

Linux Labs – Firefox

Του Βασίλη Καρακώδα vassilios.karakoidas@gmail.com



Ο Βασίλης Καρακώδης είναι υποψήφιος διδάκτορας του τμήματος Διοικητικής Επιστήμης και Τεχνολογίας.

Firefox: Ανάπτυξη add-ons (Μέρος II)

Σε αυτή τη σειρά άρθρων κάνουμε μία εισαγωγή στην ανάπτυξη επεκτάσεων για τον Firefox, με βάση τη γλώσσα XUL και την πολύ αγαπημένη Javascript.



Τεκμηρίωση
για addons

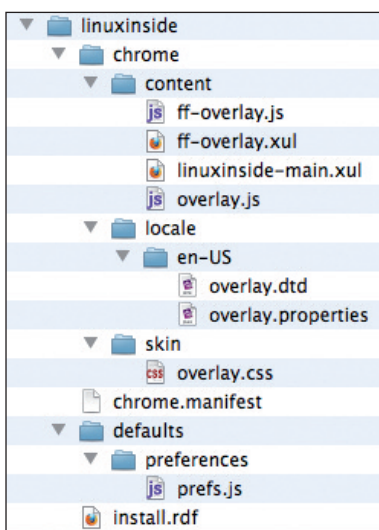
Εργαλεία: Firefox

Δυσκολία: ★★☆☆☆

URL: <http://goo.gl/74mFQ>

Στο προηγούμενο τεύχος, είδαμε με τη σειρά τα δομικά στοιχεία από τα οποία αποτελείται ένα κανονικό Add-On του Firefox. Αναλύσαμε ένα προς ένα τα βασικά αρχεία (**Εικόνα 1**) που υπάρχουν και είδαμε πώς μπορούμε να φτιάξουμε και να κάνουμε install ένα βασικό Add-On, χωρίς ιδιαίτερη λειτουργικότητα.

Σε αυτό το άρθρο, θα επιχειρήσουμε να πάμε λίγο πιο βαθιά. Δηλαδή, θα επικεντρωθούμε στα αρχεία που περιγράφουν το γραφικό περιβάλλον (user interface) των Add-On. Αυτά τα αρχεία είναι γνωστά και ως XUL. Προτού προχωρήσουμε παρακάτω, όμως, καλό είναι να ρίξετε μία ματιά στο προηγούμενο άρθρο (σελ. 62-63) για να φρεσκάσετε τη μνήμη σας...



1 Η βασική δομή ενός Add-On.

Συνεχίζοντας... Το αρχείο Overlay.js

Όπως είχαμε αναφέρει στο προηγούμενο τεύχος, με το αρχείο overlay.xul ορίζουμε ότι το add-on μας θα δημιουργούσε μία νέα καταχώριση στο μενού του Firefox, η οποία αν επιλεγεί από το χρήστη θα εκτελέσει τη μέθοδο onMenuItemCommand():

```
<menupopup id="menu_ToolsPopup">
<menuitem id="linuxinside-hello"
label="&Linuxinside.label;"
oncommand="linuxinside.onMenuItemCommand(event);"/>
</menupopup>
```

Όπως έχουμε πει, οι βασικές τεχνολογίες που κρύβονται πίσω από τα Add-on του Firefox είναι η XML και η Javascript. Κατ' αντιστοιχία, εφόσον το αρχείο overlay.xul περιγράφει αυτή την δομική αλλαγή στο μενού του browser σε επίπεδο XML, στο αρχείο overlay.js υλοποιείται η λειτουργικότητα του σε επίπεδο Javascript. Εκεί, δηλαδή, ορίζουμε όλες τις μεθόδους και εκεί γράφουμε όλον τον κώδικα του Add-on μας. Ακολουθούν, λοιπόν, τα περιεχόμενα του αρχείου overlay.js:

```
var linuxinside = {
  onLoad: function() {
    // αρχικοποίηση
    this.initialized = true;
    this.strings = document.getElementById("linuxinside-strings");
  },
  onMenuItemCommand: function(e) {
    window.location.replace("chrome://linuxinside/content/linuxinside-main.xul");
  },

```

```
onToolbarButtonCommand: function(e) {
  // επαναχρησιμοποιούμε την ίδια μέθοδο
  linuxinside.onMenuItemCommand(e);
};
window.addEventListener("load", function ()
{ linuxinside.onLoad(); }, false);
```

Ο παραπάνω κώδικας ουσιαστικά υλοποιεί τη λειτουργικότητα του μενού που προσθέσαμε με τη δήλωση του overlay.js. Πιο συγκεκριμένα, λέει ότι αν ο χρήστης επιλέξει τη συγκεκριμένη εγγραφή στο μενού, τότε πρέπει να «ανοίξει» το παράθυρο linuxinside-main.xul.

