

Linux Labs – Firefox

Του Βασίλη Καρακώδα <vassilios.karakoidas@gmail.com>



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

Firefox: Ανάπτυξη add-ons (Μέρος I)

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



Τεκμηρίωση για add-ons

Εργαλεία: Firefox

Δυσκολία: ★★☆☆☆

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

Ο Firefox είναι ίσως, μετά το Linux, το πιο διάσημο έργο ανοιχτού λογισμικού. Είναι ίσως από τα λίγα έργα που ανταγωνίστηκαν ευθέως και με επιτυχία αντίστοιχα έργα κλειστού κώδικα, όπως ο Internet Explorer της Microsoft και ο Chrome της Google.

Από τις πρώτες εκδόσεις του, οι δημιουργοί του Firefox, ακολουθώντας σωστές σχεδιαστικές πρακτικές και το πνεύμα του λογισμικού ανοιχτού κώδικα, επέτρεψαν στους χρήστες να αναπτύξουν δικές τους επεκτάσεις και πρόσθετα (add-ons), δίνοντας έτσι τη δυνατότητα σε κάθε χρήστη να χρησιμοποιήσει τον browser με πιο αποδοτικό τρόπο [1].

Πρέπει να είναι πολύπλοκο ...

Συνήθως όταν κάποιος σκέφτεται την ανάπτυξη add-on για κάποια εφαρμογή, φαντάζεται πολύπλοκες προγραμματιστικές διεπαφές και περιβάλλοντα ανάπτυξης που απαιτούν... αποκρυφιστικές γνώσεις προγραμματισμού. Ίσως τις περισσότερες φορές αυτό να ισχύει, σίγουρα όμως, όχι για τον Firefox. Αν εξαιρέσουμε τη βασική γραφειοκρατία για την ενεργοποίηση του add-on, όλα τα υπόλοιπα που χρειάζεται να γνωρίζετε, περιορίζονται στην XUL (XML, δηλαδή) και στην Javascript. Για την ακρίβεια, η ανάπτυξη επεκτάσεων για τον Firefox είναι παρόμοια διαδικασία με τον προγραμματισμό ιστοσελίδων HTML με τη χρήση Javascript.

Εν αρχή...

Στο developer center [2] του Firefox υπάρχει ένα πολύ ωραίο εργαλείο που ονομάζεται add-on Builder [3] και επιτρέπει την εύκολη δημιουργία ενός βασικού add-on, ορίζοντας απλώς τις βασικές προδιαγραφές του. Μόλις εισαγούμε μία σειρά από στοιχεία (εικόνα 1), δημιουργείται αυτόματα ένα zip αρχείο που περιέχει τα βασικά αρχεία του add-on, το οποίο μπορούμε να κατεβάσουμε.

Όσο και αν σας φαίνεται παράξενο, το add-on στην παρούσα μορφή του είναι έτοιμο προς εγκατάσταση, παρ' όλο που, φυσικά, δεν κάνει ακόμα κάτι ουσιαστικό. Αρκεί, λοιπόν, να μετονομάσουμε το αρχείο από linuxinside.zip σε linuxinside.xpi (γνωστό και ως zippy), να το σύρουμε πάνω σε ένα παράθυρο του Firefox και αυτό θα εγκατασταθεί (εικόνα 1).

Αν αποσυμπιέσουμε το linuxinside.zip μας αποκαλύπτονται τα βασικά αρχεία που πρέπει να περιέχονται σε ένα βασικό add-on (εικόνα 2). Σε αυτό το πρώτο άρθρο θα καλύψουμε μόνο τις λειτουργίες των βασικών αρχείων. Στα επόμενα τεύχη θα επεκταθούμε σε πιο εξειδικευμένα θέματα που αφορούν στην ανάπτυξη add-ons.

Το install.rdf είναι ουσιαστικά ένα XML αρχείο που περιέχει τη βασική περιγραφή ενός add-on. Ακολουθεί ένα τυπικό παράδειγμα:

```
<?xml version="1.0" encoding="UTF-8"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:em="http://www.mozilla.org/2004/em-rdf#">
```

