Universal Interface

Customer and Media Web Service

This document contains strictly confidential information and may not be circulated or copied without prior permission of S&B!

This page is intentionally left blank.

Author: Peter Kubánek

Modification log:
Refer to chapter "Record of Revisions"

Since all information and facts are subject to technical changes, any liability for the data contained is hereby disclaimed. Modifications of technical details, in terms of information and illustrations are reserved. Make sure to follow the updating index.

Scheidt & Bachmann GmbH cannot be made responsible for direct damage and/or possible consequential damage due to maloperation of the product by the customer or by third parties, unless the Product Liability Act (ProdHaftG) is concerned.

© 2020 Scheidt & Bachmann GmbH, Germany
Parking and leisure centre systems
Breite Straße 132
41238 Mönchengladbach
www.scheidt-bachmann.com

Subject to change.

This page is intentionally left blank.

Contents

20201023UI0504

A

1 Introduction

> *This document replaces the "RESTCustomerMediaService-resources_user_*_*" document.*
> *The asterisk (*) stands for the version identification.*

In this document, the resources for REST *Customer and Media Web Service* interface are described together with input and output parameters:

- Http method
- Path
- RolesAllowed
- XSD element for input data
- Path parameter
- Query parameter
- XSD element for response data
- Method name in *CustomerMediaWebService*

This documentation is valid for the Interface version 1.23.0@0.

1.1 Interface Overview

The *Customer and Media Web Service* interface enables our client to set up a website for their customers and transfer tedious and time-consuming tasks from the back-office team to the customers.

The Customer and Media Web Service interface provides the following features:

- Administration of companies
  - Create company
  - Update company information
  - Request and receive a list of all available companies or a specific company
  - Request and receive details about a specific company (contract details)
  - Request and receive a list of all card holders (participants) for a specific company
  - Request and receive account information about a specific company
  - Request and receive a list of all sales transactions for a specific company
  - Request and receive, update or delete company`s pooling data
- Comprehensive administration of card holders (participants)
  - Create participant
  - Delete participant
  - Request and receive a list of all available card holders or a specific card holder
  - Request and receive detailed card holder information
- Solid invoicing administration and control

  - Request and receive a list of all invoices for a specific company (response includes invoice number, invoice amount and invoice date)

- □ Company and card holder related parking and sales transaction queries
  - Request and receive a list of sales transactions for a specific company
  - Request and receive a list of all parking transactions for a specific card holder (participant)

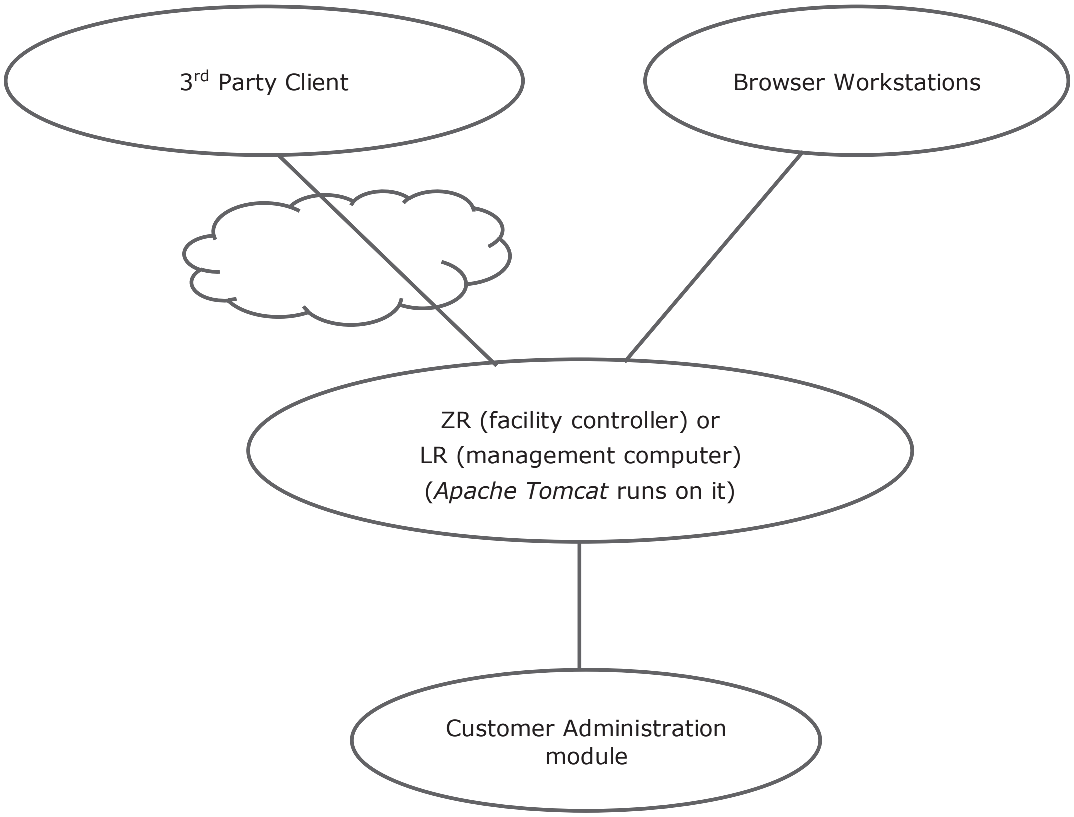
> *Not all data fields provided by the entervo Customer administration module are accessible via the current interface versions. The detailed functional volume is described in this document.*

1.2 Protocol

The following port is used in the entervo system for communication via network.

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

Directions of communication


3rd Party Client
Browser Workstations
ZR (facility controller) or
LR (management computer)
(*Apache Tomcat* runs on it)
Customer Administration module

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

Schema Location
Many requests contain XML. How this data is structured is defined in an XML-Schema file *custmedia.xsd*.

2.1 REST Requests

This chapter describes how to make REST requests to the Scheidt & Bachmann *Customer and Media 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 *Customer and Media Web Service*, and waiting for the HTTP response. Like any HTTP request, a REST request to *Customer and Media 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 the service name “custmedia”. The remainder of the URI indicates the particular resource you want to act on. For example, when creating a new contract, the following URI is used:
https://localhost:8443/CustomerMediaWebService/contracts.

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

Header Name	Description	Required
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

Customer and Media Web Service responses are standard HTTP responses. Some of the actions return special information specific to *Customer and Media 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 the particular action.

2.2.1 Error Responses

If a REST request results in an error, the HTTP response 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 example shows the XML error document in a REST error response:

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

The list of error descriptions can be found in chapter 4.5 "Appendix 5: List of Error Descriptions".

3 Resources

3.1 Resources Overview

The *Customer and Media Web Service* provides access to the following resources:

Administrative Resources

Resource	Description
Version	Provides access to the version of the web service
Resources	Provides a dictionary of all available application resources and there URIs

Primary Application Resources

Resource	Description
Contracts	The contract resource provides information about the contractor (personal data, address) and contract details like validity and contract conditions. Regarding the classic customer administration it is comparable to the Invoice Addressee.
Consumer	The consumer is the person who accesses a car park with an assigned identification media. In terms of the classic customer administration it is comparable to the participant. In this case, consumer can have only one identification media: the season card.
Invoices	Resources providing information about invoices and accounts. It is also used to start invoicing jobs.
Sales-/Park-transactions	Resources providing information about sales and park-transactions
Profiles	Resource providing information about usage profiles
Template	Resource providing information about templates used for creating contracts and consumers
Job	Resource providing information about invoicing, reminding or clearing job
BatchRequest	In order to provide a kind of import functionality a client can send a *batchrequest* with a set of contracts or consumers.

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

3.2 Resources URI Overview

Resource	Path	Method	Description	From version
Version	/version	GET	List of all available versions of the REST interface	1.3.1
Resources	/resources	GET	List of all primary resources of the REST interface	1.3.1
Contracts	/contracts	GET	List of all contracts	1.3.1
Contracts	/contracts	POST	Creates a new contract	1.3.1
Contracts	/contracts/{id}	GET	Gets a single contract	1.3.1
Contracts	/contracts/{id}	DELETE	Deletes a single contract	1.3.1
Contract Details	/contracts/{id}/detail	GET	Gets detailed Contract information	1.3.1
Contract Details	/contracts/{id}/detail	PUT	Updates detailed Contract information	1.3.1
Consumers of Contract	/contracts/{id}/consumers	GET	List of consumers belonging to a contract	1.3.1
Consumers of Contract	/contracts/{id}/consumersdetail	GET	List of consumers detail belonging to a contract	1.8.2
Consumers of Contract	/contracts/{id}/consumers	POST	Creates a new consumers belonging to a contract	1.3.1
Invoices of Contract	/contracts/{id}/invoices	GET	List of all invoices belonging to a contract	1.3.1