Η Δομή των Αρχείων Περιγραφής XUL

Όπως έχουμε αναφέρει, τα αρχεία XUL στην πράξη περιγράφουν τη γραφική διεπαφή των add-ons. Αυτά συνήθως μπορεί να είναι δύο πράγματα, προσθήκες (overlay) στο υπάρχουσα γραφική διεπαφή του Firefox ή παράθυρα (window) που υλοποιούν τη βασική λειτουργικότητα του add-on. Κάποιος που έχει μία βασική εμπειρία στο προγραμματισμό και ειδικότερα στη δημιουργία γραφικών διεπαφών, θα καταλάβει αμέσως ότι βρίσκεται σε γνώριμα μονοπάτια.

Τα βασικά συστατικά που προσφέρονται είναι κλασικού τύπου κουμπιά, λίστες, πεδία εισόδου τιμών κ.ά. Στους «Συνδέσμους» του παρόντος άρθρου μπορείτε να βρείτε μία ιστοσελίδα στο development site του Firefox, που περιγράφει αυτά τα συστατικά ένα-ένα με παραδείγματα. Στο πλαίσιο αυτού του άρθρου, όμως, για οικονομία χώρου, δεν έχει νόημα να επεκταθούμε περισσότερο, οπότε θα προχωρήσουμε απευθείας στη δομή ενός αρχείου XUL. Να ο κώδικας λοιπόν:

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/global.css"
type="text/css"?>
<window xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul" title="Linux Inside Test App">
  <head>
    <script type="application/x-javascript" src="lib/jquery-1.3.2.min.js" />
    ...
  </head>
  <!-- ΣΥΝΤΟΜΕΥΣΕΙΣ -->
  <keyset>
    <key id="reload" keycode="VK_F1" oncommand="reload();" />
    ...
  </keyset>
```

```

<!-- POPUP -->
<popupset>
  <menupopup id="treepopupmenu" onpopupshowing=
    "populatePopupMenu();"
    <!-- ΑΝΤΙΚΕΙΜΕΝΑ ΕΔΩ-->
  </menupopup>
</popupset>
<!-- ΣΥΣΤΑΤΙΚΑ -->
<hbox align="center">
  ...
</hbox>

