LOYALTY PARTNER

POS-Terminal Interface Definition for LMS 4 - “Instore Product Reward”

Author:	Andreas Schwartze
Creation Date:	19.05.2005
Last changes:	11.01.2007
Version:	2.6

1 Document Control

Change Record

Date	Author	Version	Change Reference
19.05.2005	ASchwartze	2.0	Initial Revision
26.07.3005	ASchwartze	2.1	Additional Command "CheckCardRedemption", Updated error codes in Authenticate, AuthenticateAlternate and AccountDetail to reflect latest application release
17.08.2005	ASchwartze	2.2	• Date Format adjusted in in – parameter "DOB" of command AuthenticateAlternateAndRedeemPoints • Parameter "redeemable" renamed to "registered" in command CheckCardForRedemption
18.08.2005	ASchwartze	2.3	• Parameter "registered" removed from command CheckCardForRedemption • New error Text for error –203 • Notation of parameter names adapted to LP-standards.
01.09.2005	Szausch	2.4	• Changed xml samples for command CheckCardForRedemption • Changed terminaltransactionID to terminalTransactionID
13.09.2005	Aschwartze	2.5	• New Error Code –287 added to command CheckCardForRedemption • Error in Example for command AutoRefundPoints removed
11.01.2007	Aschwartze	2.6	• Added Cardnumber concept in appendix A • Added description of allowed cardnumber parameters in command specification

2 Contents

3 Introduction

This document defines the API that is to be used by terminals at partner's POS (e.g. ec-cash Terminals or POS Terminals) to process redemption of type "Instore Product Rewards" in the Payback system. The API describes an interface to the Payback System (POS System) that consists of commands that can be submitted to the interface an receive responses from the interface. The following chapters contain:

- A business description of the redemption functionality required,
- An overview of the technical architecture,
- A definition of the XML schema which will be the interface between the systems,
- The commands used to perform an Instore Product Reward.

4 Specification

The workflow that incorporates the interface commands described is called "Instore Product Reward" or "Pay with Payback Points". Its main purpose is to redeem Payback points in real-time from the customers Payback account via an online interface in partner stores and receive goods in exchange. In that way Payback points can be regarded as a currency –equivalent where 1 Payback Point ~ 1 EuroCent.

The purpose and interaction of the commands is as follows:

- CheckCardForRedemption: When a card is swiped at the client- terminal this command is used for checking the Payback card for validity and obtaining the account balance and the points that are available for redemption. There exists an upper limit for the number of points that can be redeemed by a single transaction over this interface.

- AuthenticateAndRedeem: Authenticates the Payback member via its PIN and redeems the given amount of points. The client provides a unique transactionID for each submitted redemption-command that identifies the redemption transaction on the client side. The Payback interface returns a receiptID that identifies the redemption within the Payback system and is used for follow up handling within the Payback program. The receiptID can be printed on the receipt that is handed out to the client. The redemption command fails, no further refund is required.

- AuthenticateAlternateAndRedeem: Analogue to AuthenticateAndRedeem command. Authenticates the Payback member via ZIP-Code and date of birth and redeems the given amount of points.

- AutoRefundPoints: In case the client submitted an AuthenticateAndRedeem and received no response from the interface it can submit an auto-refund to cancel the redemption identified by the transactionID. If no redemption could be identified nothings happens. The auto –refund can be submitted several times using the same transactionID, at most one refund will be executed. This command cannot be used to process user initiated refunds.

All commands incorporate in the workflow are stateless, thus can be independently called from any of the commands.

5 Architecture

The following diagram shows the architecture of the POS system. It shows how a request is passed from a POS terminal to the payback system and how the response is passed back to the terminal.

