

ESPocketBiz – Core & Basic

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	Basic Configuration.....	4
1.1	Introduction	4
1.2	Customer List and Customer Tasks parameters	4
1.3	Formatting the Customer List	5
1.3.1	XML example 1 (queries, near)	8
1.3.2	XML example 2 (show datamembers)	10
1.3.3	XML example 3 (badge stored in the customer row)	11
1.3.4	XML example 4	12
1.4	Badges (on Tasklist, Customerlist etc)	13
1.5	Customer Card (extra financial data).....	15
1.6	sqlOffGetCustomersTasks (configuration)	17
1.7	Customer Actions in tasklist (tblSLMTasks)	18
1.8	sqlOffGetCustomerDetails (configuration).....	19
1.9	Localization	23
1.10	Greek language support in search	24
1.11	HTML Markup.....	25
1.12	Five Days Weather Forecast.....	26
1.13	Open URL to an external browser	27
1.14	New method network call	28

1 Basic Configuration

1.1 Introduction

This manual describes the **basic configuration** and functionality of the **Core** application in Android environment.

1.2 Customer List and Customer Tasks parameters

Parameter (configuration)	Description
CUSTOMERLISTXML	XML for Customer List display customization
BLOCKVISITSQL	Contains the SQL query, the result of which determines if a specific client may be selected from the grid. The SQL query should return the fields: Result and MessageText . <u>Replaces the \$TODAY\$ and \$CUSID\$</u>
sqlOffGetCustomerTasks	The SQL to be executed to bring the customer-specific tasks in offline mode. <u>Replaces the \$CUSID\$ and \$SLMID\$</u> [Mandatory]
sqlOffGetCustomerTasksSimpleVisit	The SQL to be executed to bring the customer-specific tasks in offline mode in case of simple visit. <u>Replaces the \$CUSID\$, \$SLMID\$ and \$SIMPLEVISITSLMTASKID\$</u> [Mandatory]
sqlOffGetCustomerDetails	The SQL to be executed to bring the customer-specific information details in offline mode. <u>Replaces the \$CUSID\$, \$SLMID\$, \$ADRIDDEST\$</u>
SimpleVisitSLMTaskID	If it is 0 it doesn't appear the option for simple visit . If it is set, then in customer information displays and the option for simple visiting with the actions (tasks) of the corresponding slmtaskid
CUSBalanceXML	the xml that overrides the default, which is responsible for the design of the financial data grid.
kartelles	0 (default), 1 displays all extra financial customer info (DSO, open balance)
SQLFINANCIAL	If it has an extraconfigvalue (SQL), checks which fields will appear in the financial tab (overrides default)

1.3 Formatting the Customer List

Depending on the list, we have to place the appropriate XML into the CustomerListXML parameter in the Configuration table:

Parameter name	Description
CUSTOMERLISTXML	XML for Customer List display customization

The XML supports rows and column. Badges are also supported. The field **coordsx** and the field **coordsy** must be completed to show the spots on the map.

Attention: After each change in the XML, must synchronize (in offline mode) and force stop the application (in every case).

The general format of the XML is as follows:

```
<AndroidLayout>
  <sql>
    <online
      query="..." routequery="..." near="..." />
    <offline
      query="..."
      routequery="..."
      near="..." />
  </sql>
  <columns>
    <column minWidth="300" position="1">
      <row datamember="..." textsize="..." color="..." />
      <row datamember="..." textsize="..." color=".." />
      <row datamember="..." color="..." />
      <row ..... />
    </column> <column .... </column>
  </columns>
  <filters>
    <filter caption="..." replacetoken="..." fieldname="..."
      includefieldname="..." fieldtype="..." SQL="select ... as displayvalue, ... as storevalue from .... where
        ...." Alphabetreplace="1"/>
    <filter ..... /> </filters>
</AndroidLayout>
```

TAG <sql>:

INNER TAGS: **online** for online queries **OPTIONAL**
and **offline** for offline queries. **REQUIRED**

- **query:** the select sql query to execute. **REQUIRED**
- **routequery:** the select sql query, if the user has selected to filter the customer list based to route (checkbox in the list and date selection). Leave blank " " if route option is not required. For the date that the user has selected, use the \$DATE\$. **REQUIRED**
- **near:** the select sql query we want to be replaced in the parameter \$NEAR\$ of the above sql. Leave blank " " to disable it, if it is not a desirable option. **REQUIRED**
- **maxrecords:** determines the maximum number of rows that will appear on the grid. (default value is 300 rows). If dataset rows > maxrecords then the grid will be populated only with the first rows and a relative message will inform user that he has to input some criteria.

Note: In any case the required fields are: **cusid, leename, addriddest**
+ any other required field from the modules (field service)

TAG <columns>:

INNER TAGS: Every column shows field values from a record in the result dataset. **REQUIRED at least 1.**

<column>:

- **minWidth:** like "250", sets the minimum width of the column. **REQUIRED**
- **position:** sets the position of the column like "1". **REQUIRED**

TAG <rows>:

INNER TAGS: Every row displays a line included as a record in the result dataset. **REQUIRED at least 2.**

<row>:

- **datamember:** displays the field value from the select query. **REQUIRED**
- **textsize:** the size of the font. **OPTIONAL**
- **color:** the color of the row. **OPTIONAL**, useful website: <http://cloford.com/resources/colours/500col.htm>
- **useInSearch='true':** If you put in each row the useInSearch = 'true', you will be able to do a search in this field through the General search (regardless of filters). If you don't put in any row the useInSearch = 'true', the General search, will be looking only at leename. (the General search looking like 'contains in each field').
- **badge="badge1":** For each customer, there is a similar color badge so customer list and map. The Badge color is apparent from the company that the customer works. See next chapter about Badges.

NOTE: the first row has a bold font. From the 3rd line and after we could change the color.

TAG <filters>:

INNER TAGS: Every *filter* shows an additional filter for searching replaced in the select. **OPTIONAL**

<row>:

- **caption:** the name that will appear as a description of the filter. **REQUIRED**
- **replacetoken:** The token that will be replaced in sql \$... \$ from what you will write the user. **REQUIRED**
- **includefieldname:** values - "0" or "1". **REQUIRED**
 - If "1" then the criterion is manufactured using this fieldname as follows: {fieldname} = {user value}. If "1" and the user leave the criterion blank then the token is replaced with **1 = 1, otherwise with {fieldname} = {user value}.**
 - If "0" then the criterion be replaced directly with **the value put the user**, without having to add field name. If the user leaves it blank then the token is replaced with **blank.**
 - **ATTENTION:** SQL filter with parameter includefieldname = 0 not supported.
- **fieldname:** the name of the field which will replace in sql. **REQUIRED (if includefieldname=1)**
- **fieldtype:** values INT, NVARCHAR, FLOAT, SQL (used to populate options list), SEARCHABLE_LIST_TYPE, MULTI_SELECT_LIST_TYPE, BOLLEAN (In case True the replaced values is(1=1), in case False(1=0)). The input that will be available to the user. **REQUIRED**
- **SQL:** The sql query to be executed to bring all available options for the user. **REQUIRED (if fieldtype=SQL)**
 - displayvalue:** the value that will be visible to the user when selecting **REQUIRED**
 - storevalue:** the value to be replaced in the select **REQUIRED**
- **alphabetreplace:** 0/1 replacement characters for greekish and case insensitive searches. See. Greek support in search

NEW FILTERS TYPE ESSENTIAL NOTES

For filters type SEARCHABLE_LIST_TYPE and MULTI_SELECT_LIST_TYPE when the

includefieldname = 1 then the TOKEN will replace with values with id. That values comes from SQL (display values, store values).

