Universal Interface

Payment

File Name: INTERFACE_SPEC_REST-Payment-Interface_en_GB.docx
Creation: 12.02.2016
Index: -

This page is intentionally left blank.

Author: Karsten Schroeren/Jana Srponova

Modification log:
Refer to chapter "Record of Revisions"

Parking and leisure centre systems
Breite Straße 132
41238 Mönchengladbach
www.scheidt-bachmann.com

Subject to change.

20201023UI0504

This page is intentionally left blank.

Contents

1 Introduction

This document replaces the "REST Payment Interface 120206" document.

This documentation is valid for the Interface version from 1.3.1 to 1.4.0.

API Version and Schema Location
The Scheidt & Bachmann (S&B) *Payment Web Service* REST API is versioned. The current version is 1.1.0.

Many requests contain XML or JSON. How this data is structured is defined in an XML-Schema file *payment.xsd*.

1.1 Interface Overview

The *Payment* interface enables a third party system to book revenue data into S&B system, and to set barcode ticket as paid in S&B system optionally. That means if a customer pays a barcode ticket via the *Payment* interface, they can go out at exit without payment.

Combination of the Payment interface with the Ticket classification interface might be necessary.

The Payment interface provides the following features:

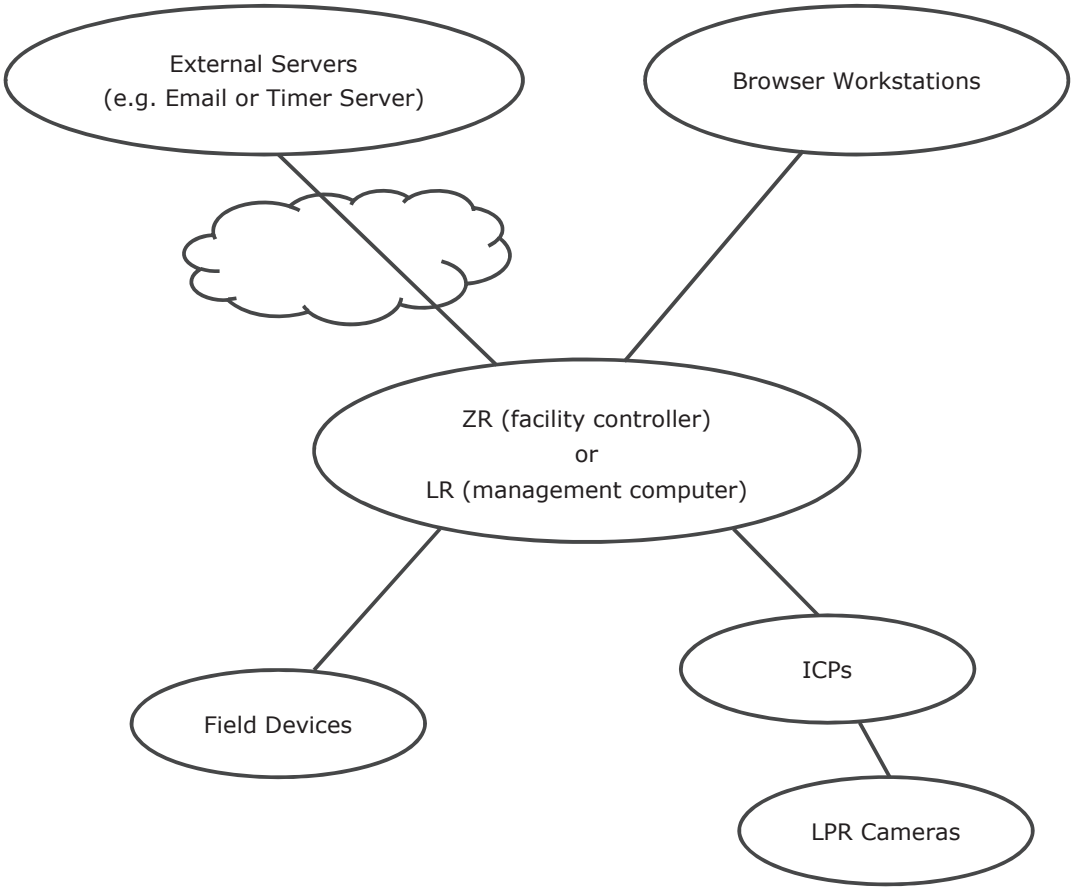
- ☐ Comprehensive entervo shift handling
  - Information about all cashiers (shift list includes shift ID, device, cashier ID, shift creation date and time, shift status, …)
  - Information about a specific cashier (shift list includes shift ID, device, cashier ID, shift creation date and time, shift status, …)
  - Create new entervo shift
  - Close shift
- ☐ Requesting and receiving entervo configured point of sale, cashier and sales article information
  - Point of sale devices list (response includes computer ID, device ID, device short name and long name, …)
  - Cashiers list (response includes cashier ID, cashier first name and last name, …)
  - Article list (response includes article ID, article short name and long name, quantity, amount, …)
- ☐ Booking capability
  - Book entervo pre-configured articles into an open shift
  - Pay for parking tickets
- ☐ Requesting and receiving shift sales transactions
  - Request sales transactions for a specific shift

1.2 Protocol

The following ports are used in the entervo system for communication via network.

Protocol	Source Port	Destination Port	
TCP	any	8443	https://
TCP	any	8580	http://

Directions of communication



1.3 Authentication and Authorisation

The application communicating with our web service requires the client authentication when sending a request. Based on the authentication information, the service will authorize (allow) or deny the request.

The authentication procedure will follow the http Basic Authentication Standard which is described in RFC 2617 (http://www.ietf.org/rfc/rfc2617.txt).

The authentication credentials (username and password) will be provided by Scheidt & Bachmann.

2 REST Interface

2.1 REST Requests

This chapter describes how to make REST requests to the Scheidt & Bachmann *Payment Web Service*.

The requests must be HTTPS requests as defined by RFC 2616 (for more information, go to http://www.ietf.org/rfc/rfc2616.txt).

A typical REST action consists of sending a single HTTPS request to *Payment Web Service* and waiting for the HTTP response. Like any HTTP request, a REST request to *Payment Web Service* contains a request method, a URI, request headers, and sometimes a query string or request body. The response contains an HTTP status code, response headers, and sometimes a response body.

The request URI always starts with a forward slash. The remainder of the URI indicates the particular resource you want to act on. Here is an example of the URI you use when listing shifts:
https://<sample.host.com>/PaymentWebService/version

2.1.1 Request Headers

The following table shows how the request headers must be structured:

Header Name	Description	Required
Authorisation	The information needed for request authentication.	Yes
Content-Length	Length of the message (without the headers) according to RFC 2616. Condition: Required if the request body itself contains information (most toolkits add this header automatically).	Conditional
Content-Type	The content type of the resource. Example: *application/xml*. Condition: Required for POST and PUT requests.	Conditional
Date	The format must be one of the full date formats specified in RFC 2616 section 3.1.1. For example: Wed, 05 Apr 2006 21:12:00 GMT. For more information, go to the RFC 2616 specification.	Yes
Host	The host being requested. The value must be *gsph.sub.com*.	Yes

Request Time Stamp

For security reasons the request header must contain a server-generated time stamp (Date Header).

The request must be received within 15 minutes of the system time (before the time stamp expires). If it is not, the request fails with the RequestExpired error code. This is to prevent replays of the requests by an adversary.

2.2 REST Responses

Payment Web Service responses are standard HTTP responses. Some of the actions return special information specific to *Payment Web Service* in the form of an HTTP header or XML in the body of the response. The specific details are covered in the API reference topic for particular action.

2.2.1 Error Responses

If a REST request results in an error, the HTTP reply has:

- An XML error document as the response body
- Content-Type header: application/xml
- An appropriate 3xx, 4xx or 5xx HTTP status code

The following is an example of the XML error document in a REST error response:

```
<ErrorResponse xmlns="http://gsph.sub.com/payment.xsd">
  <Error>
    <Code>14711</Code>
    <ShortMsg>InvalidURI</ShortMsg>
    <Message>Could not parse the specified URI.</Message>
  </Error>
</ErrorResponse>
```

A list of errors can be found in chapter 4 “Errors Handling”.

3 Resources

3.1 Resources Overview

The *Payment Web Service* provides access to the following resources:

Administrative Resources

Resource	Description
Version	Provides the version of the REST interface.
Resources	Provides a dictionary of all available application resources and there URIs.

Primary Application Resources

Resource	Description
Shift	Provides the head information of each shift which is assigned to a device and cashier.
Sales Block	Contains the information to each single payment in a shift.

For all resources listed above collection resources are provided which facilitate list queries.