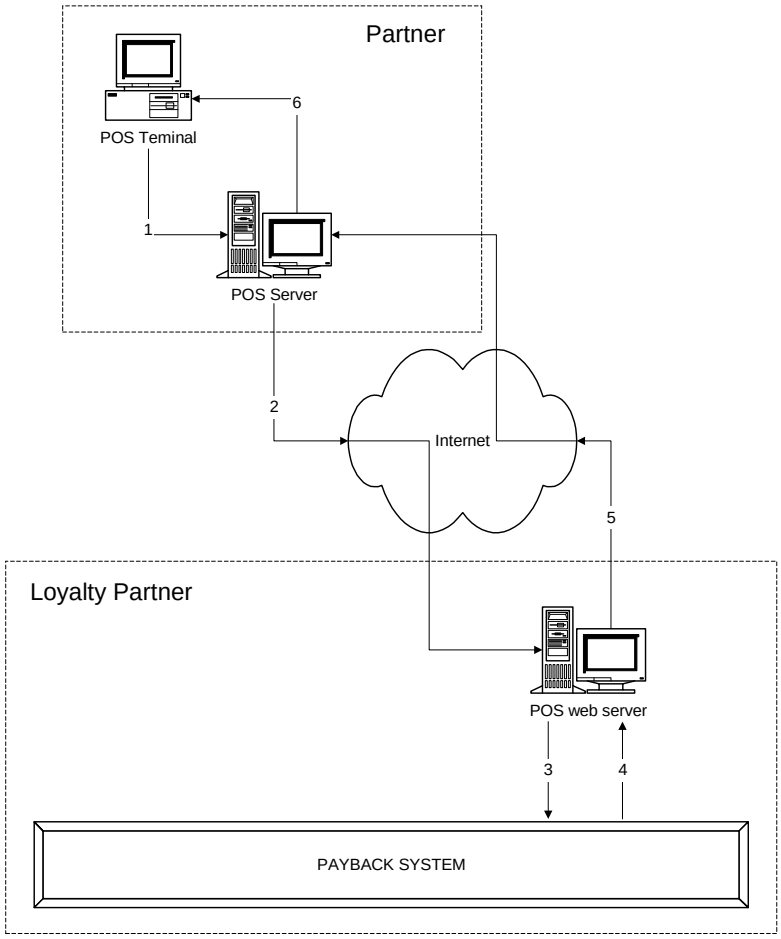


Figure1: Arcitecture of PAYBACK POS Terminal Systems

1) POS Terminal requests PAYBACK data via HTTPS
2) POS terminal server request information from PAYBACK by sending an XML request via HTTPS
3) POS web server requests PAYBACK data from the PAYBACK system
4) PAYBACK system returns the required data to the POS web server
5) POS web server send an XML response to the POS terminal server via HTTPS
6) POS terminal server renders data and sends the HTML page to the POS terminal via HTTPS

5.1 System security

Loyalty Partner will use the following main methods to ensure security:

1) The payback system will only accept calls from specific IP addresses. This will mean that only the POS Servers will be able to retrieve information from the payback system. I.e. that the IP-addresses of the machines establishing the connection via the Internet to Loyalty Partner's communication server have to be listed explicitly and must not be changed without prior notification to Loyalty Partner. If this notification does not occur then the POS system will not be functional.

2) The protocol used is https. Using SSL the data is encrypted between the POS Server and the POS web server.

