

ESPocketBiz – Scripting

Entersoft Business Suite® | Entersoft CRM® | Entersoft Mobile Suite®

Τεχνικό εγχειρίδιο

Ταυτότητα

Έκδοση κειμένου	2.0
Τελ. ενημέρωση	Ιούνιος 2021

Copyright

© Copyright 2011 Entersoft A.E. Με την επιφύλαξη όλων των νόμιμων δικαιωμάτων.

Δεν επιτρέπεται η αναδημοσίευση ή μετάδοση με οποιαδήποτε μορφή και για οποιοδήποτε λόγο μέρους ή όλου του παρόντος εγχειριδίου, χωρίς τη ρητή άδεια της Entersoft A.E.

Όσον αφορά το περιεχόμενο του παρόντος...

- Μπορεί να μεταβληθεί οποιαδήποτε στιγμή.
- Εξυπηρετεί αποκλειστικά πληροφοριακούς σκοπούς.
- Δεν παρέχεται εγγύηση για την ύπαρξη τυχόν λαθών ή για τη λανθασμένη χρήση ή για μη επιθυμητά αποτελέσματα από τη χρήση των προτεινόμενων στο παρόν διαδικασιών.

Περιεχόμενα

1.	Παραμετροποίηση	6
1.1	Σχετικοί Πίνακες	6
1.2	Αντικείμενο JavaScript.....	6
2.	Γενικές Οδηγίες Χρήσης	7
3.	Core	8
3.1	Γενικές Συναρτήσεις.....	8
3.2	Γενικές Συναρτήσεις για Report	11
3.3	Events-Actions	12
3.3.1	Λίστα πελατών	12
3.3.2	Στοιχεία πελάτη	13
3.3.3	Λίστα ενεργειών	14
3.3.4	Νέος πελάτης	15
3.3.5	Report	16
3.3.6	Merchandising	17
4.	Sales Order (old).....	18
4.1	Γενικές Συναρτήσεις.....	18
4.1.1	Λίστες	18
4.1.2	Grid	18
4.1.3	Products list in details	19
4.2	Events	19
5.	Xvan.....	23
5.1	Γενικές Συναρτήσεις.....	23
5.1.1	Λίστες	23
5.1.2	Grid	23
5.1.3	Products list in details	23
5.2	Events	24
6.	Sales Order Plus (New)	25
6.1	Σημεία Scripting	25
6.2	Πρόσβαση Στα Αντικείμενα Sales Order Plus.....	25
6.3	Γενικές Συναρτήσεις.....	26
6.4	Πεδία Αντικειμένων	28
6.4.1	Headers Standard Fields	28
6.4.2	Product Standard Fields	29
6.4.3	Discount Fields	30
6.5	Extra Fields	32
6.6	Events	33
7.	Collection	35
7.1	Γενικές συναρτήσεις.....	35

7.2	Events	36
8.	Field Service	37
8.1	Γενικές Συνρτήσεις.....	37
8.2	Events	37
9.	Surveys	38
9.1	Γενικές Συνρτήσεις.....	38
9.2	Events	39
10.	Debugger	40

Εισαγωγή

Το scripting δίνει την δυνατότητα σε κάποιον να προσθέσει το δικό του κομμάτι κώδικα προκειμένου να καλύψει μερικές ειδικές περιπτώσεις για τον κάθε πελάτη που με την υπάρχουσα παραμετροποίηση δεν καλύπτονται. Η υποστηριζόμενη γλώσσα είναι η Javascript. Το συγκεκριμένο έγγραφο θα βρίσκεται διαρκώς υπό ανάπτυξη ανάλογα με τις απαιτήσεις αλλά και τις δικές σας προσθήκες.

1. Παραμετροποίηση

1.1 Σχετικοί Πίνακες

tblCodeUnits - Σε αυτόν τον πίνακα αποθηκεύονται οι εγγραφές για κάθε javascript κώδικα.

tblCodeUnits	
Column Name	Descriptions
CUName	Το περιγραφικό όνομα του κώδικα JavaScript. Για το android είναι συγκεκριμένες οι περιγραφές (βλ. παρακάτω)
CUType	Τύπος κώδικα. Στην περίπτωση του android υποστηρίζεται το Module
CUCode	Ο κώδικας για το javascript
CULanguage	Για το android καταχωρούμε JavaScript
CUIInactive	Για να είναι ενεργός ο κώδικας βάζουμε 0. Αν θέλουμε να απενεργοποιήσουμε τον κώδικα καταχωρούμε 1

1.2 Αντικείμενο JavaScript

SF: αποτελεί ένα javascript αντικείμενο που μπορείτε να χρησιμοποιείτε εσωτερικά στον κώδικα σας για να έχετε πρόσβαση στις γενικές συναρτήσεις.

Οι παρακάτω συναρτήσεις θα πρέπει να μπαίνουν στο ανάλογο script του επιλεγμένου apk. Αυτό καθορίζεται με το όνομα script(CUName) .

2. Γενικές Οδηγίες Χρήσης

Πρέπει να προσέξουμε όταν κάνουμε χρήση των **events**, **action** και **γενικών συναρτήσεων** τον τρόπο συγγραφής τους γιατί είναι **case sensitive**.

3. Core

Όνομα script **CUName = andCustomerList** αφορά την λίστα πελατών.

3.1 Γενικές Συναρτήσεις

- **showAlert(message, buttonText, buttonFunction):** είναι μία μέθοδος που σας επιτρέπει να προβάλετε διαλόγους με μηνύματα και να παίρνετε feedback από τον χρήστη. Το buttonText είναι το αναγραφόμενο κείμενο στο κουμπί και το buttonFunction είναι το όνομα της javascript μεθόδου που θα εκτελεστεί εάν ο χρήστης πατήσει το κουμπί.
- Επίσης μπορείτε να αφήσετε το buttonText και το okFunction κενό ή null εάν δεν επιθυμείται κάποιο συγκεκριμένο όνομα στο alert
- **showAlert(message, okButtonText, okFunction, cancelButtonText, cancelFunction):** λειτουργεί αντίστοιχα με την παραπάνω μέθοδο. Εμφανίζει ένα alert με 2 κουμπία για yes/no απαντήσεις. Τα 2 επιπλέον ορίσματα καθορίζουν το όνομα του 2^{ου} κουμπιού όπως επίσης και την μέθοδο που θα εκτελεστεί αφού πατήσει το συγκεκριμένο κουμπί.



Παράδειγμα

```
SF.showAlert('This is an alert!', 'OK', 'okFunction');
SF.showAlert('This is another alert', 'Yes', null);
SF.showAlert('This is another alert', '', 'okFunction');
SF.showAlert('This is another alert', '', null);
```

- **query(SQL Query):** η συγκεκριμένη μέθοδος παίρνει σαν παράμετρο ένα scalar sql, το εκτελεί στην τοπική βάση και επιστρέφει την συγκεκριμένη τιμή σαν λεκτικό.



Παράδειγμα

```
SF.query('select tdogrossvalue from tblinvmaster where slmid=$SLMID$ and cusid=$CUSID$ order by dockatahorisidate desc Limit 1');
```

- **command(SQL Query):** η συγκεκριμένη μέθοδος παίρνει σαν παράμετρο ένα command(insert, delete, update) και το εκτελεί στη βάση.



Παράδειγμα

```
SF.command("update configuration set configcode='newcode' where configid=10");
```

- **getSettings(setting):** η συγκεκριμένη μέθοδος παίρνει σαν παράμετρο ένα λεκτικό το οποίο επιστρέφει διάφορες ρυθμίσεις της συσκευής. Υποστηρίζονται τα: **slmid** , **url**



Παράδειγμα

```
var url = SF.getSettings('url');
```

- **getListViewRows(listname):** όταν η συγκεκριμένη οθόνη έχει κάποια λίστα τότε μπορείτε να πάρετε το σύνολο των γραμμών καλώντας την συγκεκριμένη μέθοδο με παράμετρο το όνομα της αντίστοιχης λίστας.