3.2 Resources URI Overview

Resource	Path	Method	Description
Version	/version	GET	List version of the REST interface
Resources	/resources	GET	List all primary resources of the REST interface
Articles	/articles	GET	List of articles
Devices	/devices	GET	List of all point of sale devices
Cashiers	/cashiers	GET	List of all cashiers
Cashier Shifts	/cashiers/{cashierContractId},{cashierConsumerId}/shifts	GET, POST	List of all cashier shifts
Cashier Shift	/cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftId}	GET, PUT	Single cashier shift
Money Value of personalized money value card	/media/personalizedMoneyValue/{ContractId},{ConsumerId}	GET	Get actual money value of a personalized money value card

Resource	Path	Method	Description
Sales Transactions of Cashier Shift	/cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftId}/salestransactions	PUT	List of all sales transactions of a cashier shift
Sales Transaction of Cashier Shift	/cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftId}/salestransactions/{salesTransactionId}	GET	Sales Transaction of a cashier shift
Shifts	/shifts	GET, POST	List of all shifts
Shift	/shifts/{shiftId}	GET, PUT	Single shift
Sales Transactions of Shift	/shifts/{shiftId}/salestransactions	PUT	List of all sales transactions of a shift
Sales Transaction of Shift	/shifts/{shiftId}/salestransactions/{salesTransactionId}	GET	Sales Transaction of a shift
FastPayment - generate UUID	/uuid/	GET	Get new UUID for a new FastPayment transaction
FastPayment – sales transaction	/uuid/salestransactions/{transactionUUID}	PUT	Create new FastPayment transaction
FastPayment – get sales transaction details	/uuid/salestransactions/{transactionUUID}	GET	Get FastPayment sales transaction details
FastPayment transaction status	/uuid/salestransactions/{transactionUUID}/status	GET	Get FastPayment transaction status

3.2.1 Base URI

Base URI: *https://<sample.host.com>/PaymentWebService*

All other URIs described in this document will be appended to this URI.

3.2.2 Resources API

The following chapter contains the reference of the *Payment Web Service* API.

The first section contains an overview over all resources provided by the service and which actions (GET, PUT, POST, and DELETE) can be performed.

Subsequently all actions are described in detail. Each description comprises the following information:

Item	Description
Description	Textual description of the action performed by the service when receiving the request

Item	Description
URI	URI of the request. The URI may contain variables which have to be filled in by the client. The variables are placed in brackets.
HTTP-Method	Depending on the HTTP-Method sent by the request, the service executes the according action (values GET, PUT, POST, and DELETE).
Formats	The format of the data which can be sent by the client in the request body and will be accepted by the service. Depending on the client requirements, the data in the response body will be returned in the same format (e.g. JSON, XML).
Parameters	Parameters which can be appended to the URI (e.g. for specifying search parameters)
Example Header	Sample request performed with the curl utility (normally clients would use a http library appropriate for the programming language)
Example request body	Data expected by the service to be sent by the client in JSON or XML format
Example response body	Data sent by the service in the response to the client request

The list of parameters together with description and possible values can be found in chapter 6.4 "Appendix 4: List of Parameters".

3.2.2.1 Version

List a version of the REST interface.

REST-Resource

HTTP-Method	GET
REST-URI	/version
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	-
Response body	<version> element

Example Header

```
GET /payment/rest/version HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/json
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Ok	200	-	<version> element
Error	500	-	<error> element

Example Response Body (XML)

```
<?xml version=”1.0” encoding=”UTF-8”?>
<version xmlns="http://gsph.sub.com/payment/types">
  <version>1.1.0</version>
</version>
```

3.2.2.2 Resource Dictionary

List all available resources which are provided by the required version. This information can be used by clients to access URIs without hardcoding them.

REST-Resource

HTTP-Method	GET
REST-URI	/resources
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	-
Response body	<resources> element

Example Header

```
GET /payment/rest/resources HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/json
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Ok	200	-	<resources> element
Error	500	-	<error> element

Example Response Body (XML)

```
<?xml version=”1.0” encoding=”UTF-8”?>
<resources xmlns="http://gsph.sub.com/payment/types">
  <resource>devices</resource>
  <resource>cashiers</resource>
  <resource>shifts</resource>
  <resource>articles</resource>
</resources>
```

3.2.2.3 List Articles

List the key information for configured articles in the S&B system.

REST-Resource

HTTP-Method	GET
REST-URI	/articles
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	-
Response body	<articles> element

Example Header

```
GET /payment/rest/articles HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/xml
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Ok	200	-	<articles> element
Error	500	-	<error> element

Example Response Body (XML)

```
<?xml version=”1.0” encoding=”UTF-8”?>
<pay:articles xmlns:pay="http://gsph.sub.com/payment/types"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10100</pay:articleRef>
      <pay:articleShortName>Shortterm</pay:articleShortName>
      <pay:articleLongName>Shortterm Ticket</pay:articleLongName>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>0</pay:amount>
      <pay:influenceRevenue>1</pay:influenceRevenue>
      <pay:influenceCashFlow>1</pay:influenceCashFlow>
    </pay:article>
</pay:articles>
```

3.2.2.4 List Devices

List the key information for configured devices in the S&B system.

REST-Resource

HTTP-Method	GET
REST-URI	/devices
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	-
Response body	<devices> element

Example Header

```
GET /payment/rest/devices HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Content-Type: application/xml
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/xml
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Ok	200	-	<devices> element
Error	500	-	<error> element

Example Response Body (XML)

```
<?xml version=”1.0” encoding=”UTF-8”?>
<pay:devices xmlns:pay="http://gsph.sub.com/payment/types"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pay:device>
      <pay:computerId>2010</pay:computerId>
      <pay:deviceId>801</pay:deviceId>
      <pay:deviceShortName>POS801</pay:deviceShortName>
      <pay:deviceLongName>Exit Cash 801</pay:deviceLongName>
    </pay:device>
    <pay:device>
      <pay:computerId>2010</pay:computerId>
      <pay:deviceId>802</pay:deviceId>
      <pay:deviceShortName>POS802</pay:deviceShortName>
      <pay:deviceLongName>Exit Cash 802</pay:deviceLongName>
    </pay:device>
</pay:devices>
```

3.2.2.5 List Cashiers

List the key information for configured cashiers in the S&B system.

REST-Resource

HTTP-Method	GET
REST-URI	/cashiers
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	-
Response body	<cashiers> element

Example Header

```
GET /payment/rest/cashiers HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Content-Type: application/xml
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/xml
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Ok	200	-	<cashiers> element
Error	500	-	<error> element

Example Response Body (XML)

```
<?xml version=”1.0” encoding=”UTF-8”?>
<pay:cashiers xmlns:pay="http://gsph.sub.com/payment/types"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pay:cashier>
      <pay:cashierContractId>50000</pay:cashierContractId>
      <pay:cashierConsumerId>1</pay:cashierConsumerId>
      <pay:firstName>John</pay:firstName>
      <pay:surname>Deere</pay:surname>
    </pay:cashier>
    <pay:cashier>
      <pay:cashierContractId>50000</pay:cashierContractId>
      <pay:cashierConsumerId>2</pay:cashierConsumerId>
      <pay:firstName>Jack</pay:firstName>
      <pay:surname>Deere</pay:surname>
    </pay:cashier>
</pay:cashiers>
```

3.2.2.6 List Shifts

List the key information for shifts. An optional list of search parameters can be appended to the URI in order to reduce the result list.

The shift list contains key information of the shifts. To get the full information for a shift, the shift resource itself must be requested. The URI is part of the shift key information.

REST-Resource

HTTP-Method	GET
REST-URI	/shifts /cashiers/{cashierContractId},{cashierConsumerId}/shifts
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	-
Response body	<shifts> element

Example Header

```
GET
/payment/rest/shifts?computerId=2010&deviceId=801&cashierContractId=
50000&cashierConsumerId=1 HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Content-Type: application/xml
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/xml
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Ok	200	-	<shifts> element
Error	500	-	<error> element

Example Response Body (XML)

```
<?xml version=”1.0” encoding=”UTF-8”?>
<pay:shifts xmlns:pay="http://gsph.sub.com/payment/types"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pay:shift href="shifts/1295447017">
      <pay:shiftId>1295447017</pay:shiftId>
      <pay:computerId>2010</pay:computerId>
      <pay:deviceId>801</pay:deviceId>
      <pay:cashierContractId>50000</pay:cashierContractId>
      <pay:cashierConsumerId>1</pay:cashierConsumerId>
      <pay:shiftNo>10</pay:shiftNo>
      <pay:createDateTime>2010-12-
        31T14:10:55</pay:createDateTime>
      <pay:shiftStatus>1</pay:shiftStatus>
      <pay:lastSalesTransactionId>0</pay:lastSalesTransactionId>
    </pay:shift>
    <pay:shift href="shifts/1294290035">
      <pay:shiftId>1294290035</pay:shiftId>
      <pay:computerId>2010</pay:computerId>
      <pay:deviceId>801</pay:deviceId>
      <pay:cashierContractId>50000</pay:cashierContractId>
      <pay:cashierConsumerId>1</pay:cashierConsumerId>
      <pay:shiftNo>9</pay:shiftNo>
      <pay:createDateTime>2010-12-
        30T19:10:55</pay:createDateTime>
      <pay:finishDateTime>2010-12-31T03:14:56</pay:finishDateTime>
      <pay:shiftStatus>4</pay:shiftStatus>
      <pay:lastSalesTransactionId>153</pay:lastSalesTransactionId>
    </pay:shift>
</pay:shifts>
```

3.2.2.7 Actual Money Value on Personalized Money Value Card

Retrieves the actual money value on a personalized money value card (PVC).