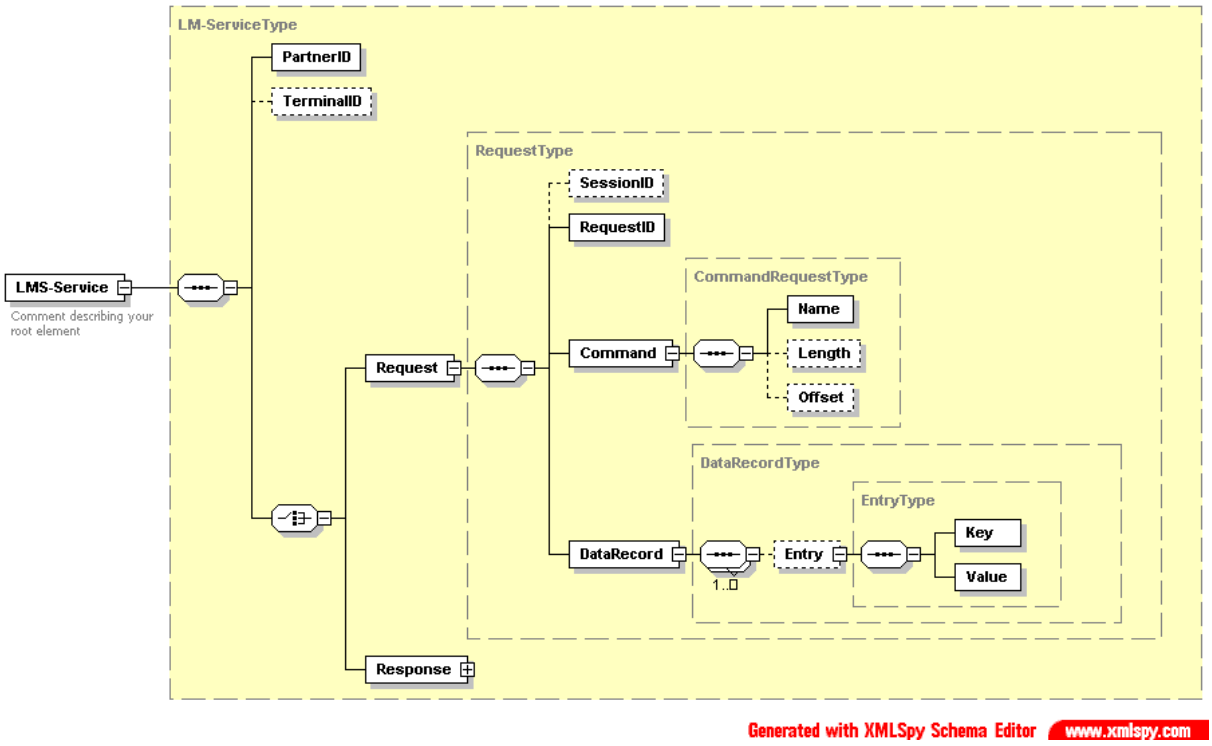
6 XML interface definition

The XML interface basically consists of 2 different services, the LMS-Service Request and the LMS-Service Response. A partner asks for a specific service issuing an LMS-Service Request with his PartnerID, defining the desired task with command.name and passing the needed parameters as key/value pairs in the dataRecord section of the XML structure. The RequestID generated by the partner can uniquely identify each request. After successful authentication subsequent LMS-Service requests additionally transfer the given SessionID.

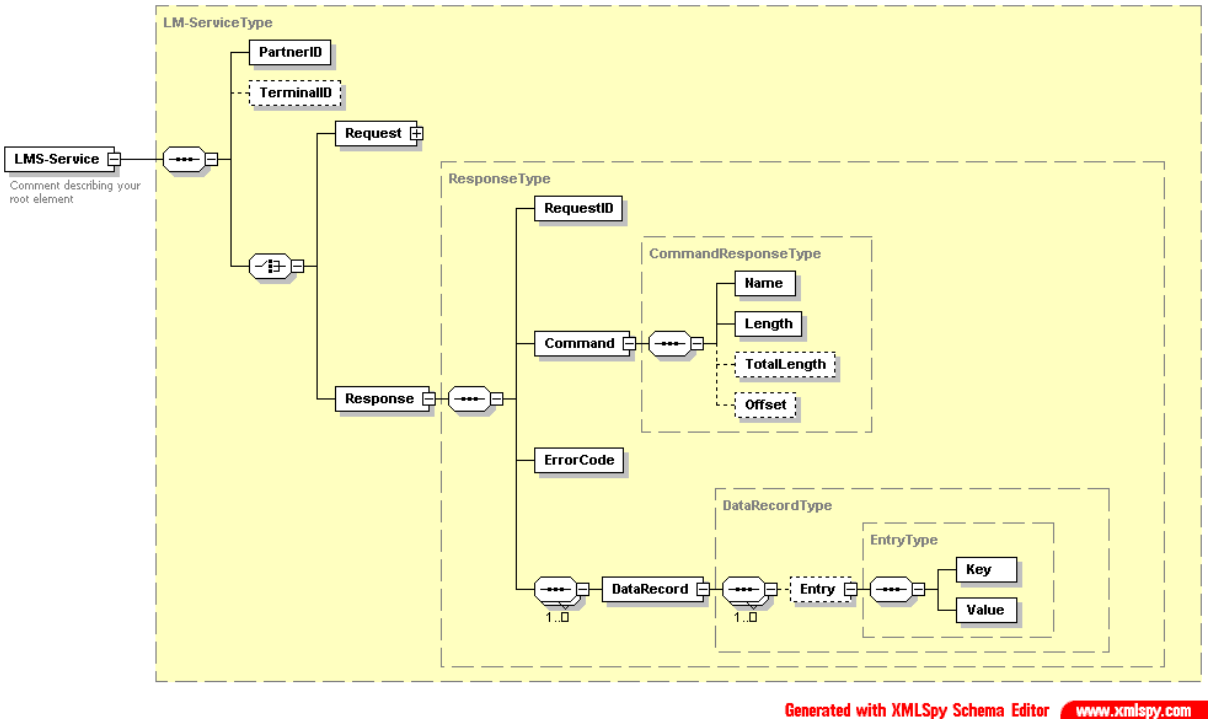
The LMS-Server answers using the LMS-Service Response XML structure. In addition to the dataRecord and command tags, the response contains an error element indicating a successful or erroneous execution of the service.