Example:

- `<filter caption="Δ.Ο.Υ" replacetoken="$TXOID$" fieldname="lee.txoid" includefieldname="1" fieldtype="MULTI_SELECT_LIST_TYPE" SQL="select txoname as displayvalue,txoid as storevalue from txo order by txoname" keepvalue="1"/>`

The replaced token will become lee.txoid IN (151142, 454752, 5475,...).

- In case the filter have empty value then the token will become (1 = 1)

When the includefieldname = 0 then the TOKEN will be replace with strings .That values comes from SQL (display values, store values).

Example:

- `<filter caption="Πόλη" replacetoken="$CITY$" fieldname="adr.adrcity" includefieldname="0" fieldtype="MULTI_SELECT_LIST_TYPE" SQL="select distinct adrcity as displayvalue, adrcity as storevalue from ADR order by adrcity" keepvalue="1"/>`

The replaced token will become with strings IN ('ΘΕΣΣΑΛΟΝΙΚΗ', 'ΠΑΤΡΑ', 'ΑΙΓΙΟ',,...).

- In case the filter have empty value then the token will become LIKE '%'

The same functionalities is for the filter SEARCHABLE_LIST_TYPE.

ATTENTION:

1) In case we want to replace SQL filter with one of the above filters need to be careful the syntax in offline query with the replaced token.

Example: `<offlinequery="SELECT...FROM...WHERE like $TXOID$ and ifnull(adr.adrcity,") like '$CITY$%' ... "`

Then the replaced token with above **filters will fail** to do the query because of the replacement.

The query will become

```
SELECT...FROM...WHERE like IN(245444,45774,...) and ifnull(adr.adrcity,") like 'LIKE IN('ΘΕΣΣΑΛΟΝΙΚΗ', 'ΠΑΤΡΑ', 'ΑΙΓΙΟ',...)'.
```

Need to reconstruct the offlinequery to become like that

```
<offlinequery=SELECT...FROM...WHERE $TXOID$ and ifnull(adr.adrcity,") $CITY$...
```

Now the query will become

```
SELECT...FROM...WHERE IN(245444,45774,...) and ifnull(adr.adrcity,") LIKE IN('ΘΕΣΣΑΛΟΝΙΚΗ', 'ΠΑΤΡΑ', 'ΑΙΓΙΟ',...).
```

2) Becerful the displays values not include comma sperated values. Example of display values «A, B, Γ ΑΘΗΝΩΝ», this value not allowed and must be convert in other format like «A-B-Γ ΑΘΗΝΩΝ».

TAG **<map>**:

- **<map badge='badge1' subtitle='revenue'/>**
Specifies the color of the dot will appear on the map in the client coordinates. The subtitle defines what will be displayed if the user clicks on it.

1.3.1 XML example 1 (**queries, near**)

<AndroidLayout>

<sql>

```
<online query="exec mobGetCustomers $SLMID$,null, $LAT$, $LONG$"
  routequery="exec mobGetCustomers $SLMID$, $DATE$, $LAT$, $LONG$"
  near="(CentroidX between abs( $LAT$ - (10 * 0.009)) and abs( $LAT$ + (10 * 0.009)))" />
<offline query="SELECT cusid,leename,adriddest, adrstreet as destadrstreet,adnumber as
destadnumber,adrcity as destadrcity, IFNULL(adrstreet,") || ' ' ||
IFNULL(adnumber,") || ' ' || IFNULL(adrzipcode,") || ' ' || IFNULL(adrdistrict,") AS
destaddress,adrzipcode as destadrzipcode,adrdistrict as destadrdistrict,centroidx as
coordsx, centroidy as coordsy, case when exists (select visitid from tblVisits where
cusid=CUS.cusid and date(starttime)=$TODAY$) then 1 else 0 end as visited from CUS
```

```

INNER JOIN LEE ON CUS.leeid=LEE.leeid INNER JOIN ADR ON ADR.adrid=CUS.adriddest
where (cus.slmid=$SLMID$ or cus.slmid2=$SLMID$ or cus.slmid3=$SLMID$) and
$NEAR$"
routequery="SELECT cusid,leename,adriddest,adrstreet as destadrstreet,adnumber as
destadnumber,adrcity as destadrcity, IFNULL(adrstreet,") || ' ' ||
IFNULL(adnumber,") || ' ' || IFNULL(adrzipcode,") || ' ' || IFNULL(adrdistrict,") AS
destaddress, adrzipcode as destadrzipcode,adrdistrict as destadrdistrict,centroidx as
coordsx, centroidy as coordsy, case when exists (select visitid from tblVisits where
cusid=CUS.cusid and date(starttime)=$TODAY$) then 1 else 0 end as visited from CUS
INNER JOIN LEE ON CUS.leeid=LEE.leeid INNER JOIN ADR ON ADR.adrid=CUS.adriddest
where (cus.slmid=$SLMID$ or cus.slmid2=$SLMID$ or cus.slmid3=$SLMID$) and
(CUS.cusid=$CUSID$ or $CUSID$=0) and $NEAR$"
near="( CentroidX between abs( $LAT$ - (10 * 0.009)) and abs( $LAT$ + (10 * 0.009))) and
(CentroidY between abs( $LONG$ - (10* 0.011)) and abs( $LONG$ + (10 * 0.011)))\" />
</sql>
<rows>
  <row datamember='leename' textsize='25' color='#FFFF00'/>
  <row datamember='destaddress' textsize='20' color='#FFFFFF'/>
</rows> </AndroidLayout>

```

1.3.2 XML example 2 (show datamembers)

```
<AndroidLayout>
<sql><online query="...." routequery="...." near="" />
<offline query="SELECT cus.cusid, leename, adriddest, adrstreet as destadrstreet, adrnumber as destadrnumber, adrcity as
destadrcity, IFNULL(adrstreet, '') AS destaddress, 'Cus. Code:' || ifnull(cus.cuscode, '') as extrainfo, adrzipcode as
destadrzipcode, 'Tel: ' || ifnull(adr.adrphone1, '') as adrphone1, adrdistrict as destadrdistrict, centroidx as coordsx, centroidy as
coordsy, cus.udfint1 as badge, case when exists (select visitid from tblVisits where cusid=CUS.cusid and
date(starttime)=$TODAY$) then 1 else 0 end as visited from CUS INNER JOIN LEE ON CUS.leeid=LEE.leeid INNER JOIN
ADR ON ADR.adrid=CUS.adriddest where cus.cusid not in (-1,-2) order by leename "
routequery="... inner join tblRouteScheduledLines on tblRouteScheduledLines.cusid=cus.cusid and
tblRouteScheduledLines.scheduledDate=$DATE$ where cus.cusid not in (-1,-2) order by leename" near="" /></sql>
<columns> <column minWidth="1" position="1">
<row datamember='leename' textsize=" color='#0a4879' useInSearch='true'/>
<row datamember='extrainfo' textsize='14' color='#000000' useInSearch='true'/>
</column > <column minWidth="1" position="2">
<row datamember='destaddress' textsize=" color='#1c75bc' useInSearch='true'/>
<row datamember='adrphone1' textsize='14' color='#000000' useInSearch='true'/>
</column > <column minWidth="1" position="3"> <row datamember='destadrcity' textsize=" color='#1c75bc'
useInSearch='true' badge="badge"/> </column > </columns>
<map badge='badge' subtitle='tziros'/> </AndroidLayout>
```

```
<columns> <column minWidth="1" position="1">

<row datamember='leename' textsize=" color='#0a4879'
useInSearch='true'/>

<row datamember='extrainfo' textsize='14' color='#000000'
useInSearch='true'/>

</column > <column minWidth="1" position="2">

<row datamember='destaddress' textsize=" color='#1c75bc'
useInSearch='true'/>

<row datamember='adrphone1' textsize='14' color='#000000'
useInSearch='true'/>

</column > <column minWidth="1" position="3"> <row
datamember='destadrcity' textsize=" color='#1c75bc' useInSearch='true'
badge="badge"/> </column > </columns>

<map badge='badge' subtitle='tziros'/>

</AndroidLayout>
```