Resource	Path	Method	Description	From version
Invoices of Contract	/contracts/{id}/invoices	POST	Creates a new invoice belonging to a contract	1.3.1
Account of Contract	/contracts/{id}/account	GET	Shows account information about a contract	1.3.1
Sales Transactions	/contracts/{id}/sales	GET	List of all sales transactions of a contract	1.3.1
Sales Transactions	/contracts/sales	PUT	Updates sales transactions	1.7.0
Sales Transactions	/contracts/{id}/sales	POST	Create sales transactions of a contract	1.14.0
Park. Transactions	/contracts/{id}/parktrans	GET	List of all parking transactions of a contract	1.8.2
Contracts	contracts/{id}/freefacilities	GET	List of free facilities for contract	1.9.0
Contracts	contracts/{id}/freefacilities	POST	Creates a new free facility for contract	1.9.0
Contracts	contracts/{id}/freefacilities	DELETE	Deletes all free facilities for contract valid from actual date	1.9.0
Contracts	contracts/{id}/dynamiccounters	GET	Get the contract`s dynamic counters (present counter and overruns values)	1.23.0
Contracts	contracts/{id}/dynamiccounters	PUT	Update contract`s dynamic counters (present counter and overruns values)	1.23.0
Contracts	contracts/{id}/poolingentries	GET	Get the contract`s pooling data	1.23.0
Contracts	contracts/{id}/poolingentries	PUT	Update contract`s pooling data (except for present counter and overruns values)	1.23.0
Contracts	contracts/{id}/poolingentries/cdc	DELETE	Delete contract`s pooling data per company	1.23.0
Contracts	contracts/{id}/poolingentries/pcdc	DELETE	Delete contract`s pooling data per certain access profile	1.23.0

Resource	Path	Method	Description	From version
Contracts	contracts/{id}/poolingentries/fcdc	DELETE	Delete contract`s pooling data per certain facility	1.23.0
Consumers	/consumers	GET	Gets a list of all consumers	1.3.1
Consumers	/consumers/{Cid},{id}	GET	Gets a single consumer	1.3.1
Consumers	/consumers/{Cid},{id}	DELETE	Deletes a single consumer	1.3.1
Consumer Details	/consumers/{Cid},{id}/detail	GET	Gets detailed Consumer information	1.3.1
Consumer Details	/consumers/{Cid},{id}/detail	PUT	Updates detailed Consumer information	1.3.1
Sales Transactions	/consumers/{Cid},{id}/sales	GET	List of all sales transactions of a consumer	1.3.1
Sales Transactions	/consumers/sales	PUT	Updates sales transactions of a consumer	1.7.0
Sales Transactions	/consumers/{Cid},{id}/sales	POST	Create sales transactions of a consumer	1.14.0
Park. Transactions	/consumers/{Cid},{id}/parktrans	GET	List of all parking transactions of a consumer	1.3.1
Park. Transactions	/consumers/{Cid},{id}/parktrans	POST	Create parking transactions of a consumer	1.14.0
Identifiers	/consumers/{Cid},{id}/identifiers	GET	List of all identifiers of a consumer	1.24.0
Identifiers	/consumers/{Cid},{id}/identifiers	PUT	Update or create identifiers of a consumer	1.24.0
Identifiers	/consumers/{Cid},{id}/identifiers /{Iid}	DELETE	Delete single identifier of a consumer	1.24.0
Usage profiles	/profiles	GET	Gets a list of all usage profiles	1.8.2
Usage profile	/profiles/{id}/detail	GET	Gets a single usage profile	1.8.2
Templates	/templates/contracts	GET	List of all templates for contracts	1.8.2

Resource	Path	Method	Description	From version
Templates	/templates/consumers	GET	List of all templates for consumers	1.8.2
Jobs	/jobs	GET	List of all jobs (Invoice, Reminder, Clearing, etc.)	1.3.1
Job	/jobs/{id}	GET	Gets a status of a single Job	1.3.1
Batchs	/contracts/batchRequest	POST	Creates contracts as batch request	1.3.1
Batchs	/consumers/batchRequest	POST	Creates consumers as batch request	1.3.1
Batchs	/batchRequest/{id}	GET	Gets the batch request status	1.3.1
Invoices	/invoices	GET	Gets a list of all invoices	1.3.1
Invoices	/invoices/{id}	GET	Gets a single invoice	1.3.1
Invoices	/invoices	POST	Creates a new invoice	1.3.1
Invoices	/invoices/{id}/pdf	GET	Get detail invoice information	1.15.0
weekdayprogprof	/weekdayprogprof	GET	Get list of all week profiles from entervo	1.16.0

3.3 API Reference

The following chapter contains the reference of the *Customer and Media Web Service* API.

The first section contains an overview of all resources provided by the service and what 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
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. 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 XML format
Example response body	Data sent by the service in the response to the client request

The success of a response is indicated by a 3-digit code. The list of these codes can be found in chapter 0 "

Appendix 4: List of Status Response Codes".

The parameters are described in particular chapters. More parameters together with description and possible values can be found in chapter 4.2 "Appendix 2: List of Parameters".

3.3.1 Base URI

The base URI is the following:

https://hostname:port/CustomerMediaWebService

All other URIs described in this document will be appended to the base URI. Default port is 8443.

3.3.2 Common Resources

3.3.2.1 Version

Get information about used version of REST *CustomerMediaWebService* and *CustomerAdministration* service.

REST-Resource

HTTP-Method	GET
REST-URI	/version
Formats	Text, XML
Path parameters	-
Query parameters	-
Request body	-
HTTP Response codes	200, 500
Response body	<version> element

Example Header

```
GET /customer/rest/version HTTP/1.1
Host: gsph.sub.com
Authorization: Basic <Credential>
Date: Fri, 24 Dec 2010 18:00:00 GMT
Accept: application/xml
```

Example Request

https://hostname:port/CustomerMediaWebService/version

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<version xmlns="http://gsph.sub.com/cust/types">
  <version>
    1.8.8 (CustomerAdministrationService: 6.18.5)
  </version>
</version>
```

3.3.2.2 Resources

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	Text, XML
Path parameters	-
Query parameters	-
Request body	-

HTTP Response codes	200, 500
Response body	<resources> element

Example Request
https://hostname:port/CustomerMediaWebService/resources

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<resources xmlns="http://gsph.sub.com/cust/types">
  <resource>contracts</resource>
  <resource>consumers</resource>
  <resource>invoices</resource>
  <resource>profiles</resource>
  <resource>templates</resource>
  <resource>jobs</resource>
  <resource>batchRequest</resource>
</resources>
```

3.3.3 Contract Resources

3.3.3.1 Get a List of Contracts

Get a list of key information about contracts. An optional list of search parameters can be appended to the URI in order to reduce the result list.

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

REST-Resource

HTTP-Method	GET
REST-URI	/contracts
Formats	Text, XML
Path parameters	-
Query parameters	minId, maxId, validFrom, validUntil, name, contractNo, status, delete, instanceId
Request body	-
HTTP Response codes	200, 204, 500
Response body	<contracts> element

Query Parameters

Parameter	Short description	Format	Possible values
minId	Minimum contract id to retrieve	number	
maxId	Maximum contract id to retrieve	number	
validFrom	Contract valid from date	YYYY-MM-DD	

Parameter	Short description	Format	Possible values
validUntil	Contract valid until date	YYYY-MM-DD	
status	Blocking status	number	0: not blocked 2: blocked
delete	Contract is or is not marked for deletion.	number	0: not marked for deletion 1: marked for deletion
contractNo	Contract number (not identical with contract id)	string	
name	Name of contract holder	string [33]	
instanceId	Instance id	number	

> *The parameters **status**, **delete**, **contractNo** and **name** are available since the interface version 1.5.0.*

Example Request

https://hostname:port/CustomerMediaWebService/contracts?minId=50000&delete=0

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<contracts xmlns="http://gsph.sub.com/cust/types">
  <contract>
    <id>5001</id>
    <name>S&B Moenchengladbach</name>
    <xValidFrom>1872-07-14</xValidFrom>
    <xValidUntil>2100-31-12</xValidUntil>
    <filialId>1001</filialId>
  </contract>
  <contract>
    <id>50002</id>
    <name>S&B Slovakia</name>
    <xValidFrom>1995-07-14</xValidFrom>
    <xValidUntil>2030-31-12</xValidUntil>
    <filialId>1002</filialId>
  </contract>
</contracts>
```

3.3.3.2 Create a Contract

Create a new contract. The contract parameters have to be filled in the request body according to the *contractDetail* element specified in the XML schema *cust_media.xsd*.

Information which is not included in the *contractDetail* element is filled from a default contract template (or from template defined by templateId query parameter). All information from the template is filled in, mandatory as well as optional.

> *List of parameters of default contract template can be found in chapter 4.9 “Appendix 9: Default contract template parameters”*

If contract id is not defined (or is 0), it will be generated (replaced) by system (first available id from appropriate range will be assigned). Range for contract id is 2-49999 for local contracts, 50000-99999 for global contracts.

The complete information will be sent to the client in the response. If the contract is created successfully the http error code will be set to 201 and the location header will contain the URI of the new created resource.

REST-Resource

HTTP-Method	POST
REST-URI	/contracts
Formats	Text, XML
Path parameters	-
Query parameters	templateId, instanceId
Request body	<contractDetail> element
HTTP Response codes	201, 500
Response body	<contractDetail> element

Query Parameters

Parameter	Short description	Format	Possible values
templateId	Id of template used for creating contracts. Otherwise use id defined in property file (when not defined in property file, default contract template id 0 is used).	number	
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<contractDetail xmlns="http://gsph.sub.com/cust/types">
  <contract>
    <id>5001</id>
    <name>Beispielfirma1</name>
    <xValidFrom>2014-01-01</xValidFrom>
    <xValidUntil>2019-12-31</xValidUntil>
  </contract>
```

```
   <person>
      <surname> Beispielfirma1</surname>
      <phone1>0049 555 123123123</phone1>
      <email1>mail@mail.eu</email1>
   </person>
   <stdAddr>
      <street>Templergraben 1</street>
      <town>Citi</town>
   </stdAddr><pooling>
    <poolingDetail>
      <maxCounter>100</maxCounter>
      <poolingType>0</poolingType>
    </poolingDetail>
    <poolingDetail>
      <extCardProfile>1</extCardProfile>
      <maxCounter>20</maxCounter>
      <poolingType>2</poolingType>
    </poolingDetail>
  </pooling>
</contractDetail>
```