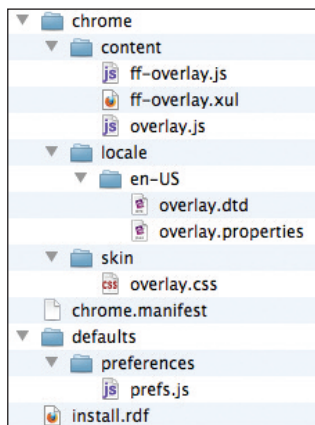


1 Το add-on έχει εγκατασταθεί!

```
<Description about="urn:mozilla:install-manifest">
<em:id>vassilios.karakoidas@gmail.com</em:id>
<em:type>2</em:type>
<em:name>Linux Inside</em:name>
<em:version>0.1</em:version>
<em:creator>Vassilios Karakoidas</em:creator>
<em:contributor></em:contributor>
<em:description>Linux Inside Test
add-on</em:description>
<em:targetApplication>
<Description>
<em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
<em:minVersion>3.5</em:minVersion>
<em:maxVersion>4.0.*</em:maxVersion>
</Description>
</em:targetApplication>
</Description>
</RDF>
```

Τα πιο σημαντικά πεδία του install.rdf είναι:

- **em:id:** Μοναδικό αναγνωριστικό που χρησιμοποιείται για την εφαρμογή. Μπορεί να έχει τη μορφή ενός e-mail (παρ' όλο που δεν χρειάζεται να υπάρχει) ή κάποιο GUID. Προσοχή, το id παραμένει πάντα το ίδιο σε όλες τις εκδόσεις του add-on.
- **em:name:** Το όνομα του add-on
- **em:type:** Για τα Firefox add-ons πρέπει πάντα να έχει την τιμή 2
- **em:version:** Η έκδοση του add-on, σε μορφή [major].[minor], π.χ., 0.1
- **em:creator:** Το όνομα του δημιουργού
- **em:targetApplication:** Περιέχει κατάλογο των εφαρμογών με τις οποίες συνεργάζεται το add-on. Μπορεί να είναι Firefox, Thunderbird και όλες οι άλλες «συγγενείς» εφαρμογές. Στο παράδειγμα μας, το em:id "{ec8030f7-c20a-464f-9b0e-13a3a9e97384}", δηλώνει ότι είναι add-on για τον Firefox, με ελάχιστη συμβατή έκδοση 3.5 και μέγιστη οποιαδήποτε patch έκδοση της 4.0.



2 Η βασική δομή του add-on μας!

Τι είναι το chrome.manifest

Το δεύτερο πιο σημαντικό αρχείο ενός add-on είναι το chrome.manifest.

content	linuxinside	chrome/content/
skin	linuxinside	classic/1.0 chrome/skin/
locale	linuxinside	en-US chrome/locale/en-US/
overlay	chrome://browser/content/browser.xul	
	chrome://linuxinside/content/ff-overlay.xul	

Στην περίπτωση μας έχει τέσσερις γραμμές. Καθεμία δηλώνει στον Firefox πού μπορεί να βρει συγκεκριμένα αρχεία, ώστε να λειτουργήσει το add-on. Συγκεκριμένα, με τον ορισμό του content, ορίζουμε τον κατάλογο όπου βρίσκονται αρχεία γραφικών διεπαφών XUL, Javascript κ.λπ., με το locale υποδεικνύουμε αρχεία που περιέχουν στοιχεία για την υποστήριξη πολλών γλωσσών και με το skin δηλώνουμε τις απαραίτητες πληροφορίες για υποστήριξη πολλαπλών themes, πάντα στο πλαίσιο λειτουργίας του add-on. Θα παρατηρήσετε ότι υπάρχει πάντα το keyword linuxinside στις πρώτες τρεις δηλώσεις. Πρακτικά, αυτό είναι ένα «αναγνωριστικό» που δώσαμε στον add-on builder όταν φτιάχναμε το add-on και χρησιμοποιείται σε αρκετά σημεία στον κώδικα. Συγκεκριμένα, στο chrome.manifest ουσιαστικά δημιουργεί το namespace το οποίο χρησιμοποιείται μετά, για να αποκτηθεί πρόσβαση στα αρχεία του add-on. Το παραπάνω μπορούμε να διαπιστώσουμε από τη δήλωση του overlay. Με απλά λόγια, αυτό που θέλουμε να κάνουμε με αυτή τη δήλωση, είναι να πούμε στον Firefox να βάλει μία επιπλέον επιλογή στο κεντρικό μενού του, ώστε από εκεί να μπορούμε να ξεκινήσουμε το add-on. Συγκεκριμένα, του λέμε, «επέκτεινε το browser.xul με τις επιπλέον λειτουργίες που θα βρεις στο ff-overlay.xul».