REST-Resource

HTTP-Method	GET
REST-URI	/media/personalizedMoneyValue/{ContractId},{ConsumerId}
Formats	JSON, XML
Path parameters	-

Query parameters	-
Request body	-
Response body	<cardInfo> element

Example Header

```
GET /payment/rest/media/personalizedMoneyValue/4,10 HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Content-Type: application/xml
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/xml
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Ok	200	-	<shifts> element
Error	500	-	<error> element

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?><cardInfo
xmlns="http://gsph.sub.com/payment/types">
  <epan>02999983010060000400010?????</epan>
  <moneyValue>9500</moneyValue>
</cardInfo>
```

3.2.2.8 Create New Shift

Create a new shift. The shift parameters have to be passed in the request body according to the element <shift> specified in the XML schema *payment.xsd*.

REST-Resource

HTTP-Method	POST
REST-URI	/shifts /cashiers/{cashierContractId},{cashierConsumerId}/shifts
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	<shift> element
Response body	<shift> element

Example Header

```
POST /payment/rest/shifts HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Content-Type: application/xml
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/json
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Ok	201	Location: URI of new Resource	<shift> element
Error	500	-	<error> element

Example Request Body (XML)

```
<pay:shift xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:computerId>2010</pay:computerId>
  <pay:deviceId>801</pay:deviceId>
  <pay:cashierContractId>50000</pay:cashierContractId>
  <pay:cashierConsumerId>1</pay:cashierConsumerId>
  <pay:shiftNo>10</pay:shiftNo>
  <pay:createDateTime>2010-12-31T14:10:55</pay:createDateTime>
</pay:shift>
```

Example Response Body (XML)

```
<pay:shift xmlns:pay="http://gsph.sub.com/payment/types"
        href="shifts/1295447017">
  <pay:shiftId>1295447017</pay:shiftId>
  <pay:computerId>2010</pay:computerId>
  <pay:deviceId>801</pay:deviceId>
  <pay:cashierContractId>50000</pay:cashierContractId>
  <pay:cashierConsumerId>1</pay:cashierConsumerId>
  <pay:shiftNo>10</pay:shiftNo>
  <pay:createDateTime>2010-12-31T14:10:55</pay:createDateTime>
  <pay:shiftStatus>1</pay:shiftStatus>
  <pay:lastSalesTransactionId>0</pay:lastSalesTransactionId>
</pay:shift>
```

3.2.2.9 Close Shift

Close a shift. The shift parameters have to be passed in the request body according to the element <shift> specified in the XML schema *payment.xsd*.

REST-Resource

HTTP-Method	PUT

REST-URI	/shifts/{shiftId} /cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftId}
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	<shift> element
Response body	<shift> element

Example Header

```
PUT /payment/rest/shifts HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Content-Type: application/xml
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/json
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Ok	200	-	<shift> element
Shift with {shiftId} does not exist	404	-	<error> element
Error	500	-	<error> element

Example Request Body (XML)

```
<pay:shift xmlns:pay="http://gsph.sub.com/payment/types">
   <pay:shiftId>1295447017</pay:shiftId>
   <pay:computerId>2010</pay:computerId>
   <pay:deviceId>801</pay:deviceId>
   <pay:cashierContractId>50000</pay:cashierContractId>
   <pay:cashierConsumerId>1</pay:cashierConsumerId>
   <pay:finishDateTime>2010-12-31T21:18:55</pay:finishDateTime>
   <pay:shiftStatus>2</pay:shiftStatus>
</pay:shift>
```

Example Response Body (XML)

```
<pay:shift xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:shiftId>1295447017</pay:shiftId>
  <pay:computerId>2010</pay:computerId>
  <pay:deviceId>801</pay:deviceId>
  <pay:cashierContractId>50000</pay:cashierContractId>
  <pay:cashierConsumerId>1</pay:cashierConsumerId>
  <pay:shiftNo>10</pay:shiftNo>
  <pay:createDateTime>2010-12-31T14:10:55</pay:createDateTime>
  <pay:finishDateTime>2010-12-31T21:18:55</pay:finishDateTime>
  <pay:shiftStatus>2</pay:shiftStatus>
  <pay:lastSalesTransactionId>159</pay:lastSalesTransactionId>
</pay:shift>
```

3.2.2.10 Create Sales Transaction

Create a sales transaction on a shift. Sales transactions are always related to shifts. The sales transaction parameters have to be passed in the request body according to the element <salesTransactionDetail> specified in the XML schema *payment.xsd*.

REST-Resource

HTTP-Method	PUT
REST-URI	/shifts/{shiftId}/salestransactions / cash-iers/{cashierContractId},{cashierConsumerId}/ shifts/{shiftId}/salestransactions
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	<salesTransactionDetail> element
Response body	<salesTransaction> element

Example Header

```
PUT /payment/rest/shifts/1295447017/salestransactions HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Content-Type: application/xml
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/json
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Created	201	Location: URI of new resource	<salesTransaction> ele-ment

Result	HTTP Error Code	HTTP Header	HTTP Body
Shift with {shiftId} does not exist	404	-	<error> element
Error	500	-	<error> element

Example Request Body (XML)

```
<pay:salesTransactionDetail
        xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:salesTransaction>
    <pay:shiftId>1295447017</pay:shiftId>
    <pay:computerId>2010</pay:computerId>
    <pay:deviceId>801</pay:deviceId>
    <pay:cashierContractId>50000</pay:cashierContractId>
    <pay:cashierConsumerId>1</pay:cashierConsumerId>
    <pay:salesTransactionID>1</pay:salesTransactionID>
    <pay:salesTransactionDateTime>2010-12-
        31T14:10:55</pay:salesTransactionDateTime>
  </pay:salesTransaction>
  <pay:articles>
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10100</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>1000</pay:amount>
    </pay:article>
  </pay:articles>
  <pay:payments>
      <pay:artClassRef>1004</pay:artClassRef>
      <pay:articleRef>1300</pay:articleRef>
      <pay:amount>1000</pay:amount>
      <pay:creditClass>0</pay:creditClass>
      <pay:custId1></pay:custId1>
      <pay:custId2></pay:custId2>
  </pay:payments>
</pay:salesTransactionDetail>
```

Example Response Body (XML)

```
<pay:salesTransaction xmlns:pay="http://gsph.sub.com/payment/types"
        href=" salestransactions/1>
  <pay:shiftId>1295447017</pay:shiftId>
  <pay:computerId>2010</pay:computerId>
  <pay:deviceId>801</pay:deviceId>
  <pay:cashierContractId>50000</pay:cashierContractId>
  <pay:cashierConsumerId>1</pay:cashierConsumerId>
  <pay:salesTransactionID>1</pay:salesTransactionID>
</pay:salesTransaction>
```

Example Request Body for paid ticket with EPAN

```
<pay:salesTransactionDetail
        xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:salesTransaction>
    <pay:shiftId>1295447017</pay:shiftId>
    <pay:computerId>2010</pay:computerId>
    <pay:deviceId>801</pay:deviceId>
    <pay:cashierContractId>50000</pay:cashierContractId>
    <pay:cashierConsumerId>1</pay:cashierConsumerId>
    <pay:salesTransactionID>1</pay:salesTransactionID>
    <pay:salesTransactionDateTime>2010-12-
        31T14:10:55</pay:salesTransactionDateTime>
  </pay:salesTransaction>
  <pay:articles>
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10100</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>1000</pay:amount>
      <pay:cardInfos>
        <pay:cardInfo>
          <pay:transType>41</pay:transType>
          <pay:facility>2011</pay:facility>
          <pay:epan>02999983010011011220584770??</pay:epan>
          <pay:cardType>1</pay:cardType>
          <pay:tariffTimeStart>2010-12-
        31T10:11:05</pay:tariffTimeStart>
          <pay:tariffTimeEnd>2010-12-31T14:10:55</pay:tariffTimeEnd>
          <pay:paymentCounter>1</pay:paymentCounter>
        </pay:cardInfo>
    </pay:article>
  </pay:articles>
  <pay:payments>
      <pay:artClassRef>1004</pay:artClassRef>
      <pay:articleRef>1300</pay:articleRef>
      <pay:amount>1000</pay:amount>
      <pay:creditClass>0</pay:creditClass>
      <pay:custId1></pay:custId1>
      <pay:custId2></pay:custId2>
  </pay:payments>
</pay:salesTransactionDetail>
```

Example Response Body

```
<pay:salesTransaction xmlns:pay="http://gsph.sub.com/payment/types"
          href=" salestransactions/1>
  <pay:shiftId>1295447017</pay:shiftId>
  <pay:computerId>2010</pay:computerId>
  <pay:deviceId>801</pay:deviceId>
  <pay:cashierContractId>50000</pay:cashierContractId>
  <pay:cashierConsumerId>1</pay:cashierConsumerId>
  <pay:salesTransactionID>1</pay:salesTransactionID>
</pay:salesTransaction>
```

Example request body for paid and recoded ticket with HashEPAN (External Barcode)

