



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη plugin σε Mozilla Firefox για αποστολή  
ενός HTTP Timezone Header σε Windows



Του φοιτητή  
Ερπιοιμίδη Γεώργιου  
Αρ. Μητρώου: 01/1721

Επιβλέπων Καθηγητής  
Χρήστος Ηλιούδης

Θεσσαλονίκη 2011

*Στους γονείς και την αδερφή μου  
που με στήριξαν όλα αυτά τα χρόνια.*

*Στους φίλους μου που μου δίνουν  
ένα χαμόγελο ευτυχίας σε δύσκολες στιγμές.*

*Ερπισμιδής Γεώργιος*

# Πρόλογος

---

Στα σύγχρονα υπολογιστικά συστήματα, οι εφαρμογές όπου η επικοινωνία μεταξύ client και server είναι απαραίτητη, γίνονται ολοένα και περισσότερες. Δυστυχώς όμως, η επικοινωνία μέσω διαδικτύου πρέπει να ικανοποιεί ένα γενικό σύνολο, ανεξάρτητα από παράγοντες όπως, γλώσσα, τοποθεσία, λειτουργικό σύστημα, χαρακτηριστικά υλικού ηλεκτρονικού υπολογιστή και ακόμη τόσους, οι οποίοι δημιουργούν εμπόδια για την επίτευξη της επικοινωνίας αυτής.

Ένας από τους παράγοντες αυτούς είναι και η ζώνη ώρας, η οποία στις περισσότερες των περιπτώσεων καθιστά την ακρίβεια των δεδομένων ελλιπή ως προς το θέμα χρόνου.

Υπήρξαν πολλές προσπάθειες για την εξάλειψη του προβλήματος αυτού. Μια συνηθισμένη μέθοδος είναι η είσοδος χρήστη με κάποιο χαρακτηριστικό όνομα μαζί με κωδικό και επιλογή προτιμήσεων για την μετατροπή της ώρας σε τοπική. Ωστόσο, το γεγονός αυτό της διαδικασίας εγγραφής και εισαγωγής χαρακτηριστικών χρήστη είναι από μόνο του χρονοβόρο και απαιτεί χώρο σε βάσεις δεδομένων.

Η πλειονότητα της διαδικτυακής επικοινωνίας γίνεται μέσω ενός προγράμματος περιηγητή ή κοινώς browser. Μία απλή λύση στο πρόβλημα θα ήταν η αποστολή των δεδομένων για τοπικές ρυθμίσεις που αφορούν την ώρα με επιβεβαίωση του χρήστη μία μόνο φορά και όχι σε κάθε είσοδο του σε κάποιο ιστότοπο. Η λύση αυτή είναι μία ιδέα του κ. Χαρχαλάκη Στέφανου, σύμφωνα με μία πρόταση του προς την επιτροπή IEEE.

Πάνω σε αυτήν την ιδέα βασίζεται και η υλοποίηση της πτυχιακής 'Ανάπτυξη plugin για αποστολή Timezone HTTP Header στον Mozilla Firefox σε Windows'.

# Περίληψη

---

Ο περιηγητής ιστοσελίδων Mozilla Firefox είναι ένας από τους περισσότερο διαδεδομένους περιηγητές παγκοσμίως. Το περιβάλλον του περιηγητή αυτού, καθώς συμμορφώνεται με τους κανόνες ανοικτού κώδικα, προσφέρει στον χρήστη τη δυνατότητα να το παραμετροποιήσει και να το βελτιώσει. Η παραμετροποίηση αυτή με τη σειρά της, συμβάλλει στην εξατομίκευση της εφαρμογής, στην ανάπτυξη νέων ιδεών καθώς και στην υλοποίηση λύσεων. Το αποτέλεσμα από μία τέτοια υλοποίηση είναι μία πιο ευχάριστη, περισσότερο άμεση και γενικότερα, πιο βελτιωμένη εμπειρία του χρήστη.

Η πτυχιακή αυτή εργασία, πηγάζει από την ανάγκη για ανάπτυξη ενός plugin που στόχο έχει να καλυτερεύσει την εμπειρία του χρήστη με την εφαρμογή αυτή. Ανάγκη που δημιουργείται από την ανακρίβεια σε χρονικά δεδομένα που συμβαίνουν σε μία επικοινωνία. Η αιτία αυτού, όπως είναι λογικό, είναι η διαφορά ώρας, πρόβλημα πολύ συχνό στο διαδίκτυο καθώς η επικοινωνία είναι μεταξύ υπολογιστών σε διαφορετικές χώρες.

Το αντικείμενο του plugin είναι η εύρεση μίας σωστής και απλής υλοποίησης για την αποστολή πληροφοριών σε μία επικοινωνία η οποία θα γίνεται ακριβής ανεξαρτήτως γεωγραφικού μέρους.

Το κείμενο αυτό θα γνωστοποιήσει στον αναγνώστη τις πληροφορίες που χρειάστηκαν, θα παρουσιάσει τις διαδικασίες που ακολουθήθηκαν, θα αναλύσει την μεθοδολογία και θα εκτιμήσει τα αποτελέσματα από αυτήν την υλοποίηση.

# Abstract

---

The web browser Mozilla Firefox is one of the most commonly used browsers worldwide. The interface of this browser, as it complies with the rules of open source software, offers to the user the ability for configuration and improvement. This configuration, in turn, contributes to the customization of the application, development of new ideas and implementation of solutions. The result of such an implementation is a more pleasant, more direct and generally, more enhanced user experience.

This thesis stems from the need of the development of a plugin which aims to get a better user experience with the application. A necessity created by the inaccuracy of the time data occurring during a communication. The cause of this, quite logically, is the time difference, a very common problem all over the internet, as the communication takes part between computers in different countries.

The object of this plugin is to search for a good and simple implementation that will send information during a communication in order to be accurate, regardless of the geographical location.

The current document will inform the reader of the required elements, will present the procedures followed, will analyze the methodology and assess the results of this implementation.

# Ευρετήριο Περιεχομένων

---

Πρόλογος .....	2
Περίληψη.....	3
Abstract.....	4
Ευρετήριο Περιεχομένων .....	5
Ευρετήριο Εικόνων.....	7
Ευρετήριο Σχημάτων Οδηγού Χρήσης Λογισμικού .....	9
Εισαγωγή .....	10
1. Hypertext Transfer Protocol .....	11
Εισαγωγή.....	11
1.1 Βασικά στοιχεία του HTTP.....	12
1.2 HTTP-Messages.....	15
1.3 HTTP-Headers .....	16
Επίλογος.....	18
2. Timezone HTTP-Header .....	19
Εισαγωγή.....	19
2.1. Η ζώνη ώρας και η θερινή ώρα .....	20
2.2. Το πρόβλημα της διαφοράς στην ζώνη ώρας.....	23
2.3. Περιορισμοί και συνθήκες κατά το Timezone Header .....	24
2.4. Η μορφή του Timezone Header.....	27
Επίλογος.....	30
3. Ο Mozilla Firefox και το περιβάλλον ανάπτυξης.....	31
Εισαγωγή.....	31
3.1 Η μηχανή Gecko.....	32
3.2. Η δομή ενός plugin .....	34

3.3. Απαραίτητα Εργαλεία για την ανάπτυξη plugin.....	36
Επίλογος.....	39
4. Ανάπτυξη του κώδικα για το plugin .....	40
Εισαγωγή.....	40
4.1. Προετοιμάζοντας τον Firefox για ανάπτυξη του plugin.....	41
4.2. Δημιουργώντας τη δομή και το σημείο σύνδεσης με την εφαρμογή.....	43
4.3. Δημιουργώντας τη διεπαφή με το χρήστη.....	45
4.4. Ανάπτυξη του κώδικα για τη λειτουργία του plugin .....	47
4.5 Δοκιμή λειτουργίας και ολοκλήρωση του plugin.....	50
Επίλογος.....	53
Επίλογος .....	54
Συμπεράσματα και προτάσεις.....	55
Βιβλιογραφία .....	56
Παραρτήματα .....	58
Α. Προσχέδιο: “Πληροφορίες ζώνης ώρας στο HTTP” .....	58
Β. Πηγαίος Κώδικας Plugin.....	73
Οδηγός Χρήσης Λογισμικού.....	96
Εγκατάσταση .....	96
Οδηγίες Χρήσης Του Plugin .....	98
Απεγκατάσταση.....	99

# Ευρετήριο Εικόνων

---

Εικόνα 1 Η Δομή Ενός URL Κατά Το HTTP Πρωτόκολλο.....	12
Εικόνα 2 Get Μέθοδος Μέσω Κονσόλας Cygwin .....	14
Εικόνα 3 General Header 'Date'.....	16
Εικόνα 4 Request Header 'Accept'.....	17
Εικόνα 5 Response Header 'Server'.....	17
Εικόνα 6 Entity Header 'Content Type'.....	17
Εικόνα 7 Extension Header 'X-Serial-Number'.....	17
Εικόνα 8 Παγκόσμιος Χάρτης Με Τις Ζώνες Ώρας .....	20
Εικόνα 9 Ολοκληρωμένη Μορφή Ενός Timezone header.....	27
Εικόνα 10 Μορφή Timezone Header Χωρίς Απόκλιση Από UTC .....	28
Εικόνα 11 Συμπαγής Μορφή Ενός Timezone Header .....	28
Εικόνα 12 Τελική Μορφή Ενός Timezone Header.....	29
Εικόνα 13 Τελική Μορφή Ενός Μη Προσδιορισμένου Timezone Header.....	29
Εικόνα 14 Το Λογότυπο Του Mozilla Firefox.....	32
Εικόνα 15 Το Περιβάλλον Του XUL Explorer .....	36
Εικόνα 16 Το Περιβάλλον Του Notepad++.....	37
Εικόνα 17 Το Περιβάλλον Του Live HTTP Headers .....	38
Εικόνα 18 Οι Παράμετροι Στο about:config.....	41
Εικόνα 19 Δημιουργία Νέου Preference.....	41
Εικόνα 20 Πεδίο Νέας Μεταβλητής .....	42
Εικόνα 21 Παράδειγμα Απόλυτης Διαδρομής.....	43
Εικόνα 22 Δομή Φακέλων .....	43
Εικόνα 23 Δείγμα RDF Αρχείου.....	44
Εικόνα 24 Δείγμα Manifest Αρχείου .....	44
Εικόνα 25 Κώδικας XUL Για Τοποθέτηση Menu Plugin .....	45
Εικόνα 26 Μέθοδος Open() Για Αρχικοποίηση Plugin Και Παραμέτρων .....	46
Εικόνα 27 Αρχικοποίηση UI .....	46
Εικόνα 28 Αρχικοποίηση Preferences Κατά Την Εκκίνηση .....	46
Εικόνα 29 Δημιουργία Contract Για Χρήση Component Του Firefox.....	47
Εικόνα 30 Register Component With Browser .....	47



Εικόνα 31 Μέθοδος Observe Για Έλεγχο HTTP Request .....	48
Εικόνα 32 Timezone Μέθοδοι .....	49
Εικόνα 33 Εκσφαλμάτωση Με dump() .....	50
Εικόνα 34 Έξοδος Σε Κονσόλα Του Firefox .....	51
Εικόνα 35 Καταγραφή Headers Με Το Plugin Live HTTP header .....	51
Εικόνα 36 Ορισμός Πεδίου Για Localization .....	52
Εικόνα 37 Localization Μέσω DTD .....	52

# Ευρετήριο Σχημάτων Οδηγού Χρήσης Λογισμικού

---

Σχήμα 1 Άνοιγμα Του Μενού Add-ons .....	96
Σχήμα 2 Οθόνη Με Extensions .....	96
Σχήμα 3 Επιβεβαίωση Εγκατάστασης.....	97
Σχήμα 4 Επιβεβαίωση Επανεκκίνησης .....	97
Σχήμα 5 Άνοιγμα Επιλογών Του Plugin .....	98
Σχήμα 6 Ενεργοποίηση/Απενεργοποίηση Του Plugin.....	98
Σχήμα 7 Απεγκατάσταση Timezone Header .....	99
Σχήμα 8 Επιβεβαίωση Επανεκκίνησης Μετά Την Απεγκατάσταση .....	99

## Εισαγωγή

---

Το διαδίκτυο είναι το πρωταρχικό μέσο επικοινωνίας στη σημερινή εποχή. Ταυτόχρονα αποτελεί και ένα μέσο ενημέρωσης. Η ανάγκη για έγκυρη ενημέρωση είναι μείζον σημασίας, όμως η διαφορά της ώρας μεταξύ χωρών θέτει ένα σημαντικό εμπόδιο στην εγκυρότητα των δεδομένων.

Η εργασία αυτή, μελετά και αναπτύσσει ένα plugin πάνω στο πρόγραμμα περιήγησης Mozilla Firefox και βασίζεται σε ένα προσχέδιο που σκοπό έχει να συμβάλλει στην εγκυρότητα των δεδομένων που έχουν σχέση με το χρόνο. Η δομή του κειμένου είναι με τέτοιο τρόπο ώστε να περιγράψει συνολικά το πρόβλημα.

Αρχικά, θα παρουσιάσουμε την τεχνολογία των πρωτοκόλλων από την οποία διέπεται αυτή η επικοινωνία. Πρωτόκολλα άγνωστα προς τον κοινό χρήστη με θεμελιώδη όμως σημασία για τον τρόπο επικοινωνίας μεταξύ δύο υπολογιστών. Έτσι θα αποκτήσουμε χρήσιμες πληροφορίες που θα βοηθήσουν παρακάτω στην επίλυση του προβλήματος.

Στο επόμενο κεφάλαιο θα επεκταθούμε στο πρόβλημα και θα αναλύσουμε διεξοδικά την αιτία. Επίσης θα ορίσουμε συνθήκες και περιορισμούς έτσι ώστε να κάνουμε μία προσεκτική πρόταση που θα αποτελέσει τον άξονα για να βρούμε την λύση.

Στο τρίτο κεφάλαιο θα έρθουμε σε επαφή με βασικά στοιχεία του Firefox και δομικά στοιχεία των plugin που βασίζονται στο πρόγραμμα αυτό. Ακόμη, θα γνωρίσουμε κάποια βοηθητικά εργαλεία που θα αποτελέσουν το μέσο για την ανάπτυξη της εφαρμογής.

Στο τελευταίο κεφάλαιο θα αναπτύξουμε τον κώδικα και δούμε πως παίρνει μορφή. Τέλος, θα συνδέσουμε τις λειτουργίες, σύμφωνα με τη λύση που προτείναμε και θα αναλύσουμε όλα τα σημεία που είναι κλειδιά για την ολοκλήρωση του plugin.

# 1. Hypertext Transfer Protocol

---

## Εισαγωγή

Στο ακόλουθο κεφάλαιο περιγράφεται το πρωτόκολλο πάνω στο οποίο βρίσκονται τα θεμέλια της πτυχιακής. Θα γνωρίσουμε πώς πραγματικά είναι η δομή μιας διαδικασίας, που ενώ είναι κυρίαρχο στοιχείο στην επικοινωνία μέσα στον παγκόσμιο ιστό, πολλές φορές αγνοείται.

Ξεκινώντας με την βασική ιδέα του πρωτοκόλλου αυτού θα έρθουμε σε επαφή με όλα εκείνα τα στοιχεία που μας είναι απαραίτητα για την υλοποίηση του plugin.

Η πρώτη ενότητα αναφέρει βασικά στοιχεία του HTTP, όπως το URL και τις μεθόδους με τις οποίες στέλνονται, όμως είναι διαφανή προς τον χρήστη.

Η ενότητα που ακολουθεί, παρουσιάζει τους διάφορους τύπους μηνυμάτων καθώς και τα πεδία που τα απαρτίζουν.

Η τρίτη ενότητα περιγράφει και αναλύει τα headers, όρος του πρωτοκόλλου τον οποίο θα χρησιμοποιήσουμε ως μέσο για την λειτουργία του plugin.

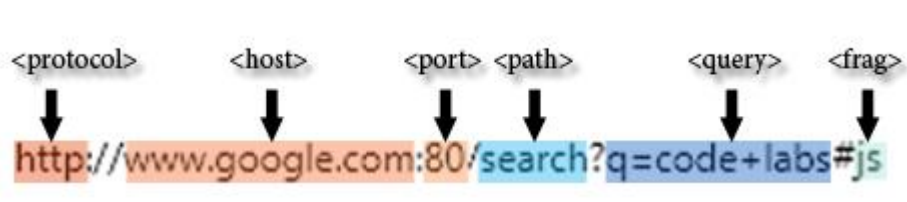
Το αποτέλεσμα από την ανάλυση αυτή θα είναι ένα πολύτιμο σύνολο από πληροφορίες, πάνω στις οποίες θα βασιστεί η εφαρμογή μας.

## 1.1 Βασικά στοιχεία του HTTP

Το Hypertext Transfer Protocol, ή κοινώς HTTP, είναι ένα πλαίσιο από κανόνες, ένα πρωτόκολλο που σκοπό του έχει να καθορίσει τον τρόπο επικοινωνίας μεταξύ δύο ή και περισσότερων υπολογιστών. Η εκδοχή του, έκδοση 1.1, που χρησιμοποιείται έως και σήμερα, προτάθηκε τον Ιούνιο του 1999. Την πρόταση για το σχεδιασμό κανόνων επικοινωνίας έκαναν διάφοροι παράγοντες όπως ιδιώτες, πανεπιστήμια και επιχειρήσεις με αντικείμενο εξειδίκευσης την πληροφορική. Ο συντονισμός ήταν υπό την εποπτεία των οργανισμών World Wide Web Consortium (W3C) και Internet Engineering Task Force (IETF), υπεύθυνων για την δημιουργία και προώθηση ανάλογων προτύπων που έχουν στόχο την ομαλή επικοινωνία.

Η λογική του προτύπου αυτού βασίζεται στη διαδικασία αίτησης-απάντησης μεταξύ δύο υπολογιστών, συγκεκριμένα, μεταξύ client και server. Ο τρόπος είναι πολύ απλός: ο client κάνει μία αίτηση με τη μορφή URL, στέλνοντας μηνύματα που περιέχουν πληροφορίες για τον ίδιο και προσδιορίζουν το περιεχόμενο και τον τρόπο με τον οποίο θέλει να το λάβει. Με τη σειρά του, ο υπολογιστής που έχει το ρόλο του server λαμβάνει το μήνυμα, το επεξεργάζεται και απαντάει με μήνυμα που αναφέρει την κατάσταση με συγκεκριμένους κωδικούς, πληροφορίες για τον ίδιο και αναλόγως τον κωδικό κατάστασης, το περιεχόμενο.

Το Uniform Resource Locator (συντμ. URL), είναι μία ηλεκτρονική διεύθυνση που προσδιορίζει μοναδικά κάποιο περιεχόμενο σε ένα server στο διαδίκτυο. Η ηλεκτρονική αυτή διεύθυνση εισάγεται σε ένα πρόγραμμα περιηγητή (αγγλ. Browser) και πρέπει να έχει συγκεκριμένη δομή. Η δομή αυτής όταν πρόκειται για το πρωτόκολλο HTTP είναι η εξής: `<protocol>://<host>:<port>/<path>?<query>#<frag>` όπως παρουσιάζεται αναλυτικά και στην Εικόνα 1.



Εικόνα 1 Η Δομή Ενός URL Κατά Το HTTP Πρωτόκολλο.

Όπως παρατηρούμε, το κομμάτι <protocol> προσδιορίζει το πρωτόκολλο που χρησιμοποιείται μέσα από ένα μεγάλο πλήθος επιλογών, όπως FTP, RTSP, κ.ά. όπου το κάθε ένα έχει διαφορετικό λόγο χρήσης.

Το <host> διαχωρίζεται με το ‘://’ και είναι το όνομα που προσδιορίζει μοναδικά κάποιον server, ο οποίος φυσικά αντιστοιχεί σε μία μοναδική διεύθυνση IP στο διαδίκτυο. Να σημειωθεί εδώ ότι το <host> κατά πολλούς διαχωρίζεται σε <host> και <domain> (Charman, et al., 2005), όμως στη συγκεκριμένη περίπτωση περιλαμβάνει και τους δύο όρους.

Το <port> που ακολουθεί το σύμβολο ‘:’ είναι ο αριθμός θύρας για το συγκεκριμένο URL και για το HTTP, όταν δεν διευκρινίζεται συγκεκριμένα, είναι το 80.

