ΟΠΤΙΚΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ ΜΕ ΤΗΝ ΥΠΟΣΤΗΡΙΞΗ ΤΗΣ PHP

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΟΙΚΟΝΟΜΙΚΩΝ ΚΑΙ ΚΟΙΝΩΝΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΚΑΡΑΚΟΥΓΙΟΥΜΤΖΗΣ ΓΕΩΡΓΙΟΣ

Δεδομενα-ορισμος-χρηση

Τα δεδομένα είναι ένα σύνολο αληθινών η μη στοιχείων, προσαρμοσμένα συνήθως στους κανόνες των μαθηματικών και έχουν δημιουργηθεί για να αποτελέσουν αντικείμενο επεξεργασίας, ανάλυσης και οπτικοποιησης. Δεδομένα χρησιμοποιούνται όταν μοντελοποιείται ένα πρόβλημα σε σύγχρονα επιστημονικά πεδία και παραλληλα σημαντικούς για την ανθρωπότητα κλάδους : Είτε της στατιστικής όπου έχουμε(συνηθως) αριθμητικά δεδομένα για περιγραφική στατιστική ανάλυση όπως μέτρα κεντρικής τάσης, διασποράς, κύρτωσης, συμμετρίας/ασυμμετρίας είτε της ιατρικής όπου υπάρχουν άπειρα δεδομένα ασθενών και ασθενειών, είτε κρατικά(δημοσια η μη) δεδομένα, δεδομένα πωλήσεων/αγορών/μάρκετινγκ, χρηματοοικονομικά ή οπουδήποτε χρησιμοποιούνται δεδομένα και παράλληλα η ανάλυση –άρα και η οπτικοποίηση τους.

Βασικα χαρακτηριστικα των δεδομενων

Χαρακτηριστικά συνόλου δεδομένων	Χαρακτηριστικά ενός στοιχειου του συνολου
Οργανώνονται σε :	***Ελλειψη(missingness)***:
Μεταβλητές (variables)	υπαρκτό-ορατή παρατήρηση
Πίνακες (tables)	Μη υπαρκτό-απαρατήρητο(missing)
Μήτρες (matrices)	
Συγκρινονται και μετρωνται σε:	***Κατάσταση***:
Κατηγορίες(categories)	Ενεργό (activated)
Ταξινομημένες κατηγορίες	Ανενεργό (unactivated)
Αριθμητικά μεγέθη(magnitude)	
Συχνότητες(frequencies)	

Τύποι δεδομένων

Ο τύπος των δεδομένων ενός συνόλου καθορίζεται από το τρόπο που αυτά είναι οργανωμένα αλλά και από το τρόπο σύγκρισης του κάθε υποσυνόλου που θα συμπεριληφθεί στην ανάλυση.Έτσι ενώ αναγνωρίσαμε 3 τρόπους οργάνωσης (σε μεταβλητές, σε πίνακες και σε μήτρες) και τρεις τρόπους μέτρησης τους (σε κατηγορίες, σε συχνότητες και σε αριθμητικά μεγέθη) δε μπορούμε υπεραπλουστεύοντας να υποθέσουμε ότι υπάρχουν 9 τύποι δεδομένων.Έτσι αν σε ένα πίνακα έχουμε τη μια από τις εξεταζόμενες μεταβλητές να είναι τύπου συχνοτήτων δε μπορούμε να πούμε ότι το σύνολο των δεδομένων του πίνακα είναι επίσης τύπου συχνοτήτων.

Η προσπάθεια να προσδιορίσουμε το τύπο ενός συνόλου δεδομένων περιπλέκεται ακόμη περισσότερο από το τρόπο που θα τα συγκρίνουμε και θα τα οργανώσουμε. Π.χ. δεδομένα οργανωμένα σε μεταβλητές συνήθως πάντα έχουν πάνω από μια μεταβλητές που πρέπει να αναλυθούν και αυτές οι μεταβλητές δεν μπορούν να συγκριθούν μεταξύ τους. Ακόμα αν λάβουμε υπόψη μας ότι η κατάσταση τους(ενεργά ή μη) μπορεί να αλλάξει κατά τη διάρκεια της ανάλυσης μας αυτό σημαίνει ότι και η σύγκριση θα είναι άνιση αν σε μια μεταβλητή ο ενεργός πληθυσμός αλλάξει μέγεθος.Αυτό το γεγονός κάνει το προσδιορισμό ή μάλλον τη προσπάθεια να προσδιορίσουμε το τύπο των δεδομένων ένα πολύ σημαντικό σημείο της ανάλυσης.

Τύπος δεδομένων	Τύπος	Αριθμός	Τροπος μετρησης βασει:
Συχνότητας			
Πίνακας	Ν πίνακας	Ν=1	Συχνότητας & κατηγορίας
Κατηγορία	Μεταβλητή	Ν	κατηγορίας
Μεγέθους			
Μονομεταβλητο	Μεταβλητή	ν=1	μεγέθους
διμεταβλητο	Μεταβλητή	ν=2	μεγέθους
πολυμεταβλητο	Μεταβλητή	ν>2	μεγέθους
Οργανωμένο σε υποσυνολο	Μεταβλητή	v+a	Μεγέθους και υποσυνολου
Μη υπαρκτό(missing)	Μεταβλητή	n	μεγέθους
Συσχέτισης			

Οπτικοποιωντας τα δεδομενα συχνοτητας και κατηγοριοποιημενα δεδομενα.