```
<pay:salesTransactionDetail
        xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:salesTransaction>
    <pay:shiftId>1295447017</pay:shiftId>
    <pay:computerId>2010</pay:computerId>
    <pay:deviceId>801</pay:deviceId>
    <pay:cashierContractId>50000</pay:cashierContractId>
    <pay:cashierConsumerId>1</pay:cashierConsumerId>
    <pay:salesTransactionID>1</pay:salesTransactionID>
    <pay:salesTransactionDateTime>2010-12-
        31T14:10:55</pay:salesTransactionDateTime>
  </pay:salesTransaction>
  <pay:articles>
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10100</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>1000</pay:amount>
      <pay:cardInfos>
        <pay:cardInfo>
          <pay:transType>41</pay:transType>
          <pay:facility>2011</pay:facility>
          <pay:cardType>1</pay:cardType>
          <pay:tariffTimeStart>2010-12-
        31T10:11:05</pay:tariffTimeStart>
          <pay:tariffTimeEnd>2010-12-31T14:10:55</pay:tariffTimeEnd>
          <pay:paymentCounter>1</pay:paymentCounter>
         <pay:recodeTicketByTransType>1</pay:recodeTicketByTransType>

        <pay:hashEpan>60085606900110117163719000000019999800000000
        00013011110101000011123123590000000000000000000300000000000
        00000000000000000000000001003195114697</pay:hashEpan>
        </pay:cardInfo>
    </pay:article>
  </pay:articles>
  <pay:payments>
      <pay:artClassRef>1004</pay:artClassRef>
      <pay:articleRef>1300</pay:articleRef>
      <pay:amount>1000</pay:amount>
      <pay:creditClass>0</pay:creditClass>
      <pay:custId1></pay:custId1>
      <pay:custId2></pay:custId2>
  </pay:payments>
</pay:salesTransactionDetail>
```

Example Response Body

```
<pay:salesTransaction xmlns:pay="http://gsph.sub.com/payment/types"
        href=" salestransactions/1>
  <pay:shiftId>1295447017</pay:shiftId>
  <pay:computerId>2010</pay:computerId>
  <pay:deviceId>801</pay:deviceId>
  <pay:cashierContractId>50000</pay:cashierContractId>
  <pay:cashierConsumerId>1</pay:cashierConsumerId>
  <pay:salesTransactionID>1</pay:salesTransactionID>
</pay:salesTransaction>
```

3.2.2.11 Top Up/Down Personalized Money Value Card

Topping Up/Down the money value of a Personalized Money Value Card will be done by creating a sales transaction on a shift. Sales transactions are always related to shifts. The sales transaction parameters have to be passed in the request body according to the element <salesTransactionDetail> specified in the XML schema *payment.xsd*.

REST-Resource

HTTP-Method	PUT
REST-URI	/shifts/{shiftId}/salestransactions / cash- iers/{cashierContractId},{cashierConsumerId}/ shifts/{shiftId}/salestransactions
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	<salesTransactionDetail> element
Response body	<salesTransaction> element

Example Header

```
PUT /payment/rest/shifts/1295447017/salestransactions HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Content-Type: application/xml
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/json
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Created	201	Location: URI of new resource	<salesTransaction> ele- ment
Shift with {shiftId} does not exist	404	-	<error> element

Result	HTTP Error Code	HTTP Header	HTTP Body
Error	500	-	<error> element

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<pay:salesTransactionDetail
        xmlns:pay="http://gsph.sub.com/payment/types">
 <pay:salesTransaction>
  <pay:shiftId>1328516753</pay:shiftId>
  <pay:computerId>2010</pay:computerId>
  <pay:deviceId>701</pay:deviceId>
  <pay:cashierContractId>1</pay:cashierContractId>
  <pay:cashierConsumerId>6</pay:cashierConsumerId>
  <pay:salesTransactionID>1</pay:salesTransactionID>
  <pay:salesTransactionDateTime>2012-02-
        06T13:49:00</pay:salesTransactionDateTime>
 </pay:salesTransaction>
 <pay:articles>
  <pay:article>
   <pay:artClassRef>0</pay:artClassRef>
   <pay:articleRef>10620</pay:articleRef>
   <pay:quantity>1</pay:quantity>
   <pay:quantityExp>0</pay:quantityExp>
   <pay:amount>1000</pay:amount>
   <pay:influenceRevenue>1</pay:influenceRevenue>
   <pay:influenceCashFlow>1</pay:influenceCashFlow>
   <pay:personalizedMoneyValueCard>
            <pay:ContractId>4</pay:ContractId>
            <pay:ConsumerId>2</pay:ConsumerId>
            <pay:addMoneyValue>1000</pay:addMoneyValue>
   </pay:personalizedMoneyValueCard>
  </pay:article>
 </pay:articles>
</pay:salesTransactionDetail>
```

Example Response Body (XML)

```
<pay:salesTransaction xmlns:pay="http://gsph.sub.com/payment/types"
        href=" salestransactions/1>
  <pay:shiftId>1295447017</pay:shiftId>
  <pay:computerId>2010</pay:computerId>
  <pay:deviceId>701</pay:deviceId>
  <pay:cashierContractId>1</pay:cashierContractId>
  <pay:cashierConsumerId>6</pay:cashierConsumerId>
  <pay:salesTransactionID>1</pay:salesTransactionID>
</pay:salesTransaction>
```

3.2.2.12 Get Sales Transaction

Get a sales transaction from a shift. Sales transactions are always related to shifts. The sales transaction parameters have to be passed in the request body according to the element <salesTransaction> specified in the XML schema *payment.xsd*.

REST-Resource

HTTP-Method	GET
REST-URI	/shifts/{shiftId}/salestransactions/{salestransactionId} /cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftId}/salestransactions/{salesTransactionId}
Formats	JSON, XML
Path parameters	-
Query parameters	-
Request body	-
Response body	<salesTransactionDetail> element

Example Header

```
GET /payment/rest/shifts/1295447017/salestransactions/1 HTTP/1.1
Host: gsph.sub.com
Authorisation: SUBWS <SUBWSAccessKeyId>:<Signature>
Content-Type: application/xml
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/json
```

Response

Result	HTTP Error Code	HTTP Header	HTTP Body
Ok	200	-	<salesTransactionDetail> element
Not Found	404	-	-
Error	500	-	<error> element

Example Response Body (XML)

```
<pay:salesTransactionDetail
        xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:salesTransaction>
    <pay:shiftId>1295447017</pay:shiftId>
    <pay:computerId>2010</pay:computerId>
    <pay:deviceId>801</pay:deviceId>
    <pay:cashierContractId>50000</pay:cashierContractId>
    <pay:cashierConsumerId>1</pay:cashierConsumerId>
    <pay:salesTransactionID>1</pay:salesTransactionID>
    <pay:salesTransactionDateTime>2010-12-
        31T14:10:55</pay:salesTransactionDateTime>
  </pay:salesTransaction>
  <pay:articles>
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10100</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>1000</pay:amount>
    </pay:article>
  </pay:articles>
  <pay:payments>
      <pay:artClassRef>1004</pay:artClassRef>
      <pay:articleRef>1300</pay:articleRef>
      <pay:amount>1000</pay:amount>
      <pay:creditClass>0</pay:creditClass>
      <pay:custId1></pay:custId1>
      <pay:custId2></pay:custId2>
  </pay:payments>
</pay:salesTransactionDetail>
```

4 Errors Handling

4.1 External Errors

Error Code	Error	Description	HTTP Status Code
101	Access Denied	Access Denied	403
201	Inappropriate XML	The XML document provided was not appropriate for this operation	400
202	Malformed XML	The XML you provided was not well-formed or did not validate against our published schema	400
300	Request Expired	The request contains a timestamp in its header which has already expired	403

4.2 Internal Errors

All internal errors will be signalled with the HTTP status code 500. More detailed information will be included in the response body. The response will be formatted in XML according to the <error> element in the *payment.xsd*.

Error Code	Short Message	Message
Create Errors		
10001	Dataset not created	The dataset could not be created due to an internal error
10002	Dataset already exists	The dataset could not be created because it exists already
10003	Wrong Sequence	The dataset could not be created because wrong sequence
10004	Shift already created	The requested shift is already created
10005	Shift still open	The requested shift cannot be created because there is still an open shift
10006	Wrong Device – shift not created	The requested shift cannot be created because the device of the shift does not exist
10007	Wrong Cashier – shift not created	The requested shift cannot be created because the cashier of the shift does not exist
10008	Wrong Article – Dataset not created	The dataset could not be created because one of the booked articles does not exist

Error Code	Short Message	Message
10009	Wrong Device - salestransaction not created	The requested salestransaction cannot be created because the device of the shift does not exist
10010	Wrong Cashier - salestransaction not created	The requested salestransaction cannot be created because the cashier of the shift does not exist
10011	Wrong Device - shift not closed	The requested shift cannot be closed because the device of the shift does not exist
10012	Wrong Cashier - shift not closed	The requested shift cannot be closed because the cashier of the shift does not exist
10013	Wrong application id - salestransaction not created	The requested salestransaction cannot be created because the shift is created by foreign process
10014	Wrong application id - shift not closed	The requested shift cannot be closed because the shift is created by foreign process
10015	MediaDynamic dataset not created	The requested MediaDynamic dataset could not be created
10016	MediaDynamicFacility dataset not created	The requested MediaDynamicFacility dataset could not be created
10017	MediaDynamicPresence dataset not created	The requested MediaDynamicPresence dataset could not be created
10018	MediaDynamicWallet dataset not created	The requested MediaDynamicWallet dataset could not be created
…	…	…
Update Errors		
20001	Shift Status not changed	The shift status could not be changed due to an internal error
20002	Shift already closed	The shift is already closed
20003	MediaDynamic dataset not changed	The requested MediaDynamic dataset could not be changed
20004	MediaDynamicFacility dataset not changed	The requested MediaDynamicFacility dataset could not be changed
20005	MediaDynamicPresence dataset not changed	The requested MediaDynamicPresence dataset could not be changed
20006	MediaDynamicWallet dataset not changed	The requested MediaDynamicWallet dataset could not be changed
…	…	…
Input Errors		
30001	Wrong shift cashier	The user is not allowed to book for another cashier

Error Code	Short Message	Message
30002	Shift not found	The shift data could not be found
30003	Wrong shift	The shift request is not valid
30004	Wrong barcode	The data determination for ticket re-coding failed
30005	Transaction data not found	The requested transaction data could not be found
30006	Card data not found	The requested personalized money value card data could not be found
30007	MoneyValueAdd is invalid	The requested MoneyValueAdd value is invalid - the total value range 0 .. 999999 will be violated
…	…	…
Shift Service Errors		
40001	Connection error	No connection to service
40002	Invalid parameters	Some of the URI-Path parameters are invalid
…	…	…

5 Use Cases

The list of parameters together with description and possible values can be found in chapter 6.4 "Appendix 4: List of Parameters".

5.1 Creating a Shift

The following information is required to create a shift:

- ☐ Correct device consisting of: *computerId* and *deviceId*
- ☐ Correct Cashier consisting of: *cashierContractId* and *cashierConsumerId*
- ☐ Correct shift creation date/time: *createDateTime*
- ☐ The *shiftNo* is created and managed by the S&B system

