

Web service Loy4Casse

Sommario

Prefazione.....	2
Restituzione saldo ed anagrafica tessera	3
Acquisizione saldo movimenti punti	6
Informazioni relative ad un coupon passato in cassa	8
Notifica utilizzo coupon.....	8
Tracciato file xml contenente i dati da inviare al web service	9
Initial load dei saldi utilizzando file xml	10

Prefazione

Il web service è utilizzato dalle casse per la gestione dei saldi tessera e risponde al seguente indirizzo:
<http://webservice.crai.org/Loy4CasseWS/Loy4CasseService.aspx>

Il servizio contiene i seguenti metodi divisi per categoria:

- per la gestione dello scontrino online:
 - setMovimento
 - setMovimento2
 - setMovimentoArray
- per la gestione loyalty online:
 - ISAlive: ritorna True
 - PuntoVenditaAlive
 - ReturnLogFile
 - ReturnLogFileFilter
 - ScontrinoEsistente: servizio windows da installare sui pc dei punti vendita, nel caso in cui la cassa abbia problemi di rete e non riesca ad inviare il saldo punti crea un xml che poi viene inviato dal servizio TLSaldiPuntiOnLine
 - SaldoTessera
 - AnagraficaTessera
 - SaldoMovimentiPunti: punti spesi di una tessera
 - SaldoMovimentiPuntiServizio
 - InfoTessera: per conoscere i punti di una tessera
 - ElencoMovimentiFidelity
 - InfoCoupon
 - AggiornaFlagCoupon
 - getAnagraficaTessera
 - getAnagraficaTesseraArray
 - getStatoTessera
 - getStatoTesseraArray
 - getMontanti
 - getMontantiArray
- per la gestione dell'aggiornamento dei programmi di supporto:
 - TLSaldiOnLine_versione: ritorna la versione di TLSaldiPuntiOnLine
 - iShuttleInterface_versione: ritorna la versione di
- per il ping:
 - ping
- vuoti:
 - getIniziativaCassaByType
 - getPdvMonitorConfiguration
 - getPrenotazioni
 - getPrenotazioniArray
 - getStatoTessera2
 - getStatoTesseraArray2
 - invokeMonitorService
 - setMovimentoArray2
 - setStatisticheMonitorOnline

Restituzione saldo ed anagrafica tessera

Il web service espone il metodo *InfoTessera (CodiceTessera, CodECRPV)* e in input richiede i seguenti parametri:

1. Codice tessera fidelity (tipo stringa)
2. Codice ECR di TLoalty del punto vendita che ha emesso lo scontrino (tipo stringa)

il response è di questo tipo:

```
<?xml version="1.0" encoding="utf-8"?>
<InfoTesseraFidelity xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://webservices.loy4casse.it/">
  <CodTesLoy></CodTesLoy>
  <Collections>
    <Collection>
      <codcol ></codcol>
      <monCentrale>
        <mongod></mongod>
        <monmat></monmat>
      </monCentrale>
      <monEsterni>
        <mongod></mongod>
        <monmat></monmat>
      </monEsterni>
      <monLocali>
        <mongod></mongod>
        <monmat></monmat>
      </monLocali>
      <tipcol></tipcol>
    </Collection>
  </Collections>
  <cogTesLoy></cogTesLoy>
  <ablTesLoy></ablTesLoy>
  <circolarita></circolarita>
  <Clusters>
    <Cluster>
      <tipo></tipo>
      <valore></valore>
    </Cluster>
  </Clusters>
</InfoTesseraFidelity>
```

All'url <http://webservice.crai.org/Loy4CasseWS/Loy4CasseService.asmx?op=InfoTessera> è disponibile un esempio completo di richiesta e risposta SOAP.

I campi gestiti attualmente sono:

1. *codtesloy*: codice della tessera fidelity ricevuta in input
2. *codcol*: codice della raccolta fidelity
3. *monCentrale (montante centrale)*:
 - a. *mongod*: punti goduti
 - b. *monmat*: punti maturati

Il saldo totale risulta quindi essere dato da: *punti maturati – punti goduti*

Viene restituito un nodo *<collection>* per ogni raccolta attiva (identificata dal campo *codcol*), ossia per ogni raccolta per cui vale: data inizio <= data odierna <= data fine ritiro premi.

I montanti esterni (*<monEsterni>*) e locali (*<monLocali>*) non sono gestiti.