<!-- ΕΚΤΕΛΕΣΗ JS ΕΔΩ -->
<script type="application/x-javascript">
  ...
</script>
</window>

```

Όπως κάθε έγγραφο XML, τα αρχεία XUL ξεκινούν με την κλασική ετικέτα "`<?xml version="1.0"?>`". Στη συνέχεια, με τη χρήση της ετικέτας "`xml:stylesheet`" ορίζουμε το αρχείο CSS, το οποίο χρησιμοποιείται για το styling του νέου παράθυρου. Ναι, καλά είδατε, όλες οι αλλαγές σε επίπεδο χρωμάτων κ.λπ., γίνονται με βάση τα αρχεία CSS. Συγκεκριμένα, στο παράδειγμα λέμε στον Firefox ότι επιθυμούμε αυτό το παράθυρο να χρησιμοποιεί το στυλ που είναι ενεργό στον Firefox. Αυτό φυσικά μπορεί να τροποποιηθεί κατά το δοκούν...

Στη συνέχεια, παρατηρώντας το XUL, μπορούμε να δούμε ότι κάθε αρχείο αυτού του είδους χωρίζεται σε τρεις βασικές ενότητες.

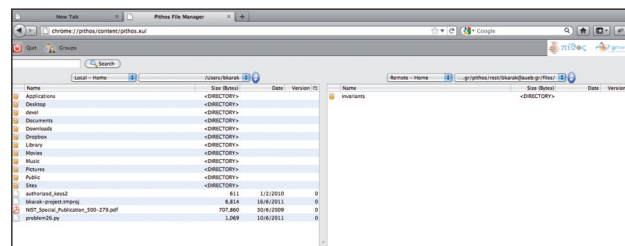
Στην πρώτη ενότητα, που περιλαμβάνει τα περιεχόμενα της ετικέτας "`head`", ορίζουμε κάποια βασικά χαρακτηριστικά (που δεν φαίνονται άμεσα στο χρήστη) του παραθύρου. Για παράδειγμα, μπορούμε να εισαγάγουμε αρχεία Javascript με βιβλιοθήκες, ώστε να τις χρησιμοποιήσουμε. Ο γενικός κανόνας λέει ότι συνήθως, ό,τι λειτουργεί σε μία Web εφαρμογή, θα λειτουργήσει και εδώ, οπότε στον παραπάνω κώδικα ορίζουμε ότι θέλουμε να χρησιμοποιήσουμε τη βιβλιοθήκη jQuery (η οποία δουλεύει κανονικά :)).

Στη δεύτερη ενότητα, ορίζουμε την ετικέτα "`keyset`", η οποία περιέχει τις συντομεύσεις και τους συνδυασμούς πλήκτρων που θα χρησιμοποιούνται στο παράθυρο. Στο παραπάνω παράδειγμα, ορίζουμε ότι κάθε φορά που πατάμε το πλήκτρο F1 θέλουμε να εκτελεστεί η μέθοδος `reload()`. Αυτή η μέθοδος μπορεί να υπάρχει στις τυπικές βιβλιοθήκες που παρέχονται από τις προγραμματιστικές διεπαφές του Firefox ή να έχει υλοποιηθεί από το χρήστη. Στη δεύτερη περίπτωση θα πρέπει απαραίτητα το αρχείο Javascript που την περιέχει να έχει δηλωθεί στην ετικέτα "`head`".

Αφού ορίσαμε και τις συντομεύσεις, περνάμε στην τρίτη ενότητα όπου δηλώνουμε τα συστατικά που θέλουμε να εμφανίζουμε στο παράθυρο. Η ετικέτα "`popupset`" δηλώνει ότι

Σύνδεσμοι

- [1] Firefox add-on - <https://addons.mozilla.org/>
- [2] Mozilla Development Center - <https://developer.mozilla.org/en/extensions>
- [3] Add-on Builder - <http://goo.gl/rgLJH>
- [4] Βασικά Συστατικά XUL - https://developer.mozilla.org/en/XUL_controls



2 Παράδειγμα ολοκληρωμένης γραφικής διεπαφής XUL (Pithos File Manager).

Το αρχείο `overlay.xul` περιγράφει τις δομικές αλλαγές στο μενού του Firefox σε XML, ενώ στο αρχείο `overlay.js` υλοποιείται η λειτουργικότητα του μενού σε Javascript.

Θέλουμε να εμφανίζουμε ένα παράθυρο με επιλογές. Στην Εικόνα 2 περιέχεται ένα παράδειγμα μίας ολοκληρωμένης γραφικής διεπαφής με τη χρήση του XUL.

Για λόγους οικονομίας παραθέτουμε ένα μέρος αυτού του αρχείου, ώστε να δούμε πώς δηλώνονται τα συστατικά στην πράξη.

```

<toolbar id="toolbarid">
  <toolbarbutton id="toolbarbutton-quit" label="Quit"
    image="./skin/quitgray.png" onclick="quit_client()" />
  <toolbarbutton id="toolbarbutton-group" label="Groups"
    image="./skin/groupsgray.png" onclick="open_groups()" />
  <spacer flex="1" />
  <image src="./skin/pithos-gnet-full.png" width="200"
    height="30" />
</toolbar>
<hbox align="center">
  <textbox id="textbox-search" />
  <button id="button-search" label="Search" image="./skin/gss/search_16.png"
    onclick="search_files();" />
  <label id="label-system-message" flex="1" />
</hbox>

```

Ο παραπάνω κώδικας δηλώνει το μενού στην κορυφή του παραθύρου και το πεδίο που χρησιμοποιείται για την αναζήτηση.

Επίλογος

Σε αυτό το άρθρο συνεχίσαμε την εισαγωγή στα Add-On του Firefox. Στόχος μας ήταν να δούμε τη γλώσσα περιγραφής XUL και να καταλάβουμε πώς αυτή λειτουργεί. Στα επόμενα τεύχη θα δούμε αναλυτικά τις τεχνικές δυνατότητες που μας δίνονται για την ανάπτυξή τους και να εμβαθύνουμε σε διάφορα δύσκολα τεχνικά θέματα. Επίσης, θα προχωρήσουμε στη δημιουργία ενός add-on που θα υλοποιεί κάτι συγκεκριμένο (αναμένουμε προτάσεις :).

Χρήσιμα Development Add-ons

- Firebug: <http://goo.gl/ht3co>
- FireQuery: <http://goo.gl/6oWs1>
- DOM Inspector: <http://goo.gl/zDKRm>
- XUL Gear: <http://goo.gl/yvr86>