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<contractDetail xmlns="http://gsph.sub.com/cust/types">
   <contract href="/contracts/2">
      <id>2</id>
      <name>company A</name>
      <xValidFrom>2013-07-26</xValidFrom>
      <xValidUntil>2015-01-30</xValidUntil>
      <filialId>2010</filialId>
   </contract>
   <contractno>1</contractno>
   <person>
      <salutation>[not supported]</salutation>
      <title>Firma</title>
      <firstName>name</firstName>
      <middleName>[not supported]</middleName>
      <surname>Bachmann</surname>
      <gender>[not supported]</gender>
      <birthday>1980-07-02</birthday>
      <lang>[not supported]</lang>
      <contractLang>0</contractLang>
      <matchCode>d</matchCode>
      <phone1>0241/5060360</phone1>
      <phone3>[not supported]</phone3>
      <fax>345345</fax>
      <email1>info@mail.sb</email1>
```

```
    <email2>[not supported]</email2>
    <homepage>www.firma.eu</homepage>
  </person>
  <stdAddr>
    <street>Roonstra??e 7</street>
    <town>City</town>
    <postbox>52070</postbox>
    <country>[not supported]</country>
    <state>CO</state>
  </stdAddr>
  <deliveryAddr>
    <street>Breite Strasse 132</street>
    <town>City</town>
    <postbox>41238</postbox>
    <state>asCO</state>
  </deliveryAddr>
  <paymentInfo>
    <paymentType>10</paymentType>
    <bankacct>
      <acctno>12</acctno>
      <bankcode>31050000</bankcode>
      <bankname>Testrewq</bankname>
      <sepamandatref>DE123456789</sepamandatref>
      <sepamandatdate>2016-03-26</sepamandatdate>
      <sepamandatstatus>0</sepamandatstatus>
      <sepaprenotification>15</sepaprenotification>
      <sepaiban>DE112233445566778899</sepaiban>
      <sepabic>DE987654321</sepabic>
      <sepacreditoridentifier>DE02ZZZ01234567890
      </sepacreditoridentifier>
    </bankacct>
    <ccno1>
      <cardtype>0</cardtype>
      <cardno/>
      <cardname/>
      <cardvaliduntil>1900-01-01</cardvaliduntil>
    </ccno1>
    <ccno2>
      <cardtype>-1</cardtype>
      <cardno/>
      <cardname/>
      <cardvaliduntil>1900-01-01</cardvaliduntil>
    </ccno2>
  </paymentInfo>
  <counting>9999</counting>
  <present>0</present>
  <status>0</status>
  <delete>0</delete>
  <memo/>
  <invoicegroup>0</invoicegroup>
  <pooling>
   <poolingDetail>
     <maxCounter>100</maxCounter>
     <poolingType>0</poolingType>
   </poolingDetail>
   <poolingDetail>
     <extCardProfile>1</extCardProfile>
     <maxCounter>20</maxCounter>
     <poolingType>2</poolingType>
   </poolingDetail>
 </pooling>
</contractDetail>
```

With previously mentioned request (without instanceId query parameter), it is possible to create local contract (contract id from range 2-49999) on ZR and global contract (contract id from range 50000-99999) on LR. To create local contract on LR, or global contract on ZR, it is necessary to include query parameter instanceId of computer, where contract should belong, to request URI:

Example Request for creating local contract (belonging to ZR with instance id 2010) on LR
https://hostname:port/CustomerMediaWebService/contracts?instanceId=2010

Example Request for creating global contract (belonging to LR with instance id 9010) on ZR
https://hostname:port/CustomerMediaWebService/contracts?instanceId=9010

*In general, instanceId query parameter has to be used in most of the requests provided by Customer and Media Web Service, if it is needed to get, update, create or delete **local** contract, consumer, parking transaction etc. **on LR**, or to get, update, create or delete **global** contract, consumer, parking transaction etc. **on ZR**.*

3.3.3.3 Get a Contract

Get contract information about a specific contract. The id of the contract requested is part of the URI. If the contract with the specified id exists, the response body contains the *contractDetail* element.

REST-Resource

HTTP-Method	GET
REST-URI	/contracts/{id}
Formats	Text, XML
Path parameters	id
Query parameters	instanceId

Request body	-
HTTP Response codes	200, 404, 500
Response body	<contract> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contracts/5001

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<contracts xmlns="http://gsph.sub.com/cust/types">
  <contract>
    <id>5001</id>
    <name>S&B Moenchengladbach</name>
    <xValidFrom>1872-07-14</xValidFrom>
    <xValidUntil>2100-31-12</xValidUntil>
    <filialId>1001</filialId>
  </contract>
</contracts>
```

3.3.3.4 Delete a Contract

Normally, contracts are deleted automatically by the background system. Deleting a contract directly by submitting this request is only permitted with service authorization. The contract itself and all related information (i.e. consumers and their identification media) are deleted.

REST-Resource

HTTP-Method	DELETE
REST-URI	/contracts/{id}
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 404, 500
Response body	<contract> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/5001

3.3.3.5 Get Contract Detail

Get detailed information about a specific contract. The id of the contract requested is part of the URI. If the contract with the specified id exists the response body contains the *contractDetail* element.

REST-Resource

HTTP-Method	GET
REST-URI	/contracts/{id}/detail
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 404, 500
Response body	<contractDetail> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/5001/detail

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<contractDetail xmlns="http://gsph.sub.com/cust/types">
   <contract href="/contracts/5001">
      <id>5001</id>
       …
   </contract>
   …
</contractDetail>
```

Detailed response body can be found in chapter 3.3.3.2 "Create a Contract".

3.3.3.6 Update a Contract

Update a contract specified by {*id*} with the information contained in the *contract* element sent in the HTTP-Body.

REST-Resource

HTTP-Method	PUT
REST-URI	/contracts/{id}/detail
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	<contractDetail> element
HTTP Response codes	200, 404, 500
Response body	<contractDetail> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contracts/2/detail

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<contractDetail xmlns="http://gsph.sub.com/cust/types">
  <contract>
    <id>2</id>
    <xValidUntil>2021-12-31</xValidUntil>
  </contract>
  <person>
    <phone1>0049 555 123123123</phone1>
    <email1>mail@firma.eu</email1>
  </person>
</contractDetail>
```

Example Response Body (XML)

See response body in chapter 3.3.3.2 "Create a Contract".

3.3.4 Contract Data Resources

*The **lpn** parameter internally searches in all three columns (lpn1, lpn2, lpn3).*

3.3.4.1 Get a List of Contract Consumers

Retrieve a list of consumers belonging to a contract with the specified contract id. The list returned by the response contains key information about a single consumer and a URI to the consumer resource itself. To retrieve the detailed consumer data a client must access this URI.

REST-Resource

HTTP-Method	GET
REST-URI	/contracts/{id}/consumers
Formats	Text, XML
Path parameters	id
Query parameters	minId, maxId, validFrom, validUntil, name, status, delete, cardno, lpn1, lpn2, lpn3, ignorePresence, lpn, instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<consumers> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values

Parameter	Short description	Format	Possible values
minId	Minimum consumer id to retrieve	number	
maxId	Maximum consumer id to retrieve	number	
validFrom	Consumer valid from date	YYYY-MM-DD	
validUntil	Consumer valid until date	YYYY-MM-DD	
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/consumers

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<consumers xmlns="http://gsph.sub.com/cust/types">
  <consumer href="/consumers/2,1">
    <id>1</id>
    <contractid>2</contractid>
    <name>Default Teilnehmer</name>
    <filialId>2010</filialId>
  </consumer>
  <consumer href="/consumers/2,2">
    <id>2</id>
    <contractid>2</contractid>
    <name>Default Teilnehmer</name>
    <filialId>2010</filialId>
  </consumer>
  …
</consumers>
```

3.3.4.2 Get a Detailed List of Contract Consumers

Retrieve a list of consumers belonging to a contract with the specified contract id. The list returned by the response contains key information about a single consumer and a URI to the consumer resource itself. To retrieve the detailed consumer data a client must access this URI.

REST-Resource

HTTP-Method	GET
REST-URI	/contracts/{id}/consumersdetail
Formats	Text, XML
Path parameters	id
Query parameters	minId, maxId, validFrom, validUntil, name, status, delete, cardno, lpn1, lpn2, lpn3, ignorePresence, lpn, instanceId

Request body	-
HTTP Response codes	200, 404, 500
Response body	<consumerDetails> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
minId	Minimum consumer id to retrieve	number	
maxId	Maximum consumer id to retrieve	number	
validFrom	Consumer valid from date	YYYY-MM-DD	
validUntil	Consumer valid until date	YYYY-MM-DD	
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contracts/2/consumersdetail

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<consumerDetails xmlns="http://gsph.sub.com/cust/types">
  <consumerDetail>
    <consumer href="/consumers/2,1">
      <id>1</id>
      <contractid>2</contractid>
      <name/>
      <filialId>2010</filialId>
    </consumer>
    <person>
      <firstName>TN</firstName>
      <surname>Default Teilnehmer</surname>
    </person>
    <identification>
      <ptcptType>2</ptcptType>
      <cardno>656456546546</cardno>
      <cardclass>1</cardclass>
      <identificationType>51</identificationType>
      <validFrom>2014-08-15</validFrom>
      <validUntil>2015-01-30</validUntil>
      <usageProfile href="/usageProfile/1">
        <id>1</id>
        <name>Standardh</name>
        <description/>
      </usageProfile>
      <admission/>
      <ignorePresence>0</ignorePresence>
      <present>false</present>
      <status>6</status>
      <ptcptGrpNo>-1</ptcptGrpNo>
      <chrgOvdrftAcct>1</chrgOvdrftAcct>
    </identification>
    <displayText>-1</displayText>
    <limit>9999900</limit>
    <status>6</status>
    <delete>0</delete>
    <userfield3>text</userfield3>
  </consumerDetail>
  <consumerDetail>
    <consumer href="/consumers/2,2">
      <id>2</id>
      <contractid>2</contractid>
  …
  </consumerDetail>
</consumerDetails>
```

3.3.4.3 Create a Contract Consumer

Add a new consumer to a specified contract. The consumer details have to be filled in the request body according to the *consumerDetail* element specified in the XML schema *cust_media.xsd*.