Αν και τα δεδομενα συνηθως ερχονται ως νουμερα και αριθμοι δεν ειναι παντα ετσι. Μπορει να ειναι προσαρμοσμενα στους κανονες των μαθηματικων, αλλά υπάρχουν και δεδομένα που ενώ δεν είναι αριθμοί μπορούν να κατηγοριοποιηθούν, να μετρηθουν και να περιληφθουν στην ανάλυση μας.Τέτοιου τύπου δεδομένα μπορούν να οργανωθούν με 2 τρόπους. Ως κατηγοριοποιημένες μεταβλητές, ή ως πίνακας συχνοτήτων. Όταν οργανώνονται ως μεταβλητές σε κατηγορίες η θεμελιώδης μονάδα παρατήρησης είναι η κατηγορία τους, ενώ όταν οργανώνονται ως πίνακας η βασική μονάδα παρατήρησης είναι το πλήθος. Χρησιμοποιείται ο όρος δεδομένα συχνότητας για αναπαραστάσεις δεδομένων υπο μορφή πινάκα ποσοστων %, ενώ η ονομασία κατηγοριοποιημένα δεδομένα χρησιμοποιείται για σύνολα φτιαγμένα αποκλειστικά απο κατηγοριοποιημένες μεταβλητές.

Εργαλεία οπτικοποιησης δεδομένων συχνότητας και κατηγοριοποιημενων δεδομενων

Δεδομενα συχνοτητας:

Πινακες: Απλοι, πολυδιαστατοι (n-way contigency tables)

Γραφηματα συχνοτητων:

Μωσαικα (Οι συχνότητες σε ένα πίνακα ν διαστάσεων αντικατοπτρίζονται σε ένα μωσαικό γραφημα ως μια συλλογή διάφορετικών χρωματισμενων ή σκιασμενων ν τετραγωνισμένων επιφανειών των οποίων οι περιοχές είναι σχετιζόμενες με τις συχνότητες των κελιών του πίνακα)

Κατηγοριοποιημενα δεδομενα:

Μωσαικα, sieve plots (αποτελουν βελτιωσεις των μωσαικων), assocation plots

Οπτικοποιωντας αριθμητικα δεδομενα (ποσοτικα)

Μονομεταβλητοι πληθυσμοι:

Dotplot, Boxplot και diamond plot(παρέχουν πληροφορία για τη διασπορά και τη κεντρική τάση της κατανομής περιλαμβάνωντας το μέσο, τα τεταρτημόρια και ούτω καθεξής), Histogram και frequency polygon , Cumulative distribution plot, Lag plot, Sequence plot

Δι-μεταβλητοι πληθυσμοι:

Scatter plot [Διαγραμμα διασπορας που χρησιμοποιειται για να δουμε τη σχεση που υπαρχει αναμεσα σε 2 μεταβλητες. Αν υπαρχει γραμμικη η μη σχεση, αν αυτη ειναι θετικη(αυτο γινεται ευκολα αντιληπτο απο το διαγραμμα αν δουμε ενα pattern στη κατευθυνση των σημειων να κινουνται απο κατω αριστερα προς τα επανω δεξια) η αρνητικη η αν δεν υπαρχει σχεση καθολου]

Οπτικοποιωντας αριθμητικα δεδομενα (...συνεχεια)

Πολυμεταβλητοι πληθυσμοι:

3 οικογενειες γραφηματων:

1)Ορθογώνιων αξόνων: Περιλαμβανει τα spin plot, orbit plot και το multivariate distribution comparison plot

2)Παράλληλων αξόνων: Περιλαμβάνονται τα parallel coordinates plots, τα parallel comparison plots και οι παράλληλες σειρές των γνωστών μας μονομεταβλητών διαγραμμάτων: parallel dot plot, parallel boxplot,parallel diamond plot.

3)Ζευγαρωτών_αξόνων(paired axes plots): Εδώ συναντάμε τα scatter plots matrices, τα quantile plot matrices, numerical plot matrices και boxplot matrices

Αλλα πολυμεταβλητα γραφικα: Data glyphs, δεντρα(ιεραρχικα η/και επαναληπτικα με κυριοτερα τα δεντρα αποφασεων).

Αρχές σχεδιασης οπτικών εργαλείων και γραφικός σχεδιασμός

Αρχες: Σωστη απεικόνιση των δεδομένων, απλούστευση(ουσια και οχι εφε), μειώση της σύγχυσης(να μη προκαλειται αλλοιωση απο συνωστισμο σημειων δεδομενων), ειλικρίνεια, ρυθμιση του λόγου των διαστάσεων του γραφηματος(aspect ratio), ρύθμιση της κλίμακας με οχι ακανονιστα διαστήματα, αναθεωρηση(αν δεν ειμαστε σιγουροι).