The encoding used by the application is ISO-8859-1. This is specified in both the XML and the HTTP headers.
Remark: The implementation expects a "newline" at the end od the first tag (<?xml …>).

6.1 XML Request



6.2 XML Response



6.3 Element definitions

The following table defines each of the XML entries:

Name	Type	Size	Description
PartnerID	Number	11	The partnerID of the partner that is making the call.
TerminalID	String		The TerminalID of the POS Terminal that is making the call.
SessionID	String	60	The sessionID for this session. This is generated by payback after first call and is returned by the calling system on each subsequent request. The commands which will generate the sessionID are: AuthenticateAndRedeemPoints and AuthenticateAlternateAndRedeemPoints. The session will be invalidated at logout or in case of a timeout. For commands where “Session Required” = false, the input parameter sessionid is optional. If there is no sessionID then the element should be left out of the XML completely. A SessionID of “” will not be parsed. If sessionID is required and does not exist then –221 (invalid session) will be returned rather than –224 (invalid XML parameter)
RequestID	Number	11	A way of pairing responses and requests. This is a random number generated by the calling system.
Name	String		The name of the command being called.
Length	Number		On request the length of the recordset to be returned (mandatory for AccountDetail, optional for other requests). On response the number of records returned (always mandatory). The default is 1. In cases where no key value pair exists (e.g. just confirmation of service execution with errorcode 0) Length equals 0.
TotalLength	Number		The total number of records that could be returned.
Offset	Number		The offset of the records that have been returned from the start of the total length. The default is 0.
Key	String		The key of a key-value pair. The pairings are described in depth in the commands and parameters section of this document.
Value	String		The value of a key value pair.
ErrorCode	Number	11	The system error code for any errors that have occurred. If there have been no errors then it will be equal to 0.

6.4 Error Codes

The following error codes can be expected from problems with the XML request. All errors cause processing to stop.

- -202 – internal error
  One or more of the required parameters do not fit the format conditions or are not present or System error that does not allow any further processing.

- - 203 – invalid cardnumber
  The parameter 'cardnumber' is not identified by the Luhn- algorithm as a valid Payback cardnumber because no data for this card can be found in the database, the account ist diabled or no member could be found for the card. This error results in a cancellation of the workflow.

- -204 – invalid authentication
  The supplied PIN value does not correspond to the one in the DB or the supplied ZIP-code and DOB are not valid.

- -216 – not enough points
  The member does not have enough available points to complete this transaction or the number of points requested for redemption is below the mininum number allowed or the number of points to redeem exceeds the maximum allowed number of points for redemption.

- -287 – card is not registered
  The given cardnumber is a valid Payback card but not yet registered for the program, thus no member is assigned to the card and card is not assigned to an account.

If the XML is unreadable (e.g. there is a tag that has not been correctly closed or there is an element missing) then an HTTP error with the code 500 will be returned instead of a response file containing an error.

7 Commands

This section describes all of the commands that are needed to return the data required by the functionality described above.

For each command the following are provided:

- A description of what the command does and when it should be used.
- A list of all IN parameters.
- A list of all OUT parameters.
- A statement as to whether a session is required to call the particular command.
- A list of all error states in the order in which they are invoked.
- An example of the XML that should be generated by a response and a request from each command.