Customers

LIDL ΕΛΛΑΣ & ΣΙΑ Ο.Ε. Cus. Code:0010261270	ΒΙ.ΠΕ. ΠΑΤΡΑΣ Tel:	ΠΑΤΡΑ
SIEBEN T.C SHIP TO Cus. Code:0010310688	ΑΡΓΟΥΣ 44 Tel: 2751022533	ΝΑΥΠΛΙΟ
SIEBEN T.C SOLD TO Cus. Code:0010310684	ΔΗΜΟΚΡΑΤΙΑΣ 22 Tel: 2108196267	ΜΑΡΟΥΣΙ
SIEBEN T.C SOLD TO Cus. Code:0010310685	ΣΥΓΓΡΟΥ 22 Tel:	ΠΑΙΑΝΙΑ
SIEBEN T.C SOLD TO Cus. Code:0010310686	Α.ΣΥΓΓΡΟΥ Tel: 2108196227	ΠΑΙΑΝΙΑ
SIEBEN T.C SOLD TO Cus. Code:0010310692	ΔΕΚΕΛΕΙΑΣ 34 Tel: 2232063636	ΛΑΡΙΣΑ

1.3.3 XML example 3 (badge stored in the customer row)

```
<AndroidLayout> <sql> <online query="..." routequery="..." near="" />
    <offline query="SELECT cus.cusid, leename, adriddest, adrstreet as destadrstreet, adrnumber as
destadrnumber, adrcity as destadrcity, IFNULL(adrstreet, '') AS destaddress, 'Cus. Code:' || ifnull(cus.cuscode, '') as extrainfo,
adrzipcode as destadrzipcode, 'Tel: ' || ifnull(adr.adrphone1, '') as adrphone1, adrdistrict as destadrdistrict, centroidx as coordsx,
centroidy as coordsy, cus.badge as badge, case when exists (select visitid from tblVisits where cusid=CUS.cusid and
date(starttime)=$TODAY$) then 1 else 0 end as visited from CUS INNER JOIN LEE ON CUS.leeid=LEE.leeid INNER JOIN
ADR ON ADR.adrid=CUS.adriddest where cus.cusid not in (-1,-2) order by leename "
routequery="... inner join tblRouteScheduledLines on tblRouteScheduledLines.cusid=cus.cusid and
tblRouteScheduledLines.scheduledDate=$DATE$ where cus.cusid not in (-1,-2) order by leename" near="" /> </sql>
    <columns>
        <column minWidth="1" position="1">
            <row datamember='leename' textsize="" color='#0a4879' useInSearch='true'/>
            <row datamember='extrainfo' textsize='14' color='#000000' useInSearch='true'/>
        </column >
        <column minWidth="1" position="2">
            <row datamember='destaddress' textsize="" color='#1c75bc' useInSearch='true'/>
            <row datamember='adrphone1' textsize='14' color='#000000' useInSearch='true'/>
        </column >
        <column minWidth="1" position="3">
            <row datamember='destadrcity' textsize="" color='#1c75bc' useInSearch='true' badge="badge"/>
        </column > </columns>
    <map badge='badge' subtitle='tziros'/>
</AndroidLayout>
```

The Cus table has already a field named badge for each customer. Another scenario is a custom field like **cus.udfint1 as badge**

Eg. ...CASE WHEN ifnull(udfint3, 1) =1 THEN 7 WHEN ifnull(udfint3, 1) =3 then 18 else 25 END AS badge

1.3.4 XML example 4

```
<AndroidLayout> <sql>
    <online query="exec mobGetCustomers $SLMID$,null, $LAT$, $LONG$" routequery="exec
mobGetCustomers $SLMID$, $DATE$, $LAT$, $LONG$"      near="(CentroidX between abs( ($LAT$) - (0.25 * 0.009)) and
abs( ($LAT$) + (0.25 * 0.009))) and (CentroidY between abs( ($LONG$) - (0.25 * 0.011))      and abs( ($LONG$) + (0.25 *
0.011))) and (CUS.cusid=$CUSID$ or $CUSID$=0)" />
    <offline query="SELECT  cus.cusid,ifnull(cus.cusname,leename) as
leename,cusname,adriddst,ifnull(adr.CentroidX,") as Coordsx,ifnull(adr.CentroidY,") as CoordsY,Πόλη: ' || IFNULL(adrcity,")
as destadrcity, IFNULL(adrstreet,") AS destaddress,IFNULL(adrdistrict,")as adrdistrict,Κωδ. Πελάτη: ' || ifnull(cus.cuscode,")
as extrainfo,'Α.Φ.Μ: '||ifnull(lee.leeafm,")as leeafm, 'Τηλέφωνο: '|| ifnull(adr.adrphone1,") as adrphone1,'Τ.Κ.: ' ||
ifnull(adrzipcode,") as adrzipcode,ifnull(adr.adremail,") as email, case when exists (select visitid from tblVisits where
cusid=CUS.cusid and date(starttime)=TODAY$) then 1 else 0 end as visited,SLMtasklistid as cusslmtasklistid,cus.badge as
badge1,'Τζίρος: '|| ifnull(cus.tziros,0) as tziros,Companies.CompanyDescr as Company from CUS INNER JOIN LEE ON
CUS.leeid=LEE.leeid INNER JOIN ADR ON ADR.adrid=CUS.adriddest LEFT OUTER JOIN CustomKriKriCompanies
Companies ON Companies.KrikriCompanyID=CUS.CompanyID where cus.cusid not in (1,2,3,4,-1,-2) and $RouteID$ and
$CUSCategoryID$ and ( ifnull(cus.tziros,0)>= cast(ifnull(nullif('$tzirosMegaluteros$',"),-100000)as int) and
cast(ifnull(nullif('$tzirosMikroteros$',"),100000000)as int )>=ifnull(cus.tziros,0) )  order by leename,cuscode " routequery=""
near="" /></sql>
    <columns>
        <column minWidth="250" position="1">
            <row datamember='leename' textsize='18' color='#0a4879' useInSearch='true' />
            <row datamember='extrainfo' textsize='16' color='#000000' useInSearch='true' />
            <row datamember='leeafm' textsize='14' color='#000000' useInSearch='true' />
        </column >
        <column minWidth="250" position="2">
            <row datamember='destaddress' textsize='16' color='#1c75bc' useInSearch='true' />
            <row datamember='destadrcity' textsize='16' color='#00297A' useInSearch='true' />
            <row datamember='adrphone1' textsize='14' color='#000000' useInSearch='true' />
        </column >
        <column minWidth="200" position="3">
            <row datamember='Company' textsize='16' color='#00297A' useInSearch='true'
badge="badge1"/>
            <row datamember='tziros' textsize='16' color='#00297A' useInSearch='true' />
        </column >
    </columns>
    <filters>
        <filter caption="Κατηγορία Πελάτη:" replacetoken="$CUSCategoryID$" fieldname="cus.Category"
includefieldname="1" fieldtype="SQL" SQL="select Description as displayvalue,CUSCategoryID as storevalue from
CustomkriKriCusCategory" keepvalue="1"/>
        <filter caption="Δρομολόγιο:" replacetoken="$RouteID$" fieldname="cus.RouteID"
includefieldname="1" fieldtype="SQL" SQL="select RouteDescr as displayvalue,KriKriRouteID as storevalue from
CustomKriKriRoutes" keepvalue="1" />
        <filter caption="Τζίρος >:" replacetoken="$tzirosMegaluteros$" fieldname="cus.RouteID"
includefieldname="0" fieldtype="FLOAT" SQL="" keepvalue="1" />
        <filter caption="Τζίρος <:" replacetoken="$tzirosMikroteros$" fieldname="cus.RouteID"
includefieldname="0" fieldtype="FLOAT" SQL="" keepvalue="1" />
    </filters>
    <map badge='badge1' subtitle='tziros' />
</AndroidLayout>
```