Information which is not included in the *consumerDetail* element is filled from a default consumer template (or from template defined by templateId query parameter). All information from the template is filled in, mandatory as well as optional.

List of parameters of default consumer template can be found in chapter 4.10 "Appendix 10: Default consumer template parameters"

If consumer id is not defined, it will be generated by system (first available id will be assigned). Range for consumer id is 1-99998.

The complete data is returned in the response body.

REST-Resource

HTTP-Method	POST
REST-URI	/contracts/{id}/consumers
Formats	Text, XML
Path parameters	id

Query parameters	templateId, instanceId
Request body	<consumerDetail> element
HTTP Response codes	201, 404, 500
Response body	<consumerDetail> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
templateId	Id of template used for creating consumer. Otherwise use Id defined in property file (when not defined in property file, default consumer template id 1 is used).	number	
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/consumers

Example Request/Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<consumerDetail xmlns="http://gsph.sub.com/cust/types">
  <consumer href="/consumers/2,1">
    <id>1</id>
    <contractid>2</contractid>
    <name/>
    <filialId>2010</filialId>
  </consumer>
  <person>
    <firstName>TN</firstName>
    <surname>Default Teilnehmer</surname>
  </person>
  <identification>
    <ptcptType>2</ptcptType>
    <cardno>656456546546</cardno>
    <cardclass>1</cardclass>
    <identificationType>51</identificationType>
    <validFrom>2014-08-15</validFrom>
    <validUntil>2015-01-30</validUntil>
    <usageProfile href="/usageProfile/1">
      <id>1</id>
      <name>Standarth</name>
      <description/>
    </usageProfile>
    <admission/>
    <ignorePresence>0</ignorePresence>
    <present>false</present>
    <status>6</status>
    <ptcptGrpNo>-1</ptcptGrpNo>
    <chrgOvdrftAcct>1</chrgOvdrftAcct>
  </identification>
  <displayText>-1</displayText>
  <limit>9999900</limit>
  <status>6</status>
  <delete>0</delete>
  <userfield3>bug6399</userfield3>
</consumerDetail>
```

3.3.4.4 Get Contract Invoices

Get the list of invoices related to a specified contract. As response a list of invoices will be provided which contains main information about an invoice. To get more detailed information about an invoice the client can follow the URI included in each list element.

REST-Resource

HTTP-Method	GET
REST-URI	/contracts/{id}/invoices
Formats	Text, XML
Path parameters	id
Query parameters	minId, maxId, minDate, maxDate, instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<invoices> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
minId	Minimum consumer id to retrieve	number	
maxId	Maximum consumer id to retrieve	number	

Parameter	Short description	Format	Possible values
minDate	Minimum invoice date	YYYY-MM-DD	
maxDate	Maximum invoice date	YYYY-MM-DD	
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/invoices

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<invoices xmlns="http://gsph.sub.com/cust/types">
  <invoice href="/invoices/66">
    <id>66</id>
    <invcDate>2013-12-18T11:25:00.000+01:00</invcDate>
    <invcAmount>1200</invcAmount>
    <companyRef>2</companyRef>
    <companyUrl>contracts/2</companyUrl>
  </invoice>
  <invoice href="/invoices/65">
    <id>65</id>
    <invcDate>2013-12-18T11:07:44.000+01:00</invcDate>
    <invcAmount>200</invcAmount>
    <companyRef>2</companyRef>
    <companyUrl>contracts/2</companyUrl>
  </invoice>
  …
</invoices>
```

3.3.4.5 Create a Contract Invoice

Create an invoice job for a given contract. The contract id has to be specified as part of the URI. If the request is processed successfully by the server, the response contains an URI to a job resource which status can be requested by the client.

REST-Resource

HTTP-Method	POST
REST-URI	/contracts/{id}/invoices
Formats	Text, XML
Path parameters	id
Query parameters	startDateTime, refDate, flatRate, bestPrice, instanceId
Request body	-
HTTP Response codes	201, 404, 500

Response body	<invoiceJob> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
startDateTime	Start date and time for invoicing job	YYYY-MM-DDTHH:MM:SS	
refDate	Reference date for invoice job	YYYY-MM-DDTHH:MM:SS	
flatRate	Take monthly flat rate into account in this invoicing job.	number	0, 1
bestPrice	Take bestprice contract parameters into account in this invoicing job.	number	0, 1
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/invoices

3.3.4.6 Get Contract Account

Get detailed information about an account assigned to a contract.

REST-Resource

HTTP-Method	GET
REST-URI	/contracts/{id}/account
Formats	Text, XML
Path parameters	id
Query parameters	minDate, maxDate, instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<account> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
minDate	Retrieve account information from <minDate>.	YYYY-MM-DD	
maxDate	Retrieve account information until <maxDate>.	YYYY-MM-DD	
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contracts/2/account

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<account xmlns="http://gsph.sub.com/cust/types">
  <accountTransaction>
    <id>1</id>
    <transactType>0</transactType>
    <createTime>2013-07-26T00:00:00.000+02:00</createTime>
    <bookingTime>2013-07-26T17:37:38.000+02:00</bookingTime>
    <amount>0</amount>
  </accountTransaction>
  …
</account>
```

3.3.4.7 Get Contract Sales Transactions

Get sales transactions of a contract which have not been invoiced yet. By adding the *minDate* and/or *maxDate* parameters to the request URI the results can be filtered.

REST-Resource

HTTP-Method	GET
REST-URI	/contracts/{id}/sales
Formats	Text, XML
Path parameters	id
Query parameters	minDate, maxDate, exported, instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<salesTransactions> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
minDate	Retrieve all sales transactions from <minDate>.	YYYY-MM-DD	
maxDate	Retrieve all sales transactions until <maxDate>.	YYYY-MM-DD	
exported	Retrieve sales transactions: - all - not sent yet - already sent	number	-1: Retrieve all sales transactions (default). 0: Retrieve only sales transactions which occurred from the last request. 1: Retrieve only sales transactions which have already been sent.
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/account

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?><salesTransactions
xmlns="http://gsph.sub.com/cust/types">
  <salesTransaction>
    <contractid>2</contractid>
    <consumerid>1</consumerid>
    <articleid>2</articleid>
    <articleclass>5</articleclass>
    <articlename>Kaution Einn. Rech</articlename>
    <quantity>1</quantity>
    <singleprice>0.0</singleprice>
    <totalamount>1000.0</totalamount>
    <servicetime>2014-01-24T10:44:24.000+01:00</servicetime>
    <selltime>2014-01-24T10:44:24.000+01:00</selltime>
    <vattype>A</vattype>
    <vatamount>0.0</vatamount>
    <pointofsale>0</pointofsale>
    <extexpmark>0</extexpmark>
  </salesTransaction>
  …
</salesTransactions>
```

3.3.4.8 Create a Contract Sales Transactions

Create a new sales transaction for a contract.

REST-Resource

HTTP-Method	POST

REST-URI	/contracts/{id}/sales
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	<salesTransactions> element
HTTP Response codes	201, 404, 500
Response body	<salesTransactions> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<salesTransactions xmlns="http://gsph.sub.com/cust/types">
    <salesTransaction>
        <contractid>2</contractid>
        <articleid>410</articleid>
        <articleclass>1</articleclass>
        <quantity>2</quantity>
        <totalamount>230.0</totalamount>
        <selltime>2016-03-04T14:20:53.000+01:00</selltime>
    </salesTransaction>
</salesTransactions>
```

Example Response Body (XML)

See response body in chapter 3.3.6.1 "Get Consumer Sales Transactions".

3.3.4.9 Update Contract Sales Transactions

Update sales transactions of a contract which have not been invoiced yet.

REST-Resource

HTTP-Method	PUT
REST-URI	/contracts/sales
Formats	Text, XML
Path parameters	-
Query parameters	instanceId
Request body	<salesTransactions> element
HTTP Response codes	200, 404, 500

Response body	<salesTransactions> element

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contracts/sales

Example Request/Response Body (XML)

See response body in chapter 3.3.4.7 "Get Contract Sales Transactions".

3.3.4.10 Block and Delete Mark the Contract

The contract is possible to block, unblock mark as delete with command update contract.

For block contract is used the element ***contractDetail –status***.

For mark delete consumer is used the element ***consumerDetail – delete.***

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<contractDetail xmlns="http://gsph.sub.com/cust/types">
  <contract>
    <id>3</id>
  </contract>
  <status>0</status>
  <delete>0</delete>
</contractDetail>
```

3.3.4.11 Get Contract Parking Transactions

Get parking transactions within a contract. By adding the *minDate* and/or *maxDate* parameters to the request URI the results can be filtered.