```
########## create shift ##############################
###   POST   /shifts
###   POST   /cashiers/{cashierContractId},{cashierConsumerId}/shifts
######################################################
<?xml version="1.0" encoding="UTF-8"?>
<pay:shift xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:computerId>2010</pay:computerId>
  <pay:deviceId>701</pay:deviceId>
  <pay:cashierContractId>1</pay:cashierContractId>
  <pay:cashierConsumerId>6</pay:cashierConsumerId>
  <pay:shiftNo>1</pay:shiftNo>
  <pay:createDateTime>2012-02-06T07:00:00</pay:createDateTime>
</pay:shift>
```

5.2 Booking Sales Transactions

The following basic information is required to book a sales transaction:

- ☐ Correct internal shift ID (*shiftId*) which will be created at shift creation by the S&B system
- ☐ Correct device consisting of: *computerId* and *deviceId*
- ☐ Correct Cashier consisting of: *cashierContractId* and *cashierConsumerId*
- ☐ Sales Transaction ID (*salesTransactionID*) this is a sequence number inside the running shift which has to be tracked by the booking client
- ☐ Correct sales transaction date/time: *salesTransactionDateTime*

5.2.1 Simple Sales Transaction

The following example shows a simple sales transaction which books an article (artClassRef=0/articleRef=10100).

Article definitions in the S&B system are documented in the document "ArticleAllocation_(1.3)_EN_20120113.doc".

The S&B settlement module interprets this transaction as cash payment, because no payment is included.

```
########## create sales transaction ####################
###   PUT   /shifts/{shiftId}/salestransactions
###   PUT
/cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftid}/salestra
nsactions
##############################################
<?xml version="1.0" encoding="UTF-8"?>
<pay:salesTransactionDetail
        xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:salesTransaction>
    <pay:shiftId>1328516753</pay:shiftId>
    <pay:computerId>2010</pay:computerId>
    <pay:deviceId>701</pay:deviceId>
    <pay:cashierContractId>1</pay:cashierContractId>
    <pay:cashierConsumerId>6</pay:cashierConsumerId>
    <pay:salesTransactionID>1</pay:salesTransactionID>
    <pay:salesTransactionDateTime>2012-02-
        06T13:49:00</pay:salesTransactionDateTime>
  </pay:salesTransaction>
  <pay:articles>
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10100</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>100</pay:amount>
    </pay:article>
  </pay:articles>
</pay:salesTransactionDetail>
```

5.2.2 Sales Transaction with discount

The following example shows a sales transaction which books two articles, (artClassRef=0/articleRef=10100) and (artClassRef=0/articleRef=10101). Additional there is a discount on the sale (artClassRef=901/articleRef=13153). Discounts in S&B system are organized in the Article Class (artClassRef=901).

Article definitions in the S&B system are documented in the document "ArticleAllocation_(1.3)_EN_20120113.doc".

The difference between sale and discount will be interpreted as cash payment, because no payment is included.

```
########## sale with discounts ######################
###   PUT   /shifts/{shiftId}/salestransactions
###   PUT
/cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftid}/salestra
nsactions
##############################################
<?xml version="1.0" encoding="UTF-8"?>
<pay:salesTransactionDetail
        xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:salesTransaction>
    <pay:shiftId>1328516753</pay:shiftId>
    <pay:computerId>2010</pay:computerId>
    <pay:deviceId>701</pay:deviceId>
    <pay:cashierContractId>1</pay:cashierContractId>
    <pay:cashierConsumerId>6</pay:cashierConsumerId>
    <pay:salesTransactionID>2</pay:salesTransactionID>
    <pay:salesTransactionDateTime>2012-02-
        06T13:49:00</pay:salesTransactionDateTime>
  </pay:salesTransaction>
  <pay:articles>
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10100</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>100</pay:amount>
    </pay:article>
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10101</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>120</pay:amount>
    </pay:article>
  </pay:articles>
  <pay:discounts>
    <pay:discount>
      <pay:artClassRef>901</pay:artClassRef>
      <pay:articleRef>13153</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:amount>15</pay:amount>
      </pay:discount>
  </pay:discounts>
</pay:salesTransactionDetail>
```

5.2.3 Sales Transaction with discount and with ticket data

```
##### sale with discount and card info (epan) without recode ######
###   PUT   /shifts/{shiftId}/salestransactions
###   PUT
/cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftid}/salestra
nsactions
#############################################
<?xml version="1.0" encoding="UTF-8"?>
<pay:salesTransactionDetail
        xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:salesTransaction>
    <pay:shiftId>1328516753</pay:shiftId>
    <pay:computerId>2010</pay:computerId>
    <pay:deviceId>701</pay:deviceId>
    <pay:cashierContractId>1</pay:cashierContractId>
    <pay:cashierConsumerId>6</pay:cashierConsumerId>
    <pay:salesTransactionID>3</pay:salesTransactionID>
    <pay:salesTransactionDateTime>2012-02-
        06T13:50:00</pay:salesTransactionDateTime>
  </pay:salesTransaction>
  <pay:articles>
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10100</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>100</pay:amount>
      <pay:influenceRevenue>1</pay:influenceRevenue>
      <pay:influenceCashFlow>1</pay:influenceCashFlow>
      <pay:cardInfos>
        <pay:cardInfo>
          <pay:transType>41</pay:transType>
          <!-- transType -> SR_ART_EXT_PARK_2.wTranstyp -->
          <pay:transMark>0</pay:transMark>
          <!-- transMark -> SR_ART_EXT_PARK_2.wTransMark -->
          <pay:facility>3010</pay:facility>
          <!-- facility -> SR_ART_EXT_PARK.wAnwAnl -->
          <pay:epan>02999983010011010341313330??</pay:epan>
          <!-- epan -> SR_ART_EXT_PARK.tEPAN -->
          <pay:cardType>1</pay:cardType>
          <!-- cardType -> SR_ART_MAGNET.wKarttyp -->
          <pay:cardSubType>0</pay:cardSubType>
          <!-- cardSubType -> SR_ART_MAGNET.bUnttyp -->
          <pay:cardClass>0</pay:cardClass>
```

```
            <!-- cardClass -> SR_ART_MAGNET.wGatt -->
            <pay:tariffTimeStart>2012-02-
          06T10:50:00</pay:tariffTimeStart>
            <!-- tariffTimeStart -> SR_ART_EXT_PARK.tIn -->
            <pay:tariffTimeEnd>2012-02-06T13:50:00</pay:tariffTimeEnd>
            <!-- tariffTimeEnd -> SR_ART_EXT_PARK.tOut -->
            <pay:moneyValue>0</pay:moneyValue>
            <!-- moneyValue -> SR_ART_MAGNET.zGguthm -->
            <pay:pointValue>0</pay:pointValue>
            <!-- pointValue -> SR_ART_MAGNET.wPguth -->
            <pay:timeValue>0</pay:timeValue>
            <!-- timeValue -> SR_ART_MAGNET.tZguth -->
            <pay:paymentCounter>1</pay:paymentCounter>
            <!-- paymentCounter -> SR_ART_MAGNET.bBezhau -->
            <pay:wkDayProgRef>0</pay:wkDayProgRef>
            <!-- wkDayProgRef -> SR_ART_MAGNET.bWoche -->
            <pay:meetingNumber>0</pay:meetingNumber>
            <!-- meetingNumber -> SR_ART_MAGNET.wVeranst -->

