταία παράμετρος είναι οι ρυθμιζόμενες επιλογές μας. Έχουμε ρυθμιστεί την αναζήτηση geolocation μας να χρησιμοποιεί υψηλή ακρίβεια. Αυτό θα επιχειρήσει να χρησιμοποιήσει GPS για την τοποθέτηση εάν υποστηρίζεται. Μπορούμε επίσης να ρυθμίσουμε το χρονικό όριο σε 6 δευτερόλεπτα. Αν αποτύχγουμε να βρούμε τη θέση των χρηστών μετά από 6 δευτερόλεπτα, θα γίνει επίκληση λάθους επανάκλησης. Τέλος , έχουμε

ρυθμίσει ότι η αναζήτηση μπορεί να χρησιμοποιήσει μια προσωρινά αποθηκευμένη θέση, εάν είναι λιγότερο από 5 λεπτά παλιά στη μνήμη.

3 . Όταν βρεθεί μια επιτυχημένη θέση θα γίνει επίκληση της επιτυχούς επανάκλησης. Σε αυτή τη περίπτωση , θα σχεδιάσουμε το χάρτη μας χρησιμοποιώντας συντεταγμένες θέσης:

```
function success(pos) {
drawMap(new google.maps.LatLng(
pos.coords.latitude, pos.coords.longitude))
}
```

4. Τέλος, η μέθοδος drawMap θα συντάξει ένα Google Map με ένα εικονίδιο επικάλυψης που θα εμφανίζεται στο κέντρο του χάρτη στη θέση που είχε προσδιοριστεί η θέση μας. Ωστόσο, εάν το Geolocation δεν υποστηρίζεται ή αν δεν εδραιώθηκε θέση, Hollywood, CA θα είναι εμφανίζονταιι ως η προεπιλεγμένη θέση (βλέπε Καταχώρηση 9-18).

ΚΕΦΑΛΑΙΟ 9:

ΓΝΩΡΙΖΟΝΤΑΣ ΤΟ PHONEGAP

Native εφαρμογές φαίνεται να έχουν δύο σημαντικά πλεονεκτήματα σε σύγκριση με Mobile Web εφαρμογές. Πρώτον , οι native εφαρμογές μπορούν να διανεμηθούν σε ένα κατάστημα εφαρμογών . Τα πιο αξιοσημείωτα καταστήματα εφαρμογών περιλαμβάνουν το App Store της Apple , το Android Market , HP App Catalog , BlackBerry AppWorld και Windows Marketplace . Αυτά τα καταστήματα απλοποιούν την εμπειρία του χρήστη , στις περιπτώσεις όπου οι καταναλωτές πρέπει να αναζητήσουν, να αγοράσουν, να εγκαταστασήσουν, ή και να αξιολογήσουν τις native εφαρμογές. Ένα άλλο πλεονέκτημα των native εφαρμογών είναι η ικανότητά τους να αλληλεπιδρούν με APIs συσκευών. Έχουν τη δυνατότητα να επικοινωνούν με τα περισσότερα APIs συσκευών, συμπεριλαμβανομένων επαφών, ημερολόγιο , φωτογραφική μηχανή.

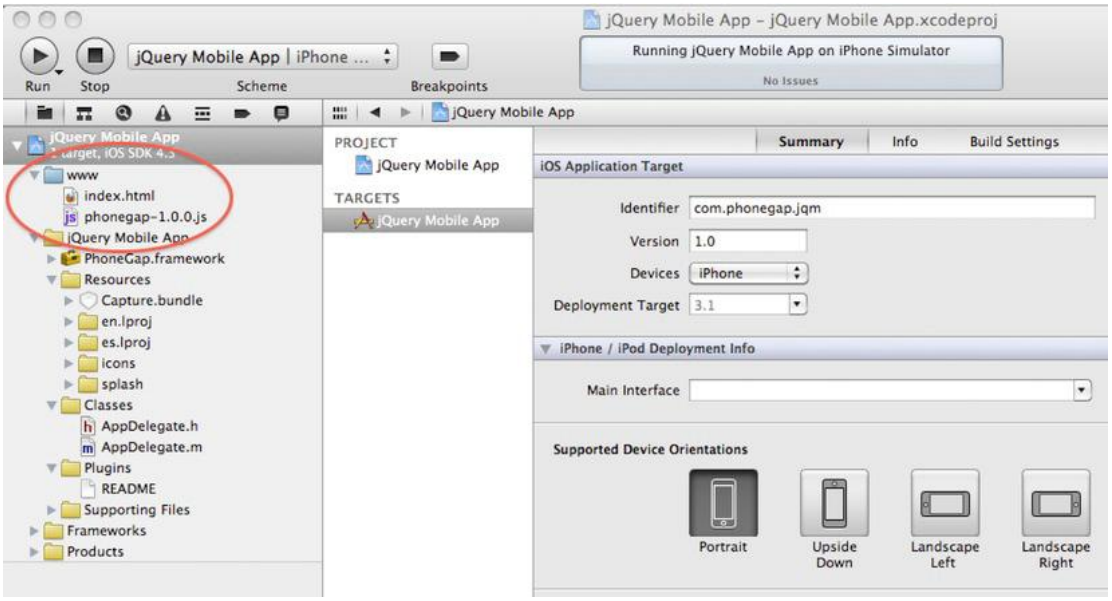
Σε αυτό το κεφάλαιο , θα συζητήσουμε πώς μπορούμε να σπάσουμε αυτά τα εμπόδια Mobile Web. Ειδικότερα, θα παρουσιάσουμε το PhoneGap και θα δείξουμε πώς το PhoneGap μπορεί να βοηθήσει στη γεφύρωση αυτών των κενών για τις Jquery Mobile εφαρμογές μας . Ως παράδειγμα , θα πάρουμε ένα υπάρχον Jquery Mobile app , θα το «τυλίξουμε» με PhoneGap και θα δούμε την εφαρμογή μας σε iOS και Android πλατφόρμες .Θα δούμε επίσης πώς μπορούμε να διανείμουμε jQuery Mobile εφαρμογές μας σε ένα κατάστημα app χωρίς PhoneGap . Το Open Market App είναι ένα κατάστημα εφαρμογών για HTML5 εφαρμογές κινητής τηλεφωνίας που μπορεί να είναι μια εναλλακτική λύση για εκείνους που βρίσκουν τη διαδικασία διανομής δυσκίνητη και αργή.

ΤΙ ΕΙΝΑΙ ΤΟ PHONEGAP

Το PhoneGap είναι ένα πλαίσιο ανάπτυξης λογισμικού ανοικτού κώδικα που μας επιτρέπει να οικοδομήσουμε cross-platform native εφαρμογές με τεχνολογίες web, όπως Jquery Mobile. Για παράδειγμα, μπορούμε να πάρουμε ένα υπάρχον web app Jquery Mobile, να το τυλίξουμε το με το PhoneGap και να το διανείμουμε σε όλες τις πλατφόρμες που υποστηρίζει το PhoneGap. Επί του παρόντος, το PhoneGap υποστηρίζει τις μητρική iOS, Android, BlackBerry, webOS, και Symbian πλατφόρμες. Εκθέτει επίσης ένα API που επιτρέπει στις Mobile εφαρμογές Web να αλληλεπιδρούν με APIs συσκευών, συμπεριλαμβανομένων το σύστημα αρχείων, ειδοποιήσεις, και τη φωτογραφική μηχανή για να αναφέρουμε μερικές. Για την πλήρη λίστα, ανατρέξτε στην Υποστηριζόμενες λειτουργίες PhoneGap κατά platform.2 Το PhoneGap API επιτρέπει να επεκτείνουμε τις Jquery Mobile εφαρμογές μας με τρόπους που προηγουμένως ήταν δυνατό μόνο με τη βοήθεια του SDK.

ΕΚΤΕΛΩΝΤΑΣ jQuery Mobile ΩΣ ΜΙΑ iOS ΕΦΑΡΜΟΓΗ

Σε αυτή την ενότητα, πρόκειται να τυλίξουμε μία Jquery Mobile εφαρμογή με PhoneGap και να το εκτελέσουμε στη πλατφόρμα iOS. Για να συσταθεί το PhoneGap για την πλατφόρμα iOS, μπορούμε να αναφερθούμε στο “Getting Started Guide with iOS.”3 Το PhoneGap έχει βήμα-βήμα οδηγίες για την εγκατάσταση του σε κάθε πλατφόρμα και οι οδηγίες τους είναι πολύ λεπτομερείς σε συνδιασμό με τα screenshots για βοήθεια. Εγκατάσταση του Xcode, το IDE για iOS ανάπτυξη4 αποτελεί προϋπόθεση για την ανάπτυξη στην πλατφόρμα iOS. Ενώ κάθε πλατφόρμα έχει ειδικές IDE οδηγίες εγκατάστασης, τη γενική διαδικασία εγκατάστασης του PhoneGap, για τη δημιουργία του έργου, και την ανάπτυξη είναι βήματα για την όλες τις πλατφόρμες. Αφού η πλατφόρμα iOS μας έχει συσταθεί, θα πρέπει να έχουμε ένα νέο Xcode project που θα μοιάζει με το σχήμα 10-1.



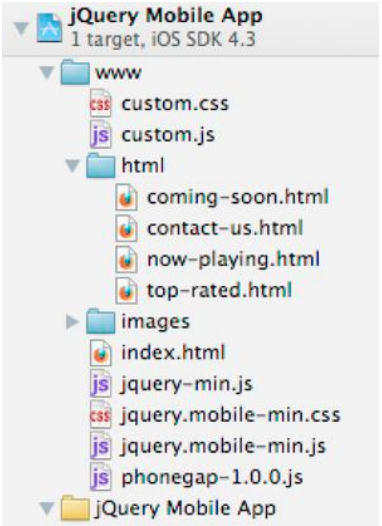
Σχήμα 10–1. ***Initial Xcode project with PhoneGap support***

Ο "www" κατάλογος που φαίνεται στο Σχήμα 10-1 είναι ο root κατάλογος της εφαρμογής. Μέσα σε αυτό το καταλόγο είναι η βιβλιοθήκη PhoneGap JavaScript και μια προεπιλεγμένη σελίδα (index.html). Η index.html σελίδα θα εμφανίζεται ως αρχική σελίδα προορισμού όταν τρέχουμε την εφαρμογή. Στο Xcode, για να δομήσουμε και να τρέξουμε την εφαρμογή, κάνουμε κλικ στο κουμπί "Run" που εμφανίζεται στην επάνω αριστερή γωνία. Αφού κάνουμε κλικ στο "Run", η εφαρμογή θα συνταχθεί, ο προσομοιωτής iOS θα ξεκινήσει, και η αρχική σελίδα θα εμφανιστεί (βλ. Εικόνα 10-2).