REST-Resource

HTTP-Method	GET
REST-URI	/contracts/{id}/parktrans
Formats	Text, XML
Path parameters	id
Query parameters	minDate, maxDate, instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<parkTransactions> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
minDate	Retrieve all park transactions from <minDate>.	YYYY-MM-DD	
maxDate	Retrieve all park transactions until <maxDate>.	YYYY-MM-DD	
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contracts/2/parktrans

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<parkTransactions xmlns="http://gsph.sub.com/cust/types">
  <parkTransaction>
    <transactionTime>2014-04-
       23T10:45:49.000+02:00</transactionTime>
    <transactionType>42</transactionType>
    <facilityin>0</facilityin>
    <facilityout>0</facilityout>
    <computer>2010</computer>
    <device>201</device>
    <amount>500.0</amount>
    <contractid>2</contractid>
    <consumerid>3</consumerid>
  </parkTransaction>
  …
</parkTransactions>
```

3.3.4.12 Get Contract Free Facilities

Get free facilities within a contract. This resource must be activated on *CustomerAdministration* module.

REST-Resource

HTTP-Method	GET
REST-URI	contracts/{id}/freefacilities
Formats	Text, XML
Path parameters	-
Query parameters	instanceId
Request body	-

HTTP Response codes	200, 204, 404, 500
Response body	<contractFreeFacilities> element

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/freefacilities

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<contractFreeFacilities xmlns="http://gsph.sub.com/cust/types">
  <contractFreeFacility>
    <xValidFrom>2013-11-20</xValidFrom>
    <xValidUntil>2013-11-20</xValidUntil>
    <facilityid>2011</facilityid>
  </contractFreeFacility>
  <contractFreeFacility>
    <xValidFrom>2014-11-19</xValidFrom>
    <xValidUntil>2014-11-19</xValidUntil>
    <facilityid>2012</facilityid>
  </contractFreeFacility>
</contractFreeFacilities>
```

3.3.4.13 Create Contract Free Facilities

Create free facilities for a given contract. The contract id has to be specified as part of the URI. This resource must be activated on *CustomerAdministration* module.

REST-Resource

HTTP-Method	POST
REST-URI	contracts/{id}/freefacilities
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	<contractFreeFacilities> element
HTTP Response codes	201, 404, 500
Response body	-

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/freefacilities

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<contractFreeFacilities xmlns ="http://gsph.sub.com/cust/types">
  <contractFreeFacility>
    <xValidFrom>2015-07-02</xValidFrom>
    <xValidUntil>2015-07-02</xValidUntil>
    <facilityid>2011</facilityid>
  </contractFreeFacility>
</contractFreeFacilities>
```

3.3.4.14 Delete Contract Free Facilities

Delete all free facilities for a given contract valid from actual date. The contract id has to be specified as part of the URI. This resource must be activated on *CustomerAdministration* module.

REST-Resource

HTTP-Method	DELETE
REST-URI	contracts/{id}/freefacilities
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 404, 500
Response body	-

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contracts/2/freefacilities

3.3.4.15 Get Contract dynamic counters

Get dynamic counters (present counter and overruns values) from all pooling records belonging to a contract with the specified contract id.

This resource will provide the data only when ***extended pooling*** *functionality is activated in* CustomerAdministration *module.*

REST-Resource

HTTP-Method	GET
REST-URI	/contracts/{id}/dynamiccounters

Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<pooling> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/dynamiccounters

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<pooling xmlns="http://gsph.sub.com/cust/types">
  <poolingDetail>
    <facility>0</facility>
    <extCardProfile>0</extCardProfile>
    <presentCounter>1</presentCounter>
    <overruns>1</overruns>
  </poolingDetail>
  <poolingDetail>
    <facility>0</facility>
    <extCardProfile>1</extCardProfile>
    <presentCounter>20</presentCounter>
    <overruns>6</overruns>
  </poolingDetail>
  <poolingDetail>
    <facility>1819</facility>
    <extCardProfile>0</extCardProfile>
    <presentCounter>0</presentCounter>
    <overruns>0</overruns>
  </poolingDetail>
   …
</pooling>
```

3.3.4.16 Get Contract pooling entries

Get pooling data belonging to a contract with the specified contract id.

> *This resource will provide the data only when* ***extended pooling*** *functionality is activated in* CustomerAdministration *module.*

REST-Resource

HTTP-Method	GET
REST-URI	/contracts/{id}/poolingentries
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<pooling> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contracts/2/poolingentries

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<pooling xmlns="http://gsph.sub.com/cust/types">
  <poolingDetail>
    <facility>0</facility>
    <extCardProfile>0</extCardProfile>
    <maxCounter>69</maxCounter>
    <presentCounter>1</presentCounter>
    <overruns>1</overruns>
    <poolingType>2</poolingType>
    <poolingProfile>2</poolingProfile>
  </poolingDetail>
<poolingDetail>
    <facility>0</facility>
    <extCardProfile>1</extCardProfile>
    <maxCounter>20</maxCounter>
    <presentCounter>20</presentCounter>
    <overruns>6</overruns>
    <maxAssigned>30</maxAssigned>
    <currentlyAssigned>3</currentlyAssigned>
    <poolingType>0</poolingType>
    <poolingProfile>0</poolingProfile>
  </poolingDetail>
  <poolingDetail>
    <facility>1819</facility>
    <extCardProfile>0</extCardProfile>
    <maxCounter>50</maxCounter>
    <presentCounter>0</presentCounter>
    <overruns>0</overruns>
    <poolingType>1</poolingType>
    <poolingProfile>0</poolingProfile>
  </poolingDetail>
</pooling>
```

3.3.4.17 Update Contract dynamic counters

Update dynamic counters (present counter and overruns values) of certain pooling records belonging to a contract with the specified contract id.

> *This resource will allow changing the data only when* ***extended pooling*** *functionality is activated in* CustomerAdministration *module.*

REST-Resource

HTTP-Method	PUT
REST-URI	/contracts/{id}/dynamiccounters
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	<pooling> element
HTTP Response codes	200, 204, 404, 500
Response body	<pooling> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contracts/2/dynamiccounters

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<pooling xmlns="http://gsph.sub.com/cust/types">
  <poolingDetail>
    <presentCounter>1</presentCounter>
    <overruns>1</overruns>
  </poolingDetail>
  <poolingDetail>
    <extCardProfile>1</extCardProfile>
    <presentCounter>20</presentCounter>
    <overruns>6</overruns>
  </poolingDetail>
  <poolingDetail>
    <facility>1819</facility>
    <presentCounter>0</presentCounter>
    <overruns>0</overruns>
  </poolingDetail>
   …
</pooling>
```

3.3.4.18 Update or create Contract pooling entries

Update pooling data (except for present counter and overruns values) or create pooling data of a contract with the specified contract id.

This resource will allow changing the data only when ***extended pooling*** *functionality is activated in* CustomerAdministration *module.*

REST-Resource

HTTP-Method	PUT
REST-URI	/contracts/{id}/poolingentries
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	<pooling> element
HTTP Response codes	200, 204, 404, 500
Response body	<pooling> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/poolingentries

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<pooling xmlns="http://gsph.sub.com/cust/types">
  <poolingDetail>
    <facility>0</facility>
    <extCardProfile>0</extCardProfile>
    <maxCounter>69</maxCounter>
    <poolingType>2</poolingType>
    <poolingProfile>2</poolingProfile>
  </poolingDetail>
  <poolingDetail>
    <facility>0</facility>
    <extCardProfile>1</extCardProfile>
    <maxCounter>20</maxCounter>
    <maxAssigned>30</maxAssigned>
    <poolingType>0</poolingType>
    <poolingProfile>0</poolingProfile>
  </poolingDetail>
  <poolingDetail>
    <facility>1819</facility>
    <extCardProfile>0</extCardProfile>
    <maxCounter>50</maxCounter>
    <poolingType>1</poolingType>
    <poolingProfile>0</poolingProfile>
  </poolingDetail>
   …
</pooling>
```

3.3.4.19 Delete Contract pooling entries (limits per company)

Delete pooling data (limits per company) of a contract with the specified contract id.

> *This resource will allow deleting the data only when **extended pooling** functionality is activated in* CustomerAdministration *module.*

REST-Resource

HTTP-Method	DELETE
REST-URI	/contracts/{id}/poolingentries/cdc
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500

Response body	<pooling> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/poolingentries/cdc

3.3.4.20 Delete Contract pooling entries (limits per access profile)

Delete pooling data (limits per specified access profile) of a contract with the specified contract id.

> *This resource will allow deleting the data only when **extended pooling** functionality is activated in* CustomerAdministration *module.*

REST-Resource

HTTP-Method	DELETE
REST-URI	/contracts/{id}/poolingentries/pcdc
Formats	Text, XML
Path parameters	id
Query parameters	extProfileId, instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<pooling> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
extProfileId	Access profile Id	number	
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/contracts/2/poolingentries/pcdc?extProfileId=2

20201023UI0504

3.3.4.21 Delete Contract pooling entries (limits per facility)

Delete pooling data (limits per specified facility) of a contract with the specified contract id.

> *This resource will allow deleting the data only when* ***extended pooling*** *functionality is activated in* CustomerAdministration *module.*

REST-Resource

HTTP-Method	DELETE
REST-URI	/contracts/{id}/poolingentries/fcdc
Formats	Text, XML
Path parameters	id
Query parameters	facilityId, instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<pooling> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Contract id	number	

Query Parameters

Parameter	Short description	Format	Possible values
facilityId	Facility Id	number	
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contracts/2/poolingentries/fcdc?facilityId=1819

3.3.5 Consumer Resources

> *The* ***lpn*** *parameter internally searches in all three columns (lpn1, lpn2, lpn3).*

3.3.5.1 Get Consumers

Get a list with key information about consumers. An optional list of search parameters can be appended to the URI in order to reduce the result list.

The consumer list contains key information about the consumers. To get the full information for a consumer the consumer resource must be requested. The URI is part of the contract key information.

REST-Resource

HTTP-Method	GET
REST-URI	/consumers
Formats	Text, XML

Path parameters	-
Query parameters	minContractId, maxContractId, minId, maxId, validFromContract, validUntilContract, validFromConsumer, validUntilConsumer, name, status, delete, cardno, lpn1, lpn2, lpn3, ignorePresence, lpn, instanceId
Request body	-
HTTP Response codes	200, 204, 500
Response body	<consumers> element

Query Parameters

Parameter	Short description	Format	Possible values
minContractId	Minimum Contract id to retrieve	number	
maxContractId	Maximum Contract id to retrieve	number	
minId	Minimum Consumer id to retrieve	number	
maxId	Maximum Consumer id to retrieve	number	
validFromConsumer	Consumer valid from date	YYYY-MM-DD	
validUntilConsumer	Consumer valid until date	YYYY-MM-DD	
validFromContract	Contract valid from date	YYYY-MM-DD	
validUntilContract	Contract valid until date	YYYY-MM-DD	
status	Blocking status	number	0: not blocked 2: blocked
delete	Consumer is or is not marked for deletion.	number	0: not marked for deletion 1: marked for deletion
lpn1	Licence plate 1	string [15]	
lpn2	Licence plate 2	string [15]	
lpn3	Licence plate 3	string [15]	
cardNo	Identifier number of consumer	string [9]	
name	Name of the consumer	string [33]	

Parameter	Short description	Format	Possible values
ignorePresence	Presence check is enabled or disabled.	number	0: presence check is enabled 1: presence check is disabled once >1: presence check is disabled permanently
instanceId	Instance id	number	

The parameters ***status****,* ***delete****,* ***lpn1****,* ***lpn2****,* ***lpn3****,* ***cardNo****,* ***name*** *and* ***ignorePresence*** *are available since the interface version 1.5.0.*

Example Request
https://hostname:port/CustomerMediaWebService/consumers

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<consumers xmlns="http://gsph.sub.com/cust/types">
  <consumer href="/consumers/2,1">
    <id>1</id>
    <contractid>2</contractid>
    <name/>
    <filialId>2010</filialId>
  </consumer>
  <consumer href="/consumers/2,2">
    <id>2</id>
    <contractid>2</contractid>
    <name/>
    <filialId>2010</filialId>
  </consumer>
  …
</consumers>
```