          <pay:recodeTicketByTransType>0</pay:recodeTicketByTransType
          >
            <!-- recodeTicketByTransType ->
          SR_ART_EXT_PARK_RECODE_HEPAN.bRecodeTicketByTransType --
          >
            <pay:hashEpan></pay:hashEpan>
            <!-- hashEpan ->
          SR_ART_EXT_PARK_RECODE_HEPAN.cHashEpan  -->
          </pay:cardInfo>
        </pay:cardInfos>
      </pay:article>
    </pay:articles>
    <pay:discounts>
      <pay:discount>
        <pay:artClassRef>901</pay:artClassRef>
        <pay:articleRef>13153</pay:articleRef>
        <pay:quantity>1</pay:quantity>
        <pay:amount>15</pay:amount>
        </pay:discount>
    </pay:discounts>
</pay:salesTransactionDetail>
```

5.2.4 Sales Transaction with discount and with credit card payment

```
### sale with discount and card info (epan) and credit card payment ###
###   PUT   /shifts/{shiftId}/salestransactions
###   PUT
/cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftid}/salestra
nsactions
##########################################################
<?xml version="1.0" encoding="UTF-8"?>
<pay:salesTransactionDetail
        xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:salesTransaction>
    <pay:shiftId>1328516753</pay:shiftId>
    <pay:computerId>2010</pay:computerId>
    <pay:deviceId>701</pay:deviceId>
    <pay:cashierContractId>1</pay:cashierContractId>
    <pay:cashierConsumerId>6</pay:cashierConsumerId>
    <pay:salesTransactionID>4</pay:salesTransactionID>
    <pay:salesTransactionDateTime>2012-02-
        06T13:52:00</pay:salesTransactionDateTime>
  </pay:salesTransaction>
  <pay:articles>
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10100</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>100</pay:amount>
      <pay:influenceRevenue>1</pay:influenceRevenue>
      <pay:influenceCashFlow>1</pay:influenceCashFlow>
      <pay:cardInfos>
        <pay:cardInfo>
          <pay:transType>41</pay:transType>
          <!-- transType -> SR_ART_EXT_PARK_2.wTranstyp -->
          <pay:transMark>0</pay:transMark>
          <!-- transMark -> SR_ART_EXT_PARK_2.wTransMark -->
          <pay:facility>3010</pay:facility>
          <!-- facility -> SR_ART_EXT_PARK.wAnwAnl -->
          <pay:epan>02999983010011010341313330??</pay:epan>
          <!-- epan -> SR_ART_EXT_PARK.tEPAN -->
          <pay:cardType>1</pay:cardType>
          <!-- cardType -> SR_ART_MAGNET.wKarttyp -->
          <pay:cardSubType>0</pay:cardSubType>
          <!-- cardSubType -> SR_ART_MAGNET.bUnttyp -->
```

```
          <pay:cardClass>0</pay:cardClass>
          <!-- cardClass -> SR_ART_MAGNET.wGatt -->
          <pay:tariffTimeStart>2012-02-
        06T10:52:00</pay:tariffTimeStart>
          <!-- tariffTimeStart -> SR_ART_EXT_PARK.tIn -->
          <pay:tariffTimeEnd>2012-02-06T13:52:00</pay:tariffTimeEnd>
          <!-- tariffTimeEnd -> SR_ART_EXT_PARK.tOut -->
          <pay:moneyValue>0</pay:moneyValue>
          <!-- moneyValue -> SR_ART_MAGNET.zGguthm -->
          <pay:pointValue>0</pay:pointValue>
          <!-- pointValue -> SR_ART_MAGNET.wPguth -->
          <pay:timeValue>0</pay:timeValue>
          <!-- timeValue -> SR_ART_MAGNET.tZguth -->
          <pay:paymentCounter>1</pay:paymentCounter>
          <!-- paymentCounter -> SR_ART_MAGNET.bBezhau -->
          <pay:wkDayProgRef>0</pay:wkDayProgRef>
          <!-- wkDayProgRef -> SR_ART_MAGNET.bWoche -->
          <pay:meetingNumber>0</pay:meetingNumber>
          <!-- meetingNumber -> SR_ART_MAGNET.wVeranst -->
         <pay:recodeTicketByTransType>0</pay:recodeTicketByTransType>
         <!-- recodeTicketByTransType ->
        SR_ART_EXT_PARK_RECODE_HEPAN.bRecodeTicketByTransType -->
          <pay:hashEpan></pay:hashEpan>
          <!-- hashEpan ->
        SR_ART_EXT_PARK_RECODE_HEPAN.cHashEpan  -->
        </pay:cardInfo>
      </pay:cardInfos>
    </pay:article>
  </pay:articles>
  <pay:discounts>
    <pay:discount>
      <pay:artClassRef>901</pay:artClassRef>
      <pay:articleRef>13153</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:amount>15</pay:amount>
    </pay:discount>
  </pay:discounts>
  <pay:payments>
    <pay:payment>
      <pay:artClassRef>1003</pay:artClassRef>
      <pay:articleRef>1259</pay:articleRef>
      <pay:amount>100</pay:amount>
      <pay:creditClass>259</pay:creditClass>
      <pay:custId1>01445301010101010101010101??</pay:custId1>
      <pay:custId2>01445302020202020202020202??</pay:custId2>
      <pay:expireDate>2015-01-01T00:00:00</pay:expireDate>
      <pay:AIDDate>2011-07-04T13:56:00</pay:AIDDate>
```

```
        <!-- authNumber in version 1.5.0 -->
        <pay:authNumber>01234567890123456789</pay:authNumber>
      </pay:payment>
    </pay:payments>
</pay:salesTransactionDetail>
```

5.2.5 Extended validation with discount

```
############ sale with discounts and validation ############
###   PUT   /shifts/{shiftId}/salestransactions
###   PUT
/cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftid}/salestra
nsactions
##########################################################
<?xml version="1.0" encoding="UTF-8"?>
<pay:salesTransactionDetail
   xmlns:pay="http://gsph.sub.com/payment/types">
   <pay:salesTransaction>
      <pay:shiftId>1328516753</pay:shiftId>
      <pay:computerId>2010</pay:computerId>
      <pay:deviceId>701</pay:deviceId>
      <pay:cashierContractId>1</pay:cashierContractId>
      <pay:cashierConsumerId>6</pay:cashierConsumerId>
      <pay:salesTransactionID>2</pay:salesTransactionID>
      <pay:salesTransactionDateTime>2018-07-
         06T13:49:00</pay:salesTransactionDateTime>
   </pay:salesTransaction>
   <pay:articles>
      <pay:article>
         <pay:artClassRef>0</pay:artClassRef>
         <pay:articleRef>10100</pay:articleRef>
         <pay:quantity>1</pay:quantity>
         <pay:quantityExp>0</pay:quantityExp>
         <pay:amount>100</pay:amount>
      </pay:article>
      <pay:article>
         <pay:artClassRef>0</pay:artClassRef>
         <pay:articleRef>10101</pay:articleRef>
         <pay:quantity>1</pay:quantity>
         <pay:quantityExp>0</pay:quantityExp>
         <pay:amount>120</pay:amount>
      </pay:article>
   </pay:articles>
   <pay:discounts>
      <pay:discount>
         <pay:artClassRef>901</pay:artClassRef>
         <pay:articleRef>701</pay:articleRef>
         <pay:quantity>1</pay:quantity>
         <pay:amount>15</pay:amount>
         <pay:validation>
            <pay:validationClassification>1</pay:validationClassification>
            <pay:sequence>1</pay:sequence>
            <pay:validationTime>2018-07-
```

```
                05T13:50:00</pay:validationTime>
            <pay:validationFacility>2011</pay:validationFacility>
            <pay:class>1</pay:class>
            <pay:percentValue>500</pay:percentValue>
            <pay:timeValue>86000</pay:timeValue>
            <pay:moneyValue>500</pay:moneyValue>
            <pay:distributorID>1001</pay:distributorID>
            <pay:customerRef>1003</pay:customerRef>
            <pay:discount>0</pay:discount>
            <pay:validatorID>10</pay:validatorID>
            <pay:validationID>UUID</pay:validationID>
            <pay:maxValParkDuration>700</pay:maxValParkDuration>
            <pay:maxValParkFeeAmnt>86000</pay:maxValParkFeeAmnt>
            <pay:timeValueRemaining>0</pay:timeValueRemaining>
            <pay:moneyValueRemaining>0</pay:moneyValueRemaining>
            <pay:receiptText>any text</pay:receiptText>
         </pay:validation>
      </pay:discount>
   </pay:discounts>
</pay:salesTransactionDetail>
```

5.2.6 Extended validation with article