Παράδειγμα

```
function getList() {
    var rows = SF.getListViewRows('header');
    if( rows > 10) {
        SF.showAlert('You have more than 10 items in your list', '', null);
    } else {
        SF.showAlert('You have less than 10 items in your list', '', null);
    }
}
```

- **getListViewDetailByIndex (listname, index, value):** όταν η συγκεκριμένη οθόνη έχει κάποια λίστα τότε μπορείτε να πάρετε μία συγκεκριμένη τιμή που αφορά την εκάστοτε γραμμή της λίστας με βάση τη θέση της. Η πληροφορίες που μπορείτε να πάρετε ποικίλει από λίστα σε λίστα και από οθόνη σε οθόνη. Για περισσότερες λεπτομέρειες μπορείτε να ανατρέξετε στην εκάστοτε παραμετροποίηση της κάθε οθόνης.



Παράδειγμα

```
var customerName = SF.getListViewDetailsByIndex('cuslist', 4, 'cusname');
```

- **getListViewDetailByKey (listname, key, value):** όταν η συγκεκριμένη οθόνη έχει κάποια λίστα τότε μπορείτε να πάρετε μία συγκεκριμένη τιμή που αφορά την εκάστοτε γραμμή της λίστας με βάση μία συγκεκριμένη λεπτομέρεια που χαρακτηρίζει μοναδικά την γραμμή. Η πληροφορίες που μπορείτε να πάρετε ποικίλει από λίστα σε λίστα και από οθόνη σε οθόνη. Για περισσότερες λεπτομέρειες μπορείτε να ανατρέξετε στην εκάστοτε παραμετροποίηση της κάθε οθόνης.



Παράδειγμα

```
var sdtid = SF.getListViewDetailsByKey('headerlist', 'sdt', 'sdtid');
```

- **log(levelDebug, message):** βγάζει log μήνυμα στην κονσόλα του android studio. Αφορά μόνο dev levelDebug παίρνει τιμές i, e, d και message το μήνυμα που θέλουμε να εμφανίσουμε



Παράδειγμα

```
SF.log('i','Hello Debug');
```

- **getGridCurrentValue (key):** Αφορά το grid και επιστρέφει την τιμή από το αντίστοιχο κελί



Παράδειγμα

```
SF.getGridCellValue('udfint3');
```

- **setGridCurrentValue (key, value):** Αφορά το grid και βάζει τιμή σε ένα συγκεκριμένο κελί.



Παράδειγμα

```
SF.setGridCellValue('udfint1, '<font color="red"> Hello </font>');
```

- **getActivityValueByKey (key):** Η κάθε φόρμα έχει μερικές συγκεκριμένες μοναδικές τιμές για κάθε οθόνη. Για να μπορείτε να έχετε πρόσβαση καλείτε αυτή τη μέθοδο με παραμέτρο το κάτλληλο key.



Παράδειγμα

```
var aValue = SF.getActivityValueByKey('theKey');
```

3.2 Γενικές Συναρτήσεις για Report

- **getSelectedRows()**: Υποστηρίζεται από την έκδοση του Core 8.14.0.0 > και μετά. Αυτή η μέθοδος μπορεί να χρησιμοποιηθεί μόνο σε script που αφορά report και παίρνει τις τιμές από όλες τις επιλεγμένες γραμμές του report.

Αναλυτικά η **SF.getSelectedRows()**; επιστρέφει μια λίστα, όπου κάθε γραμμή της λίστας περιέχει μια λίστα με όλες τις τιμές που επιστρέφει το query του report (μετά από την εφαρμογή των tblReportCriteria), με την σειρά που εμφανίζονται στο grid του report. Σημαντικό να τονίσουμε ότι για να χρησιμοποιηθεί κάποια τιμή από τις επιλεγμένες γραμμές πρέπει οπωσδήποτε να γίνει cast σε κείμενο ή σε αριθμό μέσα από τις μεθόδους String(x) ή Number(x).



Παράδειγμα

Έστω `var lines = SF.getSelectedRows()`; Για να πάρω την τιμή από την 5^η κολώνα της 1^{ης} επιλεγμένης γραμμής ως αριθμό θα πρέπει να χρησιμοποιήσω το `var value = Number(lines[0][4])`.

Προσθήκη script για report:

- Προσθήκη στο tblReportCriteria γραμμή με
 - RCFeldType = SCRIPT
 - ReportCode το tblReports.ReportCode το οποίο αφορά
 - RCCrystalParameterName την τιμή που θα περνάει ως crystalParameter στο script.
- Στον **tblCodeUnits** προσθέτουμε γραμμή όπου **CUName** = "andReport" και **CUType** περιέχει: ***function UseSelectedReportLines(reportCode, crystalParameter) { }***

function UseSelectedReportLines:

- Η παράμετρος reportCode παίρνει σαν τιμή αυτό που έχει οριστεί στο tblReports.reportCode για το συγκεκριμένο report
- Η παράμετρος crystalParameter παίρνει σαν τιμή αυτό που έχει οριστεί στο tblReportCriteria.RCCrystalParameterName



Σημείωση

Στα reports συχνά γίνεται χρήση html ώστε τα αντικείμενα μια στήλης να εμφανίζονται bold, με άλλα χρώματα κλπ.

Επειδή τα reports δεν κάνουν καμία αλλαγή στα δεδομένα και απλά δείχνουν τις τιμές που φέρνει το query του report, οι τιμές που επιστρέφει η SF.getSelectedRows() είναι αυτές που επιστρέφει το query αυτούσιες.

Αυτό σημαίνει ότι αν το query επιστρέφει κάτι με html μέσα (πχ select '' || tblinvmaster.invmasterid || '' as Value) τότε όταν στο script πάρει την τιμή για εκείνη την στήλη θα πάρει κάτι του στυλ "25456" και μετά είναι ευθύνη του consultant να βρει τρόπο να αφαιρέσει τα html tags ώστε να πάρει την πραγματική πληροφορία που περιέχεται στην στήλη.

3.3 Events-Actions

3.3.1 Λίστα πελατών

Όνομα script **CUName = andCustomerList** αφορά την λίστα πελατών.

Υποστηρίζονται:

- Tokens: \$SLMID\$, \$CUSID\$
- Το αντικείμενο SF για την κλήση γενικών συναρτήσεων του Core για να έχουμε αλληλεπίδραση με την βάση και την προβολή alerts.

Event: ActivityOnCustomEvent (action) Περιμένει true/false για το αν συνεχίσει. Μπορούμε να ενσωματώσουμε κώδικα στην οθόνη της λίστας πελατών για να προβάλλουμε alerts και να εκτελέσουμε actions με βάση προκαθορισμένα κριτήρια.

Η παράμετρο action περιέχει μια από τις εξής τιμές:

- **BeforeCustomerSelection:** πριν επιλεγεί ο πελάτης. Περιμένει σαν απάντηση ένα **true/false** προκειμένου να συνεχίσει το πρόγραμμα ή όχι. Το token \$CUSID\$ αντικαθίσταται με το id του συγκεκριμένου πελάτη και το \$SLMID\$ με το ID του πωλητή-χρήστη της εφαρμογής.



Παράδειγμα

A) Έλεγχος αν έχει καταχωρηθεί το email του πελάτη με alert:

```
function ActivityOnCustomEvent(action) {  
    if (action == 'BeforeCustomerSelection') {  
        var email=SF.query("select adremail from adr inner join cus on  
cus.adriddest=adr.adrid where cus.cusid=$CUSID$");  
  
        if (email == null) {  
            SF.showAlert('Το e-mail του πελάτη είναι κενό, παρακαλώ συμπληρώστε  
το!', 'OK', null);  
        }  
    }  
    return 'true';  
}
```

Συνέχεια παραδείγματος

B) Προβολή προωθητικών ενεργειών που αφορούν τον πελάτη από τον tblMessages και προβολή των σχολίων που έχουν καταχωρηθεί για τον πελάτη σε προηγούμενη επίσκεψη

```
function ActivityOnCustomEvent(action) {  
    if (action == 'BeforeCustomerSelection') {  
        var message = '';  
        var remarks = '';  
        var marketingInst = '';  
        var marketingInst =  
            SF.query("select ifnull(msgtext, '')  
                from tblmessages where cusid=$CUSID$ and date('now') >=  
                date(fromdate) and date('now')<=date(todate)");  
        var remarks =  
            SF.query("select ifnull(remarks, '') from cus where cusid=$CUSID$");  
  
        if (marketingInst != null) {  
            message = marketingInst + "\n\n";  
        }  
        if (remarks != '') {  
            message += remarks;  
        }  
        if (message != '') {  
            SF.showAlert(message.toString(), 'OK', null);  
        }  
    }  
    return 'true';  
}
```

3.3.2 Στοιχεία πελάτη

Όνομα script **CUName = andCustomerDetails** αφορά την οθόνη στοιχείων του πελάτη.