Unless otherwise stated all IN parameters are required fields.

If you wish more information on an individual parameter or error state then you should see the following two sections of this document.

7.1 CheckCardForRedemption

7.1.1 Description

Checks the given cardnumber for validity for redemption in the Payback System and returns various account balances of card's account.

7.1.2 IN parameters:

1) TerminalID (max. 10-digits, via element <TerminalID>)
2) RequestID (max. 8-digits, via element <RequestID>): contains the sequence number of identical requests
3) cardnumber (arbitrary standard Payback cardnumber, 16-digits, prefix 308342, as well as arbitrary PAYBACK paycardnumber, 19-digits, prefix 67251099, in both cases not necessary account holder card)

7.1.3 Session Required?

NO

7.1.4 OUT parameters:

1) acctbalance (max. 10-digits): number of points on the members account
2) available (max. 10-digits): points available for redemption
3) availableForNextRedemption (max. 10-digits): points available for the next redemption via ec-cash

7.1.5 Error codes:

External Error Code	Error Text at Terminal
-202	Interner Fehler, Vorgang abgebrochen
-203	Ungültige Karte, Vorgang abgebrochen
-287	Nicht registrierte Karte, Vorgang abgebrochen

7.1.6 Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<LMS-Service xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="F:\R09006POS_Terminals\3-Design\LM-Service.xsd">
      <PartnerID value="504"/>
      <TerminalID value="123567890"/>
      <Request>
         <RequestID value="1"/>
         <Command>
            <Name value="CheckCardForRedemption"/>
         </Command>
         <DataRecord>
            <Entry>
               <Key value="cardnumber"/>
               <Value value="3083421234567890"/>
            </Entry>
         </DataRecord>
      </Request>
</LMS-Service>
```

7.1.7 Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<LMS-Service xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="F:\R09006POS_Terminals\3-Design\LM-Service.xsd">
      <PartnerID value="504"/>
      <TerminalID value="123567890"/>
      <Response>
         <RequestID value="1"/>
         <Command>
            <Name value="CheckCardForRedemption"/>
            <Length value="1"/>
         </Command>
         <ErrorCode value="0"/>
         <DataRecord>
            <Entry>
               <Key value="acctbalance"/>
               <Value value="2567"/>
            </Entry>
            <Entry>
               <Key value="available"/>
               <Value value="2567"/>
            </Entry>
            <Entry>
               <Key value="availableForNextRedemption"/>
               <Value value="2567"/>
            </Entry>
         </DataRecord>
      </Response>
</LMS-Service>
```

7.2 AuthenticateAndRedeemPoints

7.2.1 Description

Authenticates the Payback Member via Payback Card and PIN and redeems the given amount of points from the card's account within one user-transaction. Uses TerminalTransactionID and optionally RequestID to identify and references the redemption from client side for possible refund. The TerminalTransactionID is a globally unique identifier for the redemption, the terminalID describes the source or trigger of the redemption. Both the optional TerminalSequenceNo and Branchnumber are additional Information to externally reference the redemption and describe the source (or trigger) of the redemption. The command returns a receiptID and the receiptAmount to be printed or displayed.

7.2.2 IN parameters:

1) TerminalID (max. 10-digits, via element <TerminalID>)
2) RequestID (max. 8-digits, via element <RequestID>): contains the sequence number of identical requests (same TerminalTransactionID)
3) cardnumber (arbitrary standard Payback cardnumber, 16-digits, prefix 308342, as well as arbitrary PAYBACK paycardnumber, 19-digits, prefix 67251099, in both cases not necessary account holder card)
4) PIN (4 digit)
5) points (max 8 digits): points to redeem
6) terminalTransactionID (max 20-digits): globally unique identifier for the redemption transaction
7) branchnumber (optional, max. 12-digits): source branch where the terminal resides

7.2.3 Session Required?

No

7.2.4 OUT parameters:

1) receiptID (max. 20- digits): partner-specific unique redemption number to be printed onto the receipt
2) receiptAmount (decimal number, decimal separator is '.'): Redeemed amount in Euros