```
########## create sales transaction ####################
###   PUT   /shifts/{shiftId}/salestransactions
###   PUT
/cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftid}/salestra
nsactions
##############################################
<?xml version="1.0" encoding="UTF-8"?>
<pay:salesTransactionDetail
   xmlns:pay="http://gsph.sub.com/payment/types">
   <pay:salesTransaction>
      <pay:shiftId>1328516753</pay:shiftId>
      <pay:computerId>2010</pay:computerId>
      <pay:deviceId>701</pay:deviceId>
      <pay:cashierContractId>1</pay:cashierContractId>
      <pay:cashierConsumerId>6</pay:cashierConsumerId>
      <pay:salesTransactionID>1</pay:salesTransactionID>
      <pay:salesTransactionDateTime>2018-07-
         06T13:49:00</pay:salesTransactionDateTime>
   </pay:salesTransaction>
   <pay:articles>
      <pay:article>
         <pay:artClassRef>0</pay:artClassRef>
         <pay:articleRef>10100</pay:articleRef>
         <pay:quantity>1</pay:quantity>
         <pay:quantityExp>0</pay:quantityExp>
         <pay:amount>100</pay:amount>
      </pay:article>
      <pay:article>
         <pay:artClassRef>901</pay:artClassRef>
         <pay:articleRef>701</pay:articleRef>
         <pay:quantity>1</pay:quantity>
         <pay:quantityExp>0</pay:quantityExp>
         <pay:amount>120</pay:amount>
         <pay:validations>
            <pay:validation>
               <pay:validationClassification>1</pay:validationClassification>
               <pay:sequence>1</pay:sequence>
               <pay:validationTime>2018-07-
                  05T13:50:00</pay:validationTime>
               <pay:validationFacility>2011</pay:validationFacility>
               <pay:class>1</pay:class>
               <pay:percentValue>500</pay:percentValue>
               <pay:timeValue>86000</pay:timeValue>
               <pay:moneyValue>500</pay:moneyValue>
               <pay:distributorID>1001</pay:distributorID>
```

```
                <pay:customerRef>1003</pay:customerRef>
                <pay:discount>0</pay:discount>
                <pay:validatorID>10</pay:validatorID>
                <pay:validationID>UUID</pay:validationID>
                <pay:maxValParkDuration>700</pay:maxValParkDuration>

<pay:maxValParkFeeAmnt>86000</pay:maxValParkFeeAmnt>
                <pay:timeValueRemaining>0</pay:timeValueRemaining>
                <pay:moneyValueRemaining>0</pay:moneyValueRemaining>
                <pay:receiptText>any text</pay:receiptText>
            </pay:validation>
         </pay:validations>
      </pay:article>
   </pay:articles>
</pay:salesTransactionDetail>
```

5.3 Closing a Shift

The following information is required to close a shift:

- ☐ Correct internal shift ID (*shiftId*) which will be created at shift creation by the S&B system
- ☐ Correct device consisting of: *computerId* and *deviceId*
- ☐ Correct Cashier consisting of: *cashierContractId* and *cashierConsumerId*
- ☐ Correct shift closing date/time: *finishDateTime*

```
########## close shift ##############################
###   PUT   /shifts/{shiftId}
###   PUT
/cashiers/{cashierContractId},{cashierConsumerId}/shifts/{shiftId}
#####################################################
<?xml version="1.0" encoding="UTF-8"?>
<pay:shift xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:shiftId>1328516753</pay:shiftId>
  <pay:computerId>2010</pay:computerId>
  <pay:deviceId>701</pay:deviceId>
  <pay:cashierContractId>1</pay:cashierContractId>
  <pay:cashierConsumerId>6</pay:cashierConsumerId>
  <pay:finishDateTime>2012-02-06T14:00:00</pay:finishDateTime>
  <pay:shiftStatus>2</pay:shiftStatus>
</pay:shift>
```

5.4 Booking Sales Transaction with FastPayment

Can be used in the same way as booking sales transaction with following rules:

1. Generated UUID is needed from client, or with resource “getRandomUUID”.
2. The naming of objects are different, i.e. “salesTransactionDetail” must be renamed to “systemSalesTransactionDetail”.
3. With FastPayment values as “shiftId,contractId,consumerId,transactionId” are not needed.
4. To retrieve the transaction details, only uuid is needed.

5.4.1 FastPayment UUID generating

For the new transaction a new UUID is needed. For generating can use client his own generator or use this resource, with calling the request:

```
########## create sales transaction ####################
###   GET   /uuid/
#####################################################
```

The example response:

```
<?xml version="1.0" encoding="UTF-8"?><salesUUID
xmlns="http://gsph.sub.com/payment/types">
  <uuid>02c84d68-5a0a-45e0-9473-3f76f3804381</uuid>
</salesUUID>
```

5.4.2 FastPayment sales transaction

Can be used in the same way as booking sales transaction with articles, discounts, etc.

The following example shows a simple sales transaction which books an article (artClassRef=0/articleRef=10100).

Article definitions in the S&B system are documented in the document "ArticleAllocation_(1.3)_EN_20120113.doc".

The S&B settlement module interprets this transaction as cash payment, because no payment is included.

```
########## create sales transaction ####################
###   PUT
/uuid/salestransactions/{transactionUUID}/createSystemSalesTransaction
##############################################
<?xml version="1.0" encoding="UTF-8"?>
<pay:uuidSalesTransactionDetail
xmlns:pay="http://gsph.sub.com/payment/types">
  <pay:articles>
    <pay:article>
      <pay:artClassRef>0</pay:artClassRef>
      <pay:articleRef>10100</pay:articleRef>
      <pay:quantity>1</pay:quantity>
      <pay:quantityExp>0</pay:quantityExp>
      <pay:amount>100</pay:amount>
</pay:article>
  </pay:articles>
</pay:uuidSalesTransactionDetail>
```

5.4.3 FastPayment – get transaction details

To retrieve the transaction details, only uuid is needed with calling following resource.

```
########## create sales transaction ####################
###   GET   /uuid/salestransactions/{transactionUUID}
##############################################
```

5.4.4 FastPayment – get transaction status

For information, if transaction was finished or unfinished, use following resource.