Υποστηρίζονται:

- Tokens: \$SLMID\$, \$CUSID\$
- Το αντικείμενο SF για την κλήση γενικών συναρτήσεων του Core για να έχουμε αλληλεπίδραση με την βάση και την προβολή alerts.

Διαθέσιμα events:

- **StartVisitOnClick():** πριν πατήσει ο χρήστης την ενέργεια για έναρξη επίσκεψης. Περιμένει true/false για να συνεχίσει.
- **SimpleVisitOnClick():** πριν πατήσει ο χρήστης την ενέργεια για έναρξη απλής επίσκεψης. Περιμένει true/false για να συνεχίσει.



Παράδειγμα

Για να εμποδίσουμε την επίσκεψη σε νέο πελάτη ή μια συγκεκριμένη κατηγορία πελατών χρησιμοποιούμε το παρακάτω script

```
function StartVisitOnClick() {
    if ($CUSID$ == -1) {
        SF.showAlert('Δεν μπορείτε να ξεκινήσετε κανονική επίσκεψη σε νέο πελάτη!',
        '', null);
        return 'false';
    } else {
        return 'true';
    }
}
```

3.3.3 Λίστα ενεργειών

Όνομα script **CUName = andCustomerActions** αφορά την οθόνη στοιχείων του πελάτη.

Υποστηρίζονται:

- Tokens: \$SLMID\$, \$CUSID\$, \$VISITID\$
- Το αντικείμενο SF για την κλήση γενικών συναρτήσεων του [Core](#) για να έχουμε αλληλεπίδραση με την βάση και την προβολή alerts.

Event: ActivityOnCustomEvent (action, extraValue) Περιμένει true/false για το αν συνεχίσει. Μπορούμε να ενσωματώσουμε κώδικα στην οθόνη της λίστας πελατών για να προβάλλουμε alerts και να εκτελέσουμε actions με βάση προκαθορισμένα κριτήρια.

Διαθέσιμα action:

- **StartAction:** Πριν επιλεγεί η αντίστοιχη ενέργεια για τον πελάτη. Επιπλέον παράμετρος το extravalue = *slmtaskid* της συγκεκριμένης ενέργειας.
- **EndVisit:** η ενέργεια για το κλείσιμο της επίσκεψης. Το extravalue είναι το κενό και δεν έχει καμία χρησιμότητα. Περιμένει true/false για το αν θα συνεχίσει την ενέργεια.
- **CancelVisit:** η ενέργεια για την ακύρωση της επίσκεψης. Το extravalue είναι το κενό και δεν έχει καμία χρησιμότητα. Περιμένει true/false για το αν θα συνεχίσει την ενέργεια.

Παράδειγμα

```
function ActivityOnCustomEvent(action, extravalue) {
    if (action == 'CancelVisit') {
        var count =
            SF.Query('Select      count(*)      from      tblmerchpromotiondata      where
visitid=$VISITID$');

        if (count > 0) {
            SF.showAlert('Cancel is not allowed because there are '
+count + ' seconddisplays','',null);

            return 'false';
        }
    }
    return 'true';
}
```

3.3.4 Νέος πελάτης

Όνομα script **CUName = newCustomerValidateField** αφορά την οθόνη του νέου πελάτη.

Event: FieldValidator(datamember, displayValue) χρησιμοποιείται για τον έλεγχο ακεραιότητας καταγραφής των πεδίων του νέου πελάτη.

Παράδειγμα

```
function FieldValidator(datamember, displayvalue) {
    if (datamember == 'adrphone1' || datamember == 'adrphone2') {
        var phone = displayvalue;
        phone = phone.toString();
        phone = phone.trim();
        phone = phone.length;

        if (phone != 10) {
            sMessage = "Ο τηλεφωνικός αριθμός που καταχωρίσατε δεν είναι
έγκυρος.\n";

            SF.showAlert(sMessage, '', null);
            return 'false';
        }
    }
}
```

3.3.5 Report

Προσθήκη script για report:

- Προσθήκη στο tblReportCriteria γραμμή με
 - RCFieldType = SCRIPT
 - ReportCode το tblReports.ReportCode το οποίο αφορά
 - RCCrystalParameterName την τιμή που θα περνάει ως crystalParameter στο script.

Όνομα script **CUName = andReport** αφορά την οθόνη με τα report.

Όνομα **CUType = function UseSelectedReportLines(reportCode, crystalParameter) { }**

function UseSelectedReportLines:

- Η παράμετρος reportCode παίρνει σαν τιμή αυτό που έχει οριστεί στο tblReports.reportCode για το συγκεκριμένο report
- Η παράμετρος crystalParameter παίρνει σαν τιμή αυτό που έχει οριστεί στο tblReportCriteria.RCCrystalParameterName



Παράδειγμα

```
function UseSelectedReportLines(reportCode, crystalParameter) {
    if (reportCode == "SalesOrdersTotal" && crystalParameter == "sums") {
        var lines = SF.getSelectedRows();
        var message = "";
        var totalOrders = 0;
        var totalQuantity = 0;
        var totalValue = 0;
        for (i = 0, len = lines.length; i < len; i++) {
            message += String(lines[i][0]) + " " + String(lines[i][1]) + "\n";
            totalOrders += Number(lines[i][1])
            totalQuantity += Number(lines[i][2])
            totalValue += Number(lines[i][7])
        }
        message += "orders in selection: " + totalOrders + "\n";
        message += "quantity in selection: " + totalQuantity + "\n";
        message += "value in selection: " + totalValue;
        SF.showAlert(message, 'ok', null);
    } else if (crystalParameter == "test") {
        SF.showAlert('test', 'ok', null)
    } else {
        SF.showAlert('No script available', 'ok', null);
    }
}
```

Σημείωση

Στα reports συχνά γίνεται χρήση html ώστε τα αντικείμενα μια στήλης να εμφανίζονται bold, με άλλα χρώματα κλπ.

Επειδή τα reports δεν κάνουν καμία αλλαγή στα δεδομένα και απλά δείχνουν τις τιμές που φέρνει το query του report, οι τιμές που επιστρέφει η SF.getSelectedRows() είναι αυτές που επιστρέφει το query αυτούσιες.

Αυτό σημαίνει ότι αν το query επιστρέφει κάτι με html μέσα (πχ select '' || tblinvmaster.invmasterid || '' as Value) τότε όταν στο script πάρει την τιμή για εκείνη την στήλη θα πάρει κάτι του στυλ "25456" και μετά είναι ευθύνη του consultant να βρει τρόπο να αφαιρέσει τα html tags ώστε να πάρει την πραγματική πληροφορία που περιέχεται στην στήλη.

3.3.6 Merchandising

Όνομα script **CUName = andMerchandizing** αφορά το task του merchandising και βοηθάει να ενσωματώσουμε κώδικα και να αυτοματοποιήσουμε τη συμπλήρωση των πεδίων καθώς και να προσθέσουμε ελέγχους.

Event: OnAfterGridValueChanged (columnDatamemeber, cellValue)

Event: ActivityOnCustomEvent (action, slmtaskId) το action έχει την τιμή AfterMerchandizeSavePrinting και το slmTaskId.

4. Sales Order (old)

Όνομα script **CUName = andSalesOrder**

4.1 Γενικές Συναρτήσεις

Οι πρόσβαση στις συναρτήσεις στα script γίνεται με το ίδιο τρόπο όπως το core.