Another example of filters:

```
<filters>
    <filter  caption="Επών-Διευθ-Περιοχ-ΑΦΜ:" replacetoken="$MultiFilter$"  fieldname=""  includefieldname="0"
fieldtype="NVARCHAR" SQL="" alphabetreplace="1"/>
</filters>
```

1.4 Badges (on Tasklist, Customerlist etc)

Show custom coloured badge (colored dot) in the client task, or Customerlist, or map etc.

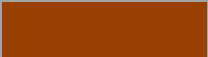
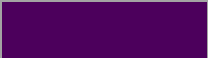
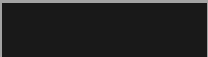
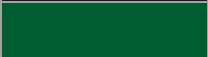
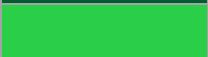


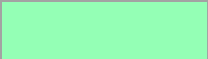
Tasklist:

You must bring in the question of `sqlOffGetCustomerTasks` a parameter column named **Badge with values from 1 to 26**. These values correspond to the colors in the table below.

Customerlist:

just populate the `xml tag badge` with appropriate badge color like `badge="badge1"`

Badge	Hex code	Sample	Name
1	#F0A3FF		Amethyst
2	#0075DC		Blue
3	#993F00		Caramel
4	#4C005C		Damson
5	#191919		Ebony
6	#005C31		Forest
7	#2BCE48		Green

8	#FFCC99		Honeydew
9	#808080		Iron
10	#94FFB5		Jade
11	#8F7C00		Khaki
12	#9DCC00		Lime
13	#C20088		Mallow
14	#003380		Navy
15	#FFA405		Orpiment
16	#FFA8BB		Pink
17	#426600		Quagmire
18	#FF0010		Red
19	#5EF1F2		Sky
20	#00998F		Turquoise
21	#E0FF66		Uranium
22	#740AFF		Violet
23	#990000		Wine
24	#FFFF80		Xanthin
25	#FFFF00		Yellow
26	#FF5005		Zinnia

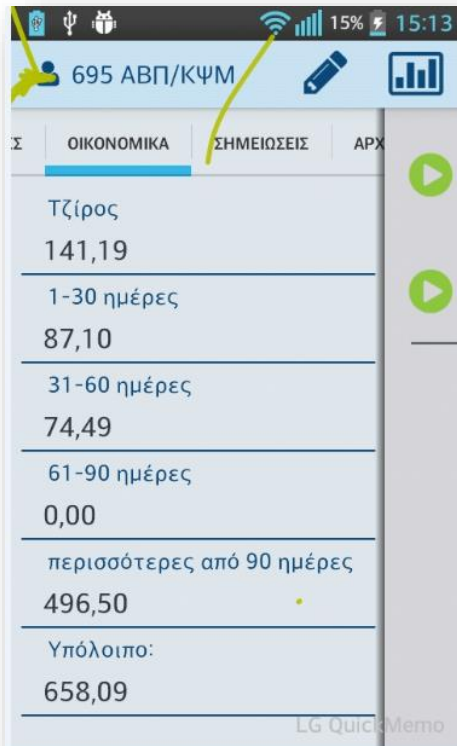
The color values with their description are also in the **[tblBadgeColor]** table.

	ColorID	Description
1	0	No color
2	1	Amethyst
3	2	Blue
4	3	Caramel
5	4	Damson
6	5	Ebony
7	6	Forest

Other sources :

http://en.wikipedia.org/wiki/Help:Distinguishable_colors

1.5 Customer Card (extra financial data)



ΠΕΡΙΟΔΟΣ	ΠΟΣΟΣΤΟ
Τζίρος	141,19
1-30 ημέρες	87,10
31-60 ημέρες	74,49
61-90 ημέρες	0,00
περισσότερες από 90 ημέρες	496,50
Υπόλοιπο:	658,09

Refers to the ability to display **extra financial data** (Days Sales Outstanding - DSO, Open value Balance) of the selected customer in the financial tab.

The data comes from the table **tblKarteles**, (per branch CUSid - not LEEid)

To activate it, put into the configuration, the parameter **Kartelles with 1 as configvalue**.

Parameter **CUSBalanceXML**: the xml that overrides the default, which

is responsible for the design of the financial data grid  There is no option to change the select sql query, only the customization of the columns derived from the tblkarteles. By default the app adds a column named "Running Balance" which is the growing balance of each row. It can be viewed at any location within the grid by putting as datamember="runningbalance". Example:

```
<AndroidLayout><columns>
  <column width="120" position="1" caption="Date"
    datamember="TRNDate"/>
  <column width="120" position="2" caption="Document"
    datamember="TRNReason"/>
  <column width="100" position="3" caption="Charge" datamember="Charge"/>
  <column width="100" position="4" caption="Credit" datamember="Credit"/>
  <column width="120" position="5" caption="Document No"
    datamember="DosCode"/>
  <column width="120" position="6" caption="Due Date"
    datamember="udfdate1"/>
  <column width="120" position="7" caption="Weighted Avg"
    datamember="udfdate2"/>
  <column position="8" caption="Amount" datamember="udffloat1"/>
</columns> </AndroidLayout>
```

Old not supported: AGINGALGORITHM,ENILPERIOD1..3

DETAILS	FINANCIAL	NOTES
VAT no	081749529	
Last Visit	2017-02-16	
Turnover	1738.7	
Credit Limit	9000	
Balance	100	