```
########## create sales transaction ####################
###   GET   /uuid/salestransactions/{transactionUUID}/status
#########################################################
```

6 Appendices

6.1 Appendix 1: List of Interface Versions and Related Functionalities

Interface Version	Functionality
1.1.0	REST Payment Interface
1.2.3	Extended log output Bugfix: ☐ Booking with URI "/cashiers/{cashierContractId}, {cashierConsumerId}/shifts/{shiftid}/salestransactions" not working ☐ Syntax error in @PathParams of URI
1.3.1	Bugfix: ☐ Problem with creating new shift by URI "/cashiers/{cashierContractId}, {cashierConsumerId}/shifts/" <message>. The user is not allowed to book for another cashier</message>.
1.4.0	Extension of REST Payment Interface for booking into InvTransData for invoicing purpose. Dedicated articles with enabled flag "Must be invoiced" in Artbase can be booked by REST Payment Interface for invoicing purpose.
1.5.0	Extension of REST Payment Interface for booking an external Authorization number into entervo system with: <pay:authNumber>01234567890123456789012345678901234567890123456789</pay:authNumber>
1.7.1	Support of Top up/down personalized money value cards (PVC) (FE 7940) and MediaDynamic support (FE 8285)

6.2 Appendix 2: Transaction Types

Transactions that have led to pass-through, production or payment:

Code	Description
1	Entry reserved spaces
2	Entry non reserved spaces
3	Entry manual open
4	Entry locked
5	Entry broken barrier
6	Entry illegal exit
11	Exit reserved spaces

Code	Description
12	Exit non reserved spaces
13	Exit manual open
14	Exit locked
15	Exit broken barrier
16	Exit illegal entry
18	Exit within grace period
20	Credit card payment at exit
21	Pass through reserved / reserved
22	Pass through non reserved / non reserved
23	Pass through reserved / non reserved
24	Pass through non reserved / reserved
25	Pass through manual open
26	Pass through locked
27	Pass through broken barrier
28	Door passing
29	Door manual open
30	Door locked
31	Level counting
32	Each barrier opening
41	POS (point of sale) card production
42	POS payment transaction
43	POS extra payment
44	POS reloading
45	POS change transaction
46	POS season ticket production with the new card coding
47	Payment is from previous day (For internal use only!)
48	Change expiration time (PKA + 1 month)

Transactions that have not led to pass-through, production or payment:

Code	Description
51	Backout ticket created
52	Backout ticket used
53	Backout ticket switched to regular ticket
61	Grace time exceeded (extra payment required)
62	Correction button used (automatic pay station)
63	Sales transaction cancelled
71	Production error

Code	Description
72	Evaluation error
73	Change error
81	Blocking manually
82	Blocking by supervisor
83	Blocking by difference counting
84	Blocking by wrong facility
85	Blocking by weekly program
86	Blocking by card expired
87	Blocking by card production counter
88	Blocking by anything else
89	Blocking by anything else
90	Blocking by anything else
91	Entry manually entered
92	Exit manually entered
93	Credit card entry without entering in TTD
94	Credit card exit without deleting from TTD
95	Backout created without entering in TTD
96	Backout switched without deleting from TTD

6.3 Appendix 3: Card Types and Subtypes

Card Type	Sub-type	Description	Production Article
1	0	parking ticket	100
1	1	prepaid ticket	101
1	2	free ticket	102
1	3	exit ticket	103
1	4	unreadable ticket	104
1	6	lost ticket at automatic pay station	106
1	7	lost ticket at manual sales device	107
2	0	contract parker card	200
3	0	congress card	300
4	0	parking cheque	400
5	0	congress cheque	500
6	0	money value card without control function	600
6	0	personalized money value card without control function	620
6	1	time value card without control function	601

Card Type	Sub-type	Description	Production Article
6	2	percentage discounts card – multiple	602
6	3	company invoice card: 3 (credit): money cheque 3 (class): tariff cheque (tariff change card) 3 (time): time cheque	 603 610 611
6	4	money value card with control function	604
6	4	personalized money value card with control function	624
6	5	time value card with control function	605
6	7	money voucher	607
6	8	time voucher	608
6	9	percentage discounts card – once	609
7	0	meeting card	
8	0	staff card	
9	0	refund card	
10	0	provider card	
90	0	service card	

6.4 Appendix 4: List of Parameters

Parameter	Short description	Format	Possible values
addMoney-Value	amount to add on the money value of the card; amount in smallest currency unit	number	(default value is 0)
AIDDate	Authorisation date and time of cash-less payment	YYYY-MM-DDTHH:MM:SS	
amount	Total amount of sales article	number	(default value is 0)
amount	Total amount of discount article	number	(default value is 0)
amount	Total amount of payment article	number	(default value is 0)

Parameter	Short description	Format	Possible values
artClassRef	Article class ID reference	number	sales: 0: sales articles 1: special income/outcome 10: entry transactions 11: exit transactions discounts: 901: discount articles payments: 1003: credit articles 1004: cash payment and cash contents articles
articleRef	Article ID reference	number	e.g. 101, 251, 12605, 20705000, … (to get the list of articles, follow the steps in chapter 3.2.2.3)
articleShortcleShort-Name	Article short name	string [18]	
authNumber	External Authorization number (e.g. PayPal) (version 1.5.0)	string [50]	
cardClass	Specific card class	number	0-999 (default value is 0)
cardSub-Type	S&B card subtype	number	(see chapter 6.3 "Appendix 3: Card Types and Subtypes")
cardType	S&B card type	number	(see chapter 6.3 "Appendix 3: Card Types and Subtypes")
cash-ierConsum-erId	Cashier Consumer ID	number	1-99999
cashierCon-tractId	Cashier contract ID	number	1-99999
class	Cardclass for Traiff switch	number	(default value is 0)
computerId	Computer ID	number	1-9999 ZR: 2000-8999 LR: 9000-9999
Consum-erId	consumer ID of personalized money value card	number	1-99999

Parameter	Short description	Format	Possible values
ContractId	contract ID of personalized money value card	number	1-99999
create-ate-DateTime	Shift create date and time	YYYY-MM-DDTHH:MM:SS	
creditClass	Internal credit or cashless payment type identifier	number	(default value is 0)
custId1	Track 1 information	string [28]	
custId2	Track 2 information	string [28]	
customer-Ref	Customer reference	number	(default value is 0)
deviceId	Device ID	number	1-999 101-199: entry 201-299: exit 301-399: pass through 401-499: door reader 501-599: validator 601-699: PKA/PoF 701-799: hand cashier 801-899: exit cashier
device-LongName	Device long name	string [25]	
device-ShortName	Device short name	string [5]	
discount	Discount Flag	number	(default value is 0)
distribu-torID	Distributor ID	number	
epan	Unique ticket number (Extended Personal Account Number)	string [28]	e.g. 0299998301001101122058 4770??
expireDate	Expiry date of credit card	YYYY-MM-DDTHH:MM:SS	
facility	ID of entered facility	number	0-9999 e.g. 2010, 3010, …

Parameter	Short description	Format	Possible values
finishDateTime	Shift finish date and time	YYYY-MM-DDTHH:MM:SS	
firstName	First name	string [33]	
hashEpan	Unique ticket number for external barcodes	string [250]	
influenceCashFlow	Indicates that the article has influence on cash flow / cash contents.	boolean	(default value is "true")
influenceRevenue	Indicates that the article has influence on revenue.	boolean	(default value is "true")
lastSalesTransactionId	ID of the last booked sales transaction	number	
maxValParkDuration	Maximum Parking Duration (validation limitation), which will be validated (used for full validation) (in seconds)	number	(default value is 0)
maxValParkParkFeeAmnt	Maximum Parking Fee Amount (validation limitation), which will be validated (used for full validation)	number	(default value is 0)
meetingNumber	Specific meeting number	number	(default value is 0)
moneyValue	Remaining stored money value on a money value card (in smallest currency unit)	number	(default value is 0)
moneyValueRemaining	Remaning money value (smallest denomination)	number	(default value is 0)
paymentCounter	Number of payments of particular ticket	number	
percentValue	Percent Value	number	(default value is 0)

Parameter	Short description	Format	Possible values
pointValue	Remaining stored point value on a point value card	number	(default value is 0)
quantity	Quantity of sales articles or quantity of discount articles	number	(default value is 1)
quantityExp	Quantity exponent	number	(default value is 0)
receiptText	Transparent receipt text	string [44]	
recodeTicketByTransType	Recode ticket in the S&B system related to transaction type	boolean	(default value is "true")
salesTransactionDateTime	Date and time of sales transaction occurrence	YYYY-MM-DDTHH:MM:SS	
salesTransactionID	Sales transaction ID	number	
sequence	Unique wallet id	number	
shiftId	Shift ID	number	e.g. 1294290035, 1295447017, …
shiftNo	Shift number	number	e.g. 5, 9, 10, …
shiftStatus	Shift status	number	1: open 2: closed (interim status) 3: in processing (interim status) 4: finished and settled 5: logically deleted
surname	Surname	string [33]	
tariffTimeEnd	Time frame end used for tariff calculation	YYYY-MM-DDTHH:MM:SS	
tariffTimeStart	Time frame start used for tariff calculation	YYYY-MM-DDTHH:MM:SS	
timeValue	Remaining stored time value on a time value card (in seconds)	number	(default value is 0)
timeValueRemaining	Remaining time value (in seconds)	number	(default value is 0)

Parameter	Short description	Format	Possible values
transac-tionUUID	UUID of transaction	UUID	Only UUID format
transMark	Specific transaction marker	number	0-99 (default value is 0) 0x00: normal transaction 0x01: tolerated negative transaction 0x02: transaction manually created (at cell computer) 0x03: wrong presence tol-erated 0x13: wrong presence tol-erated (at entry) 0x10: transaction not sent to TTD 0x20: do not distribute transaction 0x80: transaction not stored in TTD
transType	Transaction type	number	(see chapter 6.2 "Appendix 2: Transaction Types")
validation-Classifica-tion	Wallet type	number	1= money value 2= time value 3= point value 4= percentage value 5= tariff data switch 6= facture by customer 7 = full validation 8 = change of tariff relevant time 9 = time list for tiered park-ing
validation-Facility	Facility for Valida-tion	number	(default value is 0)
valida-tionID	Unique Validation ID (UUID)	string [36]	
validation-Time	Time of validation	YYYY-MM-DDTHH:MM:SS	
validatorID	Preshared validator ID	number	

Parameter	Short description	Format	Possible values
wkDayProgRef	Specific week day profile for access card holders. It is a preconfigured number in the master data "Weekly Profile".	number	(default value is 0)

7 Record of Revisions

Revision	Date	Description	Author
-	15/02/2011	Creation	K. Schroeren
1.1	15/04/2011	Deleted Authentication by HMAC	K. Schroeren
1.2	22/07/2011	Changed Authentication Changed Versioning Changed XSD Version to v00.02	K. Schroeren
1.3	02/09/2011	Extended "Create Sales Transaction" to recode tickets as paid in the system and to except external barcodes by defining a new type "HashEpan" Extended XSD for recoding tickets in the S&B system and external barcodes "HashEpan" (Version v00.03)	K. Schroeren
1.4	09/09/2011	Deleted Request body in chapter 3.2.2.11 "Get Sales Transaction"	K. Schroeren
1.5	06/02/2012	Minor changes Attachment – Examples	K. Schroeren
1.6	05/08/2015	Document adapted for the new template Chapter 1.1 "Interface Overview" added Examples moved to chapter 5 "Use Cases" Chapter 6.1 "Appendix 1: List of Interface Versions and Related Functionalities" added Chapter 6.2 "Appendix 2: Transaction Types" added Chapter 6.3 "Appendix 3: Card Types and Subtypes" added Chapter 6.4 "Appendix 4: List of Parameters" added	J. Srponova
1.7	06/08/2015	New interface version 1.5.0 supports now external Authorization numbers	K. Schroeren
1.8	02/12/2016	New interface version 1.7.1 Support of Top up/down personalized money value cards (PVC) (FE 7940) and MediaDynamic support (FE 8285)	K. Schroeren

1.9	26/8/2016	New interface version 1.9.0 Chapter 5.4 - FastPayment	M. Hostak
1.10	13/11/2018	New interface version 1.13.0 Extended validation support - Chapters 5.2.5, 5.2.6	M. Hostak