Οταν καποιος σχεδιαζει το τελικό γράφημα πρέπει να λάβει υποψη του: Την οπτική ιεραρχία(Το περιεχομενο πρέπει να τοποθετηθεί στο επάνω αριστερά σημείο της περιοχής, οι διαφορες στο περιεχομένο οφείλουν να διαχωρίζονται απο λευκό φόντο, πρεπει να χρησιμοποιούμε μεγάλους και εμφανείς χαρακτήρες), την οπτική ροή(Ο τρόπος με τον οποίο τα μάτια ανιχνεύουν το περιεχομένο της ενότητας με το γράφημα. Ο τυπικός τρόπος που έχει επικρατήσει στις δυτικές κοινωνίες είναι απο επάνω προς τα κάτω και απο αριστερά προς τα δεξια), την ομαδοποίηση γραφικών στοιχείων,το χρώμα (δινει οντοτητα στα δεδομενα και αλλου δινει εμφαση), την τυπογραφία(μεγεθος χαρακτηρων, χρωματισμος φοντων)

Η ανατομία ενός γραφήματος διασπορας

Τα κυριοτερα μερη ενος διαγραμματος διασπορας που ειναι και το πιο διαδεδομενο γραφημα για αναλυση των σχεσεων αναμεσα στις μεταβλητες ειναι:

Ο χώρος των δεδομένων (data rectangle), το σύμβολο απεικόνισης στοιχείου και η ταμπέλα του(plotting symbol and data label), το παραλληλόγραμμο με τις γραμμές της κλίμακας(scale-line rectangle), οι γραμμές αναφοράς(reference lines), οι κλίμακες των αξόνων και οι τιμές τους(scales and scales labels), τα σημάδια και ταμπέλες(tick marks and labels), οι ταμπέλες δεδομένων κατηγοριοποιημένων μεταβλητών (Key Data Label), ο τίτλος και η λεζάντα (Title and Caption), ο λόγος διαστάσεων του γραφήματος (aspect ratio), τα πανελ (Panels)

Σχεδιασμος της εφαρμογης

Μεθοδολογια:

Για την ανάπτυξη της εφαρμογής έγινε χρήση της τεχνικής MVC δηλαδή του μοντέλου όψεως-ελεγκτή (model-view-controller). Η τεχνική αυτή συγκεκριμένα στη κατασκευή ιστοτόπων διαχωρίζει τη λογική του user input από το input control έτσι ώστε να γίνεται ασφαλής η καταχώρηση των δεδομένων του χρήστη και η διαχείριση των εισαχθέντων δεδομένων από το διαχειριστή-προγραμματιστή να γίνεται απρόσκοπτα, ασφαλέστερα, ξεχωριστά.

Χαρακτηριστικο ειναι το παραδειγμα του dgrid οπου το input του χρηστη και η αλληλεπιδραση με το grid διαχωριζεται στην οψη-τη σελιδα παρουσιασης του grid(rest1.php) και στο store(π.χ george-myData.php) που ειναι ο controller -ο ενδιαμεσος- η σελιδα που αλληλεπιδρα με το σερβερ για την αποθηκευση των αλλαγων στα δεδομενα του grid που πραγματοποιει ο χρηστης απο τη σελιδα της παρουσιασης. Παρομοια ειναι η λογικη και για τη παρουσιαση των γραφηματων αλλα και για τις φορμες register και login.

Αρχιτεκτονικη-REST

Η εφαρμογη στο κυριως μερος της χρησιμοποιει τη REST φιλοσοφια. Το REST επεκτεινει ουσιαστικα το HTTP πρωτοκολλο αλλα δεν ειναι το ιδιο πρωτοκολλο με την αυστηρη ορολογια του ορου. Η βασική του αρχή είναι η ύπαρξη πόρων (πηγών δεδομένων ή άλλης πληροφορίας) κάθε μια απο τις οποίες αναφέρεται απο ένα μοναδικό uri. Τα δεδομένα οργανώνονται σε συλλογές απο uri (Collection of uri's) της μορφής: http://example.com/resources/ ενω η αναφορα σε ενα αλλο αντικειμενο της συλλογης γινεται απο ενα αλλο uri κατω απο το συγκεκριμενο ιεραρχικα http://example.com/resources/item17 που ονομαζεται Element URI . Για να γίνει χειραγώγηση αυτών των πηγών πληροφορίας απαιτείται ένα ενδιάμεσο interface που συνήθως είναι το http και οι user agents ή τα components δικτύου που μεταφέρουν τα δεδομένα ή πιο συγκεκριμένα τη κατάσταση τους.

Αρχιτεκτονικη-REST(...συνεχεια)

Το REST οφείλει να ακολουθεί κάποιους περιορισμούς:

Να είναι πιστό στην αρχιτεκτονική client-server, να είναι stateless, cacheable,να ακολουθεί ένα στρωματοποιημένο πρότυπο-σύστημα (layered system) όπου πιθανό να υπάρχουν ενδιάμεσοι σερβερ-σταθμοί, με σκοπό το μοίρασμα του όγκου δεδομένων, του φόρτου εργασίας, να απαιτείται να υπάρχουν δικαιώματα εγγραφής-ανάγνωσης δεδομένων και να γίνεται καταγραφή των δικαιωμάτων απο το σέρβερ.Να υπάρχει ενιαίο interface ανάμεσα σε πελάτη-διακομιστή με σκοπό να απλοποιηθεί η αρχιτεκτονική του interface επικοινωνίας έτσι ώστε να αναπτυχθούν ξεχωριστά και ανεξάρτητα το καθένα.