Οι συναρτήσεις είναι:

- **showAlert**
- **query**
- **command**
- **getSettings**
- **getListViewRows**
- **getListViewDetailByIndex**

Αυτές οι συναρτήσεις είναι ίδιες με του core και έχουν αναλυθεί εκτενώς στο [Κεφάλαιο 1.1](#)

Επιπρόσθετες είναι:

4.1.1 Λίστες

Λίστα του header με listname=headerlist.

- Η **getListViewDetailByKey(listName, key, value)** μπορεί να λειτουργήσει βάζοντας ως key το αντίστοιχο datamember που αντιστοιχεί στη γραμμή.

4.1.2 Grid

Grid του καταλόγου

- **getGridCurrentRowValue(String key):** περνώντας το key(=datamember) σαν παράμετρο επιστρέφει σαν String την τιμή του αντίστοιχου κελιού στην γραμμή που έχει επιλέξει ο χρήστης.
- **setGridCurrentRowValue(String key, String value):** περνώντας το key(=datamember) σαν παράμετρο μπορεί κανείς να θέσει σαν String την τιμή του αντίστοιχου κελιού στην γραμμή που έχει επιλέξει ο χρήστης.

Σημείωση

Όταν φέρνουμε τιμή από τον κατάλογο (SF.getGridCurrentRowValue()) διορθώθηκε στην περίπτωση του κενού κελιού να φέρνει κενό String (έφερνε ειδικό χαρακτήρα unicode(160)=Non Breaking Space) . Έτσι, έπρεπε να γίνει έλεγχος στο scripting για αυτόν τον χαρακτήρα. Πριν από κάποια εφαρμογή scripting καλό είναι να ελέγχεται η περίπτωση του κενού στο κελί, ειδικά αν χρησιμοποιείται κάποια συνάρτηση με χρήση της τιμής του πχ. parseFloat.

Στην αλλαγή τιμής ενός κελιού μέσω scripting προστέθηκε η δυνατότητα για HTML formatting. Καλό είναι να γίνεται σε ξεχωριστή μεταβλητή το concatenation και όχι απευθείας στη μέθοδο SF.setGridCurrentRowValue().

```
Πχ.      var codcodehtml = '<font color=\'blue\'>'+codcode+'</font>';  
  
          SF.setGridCurrentRowValue('codcode', codcodehtml.toString());  
  
//var codcode2 = [codcode,'bla'].join(""); //άλλος τρόπος
```

4.1.3 Products list in details

- **getProductsRowCount()**: μετρά ποσες εγγραφές υπάρχουν στο details
- **getProductsRowCountByField(String dataMember)**: επιστρέφει σε ποσες εγγραφές του details έχει συμπληρωθεί το πεδίο dataMember
- **setListViewDetailByIndex (listView, index, option, value)** --Αγνωστο
- **setListViewDetailByKey (listView, index, option, value)** --ΑΓνωστο

4.2 Events

Όνομα script **CUName = andSalesOrder**

Υποστηρίζονται:

- Tokens: \$SLMID\$, \$CUSID\$, \$ID\$ (invmaster id)
- Το αντικείμενο SF για την κλήση γενικών συναρτήσεων του [Sales old](#).
- getActivityValueByKey(key): υποστηρίζονται τα netvalue, vatvalue της συγκεκριμένης παραγγελίας.

ActivityOnBeforeSave() εκτελείται πριν γίνει το save. περιμένει true/false για το αν θα συνεχίσει.

CatalogueOnAfterChanged(dataMember) εκτελείται αφού εισάγει μια τιμή στο grid.



Παράδειγμα ActivityOnBeforeSave

```
function ActivityOnBeforeSave() {  
    var sdt = SF.getListViewDetailByKey('headerlist','sdtid','storevalue');  
    if(sdt != '1') { //if the sdtid of the current header is different than 1 then the  
save operation is continued  
        return 'true';  
    }  
    var netvalue = SF.getActivityValueByKey('netvalue');  
    if(netvalue >= 10) { //if the current netvalue is more than 10 then the save operation  
is continued  
        return 'true';  
    }  
    //if the sdtid=1 and netvalue<10 then the save operation must not be completed. A  
query is executed to get the last price for the current customer and an alert is shown to the  
user.  
    var result = SF.query('select tdogrossvalue from tblinvmaster where slmid=$SLMID$ and  
cusid=$CUSID$ order by dockatahorisidate desc Limit 1');  
  
    SF.showAlert('Saving of the document is not allowed. Last Price = '+result, '', null);  
    return 'false';  
}
```



Παράδειγμα CatalogueOnAfterChanged

```
function CatalogueOnAfterChanged(datamember) {  
    //if the user has edited cusstock column  
    if(datamember == 'cusstock') {  
        //we get the current cusstock value as a String  
        var stringValue = SF.getGridCurrentRowValue('cusstock');  
        //convert to a float value in order to do mathematical computations  
        var cusstock = parseFloat(stringValue)+1;  
        //set the value of qtya to cusstock,  
        after the conversion from float to String  
        SF.setGridCurrentRowValue('qtya', cusstock.toString());  
    }  
}
```



Παράδειγμα πρότασης παραγγελίας

```
function CatalogueOnAfterChanged(datamember) {  
    if(datamember == 'cusstock') {  
        //if the user has edited cusstock column  
        //we get the current cusstock value as a String  
        var cusstockstring = SF.getGridCurrentRowValue('cusstock');  
        var previousstockstring = SF.getGridCurrentRowValue('previousstock');  
        var previousorderstring = SF.getGridCurrentRowValue('previousorder');  
  
        //convert to a float value in order to do mathematical computations  
        var suggestedqty =  
            parseFloat(previousorderstring) + parseFloat(previousstockstring) -  
            parseFloat(cusstockstring);  
  
        //set the value of qtya, after the conversion from float  
        //to String  
        if (suggestedqty<0){  
            suggestedqty=0;  
            SF.setGridCurrentRowValue('qtya', suggestedqty.toString());  
        } else {  
            if(isNaN(suggestedqty)) {  
                suggestedqty=0;  
                SF.setGridCurrentRowValue('qtya', "");  
            } else {  
                SF.setGridCurrentRowValue('qtya', suggestedqty.toString());  
            }  
        }  
    }  
}
```



Παράδειγμα ελέγχου ημερομηνίας παράδοσης βάσει του πεδίου nonDeliveryDays

```
function ActivityOnBeforeSave() {

    //Get the value of nondeliverydays for the customer
    var DeliveryDays = SF.query('select nondeliverydays from cus where cusid=$CUSID$');
    //Get the docrequestdate value for this order
    Var DeliveryDay = SF.getListViewDetailByKey('headerlist', 'docrequestdate',
'storevalue');

    if (DeliveryDays.length == 5) {
        DeliveryDays = DeliveryDays + '00';
    }

    //If the customer accepts deliveries for every day of the week save the order
    if (DeliveryDays == '0000000') {
        return 'true';
    } else {

        //split the date string in 3 parts according to the dashes
        var parts = DeliveryDay.split('-');

        //Parse string YYYY-MM-DD to date format compatible with getDay()
        var dDeliveryDay = new Date(parts[0], parts[1] - 1, parts[2]);

        //Get the weekday number of the date
        var DayOfWeek = dDeliveryDay.getDay();

        //Get the correct character of the nondeliverydays field according to the
above weekday.
        //Sunday is 0 and Monday is 1, so subtract one to make Monday 0
        var AllowOrder = DeliveryDays.charAt(DayOfWeek - 1);

        //if the customer does not allow delivery on this date
        if (AllowOrder == '1') {

            //Get the name of each day
            var WeekDay = new Array(7);
            WeekDay[0] = 'Κυριακή';
            WeekDay[1] = 'Δευτέρα';
            WeekDay[2] = 'Τρίτη';
            WeekDay[3] = 'Τετάρτη';
            WeekDay[4] = 'Πέμπτη';
            WeekDay[5] = 'Παρασκευή';
            WeekDay[6] = 'Σάββατο';

            var WeekName = WeekDay[DayOfWeek];

            //Alert the user
            SF.showAlert('Ο πελάτης δεν παραλαμβάνει την ' + WeekName + '.
Παρακαλώ επιλέξτε άλλη ημερομηνία παράδοσης.', 'OK', null);
            //Do not save the order
            return 'false';
        } else {
            return 'true';
        }
    }
}
```