4. *cogtesloy*: cognome dell'intestatario della tessera
5. *abltesloy*: vale "S" se la tessera è abilitata, "N" altrimenti
6. *circolarita*: vale "S" se la tessera è in circolarità, "N" altrimenti (ossia se appartiene al pv passato come parametro al metodo)
7. *cluster*: cluster della tessera (vengono esportati tutti i cluster, considerare solo quello con tipo = "01")

Se il campo "*codcol*" vale "9999" si tratta di borsellino elettronico: in tal caso i valori del montante centrale dovranno essere intesi come valore in euro a virgola implicita (ossia valore espresso in centesimi di euro) e non punti.

Acquisizione saldo movimenti punti

Il metodo da interrogare per inviarci il saldo movimenti punti è *SaldoMovimentiPunti*.

Rispetto alla versione precedente è stato modificato per consentire alla cassa di fare una sola chiamata.

Di seguito un esempio di richiesta SOAP:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<SaldoMovimentiPunti xmlns="http://webservices.loy4casse.it/">
  <CodECPV>string</CodECPV>
  <CodiceTessera>string</CodiceTessera>
  <NumeroCassa>int</NumeroCassa>
  <NumeroScontrino>int</NumeroScontrino>
  <DataOraScontrino>dateTime</DataOraScontrino>
  <TotaleScontrino>decimal</TotaleScontrino>
  <ListaInfoRaccolta >
    <InfoRaccolta>
      <CodiceRaccolta>string</CodiceRaccolta>
      <PuntiRaccolta>int</PuntiRaccolta>
      <CodiceVenditaPremio>string</CodiceVenditaPremio>
      <QuantitaPremi>int </ QuantitaPremi >
      <FlagPrenotazione>int</ FlagPrenotazione>
    </InfoRaccolta>
    <InfoRaccolta>
      <CodiceRaccolta>string</CodiceRaccolta>
      <PuntiRaccolta>int</PuntiRaccolta>
      <CodiceVenditaPremio>string</CodiceVenditaPremio>
      <QuantitaPremi>int </ QuantitaPremi >
      <FlagPrenotazione>int</ FlagPrenotazione>
    </InfoRaccolta>
  </ListaInfoRaccolta>
</SaldoMovimentiPunti>
</soap:Body>
</soap:Envelope>
```

All'url <http://webservice.crai.org/Loy4CasseWS/Loy4CasseService.asmx?op=SaldoMovimentiPunti> è disponibile un esempio completo di richiesta e risposta SOAP.

Esso richiede in input i seguenti parametri:

1. Codice ECR di TLoyalty del punto vendita (tipo stringa)
2. Codice tessera fidelity (tipo stringa)
3. Numero di cassa che ha emesso lo scontrino (tipo intero)
4. Numero di scontrino (tipo intero)
5. Data e ora dello scontrino (tipo datetime)
6. Totale dello scontrino (tipo decimal)
7. Una lista (di lunghezza arbitraria) di elementi *InfoRaccolta* che contengono i valori:
 - a. *CodiceRaccolta* : codice della raccolta fidelity (tipo stringa): corrispondente al valore del campo *codcol* ritornato dal metodo *InfoTessera*
 - b. *PuntiRaccolta* : numero di punti (tipo intero: con segno positivo in caso di punti erogati o negativo in caso di punti riscossi)
 - c. *CodiceVenditaPremio*: codice a barre del premio (solo per il caso di prenotazione/ritiro premio)
 - d. *QuantitaPremi*: quantità dei premi ritirati/prenotati (solo per il caso di prenotazione/ritiro premio)
 - e. *FlagPrenotazione*: 1 = prenotazione, 0 = ritiro (solo per il caso di prenotazione/ritiro premio)

Analogamente a quanto scritto per il metodo *InfoTessera*, nel caso di borsellino elettronico il valore monetario dovrà essere inviato con virgola implicita nel campo punti, facendo riferimento alla raccolta "9999".

Nel caso di sovrapposizione di raccolte l'erogazione dei punti avviene sulla nuova raccolta, per cui dovrà essere inviata l'informazione relativa al codice della raccolta nuova e il totale dei punti erogati (valore positivo), ma anche, in caso di regalo della raccolta precedente, l'informazione relativa al codice della raccolta precedente ed il valore negativo dei punti per il ritiro regalo. Quindi in caso di regalo/buono è sempre necessario inviare l'informazione relativa al codice della raccolta a cui si riferisce il regalo/buono ed il totale punti riscossi dal cliente (con segno negativo).

In caso di acquisizione dati effettuata senza errori il metodo restituirà 1, altrimenti uno dei seguenti codici di errore a valore negativo:

- -1: punto vendita non trovato
- -2: tessera fidelity non trovata
- -3: raccolta fidelity non trovata
- -4: premio non trovato
- -5: non è stato trovato un accumulatore valido
- -6: il punto vendita non è abilitato all'uso del web server
- -8: data scontrino troppo vecchia rispetto alla data corrente (>= 24h di differenza)
- -9: Data/ora scontrino più vecchia di 8 ore e scontrino già esistente nel datacollect