Οι 4 βασικές αρχές του interface είναι: Ταυτοποίηση και αναγνώριση των πόρων, χειραγώγηση των πόρων μέσα απο τις καταστασιακές αναπαραστάσεις,αυτό-επεξεγηματικά μηνύματα, τα υπερμέσα να είναι η κυρίως λειτουργια- μηχανή της εφαρμογής HATEOAS. REST λοιπον ειναι ενας τροπος μεταφορας καταστασης στην οποια βρισκονται οι πληροφοριες,η συλλογη δεδομενων μας. Για να πουμε πως μια εφαρμογη σχεδιαστηκε με REST φιλοσοφια χρειαζομαστε ενα REST client απο τη μια μερια και ενα REST server απο την αλλη. Σαν rest client χρησιμοποιηθηκαν ajax requests για τη μεταφορα των αιτηματων μας με βαση το API του dojo.

Αρχιτεκτονικη-REST(...συνεχεια)

Rest μοντελα:

Το RESTFUL web service:Λεγοντας οτι ενας ιστοχωρος παρεχει RESTFUL web services εννοουμε οτι καθε resource που διαθετει ο εξυπηρετητης αναγνωριζεται μοναδικα απο το δικο του URI.

Το hybrid REST-RPC communication system, απο την αλλη ειναι πιο ελαστικο απο την αποψη οτι δεν απαιτει καθε πορος να ειναι ξεχωριστο αρχειο η να εχει διαφορετικο Unified Resource Identification. Σε αυτου του ειδους το REST μιλαμε για μια συλλογη ολων των πορων σε ενα μοναδικο uri. Η δικη μου εφαρμογη ειναι αυτου του τυπου.

Rest μεθοδοι:

GET: Μας επιστρεφει μια λιστα της συλλογης με τους πορους στο φορματ που εχουμε επιλεξει να χρησιμοποιησουμε εμεις(json,xml)

PUT: Οταν μιλαμε για συλλογη δεδομενων σημαινει αντικατεστησε τη συλλογη με μια αλλη η αντιστοιχα αν δινουμε πληροφορια για το πορο ειτε μεσω του PATH ειτε του QUERY αντικατεστησε το πορο που ερχεται μεσω PATH_INFO η QUERY_STRING με ενα αλλο που σου στελνω ως σωμα αιτηματος.

Αρχιτεκτονικη-REST(...συνεχεια)

DELETE: Σημαινει διεγραψε μου ολη τη συλλογη η αντιστοιχα το πορο που ερχεται με το url path info η query string σαν πληροφορια.

POST: Τοποθετησε η εκχωρησε τη νεα πληροφορια στη συλλογη η εκχωρησε τη νεα πληροφορια ως συλλογη απο μονη της.

Για τη ποιο ευκολη κατανοηση των REST(HTTP) μεθοδων μπορουμε να πουμε οτι οταν γινεται GET στο SERVER μεταφραζεται ως SQL ερωτημα στη βαση δεδομενων μας για SELECT FROM table, οταν το request ειναι τυπου POST το αντιστοιχο SQL ερωτημα ειναι INSERT INTO / ALTER TABLE , για PUT ειναι UPDATE / REPLACE INTO ενω για DELETE ειναι αντιστοιχο SQL DELETE FROM TABLE.

Ο κωδικας στη σελιδα επεξεργασιας: Server-side λογικη

Αφου ο χρηστης εγγραφει και στη συνεχεια συνδεθει στο σαιτ αν επιλεξει να ανεβασει αρχειο excel για οπτικοποιηση στη σελιδα επεξεργασιας του σαιτ γινεται κατα σειρα:

Η αναγνωση των δεδομενων της 1ης γραμμης του αρχειου με την εφαρμογη PHPExcel, ο "καθαρισμος" των κελιων της απο επιβλαβη κωδικα και η αποθηκευση σε μεταβλητες συνοδου εργασιας $_SESSION των τιμων των στηλων-μεταβλητων.Η αναγνωση ολων των υπολοιπων γραμμων του αρχειου excell και ο "καθαρισμος" των κελιων απο επιβλαβη κωδικα. Ο διαχωρισμος των στηλων που περιεχουν μονο αριθμητικα δεδομενα και η αποθηκευση τους σε ξεχωριστες μεταβλητες συνοδου εργασιας $_SESSION για να οπτικοποιηθουν αυτοματα στατιστικα και γραφηματα για αυτες τις στηλες(Screenshot 1).

Η δημιουργια του πινακα της βασης δεδομενων με ολα τα δεδομενα που υπαρχουν σε κελια μεσα στο excel αρχειο με βαση τις εγγραφες-τιμες που υπαρχουν στα κελια του αρχειου αλλα και τις στηλες που ειναι αποθηκευμενες σε μεταβλητες.