Το <path> είναι το όνομα της θέσης στον server που περιέχει το περιεχόμενο που αναζητούμε κατά την επικοινωνία.

Το <query> είναι ένας τρόπος για να περάσουμε διάφορες παραμέτρους σε μια σελίδα ή μία online εφαρμογή.

Το τελευταίο κομμάτι του URL είναι το <frag>, το οποίο δείχνει σε ένα συγκεκριμένο μέρος του περιεχομένου, κάτι που δεν μπορεί να προσδιορίσει το <path>.

Κατά την εισαγωγή του URL ακολουθείται κάποια μέθοδος που δίνει οδηγίες στον server για τις ενέργειες που θα κάνει.

- Η μέθοδος ‘GET’ μεταβιβάζει το όνομα του περιεχομένου από τον server στον client.
- Η ‘PUT’ δίνει εντολή για αποθήκευση δεδομένων από τον client μία απομακρυσμένη διεύθυνση στον server (Εικόνα 2).
- Η ‘DELETE’ διαγράφει το περιεχόμενο που αντιστοιχεί σε ένα URL από τον server.
- Η ‘POST’ στέλνει δεδομένα από τον client στο server τα οποία απαιτεί κάποια εφαρμογή που του περιεχομένου που αναζητήσαμε.
- Η ‘HEAD’ στέλνει τα μόνο τα Headers από την απάντηση του διακομιστή κατά την αποστολή του URL.
- Η ‘TRACE’ μέθοδος δίνει την δυνατότητα σε έναν client να μπορεί να ανιχνεύσει το αίτημά του, σε περίπτωση που αλλάξουν την αρχική του μορφή τρίτοι

servers ή άλλες εφαρμογές, όπως proxy ή firewalls. Όταν φτάσει στον τελικό server προορισμό, θα μπορεί να γνωρίζει το αίτημά του σε αυτόν.

- Η 'Options' απευθύνεται στον server και ενημερώνει τον client για τις δυνατότητες που διαθέτει.
- Η 'Connect' είναι μία μέθοδος που μπορεί να χρησιμοποιηθεί σε συνδυασμό με ένα proxy server για δυναμική αλλαγή αυτού σε λειτουργία tunnel, που είναι μία τεχνική ενθυλάκωσης διαφόρων άλλων πρωτοκόλλων στο HTTP (π.χ. SSL tunneling).

```
$ wget -S http://www.anandtech.com
--2011-09-06 02:00:13-- http://www.anandtech.com/
Resolving www.anandtech.com (www.anandtech.com)... 208.65.201.105
Connecting to www.anandtech.com (www.anandtech.com):208.65.201.105:80... connecte
d.
HTTP request sent, awaiting response...
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNetMvc-Version: 2.0
X-AspNet-Version: 2.0.50727
Set-Cookie: atusessionw=92ab69b1-5cdc-4fff-b45c-c93df2aad281; domain=.anandtec
h.com; expires=Wed, 01-Jan-2020 05:00:00 GMT; path=/
X-Powered-By: ASP.NET
Date: Mon, 05 Sep 2011 23:00:12 GMT
Connection: keep-alive
Content-Length: 68431
Length: 68431 (67K) [text/html]
Saving to: `index.html.4'

100%[=====>] 68,431      105K/s   in 0.6s
2011-09-06 02:00:14 (105 KB/s) - `index.html.4' saved [68431/68431]
```

Εικόνα 2 Get Μέθοδος Μέσω Κοινότητας Cygwin

## 1.2 HTTP-Messages

Σύμφωνα με το πρωτόκολλο HTTP, για να εξασφαλιστεί μία επικοινωνία μεταξύ δύο υπολογιστών, τα τερματικά θα πρέπει να στέλνουν και να λαμβάνουν μηνύματα.

Τα μηνύματα αυτά είναι κοινώς γνωστά σαν Request/Response messages. Όπως είναι λογικό, για να είναι επιτυχής μία τέτοια επικοινωνία πρέπει να υπάρχει και ένα πρότυπο κατά την αποστολή τους. Έτσι, σύμφωνα με το προσχέδιο RFC 2616, ένα τέτοιο μήνυμα αποτελείται από, ουσιαστικά, 5 πεδία.

Τα πεδία αυτά είναι ο τύπος του μηνύματος, κεφαλίδες μηνύματος (εφεξής headers), το σώμα του μηνύματος, το μήκος κειμένου και άλλα γενικά πεδία κεφαλίδων. Ο σχηματισμός τους θα πρέπει να ξεκινάει με μια γραμμή εκκίνησης στην οποία δηλώνεται το πρωτόκολλο που θα χρησιμοποιηθεί, κανένα ή περισσότερα headers, μία κενή γραμμή που δηλώνει το τέλος των headers και το σώμα του μηνύματος.

Ο τύπος του μηνύματος μπορεί να είναι είτε request ή response, ενώ τα headers περιέχουν όλες εκείνες τις πληροφορίες που καθορίζουν το περιβάλλον κατά το οποίο θα γίνει η μετάδοση. Ωστόσο, οι παράμετροι που μπορεί να πάρουν είναι διαφορετικοί για κάθε τύπο μηνύματος.

Το σώμα του μηνύματος είναι όλα εκείνα τα μηνύματα που θα αποσταλούν προσδιορισμένα από ένα URL και αποτελούν ένα σύνολο. Το περιεχόμενο του εξαρτάται από τα headers, καθώς και τους κωδικούς κατάστασης που στέλνονται κατά την απάντηση. Για παράδειγμα, ο κωδικός κατάστασης “100” που σύμφωνα με το πρωτόκολλο μεταφράζεται σαν “CONTINUE”, δηλαδή, ο client να συνεχίσει με τη αποστολή της αίτησης, δεν μπορεί να περιέχει το σώμα του μηνύματος.

Το μήκος του μηνύματος προσδιορίζει το μήκος του σώματος που θα σταλεί και θα πρέπει να το επαληθεύει.

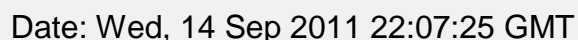
Το τελευταίο κομμάτι του μηνύματος, τα γενικά πεδία κεφαλίδων, δεν είναι τίποτα άλλο από τα προαναφερθέντα headers που όμως δεν προσδιορίζουν το σύνολο του μηνύματος. Αυτά τα headers δεν ακολουθούν, απαραίτητα, το HTTP/1.1 πρωτόκολλο και προσδιορίζουν τα επιμέρους μηνύματα (Fielding, et al., 1999).



## 1.3 HTTP-Headers

Τα headers λειτουργούν συνδυαστικά μαζί με τις μεθόδους για να δώσουν πληροφορίες για τις ενέργειες που θα ακολουθήσουν ο client ή ο server. Πρακτικά, ορίζουν παραμέτρους για τη λειτουργία μίας HTTP σύνδεσης. Υπάρχουν headers που είναι ειδικά για κάθε τύπο μηνύματος και headers που έχουν γενικότερο σκοπό, παρέχοντας πληροφορίες τόσο σε μηνύματα αίτησης όπως και απάντησης. Τα headers αποτελούνται από πέντε κατηγορίες (Gourley & Totty, 2002).

Τα γενικά πεδία κεφαλίδων, τα οποία χρησιμοποιούνται και από τον client και από τον server. Εξυπηρετούν γενικούς σκοπούς και είναι χρήσιμες και στις δύο πλευρές καθώς και σε άλλες εφαρμογές που υποστηρίζουν η μία την άλλη. Για παράδειγμα, το header 'Date' (Εικόνα 3) είναι ένα γενικής χρήσης header που επιτρέπει στις δύο πλευρές να δείχνουν την ώρα και την ημερομηνία κατά την οποία το μήνυμα κατασκευάστηκε.



```
Date: Wed, 14 Sep 2011 22:07:25 GMT
```

*Εικόνα 3 General Header 'Date'.*

Τα request headers, όπως είναι αυτονόητο, είναι headers τα οποία στέλνονται συγκεκριμένα κατά το μήνυμα τύπου request. Παρέχουν επιπλέον πληροφορίες στον server, όπως για παράδειγμα το είδος που ο client είναι διατεθειμένος να λάβει. Ένα συχνό τέτοιο header είναι το 'Accept' (Εικόνα 4). Στην συγκεκριμένη περίπτωση δηλώνεται το τον τύπο περιεχομένου που μπορεί να δεχτεί ο client. Το γράμμα 'q' είναι μία παράμετρος που εκφράζει τον βαθμό προτεραιότητας του περιεχομένου και ακολουθεί τον τύπο του διαχωρισμένο από το σύμβολο ':'. Το μέγιστο είναι '1' και το ελάχιστο '0'. Δηλαδή, στο παρακάτω σχήμα δηλώνει ο client ότι θέλει κατά προτεραιότητα 'text/html' και εφόσον δεν ακολουθεί το 'q', υπονοείται το 'q=1'. Αν δεν υπάρχει τέτοιος τύπος, η επόμενη περίπτωση είναι να στείλει περιεχόμενο τύπου 'application/xhtml+xml'. Αν δεν είναι διαθέσιμο τότε θα στείλει κατά προτίμηση 'application/xml' περιεχόμενο με προτεραιότητα '0.9' και τέλος '\*/\*', δηλαδή οτιδήποτε άλλο περιεχόμενο με προτεραιότητα '0.8'.

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

*Εικόνα 4 Request Header 'Accept'.*

Αντίστοιχα, ένα response header, θα σταλεί μόνο όταν το μήνυμα είναι τύπου response και θα δώσει πληροφορίες στον client. Στο ακόλουθο σχήμα (Εικόνα 5), ο server ενημερώνει τον client ότι απευθύνεται σε ένα server τύπου 'gws', που δεν είναι άλλος από τον Google Web Server.

```
Server: gws
```

*Εικόνα 5 Response Header 'Server'.*

Τα entity headers αναφέρονται σε headers που ασχολούνται με το σύνολο των μηνυμάτων. Για παράδειγμα, το ακόλουθο header (Εικόνα 6) φανερώνει τον τύπο του αντικειμένου που υπάρχει στο μήνυμα. Συγκεκριμένα, ένας 'text/html' τύπος με κωδικοποίηση χαρακτήρων 'UTF-8'.

```
Content-Type: text/html; charset=UTF-8
```

*Εικόνα 6 Entity Header 'Content Type'.*

Τέλος, τα extension headers είναι headers που δεν υπακούουν στο HTTP πρωτόκολλο και έχουν αναπτυχθεί από τρίτους προγραμματιστές. Ωστόσο, οποιαδήποτε εφαρμογή χρησιμοποιεί το πρωτόκολλο αυτό, οφείλει να τους στέλνει ακόμη και αν δεν είναι εφικτό να κατανοεί τη λειτουργία τους. Στο ακόλουθο παράδειγμα (Εικόνα 7), μία εφαρμογή στέλνει έναν extension header που ονομάζεται 'X-Serial-Number' και ενημερώνει τον αποδέκτη για τον σειριακό αριθμό της εφαρμογής.

```
X-Serial-Number: 0123456789
```

*Εικόνα 7 Extension Header 'X-Serial-Number'.*

## Επίλογος

Στο κεφάλαιο αυτό, είδαμε βασικά στοιχεία του πρωτόκολλου HTTP, το URL και τις μεθόδους με τις οποίες στέλνονται. Ακόμη, αναλύσαμε τα headers που λειτουργούν συνδυαστικά με τις μεθόδους. Αυτή η ανάλυση μας βοηθάει να ορίσουμε και να εστιάσουμε καλύτερα στο 'που'. Έχοντας αυτή τη γνώση μπορούμε να καταλάβουμε σε ποιο σημείο θα μπορούσαμε να μεταβάλλουμε στοιχεία ικανά να μας δώσουν τη λύση.

Στο επόμενο κεφάλαιο θα δούμε το πρόβλημα περισσότερο διεξοδικά, παρουσιάζοντας περιεχόμενο που αναλύει τις αιτίες εξαιτίας των οποίων δημιουργείται το πρόβλημα. Επίσης, θα θέσουμε ένα πλαίσιο που θα λειτουργεί σαν οδηγός για τον τρόπο με τον οποίο μεταβάλλονται τα παραπάνω στοιχεία.

## 2. Timezone HTTP-Header

---

### Εισαγωγή

Η γεωγραφική τοποθεσία των χωρών και οι ανάγκες των πολιτών σε κάθε χώρα έχουν επιφέρει σημαντικές αλλαγές που έχουν στόχο να καλυτερεύσουν τις συνθήκες σε όφελος των πολιτών. Ωστόσο, αυτό συνεπάγεται επιπλοκές στην επικοινωνία μεταξύ υπολογιστικών συστημάτων. Στο ακόλουθο κεφάλαιο θα δούμε την αιτία από όπου πηγάζουν αυτές οι επιπλοκές.

Στην πρώτη ενότητα θα δούμε την λογική με την οποία χωρίζονται οι ζώνες ώρας, ενώ παράλληλα θα αναλύσουμε και την λειτουργία της θερινής ώρας. Επίσης, θα αναφέρουμε κάποιους λόγους για τους οποίους έχουν θεσπιστεί αυτοί οι όροι.

Συνεχίζοντας, θα παρουσιάσουμε το πρόβλημα, το πού συμβαίνει και το πώς επηρεάζει την επικοινωνία.

Στην τρίτη ενότητα, θα θέσουμε τους περιορισμούς και τις συνθήκες στα οποία πρέπει να υπακούει ένα Timezone Header για την λύση των επιπλοκών που δημιουργούνται.

Βασιζόμενοι στα παραπάνω, η τέταρτη ενότητα θα θέσει την μορφή και την δομή η οποία θα πρέπει να υπάρχει κατά την αποστολή του header, για μία περισσότερο αποδοτική και έγκυρη επικοινωνία μεταξύ υπολογιστών.

## 2.1. Η ζώνη ώρας και η θερινή ώρα

Όλες σχεδόν οι χρονικές ζώνες στη γη έχουν συγκεκριμένα νομικά σύνορα τα οποία συμπίπτουν με τα σύνορα της χώρας, επιβάλλοντας την ώρα ή κάποια υποδιαίρεση αυτής. Από τις 40 διαφορετικές ζώνες ώρας στην ξηρά, οι περισσότερες είναι αποκλίσεις από την Συντονισμένη Παγκόσμια Ώρα (αγγ. Universal Coordinated Time ή UTC) με έναν ακέραιο αριθμό των ωρών (UTC-12 έως UTC +14) . Βέβαια, υπάρχουν και κάποιες οι οποίες έχουν απόκλιση 30 ή 45 λεπτά από μία γειτονική ζώνη ώρας.

Εκτός από τις ζώνες ώρας στην ξηρά, υπάρχουν 25 ναυτικά προσδιορισμένες ζώνες ώρας, που χωρίζονται από τις γραμμές του γεωγραφικού μήκους. Οι περισσότερες ζώνες με τιμές από UTC-11 έως UTC+11 απέχουν μεταξύ τους 15° γεωγραφικού μήκους (Εικόνα 8). Αυτή η τιμή αντιστοιχεί σε μία ώρα της περιστροφής της Γης σε σχέση με τον Ήλιο, αλλά σε ζώνη ώρας του κεντρικού Ειρηνικού Ωκεανού έχει διαιρεθεί σε δύο 7,5 ° ευρύ ζώνες (UTC ± 12 ) από το 180ο μεσημβρινού, μέρος του οποίου συμπίπτει με τη διεθνή γραμμή ημερομηνίας. Πολλές χρονικές ζώνες ξηράς τείνουν προς τα δυτικά σε σχέση με τις αντίστοιχες ναυτικές ζώνες ώρας.



Εικόνα 8 Παγκόσμιος Χάρτης Με Τις Ζώνες Ώρας

Οι ζώνες ώρας γράφονται ως μετατόπιση από την UTC με τη μορφή  $\pm [\Omega\Omega]:[\Lambda\Lambda]$ ,  $\pm [\Omega\Omega][\Lambda\Lambda]$ , ή  $\pm [\Omega\Omega]$ . Το σύμβολο 'ΩΩ' ισούται με την ώρα στην αντίστοιχη θέση σε διψήφιο νούμερο. Παρομοίως το σύμβολο 'ΛΛ' ισούται με λεπτά στην αντίστοιχη θέση σε διψήφιο νούμερο. Έτσι, αν η ώρα που περιγράφουμε είναι δύο ώρες μπροστά από την UTC (όπως συμβαίνει στην Ελλάδα κατά τη διάρκεια του χειμώνα), η ένδειξη ζώνης θα είναι "+02:00", "0200", ή απλά "02". Το πεδίο αυτό προσαρτάται μετά την ώρα.

Για παράδειγμα, ας υποθέσουμε ότι η ώρα στην Αγγλία αυτή τη στιγμή είναι '13:45'. Εφόσον, δεν έχει απόκλιση από το UTC η ώρα της Αγγλίας είναι '13:45Z', όπου το 'Z' απεικονίζει την μηδενική (αγγλ. Zero) απόκλιση σε σχέση με το UTC. Σύμφωνα με αυτά για την δεδομένη στιγμή στην Ελλάδα που έχει απόκλιση "+02:00" θα λέγαμε ότι η ώρα είναι "15:45+02:00", αυτό πάντα κατά την διάρκεια του χειμώνα όπου η αλλαγή σε θερινή ώρα δεν συμβαίνει.

Πολλές χώρες όμως, έχουν υιοθετήσει τον όρο 'Θερινή Ώρα' (αγγλ. DST). Ο όρος αυτός είναι η πρακτική της προσωρινής προώθησης στα ρολόγια κατά τη διάρκεια του καλοκαιριού, έτσι ώστε το απόγευμα να έχουν περισσότερο φως της ημέρας, μειώνοντας αυτό του πρωινού. Συνήθως τα ρολόγια προσαρμόζονται προς τα εμπρός μια ώρα κοντά στην αρχή της άνοιξης και προσαρμόζονται προς τα πίσω το φθινόπωρο. Στην Ελλάδα, για παράδειγμα, την Κυριακή 27 Μαρτίου του 2011 στις 3:00π.μ. οι δείκτες των ρολογιών μετακινήθηκαν μία ώρα μπροστά δείχνοντας 4:00μ.μ. . Αντίστροφα, την τελευταία Κυριακή του Οκτωβρίου του 2011 στις 4:00π.μ. οι δείκτες των ρολογιών θα μετακινηθούν μία ώρα πίσω δείχνοντας 3:00μ.μ. .

Η σύγχρονη μορφή του DST προτάθηκε για πρώτη φορά το 1895 από τον 'George Vernon Hudson' (Gibbs, 2007). Ενώ πολλές χώρες το χρησιμοποιούν από τότε, υπάρχουν πάντα αλλαγές περιστασιακά ενώ πολλές λεπτομέρειες διαφέρουν ανάλογα με τη θέση.

Η πρακτική αυτή επαινέθηκε και επικρίθηκε (Downing, 2005). Τα υπέρ και τα κατά αντικρούουν το ένα το άλλο. Προσθέτοντας το φως της ημέρας τα απογεύματα ωφελείται το λιανικό εμπόριο, ο αθλητισμός, και άλλες δραστηριότητες που εκμεταλλεύονται το φως του ήλιου μετά από τις ώρες εργασίας (Benfield, 2001), αλλά

δημιουργεί προβλήματα για την κτηνοτροφία, βραδινή διασκέδαση και άλλα επαγγέλματα που συνδέονται με τον ήλιο.

Η αλλαγή του ρολογιού κατά το DST παρουσιάζει και άλλες προκλήσεις. Μπορεί να περιπλέξει χρονομετρήσεις, να διαταράξει συναντήσεις, τα ταξίδια, τη χρέωση, τήρηση αρχείων, τα ιατρικά προϊόντα τεχνολογίας, βαρύ εξοπλισμό και τις συνήθειες ύπνου. Το λογισμικό μπορεί συχνά να ρυθμίσει αυτόματα ρολόγια υπολογιστή, αλλά αυτό είναι μία ενέργεια περιορισμένη και επιρρεπής σε λάθη, ιδιαίτερα όταν αλλάζουν τα DST πρωτόκολλα. Ωστόσο, η επίδρασή του στην υγεία και την εγκληματικότητα είναι λιγότερο σαφής (Downing, 2005).

Παρά το γεγονός ότι ένας γρήγορος στόχος της θερινής ώρας ήταν να μειώσει τη χρήση των λαμπτήρων πυράκτωσης το βράδυ, τα πρότυπα χρήσης στη σύγχρονη θέρμανση και ψύξη διαφέρουν σημαντικά. Οι έρευνες για το πώς το DST πλήττει σήμερα τη χρήση της ενέργειας είναι περιορισμένες, ελλείψεις και συχνά αντιφατικές (Aries & Newsham, 2008).