Εικόνα 10–2. ***Initial screen when running PhoneGap's default app in Xcode***

Με ένα PhoneGap project στο Xcode μπορούμε να εισαγάγουμε μια υπάρχουσα Jquery Mobile εφαρμογή στο έργο μας. Τα βήματα για την εισαγωγή μιας Jquery Mobile εφαρμογής στο Xcode και την ανάπτυξη σε iOS περιβάλλον είναι οι παρακάτω:

1. Κατ 'αρχάς, πρέπει να εισάγουμε ένα υπάρχον Jquery Mobile project στο "www" κατάλογο του Xcode. Αν εισάγουμε αρχεία Jquery Mobile και τα μεταφέρουμε στο "www" κατάλογο, η δομή του project θα εμφανίζεται όπως στην παρακάτω εικόνα:

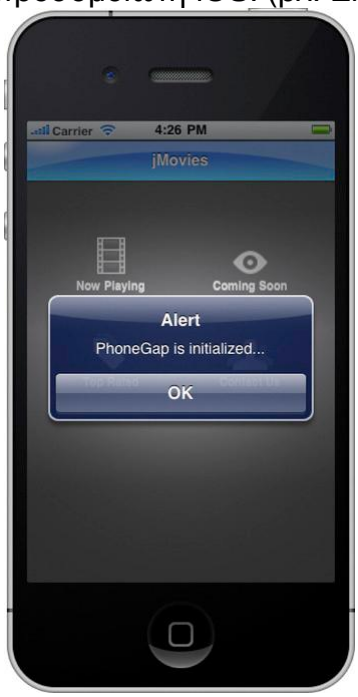


2. Μετά την εισαγωγή του Jquery κώδικα στο PhoneGap θα πρέπει να εισάγουμε JavaScript βιβλιοθήκη PhoneGap ως πόρο υψηλού επιπέδου:

```
<head>
<meta charset="utf-8">
<title>jMovies</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="jquery.mobile-min.css" />
<link rel="stylesheet" type="text/css" href="custom.css" />
<script type="text/javascript" src="phonegap-1.0.0.js"></script>
<script type="text/javascript" src="jquery-min.js"></script>
<script type="text/javascript" src="custom.js"></script>
<script type="text/javascript" src="jquery.mobile-min.js"></script>
</head>
```

Η βιβλιοθήκη PhoneGap είναι ένα API που παρέχει πρόσβαση σε πολλά συγκεκριμένα χαρακτηριστικά (κάμερα, media, αποθήκη, κ.λ.π.). Η εισαγωγή του PhoneGaplibrary είναι απαραίτητη μόνο όταν η εφαρμογή μας πρέπει να αλληλεπιδρά με native δυνατότητες PhoneGap.

3. Το τελευταίο βήμα είναι να τρέξουμε και να δοκιμάσουμε την εφαρμογή μας. Στο Xcode, κάνουμε κλικ στο κουμπί "Run". Αυτό θα καταρτίσει την εφαρμογή και να ξεκινήσει εντός του προσομοιωτή iOS. (βλ. Εικόνα 10-3).



Εικόνα 10–3. *jQuery Mobile running as a native iOS app*

Για να βοηθήσουμε στην επικύρωση ότι έχει εγκατασταθεί σωστά η βιβλιοθήκη PhoneGap προσθέσαμε έναν ακροατή για το συμβάν όπου η συσκευήθα είναι έτοιμη. (βλέπε Καταχώρηση 10-1).

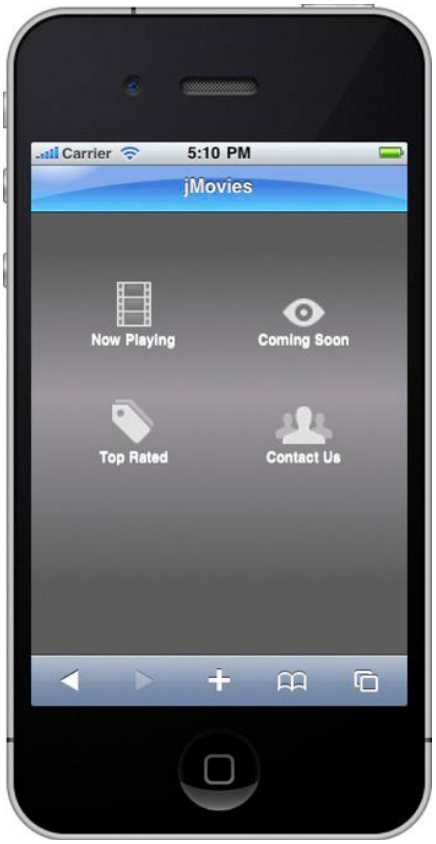
Καταχώρηση 10–1. *PhoneGap is ready*

```
$(document).bind("deviceready", function(){
navigator.notification.alert("PhoneGap is initialized...");
```

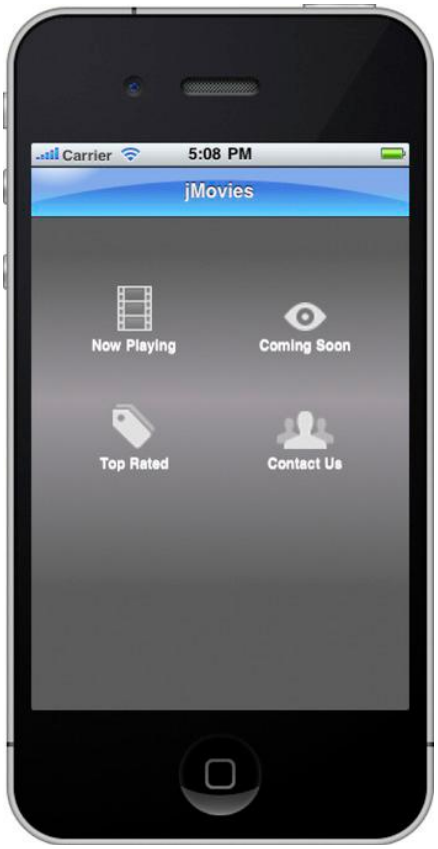
});

Όπως φαίνεται στην Εικόνα 10-3, όταν το PhoneGap είναι σε κατάσταση ετοιμότητας προβάλλουμε μια άποψη συναγερμού υποδεικνύοντας ότι έχει προετοιμαστεί. Η ανακοίνωση στη Καταχώρηση 10-1 είναι ένα παράδειγμα για το πώς μπορούμε να αλληλεπιδράσουμε προγραμματιστικά με τη λειτουργικότητα του PhoneGap.

Παρατηρούμε οτι δεν υπάρχουν διαφορές κατά τη σύγκριση της Jquery Mobile εφαρμογή μας εντός ενός Safari περιηγητή (βλ. Εικόνα 10-4) σε σχέση με την εγγενή εφαρμογή που τρέχει σε iOS (βλ. Εικόνα 10-5)



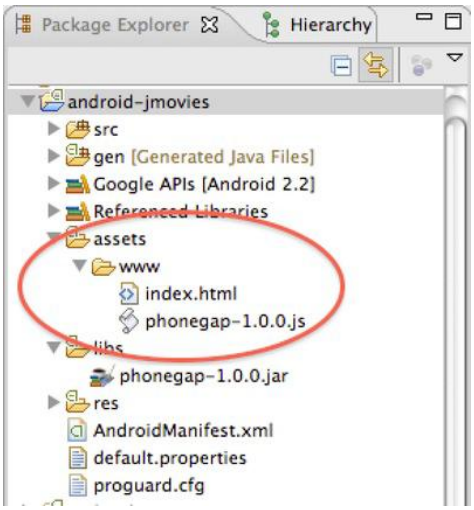
Εικόνα 10–4.
Running within a browser



Εικόνα 10–5. ***jQuery***
Running as a native iOS app

Εκτελώντας jQuery Mobile ώς μία Android εφαρμογή

Σε αυτή την ενότητα, πρόκειται να τυλίξουμε μία Jquery Mobile εφαρμογή με PhoneGap και να το εκτελέσουμε σε πλατφόρμα Android. Για να συσταθεί το PhoneGap στη πλατφόρμα Android, θα εγκαταστήσουμε το Eclipse, η IDE για το Android ανάπτυξη. Αφού η πλατφόρμα Android έχει δημιουργηθεί, θα έχει ένα νέο Eclipse project που μοιάζει με την Εικόνα 10-11.



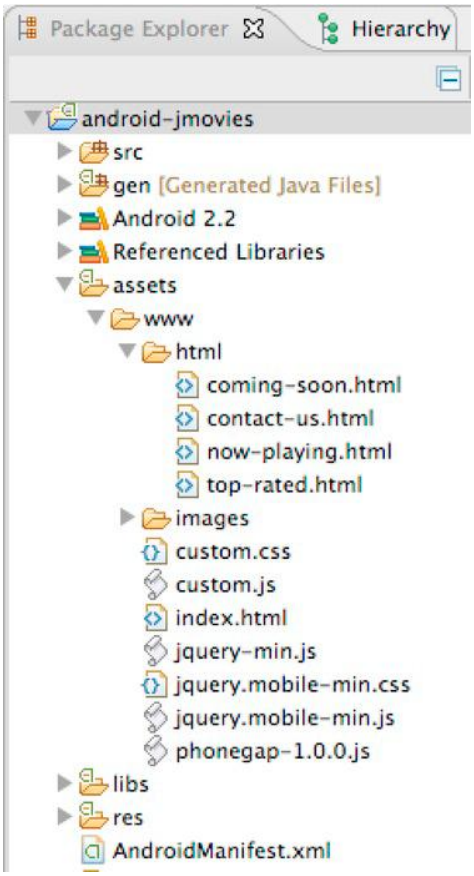
Εικόνα 10–11. *Initial Eclipse project with PhoneGap support*

Ο "www" κατάλογος που φαίνεται στην Εικόνα 10-11 είναι ο root κατάλογος της εφαρμογής. Μέσα σε αυτό το καταλόγο είναι η βιβλιοθήκη PhoneGap JavaScript και μια προεπιλεγμένη σελίδα (index.html). Η index.html σελίδα θα εμφανίζεται ως αρχική σελίδα προορισμού όταν τρέχουμε την εφαρμογή. Στο Eclipse, για να οικοδομήσουμε και να τρέξουμε την εφαρμογή, κάνουμε κλικ στο μενού Run, επιλέγουμε Run As, και επιλέγουμε Android Εφαρμογών. Αφού εκτελεστεί η εφαρμογή, ο προσομοιωτής Android θα ξεκινήσει και η αρχική σελίδα θα εμφανιστεί (βλ. Εικόνα 10-12).