7.2.5 Error codes:

External Error Code	Error Text at the Terminal
-202	Interner Fehler, Vorgang abgebrochen
-203	Ungültige Karte, Vorgang abgebrochen
-204	Ungültige Authentifizierung, Vorgang abgebrochen
-216	Nicht genügend Punkte, Vorgang abgebrochen

7.2.6 Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<LMS-Service xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="F:\R09006POS_Terminals\3-Design\LM-Service.xsd">
```

```
  <PartnerID value="504"/>
  <TerminalID value="123567890"/>
  <Request>
     <RequestID value="1"/>
     <Command>
        <Name value="AuthenticateAndRedeemPoints"/>
     </Command>
     <DataRecord>
        <Entry>
           <Key value="cardnumber"/>
           <Value value="3083421234567890"/>
        </Entry>
        <Entry>
           <Key value="PIN"/>
           <Value value="1234"/>
        </Entry>
        <Entry>
           <Key value="points"/>
           <Value value="4321"/>
        </Entry>
        <Entry>
           <Key value="terminalTransactionID"/>
           <Value value="567890654321"/>
        </Entry>
        <Entry>
           <Key value="branchnumber"/>
           <Value value="1234567890"/>
        </Entry>
     </DataRecord>
  </Request>
</LMS-Service>
```

7.2.7 Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<LMS-Service xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="F:\R09006POS_Terminals\3-Design\LM-Service.xsd">
  <PartnerID value="504"/>
  <TerminalID value="123567890"/>
  <Response>
     <RequestID value="1"/>
     <Command>
        <Name value="AuthenticateAndRedeemPoints"/>
        <Length value="1"/>
     </Command>
     <ErrorCode value="0"/>
     <DataRecord>
        <Entry>
           <Key value="receiptID"/>
           <Value value="567890654321"/>
        </Entry>
        <Entry>
           <Key value="receiptAmount"/>
           <Value value="43.12"/>
        </Entry>
     </DataRecord>
  </Response>
</LMS-Service>
```

7.3 AuthenticateAlternateAndRedeemPoints

7.3.1 Description

Authenticates the Payback Member via Payback Card and ZIP-Code plus Date of birth and redeems the given amount of points from the card's account within one user-transaction. Uses TerminalTransactionID and optionally RequestID to identify and references the Redemption from client side for possible refund. The TerminalTransactionID is a globally unique identifier

for the redemption, the terminalID describes the source or trigger of the redemption. Both the optional TerminalSequenceNo and Branchnumber are additional information to externally reference the redemption and describe the source (or trigger) of the redemption. The command returns a receiptID and the receiptAmount to be printed or displayed.

Attention: Loyalty Partner reserves the right to remove this command in case of abuse or other problems

7.3.2 IN parameters:

1) TerminalID (max. 10-digits, via element <TerminalID>)
2) RequestID (max. 8-digits, via element <RequestID>): contains the sequence number of identical requests (same TerminalTransactionID)
3) Cardnumber (arbitrary standard Payback cardnumber, 16-digits, prefix 308342, as well as arbitrary PAYBACK paycardnumber, 19-digits, prefix 67251099, in both cases not necessary account holder card)
4) zip (5 digits)
5) dob (format: DD.MM.YYYY)
6) points (max 8 digits): points to redeem
7) terminalTransactionID (max 20-digits): globally unique identifier for the redemption transaction
8) branchnumber (optional, max. 12-digits): source branch where the terminal resides

7.3.3 Session Required?

No

7.3.4 OUT parameters:

1) receiptID (max. 20- digits): partner-specific unique redemption number to be printed onto the receipt
2) receiptAmount (decimal number, decimal separator is '.'): Redeemed amount in Euros

7.3.5 Error codes:

External Error Code	Error Text at the Terminal
-202	Interner Fehler, Vorgang abgebrochen
-203	Ungültige Karte, Vorgang abgebrochen
-204	Ungültige Authentifizierung, Vorgang abgebrochen
-216	Nicht genügend Punkte, Vorgang abgebrochen