Η δημιουργια των αρχειου που θα φιλοξενησει το κωδικα που θα αποτελει το store. Λεγοντας store εννοειται ενας adapter οπου αποτελει το συνδεσμο αναμεσα στο σερβερ και στο client. Ειναι ο παροχος περιεχομενου, ο ενδιαμεσος που θα παρασχει στο καταναλωτη- πελατη τα δεδομενα που αυτος θα χρησιμοποιησει-καταναλωσει για τις αναγκες του. Προκειται για το url οπου θα γινονται τα ajax requests απο το client interface ειτε του grid ειτε των chart και ο σερβερ θα απανταει στα requests με αναλογα resposes(Screenshot 2).

```php
require_once 'PHPExcel/IOFactory.php'; //$_SERVER['DOC_ROOT'].
require_once 'PHPExcel.php';
PHPExcel_Settings::setZipClass(PHPExcel_Settings::PCLZIP);
$objPHPExcel = new PHPExcel();
$objreader = PHPExcel_IOFactory::load($_SESSION['ses_filename']);
$objWorksheet = $objreader->getActiveSheet();
$highestRow = $objWorksheet->getHighestRow();
$highestColumn = $objWorksheet->getHighestColumn();
$highestColumnIndex = PHPExcel_Cell::columnIndexFromString($highestColumn);
$column = null;
$cvalues[$column] = array();
$ses_cvals[$column] = array();
$ses_json = array();
$columndatatype[$column] = array();
$dbcoldatatype[$column] = array();
$cdatype[$column] = array();
$numeric = array();
$stuff = array();
$pat = "/^[^<>`~!/@\\#}$%:;)(_^{&*=|'+]+$/";
$p = "<?";
for ($column = 0; $column < $highestColumnIndex; $column++) {
    for ($row = 1; $row <= $highestRow; $row++) {
        $cell = $objWorksheet->getCellByColumnAndRow($column, $row);
        $cvalue = $cell->getValue();
        $cva=  mysqli_real_escape_string($dbco, $cvalue);
        $cvaluee = strip_tags(htmlentities($cva));
        $cvalues[$column][] = ($cvaluee);
```

```php
$q = "CREATE TABLE IF NOT EXISTS {$_SESSION['ses_name']} (id int auto_increment not null, primary key(id), " . $str . ")ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_
$ql = mysqli_query($dbco, $q) or die("<div class='echoingg' style='text-align:center'>ERROR: It seems that you have not read the instructions!<br />The first row of you
$question = "DESC {$_SESSION['ses_name']}";
$result = mysqli_query($dbco, $question);
if (mysqli_errno($dbco) == 1146)
    echo 'The table was not created';
else
    $values = array();
foreach ($objWorksheet->getRowIterator() as $row) {
    $cellIterator = $row->getCellIterator();
    $cellIterator->setIterateOnlyExistingCells(false);
    $values = array();
    foreach ($cellIterator as $cell) {
        $val = $cell->getValue();
        $va=  mysqli_real_escape_string($dbco, $val);
        $valuee = strip_tags(htmlentities($va));
        $values[] = "\"$valuee\"";
        $valss[] = $val;
    }
    $_SESSION['values'] = $values;
    $_SESSION['imploded'] = implode(",", $values); //print_r($_SESSION['imploded']);
    $query = "INSERT INTO {$_SESSION['ses_name']} ({$_SESSION['v']}) VALUES({$_SESSION['imploded']})"; //TODO:θέλει αλλαγή στη σύνταξη του $_SESSION
    $querr = "DELETE FROM {$_SESSION['ses_name']} where id=1";
    $imp = mysqli_query($dbco, $query) or die("ERROR into inserting values to db is this:" . mysqli_error($dbco)); //TODO:check if the rows were successfully inserted
    $impp = mysqli_query($dbco, $querr) or die("ERROR in deleting row 1 is this:" . mysqli_error($dbco)); //TODO:check if the rows were successfully inserted  */
}
if ($imp && $impp) {