3.3.5.2 Get a Consumer

Get the key information about a specific consumer. The id of the consumer requested and the id of the contract to which the consumer is associated are part of the URI. This request can be used to check if a consumer with a specified id exists without retrieving the entire detail information.

REST-Resource

HTTP-Method	GET
REST-URI	/consumers/{Cid},{id}
Formats	Text, XML
Path parameters	Cid, id
Query parameters	instanceId
Request body	-

HTTP Response codes	200, 404, 500
Response body	<consumer> element

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	
id	Consumer id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/consumers/2,1

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<consumer xmlns="http://gsph.sub.com/cust/types"
href="/consumers/2,1">
   <id>1</id>
   <contractid>2</contractid>
   <name/>
   <xValidFrom>2014-08-15</xValidFrom>
   <xValidUntil>2015-01-30</xValidUntil>
   <filialId>2010</filialId>
</consumer>
```

3.3.5.3 Get Consumer Detail

Get consumer detailed information. To specify the consumer the unique identifier of the contract (*Cid*) to which the consumer belongs and the consumer identifier (*id*) must be appended to the base consumer path.

REST-Resource

HTTP-Method	GET
REST-URI	/consumers/{Cid},{id}/detail
Formats	Text, XML
Path parameters	Cid, id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 404, 500
Response body	<consumerDetail> element

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	

Parameter	Short description	Format	Possible values
id	Consumer id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/consumers/2,1/detail

Example Response Body (XML)

See response body in chapter 3.3.4.3 "Create a Contract Consumer".

3.3.5.4 Update a Consumer

Update a consumer specified by a contract id {*Cid*} and a consumer id {*id*} with the information sent in the HTTP body.

REST-Resource

HTTP-Method	PUT
REST-URI	/consumers/{Cid},{id}/detail
Formats	Text, XML
Path parameters	Cid, id
Query parameters	instanceId
Request body	
HTTP Response codes	200, 404, 500
Response body	<consumerDetail> element

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	
id	Consumer id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/consumers/2,1/detail

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<consumerDetail xmlns="http://gsph.sub.com/cust/types">
   <consumer href="/consumers/2,1">
      <id>1</id>
      <contractid>2</contractid>
      <name/>
   </consumer>
   <person>
      <surname>Director</surname>
   </person>
   <identification>
      <cardno>4111111111111111</cardno>
      <identificationType>51</identificationType>
      <validUntil>2016-01-30</validUntil>
</consumerDetail>
```

Example Response Body (XML)

See response body in chapter 3.3.5.2 “Get a Consumer”.

3.3.5.5 Block, Delete Mark and Neutral the Consumer

The consumer is possible to block, unblock, neutralise and mark as delete with command update consumer.

For block consumer are used the elements ***consumerDetail – identification -status*** and ***consumerDetail - status***. The high priority has element in identification section.

For neutral the consumer are used the elements ***consumerDetail - identification - ignorePresence*** and ***consumerDetail - ignorePresence***. The high priority has element in identification section.

For mark delete the consumer is used the element ***consumerDetail – delete.***

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<consumerDetail xmlns="http://gsph.sub.com/cust/types">
  <consumer>
    <id>2</id>
    <contractid>3</contractid>
  </consumer>
  <ignorePresence>1</ignorePresence>
  <status>0</status>
  <delete>0</delete>
  <identification>
      <ignorePresence>0</ignorePresence>
      <status>2</status>
  </identification>
</consumerDetail>
```

3.3.5.6 Delete a Consumer

Normally, consumers are deleted automatically by the background system. Deleting a consumer directly by submitting this request is only permitted with service authorization. The consumer itself and all related information will be deleted.

REST-Resource

HTTP-Method	DELETE
REST-URI	/consumers/{Cid},{id}
Formats	Text, XML
Path parameters	Cid, id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 404, 500
Response body	-

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	
id	Consumer id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/consumers/2,1

3.3.6 Consumers Data Resources

3.3.6.1 Get Consumer Sales Transactions

Get sales transactions related to a special consumer. Only sales transactions which are not invoiced yet are returned. By adding the *minDate* and/or *maxDate* parameters to the request URI the transactions to be returned can be filtered.

REST-Resource

HTTP-Method	GET
REST-URI	/consumers/{Cid},{id}/sales
Formats	Text, XML
Path parameters	Cid, id
Query parameters	minDate, maxDate, exported, instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<salesTransactions> element

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	
id	Consumer id	number	

Query Parameters

Parameter	Short description	Format	Possible values
minDate	Retrieve all sales transactions from <minDate>.	YYYY-MM-DD	
maxDate	Retrieve all sales transactions until <maxDate>.	YYYY-MM-DD	
exported	Retrieve sales transactions: - all - not sent yet - already sent	number	-1: Retrieve all sales transactions (default). 0: Retrieve only sales transactions which occurred from the last request. 1: Retrieve only sales transactions which have already been sent.
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/consumers/2,1/sales

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<salesTransactions xmlns="http://gsph.sub.com/cust/types">
  <salesTransaction>
    <contractid>2</contractid>
    <consumerid>1</consumerid>
    <articleid>2</articleid>
    <articleclass>5</articleclass>
    <articlename>Kaution Einn. Rech</articlename>
    <quantity>1</quantity>
    <singleprice>0.0</singleprice>
    <totalamount>1000.0</totalamount>
    <servicetime>2014-01-24T10:44:24.000+01:00</servicetime>
    <selltime>2014-01-24T10:44:24.000+01:00</selltime>
    <vattype>A</vattype>
    <vatamount>0.0</vatamount>
    <pointofsale>0</pointofsale>
    <extexpmark>0</extexpmark>
  </salesTransaction>
```

```
    <salesTransaction>
      <contractid>2</contractid>
      <consumerid>1</consumerid>
      <articleid>2</articleid>
      <articleclass>5</articleclass>
      <articlename>Kaution Einn. Rech</articlename>
      <quantity>1</quantity>
      <singleprice>0.0</singleprice>
      <totalamount>1100.0</totalamount>
      <servicetime>2014-01-24T10:47:44.000+01:00</servicetime>
      <selltime>2014-01-24T10:47:44.000+01:00</selltime>
      <vattype>A</vattype>
      <vatamount>0.0</vatamount>
      <pointofsale>0</pointofsale>
      <extexpmark>0</extexpmark>
   </salesTransaction>
   …
</salesTransactions>
```

3.3.6.2 Create a Consumer Sales Transactions

Create a new sales transaction for a customer.

