Oracle® Retail MICROS Stores2
Functional Document
Web Service Centralized Customer Tracking
Release 1.20-1.39

July 2017

ORACLE®

Oracle® Retail MICROS Stores2 Functional Document Web Service Centralized Customer Tracking, Release 1.20-1.39

TECHNICAL DOCUMENTATION

Stores2 Web Service - Centralized customer interchange tracks

Report developed by:
MICROS-Retail srl with sole shareholder

Alessio Ossola

Version 1.00 of 16/07/2014

Table of contents

Note: The rebranding for the latest version of this documentation set is in development as part of post MICROS acquisition activities. References to former MICROS product names may exist throughout this existing documentation set.

1 Introduction

This document contains the track used for calling the web service of online customers.

1.1 FidelityGenericGetCustomerList

1.1.1 Description

Used to search for customers who meet certain search criteria.

1.1.2 Mandatory parameters

The mandatory parameters must always be included in the call to the page.

az	Company code
nd	Node code
pf	Workstation code

1.1.3 Optional parameters

The optional parameters determine the data set returned by the function and the format in which this information is returned.

node	Customer node
code	Customer code
bc	Customer barcode
nam	Customer name
sur	Customer last name
Email	Customer email
locnam	Name of customer location
city	Customer city
vat	Customer VAT number
fidelity	Fidelity flag
maxcust	Maximum number of customers to return
responsetype	Type of function response. *If this parameter is equal to XML the recordsets returned by the call to the page are not serialized and will be available in XML text format. With any other value (or not specifying the parameter) the data are serialized.*
nozip	Flag for zipping the result. *If this parameter is set to 1 the data returned by the function are not zipped and will be shown unzipped in the result. With any other value (or not specifying the parameter) the data are zipped.*

1.1.4 Example

http://localhost/Stores2WS/FidelityGenericGetCustomerList.aspx?az=14&pf=1&nd=001&nam=MARIO&sur=ROSSI&responsetype=XML&nozip=1

With this call the web service returns the data unzipped and not serialized for all customers with first name Mario and last name Rossi.

1.1.5 Return value

The web service returns an XML document containing the search result (or a code relative to an error which occurred during processing, 0 if there were no errors), the number of potentially located customers and the data of the first *n* customers found (where *n* is a number specified during the call phase).

```
<data error="0" totalcustomers="n">
   <customers>
      <![CDATA[...]]>
   </customers>
   <addresses>
      <![CDATA[...]]>
   </addresses>
</data>
```

In the CDATA attribute it is possible to find (tag <customers) customer data (including information taken from the ProdottiBarcodes table) and shipping address data (tag <addresses>), with or without the schema, based on the *responsetype* parameter specified in the call.

1.2   FidelityGenericSubmitCustomerData

1.2.1 Description

This is used to add or update the data of one or more customers and add or update one or more shipping addresses.

The function autonomously determines the type of operation to perform (add or update) based on whether the data is already present in the database.

1.2.2 Mandatory parameters

The mandatory parameters must always be included in the call to the page.

az	Company code
nd	Node code
pf	Workstation code

1.2.3 Optional parameters

ADODB recordset containing customer data (and possibly the barcode).

ADODB recordset containing the shipping address data.

1.2.4 Example

http://localhost/Stores2WS/FidelityGenericSubmitCustomerData.aspx?az=14&pf=1&nd=001&responsetype=XML&nozip=1

With this call the web service returns the result of the operation on customer data in unzipped format.

Instead of recordsets it is possible to enter parameters, this procedure narrows the amount of managed information:

chkemail	can be set to 1 or 0 based on whether unequivocalness of the email needs to be checked.
nozip	always to 1.
bc	Customer barcode -> 12 characters: 3 prefix (202) + 3 customer node + 6 customer code; max len 50.
node	customer node code; Stores2 field: Nodo; max len 3.
code	customer code; sequential number of 6 characters (e.g. 000001). The code needs to be unequivocal at node level; Stores2 field: codice; max len 10.
name	Name; Stores2 field: RagioneSociale; max len 255.
surname	Last name; Stores2 field: Cognome; max len 50.
address	Address; Stores2 field: Indirizzo; max len 50.

cell	Cell phone; Stores2 field: Telefono2; max len 50.
location	location; Stores2 field: Localita; max len 50.
province	province; if management is <mark>configured by nation the</mark> Stores2 code needs to be used for the provinces, otherwise any alphanumeric string; Stores2 field: Provincia; max len 5.
country	nation codes in Stores2; Stores2 field: Nazione; max len 4.
email	Email address; Stores2 field: email; max len 50.
zipcode	zip code; Stores2 field: Cap; max len 10.
son1surname	Child 1 last name; Stores2 field: CognomeFiglio1; max len 50
son1name	Child 1 name; Stores2 field: NomeFiglio1; max len 50
son1sex	Child 1 sex (M/F); Stores2 field: SessoFiglio1; max len 1
son1birthdate	Child 1 date of birth (yyyymmdd format); Stores2 field: DataNascitaFiglio1.
son2surname	Child 2 last name; Stores2 field: CognomeFiglio2; max len 50
son2name	Child 2 name; Stores2 field: NomeFiglio2; max len 50
son2sex	Child 2 sex (M/F); Stores2 field: SessoFiglio2; max len 1
son2birthdate	Child 2 date of birth (yyyymmdd format); Stores2 field: DataNascitaFiglio2.
son3surname	Child 3 last name; Stores2 field: CognomeFiglio3; max len 50
son3name	Child 3 name; Stores2 field: NomeFiglio3; max len 50
son3sex	Child 3 sex (M/F); Stores2 field: SessoFiglio3; max len 1
son3birthdate	Child 3 date of birth (yyyymmdd format); Stores2 field: DataNascitaFiglio3.
son4surname	Child 4 last name; Stores2 field: CognomeFiglio4; max len 50
son4name	Child 4 name; Stores2 field: NomeFiglio4; max len 50
son4sex	Child 4 sex (M/F); Stores2 field: SessoFiglio4; max len 1
son4birthdate	Child 4 date of birth (yyyymmdd format); Stores2 field: DataNascitaFiglio4.
dataflag0	**Stores2 field: DataFlag0;** (yyyymmdd format)**.**
dataflag1	**Stores2 field: DataFlag1;** (yyyymmdd format)**.**
dataflag2	**Stores2 field: DataFlag2;** (yyyymmdd format)**.**
dataflag3	**Stores2 field: DataFlag3;** (yyyymmdd format)**.**
dataflag4	**Stores2 field: DataFlag4;** (yyyymmdd format)**.**
dataflag5	**Stores2 field: DataFlag5;** (yyyymmdd format)**.**
dataflag6	**Stores2 field: DataFlag6;** (yyyymmdd format)**.**
dataflag7	**Stores2 field: DataFlag7;** (yyyymmdd format)**.**
dataflag8	**Stores2 field: DataFlag8;** (yyyymmdd format)**.**
dataflag9	**Stores2 field: DataFlag9;** (yyyymmdd format)**.**

1.2.5 Example

http://localhost/Stores2WS/FidelityGenericSubmitCustomerData.aspx?az=14&pf=1&nd=001&chkmail=1&nozip=1&=XML&nozip=1&bc=123456789012&node=001&code=000001&name=NOME&surname=COGNOME&address=INDIRIZZO&cell=3331234567&location=Milano&province=MI&country=ITA&email=INDIRIZZOEMAIL&zipcode=123451

1.2.6 Return value

The web service returns an XML document containing the result of the operation (or a code related to an error occurring during processing, 0 if there were no errors).

<data error="0"/>