    touch("" . $_SERVER['DOCUMENT_ROOT'] . "/REST/" . $_SESSION['ses_name'] . "/" . $_SESSION['ses_name'] . "-myData.php");
    touch("" . $_SERVER['DOCUMENT_ROOT'] . "/REST/" . $_SESSION['ses_name'] . "/" . $_SESSION['ses_name'] . "-myDownload.php");
```

```
if( \$_SESSION['ses_name']!==\$_SESSION['ses_name'] || empty(\$_SESSION['ses_name']) && \$_SESSION['agent']!==\$_SESSION['a
            header('Status: 404 Not Found');
}
            if( \$_SERVER['REQUEST_METHOD']=='GET' && http_match_request_header('X-Requested-With','XMLHttpRequest') && htt
            \$result = mysqli_query(\$dbco, \"SELECT * FROM {\$_SESSION['ses_name']}\");

           \$rows = array();
            while (\$r = mysqli_fetch_array(\$result, MYSQLI_ASSOC)) {
                \$rows[] = \$r;
            }\$js = json_encode(\$rows);
 header('Content-type: application/json');
echo \$js;
}
if( \$_SERVER['REQUEST_METHOD']=='GET' && http_match_request_header('ask','rowCount') && http_match_request_header('Accept'
\$s=\"select id from {\$_SESSION['ses_name']} order by id desc limit 1\";
\$result = mysqli_query(\$dbco,\$s);
if(\$result){
\$an=array();
while(\$an1=@mysqli_fetch_array(\$result, MYSQLI_ASSOC)){
\$an[]=\$an1;}
foreach(\$an as \$a){
\$r=implode(\$a);}
\$rr=(string)\$r;
 header('Content-type: text/html');
 header('Status: 200 OK');
echo \$rr;
}else { /*header('Status: 500 INTERNAL SERVER ERROR'); */die(\"ERROR IN SELECT ROWCOUNT:\" .mysqli_error(\$dbco));}
}
             elseif( \$_SERVER['REQUEST_METHOD']=='DELETE' && !empty(\$_SERVER['QUERY_STRING'])){
            \$trimmed=is_string(\$_SERVER['QUERY_STRING'])?\$_SERVER['QUERY_STRING']:' ';
            \$str=substr(\$trimmed,3);
            \$tr=(integer)\$str;
  \$sqld=\"DELETE FROM {\$_SESSION['ses_name']} WHERE id=\$tr\";
                \$result = mysqli_query(\$dbco, \$sqld);
```

```
elseif(\$_SERVER['REQUEST_METHOD']=='POST' && http_match_request_header(\"additional\", \"columns\")){
\$b=http_get_request_body();
if(isset(\$b)  && is_string(\$b)){
\$escc=mysqli_real_escape_string(\$dbco,\$b);
\$str=strip_tags_deep(htmlentities_deep(\$escc));
\$_SESSION['v'].=\",\".\$str;
\$se=(string)\$_SESSION['v'];
//\$_SESSION['hvalues']=array();

\$_SESSION['hvalues']=explode(\",\",\$se);// its an array of strings
}
else{
print_r(\$_POST['data']);
echo'<br />';PHP_EOL;
echo\$str;
echo'<br />';PHP_EOL;
echo \$_SESSION['v'] ;
echo'<br />';PHP_EOL;
print_r(\$_SESSION['hvalues']);
PHP_EOL;
}
if(isset(\$str) && is_string(\$str)){
\$q=\"ALTER TABLE  {\$_SESSION['ses_name']} ADD {\$str} varchar(250) \";
\$msq=mysqli_query(\$dbco,\$q);
if(\$msq){
\$res=mysqli_query(\$dbco,\"show columns from {\$_SESSION['ses_name']}\") or die('Error:'.  mysqli_error(\$dbco));
\$a2=  mysqli_num_rows(\$res) or die('Error while returning nums'. mysqli_error(\$dbco));
if(\$a2==count(\$_SESSION['hvalues'])+1){//all ok
header('Status: 204 Created');
header('Content-Type: application/javascript;application/json');
//header('Content-Type: text/html');
\$a=array('id');\$ba=array_merge(\$a,\$_SESSION['hvalues']);\$c=array(1,2);
//\$stri=mkStructureee(\$c);echo \$stri;
 echo 'require([\"dgrid/editor\", \"dojo/ready\"],function(editor,ready){evt=function(){var e='; mkStructureee(\$ba); echo 'grid.set(\"columns\",e);}
 ready(evt);
 });';
//echo 'id.' \$_SESSION['v'] ;
```