- -99: errore generico

Informazioni relative ad un coupon passato in cassa

Il metodo da interrogare per avere informazioni su un coupon è: *InfoCoupon (CodTessera, Coupon, CodECPV)*; esso prende in input i seguenti parametri:

1. Codice tessera fidelity (tipo stringa)
2. Coupon (tipo stringa)
3. Codice ECR di TLoyalty del punto vendita che ha emesso lo scontrino (tipo stringa)

All'url <http://webservice.crai.org/Loy4CasseWS/Loy4CasseService.asmx?op=InfoCoupon> è disponibile un esempio completo di richiesta e risposta SOAP.

In caso il coupon sia valido il metodo restituirà *1*, altrimenti uno dei seguenti codici di errore a valore negativo:

- **-1**: punto vendita non trovato
- **-2**: tessera fidelity non trovata
- **-3**: coupon non associato alla tessera
- **-4**: coupon associato alla tessera ma già usato dalla tessera
- **-5**: coupon associato alla tessera ma usato fuori dal periodo di validità (rispetto alla data odierna)
- **-99**: errore generico

Notifica utilizzo coupon

Il metodo da richiamare per inviarci l'informazione che uno o più coupon sono stati utilizzati da una tessera in uno scontrino è: *AggiornaFlagCoupon(CodTessera, ListaCoupon, CodECPV)*; esso prende in input i seguenti parametri:

1. Codice tessera fidelity (tipo stringa)
2. Lista di coupon (tipo stringa): indica il/i coupon utilizzati: se sono più di uno devono essere separati da virgola senza spazi; es.: 111,222,333,444
3. Codice ECR di TLoyalty del punto vendita che ha emesso lo scontrino (tipo stringa)

All'url <http://webservice.crai.org/Loy4CasseWS/Loy4CasseService.asmx?op=AggiornaFlagCoupon> è disponibile un esempio completo di richiesta e risposta SOAP.

In caso di aggiornamento effettuato con successo il metodo restituirà *1*, altrimenti uno dei seguenti codici di errore a valore negativo:

- **-1**: punto vendita non trovato
- **-2**: tessera fidelity non trovata
- **-99**: errore generico

Eventuali coupon passati come parametro ma non trovati nel database TLoyalty vengono ignorati.

Tracciato file xml contenente i dati da inviare al web service

Nel caso in cui la cassa non possa collegarsi al web service ed utilizzare il metodo SaldoMovimentiPunti, è previsto che crei un file xml nella cartella C:\Temp\DatiScontrini avente nome del tipo: CodECPV_NumCassa_NumScontrino_DataScontrino.xml, dove la data dello scontrino è nel formato AAAMMGGHHMMSS.

Di seguito lo schema XSD del file; i dati da scrivere nel file xml sono sostanzialmente gli stessi che dovrebbero essere passati come parametri al metodo SaldoMovimentiPunti; è stato aggiunto il nodo facoltativo “**ListaCoupon**” contenente il/i coupon utilizzati (se sono più di uno devono essere separati da virgola senza spazi, es.: <ListaCoupon>111,222,3333</ListaCoupon>):

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://webservices.loy4casse.it/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="SaldoMovimentiPunti">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="CodECPV"/>
        <xs:element type="xs:string" name="CodiceTessera"/>
        <xs:element type="xs:int" name="NumeroCassa"/>
        <xs:element type="xs:int" name="NumeroScontrino"/>
        <xs:element type="xs:dateTime" name="DataOraScontrino"/>
        <xs:element type="xs:decimal" name="TotaleScontrino"/>
        <xs:element type="xs:string" name="ListaCoupon" minOccurs="0" maxOccurs="1" />
        <xs:element name="ListaInfoRaccolta">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="InfoRaccolta" maxOccurs="unbounded" minOccurs="1">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="CodiceRaccolta"/>
                    <xs:element type="xs:string" name="PuntiRaccolta"/>
                    <xs:element type="xs:string" name="CodiceVenditaPremio"/>
                    <xs:element type="xs:int" name="QuantitaPremi"/>
                    <xs:element type="xs:int" name="FlagPrenotazione"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Initial load dei saldi utilizzando file xml

Se per fare l'initial load dei saldi vengono utilizzati file xml, prestare attenzione che alla fine della procedura non siano presenti all'interno del percorso **C:\Temp\DatiScontrini\Errori**, file con estensione xml.

Nel caso fossero presenti a causa di errori o di un timeout del web service, copiarli nella cartella superiore C:\Temp\DatiScontrini\ per permettere un nuovo invio.