By using the **SQLFINANCIAL** parameter (Configuration table) we can check which fields will appear in the financial tab. For example we can write an SQL as extraconfigvalue:

```
SELECT 'VAT no' AS title, leeafm AS displayvalue , 1 AS position FROM cus INNER
JOIN lee ON cus.leeid = lee.leeid WHERE cusid = $CUSID$ UNION ALL SELECT
'Last Visit' AS title ,date(max(visittime)) AS displayvalue ,2 AS position FROM
tblVisits WHERE cusid= $CUSID$ UNION ALL SELECT 'Turnover' AS title
,cus.tziros AS displayvalue ,10 AS position FROM CUS WHERE cusid =
$CUSID$ UNION ALL SELECT 'Credit Limit' AS title ,CUS.cuscreditlimit AS
displayvalue ,20 AS position FROM CUS WHERE cusid = $CUSID$ UNION ALL
SELECT 'Balance' AS title ,CUS.balance AS displayvalue ,30 AS
position FROM CUS WHERE cusid = $CUSID$
```

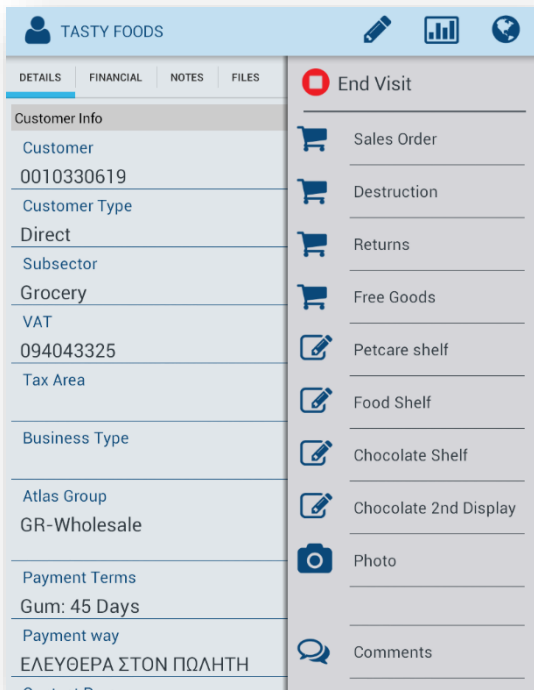
(optional): 0 as after (default) our custom fields will be displayed before the default financial fields (such as DSO)

or 1 **as after** : will be displayed after the default financial fields

We can have any combination of the parameters KARTELLS and SQLFINANCIAL. If SQLFINANCIAL parameter has no value, it displays by default:

- If we have not any aging balances, Turnover, Credit Limit and Balance fields (from CUS)
- If we aging balances, except Tab Fields (DSO) appear and the Turnover and Credit limit fields. The balance of the CUS does not appear, because the balance of the tab appears

1.6 sqlOffGetCustomersTasks (configuration)



This parameter defines the tasks that will be in the visit of the seller for the selected customer.

We can define additional column named **Badge (with values from 1 to 26)** for showing specific colors. Attention: not NULL value in the field Badge

Relative Tables:

tblSLMTaskLists, tblSLMTasks

Example:

```
SELECT SLMTaskID,TaskDescription,Action,MerchScenarioID,MerchCategoryID
      ,MerchCompanyID,SurveyID,Pos,ParentSLMTaskID,Mandatory
      ,sdtid,Badge,ActionParameter1
FROM tblSLMTasks
WHERE SLMTaskListID = (
      SELECT IFNULL(CUS.SLMTaskListID,
      (SELECT configvalue FROM Configuration
      WHERE configcode = 'SLMTaskListID' LIMIT 1))
      FROM CUS WHERE CUS.cusid = $CUSID$)
ORDER BY Pos ASC
```

```
... UNION all SELECT 'ΣΥΝΤΑΓΕΣ ΤΕΧΝΙΚΟΫ','SURVEY',0,0,0,6,15,NULL,0,NULL,0 FROM SLM
WHERE SLMID IN (1266758136,1868789165)
```

1.7 Customer Actions in tasklist (tblSLMTasks)

TaskDescription	Action
Order	SalesPlus
Collection	Collection
Pricing Survey (new)	Merchandise
Survey (old)	Survey
Photo	PhotoPlus
Geocoding	Location_Map
Expenses	Benchmarks
Document view	DOCUMENTVIEW
Find location (stigma)	Location
Simple visit	Simplevisit
SIGNATURE	SIGNATURE
Comments of Visit	Comments
Visit Reason	REASON

Action column shows the reserved words (available action types)

1.8 sqlOffGetCustomerDetails (configuration)

The SQL to be executed to bring the customer-specific information details in offline mode. It brings the following fields:

field **title**: the title of every information

field **value**: the content of information

field **Group**: Used to group information. Contains the title of the group.

field **BusinessType**: Specifies the type of information. This field takes the following values:

- PHONENUMBER : indicates that this entry on the customer's details concerns phone number. If the seller presses the entry will be dialing in the specified phone number.

- EMAIL : indicates that the specific record in the customer details concerns an email address. If the seller presses this entry in the list, will open a pop up to choose the email dispatch application wants to use.

- ADDRESSNAVIGATE : The value specified concerns in address. If the seller presses this entry in the list, will open a pop up window to choose the application to view navigation details about this address. (Google maps fully supported).

- Blank (" as BusinessType) : It doesn't add any functionality to the entry in the list.

If the field BusinessType not set at all (i.e. there is no in select) then the program automatically tries to understand the type of each field. The address can either refers to the name of the road or to the navigation point coordinates.

In case of the coordinates, the field value of select must have the following format:

```
SELECT 'navigation point coordinates', ifnull(CentroidX,"") || "," || ifnull(CentroidY,"") as value , 'ADDRESSNAVIGATE' as BusinessType FROM ADR WHERE ADRID =$ADRIDDEST$
```

Mandatory is, the value to have coordinates separated by commas.

Example 1

```
select 'OUTLET' as title,ifnull(cusname,") as value," as BusinessType from tblServiceCalls inner join CUS on
tblServiceCalls.cusid=CUS.cusid where tblServiceCalls.SCID='$SCID$'

UNION ALL select 'CUSTOMER CODE',ifnull(cuscode,")," as BusinessType from tblServiceCalls inner join CUS on
tblServiceCalls.cusid=CUS.cusid where tblServiceCalls.SCID='$SCID$'

UNION ALL select 'ADDRESS', ifnull(adastreet,")||','||ifnull(adrdistrict,")||','||ifnull(adrcity,")||','||ifnull(adrcode,") as street,
'ADDRESSNAVIGATE' as BusinessType from tblServiceCalls inner join CUS on tblServiceCalls.cusid=CUS.cusid inner join
ADR on ADR.adrid=CUS.adriddest WHERE tblServiceCalls.SCID='$SCID$'

UNION ALL select 'Customer COORDINATES', ifnull(CentroidX,")||','||ifnull(CentroidY,") as value, 'ADDRESSNAVIGATE' as
BusinessType from tblServiceCalls inner join CUS on tblServiceCalls.cusid=CUS.cusid inner join ADR on
ADR.adrid=CUS.adriddest WHERE tblServiceCalls.SCID='$SCID$'

UNION ALL select 'Phone Number', ifnull(adrcode1,") as phone,'PHONENUMBER' as BusinessType from tblServiceCalls
inner join CUS on tblServiceCalls.cusid=CUS.cusid inner join ADR on ADR.adrid=CUS.adriddest WHERE
tblServiceCalls.SCID='$SCID$'
```