## 2.2. Το πρόβλημα της διαφοράς στην ζώνη ώρας

Το διαδίκτυο είναι ένα σύνολο από υπολογιστές συνδεδεμένους μεταξύ τους σε παγκόσμιο επίπεδο. Ανεξάρτητα από το περιεχόμενο της κάθε ιστοσελίδας, αποθηκεύονται δεδομένα χρονικής σήμανσης που σκοπό έχουν να πληροφορήσουν τον επισκέπτη για την δημιουργία κάποιου σχόλιου ή γενικότερα καινούργιου περιεχομένου. Οι σημάνσεις αυτές, ενώ σε τοπικό επίπεδο, δηλαδή στην μεριά του server όπου βρίσκεται το περιεχόμενο, είναι σωστές, δεν εξασφαλίζουν στον χρήστη την ακρίβεια της χρονικής στιγμής. Ο λόγος είναι η διαφορά ώρας που υπάρχει μεταξύ δύο χωρών εξαιτίας της ζώνης ώρας.

Πολλές διαδικτυακές εφαρμογές θα μπορούσαν να ωφεληθούν από τη γνώση της ζώνης ώρας των πελατών τους. Οι περισσότερες από αυτές που παρέχουν δυναμικά περιεχόμενο, εξαρτώνται από τους λογαριασμούς χρηστών για να εμφανίζεται η ώρα και η ημερομηνία προσαρμοσμένη στη ζώνη ώρας του πελάτη. Ακόμα και αυτό δεν είναι πάντα επαρκές εφόσον υπάρχουν άτομα που μπορούν να ταξιδέψουν πέρα από αυτά τα όρια της ζώνης, όπου και υφίσταται η ανάγκη για να ενημερώσουν τους λογαριασμούς τους στο διαδίκτυο έτσι ώστε οι πληροφορίες να αντανakλούν την πραγματική τους ζώνη ώρας. Επιπλέον, αυτό ισχύει για όλα τα παρεμφερή με HTTP όπως και το ίδιο HTTP πρωτόκολλα, που κάνουν χρήση των χρονικών σημάνσεων (αγγ. timestamps).

Το πρόβλημα αυτό μπορεί να επιλυθεί με την περιγραφή ενός header που θα χρησιμοποιηθεί κατά το HTTP πρωτόκολλο έτσι ώστε οι ενδιαφερόμενοι πελάτες να μπορούν να παρέχουν τις τρέχουσες πληροφορίες ζώνη ώρας σε διαδικτυακούς διακομιστές και κατ' επέκταση σε διαδικτυακές εφαρμογές.



## 2.3. Περιορισμοί και συνθήκες κατά το Timezone Header

Πριν να δημιουργήσουμε Timezone Header θα πρέπει να θέσουμε τα όρια στα οποία θα εφαρμόζεται. Επίσης είναι απαραίτητο να ορίσουμε το περιβάλλον στο οποίο θα λειτουργεί. Το κείμενο αυτό συμβαδίζει με το 'Παράρτημα Α', όπου και βρίσκεται το προσχέδιο για την υλοποίηση.

Το Timezone Header θα πρέπει να ισχύει για κάθε HTTP client. Κάθε πελάτης του πρωτοκόλλου HTTP ο οποίος, συνήθως, είναι και ένας περιηγητής ιστοσελίδων.

Το Timezone Header θα πρέπει να είναι μία συμβολοσειρά (αγγλ. String). Η συμβολοσειρά αυτή θα περιέχει την μορφή της ζώνης ώρας, όπως θα περιγράψουμε παρακάτω.

Το Timezone Header θα πρέπει να είναι ένα HTTP header. Ως εκ τούτου, θα πρέπει να υπακούει στους όρους που ορίζεται για ένα HTTP header κατά το πρωτόκολλο HTTP/1.1. (Fielding, et al., 1999).

Λόγω της ποικιλίας των συστημάτων στο Διαδίκτυο και του μη τεχνικού χαρακτήρα της ζώνης ώρας, δεν υπάρχει απλή μέθοδος για τον πελάτη να παρέχει πληροφορίες ζώνης ώρας σε διακομιστές HTTP. Η υλοποίηση θα πρέπει να λάβει υπόψη της, τα εξής:

- Η απλότητα είναι απαραίτητη. Οι προδιαγραφές πρέπει να είναι όσο το δυνατό απλούστερες ή να επιτρέπουν τον χειρισμό με απλό τρόπο.
- Υπάρχει μια ποικιλία των μορφών ζώνης ώρας. Ορισμένες χώρες δεν υιοθετούν την θερινή ώρα (DST). Άλλες χώρες έχουν πολύ ασταθή DST.
- Η πολιτική επηρεάζει το DST. Αυτό το καθιστά ένα κινούμενο στόχο.
- Οι ρυθμίσεις θερινής ώρας ενδέχεται να αλλάζουν από χρόνο σε χρόνο.
- Το πλήρες παρελθόν και το μέλλον στις πληροφορίες ζώνης ώρας δεν μπορεί να χαρακτηριστεί χρησιμοποιώντας μία απλή συμβολοσειρά.
- Συστήματα που απευθύνονται σε τελικούς χρήστες μπορεί να έχουν ρυθμισμένη λάθος ζώνη ώρας.
- Υπάρχουν συστήματα που δεν διαθέτουν πληροφορίες για ζώνη ώρας.
- Δεν έχουν όλα τα συστήματα πρόσβαση σε χρονικούς πόρους.

Αυτή η υλοποίηση προσπαθεί να παρέχει επαρκή δεδομένα για εφαρμογές που θα επωφεληθούν από τις πληροφορίες ζώνης ώρας για να εκπληρώσουν τις ανάγκες τους. Υπάρχουν διαφορετικά επίπεδα απαιτήσεων από την κάθε εφαρμογή και απαιτήσεων που μπορεί να ικανοποιήσει ο HTTP πελάτης. Έτσι, αυτή η πρόταση είναι ένας τρόπος για τους πελάτες HTTP ώστε να παρέχει:

- Την τρέχουσα ώρα πελάτη.
- Την τρέχουσα απόκλιση ώρας.
- Τις τρέχουσες πληροφορίες ώρας ζώνης του τρέχοντος έτους.

Πλήρεις πληροφορίες ζώνης ώρας ανάλογα με τις δυνατότητές τους.

Απλές εφαρμογές server μπορούν να χρησιμοποιήσουν μόνο την τρέχουσα χρονική απόκλιση βασιζόμενοι ότι είναι σωστά ρυθμισμένο. Πιο προηγμένες εφαρμογές θα πρέπει να γνωρίζουν πότε μια αλλαγή θερινής ώρας θα προκύψει για τη σωστή αναπαράσταση μελλοντικών η παρελθοντικών χρόνων. Ακόμη πιο προηγμένες εφαρμογές (όπως τα ημερολόγια) θα πρέπει να γνωρίζουν την ακριβή ζώνη ώρας του πελάτη, η οποία μπορεί να περιγραφεί μόνο χρησιμοποιώντας μια αναφορά σε μια εισαγωγή στη βάση δεδομένων ζώνης ώρας. Τέλος, εφαρμογές μπορούν να επιλέξουν να χρησιμοποιήσουν την τρέχουσα ώρα του πελάτη ως σημείο χρονικής αναφοράς και προαιρετικά να αγνοήσουν τις πληροφορίες για την ζώνη ώρας του πελάτη.

Σύμφωνα με την παράγραφο 4.4 του [RFC3339] "Απορριφθείσα Τοπική Ώρα", συστήματα με μη έγκυρες πληροφορίες ζώνης ώρας θεωρούνται ως ακατάλληλα για το διαδικτυακές επικοινωνίες (Klyne & Newman, 2002).

Οι συμβατοί διακομιστές μπορούν να επιβεβαιώσουν την μορφή πληροφορίας της παρεχόμενης ζώνης ώρας. Συμβολοσειρές ζώνης ώρας που δεν είναι σε έγκυρη μορφή είναι δυνατόν να απορριφτούν. Οι έλεγχοι δεν θα πρέπει να πραγματοποιούνται στο περιεχόμενο της ζώνης ώρας από τους διακομιστές. Οι διακομιστές θα πρέπει να ελέγχουν μόνο τη μορφή της συμβολοσειράς. Με αυτόν τον τρόπο, μη ενημερωμένοι διακομιστές δεν θα απορρίπτουν κατάλληλες πληροφορίες.

Τέλος, οι HTTP-proxy διακομιστές, δεν θα πρέπει να αλλάζουν το περιεχόμενο της πληροφορίας αυτής. Διακομιστές με κώδικα στην πλευρά τους ο οποίος παράγει εξατομικευμένα αποτελέσματα, βασισμένα στην πληροφορία ζώνης ώρας, θα πρέπει

να επιστρέφουν έναν κατάλληλο 'Vary' header, όπως ορίζεται στην παράγραφο 14.44 του προσχέδιου RFC2616 (Fielding, et al., 1999).

## 2.4. Η μορφή του Timezone Header

Οι HTTP πελάτες μπορούν να παρέχουν τοπικές πληροφορίες ζώνης ώρας κατά την επίσκεψη σε ιστοσελίδες. Αυτές οι πληροφορίες θα αποστέλλονται με τον πελάτη χρησιμοποιώντας ένα Timezone-HTTP header και θα έχει τα εξής πεδία:

Το πεδίο `client-timezone` θα είναι μία συμβολοσειρά που θα περιέχει τους χαρακτήρες 'Timezone' και θα χωρίζεται από το υπόλοιπο κομμάτι με το σύμβολο ':' .

Το πεδίο `rfctime` θα περιέχει μία συμβολοσειρά με την τοπική ώρα, όπως ορίζεται κατά το RFC 3339 (Klyne & Newman, 2002).

Το πεδίο `posixtz` που θα περιέχει μία συμβολοσειρά ζώνης ώρας όπως ορίζεται κατά το `posixtz` (IEEE, 2004)

Το πεδίο `tzname` που θα αποτελεί όνομα της ζώνης ώρας. Πολλά συστήματα χρησιμοποιούν ένα όνομα ζώνης ώρας από μία βάση δεδομένων ζώνης ώρας. Το πεδίο αυτό αποτελεί μια τέτοια συμβολοσειρά. Περισσότερες από μία τέτοιες συμβολοσειρές μπορούν να παρέχονται. Όλες τους πρέπει να προσδιορίζουν την ίδια ζώνη ώρας.

Οι HTTP πελάτες θα πρέπει να παρέχουν και τα τρία μέρη του Header, εκτός αν δεν είναι σε θέση να το πράξουν ή να έχουν ρυθμίσει να μην στέλνουν πληροφορίες ζώνης ώρας.

Οι πελάτες που δεν έχουν κάποια αναφορά για την ώρα θα πρέπει να συμπληρώσουν με χρήση μηδενικών όλα τα ψηφία της συμβολοσειράς 'datetime', εκτός από τη απόκλιση με το UTC. Το σύμβολο "Z" θα πρέπει επίσης να χρησιμοποιηθεί όταν η απόκλιση δεν είναι διαθέσιμη.

Σύμφωνα με όλα αυτά τα στοιχεία μία πρότυπη μορφή για ένα τέτοιο header είναι η εξής όπως φαίνεται στην Εικόνα 9.

```
Timezone: 1983-09-03T13:00+0200;  
EET2EEST3,M3.2.0/02:00,M11.1.0/02:00;  
Europe/Athens;
```

Εικόνα 9 Ολοκληρωμένη Μορφή Ενός Timezone header

Στην παραπάνω εικόνα η πρώτη σειρά αποτελεί την ημερομηνία, ακολουθούμενη από το σύμβολο 'T', την ώρα και την απόκλιση. Στην δεύτερη σειρά βρίσκεται η ζώνη ώρας σύμφωνα με το posixtz (IEEE, 2004). Η τρίτη σειρά περιέχει το πεδίο tzname, που είναι το όνομα της ζώνης ώρας του πελάτη.

Για έναν πελάτη ο οποίος δεν θα διέθετε πληροφορίες για απόκλιση της ζώνης ώρας το header θα είχε την εξής μορφή που φαίνεται παρακάτω στην Εικόνα 10.

```
Timezone: 0000-00-00T00:00+0000Z;  
EET2EEST3,M3.2.0/02:00,M11.1.0/02:00;  
Europe/Athens;
```

*Εικόνα 10 Μορφή Timezone Header Χωρίς Απόκλιση Από UTC*

Επίσης μία πιο συμπαγής μορφή αυτού του header απεικονίζεται παρακάτω στην Εικόνα 11

```
Timezone: 1983-09-03T13:00+2:00
```

*Εικόνα 11 Συμπαγής Μορφή Ενός Timezone Header*

Εδώ το header αποτελείται από την ημερομηνία, το σύμβολο 'T', την ώρα και την απόκλιση από το UTC.

Βέβαια αυτή ήταν μία μορφή η οποία ήταν πρότυπη και απολύτως βασισμένη πάνω στο προσχέδιο του καθηγητή κ. Χαρχαλάκη (Παράρτημα Α). Ύστερα από πολλές συζητήσεις, προτάσεις και αναλύσεις με τον ίδιο, καταλήξαμε σε μία μορφή ελαφρώς τροποποιημένη, όμως περισσότερο ολοκληρωμένη ως προς τα δεδομένα. Σύμφωνα με αυτήν την πρόταση το header αποτελείται από τα εξής τμήματα:

- Η συμβολοσειρά 'Timezone'
- Το διαχωριστικό σύμβολο ':'
- Η ημερομηνία σε μορφή 'EEEE-MM-HH', όπου 'E' = έτος, 'M' = μήνας, 'H' = ημέρα
- Το διαχωριστικό σύμβολο 'T' που υποδεικνύει ότι ακολουθεί η ώρα

- Η ώρα σε μορφή 'ΩΩ:ΛΛ:ΔΔ', όπου 'Ω' = ώρες, 'Λ' = λεπτά, 'Δ' = δευτερόλεπτα
- Το διαχωριστικό σύμβολο ':'
- Την απόκλιση από το UTC
- Το διαχωριστικό σύμβολο ':'
- Το όνομα της ζώνης ώρας, προσδιορισμένο όσο το δυνατόν πιο σωστά γεωγραφικά, σε μορφή 'Ηπειρος/Χώρα' και σε περίπτωση που αυτό δεν είναι εφικτό, το πεδίο 'Undefined'
- Το διαχωριστικό σύμβολο ':'
- Το DST όπου θα πληροφορεί αν η χώρα διαθέτει θερινή ώρα σε μορφή 'yes' ή 'no'

Συμψηφίζοντας τις παραπάνω πληροφορίες ένα τέτοιο παράδειγμα απεικονίζεται παρακάτω στην εικόνα Εικόνα 12.

```
Timezone: 2011-09-17T03:28:29:+02:00:Europe/Athens,Istanbul:yes
```

*Εικόνα 12 Τελική Μορφή Ενός Timezone Header*

Αντίστοιχα εάν οι πληροφορίες δεν ήταν διαθέσιμες το παράδειγμα θα μετατρεπόταν όπως στην ακόλουθη Εικόνα 13.

```
Timezone: 0000-00-00T00:00:00:+00:00:Undefined:no
```

*Εικόνα 13 Τελική Μορφή Ενός Μη Προσδιορισμένου Timezone Header*

## Επίλογος

Στο προηγούμενο κεφάλαιο είδαμε πως η γεωγραφική τοποθεσία των χωρών επηρεάζει την επικοινωνία μεταξύ των υπολογιστών. Αναλύσαμε τον τρόπο με τον οποίο ορίζονται οι όροι 'ζώνη ώρας' και 'θερινή ώρα'. Περιορίσαμε και επιβάλλαμε συνθήκες οι οποίες συμβαδίζουν με ένα προσχέδιο. Πάνω σε αυτό βασιστήκαμε για τον προσδιορισμό της μορφής του Timezone Header.

Στη συνέχεια θα δούμε το περιβάλλον πάνω στο οποίο θα βασιστούμε για την υλοποίηση και δεν είναι άλλο από τον browser Mozilla Firefox. Θα δούμε τον πυρήνα του περιβάλλοντος αυτού που είναι υπεύθυνο για την επιτυχία του προγράμματος, θα έρθουμε σε επαφή με την δομή ενός plugin και θα παρουσιάσουμε τα εργαλεία που χρειαζόμαστε για την ανάπτυξή του.

## 3. Ο Mozilla Firefox και το περιβάλλον ανάπτυξης

---

### Εισαγωγή

Ο Mozilla Firefox είναι ένα πρόγραμμα περιήγησης ιστοσελίδων διάσημο, τόσο για την απλότητα που προσφέρει στον μέσο χρήστη, όσο και για το πλήθος των δυνατοτήτων που δίνει στον προχωρημένο χρήστη.

Στη αρχική ενότητα θα γνωρίσουμε την μηχανή Gecko, υπεύθυνη για το περιβάλλον του Firefox.

Στην συνέχεια του κεφαλαίου θα κάνουμε αναφορά για την δομή ενός plugin. Δομή η οποία δεν είναι τυχαία και τα δομικά της στοιχεία προσφέρουν άπειρους συνδυασμούς για σύνθεση οποιασδήποτε υλοποίησης.

Στην τελευταία ενότητα παρουσιάζονται τα εργαλεία που είναι χρήσιμα για την ανάπτυξη αυτού του plugin. Τα εργαλεία αυτά θα πρέπει να επιλεγούν με βάση την απλότητα και την αποδοτικότητα. Επίσης θα πρέπει να είναι ολοκληρωμένα και να στοχεύουν τόσο στην σωστή ανάπτυξη όσο και στην εκσφαλμάτωση του κώδικα.



### 3.1 Η μηχανή Gecko

Όλες οι εφαρμογές περιήγησης στο διαδίκτυο χρησιμοποιούν ένα πρόγραμμα ο οποίος είναι υπεύθυνος για την ανάλυση, την διάταξη, και την απεικόνιση κώδικα γραμμένου σε γλώσσες διαδικτυακού προγραμματισμού. Το πρόγραμμα αυτό ονομάζεται μηχανή και είναι τα θεμέλια πάνω στα οποία βασίζεται η εφαρμογή περιήγησης.

Αντίστοιχα, στον Mozilla Firefox, η μηχανή Gecko είναι το κύριο χαρακτηριστικό που κάνει αυτήν την εφαρμογή τόσο γνωστή. Η μηχανή περιήγησης Gecko είναι ανοικτού κώδικα και σχεδιάστηκε για να υποστηρίξει ανοικτά (αγγ. Open Source) πρότυπα διαδικτύου, όπως η HTML 5, CSS 1 και 2, το W3C DOM, XML,RDF και Javascript. Βέβαια, υποστηρίζει και γλώσσες γραφικών ή απεικόνισης διανυσμάτων όπως η SVG, η WebGL και η OpenGL.

Η Gecko είναι γραμμένη σε γλώσσα C++, υποστηρίζει ποικίλες πλατφόρμες, όπως Microsoft Windows, Linux, Mac OS X, Solaris και πολλά άλλα λειτουργικά συστήματα, καθώς η ανάπτυξη του είναι συνεχείς. Παλαιότερα αποτελούσε τον πυρήνα και για άλλες εφαρμογές, όπως ο browser Netscape με το όνομα 'NGLayout' ενώ μετά από διαφωνία για παραβίαση εμπορικού σήματος μετονομάστηκε σε 'Gecko'. Από τον Ιούλιο του 2003 έως και σήμερα, υπεύθυνος για την ανάπτυξη του είναι το ίδρυμα Mozilla και απλοί εθελοντές. Η ταχεία και συνεπής ανάπτυξη έχει φέρει την μηχανή Gecko στην έκδοση 6 και τον Mozilla Firefox στην έκδοση 6.0.2 (Εικόνα 14) .



Εικόνα 14 Το Λογότυπο Του Mozilla Firefox

Η μηχανή και το περιβάλλον του Mozilla Firefox όμως δεν σταματάει στην απεικόνιση διαδικτυακών γλωσσών προγραμματισμού. Το ισχυρό περιβάλλον προγραμματισμού που διαθέτει, επιτρέπει στον κάθε χρήστη να αναπτύξει λογισμικό για να εξατομικεύσει τις ανάγκες του. Αυτό το λογισμικό ονομάζεται plugin και απαρτίζεται από διάφορες γλώσσες και στοιχεία προγραμματισμού.