5. Xvan

Όνομα script CUName = andXVan

5.1 Γενικές Συναρτήσεις

- showAlert
- query
- command
- getSettings
- getListViewRows
- getListViewDetailByIndex

Αυτές οι συναρτήσεις είναι ίδιες με του core και έχουν αναλυθεί εκτενώς στο [Κεφάλαιο 1.1](#)

Επιπρόσθετες είναι:

5.1.1 Λίστες

- **getListViewDetailByKey(listName, key, value)**

5.1.2 Grid

- **getGridCurrentRowValue(String key)**
- **setGridCurrentRowValue(String key, String value)**

5.1.3 Products list in details

- **getProductsRowCount()**
- **getProductsRowCountByField(String dataMember)**
- **setListViewDetailByIndex (listView, index, option, value) --Αγνωστο**
- **setListViewDetailByKey (listView, index, option, value) --Αγνωστο**

Αυτές οι συναρτήσεις είναι ίδιες με του sales old και έχουν αναλυθεί εκτενώς στο [Κεφάλαιο 3.1.1](#)

5.2 Events

Υποστηρίζονται:

- Tokens: \$SLMID\$, \$CUSID\$
- Το αντικείμενο SF για την κλήση γενικών συναρτήσεων του XVan

Event: ActivityOnBeforeSave() εκτελείται πριν γίνει το save. περιμένει true/false για το αν θα συνεχίσει.

Παραδείγματα έχουν αναφερθεί στο [Sales Order](#).

Ένα ακόμα event με διαφορετικό όνομα script **CUName = xvanHeader**

Υποστηρίζονται:

- Tokens: \$SLMID\$, \$CUSID\$, \$ID\$ (invmaster id)

CalculateValuesInHeader-- Αγνωστο

6. Sales Order Plus (New)

Όνομα script **CUName = andSalesOrderPlus**

6.1 Σημεία Scripting

Υπάρχουν τρία σημεία στα οποία μπορούμε να εκτελεστούν JavaScript functions στο Sales/xVan. Τα σημεία αυτά είναι

- Όταν θέτουμε μια νέα τιμή σε κάποιο κελί του grid (CatalogueOnAfterChanged)
- Όταν θέτουμε μια τιμή σε κάποιο πεδίο του header (Η ίδια μέθοδος θα τρέξει και κατά την είσοδό μας στο Sales/xVan) (HeaderPropertyChanged)
- Όταν πατήσουμε το κουμπί αποθήκευσης. (ActivityOnBeforeSave)

6.2 Πρόσβαση Στα Αντικείμενα Sales Order Plus

Σε αυτό το σημείο, θα πρέπει να αναφέρουμε ότι τα scripts έχουν σαν ορίσματα και αλληλοεπιδρούν με Java αντικείμενα. Σαν αποτέλεσμα, οι getters και setters μέθοδοι των αντικειμένων αυτών θα πρέπει να διαχειριστούν από τον implementor ώστε να μπορεί να τις χρησιμοποιήσει στο script του.

Μια προσέγγιση θα ήταν να μετατρέψει την τιμή μιας getter μεθόδου σε κάτι που έχει αξία στο script του. Η άλλη θα ήταν να την χειριστεί χρησιμοποιώντας απευθείας Java. Για παράδειγμα, στο αντικείμενο που αναφέρεται στις γραμμές του παραστατικού το πεδίο που αποθηκεύεται η ποσότητα A είναι τύπου `java.math.BigDecimal`. Με την πρώτη προσέγγιση θα μπορούσαμε να την μετατρέψουμε σε `float` για να την χρησιμοποιήσουμε στο script (αν θελήσουμε να την θέσουμε θα πρέπει να την μετατρέψουμε πάλι σε `BigDecimal`). Με τη δεύτερη προσέγγιση απλά θα χρησιμοποιούσαμε τις μεθόδους που παρέχονται από την κλάση `BigDecimal`.

Στο [ενότητα 5.4](#) παρέχεται ο ακριβής προσδιορισμός των getters/setters του κάθε αντικειμένου με το οποίο μπορούμε να αλληλεπιδράσουμε με τα scripts

Τα παρακάτω παραδείγματα είναι ισοδύναμα.



Παράδειγμα 1η προσέγγιση

```
var qtyA = parseFloat(product.getQuantityA().toString())

If (qtyA == 0) {
    //do something
}
```



Παράδειγμα 2η προσέγγιση

```
If(product.getQuantityA().compareTo(0) == 0) {
    //do something
}
```

6.3 Γενικές Συναρτήσεις

- **showAlert(message):** Η μέθοδος αυτή μας επιτρέπει να εμφανίσουμε μια ειδοποίηση στον χρήστη. Αυτή εμφανίζεται στο κάτω μέρος της οθόνης με την μορφή `snackbar`. Η σύνταξη της είναι η ακόλουθη:



Παράδειγμα

```
showAlert('Εμφάνισε αυτό το μήνυμα');
```

- **showDialog(message):** Με τη μέθοδο αυτή εμφανίζουμε μια ειδοποίηση στο χρήστη με τη μορφή `dialog`. Η σύνταξη της φαίνεται παρακάτω



Παράδειγμα

```
showDialog('Εμφάνισε αυτό το μήνυμα');
```

- **query(sql):** Η μέθοδος αυτή μας επιτρέπει να εκτελέσουμε `selects` στην βάση (οποιοδήποτε άλλου τύπου `query` δεν εκτελείται). Η σύνταξή της είναι η ακόλουθη:



Παράδειγμα

```
var result = query('select * from somewhere');
```

Το αποτέλεσμα που επιστρέφει είναι ένας διδιάστατος πίνακας. Κάθε σειρά αντιστοιχεί σε μια εγγραφή όπως επιστρέφεται από το `select` μας. Κάθε στήλη αντιστοιχεί στις στήλες του `select` μας (με την ίδια σειρά που έχουν οριστεί σε αυτό).

showWarning(message): Η μέθοδος αυτή μας επιτρέπει να εμφανίσουμε μια ειδοποίηση στον χρήστη που μπορεί να τον αποτρέψει από το να αποθηκεύσει. Η μέθοδος εμφανίζει ένα παράθυρο με τίτλο `'Warning'`, κάποιο μήνυμα που το ορίζει ο `consultant` και επιλογές `'OK'` & `'CANCEL'`.

Όταν πατάει ο Χρήστης `CANCEL` το παράθυρο κλείνει και δεν συμβαίνει τίποτα άλλο, όταν όμως πατάει `OK` η εφαρμογή αρχίζει την διαδικασία της αποθήκευσης όπως θα γινόταν αν δεν υπήρχε το `script`.

Προσοχή

- Αν χρησιμοποιηθεί η μέθοδος αυτή το θα πρέπει αυτός που γράφει το script υποχρεωτικά να επιστρέψει **false**, διαφορετικά δεν θα δουλέψει σωστά.
- Η μέθοδος αυτή μπορεί να κληθεί μόνο μέσα από την **ActivityOnBeforeSave**. Αν κληθεί από οπουδήποτε αλλού θα προκληθεί error.