Εικόνα 10–12. *Initial screen when running PhoneGap's default app in Eclipse*

Με το PhoneGap project ενσωματομένο στο Eclipse μπορούμε τώρα να εισάγουμε μία Jquery Mobile εφαρμογή. Τα βήματα είναι τα παρακάτω:

1. Κατ 'αρχάς, θα πρέπει να εισάγουμε ένα project Jquery Mobile στο "www" root κατάλογο του Eclipse. Για παράδειγμα, αν εισάγουμε αρχεία Jquery Mobile και να τα μεταφέρουμε στο "www" κατάλογο, η δομή του Eclipse project θα πρέπει να εμφανίζεται όπως φαίνεται στην Εικόνα 10-13:

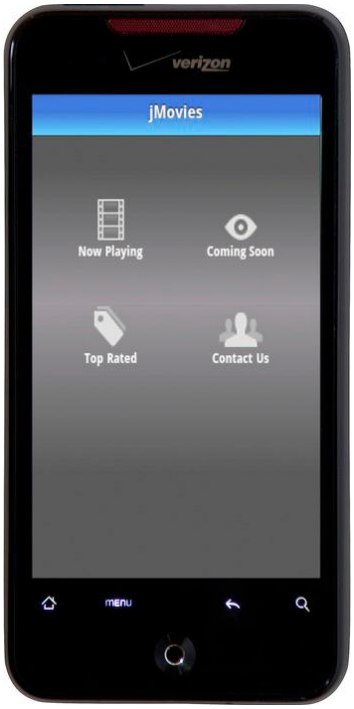


Εικόνα 10–13. ***an Eclipse project***

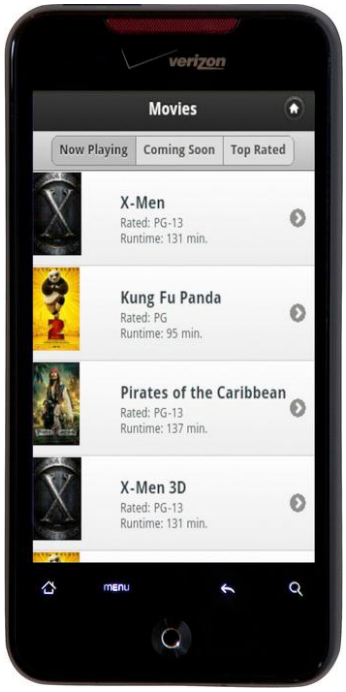
2. Μετά την εισαγωγή του Jquery Mobile κώδικα στο Eclipse project θα πρέπει να εισάγουμε JavaScript βιβλιοθήκη PhoneGap ως πόρο υψηλού επιπέδου:

```
<head>
<meta charset="utf-8">
<title>jMovies</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="jquery.mobile-min.css" />
<link rel="stylesheet" type="text/css" href="custom.css" />
<script type="text/javascript" src="phonegap-1.0.0.js"></script>
<script type="text/javascript" src="jquery-min.js"></script>
<script type="text/javascript" src="custom.js"></script>
<script type="text/javascript" src="jquery.mobile-min.js"></script>
</head>
```

3.Το τελευταίο βήμα είναι να τρέξουμε και να δοκιμάσουμε την εφαρμογή μας. Στο Eclipse, κάνουμε κλικ στο μενού Run, επιλέγουμε Run As, και επιλέγουμε το Android Εφαρμογών. Αυτό θα καταρτίσει την εφαρμογή και θα ξεκινήσει εντός του προσομοιωτή Android. (βλ. Εικόνα 10-14).



Εικόνα 10–14. ***Initial screen when running PhoneGap from Eclipse***

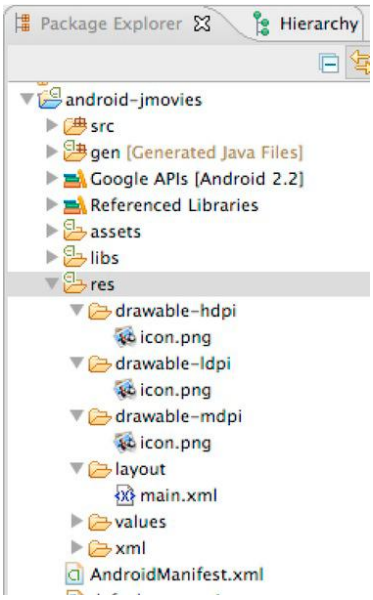


Εικόνα 10–15. ***No back button necessary on Android***

Τώρα που είμαστε σε θέση να αναπτύξουμε τη Jquery Mobile εφαρμογή μας στην Android πλατφόρμα θα προσαρμόσουμε το εικονίδιο της εφαρμογής (βλ. Εικόνα 10-16).

Εικόνα 10–16. ***PhoneGap's default app icon in Android***

Τα εικονίδια των εφαρμογών Android αποθηκεύονται στο / res/drawable- * καταλόγους με διαθέσιμες εικόνες υψηλής, μεσαίας και χαμηλής πυκνότητας (βλ. Εικόνα 10-17).



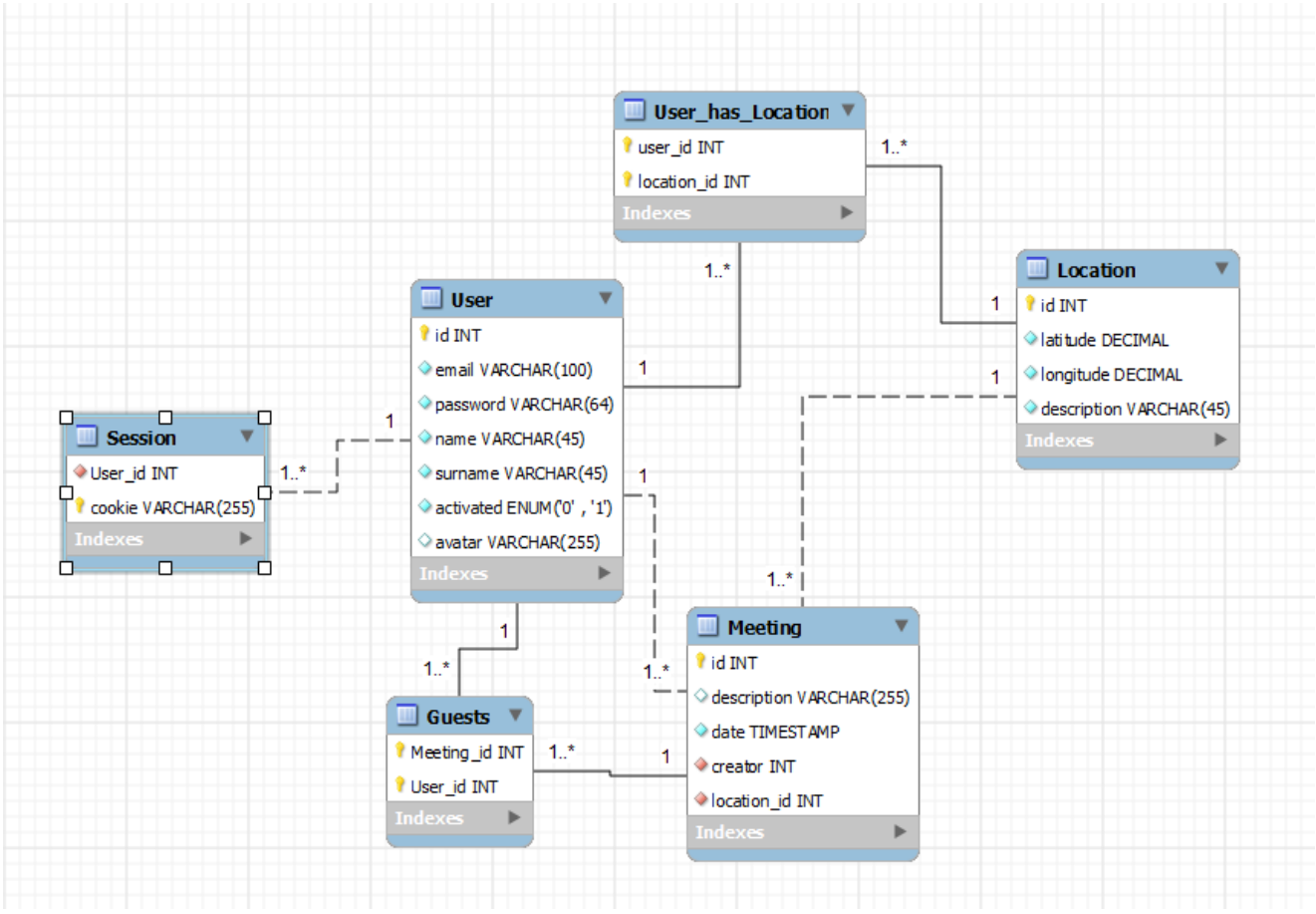
Αυτό ολοκληρώνει την όλη διαδικασία από την εγκατάσταση του PhoneGap μέχρι και την εκτέλεση της Jquery Mobile εφαρμογής για την Android πλατφόρμα.