7.3.6 Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<LMS-Service xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="F:\R09006POS_Terminals\3-Design\LM-Service.xsd">
  <PartnerID value="504"/>
  <TerminalID value="123567890"/>
  <Request>
     <RequestID value="1"/>
     <Command>
        <Name value="AuthenticateAlternateAndRedeemPoints"/>
     </Command>
     <DataRecord>
        <Entry>
```

```
            <Key value="cardnumber"/>
            <Value value="3083421234567890"/>
         </Entry>
         <Entry>
            <Key value="zip"/>
            <Value value="12345"/>
         </Entry>
         <Entry>
            <Key value="dob"/>
            <Value value="24.12.1977"/>
         </Entry>
         <Entry>
            <Key value="points"/>
            <Value value="4321"/>
         </Entry>
         <Entry>
            <Key value="terminalTransactionID"/>
            <Value value="654321"/>
         </Entry>
         <Entry>
            <Key value="branchnumber"/>
            <Value value="1234567890"/>
         </Entry>
      </DataRecord>
   </Request>
</LMS-Service>
```

7.3.7 Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<LMS-Service xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="F:\R09006POS_Terminals\3-Design\LM-Service.xsd">
  <PartnerID value="504"/>
  <TerminalID value="123567890"/>
  <Response>
     <RequestID value="1"/>
     <Command>
        <Name value="AuthenticateAlternateAndRedeemPoints"/>
        <Length value="1"/>
     </Command>
     <ErrorCode value="0"/>
     <DataRecord>
        <Entry>
           <Key value="receiptID"/>
           <Value value="567890654321"/>
        </Entry>
        <Entry>
           <Key value="receiptAmount"/>
           <Value value="43.12"/>
        </Entry>
     </DataRecord>
  </Response>
</LMS-Service>
```

7.4 AutoRefundPoints

7.4.1 Description

Refunds all points of the redemption referenced by the given TerminalTransactionID in the context of an auto-redemption after an unsuccessful redemption. The TerminalTransactionID is a globally unique identifier for the redemption, the terminalID describes the source or trigger of the redemption. No return parameters.

7.4.2 IN parameters:

1) TerminalID (max. 10-digits, via element <TerminalID>)
2) RequestID (max. 8-digits, via element <RequestID>): contains the sequence number of identical requests (same TerminalTransactionID)TerminalID (max. 8-digits)
3) terminalTransactionID (max 20-digits)

7.4.3 Session Required?

No

7.4.4 OUT parameters:

No

7.4.5 Error codes:

External Error Code	Error Text at the Terminal
-202	Interner Fehler, Vorgang abgebrochen

7.4.6 Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<LMS-Service xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="F:\R09006POS_Terminals\3-Design\LM-Service.xsd">
  <PartnerID value="504"/>
  <TerminalID value="123567890"/>
  <Request>
     <RequestID value="1"/>
     <Command>
        <Name value="AutoRefundPoints"/>
     </Command>
     <DataRecord>
        <Entry>
           <Key value="terminalTransactionID"/>
           <Value value="654321"/>
        </Entry>
     </DataRecord>
  </Request>
</LMS-Service>
```

7.4.7 Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<LMS-Service xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="F:\R09006POS_Terminals\3-Design\LM-Service.xsd">
  <PartnerID value="504"/>
  <TerminalID value="123567890"/>
  <Response>
     <RequestID value="1"/>
     <Command>
        <Name value="AutoRefundPoints"/>
        <Length value="1"/>
     </Command>
     <ErrorCode value="0"/>
     <DataRecord/>
  </Response>
</LMS-Service>
```

Appendix A Cardnumber concept

This appendix provides a brief description of the structure of the PAYBACK cardnumbers used and allowed in the context of Instore Product Reward.

Generally, both standard PAYBACK cards as well as PAYBACK paycards can be utilized for Instore Product reward processing at the client terminal when swiping the card. The magnetic stripe of the cards contains the following cardnumber structure:

Standard PAYBACK card:

Prefix	cardnumber:	Check digit (Luhn)
3 0 8 3 4 2	1 2 3 4 5 6 7 8 9	0

PAYBACK paycard:

Prefix	Accountnumber	Check digit (Luhn)
6 7 2 5 1 0 9 9	1 2 3 4 5 6 7 8 9 0	0