Η σύνταξη της είναι η ακόλουθη:

Παράδειγμα

```
showWarning('Εμφάνισε αυτό το μήνυμα');  
return false;
```

6.4 Πεδία Αντικειμένων

6.4.1 Headers Standard Fields

Οι getters/setters που παρουσιάζονται στον πίνακα είναι case sensitive

Πεδίο	Java Type	Getter	Setter
tdotraderdiscpcnt	BigDecimal	gettdotraderdiscpcnt	settdotraderdiscpcnt
tdopaymentdiscpcnt	BigDecimal	gettdopaymentdiscpcnt	settdopaymentdiscpcnt
invmasterid	Integer	getinvmasterid	setinvmasterid
sdtid	Integer	getsdtid	setsdtid
slmtaskid	Integer	getslmtaskid	setslmtaskid
dosid	Integer	getdosid	setdosid
docnumber	Integer	getdocnumber	setdocnumber
dockatahorisidate	Date	getdockatahorisidate	setdockatahorisidate
doccreatedondate	Date	getdoccreatedondate	setdoccreatedondate
docekdosidate	Date	getdocekdosidate	setdocekdosidate
docrequestdate	Date	getdocrequestdate	setdocrequestdate
cusid	Integer	getcusid	setcusid
adridlogistiriou	Integer	getadridlogistiriou	setadridlogistiriou
adridproorismou	Integer	getadridproorismou	setadridproorismou
curid	Integer	getcurid	setcurid
slmid	Integer	getslmid	setslmid
cpmid	Integer	getcpmid	setcpmid
wrhid	Integer	getwrhid	setwrhid
asdraft	Integer	getasdraft	setasdraft
replid	Integer	getReplID	setReplID
visitid	Integer	getVisitid	setVisitid
issynced	Integer	getissynced	setissynced
blockorder	Integer	getBlockorder	setBlockorder
cusiddebit	Integer	getCusIdDebit	setCusIdDebit
tdoqtya	BigDecimal	getSumQtyA	setSumQtyA
tdoqtyb	BigDecimal	getSumQtyB	setSumQtyB
sumitemdiscountvalue	BigDecimal	getSumItemDiscountValue	setSumItemDiscountValue
sumpricelistdiscountvalue	BigDecimal	getSumPricelistDiscountValue	setSumPricelistDiscountValue
tdotraderdisc	BigDecimal	getSumCustomerDiscountValue	setSumCustomerDiscountValue
tdopaymentdisc	BigDecimal	getSumPaymentDiscountValue	setSumPaymentDiscountValue
sumfirstdiscountvalue	BigDecimal	getSumFirstDiscountValue	setSumFirstDiscountValue
sumseconddiscountvalue	BigDecimal	getSumSecondDiscountValue	setSumSecondDiscountValue
sumDiscountValue	BigDecimal	getSumDiscountValue	setSumDiscountValue
tdogrossvalue	BigDecimal	getSumGrossValue	setSumGrossValue
tdonetstock	BigDecimal	getSumNetValue	setSumNetValue
tdovatstock	BigDecimal	getSumVatValue	setSumVatValue

sumefkvalue	BigDecimal	getSumEfkValue	setSumEfkValue
tdosumvalue	BigDecimal	getTdoSumValue	setTdoSumValue
totallines	Integer	getTotalLines	setTotalLines
tdoweight	BigDecimal	getTdoweight	setTdoweight
Nextinvmasterid	Integer	getNextInvMasterId	setNextInvMasterId
previousinvmasterid	Integer	getPreviousInvMasterId	setPreviousInvMasterId
docidakiromeno	Integer	getDocidakiromeno	setDocidakiromeno
docidakirotiko	Integer	getDocidakirotiko	setDocidakirotiko
isprinted	Integer	getIsPrinted	setIsPrinted

6.4.2 Product Standard Fields

Οι getters/setters που παρουσιάζονται στους πίνακες είναι case sensitive

Πεδίο	Java Type	Getter	Setter
stdqtya	BigDecimal	getQuantityA	setQuantityA
stdqtyb	BigDecimal	getQuantityB	setQuantityB
packagingqty	BigDecimal	getPackagingQty	setPackagingQty
stiratiofrom	BigDecimal	getStiRatioFrom	setStiRatioFrom
stiratioto	BigDecimal	getStiRatioTo	setStiRatioTo
wrhstock	BigDecimal	getWrhStock	setWrhStock
stiid	Integer	getStiid	setStiid
invdetailid	Integer	getInvDetailId	setInvDetailId
invmasterid	Integer	getInvMasterId	setInvMasterId
linenumber	Integer	getLineNumber	setLineNumber
wrhid	String	getWrhid	setWrhid
setid	Integer	getSetId	setSetId
issynced	Integer	getIsSynced	setIsSynced
isclone	Integer	getIsClone	setIsClone
ssdweight	BigDecimal	getRowWeight	setRowWeight
weightPerUnit	BigDecimal	getWeightPerUnit	setWeightPerUnit
expiredqty	BigDecimal	getExpiredQuantityA	setExpiredQuantityA
expiredqtyb	BigDecimal	getExpiredQuantityB	setExpiredQuantityB
freshqty	BigDecimal	getFreshQuantityA	setFreshQuantityA
freshqtyb	BigDecimal	getFreshQuantityB	setFreshQuantityB
updqtya	Integer	getUpdQtyA	setUpdQtyA
uniida	Integer	getUniida	setUniida
replid	Integer	getReplid	setReplid
isDeleted	Integer	getIsDeleted	setIsDeleted

6.4.3 Discount Fields

Για να προσπελάσουμε πεδία των εκπτώσεων, πρέπει πρώτα καλέσουμε την μέθοδο `getProductItemDiscounts()` και μετά τον αντίστοιχο `getter/setter`. Για παράδειγμα στο όρισμα `product` της `CatalogueOnAfterChanged` θα έπρεπε να γράψουμε «`product.getProductItemDiscounts().getPrice();`» για να ανακτήσουμε την τιμή του προϊόντος.

Πεδίο	Java Type	Getter	Setter
ssdunitprice	BigDecimal	getPrice	setPrice
ssditemdiscpcnt	BigDecimal	getItemDiscountPercent	setItemDiscountPercent
ssdpricelistdiscpcnt	BigDecimal	getPricelistDiscountPercent	setPricelistDiscountPercent
disc1pcnt	BigDecimal	getFirstDiscountPercent	setFirstDiscountPercent
disc2pcnt	BigDecimal	getSecondDiscountPercent	setSecondDiscountPercent
efkperunit	BigDecimal	getEfkPerUnit	setEfkPerUnit
ssdvatpcnt	BigDecimal	getVatPercent	setVatPercent
ssdvatvalue	BigDecimal	getVatValue	setVatValue
vatid	Integer	getVatId	setVatId
ssdnetvalue	BigDecimal	getNetValue	setNetValue
ssdgrossvalue	BigDecimal	getGrossValue	setGrossValue
efkvalue	BigDecimal	getEfkValue	setEfkValue
ssditemdisc	BigDecimal	getItemDiscountValue	setItemDiscountValue
ssdpricelistdisc	BigDecimal	getPricelistDiscountValue	setPricelistDiscountValue
ssdtraderdisc	BigDecimal	getCustomerDiscountValue	setCustomerDiscountValue
ssdpaymentdisc	BigDecimal	getPaymentDiscountValue	setPaymentDiscountValue
disc1	BigDecimal	getFirstDiscountValue	setFirstDiscountValue
disc2	BigDecimal	getSecondDiscountValue	setSecondDiscountValue
ssdtraderdiscpcnt	BigDecimal	getCustomerDiscountPercent	setCustomerDiscountPercent
ssdpaymentdiscpcn	BigDecimal	getPaymentDiscountPercent	setPaymentDiscountPercent

6.4.3.1 Special Discount Fields

Για να προσπελάσουμε πεδία των special discounts, πρέπει πρώτα καλέσουμε την μέθοδο getProductItemSpecialDiscounts() και μετά τον αντίστοιχο getter/setter. Για παράδειγμα στο όρισμα product της CatalogueOnAfterChanged θα έπρεπε να γράψουμε «product.getProductItemSpecialDiscounts().getSpecialDiscountType ();» για να ανακτήσουμε τον τύπο του special discount.