ΚΕΦΑΛΑΙΟ 10:

Sync Meetings

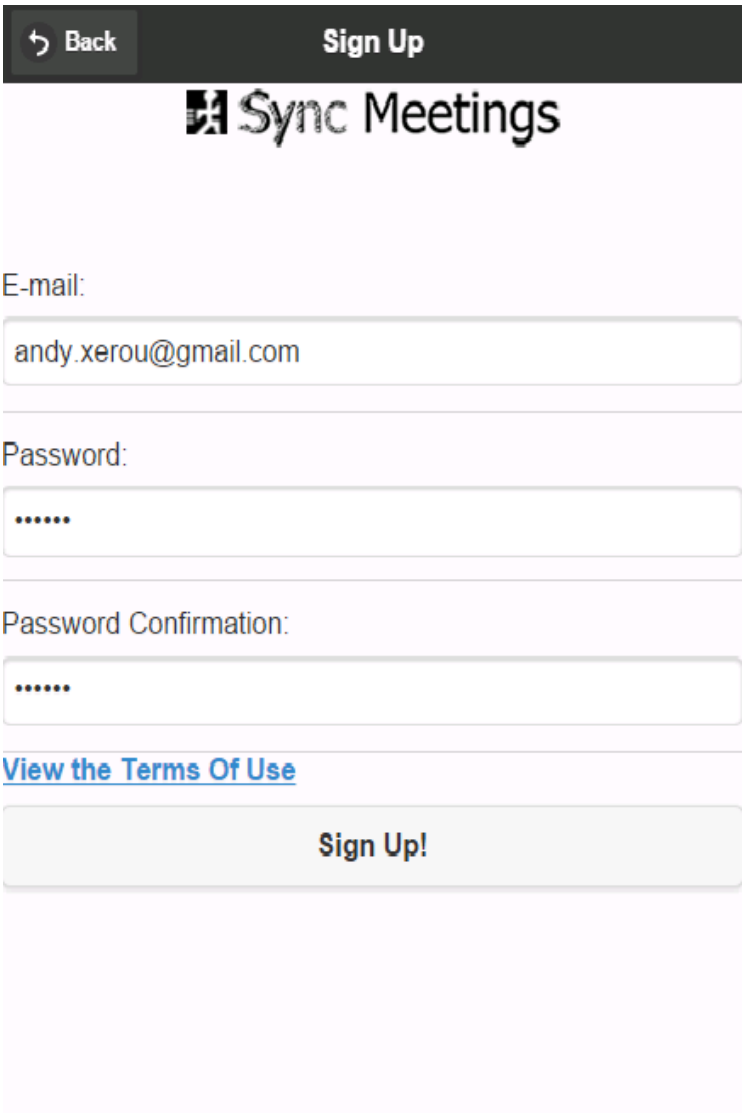
ΣΧΕΔΙΑΣΜΟΣ

Η ιστοσελίδα θα έχει τη μορφή web site και αποτελεί ουσιαστικά ένα περιβάλλον όπου υπάρχει η δυνατότητα εγγραφής χρηστών όπως και η πλήρης διαχείριση των λογαριασμών και προφίλ αυτών. Για να επιτευχθεί αυτό θα πρέπει αρχικά να δημιουργηθεί μια βάση δεδομένων στην οποία θα συσχετίζονται οι χρήστες, τα ραντεβού, η τοποθεσία των ραντεβού, η τοποθεσία των χρηστών και το προφίλ των χρηστών (βλ. Εικόνα 10.17).

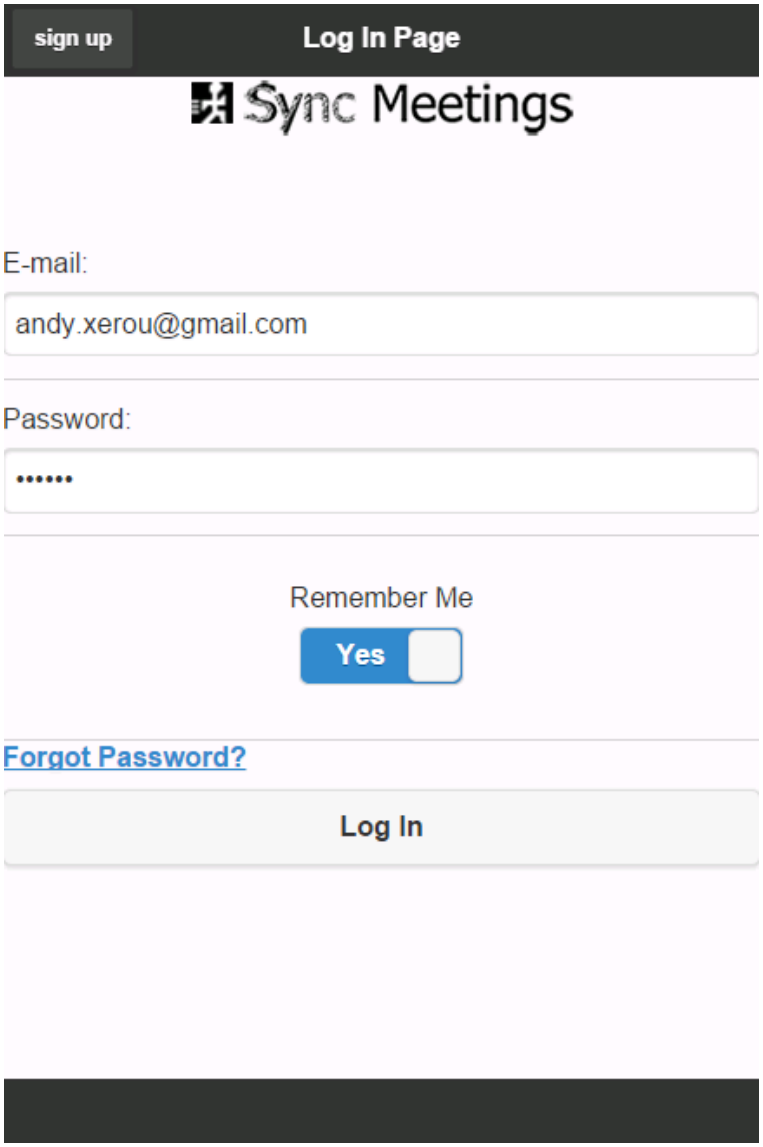


Εικόνα 10-17 Σχεδιάγραμμα Βάσης Sync Meetings

Φυσικά θα υπάρχουν οι φόρμες εγγραφής και σύνδεσης όπου ο χρήστης θα μπορεί να εγγραφεί και να συνδεθεί στην εφαρμογή (βλ. Εικόνα 10.18, 10.19).