REST-Resource

HTTP-Method	POST
REST-URI	/consumers/{Cid},{id}/sales
Formats	Text, XML
Path parameters	Cid, id
Query parameters	useShadowContract, instanceId
Request body	<salesTransactions> element
HTTP Response codes	201, 404, 500
Response body	<salesTransactions> element

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	
id	Consumer id	number	

Query Parameters

Parameter	Short description	Format	Possible values
useShaowContract	Define a possibility of creating this sales for "shadow" contracts of this consumer or for "official" contracts of this consumer	boolean	true / false default value: false

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<salesTransactions xmlns="http://gsph.sub.com/cust/types">
    <salesTransaction>
        <contractid>2</contractid>
        <consumerid>9</consumerid>
        <articleid>410</articleid>
        <articleclass>1</articleclass>
        <quantity>2</quantity>
        <totalamount>230.0</totalamount>
        <selltime>2016-03-04T14:20:53.000+01:00</selltime>
    </salesTransaction>
</salesTransactions>
```

Example Response Body (XML)

See response body in chapter 3.3.6.1 "Get Consumer Sales Transactions".

3.3.6.3 Update Consumer Sales Transactions

Update the ExtExpMark flag of the sales transactions of a consumer that have not been invoiced yet.

REST-Resource

HTTP-Method	PUT
REST-URI	/consumers/sales
Formats	Text, XML
Path parameters	-
Query parameters	instanceId
Request body	<salesTransactions> element
HTTP Response codes	200, 404, 500
Response body	<salesTransactions> element

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/contract/sales

Example Request Body (XML)

See response body in chapter 3.3.6.1 "Get Consumer Sales Transactions".

Example Response Body (XML)

> *See response body in chapter 3.3.6.1 "Get Consumer Sales Transactions".*

3.3.6.4 Get Consumer Parking Transactions

Get sales transactions related to a special consumer. The consumer is specified by the combination of the contract id {*Cid*} and consumer id {*id*}. By adding the *minDate* and/or *maxDate* parameters to the request URI the transactions to be returned can be filtered.

REST-Resource

HTTP-Method	GET
REST-URI	/consumers/{Cid},{id}/parktrans
Formats	Text, XML
Path parameters	Cid, id
Query parameters	minDate, maxDate, instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<parkTransactions> element

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	
id	Consumer id	number	

Query Parameters

Parameter	Short description	Format	Possible values
minDate	Retrieve all park transactions from <minDate>.	YYYY-MM-DD	
maxDate	Retrieve all park transactions until <maxDate>.	YYYY-MM-DD	
exported	Retrieve sales transactions: - all - not sent yet - already sent	number	-1: Retrieve all sales transactions (default). 0: Retrieve only sales transactions which occurred from the last request. 1: Retrieve only sales transactions which have already been sent.
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/consumers/2,1/parktrans

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?><parkTransactions
xmlns="http://gsph.sub.com/cust/types">
   <parkTransaction>
      <transactionTime>2014-07-
          30T11:34:44.000+02:00</transactionTime>
      <transactionType>42</transactionType>
      <facilityin>0</facilityin>
      <facilityout>0</facilityout>
      <computer>2010</computer>
    <device>201</device>
      <amount>500.0</amount>
   </parkTransaction>
   …
</parkTransactions>
```

3.3.6.5 Create a Consumer Park Transactions

Create a new park transaction for a customer.

REST-Resource

HTTP-Method	POST
REST-URI	/consumers/{Cid},{id}/parktrans
Formats	Text, XML
Path parameters	Cid, id
Query parameters	instanceId
Request body	<bookParkTransaction> element
HTTP Response codes	201, 404, 500
Response body	<bookParkTransaction> element

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	
Id	Consumer id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request/Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<bookParkTransaction xmlns="http://gsph.sub.com/cust/types">
   <entryParkTransaction>
       <transactionTime>2016-02-
18T15:15:41.000+01:00</transactionTime>
       <transactionType>2</transactionType>
       <facilityin>2011</facilityin>
       <facilityout>0</facilityout>
       <computer>2010</computer>
       <device>101</device>
       <amount>0.0</amount>
       <contractid>2</contractid>
       <consumerid>5</consumerid>
   </entryParkTransaction>
   <exitParkTransaction>
       <transactionTime>2016-02-
18T16:15:41.000+01:00</transactionTime>
       <transactionType>12</transactionType>
       <facilityin>0</facilityin>
       <facilityout>2011</facilityout>
       <computer>2010</computer>
       <device>201</device>
       <amount>200.0</amount>
       <contractid>2</contractid>
       <consumerid>5</consumerid>
   </exitParkTransaction>
</bookParkTransaction>
```

If is in the element exitParkTransaction the amount element set to -1, the price is calculate through clip process.

3.3.6.6 Get Consumer identifiers

Get identifiers belonging to consumer.

This resource will provide the data only when ***extended pooling*** *functionality is activated in* CustomerAdministration *module.*

REST-Resource

HTTP-Method	GET
REST-URI	/consumers/{Cid},{id}/identifiers
Formats	Text, XML
Path parameters	Cid, id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500

Response body	<identifiers> element

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	
id	Consumer id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/consumers/49,1/identifiers

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<identifiers xmlns="http://gsph.sub.com/cust/types">
  <identifier>
    <ptcptId>1</ptcptId>
    <cardNo>56553654983</cardNo>
    <licensePlate>MG SR 850</licensePlate>
  </identifier>
  <identifier>
    <ptcptId>5</ptcptId>
    <cardNo>8545098</cardNo>
    <licensePlate>MG SR 160</licensePlate>
  </identifier>
</identifiers>
```

3.3.6.7 Update or create Consumer identifiers

Update or create new identifier data of a consumer.

This resource will allow changing the data only when ***extended pooling*** *functionality is activated in* CustomerAdministration *module.*

REST-Resource

HTTP-Method	PUT
REST-URI	/consumers/{Cid},{id}/identifiers
Formats	Text, XML
Path parameters	Cid, id
Query parameters	instanceId
Request body	<identifiers> element
HTTP Response codes	200, 204, 404, 500
Response body	<identifiers> element

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	
id	Consumer id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/consumers/49,1/identifiers

Example Request Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<identifiers xmlns="http://gsph.sub.com/cust/types">
  <identifier>
    <ptcptId>5</ptcptId>
    <cardNo>545490854</cardNo>
    <licensePlate>MG SR 165</licensePlate>
  </identifier>
  <identifier>
    <cardNo>210899508</cardNo>
    <licensePlate>MG SR 170</licensePlate>
  </identifier>
…
</identifiers>
```

3.3.6.8 Delete Consumer identifiers

Delete identifier data of a consumer.

This resource will allow deleting the data only when ***extended pooling*** *functionality is activated in* CustomerAdministration *module.*

REST-Resource

HTTP-Method	DELETE
REST-URI	/consumers/{Cid},{id}/identifiers/{Iid}
Formats	Text, XML
Path parameters	Cid, id, Iid
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 204, 404, 500
Response body	<identifiers> element

Path Parameters

Parameter	Short description	Format	Possible values
Cid	Contract id	number	
id	Consumer id	number	
Iid	Identifier id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/consumers/49,1/identifiers/5

3.3.7 Invoice Resources

3.3.7.1 Get a List of Invoices

Get a list of invoices. You can retrieve a partial list by adding parameters (invoice number range, date range) to the URI. To get a list of invoices related to a special contract use the according URI for the *contract* resource (e.g. …/contracts/5001/invoices).

REST-Resource

HTTP-Method	GET
REST-URI	/invoices
Formats	Text, XML
Path parameters	-
Query parameters	minId, maxId, minDate, maxDate, instanceId
Request body	-
HTTP Response codes	200, 204, 500
Response body	<invoices> element

Query Parameters

Parameter	Short description	Format	Possible values
minId	Minimum invoice id to retrieve	number	
maxId	Maximum invoice id to retrieve	number	
minDate	Minimum invoice date	YYYY-MM-DD	
maxDate	Maximum invoice date	YYYY-MM-DD	
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/invoices

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<invoices xmlns="http://gsph.sub.com/cust/types">
  <invoice href="/invoices/75">
    <id>75</id>
    <invcDate>2014-09-25T15:14:55.000+02:00</invcDate>
    <invcAmount>1200</invcAmount>
    <companyRef>31</companyRef>
    <companyUrl>contracts/31</companyUrl>
  </invoice>
  <invoice href="/invoices/74">
    <id>74</id>
    <invcDate>2014-04-08T08:23:07.000+02:00</invcDate>
    <invcAmount>4900</invcAmount>
    <companyRef>13</companyRef>
    <companyUrl>contracts/13</companyUrl>
  </invoice>
  …
</invoices>
```

3.3.7.2 Get an Invoice Detail

Get the detailed information about an invoice specified in {*id*}.

REST-Resource

HTTP-Method	GET
REST-URI	/invoices/{id}
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 404, 500
Response body	<invoiceDetail> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Invoice id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/invoices/75

Example Response Body (XML)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<invoiceDetail xmlns="http://gsph.sub.com/cust/types">
  <invoice href="/invoices/75">
    <id>75</id>
    <invcDate>2014-09-25T15:14:55.000+02:00</invcDate>
    <invcAmount>1200</invcAmount>
    <companyRef>31</companyRef>
    <companyUrl>contracts/31</companyUrl>
  </invoice>
  <fakturadate>2014-09-25T15:14:47.000+02:00</fakturadate>
  <paystatus>1</paystatus>
  <amontToPay>1200</amontToPay>
  <clearingflag>2</clearingflag>
  <remindercnt>0</remindercnt>
  <invoiceItem>
    <contractid>31</contractid>
    <consumerid>99999</consumerid>
    <articleid>403</articleid>
    <articleclass>1</articleclass>
    <articlename>SOE Nachbuchung 3</articlename>
    <quantity>1</quantity>
    <singleprice>0.0</singleprice>
    <totalamount>100.0</totalamount>
    <servicetime>2014-02-11T13:06:57.000+01:00</servicetime>
    <selltime>2014-02-11T13:06:38.000+01:00</selltime>
    <vattype>A</vattype>
    <vatamount>16.0</vatamount>
    <pointofsale>0</pointofsale>
  </invoiceItem>
  <invoiceItem>
    <contractid>31</contractid>
    <consumerid>99999</consumerid>
    <articleid>509</articleid>
    <articleclass>1</articleclass>
    <articlename>SOA Nachbuchung 9</articlename>
    <quantity>1</quantity>
    <singleprice>0.0</singleprice>
    <totalamount>-200.0</totalamount>
    <servicetime>2014-02-11T13:07:19.000+01:00</servicetime>
    <selltime>2014-02-11T13:07:07.000+01:00</selltime>
    <vattype>A</vattype>
    <vatamount>-32.0</vatamount>
    <pointofsale>0</pointofsale>
```