Example 2

```
SELECT 'Customer Code' AS title
, cuscode AS value
, " AS BusinessType
, 'Customer Information' AS [Group]

FROM cus WHERE cusid = $CUSID$

UNION ALL SELECT 'Customer Code - main' AS title, leeid AS value
FROM cus WHERE cusid = $CUSID$

UNION ALL SELECT 'Customer Code - Branch' AS title, companyid AS value
FROM cus WHERE cusid = $CUSID$

UNION ALL SELECT 'A.Φ.Μ.' AS title
, leeafm AS afm
, " AS BusinessType
, 'Customer Information' AS [Group]

FROM cus INNER JOIN lee ON cus.leeid = lee.leeid WHERE cusid = $CUSID$

UNION ALL SELECT 'TAX' AS title
, txo.txoname AS txo
, " AS BusinessType
, 'Customer Information' AS [Group]
```

```

FROM cus

INNER JOIN lee ON cus.leeid = lee.leeid

LEFT OUTER JOIN txo ON lee.txoID = txo.txoID

WHERE cus.cusid = $CUSID$

UNION ALL

SELECT 'Payment method' AS title

        ,cpmtitle

        ," AS BusinessType

        , 'Customer Information' AS [Group]

FROM cpm INNER JOIN cus ON cus.cpmid = cpm.cpmid WHERE

cusid = $CUSID$

UNION ALL

SELECT 'City' AS title

        ,adrcity AS value

        ," AS BusinessType

        , 'Address information' AS [Group]

FROM ADR WHERE ADRID = $ADRIDDEST$

UNION ALL

SELECT 'District' AS title

        ,adrdistrict AS district

        ," AS BusinessType

        , 'Address information' AS [Group]

FROM ADR WHERE ADRID = $ADRIDDEST$

UNION ALL

SELECT 'Address' AS title

        ,adrstreet || ' ' || ifnull(adrcity, "") AS street

        , 'ADDRESSNAVIGATE' AS BusinessType

        , 'Address information' AS [Group]

FROM ADR WHERE ADRID = $ADRIDDEST$

UNION ALL

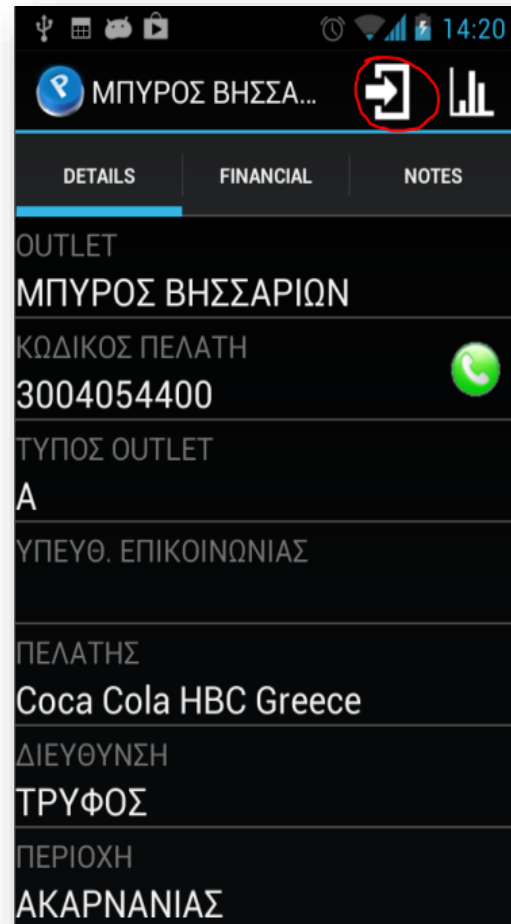
SELECT 'Postcode' AS title

        ,adrzipcode AS zip

        ," AS BusinessType

        , 'Address information' AS [Group]

```



```
FROM ADR WHERE ADRID = $ADRIDDEST$
```

```
UNION ALL
```

```
SELECT 'Phone' AS title
```

```
    ,adrphone1 AS phone
```

```
    , 'PHONENUMBER' AS BusinessType
```

```
    , 'Address information' AS [Group]
```

```
FROM ADR WHERE ADRID = $ADRIDDEST$
```

```
UNION ALL
```

```
SELECT 'Mobile'
```

```
    ,adrphone2 AS phone2
```

```
    , 'PHONENUMBER' AS BusinessType
```

```
    , 'Address information' AS [Group]
```

```
FROM ADR WHERE ADRID = $ADRIDDEST$
```

```
UNION ALL
```

```
SELECT 'FAX' AS title
```

```
    ,ADRFax AS fax
```

```
    , " AS BusinessType
```

```
    , 'Address information' AS [Group]
```

```
FROM ADR WHERE ADRID = $ADRIDDEST$
```

```
UNION ALL
```

```
SELECT 'E-MAIL' AS title
```

```
    ,ADREmail AS email
```

```
    , 'EMAIL' AS BusinessType
```

```
    , 'Address information' AS [Group]
```

```
FROM ADR WHERE ADRID = $ADRIDDEST$
```

```
UNION ALL
```

```
SELECT 'Remarks' AS title
```

```
    ,remarks AS remarks
```

```
    , " AS BusinessType
```

```
    , 'Additional' AS [Group]
```

```
FROM CUS WHERE cusid = $CUSID$
```

1.9 Localization

In case you have data that you want to appear differently in each language (e.g. product descriptions) we can use the function of localization. This function is based on two tables, **srLocalizeDescription** and **srLocalizedFields**.

In **srLocalizedFields**, add the fields for which we want to become the Localization, e.g. leename or stiname

	LFID	Tablename	Fieldname	Keycolumn
1	1	LEE	leename	leeid

The alternative translations come in to table **srLocalizeDescription**.

	ID	TableName	FieldName	ValueID	Description	LocaleID	UV
1	1	LEE	leename	4	Μικροσφτ	1032	0x0000000000010FAA
2	2	LEE	leename	4	MICROSOFT	1033	0x0000000000010DBD
3	3	LEE	leename	4	MICROSOFT other	1034	0x0000000000010DBE
4	4	LEE	leename	5	Unilever	1033	0x0000000000010DBF
5	5	LEE	leename	5	Ουνιλεβερ	1032	0x0000000000010DC3
6	6	LEE	leename	5	Unilever other	1034	0x0000000000010DC1

The ValueID in this case is the Leeid of the corresponding records. The LocaleID is the language that matches the specific translation. To know the system, the language in which will serve the data uses the field SLM.localeid. That is, each user can have their own language and when you synchronize the table LEE will get the values of the corresponding leename in their own language.

1.10 Greek language support in search

The General logic: in the criteria when the user fills in a field, then the program puts *stiname glob '[criterio]'* which allows us to import Greek letters when searching with no problem with capital-small...

The [criterio] replaced based on a **dictionary** as follows: If the user write the word «αρ» and wants to search the product “ΑΡΩΜΑ 100ML”, the where statement generated is the following: “stiname glob '[Ααά][Ρρ]’” and that successfully brings results as searching all possible combinations of letters that are in brackets.