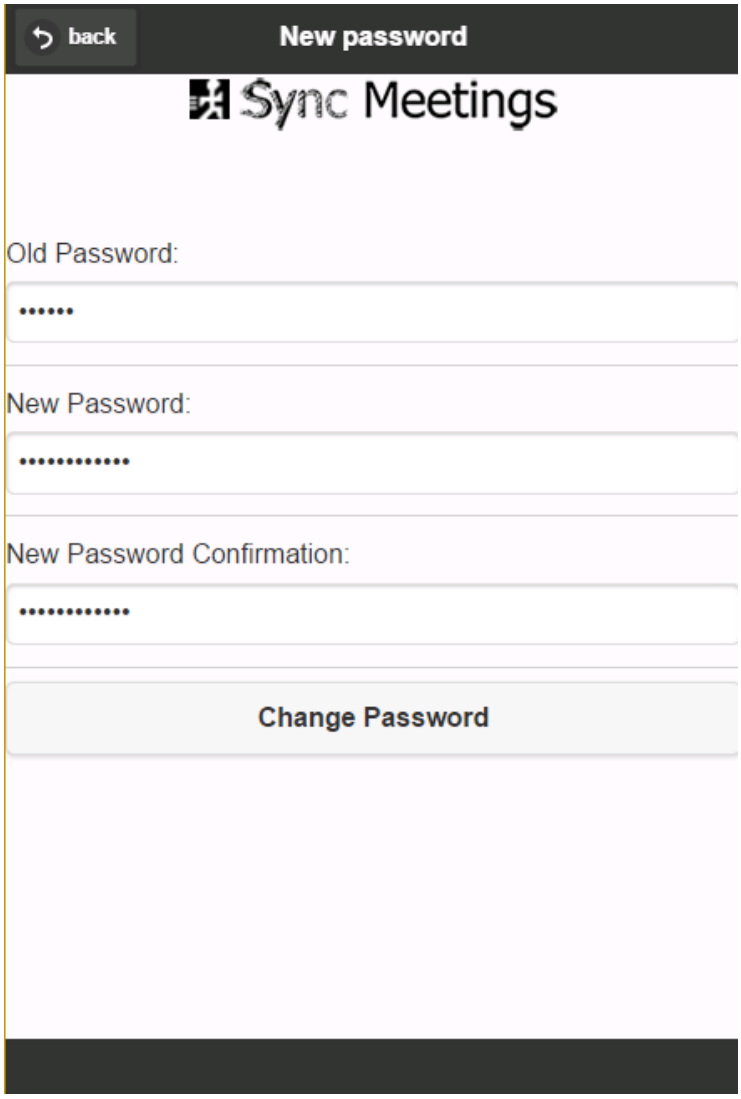


Εικόνα 10.18 Sign Up Σελιδα



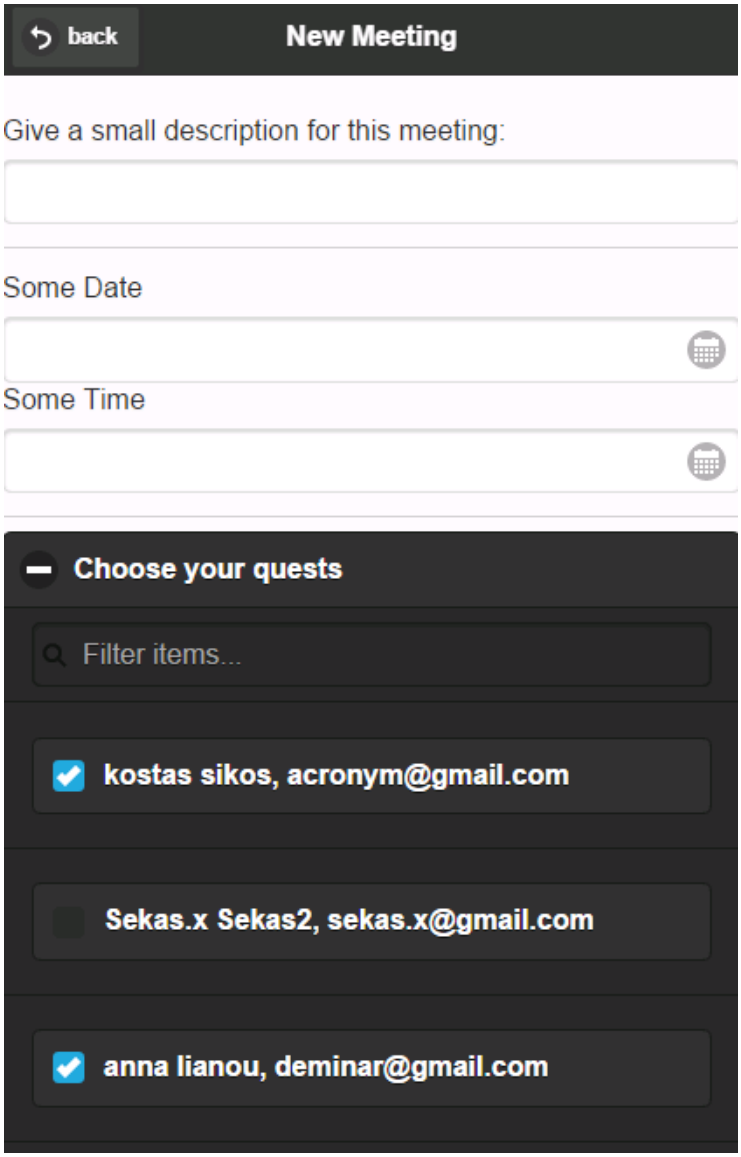
Εικόνα 10-19 Login Σελίδα

Επίσης θα υπάρχει δυνατότητα απομνημόνευσης των στοιχείων του χρήστη μετά τη πρώτη εγγραφή, έτσι ώστε όταν συνδεθεί ξανά να μεταβεί κατευθείαν στη κύριως σελίδα της εφαρμογής, όπως και η δυνατότητα αλλαγής του κωδικού του σε περίπτωση που τον έχει ξεχάσει μέσω αποστολής mail (βλ. Εικόνα 10.20).



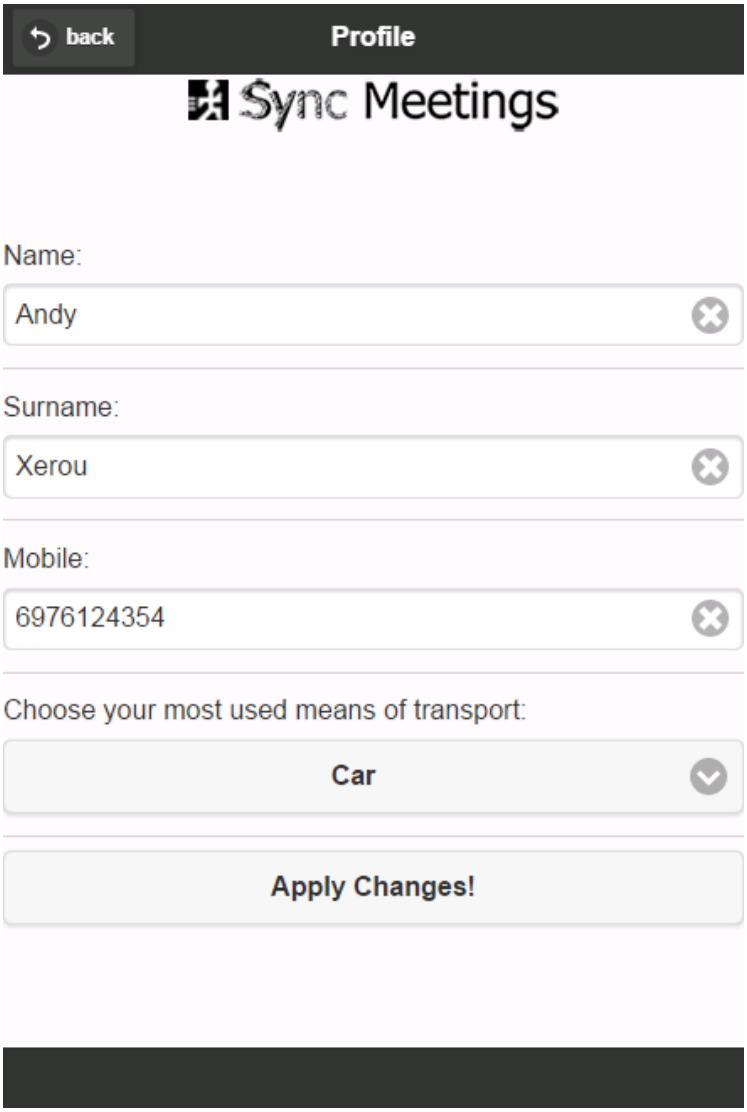
Εικόνα 10-20 Αλλαγή Κωδικού

Οι χρήστες που θα έχουν εγγραφεί θα εμφανίζονται αυτόματα στη λίστα των ατόμων που μπορεί να προσκαλέσει ο συντονιστής (βλ. Εικόνα 10.21).



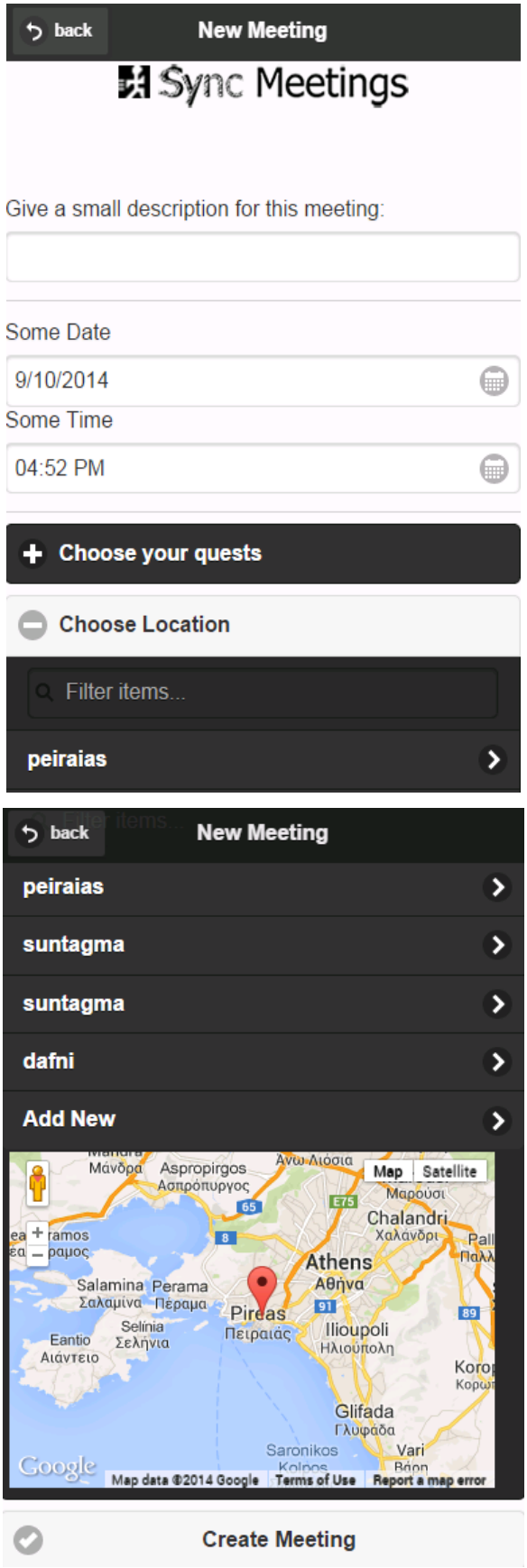
Εικόνα 10-21 Προσκεκλημένοι

Στη συνέχεια πρέπει να σκεφτούμε πως μπορεί η διεπαφή να γίνει πιο κατανοητή και φιλική προς το χρήστη. Αυτό επιτυγχάνεται με τη χρήση μικρού αριθμού κουμπιών, αποφυγή περιττών πληροφοριών και αισθητικού σχεδιασμού. Ο χρήστης θα μπορεί εύκολα να γραφτεί και να κάνει login στον ιστότοπο, επιπλέον θα μπορεί να διαβάσει πληροφοριές για την εφαρμογή, και να διαχειριστεί προσωπικά του στοιχεία όπως αλλαγή κωδικού, επιλογή ψευδόνυμου κ.α (βλ. Εικόνα 10.22).



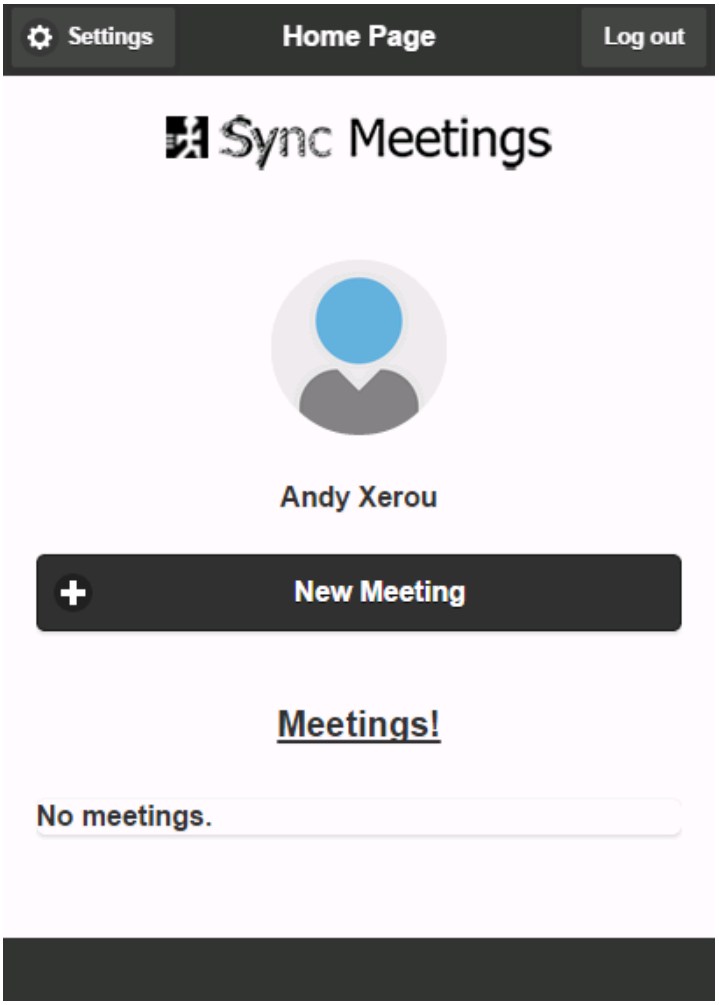
Εικόνα 10-22 Προφίλ

Παρακάτω υπάρχει η διαχείρηση των συναντήσεων που πραγματοποιείται ως εξής: ο χρήστης-συντονιστής οργανώνει συνάντηση στέλνοντας προσκλήσεις σε όσους θα συμμετέχουν καθοριζοντας ημερομηνία, ώρα και τόπο συνάντησης (βλ. Εικόνα 10.23).