Πεδίο	Java Type	Getter	Setter
specialDiscountType	Integer	getSpecialDiscountType	setSpecialDiscountType
appliedItemDiscountValue	Boolean	isAppliedItemDiscountValue	setAppliedItemDiscountValue
appliedPricelistDiscountValue	Boolean	isAppliedPricelistDiscountValue	setAppliedPricelistDiscountValue
debugMessage	String	getDebugMessage	setDebugMessage
appliedDisc1Value	Boolean	isAppliedDisc1Value	setAppliedDisc1Value
appliedDisc2Value	Boolean	isAppliedDisc2Value	setAppliedDisc2Value
itemDiscountUnitValue	BigDecimal	getItemDiscountUnitValue	setItemDiscountUnitValue
pricelistDiscountUnitValue	BigDecimal	getPricelistDiscountUnitValue	setPricelistDiscountUnitValue
disc1UnitValue	BigDecimal	getDisc1UnitValue	setDisc1UnitValue
disc2UnitValue	BigDecimal	getDisc2UnitValue	setDisc2UnitValue
stiDiscGroupId	Integer	getStiDiscGroupId	setStiDiscGroupId

6.5 Extra Fields

Τα αντικείμενα header και product εκτός από τα πεδία που έχουν ενσωματωμένα πάντα, μπορούν να έχουν και πεδία τα οποία έχουν οριστεί στο salesConfig. Για να χρησιμοποιήσει κάποιος στο script του τα πεδία αυτά, θα πρέπει να καλέσει την μέθοδο `getExtraFields()` (έχουμε το ίδιο όνομα μεθόδου και για το product και για το header). Η μέθοδος αυτή επιστρέφει ένα Map, το οποίο έχεις ως key το datamember του πεδίου και σαν value ένα αντικείμενο που περιλαμβάνει τα εξής δύο String πεδία :

- `displayValue` (τιμή που φαίνεται)
- `storeValue` (τιμή που αποθηκεύεται)

Για να προσπελάσουμε αυτά τα πεδία χρησιμοποιούμε τους παρακάτω getters/setters (οι οποίοι είναι προφανώς case sensitive):

- `displayValue` -> `getDisplayValue`, `setDisplayValue`
- `storeValue` -> `getStoreValue`, `setStoreValue`



Παράδειγμα

Έστω ότι έχουμε ορίσει ένα πεδίο για τα details μας με datamember `batman`. Για να διαβάσουμε την τιμή του στην `CatalogueOnAfterChanged` θα πρέπει να γράψουμε το εξής

```
function CatalogueOnAfterChanged(rowIndex, product, datamember) {  
    var batman = product.getExtraFields().get("batman").getDisplayValue();  
    //do something with batman  
}
```

6.6 Events

Υποστηρίζονται:

- Tokens: \$SLMID\$, \$CUSID\$, \$VISITID\$
- getActivityValueByKey(key): υποστηρίζονται τα netvalue, vatvalue της συγκεκριμένης παραγγελίας.
- **CatalogueOnAfterChanged(rowIndex, product, dataMember)**: Εκτελείται κάθε φορά που αλλάζει η τιμή ενός κελιού στο grid. Έχει τρία ορίσματα τα οποία μπορείτε να χρησιμοποιήσετε.
 - **rowIndex** -> Ο αριθμός της γραμμής που άλλαξε
 - **product** -> Αντικείμενο που περιλαμβάνει όλα τα δεδομένα της γραμμής, είτε αυτά φαίνονται είτε όχι. (Δηλαδή περιλαμβάνει τα πεδία που ορίζει ο implementor στο salesConfig αλλά και αυτά που δεν χρειάζεται να προσθέσει γιατί τα προσθέτει από μόνο του το Sales/xVan. Στο τέλος του documentation παρουσιάζονται ποια πεδία περιλαμβάνει)
 - **datamember** -> Το datamember το κελιού που άλλαξε

Σημείωση: Η γραμμή ΔΕΝ έχει αποθηκευτεί στη βάση τη στιγμή που εκτελείται το script!



Παράδειγμα

```
function CatalogueOnAfterChanged(rowIndex, product, datamember) {
    //Αν αλλάξαμε το πεδίο stdqtya, θέσε το πεδίο stdqtyb ίσο με 1
    if (datamember == 'stdqtya') {
        product.setQuantityB(new BigDecimal(1));
    }
}
```

- **HeaderPropertyChanged(header)**: Η μέθοδος αυτή καλείται κάθε φορά που αλλάζει η τιμή ενός πεδίου στο tab header καθώς και κατά την είσοδο μας στο Sales/xVan. Έχει ένα όρισμα
 - **header** -> αντικείμενο που περιλαμβάνει όλες τις τιμές του header (Δηλαδή τα πεδία που ορίζει ο implementor στο salesConfig αλλά και αυτά που δεν χρειάζεται να προσθέσει, γιατί τα προσθέτει από μόνο του το Sales/xVan. Στο τέλος του documentation παρουσιάζονται ποια πεδία περιλαμβάνει).

Η μέθοδος αυτή θα πρέπει να επιστρέφει μια boolean τιμή για να ξέρουμε αν θέλουμε να θέσουμε τις αλλαγές ή όχι. (true -> τις θέτουμε, false -> δεν τις θέτουμε)



Παράδειγμα

```
function HeaderPropertyChanged(header) {
    If(header.getcusid() == 1123) {
        //do something
    }
    return true;
}
```

- **ActivityOnBeforeSave(header)**: Η μέθοδος αυτή καλείται την στιγμή που θα πατήσουμε το κουμπί αποθήκευσης. Τρέχει ακριβώς πριν την αποθήκευση και το αποτέλεσμα της καθορίζει αν θα συνεχιστεί η αποθήκευση ή όχι. Έχει ένα όρισμα

- **header** -> αντικείμενο που περιέχει όλα τα δεδομένα της κεφαλίδας. (Δηλαδή τα πεδία που ορίζει ο implementor στο salesConfig αλλά και αυτά που δεν χρειάζεται να προσθέσει γιατί τα προσθέτει από μόνο του το Sales/xVan. Στο τέλος του documentation παρουσιάζονται ποια πεδία περιλαμβάνει).

Και σε αυτή τη μέθοδο θα πρέπει να επιστρέφουμε μια boolean τιμή για να ξέρουμε αν θα συνεχίσουμε τη ροή του προγράμματος ή όχι. (true -> συνεχίζουμε στη διαδικασία αποθήκευσης, false -> σταματάμε).



Παράδειγμα

```
function ActivityOnBeforeSave(header) {  
    If(parseInt(header.getBlockorder()) == 1) {  
        return false;  
    }  
    return true;  
}
```

7. Collection

Όνομα script **CUName = andCollection**

7.1 Γενικές συναρτήσεις

Οι πρόσβαση στις συναρτήσεις στα script γίνεται με το ίδιο τρόπο όπως το core.

Οι συναρτήσεις είναι:

- **showAlert**
- **query**
- **command**
- **getSettings**
- **getListViewRows**
- **getListViewDetailByIndex**
- **getListViewDetailByKey**

Αυτές οι συναρτήσεις είναι ίδιες με του core και έχουν αναλυθεί εκτενώς στο [Κεφάλαιο 1.1](#)

Επιπρόσθετες είναι:

- **getActivityValueByKey(key)**: υποστηρίζονται τα netvalue, vatvalue της συγκεκριμένης παραγγελίας.

Αυτές που αφορούν το **GRID**:

- **getGridCurrentRowValue (key)**: περνώντας το key(=datamember) σαν παράμετρο επιστρέφει σαν String την τιμή του αντίστοιχου κελιού στην γραμμή που έχει επιλέξει ο χρήστης
- **setGridCurrentRowValue (key, value)**: περνώντας το key(=datamember) σαν παράμετρο μπορεί να θέσει σαν String την τιμή του αντίστοιχου κελιού που έχει επιλέξει ο χρήστης.

Σημείωση

Όταν φέρνουμε τιμή από τον κατάλογο (SF.getGridCurrentRowValue()) διορθώθηκε στην περίπτωση του κενού κελιού να φέρνει κενό String (έφερνε ειδικό χαρακτήρα unicode(160)=Non Breaking Space) . Έτσι, έπρεπε να γίνει έλεγχος στο scripting για αυτόν τον χαρακτήρα. Πριν από κάποια εφαρμογή scripting καλό είναι να ελέγχεται η περίπτωση του κενού στο κελί, ειδικά αν χρησιμοποιείται κάποια συνάρτηση με χρήση της τιμής του πχ. parseFloat.