Ο κωδικας:Client-side λογικη

Αφου τελειωσουμε με το server κομματι της εφαρμογης, σειρα παιρνει ο κωδικας javascript. Ενσωματωνουμε το dojo στην εφαρμογη με script tag και κατοπιν ολα τα απαραιτητα αρχεια με require που θα απαιτηθουν για να γινουν parse τα αντικειμενα που αρχικοποιουνται μεσω html και θα χρησιμοποιηθουν για το layout μας(TabContainer,ContentPane,buttons κτλ).

Στη συνεχεια γραφουμε το κωδικα που θα αρχικοποιει το grid αντικειμενο και ολα τα αλλα απαραιτητα αντικειμενα για τη λειτουργια του(Store, grid plugins), (Screenshot 1)

Κατω απο το grid υπαρχουν δυο αλλα div's που βρισκονται 1ον: τα κουμπια που θα εκτελουν τα requests προς το store για αλλαγη των δεδομενων του grid(delete row, add row, add column, download file) και 2ον μια φορμα που ο χρηστης θα επιλεγει στηλες απο το grid και μεσω ασυγχρονων αιτηματων θα δημιουργουνται σε ContentPane γραφηματα των δεδομενων(Screenshots 2,3,4).

Ο κωδικας:Client-side λογικη(...συνεχεια)

Οταν ο χρηστης επιλεξει στηλη η στηλες εμεις με απλη javascript παιρνουμε τα περιεχομενα των επιλεγμενων select options και checkboxes και κατοπιν δημιουργουμε με το request API του dojo 1ον: ενα string που θα ζηταει απο το store μας να φερει τα δεδομενα των στηλων σε text μορφη χωρισμενα απο καθετους σε ενα πινακα και 2ον:τα headers, standard και custom που θα αποσταλουν για να καταλαβει το store τι θελουμε να κανουμε [Εδω server-side πραγματοποιειται ο ελεγχος με το http_match_request_header('make','charts')]. Οταν τα δεδομενα του POST request καταφθασουν εμεις με τη shift χωριζουμε τον αριθμο των στηλων και το τυπο του γραφηματος απο τα υπολοιπα δεδομενα που μας εστειλε η βαση και με τη split χωριζουμε τα αποτελεσματα των στηλων σε ενα πινακα. Στη συνεχεια φτιαχνουμε 2 υποπινακες αν προκειται για 2 στηλες για να αποτελεσουν τα δεδομενα των αξονων του γραφηματος και αρχικοποιουμε προγραμματιστικα 1ον: 2 νεα div που θα φιλοξενησουν το ContentPane και το γραφημα με τυχαια id, 2ον το ContentPane και 3ον το γραφημα που θα οπτικοποιει τα δεδομενα.

```
<div id="grid" class="dgrid tundra" style="width:960px; height:750px" >
    <script type="text/javascript" data-dojo-event="onShow" data-dojo-args="evt">
  require(["dojo/_base/declare", "dojo/ready", "dojo/domReady","dojo/dom-attr","dijit/registry","dojo/on","dgrid,
            "dojo/store/Observable","dojo/store/JsonRest","dojoSmore/RequestMemory", "dojo/store/Cache", "dijit/
        function(declare,ready,domReady,domattr,registry,on,Grid,OnDemandGrid,_StoreMixin,editor,Selection,CellSe
             evt=function(){
var mystructure=<?php mkStructuree($_SESSION['hvalues']); ?>
memoryStore = new Memory(<?php echo"{url:'/REST/" . $_SESSION['ses_name'] . "/" . $_SESSION['ses_name'] . "-myDa
restStore = new JsonRest(<?php echo"{target:'/REST/" . $_SESSION['ses_name'] . "/" . $_SESSION['ses_name'] . "-m
storeMaster= new Observable(restStore);
storeMemory=new Observable(memoryStore);
store = new Cache(storeMaster,storeMemory);
CustomGrid = new declare([OnDemandGrid,  editor, Selection, CellSelection, selector, Keyboard,  DnD, ColumnResize
grid = new CustomGrid({
    columns: mystructure,
    store:store,
    selectionMode: "extended", // for Selection; only select a single row at a time
    cellNavigation: false,// for Keyboard; allow only row-level keyboard navigation
    getBeforePut: false
   }, "grid");
grid.on("dgrid-datachange", function(event) {
    var roww=grid.row(event);
    var cell=grid.cell(event);
    var options={id:roww.id};
    var object=grid.row(roww.id);
    } );
delTool=function(){
    new Tooltip({connectId: ['del'],label: 'Delete rows by selecting them from the id field'});
}
downTool=function(){new Tooltip({connectId: ['down'],label: 'Download your new excel file'});}

addColumn=function(){
```

```
<script type="text/javascript" data-dojo-event="onShow" data-dojo-args="chart1,chart,chart2">
    req=require(["dojo/ready","dojo/dom","dojo/on","dojo/query","dojo/domReady","dojo/dom-form","dijit/registry"
        "dojox/charting/Chart","dojox/charting/plot2d/Scatter","dojox/charting/plot2d/Grid","dojox/charting/axis
        "dojox/charting/themes/Minty", "dojox/charting/action2d/MouseZoomAndPan","dojox/charting/action2d/TouchZ
        "dojox/charting/action2d/Highlight","dojox/charting/action2d/MouseIndicator","dojo/fx/easing","dojox/gfx
    function(ready,dom,on,query,domReady,domForm,registry,contentPane,json,Form,Button,CheckBox,request,lang,Cha
        evt4=function(){
            values=[]
            var elems = document.forms[1].elements;
            //if(){}
            // var array='';
            var counter = 0;
            var e = document.getElementById("ddlViewBy");
            strUser = e.options[e.selectedIndex].value;
            var  array=""+strUser+","+"";
            for(var i = 0; i < elems.length; i++){
                if(elems[i].checked==true){
                    values.push(elems[i].value);
                    array+=""+elems[i].value+",";
                    counter++;
                }
            }
   if(counter==2 && strUser=="Scatter"  || counter==1 && strUser=="Pie" || counter==2 && strUser=="Spider"){
                var args={
                    data:counter+','+array,
                    preventCache:false,
                    headers :{
                        'Content-Type' :'application/x-www-form-urlencoded',//'multipart/form-data',//
                        'Accept' :'text/html',
                        'X-Requested-With':'XMLHttpRequest',
                        'Make': 'chart'
                    },
                    handleAs:'text',
```

```
var def= request.post(<?php echo"'https://localhost/REST/" . $_SESSION['ses_name'] . "/" . $_SESSION['ses_name
def.then(function(response){
    rest=new Array();
    var st='';
   st+=response;
  rest=st.split("|");
    var Count=rest.shift();
    Icount=Number(Count);
    var shi=rest.shift();
    shif=String(shi);
       if(shif=="Scatter" && Icount==2){
       r1=new Array();  r2=new Array();
        var inn;var inno;
        for(var i=0; i<rest.length; i++){
            if(i%2 == 0){
                inn=Number(rest[i]);
                //inno=isNumber(inn);
                if(!isNaN(inn)){
                    r1.push(inn);}
            }  else{
                inn=Number(rest[i]);
                //inno=isNumber(inn);
                if(!isNaN(inn)){
                    r2.push(inn);}}
        }
        var obj= r1 instanceof Array && r1.length>0;
        var obj2= r2 instanceof Array && r2.length>0;
        for(var i=0; i<r2.length; i++){
        var strIt2=""+Number(r2[i])+" ";
```