Εικόνα 10-23 Οργάνωση Ραντεβού

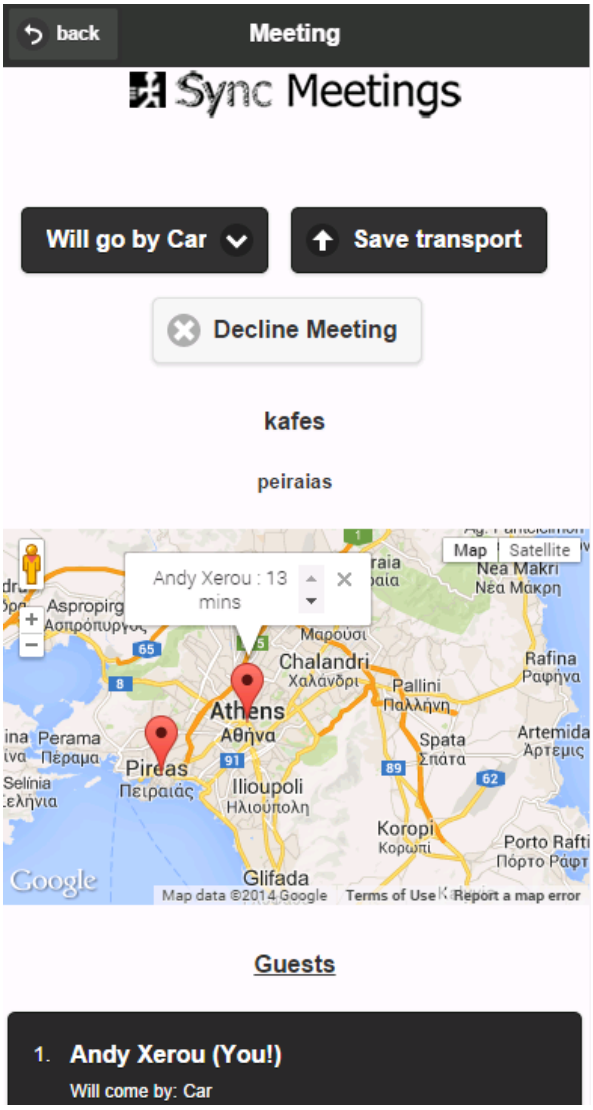
Κάθε χρήστης που προσκαλείται στο ραντεβού, καθορίζει με τι μέσο θα μετακινηθεί, για να μπορεί το σύστημα να εκτιμήσει τον χρόνο που απαιτείται ανάλογα με τη θέση του. Για τη διευκόλυνση του χρήστη το κουμπί δημιουργίας νέου meeting έχει υλοποιήθει στην αρχική σελίδα (βλ. Εικόνα 10.24).



Εικόνα 10-24 Αρχική Σελίδα

Πατώντας το θα μεταβαίνει στη σελίδα στην οποία θα καθορίζει τα εξής: θα δίνει μια μικρή περιγραφή για το ραντεβού, θα επιλέγει τα άτομα τα οποία θα έρθουν και τη διεύθυνση του ραντεβού, η οποία θα απεικονίζεται στο χάρτη (βλ. Εικόνα 10.23).

Στην πορεία θα μεταβεί στη σελίδα οπου θα δηλώσει το μέσο μεταφοράς, να το αποθηκεύσει, να δεί τη τοποθεσία στο χάρτη και να έχει τη δυνατότητα να απορρίψει το ραντεβού. Επιπλέον, θα παρακολουθεί τη θέση όλων των χρηστών με χρήση GPS , θα υπολογίζει πότε πρέπει να ξεκινήσουν για μια συνάντηση ανάλογα με τη τοποθεσία, την τρέχουσα θέση τους και το μέσο που θα έχει επιλεχτεί (πόδια, αυτοκίνητο, μέσα μαζικής μεταφοράς), όπως και την εκτίμηση σε πόση ώρα θα φτάσουν. Αυτό θα υλοποιηθεί με τον εξής τρόπο: πάνω στο χάρτη θα εμφανίζονται διαδοχικά οι markers των προσκεκλημένων στο ραντεβού δίχνοντας το όνομα και το χρόνο άφιξής τους στο ραντεβού (βλ. Εικόνα 10.25).



Εικόνα 10-25 Τελικό στάδιο ραντεβού

Για τη παρούσα πτυχιακή εργασία υλοποιήθηκαν οι τεχνολογίες: JQuery Mobile, JQuery, HTML5, Javascript, PHP, Ajax και CSS3. Ο editor που χρησιμοποιήθηκε για την υλοποίηση του site είναι το dreamweaver CS6. Ενδεικτικά σημεία κώδικα υπάρχουν στην επόμενη ενότητα. Να σημειωθεί επίσης η χρήση του xampp (Ένα πακέτο προγραμμάτων ελεύθερου λογισμικού , λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει τον εξυπηρετητή Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε PHP και Perl.) το οποίο εγκατέστησα και στο τερματικό μου όπως και στο server . Επίσης έγινε χρήση του MySQL Workbench για την υλοποίηση της βάσης δεδομένων του project και τέλος, για τη τελική δοκιμή της εφαρμογής χρησιμοποίησα τον οκεανό (Είναι μία IaaS Υπηρεσία. "IaaS" σημαίνει "Υποδομών ως Υπηρεσία".) Λόγω της φοιτικής μου ιδιότητας έχω το δικαίωμα ελεύθερης χρήσης του λογισμικού και έτσι έχω το δικό μου Virtual Machine, όπως και Virtual Networks. Στο Pithos λοιπόν του οκεανού αποθήκευσα όλο το κώδικα του project και στις Cyclades δημιούργησα το VM μου όπου και αργότερα μου δόθηκε η ip 83.212.122.103. Στη συγκεκριμένη ip έγιναν όλα τα πειράματα αν και υπήρξαν επιπλοκές στη πορεία της υλοποίησης όπως θέματα σύνδεσης με το server, κακό σήμα 3G και λίγος χρόνος για το remote connection μεταξύ client και server όπου έληγε η σύνδεση και έπρεπε εκ νέου να ξεκινήσει το πείραμα.

ΚΩΔΙΚΑΣ

Εδώ παρουσιάζουμε κομμάτια κώδικα όλης της εφαρμογής, ξεκινώντας με το κώδικα html. (βλ. Εικόνα 10-26, 10-27 10-28, 10-29)

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>jQuery Mobile Web App</title>

    <link href="js/jquery.mobile-1.4.1.min.css" rel="stylesheet" type="text/css"/>
    <link rel="stylesheet" type="text/css" href="http://dev.jtsage.com/cdn/datebox/latest/jqm-datebox.min.css" />
    <script src="js/jquery-2.1.0.min.js" type="text/javascript"></script>
    <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAU1Rh7v0tDZ9_VoJJqaOnN0HkQjC_oqqg&sensor=false">
</script>
    <script src="js/jquery.mobile-1.4.1.min.js" type="text/javascript"></script>
    <script src="js/jquery.datetimepicker.css" type="text/javascript"></script>
    <script src="js/jquery.datetimepicker.js" type="text/javascript"></script>
    <script src="js/functions.js" type="text/javascript"></script>
    <script type="text/javascript" src="http://dev.jtsage.com/cdn/datebox/latest/jqm-datebox.core.min.js"></script> timebox
    <script type="text/javascript" src="http://dev.jtsage.com/cdn/datebox/latest/jqm-datebox.mode.datebox.min.js"></script>
    <script type="text/javascript" src="http://dev.jtsage.com/cdn/datebox/latest/jqm-datebox.mode.timebox.min.js"></script>
    <script type="text/javascript" src="http://dev.jtsage.com/cdn/datebox/i18n/jquery.mobile.datebox.i18n.en_US.utf8.js"></script>
    <link rel="stylesheet" type="text/css" href="basic.css" media="screen" />
</head>
<body>
```

Εικονα 10-26

Εδώ παρατηρούμε τη παραμετροποίηση της γλώσσας, την ετικέτα με την οποία τοποθετούμε το περιεχόμενο μας σε προβολή για mobile συσκευές ,τον τίτλο της σελίδας, τη σύνδεση του html κώδικα με το css αρχείο και τις βιβλιοθήκες του jquery, datebox, όπως και google maps του project.

```
<div data-role="page" id="homePage">
    <div data-role="header" data-theme="b" data-position="fixed" data-fullscreen="true">
        <h1>Home Page</h1>
        <a href="#settings" data-role="button" data-icon="gear" data-transition="fade">
            Settings
        </a>
        <a href='javascript:logOut()' data-role="button" data-transition="fade" id="logOutButton">
            Log out
        </a>
    </div>
    <div data-role="content">
        <div align="center">
            <img src="images/logo.png" />
        </div>
        <div align="center" id="profPicDiv">
            <img name="profilePic" src="images/empty_profile.png"  alt=""  id='profilePic'>
        </div>
        <a href="#newMeeting" data-theme="b" data-role="button" data-icon="plus" data-transition="fade">
            New Meeting
        </a>
        <div id="listOfMeetings" >
            <!-- data fill from server -->
            <div align="center" id="meetingHeader">
                <p>Meetings!</p>
            </div>
            <ul data-theme="b" data-role="listview" data-inset="true" data-split-icon="delete" id='meetingsList'>
              <!-- list of meetings -->
                        </ul>
           </div>
            </div>
    <div data-role="footer" data-position="fixed" data-theme="b" data-fullscreen="true">
        <h4></h4>
    </div>
</div>
```

Εικονα 10-27

Εδώ εμφανίζουμε το κώδικα της αρχικής σελίδας του project, όπου τοποθετούμε header και footer buttons-links για άλλες σελίδες, το logo της εφαρμογής, ορίζουμε τα θέματα και τοποθετούμε τη λίστα των ραντεβού μας, η οποία θα εμφανίζεται δυναμικά στη σελίδα οταν θα οργανώνεται μία συνάντηση.