Στην αλλαγή τιμής ενός κελιού μέσω scripting προστέθηκε η δυνατότητα για HTML formatting. Καλό είναι να γίνεται σε ξεχωριστή μεταβλητή το concatenation και όχι απευθείας στη μέθοδο SF.setGridCurrentRowValue().

```
Πχ.      var codcodehtml = '<font color=\'blue\'>'+codcode+'</font>';
          SF.setGridCurrentRowValue('codcode', codcodehtml.toString());

//var codcode2 = [codcode,'bla'].join(""); //άλλος τρόπος
```

Αυτές που αφορούν τις **Products list in Details**:

- **getProductsRowCount ()**: μετρά πόσες εγγραφές υπάρχουν στο details.
- **getProductsRowCountByField (datamember)**: επιστρέφει σε ποσες εγγραφές του details έχει συμπληρωθεί το πεδίο dataMember

Αυτές που αφορούν τις **Λίστες**:

- Λίστα του header με listname=headerlist
 - Η **getListViewDetailByKey()** μπορεί να λειτουργήσει βάζοντας ως key το αντίστοιχο datamember που αντιστοιχεί στη γραμμή.
- **setListViewDetailByIndex (listView, index, option, value)** --Αγνωστο
- **setListViewDetailByKey (listView, index, option, value)** --ΑΓνωστο

7.2 Events

Υποστηρίζονται:

- Tokens: \$SLMID\$, \$CUSID\$
- Το αντικείμενο SF για την κλήση γενικών συναρτήσεων του Collection.

Event: ActivityOnBeforeSave() εκτελείται πριν γίνει το save. περιμένει true/false για το αν θα συνεχίσει.

8. Field Service

Όνομα script **CUName = andFieldService**

8.1 Γενικές Συνρτήσεις

Οι πρόσβαση στις συναρτήσεις στα script γίνεται με το ίδιο τρόπο όπως το core.

Οι συναρτήσεις είναι:

- **showAlert**
- **query**
- **command**
- **getSettings**
- **getActivityValueByKey**

Αυτές οι συναρτήσεις είναι ίδιες με του core και έχουν αναλυθεί εκτενώς στο [Κεφάλαιο 1.1](#)

Επιπρόσθετες είναι:

- **getSpareParts()**: Επιστρέφει όλες τις εγγραφές του spare parts σε ένα array Nx2. Εάν δεν υπάρχει απάντηση επιστρέφει ('#VALUE!', '#VALUE!').
- Πρώτη στήλη είναι το **stiid** και η δεύτερη είναι το **qtyA**.
- **getServices()**: Παίρνει όλες τις κατηγορίες των services που υπάρχουν στο spinner της οθόνης. Επιστρέφει ένα Nx1 όπου περιέχει το **Service id**.
- **getDamageRecordsDetails()**: Επιστρέφει όλες τις εγγραφές του damage details σε ένα array Nx2. Εάν δεν υπάρχει απάντηση επιστρέφει ('#VALUE!', '#VALUE!').
- Πρώτη στήλη είναι το **datamember** και η δεύτερη είναι το **value**.

8.2 Events

Υποστηρίζονται:

- Tokens: \$SCID\$
- Το αντικείμενο SF για την κλήση γενικών συναρτήσεων του Field Service.

Event: OnBeforeSave() εκτελείται πριν γίνει το save. περιμένει true/false για το αν θα συνεχίσει.

9. Surveys

Για να εκτελέσουμε script στο surveys πρέπει στον πίνακα **tblSurveys** να υπάρχει η στήλη **Javascript** και να τοποθετήσουμε εκεί το script.

9.1 Γενικές Συνρτήσεις

Οι πρόσβαση στις συναρτήσεις στα script γίνεται με το ίδιο τρόπο όπως το core.

Οι συναρτήσεις είναι:

- **showAlert**
- **query**
- **command**
- **getSettings**
- **getActivityValueByKey**

Αυτές οι συναρτήσεις είναι ίδιες με του core και έχουν αναλυθεί εκτενώς στο [Κεφάλαιο 1.1](#)

Επιπρόσθετες είναι:

- **getAnswersStoreValuesByQuestionId (id):** Με τη συνάρτηση αυτή, μου δίνεται η δυνατότητα να πάρω τις απαντήσεις του survey με παράμετρο το id μέσα σε script.
- Επιστρέφει ένα array Nx2 με όλες τις απαντήσεις (=storeValues) και τα αντίστοιχα picklistid μιας συγκεκριμένης ερώτησης. Εάν δεν υπάρχει απάντηση επιστρέφει ('#VALUE!', '#VALUE!')
- Πρώτη στήλη είναι το picklistid και δεύτερη το storeValue



Παράδειγμα

```
var array = SF.getAnswersStoreValuesByQuestionId(20);

for (var j in array) {
    SF.showAlert(array[j][0] + '->' + array[j][1], null, null);
}
```

- **getAnswersIds():** Επιστρέφει array μόνο με τα ids των ερωτήσεων invisible ή visible που έχουν προκύψει μέχρι στιγμής από το “μονοπάτι” που έχει ακολουθήσει ο χρήστης πάνω στο δέντρο των ερωτήσεων.



Παράδειγμα

```
var answerIds = SF.getAnswersIds();
for (var i in answerIds) {
    // do something
}
```

- **getAnswersScoreByQuestionId(id, defaultScore):** Επιστρέφει ένα array 2x1 όπου array[0] περιέχει N scores όπου N το πλήθος των απαντήσεων για κάθε ερώτηση και array[1] το weight της ερώτησης



Παράδειγμα

```
var answerScore = SF.getAnswersScoreByQuestionId(20, 5);
```

- **executequery(sql)** : Η μέθοδος αυτή μας επιτρέπει να εκτελέσουμε selects στην βάση (οποιοδήποτε άλλου τύπου query δεν εκτελείται). Η σύνταξή της είναι η ακόλουθη:



Παράδειγμα

```
var result = query('select * from somewhere');
```

Το αποτέλεσμα που επιστρέφει είναι ένας πολυδιάστατος πίνακας. Κάθε σειρά αντιστοιχεί σε μια εγγραφή όπως επιστρέφεται από το select μας. Κάθε στήλη αντιστοιχεί στις στήλες του select μας (με την ίδια σειρά που έχουν οριστεί σε αυτό).

9.2 Events

Είναι απαραίτητο στην περίπτωση που οριστεί JavaScript στο Survey, αυτό να περιέχει **οποσδήποτε** τις 3 παρακάτω μεθόδους

- **CalculateSurveyScore:**
- **OnBeforeSave:** Η συγκεκριμένη συνάρτηση, τρέχει πριν την αποθήκευση και δίνει τη δυνατότητα απαγόρευσης κάποιου survey. Εάν επιστρέψει 'true' (=return 'true') ή null (=return) το survey θα αποθηκευτεί. Σε κάθε άλλη περίπτωση δεν θα αποθηκευτεί.
- **OnNewSurvey**

Ένα ασφαλές template για να ξεκινήσετε custom javascript πάνω σε Survey είναι το παρακάτω.

```
function CalculateSurveyScore() {
    SF.showAlert('CalculateSurveyScore', 'ok', null)
    return 'noscore';
}

function OnBeforeSave(reportCode, crystalParameter) {
    SF.showAlert('OnBeforeSave', 'ok', null)
}

function OnNewSurvey(reportCode, crystalParameter) {
    SF.showAlert('OnNewSurvey', 'ok', null)
}
```

10. Debugger

Για να ενεργοποιηθεί ο debugger θα πρέπει να προστεθεί το λεκτικό "debugger;" στην αρχή του script. Όσο υπάρχει αυτό το λεκτικό, όταν το script εκτελείται θα ανοίγει ένα dialog που θα περιέχει το script μας, ώστε να μπορούμε να το τροποποιήσουμε και να το σώσουμε στην βάση. Πράγμα που σημαίνει ότι σε παραγωγικό περιβάλλον δεν θα πρέπει να περιλαμβάνεται.