## 3.2. Η δομή ενός plugin

Το plugin είναι μία εφαρμογή, βασισμένη σε ένα πρόγραμμα που έχει σκοπό να εξατομικεύσει ή να προσθέσει μία λειτουργία. Μία τέτοια εφαρμογή τους πόρους για την ανάπτυξη πάνω στο πρόγραμμα για το οποίο προορίζεται και όπως είναι λογικό, δεν μπορεί να σταθεί σαν αυτόνομη εφαρμογή. Στο περιβάλλον του Mozilla Firefox αποτελείται από διάφορα στοιχεία σε διαφορετικές γλώσσες προγραμματισμού και από διαφορετικούς τύπους αρχείων. Ο λόγος είναι ότι με αυτόν τον τρόπο μπορούν να καλυφθούν σχεδόν όλες οι περιπτώσεις για υλοποίηση μίας ιδέας.

Κατά την αρχικοποίηση ενός plugin θα πρέπει να υπάρχει ένα αρχείο RDF, το οποίο κατά λέξη μεταφράζεται ως πλαίσιο περιγραφής πόρων (αγγ. Resource Definition Framework). Το RDF αυτό αρχείο είναι ένα XML αρχείο που λειτουργεί σαν ένα αφηρημένο μοντέλο δεδομένων (Feldt, 2007). Τα δεδομένα αυτά, ορίζονται από tags σε μορφή XML και περιέχουν τιμές. Ο Firefox μεταφράζει τα δεδομένα αυτά και δημιουργεί μία δομή δεδομένων.

Επίσης, κατά την αρχικοποίηση του plugin, ένα Manifest αρχείο περιγράφει όλα εκείνα τα πακέτα που θα χρειαστούμε για το plugin καθώς και τη σχετική τους θέση στο σκληρό δίσκο του υπολογιστή (Edwards, 2009).

Στην διεπαφή με τον χρήστη συμβάλλει η XUL (σύντμηση του όρου: XML User Interface Language) που είναι μία XML-μορφής γλώσσα προγραμματισμού διεπαφής χρήστη. Φυσικά, η XUL αναλαμβάνει την απεικόνιση και μόνο ενός plugin.

Ένα plugin όμως πρέπει να έχει και κάποια λειτουργία. Για τον προγραμματισμό μίας λειτουργίας είναι υπεύθυνη η Javascript. Το μοντέλο προγραμματισμού δεν είναι διαφορετικό από την Javascript που χρησιμοποιείται κατά κόρων για ανάπτυξη ιστοσελίδων. Οι διαθέσιμες λειτουργίες όμως εμπλουτίζονται από τα XPCOM components (mozdev.org).

Όποιες λειτουργίες δεν μπορούν να καλυφθούν από την Javascript, καλύπτονται από το XPCOM συστατικό μοντέλο αντικειμένου (αγγλ. XPCOM component object model). Τα XPCOM στοιχεία που απαρτίζουν το μοντέλο είναι γραμμένα σε Javascript, Java, C++ ή Python και θα πρέπει να είναι 'compiled' σε γλώσσα μηχανής. Μπορούν να επεκτείνουν τα ήδη υπάρχοντα components ή να δημιουργούν καινούργια με διαφορετική λειτουργία.

Τέλος, ένα plugin συμπληρώνει το Cascading Style Sheets (συντμ. CSS), σκοπός του οποίου είναι να εμπλουτίσει ή ακόμη και να τροποποιήσει την εμφάνιση του plugin. Το CSS και σε αυτήν την περίπτωση δεν διαφέρει από εκείνο που χρησιμοποιείται για την ανάπτυξη ιστοσελίδων. (Feldt, 2007)

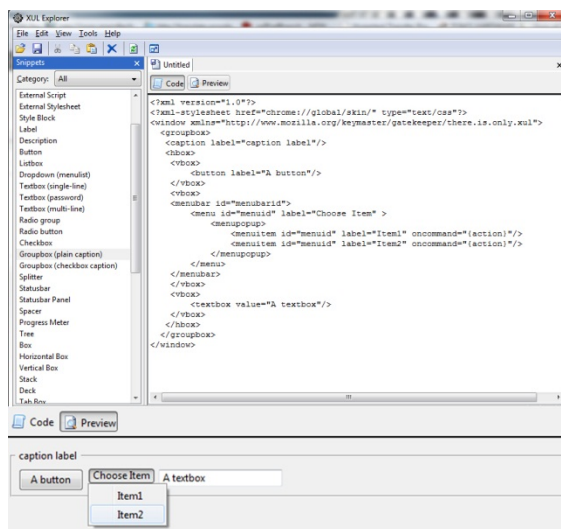
### 3.3. Απαραίτητα Εργαλεία για την ανάπτυξη plugin

Όπως είναι αυτονόητο η εγκατάσταση του προγράμματος περιήγησης Mozilla Firefox είναι απαραίτητη πριν καν να ξεκινήσουμε την ανάπτυξη του κώδικα του plugin. Για την εφαρμογή αυτή, χρησιμοποιήθηκαν όλες οι εκδόσεις του Firefox από 3.16 έως 6.0.\*. Για λόγους όμως συμβατότητας, δούλεψα σε εκδόσεις 4.\*.\* και πάνω.

Τα εργαλεία που θα χρειαστούμε κατά την ανάπτυξη του plugin θα πρέπει να είναι λειτουργικά, πρακτικά και με μηδενικό περιθώριο λάθους. Θα πρέπει να καλύπτουν όλες τις πτυχές της εφαρμογής τόσο το σχεδιαστικό κομμάτι όσο και το λειτουργικό. Παράλληλα θα πρέπει να δίνουν τη δυνατότητα για γρήγορη ανάπτυξη κώδικα και να συμβάλλουν στη γρήγορη διόρθωση. Όμως πέρα από τα εργαλεία που θα χρειαστούμε για τον κώδικα, θα ήταν χρήσιμο να παρουσιάσουμε και τα εργαλεία εκείνα που βοηθούν στον έλεγχο λειτουργίας της εφαρμογής.

Για το σχεδιαστικό κομμάτι επέλεξα τον συνδυασμό δύο εφαρμογών. Οι εφαρμογές αυτές είναι η XUL Explorer (XUL Explorer, 2009) και ο διάσημος επεξεργαστής κειμένου Notepad++ (Notepad++, 2011).

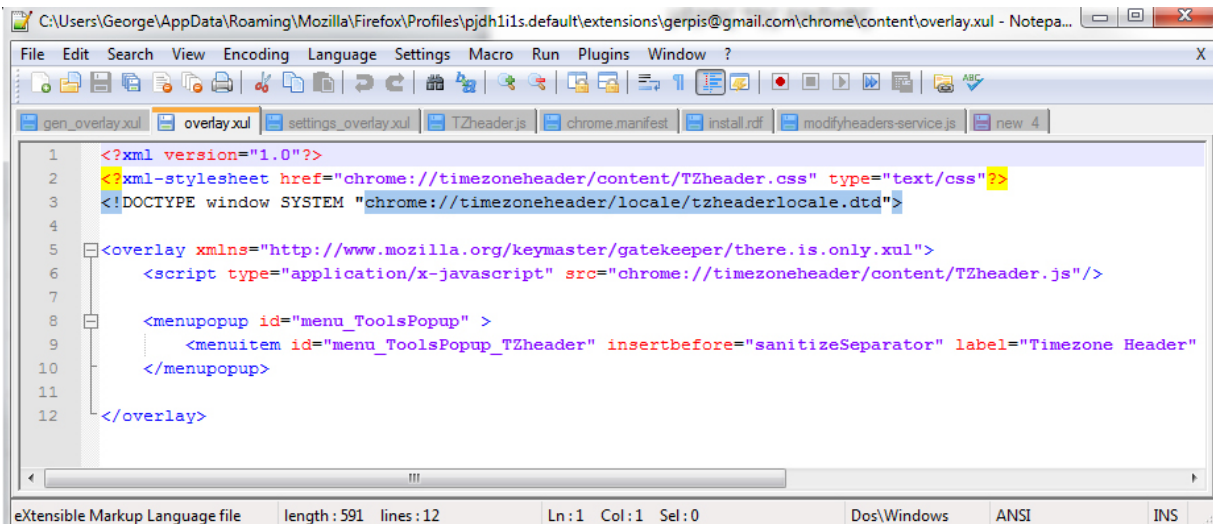
Το XUL Explorer είναι μία εφαρμογή η οποία διαθέτει όλες τις αναφορές για τα αντικείμενα της XUL, ενσωματώνει επισκόπηση κώδικα, είναι ανοιχτού λογισμικού και το σημαντικότερο, είναι άμεσο και απλό. Στην παρακάτω εικόνα απεικονίζεται το περιβάλλον του XUL Explorer (Εικόνα 15).



Εικόνα 15 Το Περιβάλλον Του XUL Explorer

Αμέσως μπορούμε να διακρίνουμε στην αριστερή στήλη όλα τα διαθέσιμα αντικείμενα που αποτελούν στοιχεία για ανάπτυξη του κώδικα, ενώ πατώντας το κουμπί 'preview', το κυρίως παράθυρο θα απεικονίσει μία επισκόπηση του κώδικα όπως φαίνεται στο κάτω μέρος της εικόνας.

Το Notepad++ είναι ένας επεξεργαστής κειμένου. Κύρια χαρακτηριστικά αυτού είναι η υποστήριξη σχεδόν όλων των γλωσσών προγραμματισμού, η υπογράμμιση κώδικα, η πρόβλεψη κώδικα ενώ παράλληλα είναι λογισμικό ανοικτού κώδικα. Παρακάτω στην Εικόνα 16 φαίνεται το περιβάλλον του 'Notepad++'. Μπορούμε να παρατηρήσουμε τον διαφορετικό χρωματισμό που υποδηλώνουν δεσμευμένες λέξεις ή απαραίτητα στοιχεία για την γραφή σωστού κώδικα στην εφαρμογή.



```

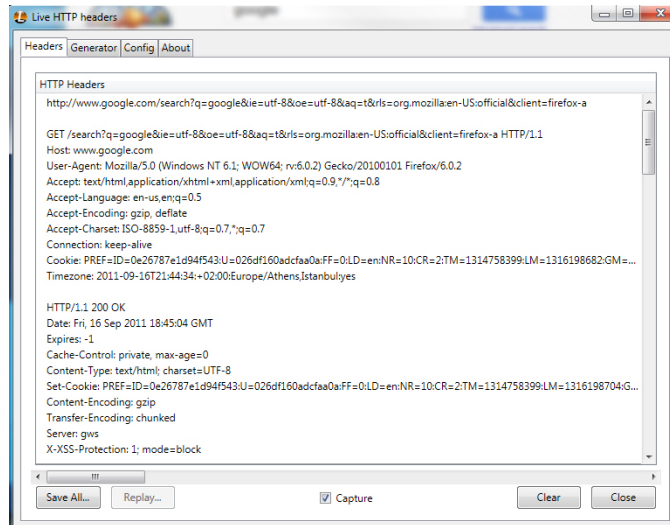
1 <?xml version="1.0"?>
2 <?xml-stylesheet href="chrome://timezoneheader/content/TZheader.css" type="text/css"?>
3 <!DOCTYPE window SYSTEM "chrome://timezoneheader/locale/tzheaderlocale.dtd">
4
5 <overlay xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
6   <script type="application/x-javascript" src="chrome://timezoneheader/content/TZheader.js"/>
7
8   <menupopup id="menu_ToolsPopup" >
9     <menuitem id="menu_ToolsPopup_TZheader" insertbefore="sanitizeSeparator" label="Timezone Header"
10    </menuitem>
11  </menupopup>
12 </overlay>

```

Εικόνα 16 Το Περιβάλλον Του Notepad++

Για το λειτουργικό κομμάτι επέλεξα επίσης την εφαρμογή Notepad++. Οι λόγοι δεν είναι διαφορετικοί από ότι στο σχεδιαστικό κομμάτι. Σημαντικό όμως πλεονέκτημα αυτής της επιλογής είναι η διαθεσιμότητα του κώδικα της εφαρμογής συγκεντρωμένη σε ένα πρόγραμμα.

Για να διαπιστώσουμε την λειτουργία της εφαρμογής, επιλέχθηκε ένα plugin για το Mozilla Firefox το οποίο ονομάζεται 'Live-HTTP headers'. Το plugin αυτό ενεργοποιώντας το, πληροφορεί το χρήστη για όλα τα headers που στέλνονται σε έναν διακομιστή (Εικόνα 17).



*Εικόνα 17 Το Περιβάλλον Του Live HTTP Headers*

Τέλος χρήσιμο θα ήταν να αναφέρουμε κάποιες άλλες επιλογές που απορρίφθηκαν. Το ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού Netbeans, διαθέτει ένα plugin που ονομάζεται Foxbeans. Το plugin αυτό βοηθάει στην ολοκληρωμένη ανάπτυξη ενός plugin για Firefox. Δυστυχώς όμως, προβλήματα συμβατότητας και υποστήριξης με τις σύγχρονες εκδόσεις Firefox, στάθηκαν εμπόδιο για την εκμετάλλευσή του.

## Επίλογος

Στο προηγούμενο κεφάλαιο είδαμε την μηχανή Gecko, υπεύθυνη για το περιβάλλον του Firefox. Ακολούθως, γνωρίσαμε τα δομικά στοιχεία ενός plugin σύμφωνα με τα οποία θα υλοποιήσουμε την εφαρμογή. Εφαρμογή η οποία χρειάζεται για την ανάπτυξή της.

Το επόμενο κεφάλαιο παρουσιάζει αναλυτικά την ανάπτυξη της εφαρμογής αυτής. Θα προετοιμάσουμε το έδαφος για την ανάπτυξη αυτού του plugin και θα δούμε πως συνδέετε η δομή με την κώδικα. Έτσι θα δημιουργήσουμε την διεπαφή και παράλληλα θα χτίσουμε την λειτουργία του. Τέλος, θα το δοκιμάσουμε και θα ολοκληρώσουμε τις λεπτομέρειες που θα το κάνουν φιλικότερο προς τον χρήστη.



## 4. Ανάπτυξη του κώδικα για το plugin

---

### Εισαγωγή

Έχοντας αναλύσει το πρόβλημα και τις αιτίες, κάναμε μία θεωρητική πρόταση για το πώς θα μπορούσαν να μειωθούν οι επιπλοκές που δημιουργεί η διαφορά ώρας μεταξύ των χωρών. Στο κεφάλαιο αυτό θα υλοποιήσουμε την εφαρμογή, βασισμένοι σε αυτήν την πρόταση και θα δούμε τον τρόπο λειτουργίας της.

Ξεκινώντας, θα προετοιμάσουμε το έδαφος για την ανάπτυξη αυτού του plugin. Θα τροποποιήσουμε παραμέτρους του Firefox, που θα μας διευκολύνουν κατά την ανάπτυξη και εκσφαλμάτωση του κώδικα.

Έπειτα, θα δημιουργήσουμε τη δομή του plugin και την σύνδεση του Firefox με τον κώδικά μας.

Έτσι, θα μπορέσουμε να αναπτύξουμε την διεπαφή και να αρχικοποιήσουμε τις απαιτούμενες παραμέτρους.

Παράλληλα θα δημιουργήσουμε την λειτουργία του, δηλαδή την κατασκευή και αποστολή του header.

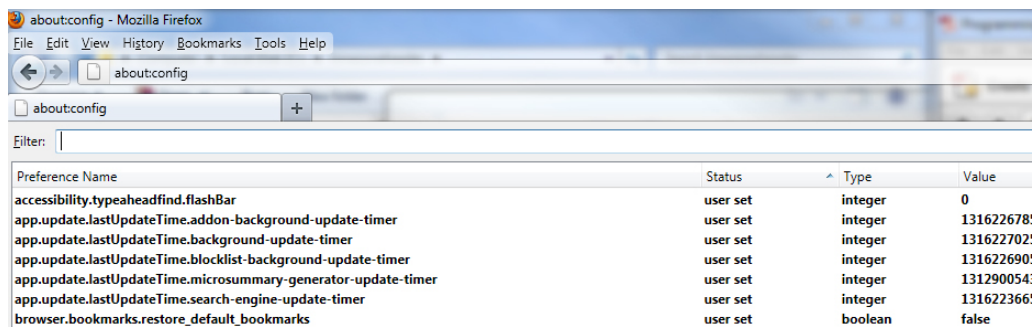
Τέλος, θα δοκιμάσουμε την λειτουργία του, τον τρόπο εκσφαλμάτωσης και θα ολοκληρώσουμε την εφαρμογή μας με τέτοιες διαδικασίες ώστε να γίνει πρακτική προς τον τελικό χρήστη.

## 4.1. Προετοιμάζοντας τον Firefox για ανάπτυξη του plugin

Ο Firefox διαθέτει ένα πλήθος από λειτουργίες που απευθύνονται σε κάθε επίπεδο χρήστη. Για προστασία όμως των αρχάριων χρηστών και βελτιωμένη απόδοση για το μέσο χρήστη, πολλές από αυτές τις επιλογές είναι ανενεργές ή δεν υπάρχουν καν,

Για να αναπτυχθεί όμως ένα plugin στον Firefox θα πρέπει να θέσουμε κάποιες παραμέτρους που θα μας διευκολύνουν κατά την εκσφαλμάτωση. Οι παράμετροι αυτοί είναι κρυφές επιλογές που αλλάζουν κάποια δυνατότητα του περιηγητή.

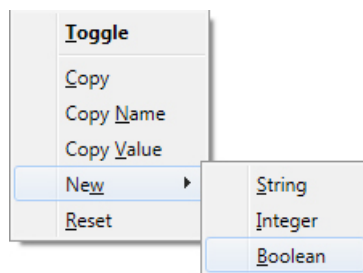
Έτσι για να αποκτήσουμε πρόσβαση σε αυτές, ανοίγοντας τον περιηγητή Mozilla Firefox πληκτρολογούμε στην URL μπάρα 'about:config'(Εικόνα 18). Η εντολή αυτή δεν είναι κάποιο URL και δεν υπάρχει σε κάθε πρόγραμμα περιήγησης.



Preference Name	Status	Type	Value
accessibility.typeaheadfind.flashBar	user set	integer	0
app.update.lastUpdateTime.addon-background-update-timer	user set	integer	1316226785
app.update.lastUpdateTime.background-update-timer	user set	integer	1316227025
app.update.lastUpdateTime.blocklist-background-update-timer	user set	integer	1316226905
app.update.lastUpdateTime.microsummary-generator-update-timer	user set	integer	1312900543
app.update.lastUpdateTime.search-engine-update-timer	user set	integer	1316223665
browser.bookmarks.restore_default_bookmarks	user set	boolean	false

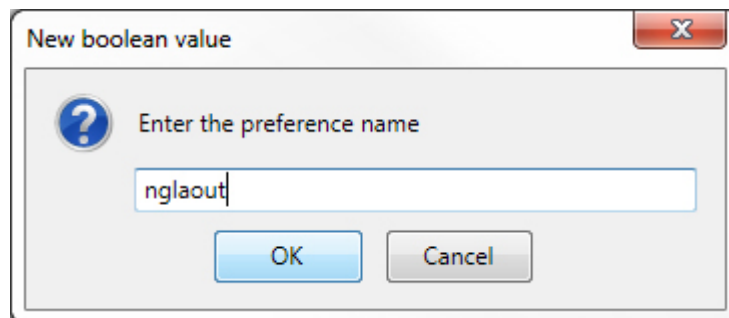
Εικόνα 18 Οι Παράμετροι Στο about:config

Θα παρατηρήσουμε όλες εκείνες τις παραμέτρους τις οποίες αναφέραμε παραπάνω. Αυτοί οι παράμετροι ονομάζονται 'Preferences' (ελλ. Προτιμήσεις) και αν δεν υπάρχει αυτό που χρειαζόμαστε αναζητώντας το στο πεδίο 'Filter', θα πρέπει να το δημιουργήσουμε. Ο τρόπος για τη δημιουργία είναι πολύ απλός. Με δεξιά 'κλικ' σε κάποιο χώρο του παραθύρου επιλέγουμε 'New > Boolean' (Εικόνα 19).



Εικόνα 19 Δημιουργία Νέου Preference

Το νέο preference που δημιουργείται είναι τύπου 'Boolean' δέχεται δηλαδή τιμές 'true' ή 'false'. Το preference αυτό θα πρέπει να έχει κάποιο όνομα το οποίο εισάγουμε στο κενό πεδίο (Εικόνα 20).