```
<!-- Log In Page-->
<div data-role="page" id="logIn" >
    <div data-role="header" data-theme="b" data-position="fixed" data-fullscreen="true">
        <h1>Log In Page</h1>
        <a href="#signUp" data-role="button" data-transition="fade">
            sign up
        </a>
    </div>
    <div data-role="content" id="logInContent">
        <div align="center"> <img src="images/logo.png" /> </div>

      <div data-role="fieldcontain" id="email">
        <label for="textinput">E-mail:</label>
        <input type="text" name="textinput2" id="textinput" value="xz"  />
      </div>

      <div data-role="fieldcontain" id="pass">
        <label for="password">Password:</label>
        <input type="password" name="password" id="password" value="sas"  />
      </div>

      <div data-role="fieldcontain" id="rememberMe" align="center">

        <div data-role="fieldcontain" id="remCheck">
          <label for="rememberCheck">Remember Me</label>
          <select name="flipswitch" id="rememberCheck" data-role="slider">
            <option value="off">No</option>
            <option value="on">Yes</option>
          </select>
        </div>
      </div>
      <a href="#forgotPass" >Forgot Password?</a>
      <div id='wrong'></div>
      <div id="logInButton">
        <button>Log In</button>
      </div>

    </div>
    <div data-role="footer" data-theme="b" data-position="fixed" data-fullscreen="true">
        <h4></h4>
    </div>
</div>
```

Εικονα 10-28

Εδώ εμφανίζεται ο κώδικας για τη login σελίδα. Υπάρχουν κλασσικά τα header και footer της σελίδας, button-links για άλλες σελίδες, φόρμα με τα πεδία εγγραφής, το forgot password button για μεταβαση στη σελίδα ανάκτησης κωδικού και το slider (remember me) για την αυτόματη αποθήκευση των στοιχείων του χρήστη για την επόμενη σύνδεσή του στο site.

```
<div data-role="page" id="editProfile">

    <header data-role="header" data-theme="b" data-position="fixed" data-fullscreen="true">
        <h1>Profile</h1>
    <a href="#settings" data-role="button" data-icon="back" data-transition="fade">back</a>
    </header>
    <div data-role="content">
        <div align="center">
            <img src="images/logo.png" />
        </div>
        <div data-role="fieldcontain" id="field">
          <label for="name">Name:</label>
          <input data-clear-btn="true" type="text" name="textinput2" id="name" value=""/>
        </div>
        <div data-role="fieldcontain" id="field">
        <label for="surName">Surname:</label>
        <input data-clear-btn="true" type="text" name="textinput3" id="surName" value=""/>
        </div>
        <div data-role="fieldcontain" id="field">
         <label for="mobile">Mobile:</label>
        <input data-clear-btn="true" type="text" name="textinput4" id="mobile" value=""/>
        </div>

        <div class="ui-field-contain">
            <label for="transport">Choose your most used means of transport:</label>
            <select name="select-native-1" id="transport">
                <option value="foot">Foot</option>
                <option value="car">Car</option>
                <option value="MMM">Public means of transport</option>
            </select>
        </div>
        <div id="applyButton">
            <button>Apply Changes!</button>
        </div>
    </div>
    <div data-role="footer" data-theme="b" data-position="fixed" data-fullscreen="true">
        <h4></h4>
    </div>
</div>
```

Εικονα 10-29

Εδώ εμφανίζεται ο κώδικας για τη profile σελίδα. Υπάρχουν κλασσικά τα header και footer της σελίδας, button-links για άλλες σελίδες, φόρμα με τα πεδία συμπλήρωσης προσωπικών στοιχείων του χρήστη, button για επιλογή μεταφορικού μέσου και τέλος το apply button για αποθήκευση των επιλογών.

Συνεχίζουμε με ενδεικτικό κώδικα javascript (βλ. Εικόνα 10.30, 10.31, 10.32, 10.33, 10.34)

```
//sing Up function
$('#signUpButton').click( function()
{
//alert("sign up func");
var e = $("#signUpContent").find('#email').find('#textinput').val(); // email
var wrong = "";
var p1 = $("#signUpContent").find('#pass').find('#password').val(); // password1
var p2 = $("#signUpContent").find('#passConf').find('#password').val(); // passworg2
var n = $("#signUpContent").find('#name').find('#textinput').val(); // name
var sN = $("#signUpContent").find('#surname').find('#textinput').val(); // surName

if ( e == "" || p1 == "" || p2 == "" )
{
    wrong += "<p>Fill out all of the form data</p>";
    $("#signUpContent").find('#wrong').html(wrong);
    return;
}
else if(!validateEmail(e))
{
    wrong += "<p>This is not an email.</p>";
    $("#signUpContent").find('#wrong').html(wrong);
    return;

}
else if(!validPass(p1,p2))
{


    wrong += "<p>This is not a valid password</p>";
    $("#signUpContent").find('#wrong').html(wrong);
    return;
}
```

Εικόνα 10-30

Εδώ παρατηρούμε τον έλεγχο για τα πεδία του κωδικού και επαλήθευσης κωδικού, όπως και του mail. Ορίζουμε μεταβλητές για τα πεδία ,στη συνέχεια κοιτάμε για κενά πεδία σε password και mail, όπως και την ορθότητά τους. Με το πάτημα του button sign up καλείται η function και πραγματοποιούνται όλοι οι έλεγχοι παράγωντας σε περίπτωση λάθους εσωτερικό μήνυμα (innerHTML) στη σελίδα για να ειδοποιηθεί ο χρήστης.

```
$('#newMeetingButton').click(function()
{
    var friendList = [];
    var date = $('#newMeeting').find('#mydate').val();
    var time = $('#newMeeting').find('#mytime').val();
    var desc = $('#newMeeting').find('#newMtngDesc').val();

    $('#friendList input:checked').each(function() {
        friendList.push($(this).data('uid'));
    });
    if(friendList.length < 1)
    {
        alert("Select at least 1 quest.");
        return false;
    }
    var locId = $('#newMeeting').data("locId");
    if(!locId)
    {
        alert("Select Location");
        return false;
    }

    var guestsList = friendList.join(',');
    //alert(guestsList);
    var dataStr = "action=newMeeting&desc="+desc+"&date="+date+"&time="+time+"&locId="+locId+"&guestsList="+guestsList;
    $.ajax({
        url: serverName+"api.php",
        type: "POST",
        async: true,
        data: dataStr,

        success: function(response)
        {
            if(response == "meeting_fail")
            {
                alert("Meeting fail");
            }
            else
            {
                //alert(response);
                meetingPage(response);
```

Εικόνα 10-31

Εδώ συνεχίζονται οι έλεγχοι, έχοντας ορίσει μεταβλητές για ώρα, ημερομηνία και επιλογή καλεσμένων. Πατώντας το meeting button καλέιται το function και πραγματοποιούνται οι εξής έλεγχοι: Αν έχει επιλεχτεί τουλάχιστον ένας χρήστης και αν έχει επιλεχτεί τοποθεσία του ραντεβού. Τέλος μέσω ajax αποστέλνονται τα στοιχεία: ώρα, ημερομηνία, καλεσμένοι χρήστες, τοποθεσία στο server μέσω post εντολής και ελέγχεται αν οργανώθηκε επιτυχώς το ραντεβού ή όχι.

```
function alertGeoloc()

    var lat = pos.coords.latitude;
        var long = pos.coords.longitude;
        alert(lat + " " + long);



function checkLogInStatus()


    var ajax = ajaxObj("POST", serverName+"api.php");

    ajax.onreadystatechange = function()
    {
        if(ajaxReturn(ajax) == true)
        {
            //alert("|"+ajax.responseText+"|");
            if(ajax.responseText == "no_user")
            {
                //alert(ajax.responseText);
                window.location = "#logIn";
            }
            else
            {
                //alert(ajax.responseText);
                window.location = "#homePage";
                clearInterval(locInterval);
                locInterval = setInterval(sendGeoLoc, repeatTime); // every 1 minute
            }

        }
    };
    //alert("This is the logOut function!!");
    ajax.send("action=checkLogInStatus");
```

Εικόνα 10-32

Εδώ ορίζεται η function alertGeoloc με τις μεταβλητές longtitude kai latitude, βγάζοντας alert μήνυμα. Στη πορεία υπάρχει η function checkLogInStatus η οποία ελέγχει αν έχει σταλθει το xml http request μήνυμα στο server (δηλαδή το object ajaxObj) , αν είναι έτοιμος ο server να το επεξεργαστεί τότε το ajax object επιστρέφει μήνυμα το οποίο αν είναι no user τότε μας επιστρέφει στη login σελίδα για να συνδεθούμε ξανά, διαφορετικά μας επιστρέφει στην αρχική σελιδα. Τέλος ανανεώνονται τα στοιχεία ανά ένα λεπτό.

```
function drop(people, meetingMap)
{

    var map = meetingMap;
    var marker;
    for (var i = 0; i < markers.length; i++) {
        markers[i].setMap(null);
    }

    markers = [];

    for (var i in people)
    {
        mark = people[i];
        var contentString = mark.name + " " + mark.surName + " : " + mark.duration;
        var infowindow = new google.maps.InfoWindow({
                              content: contentString
                          });

        var myLatLng = new google.maps.LatLng(mark.lat, mark.long);
        marker = new google.maps.Marker({
            position: myLatLng,
            map: meetingMap,
            title: mark.name,
            animation: google.maps.Animation.DROP
        });
        infowindow.open(meetingMap,marker);
        markers.push(marker);

    }

}
```

Εικόνα 10-33

Εδώ παρατηρούμε το κώδικα , όπου ορίζουμε το χάρτη και τους markers. Ταυτοποιούμε τους ανθρώπους ως markers, δίνοντάς τους όνομα, επώνυμο και χρόνο άφιξης στο τελικό προορισμό. Ορίζουμε το marker στο google maps με το γεωγραφικό μήκος ,πλάτος και απεικόνησή του.