Χρησιμοποιώντας το chrome://

Ο Firefox χρησιμοποιεί ένα ειδικό URI για βάση, το οποίο επιτρέπει την προσπέλαση σε όλα τα αρχεία που παρέχει κάθε add-on. Όπως, λοιπόν, τόσο καιρό χρησιμοποιούσατε το http:// για να προσπελάσετε σελίδες στο Internet, έτσι θα μάθετε ότι αν βάλετε το chrome:// αυτομάτως ψάχνετε μέσα στα αρχεία που παρέχονται από το περιβάλλον λειτουργίας του Firefox. Όπως είπαμε και πιο πριν, γράφοντας το linuxinside στο chrome.manifest, δηλώνουμε πρακτικά το ιδεατό μονοπάτι που θα χρησιμοποιούμε για να προσπελάσουμε τα αρχεία του add-on. Οπότε, αν θέλαμε να φορτώσουμε το αρχείο overlay.js που βρίσκεται στο chrome/content/overlay.js, θα πρέπει να χρησιμοποιήσουμε το μονοπάτι chrome://linuxinside/content/overlay.js.

Το αρχείο ff-overlay.xul

Θα κλείσουμε αυτό το εισαγωγικό άρθρο, αναλύοντας το αρχείο ff-overlay.xul:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet h
```

```
ref="chrome://linuxinside/skin/overlay.css"
type="text/css"?>
<!DOCTYPE overlay SYSTEM
"chrome://linuxinside/locale/overlay.dtd">
<overlay id="linuxinside-overlay"
xmlns="http://www.mozilla.org/keymaster/gatekeeper/
there.is.only.xul">
<script src="overlay.js"/>
<script src="ff-overlay.js"/>
<stringbundle id="stringbundleset">
<stringbundle id="linuxinside-strings"
src="chrome://linuxinside/locale/overlay.properties"/>
</stringbundleset>
<menupopup id="menu_ToolsPopup">
<menuitem id="linuxinside-hello"
label="&linuxinside.label;"
oncommand="linuxinside.onMenuItemCommand(event);"/>
</menupopup>
</overlay>
```

Όπως αναφέραμε στην αρχή του άρθρου, τα XUL είναι απλά αρχεία XML. Αν εξαιρέσουμε την αρχική γραφειοκρατία που ορίζει το overlay.css που περιέχει πληροφορίες για το theme (μην ξεχνάμε ότι τα XUL περιγράφουν τη γραφική διεπαφή του add-on κατά κύριο λόγο), μπορούμε να χωρίσουμε το αρχείο σε τρία επιμέρους τμήματα.

Στο πρώτο τμήμα φορτώνουμε αρχεία Javascript, χρησιμοποιώντας το tag <script>. Ο τρόπος χρήσης είναι καθαρά δηλωτικός, οπότε στο παραπάνω μπορούμε να υποθέσουμε ότι θα φορτωθεί πρώτα το overlay.js και αμέσως μετά το ff-overlay.js. Το δεύτερο μέρος έχει σχέση με την υποστήριξη πολλών γλωσσών και αυτό που κάνει, είναι να φορτώνει τη μετάφραση για τη γλώσσα που έχει επιλεγεί. Θα αναλύσουμε πώς δουλεύει το σύστημα των μεταφράσεων σε ένα από τα επόμενα άρθρα. Τέλος, ορίζουμε την προσθήκη στο μενού του Firefox στην κατηγορία «Tools» και δηλώνουμε ότι αν επιλεγεί, θα εκτελεστεί η μέθοδος onMenuItemCommand, η οποία έχει οριστεί στο overlay.js.

Προσοχή: Έχει τεράστια σημασία να έχει φορτωθεί το overlay.js πριν από τη δήλωση του μενού. Αν συμβεί το ανάποδο, δεν θα βρεθεί η συνάρτηση που θα πρέπει να καλεσθεί και υπάρχει η πιθανότητα να προκληθεί σφάλμα.

Επίλογος

Σε αυτό το άρθρο κάναμε μία μικρή εισαγωγή στα add-ons του Firefox. Στόχος μας ήταν να δούμε κάποια από τα δομικά στοιχεία τους και να καταλάβουμε πώς δουλεύουν. Στα επόμενα τεύχη θα δούμε αναλυτικά τις τεχνικές δυνατότητες που μας δίνονται για την ανάπτυξή τους και να εμβαθύνουμε σε διάφορα δύσκολα τεχνικά θέματα.

Σύνδεσμοι:

- [1] Firefox add-on - <https://addons.mozilla.org/>
- [2] Mozilla Development Center - <https://developer.mozilla.org/en/extensions>
- [3] Add-on Builder - <https://addons.mozilla.org/en-US/developers/tools/builder>