Εικόνα 20 Πεδίο Νέας Μεταβλητής

Έτσι δημιουργούμε τις εξής τέσσερις μεταβλητές με συγκεκριμένες λειτουργίες:

- 'browser.dom.window.dump.enabled' Το preference αυτό επιτρέπει η έξοδος μέσω συγκεκριμένης εντολής να γίνει σε κονσόλα του Firefox.
- 'devtools.errorconsole.enabled' Το preference αυτό ενεργοποιεί τον έλεγχο για τυχόν λάθη και προειδοποιήσεις σε Javascript, CSS κ.ά.
- 'nglayout.debug.disable\_xul\_cache' Το preference αυτό αποτρέπει την αποθήκευση XUL. Κατά την ανάπτυξη του κώδικα όμως, εμείς δεν θέλουμε κατι τέτοιο καθώς οποιεσδήποτε αλλαγές πρέπει να είναι άμεσα ορατές.
- 'nglayout.debug.disable\_xul\_fastload' Το τελευταίο preference είναι σχετικό με το παραπάνω. Η λειτουργία του αποτρέπει την αποθήκευση XUL/JS στο σκληρό δίσκο.

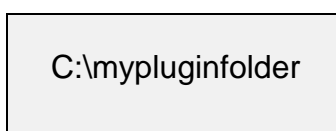
Έχοντας δημιουργήσει το τελευταίο preference, η μόνη εκκρεμότητα είναι να αλλάξουμε τις τιμές τους σε 'true'. Πράγμα το οποίο γίνεται κάνοντας διπλό 'κλικ'.

Τέλος για να εμφανίσουμε την κονσόλα θα πρέπει από τη γραμμή εντολών των Windows να καλέσουμε τον Firefox με την παράμετρο '-console'. Για παράδειγμα 'C:\Program Files (x86)\Mozilla Firefox\firefox.exe" -console'.

## 4.2. Δημιουργώντας τη δομή και το σημείο σύνδεσης με την εφαρμογή

Όλα τα plugins έχουν συγκεκριμένη δομή φακέλων και αρχείων. Επίσης, για να λειτουργήσει ένα plugin θα πρέπει να βρίσκεται σε συγκεκριμένο φάκελο που ορίζεται από τον Firefox. Ο φάκελος αυτός είναι ο '%appdata%\Mozilla\Firefox\Profiles\profile\extensions', όπου 'profile' το όνομα του προφίλ που έχουμε προεπιλεγμένο.

Εκεί δημιουργούμε ένα αρχείο '.txt' και μέσα γράφουμε την απόλυτη διαδρομή (αγγλ. Path) που θα βρίσκεται το plugin (Εικόνα 21).

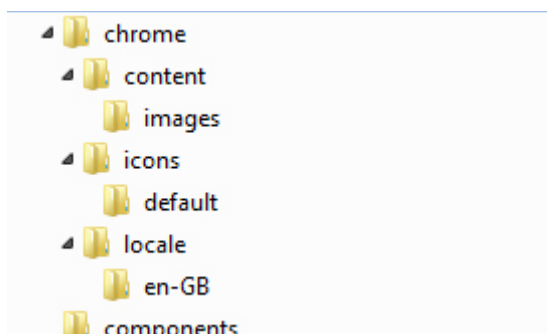


C:\mypluginfolder

Εικόνα 21 Παράδειγμα Απόλυτης Διαδρομής

Έπειτα, μετονομάζουμε το αρχείο αυτό σε ένα μοναδικό όνομα όπως 'mymail@mail.com'. Η διαδικασία αυτή δημιουργεί μία αναφορά για τον Firefox προσδιορίζοντας μια διαφορετική θέση για το plugin μας.

Στο 'root' αυτής της θέσης, οι φάκελοι θα πρέπει να έχουν την δομή όπως απεικονίζεται παρακάτω στην Εικόνα 22



Εικόνα 22 Δομή Φακέλων

Για να δημιουργήσουμε το plugin θα πρέπει να το 'περιγράψουμε'. Για το λόγο αυτό δημιουργούμε ένα αρχείο στο 'root' με κατάληξη '.RDF', το οποίο, όπως προαναφέρθηκε, έχει μορφή XUL. Επίσης, το συγκεκριμένο αρχείο είναι υπεύθυνο και για την περιγραφή του plugin, όπως 'Options', 'About' κ.ά. στην καρτέλα 'Extensions'

του Firefox. Ένα δείγμα του κώδικα και της λειτουργίας παρουσιάζεται παρακάτω στην Εικόνα 23

```
<?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:em="http://www.mozilla.org/2004/em-rdf#">
  <Description about="urn:mozilla:install-manifest">
    <em:creator>George Erpissimidis</em:creator>
    <em:id>mymail@mail.com</em:id>
  ...
    <em:targetApplication>
      <Description>
        <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
        <em:minVersion>4.0.*</em:minVersion>
        <em:maxVersion>6.0.*</em:maxVersion>
      </Description>
    </em:targetApplication>
  </Description>
</RDF>
```

Εικόνα 23 Δείγμα RDF Αρχείου

Παρατηρούμε ότι εδώ οι παράμετροι που δημιουργούν μία δομή δεδομένων. Έμφαση θα πρέπει να δοθεί στο tag ‘em:id’ το οποίο θα συμπίπτει ακριβώς με το όνομα στο αρχείο που δώσαμε παραπάνω. Έτσι ο Firefox θα το προσδιορίζει μοναδικά. Επίσης το αντίστοιχο ‘em:id’ που βρίσκεται στο ‘<em:targetApplication>’ προσδιορίζει το id της εφαρμογής που θα λειτουργήσει το plugin. Για την εικόνα συγκεκριμένα το id αυτό είναι το μοναδικό id του Firefox.

Στο root πρέπει επίσης να δημιουργηθεί ένα ‘.Manifest’ αρχείο που περιγράφει όλα εκείνα τα πακέτα που θα χρειαστούμε για το plugin καθώς και τη σχετική τους θέση στον υπολογιστή. Στην παρακάτω εικόνα (Εικόνα 24) φαίνεται το σημείο σύνδεσης του Firefox με τον κώδικά μας στις δύο πρώτες γραμμές. Στις ακόλουθες σειρές, ορίζουμε ένα πακέτο σε σχέση με το φάκελο που αναπτύσσουμε το plugin.

```
#Firefox
overlay chrome://browser/content/browser.xul   chrome://mypluginfolder/content/overlay.xul
...
#Setting paths for packages
content mypluginfolder chrome/content/
locale mypluginfolder en-GB chrome/locale/en-GB/
...
```

Εικόνα 24 Δείγμα Manifest Αρχείου

### 4.3. Δημιουργώντας τη διεπαφή με το χρήστη

Στην προηγούμενη υποενότητα είδαμε το σημείο σύνδεσης του Firefox με τον κώδικα. Στη συνέχεια, θα επεκταθούμε στην δημιουργία διεπαφής, που ορίσαμε στο σημείο εκείνο. Η διεπαφή αυτή που ορίζεται από ένα URL είναι ένα overlay, δηλαδή μία επίστρωση, η οποία χρησιμοποιεί ένα package και συγκεκριμένα το 'browser.xul'. Το package αυτό περιέχει όλα εκείνα τα στοιχεία που θα χρειαστούν για τη δημιουργία της διεπαφής. Το αρχείο, θα πρέπει να έχει το όνομα , στη περίπτωση αυτή 'overlay.xul' και τη δομή στους φακέλους όπως φαίνεται παραπάνω (Εικόνα 24).

Χρησιμοποιώντας κώδικα σε XUL, κατασκευάζουμε ένα μενού το οποίο θα βρίσκεται στα 'Tools' του Firefox. Από εκεί ο χρήστης θα έχει τη δυνατότητα να το ενεργοποιήσει ώστε να δει τις ρυθμίσεις του. Κατά την εκτέλεση, θα κληθεί μία μέθοδος στο αρχείο 'TZheader.js', ώστε να κάνουμε την λειτουργία που χρειαζόμαστε. Στην παρακάτω εικόνα (Εικόνα 25) αυτό επιτυγχάνεται από το 'oncommand='TZheader\_open()'".

```
...
<overlay xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <script type="application/x-javascript" src="chrome://timezoneheader/content/TZheader.js"/>
  <menupopup id="menu_ToolsPopup" >
    <menuitem id="menu_ToolsPopup_TZheader" insertbefore="sanitizeSeparator"
      label="Timezone Header" class="menuitem-iconic"
      oncommand="TZheader.open()" />
  </menupopup>
</overlay>
....
```

Εικόνα 25 Κώδικας XUL Για Τοποθέτηση Menu Plugin

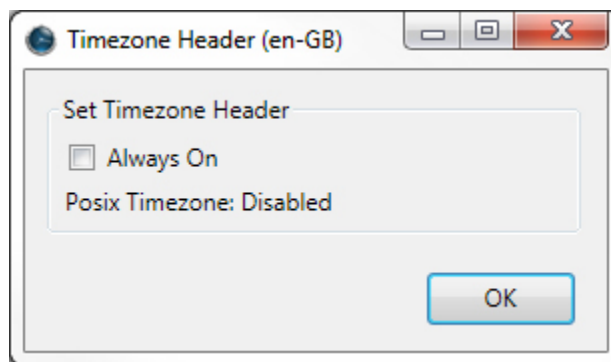
Η μέθοδος αυτή θα παράγει δυναμικά το 'User Interface' του plugin σε μορφή XUL. Στη συνέχεια θα δημιουργήσει και θα αρχικοποιήσει παραμέτρους στο 'about:config' που είδαμε σε προηγούμενη υποενότητα. Οι παράμετροι αυτοί θα χρησιμεύσουν στη λειτουργία του plugin (Εικόνα 26).

```
var TZheader = {
  open: function () {
    this.mainWindow = window.openDialog("chrome://mypluginfolder/content/gen_overlay.xul",
      "TZheader", "chrome,all,dialog=no");
  }
};
```

```
},  
initSettingsPane: function(){  
  ...  
}  
...  
}
```

Εικόνα 26 Μέθοδος *Open()* Για Αρχικοποίηση *Plugin* Και Παραμέτρων

Κατά την αρχικοποίηση του ‘User Interface’ το παράθυρο που θα ανοίξει θα έχει την μορφή όπως απεικονίζεται στην Εικόνα 27.



Εικόνα 27 Αρχικοποίηση *UI*

Ταυτόχρονα, τα preferences που θα αρχικοποιηθούν θα βοηθήσουν στην λειτουργία του plugin. Σύμφωνα με την παρακάτω εικόνα (Εικόνα 28), η πρώτη παράμετρος θα ορίζει αν το plugin βρίσκεται σε λειτουργία και εξαρτάται από το ‘Checkbox’ που βρίσκεται στο ‘User Interface’. Η δεύτερη παράμετρος θα αποθηκεύει το header στην μορφή που περιγράψαμε στο κεφάλαιο 2. Η τελευταία παράμετρος περιέχει μία χρονική σήμανση, σύμφωνα με την οποία θα γίνεται έλεγχος για αλλαγή της ‘posixvalue’ ανά συγκεκριμένα διαστήματα και συγκεκριμένα ανά 60 δευτερόλεπτα. Ο λόγος ύπαρξης της είναι για να αποφευχθούν περιττοί υπολογισμοί οι οποίοι θα επιβάρυναν το τερματικό του χρήστη.

```
timezoneheader.config.alwaysOn=false  
timezoneheader.config.posixvalue=0000-00-00T00:00:00:+0:00:Undefined:false  
timezoneheader.config.tztimestamp=0
```

Εικόνα 28 Αρχικοποίηση *Preferences* Κατά Την Εκκίνηση

## 4.4. Ανάπτυξη του κώδικα για τη λειτουργία του plugin

Η υποενότητα αυτή θα αναλύσει τις διαδικασίες για την κατασκευή και αποστολή του header. Μέσα από την Javascript θα κατασκευάσουμε και θα συνδέσουμε την εφαρμογή μας με τα Components του Firefox. Το κομμάτι αυτό επικεντρώνεται κυρίως γύρω από το αρχείο 'modify-headers-service.js'.

Για να έχει σωστή λειτουργία η εφαρμογή μας, θα πρέπει πρώτα από όλα να λειτουργεί όλη την ώρα, ανεξάρτητα αν το 'User Interface' είναι ανοικτό και εφόσον το 'AlwaysOn' είναι ενεργό. Ως εκ τούτου, θα χρειαστούμε ένα Component του Firefox το οποίο στη ουσία θα λειτουργεί ως ένα 'Background Service' όπου θα γίνονται συνεχής έλεγχοι.

Στην ακόλουθη εικόνα είναι η δημιουργία σύνδεσης, ή αλλιώς, 'συμβολαίου' για τη χρήση μίας διεπαφής και συγκεκριμένα του 'nsIObserver' (Εικόνα 29).

```
classDescription: "Modify Headers Proxy",
  classID:        Components.ID("{0eff9eeb-c51a-4f07-9823-27bc32fdae13}"),
  contractID:     "@modifyheaders.mozdev.org/proxy;1",

  QueryInterface: XPCOMUtils.generateQI([Components.interfaces.nsIObserver]),
```

*Εικόνα 29 Δημιουργία Contract Για Χρήση Component Του Firefox*

Θα πρέπει να αναφέρουμε ότι για να υπάρχει μία τέτοια σύνδεση, είναι απαραίτητο το αρχείο στο τέλος να κάνει μία καταγραφή του Component στον browser, όπως φαίνεται παρακάτω (Εικόνα 30).

```
if (XPCOMUtils.generateNSGetFactory) {
  var NSGetFactory = XPCOMUtils.generateNSGetFactory([TimezoneHeader.Proxy]);
} else {
  var NSGetModule = XPCOMUtils.generateNSGetModule([TimezoneHeader.Proxy]);
}
```

*Εικόνα 30 Register Component With Browser*



Όταν η μέθοδος `observer`, παρατηρήσει ένα HTTP request (`http-on-modify-request`), τότε θα ανακτήσει ή θα παράγει ένα Timezone header, το οποίο είναι αποθηκευμένο στο preference `'timezoneheader.config.posixvalue'`. Εφόσον γίνει αυτό, τότε θα θέσει το επιθυμητό header για αποστολή στο HTTP request. Η διαδικασία αυτή αντικατοπτρίζεται με την ίδια σειρά παρακάτω (Εικόνα 31).

```
observe: function (subject, topic, data) {
  if (topic == 'http-on-modify-request'){...
    myHeader=prefs.getCharPref("config.posixvalue");
    subject.setRequestHeader("Timezone", myHeader, "true");
    ...}
  ...}
```

*Εικόνα 31 Μέθοδος Observe Για Έλεγχο HTTP Request*

Η δημιουργία του preference `'posixvalue'`, που αποτελεί ουσιαστικά το Timezone header, συνθέτεται από 2 μέρη. Το πρώτο μέρος δηλαδή η ημερομηνία και ώρα υπολογίζεται με τη βοήθεια της `'Date()'` της Javascript.

Το δεύτερο κομμάτι δηλαδή η απόκλιση από UTC, το όνομα της ζώνης ώρας και το DST υπολογίζονται με τη βοήθεια μίας μεθόδου. Η μέθοδος αυτή ελέγχει τις ημερομηνίες αλλαγής ώρας σε μία χώρα και τις συγκρίνει με μία βάση ζώνης ώρας, που ονομάζεται Olson Database (Olson, 1986).

Ενδεικτικά στην παρακάτω εικόνα φαίνεται κώδικας από τις μεθόδους για το πρώτο και δεύτερο μέρος αντίστοιχα (Εικόνα 32).

```
grabTz: function (){
  ...
  var aDate=d.getDate();
  ...
  var afullDate = d.getFullYear() + "-" + aMonth + "-" + aDate;
  var afullTime = aHours + ":" + aMinutes + ":" + aSeconds;
```

```
...  
}  
function date_is_dst(date) {  
...  
    var base_offset = ( (date.getMonth() > 5 ? get_june_offset() : get_january_offset()) )  
...  
}  
...  
}
```

*Εικόνα 32 Timezone Μέθοδοι*

## 4.5 Δοκιμή λειτουργίας και ολοκλήρωση του plugin

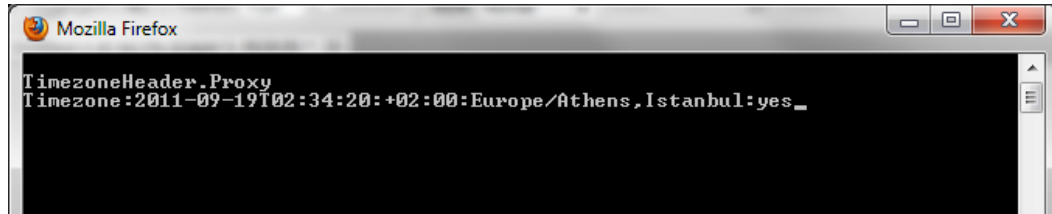
Κατά την διάρκεια του κώδικα θα πρέπει να δοκιμάζονται τα επιμέρους στάδια της ανάπτυξης καθώς και οι επιμέρους λειτουργίες του plugin. Καθώς η Javascript δεν προσφέρει ένα σταθερό περιβάλλον (Venkman) για εκσφαλμάτωση κατά την ανάπτυξη ενός plugin, θα πρέπει να εφαρμόσουμε εναλλακτικές μεθόδους. Επίσης, θα πρέπει να δοκιμάσουμε την λειτουργία του και να κάνουμε την εφαρμογή φιλικότερη προς το χρήστη.

Σε προηγούμενη ενότητα, είδαμε ότι ο Firefox διαθέτει κονσόλα στην οποία μπορούμε να εμφανίσουμε παραμέτρους γενικευμένες με κώδικα, για να εστιάσουμε το σφάλμα ή να επιβεβαιώσουμε μια λειτουργία. Σε αυτήν τη διαδικασία βοηθάει η εντολή 'dump()' η οποία είναι αντίστοιχη με την 'alert()' της Javascript με την διαφορά ότι εμφανίζει την έξοδο στην κονσόλα. Ένα απλό παράδειγμα στην εικόνα της κονσόλας παρακάτω (Εικόνα 33) , όπου εμφανίζεται αν ο έλεγχος έχει περάσει στο πρώτο κομμάτι και η τιμή που έχει το preference για το Timezone Header.

```
//1ο Έλεγχος
if (!TimezoneHeader.Proxy) {
    ...
    dump("\nTimezoneHeader.Proxy\n");
    ...
}
//2ο Έλεγχος
var showPref= config_prefs.getCharPref("config.posixvalue");
dump(showPref);
```

Εικόνα 33 Εκσφαλμάτωση Με dump()

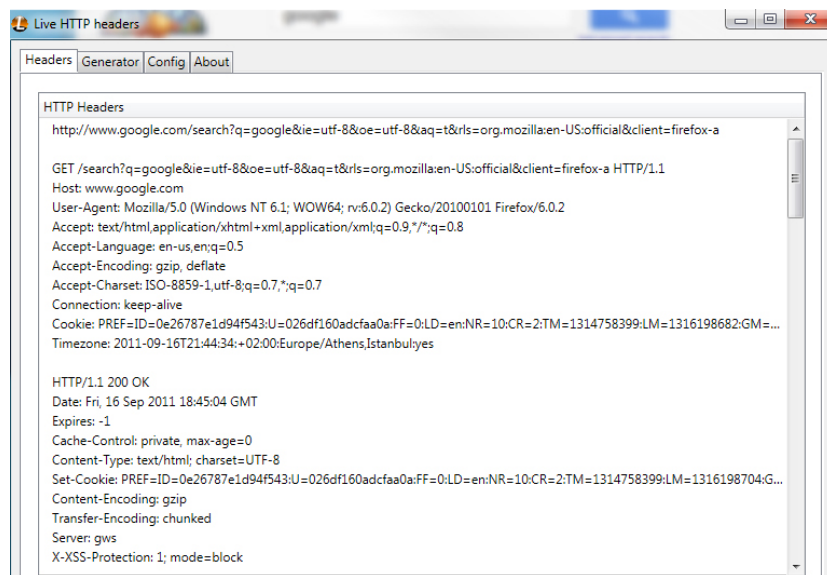
Τα αποτελέσματα από τις δύο αυτές εντολές θα εμφανιστούν στην κονσόλα ως εξής (Εικόνα 34).