How to customize the sets of letters: If it does not already exist, you must create an **entry in the configuration named AlphabetReplace** that will **contains the dictionary** mentioned above. The record must have this form, generally for all the alphabet you want to support:

Aαά,Bβ,Γγ,Δδ,Εεέ,Ζζ,Ηηή,Θθ,Ιιί,Κκ,Λλ,Μμ,Νν,Ξξ,Οοό,Ππ,Ρρ,Σσς,Ττ,Υυύ,Φφ,Χχ,Ψψ,Ωωώ,Aa,Bb,Cc,Dd,Ee,Ff,Gg,Hh,Ii,Jj,Kk,Ll,Mm,Nn,Oo,Pp,Qq,Rr,Ss,Tt,Uu,Vv,Ww,Xx,Yy,Zz

Next you need to supplement an extra attribute called alphabetreplace in the **tag <column .../>** in the field to search.

If you wish to play with the specific variable then you should set a value of 1, if you do not, then put the value 0 or do not include it at all, because the default value is 0. Else, the program does not take into consideration anything and playing with the old faulty logic.

What the program does: The time that dynamically adds the criterion, check the letters that has put the user and replace them properly. E.g. If the user has written “αρ” at the criterion and presses search then the program will take 1-1 the letters and will search if there are in the dictionary. If there is a match in the dictionary, then replaces the entire “suggestion” of dictionary that you specify in the letter. I.e. if the dictionary you are registering has the record: Ααά,Bβ,Γγ,... Then the «α» in «αρ» will be replaced with the following: [Ααά]. If there is not a letter then not replaced. I.e. if you forgot to put the «ρ» in the dictionary then the replacement in the word «αρ» will be as following: «[Αα]ρ».

NOTE: to run the searches as case and accent insensitive, it is necessary to exist in the dictionary for each letter, the capital, the small and the accented letters.

HINT: in this sense it is very feasible to create a dictionary that supports greeklish too. The dictionary in this case should be something like this (attention has English and Greek capital): ...,EeEeέ,...

1.11 HTML Markup

Html markup, actively supported (bold, italic, colors, etc), but there are some points that we should watch!

Firstly, the supported html markup are in the following link: <http://commonsware.com/blog/Android/2010/05/26/html-tags-supported-by-textview.html>

Exception: in the **font**, the only supported attribute is **color**, **not the SIZE**, **not the FACE**. The size can be changed by other properties of the XML.

USE: In the sql query where is embedded in a field, putting in front of it, the appropriate html markup.

In the columns->column->caption attribute, where it seems respectively the header of the grid, as the title of the criterion where it is used.

You need to do:

These special characters for html are also special characters for xml. If you go to put loose then the xml will not be able to parse and you will get no error message.

The steps to a proper xml with markup are the following:

1. Add the markup and we confirm that is correct and what we want.
2. Replace in markup these 5 special characters with their respective:

" -> "

' -> '

< -> <

> -> >

& -> &

3. We verify that our xml is correct with a corresponding program (like xmlpad) or simply execute, where will open your browser and will tell us if there is a mistake or not (to save time).
4. Then we try it on the pda.

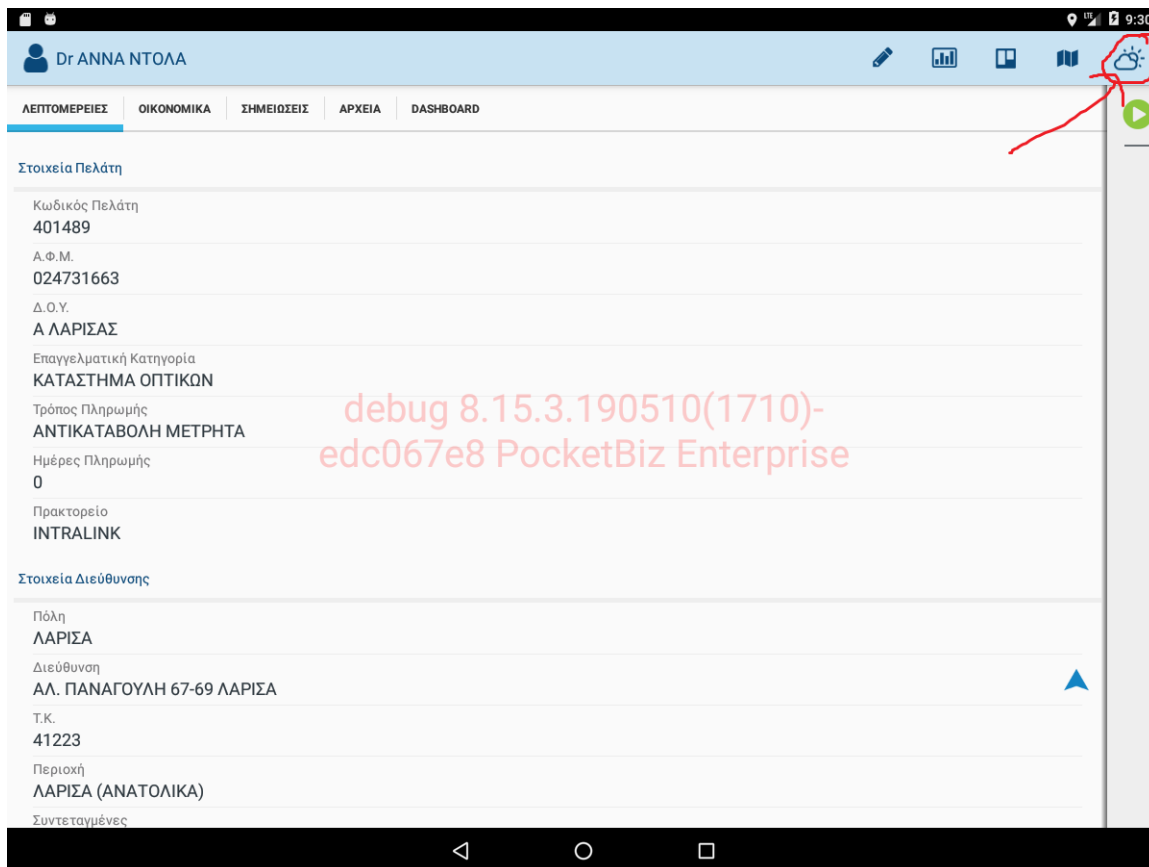
1.12 Five Days Weather Forecast

Five days weather forecast added to the customer panel. For enable the weather forecast feature we need to insert in the table Configuration in the column configcode the value(WeatherApiKey) and in the column configvalue(The key that the customer have buy)

-Key only for test 8acecfddfeb6aad6a02af4cab9b64b6 (After 40 calls the key doesn't work)

We can buy the key from <https://openweathermap.org/price>

When the weather forecast is enabled the customer panel show that option



Also in the offline query parameter need that ifnull(adr.CentroidX, "") AS Coordsx, ifnull(adr.CentroidY, "") AS CoordsY **(Be carefull need to write as it is)**. Because it needs the coordinates of the customer.