Empty string passed to getElementById().

POST https://localhost/REST/george/george-myData.php 200 OK 19ms

Headers Post Response HTML Cookies


```
2|Scatter|sadsd|asdfgasd|456|5458|789|5478|2155|258|234|125|12|563|3214|5214|547|526|512|528|632|5214
|325|546|5214|789|320|456|2145|123|3268|1254|5247|5844|1|5652|369|1354|258|4587|2145|5214|2582|587|3165
|554|2356|774|8954|5589|2147|654|2145|2358|6985|5896|547|547|254|587|258|458|123|12331|sadsd|asdfgasd
|456|5458|789|5478|2155|258|234|125|12|563|3214|5214|547|526|512|528|632|5214|325|546|5214|789|320|456
|2145|123|3268|1254|5247|5844|1|5652|369|1354|258|4587|2145|5214|2582|587|3165|554|2356|774|8954|5589
|2147|654|2145|2358|6985|5896|547|547|254|587|258|458|123|12331|sadsd|asdfgasd|456|5458|789|5478|2155
|258|234|125|12|563|3214|5214|547|526|512|528|632|5214|325|546|5214|789|320|456|2145|123|3268|1254|5247
|5844|1|5652|369|1354|258|4587|2145|5214|2582|587|3165|554|2356|774|8954|5589|2147|654|2145|2358|6985
|5896|547|547|254|587|258|458|123|12331|sadsd|asdfgasd|456|5458|789|5478|2155|258|234|125|12|563|3214
|5214|547|526|512|528|632|5214|325|546|5214|789|320|456|2145|123|3268|1254|5247|5844|1|5652|369|1354
|258|4587|2145|5214|2582|587|3165|554|2356|774|8954|5589|2147|654|2145|2358|6985|5896|547|547|254|587
|258|458|123|12331|sadsd|asdfgasd|456|5458|789|5478|2155|258|234|125|12|563|3214|5214|547|526|512|528
|632|5214|325|546|5214|789|320|456|2145|123|3268|1254|5247|5844|1|5652|369|1354|258|4587|2145|5214|2582
|587|3165|554|2356|774|8954|5589|2147|654|2145|2358|6985|5896|547|547|254|587|258|458|123|12331|sadsd
|asdfgasd|456|5458|789|5478|2155|258|234|125|12|563|3214|5214|547|526|512|528|632|5214|325|546|5214|789
|320|456|2145|123|3268|1254|5247|5844|1|5652|369|1354|258|4587|2145|5214|2582|587|3165|554|2356|774|8954
|5589|2147|654|2145|2358|6985|5896|547|547|254|587|258|458|123|12331|sadsd|asdfgasd|456|5458|789|5478
|2155|258|234|125|12|563|3214|5214|547|526|512|528|632|5214|325|546|5214|789|320|456|2145|123|3268|1254
|5247|5844|1|5652|369|1354|258|4587|2145|5214|2582|587|3165|554|2356|774|8954|5589|2147|654|2145|2358
|6985|5896|547|547|254|587|258|458|123|12331
```


```
     var randd=makeid();var randod=makeid();
  var m= document.createElement('div');
 var attr = m.setAttribute('id',randd);
 var attrb = m.setAttribute('style','width:800px;height:800px');
 cont=new contentPane({region:'right',
 style:"height:100%; width:100%;float:center",id:randod,title:"<b>Spider</b>",executeScripts:"true", preloa
  });cont.set("content",m);
 var co6=registry.byId("_container_")||TabContainer;
 co6.addChild(cont);cont.startup();
  registry.add(cont);
  chart2 = new Chart(randd);
 chart2.addPlot("default", {type:"Spider",abelOffset: -10,axisColor: "lightgray", divisions: Icount, spider
     seriesFillAlpha: 0.2, markerSize: 3, precision: 0, spiderOrigin: 0.16, spiderType: "polygon"});
 chart2.setTheme(Minty);
   var  color=["brown","blue","grey","purple","pink","black","red","green","yellow","orange"]
 chart2.addSeries(values[0], r1, { fill: color[7] });
 chart2.addSeries(values[1], r2, { fill: color[0] });
 chart2.render();
  var tex=document.createTextNode(ready(chart2));
 var x=m.appendChild(tex);
 var ss1=registry.byId(randod);
  registry.byId("_container_").selectChild(ss1);  }
```

Τελος

Ευχαριστω για τη προσοχη σας