Εικόνα 34 Έξοδος Σε Κονσόλα Του Firefox

Για να δούμε αν η λειτουργία του plugin είναι σωστή, θα χρησιμοποιήσουμε το plugin Live HTTP headers που αναφέραμε στην υποενότητα '3.3. Απαραίτητα Εργαλεία για την ανάπτυξη plugin'.

Έχοντας εγκαταστήσει το Live HTTP headers το ενεργοποιούμε από τη γραμμή 'Tools' του Firefox. Εφόσον έχουμε ενεργοποιήσει την επιλογή 'AlwaysOn' στο plugin που κατασκευάσαμε (Οδηγός Χρήσης Λογισμικού, σελ. 96 ) , κάνουμε ένα request προς μία ιστοσελίδα ή ,κοινώς, γράφουμε μία διεύθυνση στην μπάρα URL. Το Live-HTTP headers θα καταγράψει όλα τα headers που αποστέλλονται από τον client (Mozilla Firefox) στον server, μαζί και το header που δημιουργήσαμε με όνομα 'Timezone', όπως απεικονίζεται παρακάτω (Εικόνα 35).



Εικόνα 35 Καταγραφή Headers Με Το Plugin Live HTTP header

Το περιβάλλον του Firefox επιτρέπει το Language Localization, που είναι η προσαρμογή της γλώσσας του χρήστη. Αυτό, είναι ένα αρχείο με κατάληξη '.dtd' μέσα στον φάκελο 'locale/ογ-ΟΧ', όπου ογ= όνομα γλώσσας και ΟΧ= Όνομα Χώρας. Για τα ελληνικά ο αντίστοιχος φάκελος θα ήταν 'el-GR'. Το πεδίο πρέπει να ορίζεται μέσα σε ένα αρχείο XUL κατάλληλα (Εικόνα 36).

```
title="&tzheader.main.title;"
```

*Εικόνα 36 Ορισμός Πεδίου Για Localization*

Ταυτόχρονα το '.dtd' θα περιέχει την αντιστοιχία του πεδίου αυτού με τον ορισμό στην εκάστοτε γλώσσα (Εικόνα 37).

```
<!ENTITY tzheader.main.title "Timezone Header (en-GB)">
```

*Εικόνα 37 Localization Μέσω DTD*

Τέλος, για την ευκολία του χρήστη κατά την εγκατάσταση, μετατρέπουμε το plugin σε ένα πακέτο. Το πακέτο αυτό είναι '.zip' αρχείο με κατάληξη '.xpi'. Εφόσον λοιπόν μετατρέψουμε τους φάκελους μαζί με όλους τους υποφάκελους και τα αρχεία, που βρίσκονται μέσα στον 'root' φάκελο σε ένα 'zip', μετονομάζουμε το αρχείο σε '.xpi'. Το όνομα του αρχείου μπορεί να είναι οποιοδήποτε και η αναφορά για τον φάκελο του plugin μέσω αρχείου που περιγράψαμε στην παράγραφο 4.2 (σελ. 43) είναι πλέον περιττή.

## Επίλογος

Στο τελευταίο κεφάλαιο της πτυχιακής αναπτύξαμε τον κώδικα της εφαρμογής, χρησιμοποιώντας σαν άξονα την πρόταση για το Timezone Header που περιγράψαμε σε προηγούμενο κεφάλαιο.

Βασισμένοι σε αυτό, είδαμε τι παραμέτρους στον Mozilla Firefox χρειάστηκε να τροποποιήσουμε για την ανάπτυξη αυτού του plugin. Συνδέσαμε τον browser με τον κώδικά μας και δημιουργήσαμε την διεπαφή. Διεπαφή η οποία συνεργάζεται με την λειτουργία που αναπτύξαμε,. Τέλος, δοκιμάσαμε την εφαρμογή του και προσθέσαμε τις λεπτομέρειες που ολοκληρώνουν το plugin και δημιουργούν φιλικότερο περιβάλλον προς τον χρήστη.

## Επίλογος

---

Η ζώνη ώρας και η θερινή ώρα έχουν δημιουργηθεί σαν όροι για την διευκόλυνση των πολιτών και της κοινωνίας, γενικότερα. Ωστόσο, αυτό δημιουργεί προβλήματα στην παγκόσμια επικοινωνία μεταξύ υπολογιστικών συστημάτων και καθιστά την ακρίβεια των δεδομένων ελλιπή ως προς το θέμα χρόνου.

Παρατηρώντας την επικοινωνία μεταξύ υπολογιστών που χρησιμοποιούν το πρωτόκολλο HTTP, συλλέξαμε πολύτιμες πληροφορίες που μας βοηθούν να καταλάβουμε τα θεμέλια μιας επικοινωνίας και να εστιάσουμε στον τρόπο με τον οποίο θα μπορούσαμε να εφαρμόσουμε την λύση.

Αναλύοντας την ζώνη ώρας διαπιστώσαμε γιατί συμβαίνει αυτή η ανακρίβεια δεδομένων. Μέσα από την ανάγκη για απλούστευση και βελτίωση των δεδομένων αυτών, είδαμε ένα προσχέδιο σύμφωνα με το οποίο θα μπορούσε να δημιουργηθεί αυτή η λύση.

Ο Mozilla Firefox, όντας ένα μέσο πάνω στο οποίο βασίζεται η επικοινωνία και έχοντας ένα πλαίσιο που προσφέρει δυνατότητες για ανάπτυξη ήταν το κατάλληλο περιβάλλον για την εφαρμογή της λύσης. Έτσι δημιουργήθηκε η ανάγκη για ανάπτυξη ενός plugin πάνω στον Firefox.

Έπειτα, προετοιμάσαμε τις παραμέτρους που μας βοήθησαν για την ανάπτυξη ενός σωστού plugin. Γνωρίσαμε πως ο κώδικας παίρνει μορφή και αναπτύξαμε τις λειτουργίες βασισμένοι πάνω στην πρόταση που κάναμε. Στη συνέχεια συνδέσαμε με την εφαρμογή τις λειτουργίες, σύμφωνα με τη λύση που προτείναμε και ολοκληρώσαμε την εφαρμογή με το σκεπτικό ενός φιλικότερου περιβάλλοντος προς τον χρήστη.

## Συμπεράσματα και προτάσεις

---

Ο σκοπός αυτής της εργασίας ήταν η ανάπτυξη ενός plugin για τον Mozilla Firefox σε πλατφόρμα των Windows για αποστολή Timezone HTTP header βασιζόμενο στην τοποθεσία και ώρα του χρήστη, μία ιδέα σε προσχέδιο προς την IEEE, του κ. Στέφανου Χαρχαλάκη.

Έχοντας υλοποιήσει το αντικείμενο της πτυχιακής, θα λέγαμε ότι ο Mozilla Firefox προσφέρει ένα πολύ δυνατό και ολοκληρωμένο σε δυνατότητες περιβάλλον. Δίνει στον προγραμματιστή την ευκαιρία να κάνει αυτό ακριβώς που έχει σχεδιάσει, χωρίς να θέτει περιορισμούς. Το γεγονός ότι είναι ανοιχτού κώδικα βάζει την κατάλληλη υποδομή για περαιτέρω ανάπτυξη ή τροποποίηση ώστε να προσαρμοστεί στα μέτρα ενός προγραμματιστή.

Δυστυχώς, όμως, ολοκληρωμένες δυνατότητες δεν συνεπάγεται ολοκληρωμένη υποστήριξη. Οι δυσκολίες δεν ήταν λίγες και οι μέθοδοι εκσφαλμάτωσης ήταν εξαιρετικά χρονοβόρες. Αν και το διαθέσιμο υλικό για απλή κατασκευή ενός plugin ήταν πλούσιο, οι εξειδικευμένες πηγές ήταν λίγες. Παράλληλα οι συνεχείς αλλαγές σε εκδόσεις του Mozilla Firefox αύξανε την πολυπλοκότητα της ανάπτυξης.

Εδώ θα ήθελα να κάνω μια αναφορά σε γεγονότα άγνωστα προς τον αναγνώστη, σημαντικά όμως για τον προγραμματιστή. Όπως θα παρατηρήσατε το plugin είναι στην έκδοση 1.0.3. Ο λόγος είναι οι διαφορετικές εκδοχές, μέσα από έρευνα, συζητήσεις και σκέψεις με τον κ. Χαρχαλάκη.

Η πρώτη εκδοχή βασιζόταν σε μία λίστα όπου τα δεδομένα και οι προτιμήσεις παράγονταν δυναμικά σε XML αρχεία. Η εκδοχή απορρίφθηκε καθώς έδινε δυνατότητες προς τον χρήστη που δεν συμφωνούσαν με το προσχέδιο.

Η δεύτερη εκδοχή που απορρίφθηκε έκανε ανάκτηση μέσω registry keys των Windows. Η ευαισθησία των δεδομένων ήταν σημαντική αφορμή για κοινή αποδοχή της υλοποίησης.

Σαν συμπέρασμα κρίνω σκόπιμο να πω ότι με την τρίτη εκδοχή πετύχαμε μία απλή, λειτουργική και διαφανή εφαρμογή ως προς τον χρήστη. Μία υλοποίηση που ίσως γίνει πρότυπο και κατά την ανάπτυξη αποτέλεσε μία δημιουργική και ευχάριστη πρόκληση.



## Βιβλιογραφία

---

- XUL Explorer. (2009). Retrieved November 2010, from [https://developer.mozilla.org/en/xul\\_explorer](https://developer.mozilla.org/en/xul_explorer)
- Notepad++. (2011). Retrieved June 2011, from <http://notepad-plus-plus.org/>
- Aries, M. B., & Newsham, G. R. (2008, June). Effect of daylight saving time on lighting energy use: a literature review. Ottawa, Canada. Retrieved 4 2011, from <http://www.nrc-cnrc.gc.ca/obj/irc/doc/pubs/nrcc49212/nrcc49212.pdf>
- Benfield, J. C. (2001). *Energy Conservation Potential of Extended and Double Daylight Saving Time*. Statement to the U.S. House, Subcommittee on Energy, Committee on Science. Retrieved 08 19, 2011, from [http://commdocs.house.gov/committees/science/hsy73325.000/hsy73325\\_0.H TM](http://commdocs.house.gov/committees/science/hsy73325.000/hsy73325_0.H TM)
- Chapman, B., Johnson, A., Graziani, R., Horn, E., Large, A., & Rufi, W. (2005). *CCNA 1 and 2 Companion Guide* (3 ed.). Indiana, Indianapolis, USA: Cisco Press.
- Downing, M. (2005). *Spring Forward: The Annual Madness of Daylight Saving Time*. Shoemaker & Hoard.
- Edwards, J. (2009). *Build Your Own Firefox Extension*. Australia: SitePoint Pty. Ltd.
- Feldt, K. C. (2007). *Programming Firefox* (1st ed.). Sebastopol, California, USA: O'Reilly.
- Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999, June). Hypertext Transfer Protocol -- HTTP/1.1. Retrieved March 4, 2011, from <http://www.w3.org/>: <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- Flanagan, D. (2006). *JavaScript: The Definitive Guide, Fifth Edition* (5th ed.). O'Reilly.
- Gecko FAQ. (n.d.). (Mozilla Foundation) Retrieved from Mozilla Development Network: [https://developer.mozilla.org/en/Gecko\\_FAQ](https://developer.mozilla.org/en/Gecko_FAQ)
- Gibbs, G. (2007). Dictionary of New Zealand Biography. *Hudson, George Vernon 1867–1946*. Retrieved from <http://www.teara.govt.nz/en/biographies/3h42/1>
- Gourley, D., & Totty, B. (2002). *HTTP The Definitive Guide* (First ed.). O'REILLY.

IEEE. (2004). IEEE Std 1003.1. Retrieved July 2011, from [http://www.unix.org/version3/ieee\\_std.html](http://www.unix.org/version3/ieee_std.html)

Klyne, G., & Newman, C. (2002, July). Date and Time on the Internet: Timestamps. Retrieved July 23, 2011, from <http://www.ietf.org/rfc/rfc3339.txt>

*mozdev.org*. (n.d.). Retrieved July 2011, from [books.mozdev.org: http://books.mozdev.org/html/mozilla-chp-8.html](http://books.mozdev.org:books.mozdev.org/html/mozilla-chp-8.html)

Olson, A. D. (1986). *tz database*. Retrieved September 2011, from [http://en.wikipedia.org/wiki/Tz\\_database](http://en.wikipedia.org/wiki/Tz_database)

*Venkman*. (n.d.). Retrieved 9 2011, from [developer.mozilla.org: https://developer.mozilla.org/en/Venkman](http://developer.mozilla.org:developer.mozilla.org/en/Venkman)

## Παραρτήματα

---

### A. Προσχέδιο: “Πληροφορίες ζώνης ώρας στο HTTP”

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: January 28, 2010

S. Harhalakis  
TEI of Thessaloniki  
July 27, 2009

Timezone Information in HTTP  
draft-sharhalakis-httpztz-05.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on January 28, 2010.

#### Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Internet-Draft

Timezone Information in HTTP

July 2009

#### Abstract

This document defines a HTTP header for clients to provide timezone information to web servers. An ABNF description of the corresponding header is provided.

Harhalakis

Expires January 28, 2010

[Page 2]

## Discussion

Discussion about this document takes place in `http-wg` mailing list (`ietf-http-wg@w3.org`). Please CC `v13@v13.gr` too.

## Table of Contents

1. Introduction . . . . .	4
1.1. Purpose . . . . .	4
1.2. Requirements . . . . .	4
1.3. Terminology . . . . .	4
1.4. Considerations . . . . .	5
2. Definition . . . . .	7
2.1. Client support . . . . .	7
2.2. Server support . . . . .	8
2.3. Proxy considerations . . . . .	8
3. Security Considerations . . . . .	9
3.1. Client side . . . . .	9
3.2. Server side . . . . .	9
4. IANA Considerations . . . . .	10
5. Acknowledgements . . . . .	11
6. References . . . . .	12
6.1. Normative . . . . .	12
6.2. Informative . . . . .	12
Appendix A. Examples . . . . .	13
Author's Address . . . . .	14

## 1. Introduction

### 1.1. Purpose

Many web based applications could benefit from knowing the timezone of their visiting clients. Most of the dynamic content provider applications depend on user accounts to display time and date in the client's native timezone. This is a common problem and the current solution is not always possible or adequate for reasons including:

- o There may be no user accounts involved. News sites or RSS feeds for example don't require users to have accounts.
- o People may travel across timezone boundaries. Currently they need to update their web accounts to reflect their actual timezone information.

Furthermore, this is also the case for all HTTP-like or HTTP based protocols that make use of timestamps.

This document addresses this need by describing a header to be used by HTTP [RFC2616] so that interested clients may provide their current timezone information to web servers and thus to web based applications.

At the time of this writing GPS-enabled Internet access devices like mobile phones are being deployed. In the near future, those devices may be able to use GPS information and properly adjust their timezone information. This could be of great help for people that travel across timezones.

### 1.2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

An implementation is not compliant if it fails to satisfy one or more of the MUST or REQUIRED level requirements. An implementation that satisfies all the MUST or REQUIRED level and all the SHOULD level requirements is said to be "unconditionally compliant"; one that satisfies all the MUST level requirements but not all the SHOULD level requirements is said to be "conditionally compliant".

### 1.3. Terminology

This document uses the following terms:

**HTTP client**

Every client of the HTTP protocol. Commonly referred to as a web browser.

**Timezone string**

A timezone string as described in this document.

**HTTP header**

An HTTP header as described in [RFC2616].

The HTTP header specification of this document is presented in the augmented Backus-Naur Form that is described in [RFC2616].

#### 1.4. Considerations

Because of the variety of systems on the Internet and the non-technical nature of timezone, there is no simple method for a client to provide timezone information to HTTP servers. During the writing of this document the following were considered:

- o Simplicity is a must. The specification needs to be as simple as possible or allow for partial handling in a simple manner.
- o There is a variety of timezone styles. Some countries don't experience Daylight Saving Time (DST). Other countries have very unstable DST.
- o Politics affect DST. This makes it a moving target.
- o DST settings may change from year to year. For example, in 2007 the DST start and end time in some states in USA were changed.
- o Complete past and future timezone information cannot be described using a simple string.
- o End user systems may have an invalid timezone configured.
- o There are systems without timezone information.
- o Not all systems have a time source.

This document tries to provide adequate data for applications that can take advantage of the Timezone information to fulfill their needs. There are different levels of requirements that applications may have or HTTP clients may be able to satisfy. Thus this document introduces a way for HTTP clients to provide:

- o The current client time.
- o The current time offset.
- o The current year's timezone information.
- o Complete timezone information.

depending on their abilities.

Simple server side applications may just use the current time offset and hope that it is correctly set. More advanced applications (like calendars) need to know when a DST change will occur to correctly represent future or past times. Even more advanced applications may need to know the exact client timezone which can only be described using a reference to a timezone database entry. Finally, applications may choose to use the client's current time as reference time and optionally ignore client timezone information.

An issue that was discussed when proposing this header was the need of its existence. The alternative approach is to leave the time and timezone information handling to server side applications which can use geo-ip, user accounts, etc. This is not an option because:

- o Geo-ip is not suitable for HTTPS clients behind a proxy.
- o User accounts are not efficient for public pages like news sites, blogs, forums, search engines, etc.

HTTP-based protocols may also take advantage of this feature.

Another consideration is the validity of client-side time or timezone information. It seems that proper timezone information is gradually being adopted as common behavior. Even though this document attempts to help clients without timezone information and provide ultimate flexibility to server side applications, the validity of the timezone information is considered an existing precondition. This is in accordance with paragraph 4.4 of [RFC3339] "Unqualified Local Time" which considers systems with invalid timezone information as inappropriate for Internet communications.



## 2. Definition

### 2.1. Client support

HTTP clients MAY provide local timezone information to visiting web sites. This information is sent using the client-timezone HTTP header:

```
client-timezone = "Timezone" ":" tzdescr
tzdescr          = rfctime [ ";" posixtz [ ";" tznames ] ]
rfctime         = <local time>
posixtz        = <a POSIX 1003.1 timezone string>
tznames        = tzname *( "," tzname )
tzname         = <a timezone string from a timezone database>
```

Where:

**rfctime** A local time string as specified in [RFC3339]. It includes the current time offset.

**posixtz** A POSIX 1003.1 timezone string as specified in [POSIXTZ].

**tzname** A timezone name. Many systems use a timezone name from a timezone database. This is one such string. More than one such strings may be provided. All of them MUST identify the same timezone.

HTTP clients SHOULD provide all three parts of the header unless they aren't able to do so or they are configured not to send timezone information.

If a client does not provide a part of the header and there is another part following that is provided, the client MUST use the empty string for the part that is not provided.

Clients that don't have a clock source MUST use zeroes for all digits of the datetime string except from the offset. "Z" MUST be used when the offset is also unavailable.

When multiple references to timezone databases are specified the server-side SHOULD perform a left-to-right search. This means that the server-side application first searches its database for the first

provided timezone. If the search fails the search continues for the second timezone, etc. The search ends when a matching entry is found or there is no provided timezone database reference to search for. The client side MUST assume that the server side uses this search method.

## 2.2. Server support

Compliant servers MAY validate the format of the provided information. Timezone strings that are not in a valid format MAY not be accepted. Validation checks MUST NOT be performed on the content of the Timezone string by servers. Only the format of the string may be checked. This way outdated servers will not filter out proper information.

## 2.3. Proxy considerations

HTTP proxy servers MUST NOT alter this information. HTTP proxy servers MUST follow the same validation policy that was specified for servers.

Server side scripts that produce customized results based on the timezone information MUST return an appropriate "Vary" header as specified in paragraph 14.44 of [RFC2616].

### 3. Security Considerations

#### 3.1. Client side

Timezone information may consist personal information regarding the location of a person. HTTP clients MUST NOT provide this information without letting the user prevent it. Clients must either ask users or provide an option to enable/disable this feature. The latter is RECOMMENDED.

In most cases Timezone information will not disclose more personal information than an IP address. HTTP clients MAY default in enabling this to improve user experience. It is also possible that client behavior regarding timezone information disclosure be controlled by implicit privacy settings.

#### 3.2. Server side

Web based applications MUST treat this information as user input that may be invalid or malicious.

Internet-Draft

Timezone Information in HTTP

July 2009

#### 4. IANA Considerations

This specification requires registration of a Message Header Field for HTTP [RFC3864].