1.13 Open URL to an external browser

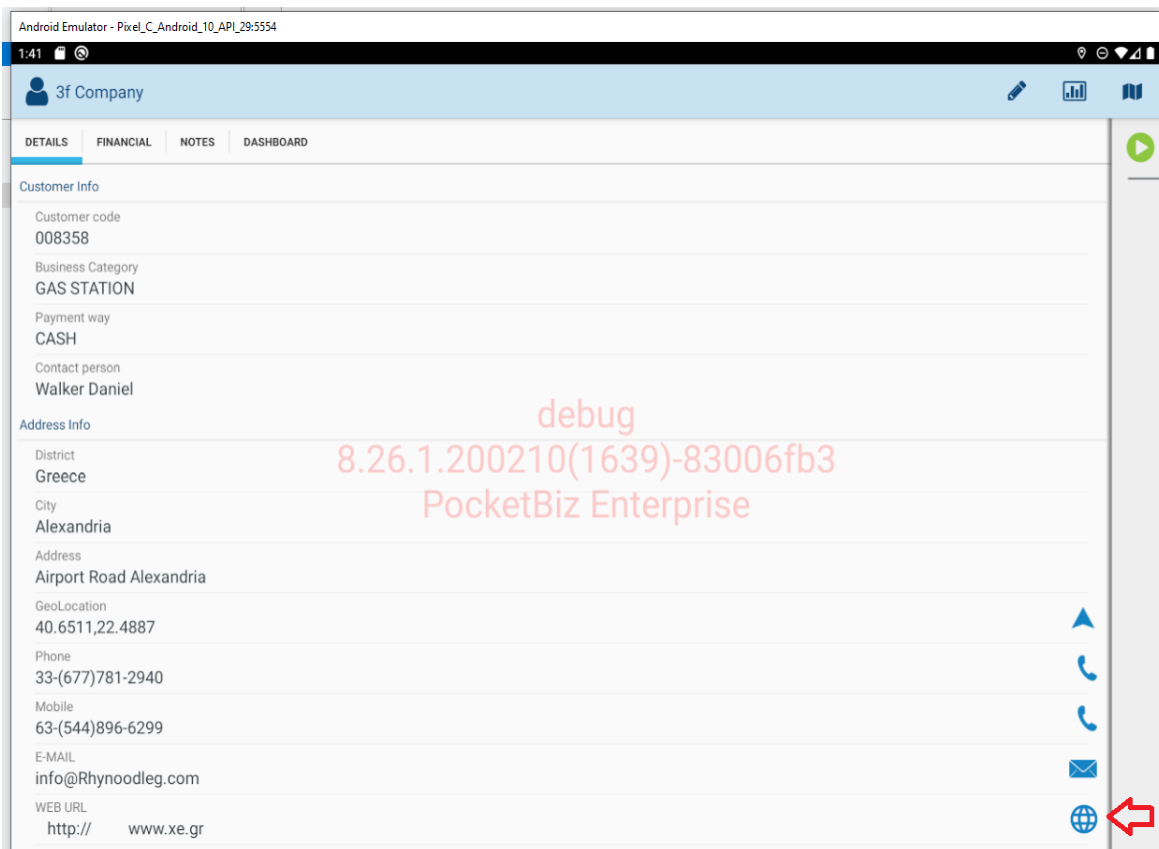
We have the ability to open url to an external browser from the customer details panel.

To do that need in **ADR** table exist the column **adrurl**. The value of the column must be a valid url. Valid url is consider the url which have **prefix https://**

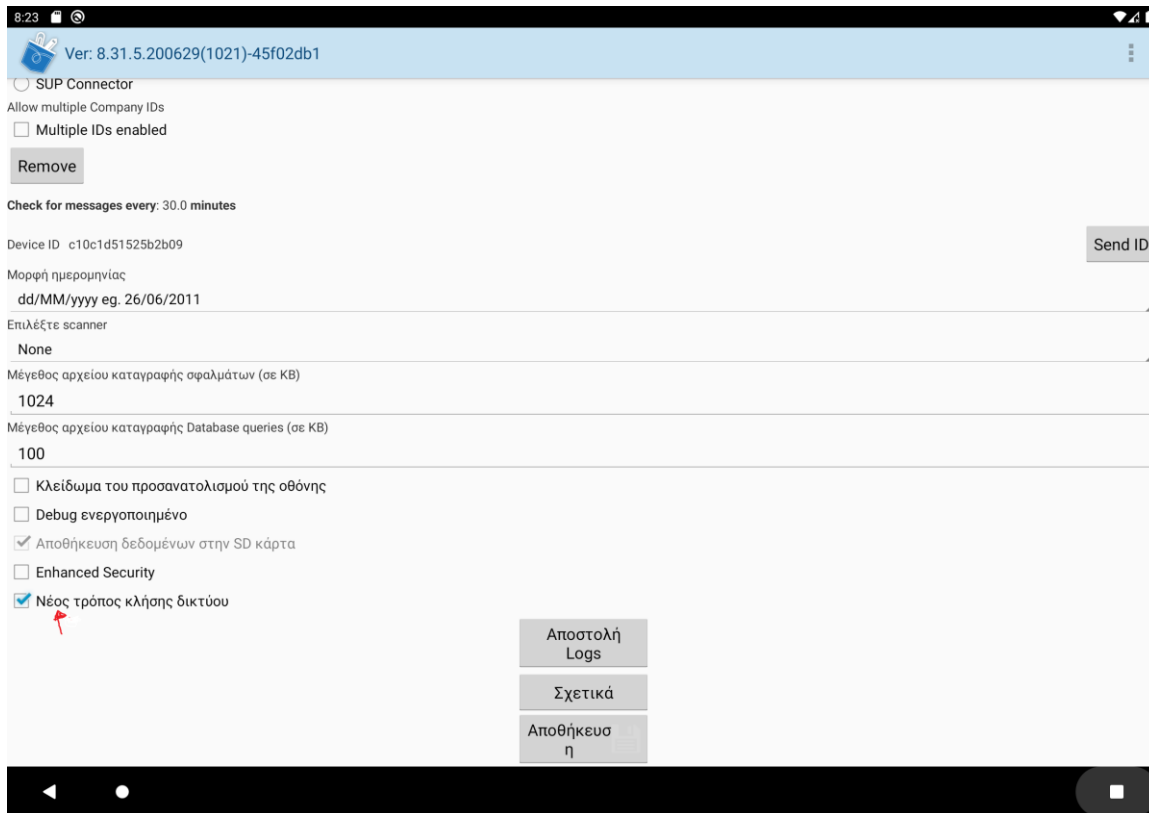
Also we need to add in the sql query the **WEB as businessType** see the below example

Example:

```
SELECT 'Website' AS title
      ,adrurl AS value
      ,'WEB' AS BusinessType
      ,'Στοιχεία Διεύθυνσης' AS [Group]
FROM ADR
WHERE ADRID = $ADRIDDEST$
```



1.14 New method network call



8:23 Ver: 8.31.5.200629(1021)-45f02db1

☐ SUP Connector

Allow multiple Company IDs

☐ Multiple IDs enabled

Remove

Check for messages every: 30.0 minutes

Device ID c10c1d51525b2b09 Send ID

Μορφή ημερομηνίας
dd/MM/yyyy eg. 26/06/2011

Επιλέξτε scanner
None

Μέγεθος αρχείου καταγραφής ασφαμάτων (σε KB)
1024

Μέγεθος αρχείου καταγραφής Database queries (σε KB)
100

☐ Κλείδωμα του προσανατολισμού της οθόνης

☐ Debug ενεργοποιημένο

☒ Αποθήκευση δεδομένων στην SD κάρτα

☐ Enhanced Security

☒ Νέος τρόπος κλήσης δικτύου

Αποστολή Logs

Σχετικά

Αποθήκευση

From the settings of the pocketbiz we can enable the new method network call as show in the image above. If this options is checked this mean the new method is enabled.

The default value of this option is the standard network call.