```
function validPass(pas1, pas2)
{
    //den graftei password h einai idio me to confirmation h <4 xarakthres
    if(pas1 == "" || pas2 != pas1 || pas1.length < 4)
    {
        return false;
    }

    return true;



}

function validName(name)
{
    re = /[0-9]/;
    //den epitrepontai ari8moi kai to onoma na einai <2 xarakthres
    if(re.test(name) || name == "" || name.length <2)
    {
        return false;
    }
    return true;
}


function getField(field, op)
{
    op = op || "get"; //default get
    var value = $("#editProfile").find('#'+field).val();

    var dataStr = "action="+field+"&op="+op;
    if(op == "update")
    {
        dataStr += "&val="+value;
    }
    var resp = "something";
    $.ajax({
      url: serverName+"api.php",
      type: "POST",
      async: false,
      data: dataStr,
```

Εικόνα 10-34

Εδώ παρατηρούμε τον έλεγχο για : αν το πεδίο του κωδικού είναι κενό, αν ταυτίζεται το πρώτο πρεδίο κωδικού με το δεύτερο και αν είναι το μήκος του κωδικου τουλάχιστον 4 χαρακτήρες. Επίσης ελέγχουμε την ορθότητα του ονόματος, αν είναι κενό το πεδίο του και αν είναι τουλαχιστον 2 χαρακτήρες μεγάλο. Αν υπάρξει λάθος τοτε εμφανίζεται αντίστοιχο alert μήνυμα με τις κατάλληλες οδηγίες για επανασυμπλήρωση σωστών στοιχείων. Τέλος με τη

function getField στέλνει τα ανανεωμένα values των πεδίων της profil σελίδας με την εντολή post στο server.

Τέλος, ενδεικτικός κώδικας php.(βλ. Εικόνα 10-35, 10-36, 10-37, 10-38, 10-39)

```php
<?php

    //check log in status
    session_start();
    include 'Model/database/connection.php';

function evalLoggedUser($conx,$DBName,$uid,$cookie)
{
    $sql = "SELECT User_id FROM $DBName.session WHERE User_id='$uid' AND cookie='$cookie' LIMIT 1";
    $query = mysqli_query($conx, $sql);
    $numrows = mysqli_num_rows($query);
    if($numrows > 0)
    {
        return true;
    }
    return false;
}

    $user_ok = false;
    $log_id = "";
    $log_cookie = "";

    // User Verify function

    if(isset($_SESSION["uid"]) && isset($_SESSION["cookie"]) )
    {
        $log_id = preg_replace('#[^0-9]#', '', $_SESSION['uid']);
        $log_cookie = preg_replace('#[^a-z0-9]#i', '', $_SESSION['cookie']);

        // Verify the user
        $user_ok = evalLoggedUser($db_conx, $DBName, $log_id,$log_cookie);
    }
    else if(isset($_COOKIE["uid"]) && isset($_COOKIE["cookie"]) )
    {
        $_SESSION['uid'] = preg_replace('#[^0-9]#', '', $_COOKIE['uid']);
        $_SESSION['cookie'] = preg_replace('#[^a-z0-9]#i', '', $_COOKIE['cookie']);

        $log_id = preg_replace('#[^0-9]#', '', $_SESSION['uid']);
        $log_cookie = preg_replace('#[^a-z0-9]#i', '', $_SESSION['cookie']);
        // Verify the user
        $user_ok = evalLoggedUser($db_conx,$DBName,$log_id,$log_cookie);
```

Εικόνα 10-35

Εδώ πιστοποιούμε το χρήστη που έχει εγγραφτεί, βλέπουμε αν έχει αυξηθεί το numrows απο 0, τελος πιστοποιούμε το function και τον χρήστη μέσω της συνθήκης if και μέσω coockies.

```php
<?php

$DBServer = "localhost:3306"; // e.g 'localhost' or '192.168.1.100'
$DBUser   = "root";
$DBPass   = "1qaz2wsx3EDC";
$DBName   = "meetinggap";

$db_conx = mysqli_connect($DBServer, $DBUser, $DBPass, $DBName);
// Evaluate the connection
if (mysqli_connect_errno()) {
    echo mysqli_connect_error();
    exit();
}
```

Εικόνα 10-36

Εδώ παρατηρούμε το κώδικα , όπου γίνεται η σύνδεση της βάσης δεδομένων με το site με την εντολή mysqli_connect. Ορίζονται φυσικά ο server, ο χρήστης, ο κωδικός και το όνομα της βάσης.

```php
<?php

function getField($db_conx,$DBName,$field,$id)
{

    $sql = "SELECT $field FROM $DBName.user WHERE id=$id LIMIT 1";
    $query = $db_conx->prepare($sql);
    $query->execute();
    $query->bind_result($name);
    $query->fetch();

    return $name;
}

function getUsers($db_conx,$DBName)
{

    $sql = "SELECT u.id, u.email, u.name, u.surname FROM $DBName.user u WHERE u.activated='1' ";
    $query = $db_conx->prepare($sql);
    $query->execute();
    $query->bind_result($id, $email, $name, $surName);

    $users = array();
    while($query->fetch())
    {
        $users[] = array($id, $email, $name, $surName);
    }

    return $users;
}


function getIdByMail($db_conx,$DBName,$email)
{

    $sql = "SELECT id FROM $DBName.user WHERE email='$email' LIMIT 1";
    $query = $db_conx->prepare($sql);
    $query->execute();
    $query->bind_result($id);
    $query->fetch();
    return $id;
}
```

Εικόνα 10-37

Εδώ παρατηρούμε την εκτέλεση 3 select queries (ερωτημάτων) στη βάση δεδομένων. Στην συνέχεια με την εντολή prepare σε δυνδιασμό με τη χρήση παραμέτρων μέσα στα ερωτήματα τα προστατεύει απο injection attacks. Μετά εκτελεί την εντολή με το execute(); και τέλος με το bind result αναθέτει το αποτέλεσμα του ερωτήματος σε μία μεταβλητή.

```
unction getUsersforReminder($db_conx, $DBName)

    $sql = "SELECT u.name, u.email, u.transport, m.description, u.Location_id, g.User_id, g.Meeting_id, m.location_id
            FROM $DBName.meeting m, $DBName.guests g, $DBName.user u, $DBName.location l
            where m.meetingDate < DATE_ADD(NOW(),INTERVAL 120 MINUTE)
            and m.meetingDate > NOW()
            and m.id = g.Meeting_id
            and g.reminderSent = '0'
            and g.User_id = u.id
            and m.location_id = l.id";
    $query = $db_conx->prepare($sql);
    $query->execute();
    $query->bind_result($name, $email, $transport, $mDesc, $lid, $uid, $mid, $locid);
    $users = array();
    while($query->fetch())
    {
        $users[] = array("name"=>$name, "email"=>$email, "mDesc"=>$mDesc, "transport"=>$transport,
                         "lid"=>$lid, "uid"=>$uid, "mid"=>$mid, "mlocid"=>$locid);
    }
    return $users;


/apo8hkeuei oti o uid eidopoih8hke gia to meeting mid
unction reminderSent($db_conx, $DBName,$uid, $mid)

    $sql = "UPDATE $DBName.guests
            SET reminderSent = '1'
            WHERE Meeting_id = $mid
            AND User_id = $uid";
    $query = $db_conx->prepare($sql);
    $query->execute();
```

Εικόνα 10-38

Εδώ απο την εκτέλεση του ερωτήματος επιστρέφεται ο πίνακας users. Στη συνέχεια η μέθοδος reminderSent ανανεώνει τη στήλη reminderSent στη βάση δεδομένων και βάζει τη τιμή «1» στη στήλη με τις συνθήκες που ακολουθούν.

```php
function insertMeeting($db_conx,$DBName, $desc, $dateTime, $uid, $locId)
{


    $sql = "INSERT INTO $DBName.meeting(description, meetingDate, creator, location_id)
                VALUES('$desc', STR_TO_DATE('$dateTime','%c/%e/%Y %h:%i %p'), $uid, $locId)";

    $query = $db_conx->prepare($sql);
    $query->execute();
    $mid = mysqli_insert_id($db_conx);
    //echo  "Prepare failed: (" . $query->errno . ") " . $query->error;

    return $mid;
}

function insertLocation($db_conx,$DBName, $lat, $long, $desc=null)
{
    $sql = "INSERT INTO $DBName.location(latitude, longitude, description)
                VALUES($lat, $long, '$desc')";

    $query = $db_conx->prepare($sql);
    $query->execute();
    $lid = mysqli_insert_id($db_conx);
    //echo  "Prepare failed: (" . $query->errno . ") " . $query->error;

    return $lid;
}



/*
 * set guest by meeting id*/
function insertGuest($db_conx,$DBName, $uid, $meetingId, $transport)
{
    //insert guest
    $sql = "INSERT INTO $DBName.guests(Meeting_id, User_id, transport)
                            VALUES($meetingId, $uid, '$transport')";
    $query = $db_conx->prepare($sql);
```

Εικόνα 10-39

Εδώ εισάγονται οι νέες εγγραφές στη βάση δεδομένων που αφορούν το νέο meeting , χρήστη και τοποθεσία.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία:

- Brad Broulik: Pro jQuery Mobile. 2011
- Kris Hadlock: jQuery Mobile Develop and Design. 2012

Sites:

- http://jquerymobile.com/
- http://jqueryui.com/themeroller/
- http://wave.webaim.org/
- http://www.apple.com/accessibility/iphone/vision.html
- http://www.google.com/analytics/
- http://www.omniture.com/
- http://developer.apple.com/library/ios/documentation/userexperience/
- conceptual/mobilehig/MobileHIG.pdf/
- http://www.w3.org/TR/css3-mediaqueries/
- http://filamentgroup.com/examples/iosScaleBug/
- http://adactio.com/journal/4470/
- https://github.com/scottjehl/Respond
- http://glyphish.com/
- http://www.webdirections.org/sotmw2011/
- http://www.quirksmode.org/html5/inputs_mobile.html
- http://code.google.com/p/mobiscroll/
- http://pukupi.com/post/1964
- http://www.westciv.com/tools/gradients/ or http://gradients.glrzad.com/
- http://www.westciv.com/tools/gradients/ or http://gradients.glrzad.com/
- http://kuler.adobe.com

- http://api.jquery.com/
- https://addons.mozilla.org/en-US/firefox/addon/jsonview/
- http://jsonlint.com/
- http://static.springsource.org/spring/docs/current/spring-frameworkreference/
  html/mvc.html
- http://www.webdirections.org/sotmw2011/
- http://code.google.com/apis/maps/documentation/javascript/basics.html
- http://www.phonegap.com/
- http://developer.apple.com/xcode/
- http://www.phonegap.com/start#android