Header field: Timezone

Applicable protocol: HTTP

Status: Experimental

Author/change controller:  
IETF (iesg@ietf.org)  
Internet Engineering Task Force

Specification document:  
[ this document ]

## 5. Acknowledgements

Timezone information in HTTP was also proposed by David Robinson in an email at HTTP Working Group back in 1995 but the replies he got were negative. It was believed that timezone information should be handled by CGI scripts and not by the Hypertext Transfer Protocol. The discussion can be found at http-wg mailing list archives: <http://www.hpl.hp.com/personal/ange/archives/archives-95/http-wg-archive/0521.html>.

This document was properly formed thanks to the remarks of Julian Reschke.

## 6. References

### 6.1. Normative

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
- [RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.
- [POSIXTZ] IEEE, The Open Group, "IEEE Std 1003.1", 2004.

### 6.2. Informative

- [I-D.rfc-editor-rfc2223bis] Reynolds, J. and R. Braden, "Instructions to Request for Comments (RFC) Authors", draft-rfc-editor-rfc2223bis-08 (work in progress), July 2004.

## Appendix A. Examples

A complete timezone information header using the Olson timezone database:

Assuming that current time is 12:00, Jul 30, 1977. Time offset is +2 hours. The full timezone string in [POSIXTZ] format is EET2EEST3,M3.2.0/02:00,M11.1.0/02:00. The timezone database identifier string is Europe/Athens:

```
Timezone: 1977-07-30T12:00+0200;  
          EET2EEST3,M3.2.0/02:00,M11.1.0/02:00;  
          Europe/Athens
```

A partial one:

GMT/UTC timezone is noted as Z(ulu).

```
Timezone: 2007-06-12T23:48Z
```

A header from a clock-less client:

```
Timezone: 0000-00-00T00:00+0200;  
          EET2EEST3,M3.2.0/02:00,M11.1.0/02:00;  
          Europe/Athens
```

A header with multiple database references (Olson and Microsoft Windows timezone databases):

```
Timezone: 1977-07-30T12:00+0200;  
          EET2EEST3,M3.2.0/02:00,M11.1.0/02:00;  
          Europe/Athens;(GMT+02:00) Athens, Istanbul, Minsk
```

A header from a clock-less client without current offset information:

```
Timezone: 0000-00-00T00:00Z
```

A header with a missing part:

```
Timezone: 1977-07-30T12:00+0200;;Europe/Athens
```

Internet-Draft

Timezone Information in HTTP

July 2009

Author's Address

Stefanos Harhalakis  
Technological Educational Institute of Thessaloniki  
Department of Information Technology  
Thessaloniki, Greece  
GR

Email: v13@v13.gr, v13@it.teithe.gr



#### Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

#### Acknowledgment

This document was produced using `xml2rfc v1.33` (of <http://xml.resource.org/>) from a source in RFC-2629 XML format.

## B. Πηγαίος Κώδικας Plugin

### Αρχείο install.rdf

```
<?xml version="1.0"?>

<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:em="http://www.mozilla.org/2004/em-rdf#">

  <Description about="urn:mozilla:install-manifest">
    <!-- Add on details definitions -->
    <em:creator>George Erpimidis</em:creator>
      <em:contributor>Stefanos Harhalakis</em:contributor>
    <em:description>Set a timezone header, based on local timezone detection.</em:description>
    <em:homepageURL>http://www.it.teithe.gr/~gerpis/Timezone_Header.html</em:homepageURL>
    <em:iconURL>chrome://timezoneheader/content/images/TZlogo.png</em:iconURL>

    <em:id>gerpis@gmail.com</em:id>
    <em:name>Timezone Header</em:name>
    <em:unpack>true</em:unpack>
    <em:optionsURL>chrome://timezoneheader/content/gen_overlay.xul</em:optionsURL>
    <em:version>1.0.3</em:version>

    <!-- Firefox 3.0 - 3.* -->
    <!-- Due to changes in XPCOM, Firefox prior to version 3 is no longer supported -->

    <em:targetApplication>
      <Description>
        <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
        <em:minVersion>4.0.*</em:minVersion>
        <em:maxVersion>6.0.*</em:maxVersion>
      </Description>
    </em:targetApplication>
  </Description>
</RDF>
```

## Αρχείο chrome.manifest

```
#Firefox
overlay chrome://browser/content/browser.xul      chrome://timezoneheader/content/overlay.xul

#Setting paths for packages
content timezoneheader chrome/content/
locale timezoneheader en-GB chrome/locale/en-GB/

#Registering Interfaces
interfaces components/mhIHeader.xpt
interfaces components/nsIModifyheaders.xpt

component {6b2f2fc7-a26c-4602-a08d-bd6d065a86e3} components/modifyheaders-service.js
contract @modifyheaders.mozdev.org/header;1 {6b2f2fc7-a26c-4602-a08d-bd6d065a86e3}

component {feb80fc3-9e72-4fc5-bc72-986957ada6cc} components/modifyheaders-service.js
contract @modifyheaders.mozdev.org/service;1 {feb80fc3-9e72-4fc5-bc72-986957ada6cc}

component {0eff9eeb-c51a-4f07-9823-27bc32fdae13} components/modifyheaders-service.js
contract @modifyheaders.mozdev.org/proxy;1 {0eff9eeb-c51a-4f07-9823-27bc32fdae13}

# The category registration is required for the Proxy service
category profile-after-change ModifyHeadersProxy @modifyheaders.mozdev.org/proxy;1
```

## Αρχείο overlay.xul

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://timezoneheader/content/TZheader.css" type="text/css"?>
<!DOCTYPE window SYSTEM "chrome://timezoneheader/locale/tzheaderlocale.dtd">

<overlay xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <script type="application/x-javascript" src="chrome://timezoneheader/content/TZheader.js"/>

  <menupopup id="menu_ToolsPopup" >
    <menuitem id="menu_ToolsPopup_TZheader" insertbefore="sanitizeSeparator"
label="Timezone Header" class="menuitem-iconic" oncommand="TZheader.open()" />
  </menupopup>

</overlay>
```

### Αρχείο gen\_overlay.xul

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/global.css"?>
<?xml-stylesheet href="chrome://timezoneheader/content/TZheader.css" type="text/css"?>
<!DOCTYPE window SYSTEM "chrome://timezoneheader/locale/tzheaderlocale.dtd">
<prefwindow id="TZheader-window"
  buttons="accept"
  title="&tzheader.main.title;"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">

  <prefpane id="settings-pane" label="Settings" selected="true"
  src="chrome://timezoneheader/content/settings_overlay.xul" flex="1"/>

  <script type="application/x-javascript" src="chrome://timezoneheader/content/TZheader.js"/>
</prefwindow>
```

### Αρχείο settings\_overlay.xul

```
<?xml version="1.0"?>
<!DOCTYPE window SYSTEM "chrome://timezoneheader/locale/tzheaderlocale.dtd">
<overlay id="settings_overlay"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <prefpane id="settings-pane" label="Timezone Settings" onpanelload="TZheader.initSettingsPane()">
    <stringbundleset>
      <stringbundle id="tzheaderstrBundle"
        src="chrome://timezoneheader/locale/tzheaderlocale.properties"/>
    </stringbundleset>
  </prefpane >
</overlay >
```

## Αρχείο TZheader.js

```
var TZheader = {  
  
    //tzheadersService:  
    Components.classes["@mozilla.org/timezoneheader/service;1"].getService(Components.interfaces.nsi  
    Modifyheaders),  
  
    open: function () {  
  
        this.mainWindow =  
        window.openDialog("chrome://timezoneheader/content/gen_overlay.xul", "TZheader",  
        "chrome,all,dialog=no");  
  
    },  
  
    initSettingsPane: function(){  
  
        /*  
  
            Function that dynamically creates the menu inside prefpane  
  
        */  
  
        //Create newmenu, set a reference to xul element with id = "settings-pane"  
        var newmenu = document.getElementById("settings-pane");  
  
        //Implement groupbox, caption required  
        var groupbox = document.createElement("groupbox");  
  
        //Implement the caption of groupbox  
        var caption = document.createElement("caption");  
        caption.setAttribute("id", "TZgroup");  
  
        caption.setAttribute("label", document.getElementById("tzheaderstrBundle").getString("tzheade  
r.label.caption.TZgroup"));  
  
        //Implement preferences tag and preference element  
        //<preferences>  
        //    <preference id=""alwaysOn" name="timezoneheader.config.alwaysOn"  
type="bool" instantApply="true" />  
        //</preferences>  
  
        var prefsElem=document.createElement("preferences");
```

```
var posixPref=document.createElement("preference");
posixPref.setAttribute("id","alwaysOn");
posixPref.setAttribute("name","timezoneheader.config.alwaysOn");
posixPref.setAttribute("type","bool");
posixPref.setAttribute("instantApply","true");
prefsElem.appendChild(posixPref);

//Implement checkbox, notice the connection to alwaysOn preference
var posixchBox=document.createElement("checkbox");
posixchBox.setAttribute("id","posixchBox");

    posixchBox.setAttribute("label",document.getElementById("tzheaderstrBundle").getString("tzheader.label.posixchBox"));

    posixchBox.setAttribute("preference","alwaysOn");

    posixchBox.setAttribute("tooltip",document.getElementById("tzheaderstrBundle").getString("tzheader.tooltip.posixchBox"));

//Set var for preference call in about:config, branch: timezoneheader.
var config_prefs = Components.classes["@mozilla.org/preferences-service;1"]
.getService(Components.interfaces.nsIPrefService).getBranch("timezoneheader.");

if (!config_prefs.prefHasUserValue("config.alwaysOn")) {
    config_prefs.setBoolPref("config.alwaysOn", "false");
}

if (!config_prefs.prefHasUserValue("config.tztimestamp")) {
    config_prefs.setCharPref("config.tztimestamp", "0");
}

if (!config_prefs.prefHasUserValue("config.posixvalue")) {
    config_prefs.setCharPref("config.posixvalue", "0000-00-
```

```

00T00:00:00:+0:00:Undefined:false");
    }

    //Check for timezoneheader.config.alwaysOn in about.config. If true, posixchBox will
    be checked.

    if (config_prefs.prefHasUserValue("config.alwaysOn") &&
    config_prefs.getBoolPref("config.alwaysOn")) {

        posixchBox.setAttribute("checked",config_prefs.getBoolPref("config.alwaysOn"));

    }

    //Check config.alwaysOn if true, output stored posixvalue that will be send as header.

    if(config_prefs.prefHasUserValue("config.alwaysOn") &&
    config_prefs.getBoolPref("config.alwaysOn")){

        //dump("\n"+config_prefs.getCharPref("config.posixvalue").match("Undefined")+"\n");

        if (config_prefs.prefHasUserValue("config.posixvalue") &&
        config_prefs.getCharPref("config.posixvalue").match("Undefined")!= "Undefined"){

            var
            posixtz=document.getElementById("tzheaderstrBundle").getString("tzheader.label.posixLabel") + "" +
            config_prefs.getCharPref("config.posixvalue");

            }

            else{

                var
                posixtz=document.getElementById("tzheaderstrBundle").getString("tzheader.label.posixLabel.undefine
                d");

            }

        }

        else {

            var
            posixtz=document.getElementById("tzheaderstrBundle").getString("tzheader.label.posixLabel.disabled"
            );

        }

        dump(posixtz);

        var posixLabel= document.createElement("label");
    
```

```
        posixLabel.setAttribute("id","posixLabel");
        posixLabel.setAttribute("value",posixtz);
        posixLabel.setAttribute("disabled","false");

        posixLabel.setAttribute("tooltiptext",document.getElementById("tzheaderstrBundle").getString("
tzheader.tooltip.posixLabel"));

        //Create XuL hierarchy dynamically
        //<prefpane id="settings-pane">
        //    <groupbox/>
        //    <caption/>
        //    <preferences>
        //        <preference/>
        //    </preferences>
        //    <checkBox />
        //    <label/>
        //</prefpane>
        newmenu.appendChild(groupbox);
        groupbox.appendChild(caption);
        groupbox.appendChild(prefsElem);
        groupbox.appendChild(posixchBox);
        groupbox.appendChild(posixLabel);
        this.refreshSelection();
    }
};
```



## Αρχείο modifyheaders-service.js

```
if (!TimezoneHeader)
    var TimezoneHeader = {};

if (!TimezoneHeader.Proxy) {
    Components.utils.import("resource://gre/modules/XPCOMUtils.jsm");
    dump("\nTimezoneHeader.Proxy\n");
    TimezoneHeader.Proxy = function () {
    };
    TimezoneHeader.Proxy.prototype = {
        classDescription: "Modify Headers Proxy",
        classID:          Components.ID("{0eff9eeb-c51a-4f07-9823-27bc32fdae13}"),
        contractID:       "@modifyheaders.mozdev.org/proxy;1",

        QueryInterface: XPCOMUtils.generateQI([Components.interfaces.nsIObserver]),
        grabTz: function (){

            var d=new Date()

            /* Optional Calculation for GMT Offset locally. I.e. if gmtHours = 3, aOffset will
            be "+3:00", formatted appropriately.

            var gmtHours
            if(-d.getTimezoneOffset()%60==0){
                gmtHours=-d.getTimezoneOffset()/60;
                if (gmtHours>=0)
                    gmtHours="+"+gmtHours+":00" ;
                else
                    gmtHours+=":00";
            }
            else if(-d.getTimezoneOffset()%60!=0){
```

```
gmtHours=parseInt(-d.getTimezoneOffset()/60) ;

if (gmtHours>=0){
    gmtHours="+"+gmtHours ;
}
var aMod=-d.getTimezoneOffset()%60;
    gmtHours=gmtHours+ ":" + Math.abs(aMod);
}
var aOffset = gmtHours;
*/
var aDate=d.getDate();
var aMonth=d.getMonth() + 1;
var aHours=d.getHours();
var aMinutes=d.getMinutes();
var aSeconds=d.getSeconds();

if (aMonth<=9)aMonth = "0" + aMonth;
if (aDate<=9)aDate = "0" + aDate;
if (aHours<=9)aHours = "0" + aHours;
if (aMinutes<=9)aMinutes = "0" + aMinutes;
if (aSeconds<=9)aSeconds = "0" + aSeconds;

var afullDate = d.getFullYear() + "-" + aMonth + "-" + aDate;
var afullTime = aHours + ":" + aMinutes + ":" + aSeconds;

var resgrabTz = afullDate + "T" + afullTime;

return resgrabTz;
```

```
    },

    getlocalTz: function (){
        var fullTz
        try
        {
            fullTz=this.grabTz();
        }
        catch(err)
        {
            alert("Clock functions could not be initiated");
            fullTz="0000-00-00T00:00Z";
        }
        return fullTz;
    },

    _xpcom_categories: [{
        category: "profile-after-change",
        entry: "Modify Headers Proxy"
    }],

    // nsIObserver interface method

    observe: function (subject, topic, data) {
        //dump("\nnsIObserver interface method\n-----"+ "\nSubject: " + subject +
        "\nTopic: " + topic + "\nData" + data);

        if (topic == 'http-on-modify-request') {
            subject.QueryInterface(Components.interfaces.nsIHttpChannel);
            var prefs = Components.classes["@mozilla.org/preferences-
```

```

service;1"]
        .getService(Components.interfaces.nsIPrefService).getBranch("timezoneheader.");

        //Check for alwaysOn Preference in about.config
        if (prefs.prefHasUserValue("config.alwaysOn") &&
prefs.getBoolPref("config.alwaysOn")) {
        calculation
                //Fetch preference timestamp preference for time difference

                //with the current localtime.
                if(prefs.prefHasUserValue("config.tztimestamp")){

                preftimeStamp=prefs.getCharPref("config.tztimestamp");

                //Set tztimeStamp and posixvalue if difference is
greater than 60 Seconds.
                if(checktimeSecs(preftimeStamp,60)){
                        var d=new Date();
                        tztimeStamp=parseInt(d.getTime()/1000);
                        prefs.setCharPref("config.tztimestamp",
tztimeStamp);

                        var
output=determine_timezone().timezone.display();
                        myHeader=this.getLocalTz() + ":" +
output["utc_offset"] + ":" + output["olson_tz"] + ":" + output["uses_dst"];
                        prefs.setCharPref("config.posixvalue",
myHeader);
                }
                }
                myHeader=prefs.getCharPref("config.posixvalue");
                subject.setRequestHeader("Timezone", myHeader, "true");
        }
    } else if (topic == 'profile-after-change') {

```

```
        if ("nsINetModuleMgr" in Components.interfaces) {
                // Should be an old version of Mozilla (before september 15,
2003
                // Do Nothing as these old versions of firefox (firebird, phoenix
etc) are not supported
        } else {
                // Should be a new version of Mozilla (after september 15,
2003)
                var observerService =
Components.classes["@mozilla.org/observer-
service;1"].getService(Components.interfaces.nsIObserverService);
                observerService.addObserver(this, "http-on-modify-request",
false);
        }
        } else {
                //dump("\nNo observable topic defined");
        }
    }

};
}

var HEMISPHERE_SOUTH = 'SOUTH';
var HEMISPHERE_NORTH = 'NORTH';
var HEMISPHERE_UNKNOWN = 'N/A';
var olson = {}

/**
 * The keys in this dictionary are comma separated as such:
 *
 * First the offset compared to UTC time in minutes.
```

```
*
* Then a flag which is 0 if the timezone does not take daylight savings into account and 1 if it does.
*
* Thirdly an optional 's' signifies that the timezone is in the southern hemisphere, only interesting for
timezones with DST.
*
* The values of the dictionary are TimeZone objects.
*/
olson.timezones = {
    '-720,0' : new TimeZone('-12:00','Etc/GMT+12', false),
    '-660,0' : new TimeZone('-11:00','Pacific/Pago_Pago', false),
    '-600,1' : new TimeZone('-11:00','America/Adak',true),
    '-660,1,s' : new TimeZone('-11:00','Pacific/Apia', true),
    '-600,0' : new TimeZone('-10:00','Pacific/Honolulu', false),
    '-570,0' : new TimeZone('-10:30','Pacific/Marquesas',false),
    '-540,0' : new TimeZone('-09:00','Pacific/Gambier',false),
    '-540,1' : new TimeZone('-09:00','America/Anchorage', true),
    '-480,1' : new TimeZone('-08:00','America/Los_Angeles', true),
    '-480,0' : new TimeZone('-08:00','Pacific/Pitcairn',false),
    '-420,0' : new TimeZone('-07:00','America/Phoenix', false),
    '-420,1' : new TimeZone('-07:00','America/Denver', true),
    '-360,0' : new TimeZone('-06:00','America/Guatemala', false),
    '-360,1' : new TimeZone('-06:00','America/Chicago', true),
    '-360,1,s' : new TimeZone('-06:00','Pacific/Easter',true),
    '-300,0' : new TimeZone('-05:00','America/Bogota', false),
    '-300,1' : new TimeZone('-05:00','America/New_York', true),
    '-270,0' : new TimeZone('-04:30','America/Caracas', false),
    '-240,1' : new TimeZone('-04:00','America/Halifax', true),
    '-240,0' : new TimeZone('-04:00','America/Santo_Domingo', false),
    '-240,1,s' : new TimeZone('-04:00','America/Asuncion', true),
```

```
'-210,1' : new TimeZone('-03:30','America/St_Johns', true),
'-180,1' : new TimeZone('-03:00','America/Godthab', true),
'-180,0' : new TimeZone('-03:00','America/Argentina/Buenos_Aires', false),
'-180,1,s' : new TimeZone('-03:00','America/Montevideo', true),
'-120,0' : new TimeZone('-02:00','America/Noronha', false),
'-120,1' : new TimeZone('-02:00','Etc/GMT+2', true),
'-60,1' : new TimeZone('-01:00','Atlantic/Azores', true),
'-60,0' : new TimeZone('-01:00','Atlantic/Cape_Verde', false),
'0,0' : new TimeZone('00:00','Etc/UTC', false),
'0,1' : new TimeZone('00:00','Europe/London', true),
'60,1' : new TimeZone('+01:00','Europe/Berlin', true),
'60,0' : new TimeZone('+01:00','Africa/Lagos', false),
'60,1,s' : new TimeZone('+01:00','Africa/Windhoek',true),
'120,1' : new TimeZone('+02:00','Asia/Beirut', true),
'120,0' : new TimeZone('+02:00','Africa/Johannesburg', false),
'180,1' : new TimeZone('+03:00','Europe/Moscow', true),
'180,0' : new TimeZone('+03:00','Asia/Baghdad', false),
'210,1' : new TimeZone('+03:30','Asia/Tehran', true),
'240,0' : new TimeZone('+04:00','Asia/Dubai', false),
'240,1' : new TimeZone('+04:00','Asia/Yerevan', true),
'270,0' : new TimeZone('+04:30','Asia/Kabul', false),
'300,1' : new TimeZone('+05:00','Asia/Yekaterinburg', true),
'300,0' : new TimeZone('+05:00','Asia/Karachi', false),
'330,0' : new TimeZone('+05:30','Asia/Kolkata', false),
'345,0' : new TimeZone('+05:45','Asia/Kathmandu', false),
'360,0' : new TimeZone('+06:00','Asia/Dhaka', false),
'360,1' : new TimeZone('+06:00','Asia/Omsk', true),
'390,0' : new TimeZone('+06:30','Asia/Rangoon', false),
'420,1' : new TimeZone('+07:00','Asia/Krasnoyarsk', true),
```

```
'420,0' : new TimeZone('+07:00','Asia/Jakarta', false),
'480,0' : new TimeZone('+08:00','Asia/Shanghai', false),
'480,1' : new TimeZone('+08:00','Asia/Irkutsk', true),
'525,0' : new TimeZone('+08:45','Australia/Eucla', true),
'525,1,s' : new TimeZone('+08:45','Australia/Eucla', true),
'540,1' : new TimeZone('+09:00','Asia/Yakutsk', true),
'540,0' : new TimeZone('+09:00','Asia/Tokyo', false),
'570,0' : new TimeZone('+09:30','Australia/Darwin', false),
'570,1,s' : new TimeZone('+09:30','Australia/Adelaide', true),
'600,0' : new TimeZone('+10:00','Australia/Brisbane', false),
'600,1' : new TimeZone('+10:00','Asia/Vladivostok', true),
'600,1,s' : new TimeZone('+10:00','Australia/Sydney', true),
'630,1,s' : new TimeZone('+10:30','Australia/Lord_Howe', true),
'660,1' : new TimeZone('+11:00','Asia/Kamchatka', true),
'660,0' : new TimeZone('+11:00','Pacific/Noumea', false),
'690,0' : new TimeZone('+11:30','Pacific/Norfolk', false),
'720,1,s' : new TimeZone('+12:00','Pacific/Auckland', true),
'720,0' : new TimeZone('+12:00','Pacific/Tarawa', false),
'765,1,s' : new TimeZone('+12:45','Pacific/Chatham', true),
'780,0' : new TimeZone('+13:00','Pacific/Tongatapu', false),
'840,0' : new TimeZone('+14:00','Pacific/Kiritimati', false)
}

/**
 * This object contains information on when daylight savings starts for
 * different timezones.
 *
 * The list is short for a reason. Often we do not have to be very specific
 * to single out the correct timezone. But when we do, this list comes in
```



\* handy.

\*

\* Each value is a date denoting when daylight savings starts for that timezone.

\*/

```
olson.dst_start_dates = {  
    'America/Denver' : new Date(2011, 2, 13, 3, 0, 0, 0),  
    'America/Mazatlan' : new Date(2011, 3, 3, 3, 0, 0, 0),  
    'America/Chicago' : new Date(2011, 2, 13, 3, 0, 0, 0),  
    'America/Mexico_City' : new Date(2011, 3, 3, 3, 0, 0, 0),  
    'Atlantic/Stanley' : new Date(2011, 8, 4, 7, 0, 0, 0),  
    'America/Asuncion' : new Date(2011, 9, 2, 3, 0, 0, 0),  
    'America/Santiago' : new Date(2011, 9, 9, 3, 0, 0, 0),  
    'America/Campo_Grande' : new Date(2011, 9, 16, 5, 0, 0, 0),  
    'America/Montevideo' : new Date(2011, 9, 2, 3, 0, 0, 0),  
    'America/Sao_Paulo' : new Date(2011, 9, 16, 5, 0, 0, 0),  
    'America/Los_Angeles' : new Date(2011, 2, 13, 8, 0, 0, 0),  
    'America/Santa_Isabel' : new Date(2011, 3, 5, 8, 0, 0, 0),  
    'America/Havana' : new Date(2011, 2, 13, 2, 0, 0, 0),  
    'America/New_York' : new Date(2011, 2, 13, 7, 0, 0, 0),  
    'Asia/Gaza' : new Date(2011, 2, 26, 23, 0, 0, 0),  
    'Asia/Beirut' : new Date(2011, 2, 27, 1, 0, 0, 0),  
    'Europe/Minsk' : new Date(2011, 2, 27, 3, 0, 0, 0),  
    'Europe/Athens,Istanbul' : new Date(2011, 2, 27, 7, 0, 0, 0),  
    'Asia/Damascus' : new Date(2011, 3, 1, 2, 0, 0, 0),  
    'Asia/Jerusalem' : new Date(2011, 3, 1, 6, 0, 0, 0),  
    'Africa/Cairo' : new Date(2011, 3, 29, 4, 0, 0, 0),  
    'Asia/Yerevan' : new Date(2011, 2, 27, 4, 0, 0, 0),  
    'Asia/Baku' : new Date(2011, 2, 27, 8, 0, 0, 0),  
    'Pacific/Auckland' : new Date(2011, 8, 26, 7, 0, 0, 0),
```

```
'Pacific/Fiji' : new Date(2010, 11, 29, 23, 0, 0, 0),
'America/Halifax' : new Date(2011, 2, 13, 6, 0, 0, 0),
'America/Goose_Bay' : new Date(2011, 2, 13, 2, 1, 0, 0),
'America/Miquelon' : new Date(2011, 2, 13, 5, 0, 0, 0),
'America/Godthab' : new Date(2011, 2, 27, 1, 0, 0, 0)
}

/**
 * The keys in this object are timezones that we know may be ambiguous after
 * a preliminary scan through the olson_tz object.
 *
 * The array of timezones to compare must be in the order that daylight savings
 * starts for the regions.
 */
olson.ambiguity_list = {
    'America/Denver' : ['America/Denver', 'America/Mazatlan'],
    'America/Chicago' : ['America/Chicago', 'America/Mexico_City'],
    'America/Asuncion' : ['Atlantic/Stanley', 'America/Asuncion',
'America/Santiago', 'America/Campo_Grande'],
    'America/Montevideo' : ['America/Montevideo', 'America/Sao_Paulo'],
    'Asia/Beirut' : ['Asia/Gaza', 'Asia/Beirut', 'Europe/Minsk', 'Europe/Athens', 'Istanbul',
'Asia/Damascus', 'Asia/Jerusalem', 'Africa/Cairo'],
    'Asia/Yerevan' : ['Asia/Yerevan', 'Asia/Baku'],
    'Pacific/Auckland' : ['Pacific/Auckland', 'Pacific/Fiji'],
    'America/Los_Angeles' : ['America/Los_Angeles', 'America/Santa_Isabel'],
    'America/New_York' : ['America/Havana', 'America/New_York'],
    'America/Halifax' : ['America/Goose_Bay', 'America/Halifax'],
    'America/Godthab' : ['America/Miquelon', 'America/Godthab']
}
```

```
/**
 * A simple object containing information of utc_offset, which olson timezone key to use,
 * and if the timezone cares about daylight savings or not.
 *
 * @constructor
 * @param {string} offset - for example '-11:00'
 * @param {string} olson_tz - the olson Identifier, such as "America/Denver"
 * @param {boolean} uses_dst - flag for whether the time zone somehow cares about daylight savings.
 */
function TimeZone(offset, olson_tz, uses_dst) {
    this.utc_offset = offset;
    this.olson_tz = olson_tz;
    this.uses_dst = uses_dst;
}

/**
 * Prints out the result.
 * But before it does that, it calls this.ambiguity_check.
 */
TimeZone.prototype.display = function() {
    this.ambiguity_check();
    var response_text = new Array();
    response_text["utc_offset"] = this.utc_offset;
    response_text["olson_tz"] = this.olson_tz;
    response_text["uses_dst"] = (this.uses_dst ? 'yes' : 'no');

    return response_text;
}
```

```
/**
 * Checks if a timezone has possible ambiguities. I.e timezones that are similar.
 *
 * If the preliminary scan determines that we're in America/Denver. We double check
 * here that we're really there and not in America/Mazatlan.
 *
 * This is done by checking known dates for when daylight savings start for different
 * timezones.
 */
TimeZone.prototype.ambiguity_check = function() {
    var local_ambiguity_list = olson.ambiguity_list[this.olson_tz];

    if (typeof(local_ambiguity_list) == 'undefined') {
        return;
    }

    var length = local_ambiguity_list.length;

    for (var i = 0; i < length; i++) {
        var tz = local_ambiguity_list[i]

        if (date_is_dst(olson.dst_start_dates[tz])) {
            this.olson_tz = tz;
            return;
        }
    }
}
```

```
/**
 * Checks whether a given date is in daylight savings time.
 *
 * If the date supplied is after june, we assume that we're checking
 * for southern hemisphere DST.
 *
 * @param {Date} date
 * @returns {boolean}
 */
function date_is_dst(date) {
    var base_offset = ( date.getMonth() > 5 ? get_june_offset() : get_january_offset() )

    var date_offset = get_date_offset(date);

    return (base_offset - date_offset) != 0;
}

/**
 * Gets the offset in minutes from UTC for a certain date.
 *
 * @param date
 * @returns {number}
 */
function get_date_offset(date) {
    return -date.getTimezoneOffset();
}

/**
 * This function does some basic calculations to create information about
```

```
* the user's timezone.
*
* Returns a primitive object on the format
* {'utc_offset' : -9, 'dst': 1, 'hemisphere' : 'north'}
* where dst is 1 if the region uses daylight savings.
*
* @returns {Object}
*/
function get_timezone_info() {
    var january_offset = get_january_offset();

    var june_offset = get_june_offset();

    var diff = january_offset - june_offset;

    if (diff < 0) {
        return {'utc_offset' : january_offset,
                'dst': 1,
                'hemisphere' : HEMISPHERE_NORTH}
    }
    else if (diff > 0) {
        return {'utc_offset' : june_offset,
                'dst' : 1,
                'hemisphere' : HEMISPHERE_SOUTH}
    }

    return {'utc_offset' : january_offset,
            'dst': 0,
            'hemisphere' : HEMISPHERE_UNKNOWN}
}
```

```
}

function get_january_offset() {
    return get_date_offset(new Date(2011, 0, 1, 0, 0, 0, 0));
}

function get_june_offset() {
    return get_date_offset(new Date(2011, 5, 1, 0, 0, 0, 0));
}

/**
 * Uses get_timezone_info() to formulate a key to use in the olson.timezones dictionary.
 *
 * Returns a primitive object on the format:
 * {'timezone': TimeZone, 'key' : 'the key used to find the TimeZone object'}
 *
 * @returns Object
 */
function determine_timezone() {
    var timezone_key_info = get_timezone_info();

    var hemisphere_suffix = ""

    if (timezone_key_info.hemisphere == HEMISPHERE_SOUTH) {
        hemisphere_suffix = ',s';
    }

    var tz_key = timezone_key_info.utc_offset + ',' + timezone_key_info.dst + hemisphere_suffix
```

```
        return {'timezone' : olson.timezones[tz_key], 'key' : tz_key}
    }

//Function to calculate the difference between the preference stored timestamp and the current time.
//Timethreshold used to define the difference in seconds allowed
function checktimeSecs(timeinput,timethreshold){
    //If true the difference is greater than the timethreshold.
    var d=new Date();
    //getTime() calculated in milliseconds 1970/01/01 and converted to seconds
    if ((checktimepassed=(parseInt(d.getTime()/1000) - parseInt(timeinput))>
timethreshold)?"True":"False")
        return checktimepassed;
    }

/* Entry point - registers the component with the browser */
if (XPCOMUtils.generateNSGetFactory) {
    var NSGetFactory = XPCOMUtils.generateNSGetFactory([TimezoneHeader.Proxy]);
} else {
    var NSGetModule = XPCOMUtils.generateNSGetModule([TimezoneHeader.Proxy]);
}
```

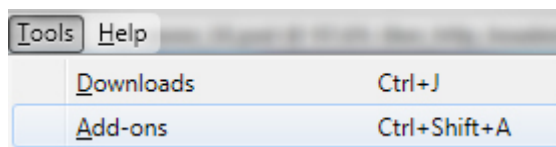


# Οδηγός Χρήσης Λογισμικού

## Εγκατάσταση

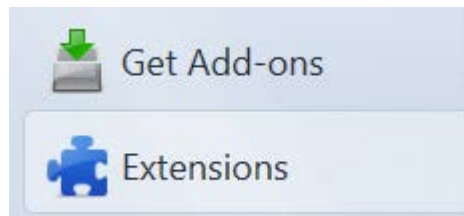
Η εγκατάσταση αυτού του plugin προορίζεται για τον **Mozilla Firefox** με έκδοση 4.\*.\* - 6.0.\* σε λειτουργικό περιβάλλον των **Windows**. Αν δεν τον έχετε εγκατεστημένο μπορείτε να τον κατεβάσετε από εδώ: <http://www.mozilla.org/>

- 1) Ανοίξτε τον browser **Mozilla Firefox**.
- 2) Επιλέξτε από την μπάρα μενού, στο πάνω μέρος, την επιλογή **Tools>Add-ons** όπως φαίνεται παρακάτω στο **Σχήμα 1**.



Σχήμα 1 Ανοιγμα Του Μενού Add-ons

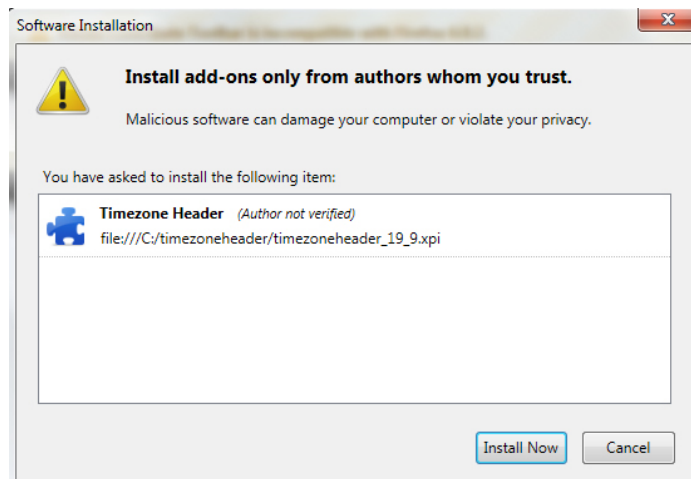
- 3) Στην οθόνη που θα ανοίξει επιλέξτε από το αριστερό μενού την επιλογή **Extensions** (Σχήμα 2), όπου και θα δείτε την οθόνη με όλα τα εγκατεστημένα Add-ons.



Σχήμα 2 Οθόνη Με Extensions

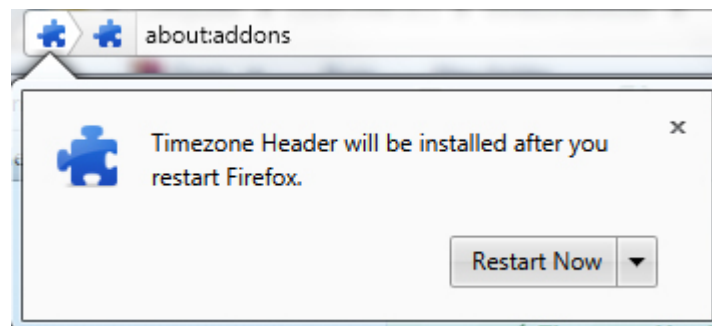
- 4) Για να συνεχίσετε, εντοπίστε το αρχείο **Timezoneheader.xpi** που βρίσκεται στο CD που παρέχεται με το παρών κείμενο.

- 5) Επιλέξτε το και σύρετέ το (Drag&Drop) στην οθόνη **Extensions** του **Βήματος 3**.
- 6) Περιμένετε 3 δευτερόλεπτα και επιλέξτε **Install Now** στον διάλογο που θα εμφανιστεί όπως φαίνεται στο Σχήμα 3



Σχήμα 3 Επιβεβαίωση Εγκατάστασης

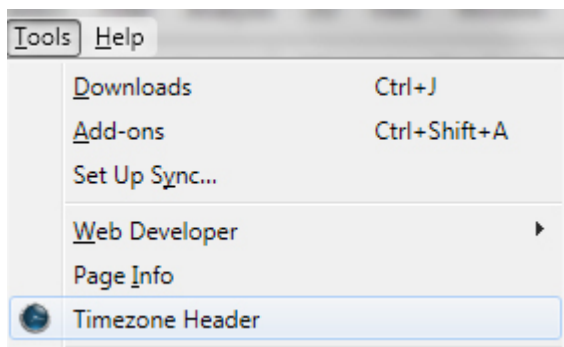
- 7) Επιλέξτε **Restart Now** για να ολοκληρώσετε την εγκατάσταση και να κάνετε επανεκκίνηση τον Mozilla Firefox.



Σχήμα 4 Επιβεβαίωση Επανεκκίνησης

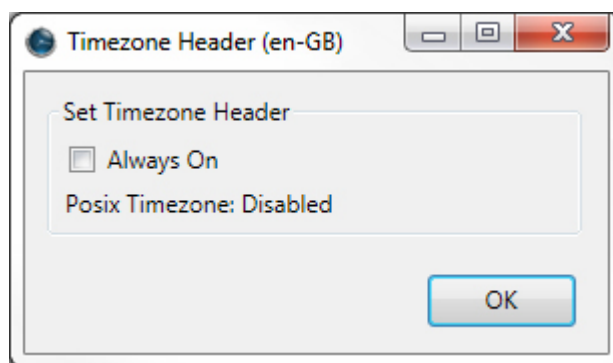
## Οδηγίες Χρήσης Του Plugin

- 1) Επιλέξτε από την μπάρα μενού, στο πάνω μέρος, την επιλογή **Tools>Timezone Header** όπως φαίνεται παρακάτω (**Σχήμα 5**)



Σχήμα 5 Άνοιγμα Επιλογών Του Plugin

- 2) Για να το Ενεργοποιήσετε/Απενεργοποιήσετε\* την λειτουργία του plugin σημειώστε ανάλογα την επιλογή **AlwaysOn** στο κουτάκι (**Σχήμα 6**). Το Timezone Header θα αποσταλεί στο επόμενο Request για μία ιστοσελίδα.

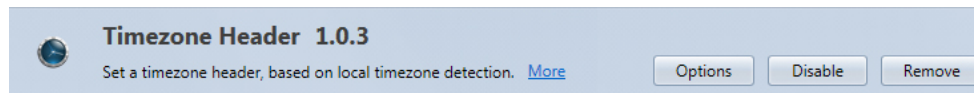


Σχήμα 6 Ενεργοποίηση/Απενεργοποίηση Του Plugin

\*Όταν Ενεργοποιείτε/Απενεργοποιείτε την επιλογή **AlwaysOn**, το πεδίο Timezone θα κάνει ανανέωση την επόμενη φορά που θα ανοίξετε τις επιλογές του plugin. Ωστόσο, η λειτουργία είναι άμεση.

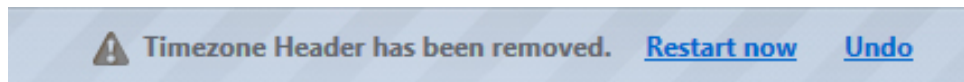
## Απεγκατάσταση

- 1) Επιλέξτε από την μπάρα μενού, στο πάνω μέρος, την επιλογή **Tools>Add-ons** όπως φαίνεται στο **Σχήμα 1**, στην παράγραφο **Εγκατάσταση**.
- 2) Στην οθόνη που θα ανοίξει επιλέξτε από το αριστερό μενού την επιλογή **Extensions (Σχήμα 2, παράγραφος Εγκατάσταση)**, όπου και θα δείτε την οθόνη με όλα τα εγκατεστημένα Add-ons.
- 3) Στην οθόνη με τα εγκατεστημένα Add-ons εντοπίστε το Timezoneheader και επιλέξτε την επιλογή **Remove (Σχήμα 7)**.



Σχήμα 7 Απεγκατάσταση Timezone Header

- 4) Επιβεβαιώστε την απεγκατάσταση επιλέγοντας **Restart now (Σχήμα 8)**.



Σχήμα 8 Επιβεβαίωση Επανεκκίνησης Μετά Την Απεγκατάσταση