```
   </invoiceItem>
   <invoiceItem>
      <contractid>31</contractid>
      <consumerid>99999</consumerid>
      <articleid>91</articleid>
      <articleclass>1</articleclass>
      <articlename>Akonto umsatzw.</articlename>
      <quantity>1</quantity>
      <singleprice>0.0</singleprice>
      <totalamount>400.0</totalamount>
      <servicetime>2014-02-11T13:09:51.000+01:00</servicetime>
      <selltime>2014-02-11T13:07:39.000+01:00</selltime>
      <vattype>A</vattype>
      <vatamount>64.0</vatamount>
      <pointofsale>0</pointofsale>
   </invoiceItem>
</invoiceDetail>
```

3.3.7.3 Get an Invoice PDF

Get the pdf file about an invoice specified in {*id*}. The pdf file is transferred as string in context of response. The string was encoded with Base64 formel.

REST-Resource

HTTP-Method	GET
REST-URI	/invoices/{id}/pdf
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 404, 500
Response body	<invoicePDF> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Invoice id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/invoices/75/pdf

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?><invoicePDF
xmlns="http://gsph.sub.com/cust/types">
  <invoice href="/invoices/78">
    <id>78</id>
    <invcDate>2015-06-25T10:02:51.000+02:00</invcDate>
    <invcAmount>351000</invcAmount>
    <companyRef>2</companyRef>
    <companyUrl>contracts/2</companyUrl>
  </invoice>
  <pdfStringData>
JVBERi0xLjQKJeLjz9MKNCAwIG9iago8PC9MZW5ndGggMTkzMy9GaWx0ZXIv
RmxhdGVEZWNvZGU+
…
MjA1NGU3YjczY2I1ZDY0Pl0vSW5mbyAzMSAwIFIvU2l6ZSAzMj4+CnN0YXJ0e
HJlZgoxNTExNwol
JUVPRgo=
</pdfStringData>
</invoicePDF>
```

3.3.7.4 Create an Invoice

Sending a POST request to this URI creates a new invoicing job. This request initiates an asynchronous handling. If the job is created successfully by the server, the response contains a reference to the job created. The client can use this URI to retrieve the status of the job.

See also the API for the job resource (chapter 0 “ *Job Resources”).*

REST-Resource

HTTP-Method	POST
REST-URI	/invoices
Formats	Text, XML
Path parameters	-
Query parameters	minContractId, maxContractId, startDateTime, refDate, downPayment, bestPrice, instanceId
Request body	-
HTTP Response codes	201, 500
Response body	<invoiceJob> element

Query Parameters

Parameter	Short description	Format	Possible values
minContracId	Minimum contract id to be invoiced. (optional)	number	Minimum value is 1.

Parameter	Short description	Format	Possible values
maxContractId	Maximum contract id to be invoiced. (optional; required only if minContractId is set).	number	
startDateTime	Start date and time for invoicing job	YYYY-MM-DDTHH:MM:SS	
refDate	Reference date for invoice job	YYYY-MM-DDTHH:MM:SS	
downPayment	Take downpayments into account in this invoicing job.	number	0, 1
bestPrice	Take bestprice contract parameters into account in this invoicing job.	number	0, 1
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/invoices

Example Response Body (XML)

```
<?xml version = "1.0" encoding = "utf-8"?>
<invoiceJob xmlns:cm="http://gsph.sub.com/cust/types"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <jobId>10001</jobId>
  <jobType>invoice</jobType>
  <startTime>2010-12-24 18:00:00</startTime>
  <refDate>2010-12-24</refDate>
  <status>0</status>
  <invoiceRef>/invoices/5</invoiceRef>
  <invoiceRef>/invoices/6</invoiceRef>
  <invoiceRef>/invoices/7</invoiceRef>
</invoiceJob>
```

3.3.8 Usage Profiles Resources

3.3.8.1 Get a List of Usage Profiles

Get a list of all usage profiles.

REST-Resource

HTTP-Method	GET
REST-URI	/profiles

Formats	Text, XML
Path parameters	-
Query parameters	onlineAvailable, instanceId
Request body	-
HTTP Response codes	200, 204, 500
Response body	<usageProfiles> element

Query Parameters

Parameter	Short description	Format	Possible values
onlineAvailable	Get usage profiles: - all - not online available - online available	number	-1: get all usage profiles (default) 0: get only usage profiles without "onlineAvailable" flag 1: get only usage profiles with "onlineAvailable" flag
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/profiles

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?><usageProfiles
xmlns="http://gsph.sub.com/cust/types">
  <usageProfile href="/profiles/1/detail">
    <id>1</id>
    <name>Standardh</name>
  </usageProfile>
  <usageProfile href="/profiles/2/detail">
    <id>2</id>
    <name>testtg</name>
  </usageProfile>
  ..
</usageProfiles>
```

3.3.8.2 Get a Usage Profile Detail

Get a detail of usage profile.

REST-Resource

HTTP-Method	GET
REST-URI	/profiles/{id}/detail
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-

HTTP Response codes	200, 404, 500
Response body	<usageProfileDetail> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Usage profile id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/profiles/1/detail

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<usageProfileDetail xmlns="http://gsph.sub.com/cust/types">
  <usageProfile>
    <id>1</id>
    <name>Standardh</name>
    <description/>
  </usageProfile>
  <usageProfileSettings>
    <profileid>1</profileid>
    <versionid>2</versionid>
    <validFrom>2014-08-13T00:00:00.000+02:00</validFrom>
    <maxCount>0</maxCount>
    <flatrateamount>0</flatrateamount>
    <flatratevattype>A</flatratevattype>
    <bestabr>0</bestabr>
    <bestabrmin>0</bestabrmin>
    <bestabrmax>0</bestabrmax>
    <datafield1>Text in Data field 1</datafield1>
    <datafield2>Text in Data field 2</datafield2>
    <datafield3>Text in Data field 3</datafield3>
  </usageProfileSettings>
  <usageProfileSettings>
    <profileid>1</profileid>
    <versionid>1</versionid>
    <validFrom>2009-06-18T00:00:00.000+02:00</validFrom>
    <validUntil>2014-03-06T23:59:59.000+01:00</validUntil>
    <maxCount>0</maxCount>
<flatrateamount>5000</flatrateamount>
    <flatratevattype>A</flatratevattype>
    <bestabr>0</bestabr>
    <bestabrmin>0</bestabrmin>
    <bestabrmax>0</bestabrmax>
    <usageProfileFacilities>
      <profileid>1</profileid>
      <versionid>1</versionid>
      <facilityid>9999</facilityid>
      <timeprofile>0</timeprofile>
      <admittance>0</admittance>
      <inprofilerate>0</inprofilerate>
      <outprofilerate>0</outprofilerate>
      <countReserved>false</countReserved>
    </usageProfileFacilities>
  </usageProfileSettings>
</usageProfileDetail>
```

3.3.9 Resources of Templates

Contract templates and consumer templates are all stored in one database table together and therefore share the same range of IDs.

3.3.9.1 Get a List of Contract Templates

Get a list of all contract templates.

REST-Resource

HTTP-Method	GET
REST-URI	/templates/contracts
Formats	Text, XML
Path parameters	-
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 204, 500
Response body	<templates> element

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/templates/contracts

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<templates xmlns="http://gsph.sub.com/cust/types">
  <template>
    <id>99999</id>
    <name>Firmennachzahler</name>
  </template>
  <template>
    <id>6</id>
    <name>SEPA</name>
  </template>
  …
</templates>
```

3.3.9.2 Get a List of Consumer Templates

Get a list of all consumer templates.

REST-Resource

HTTP-Method	GET
REST-URI	/templates/consumers
Formats	Text, XML
Path parameters	-
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 204, 500
Response body	<templates> element

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/templates/consumers

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<templates xmlns="http://gsph.sub.com/cust/types">
  <template>
    <id>8</id>
    <name>Neuer Musterteilnehmer</name>
  </template>
  <template>
    <id>7</id>
    <name>Neuer Musterteilnehmer</name>
  </template>
  …
</templates>
```

3.3.10 Job Resources

3.3.10.1 Get a List of Jobs

Get a list of jobs. To retrieve only a partial list of jobs additional parameters can be added to the URI.

REST-Resource

HTTP-Method	GET
REST-URI	/jobs
Formats	Text, XML
Path parameters	-
Query parameters	status, type, instanceId
Request body	-
HTTP Response codes	200, 204, 500
Response body	<invoiceJobs> element

Query Parameters

Parameter	Short description	Format	Possible values
status	Job status	number	0: waiting 1: running 2: successful 3: failed
type	Job type	string	0: Invoice 1: Reduction 2: Reminder 3: Open positions 4: Repeated reminder 9: Clearing 10: Card booking 11: Payroll 12: SEPA clearing
instanceId	Instance id	number	

Example Request

https://hostname:port/CustomerMediaWebService/jobs

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<invoiceJobs xmlns="http://gsph.sub.com/cust/types">
  <invoiceJob>
    <jobId>10364</jobId>
    <jobType>0</jobType>
    <startTime>2014-10-08T11:00:00.000+02:00</startTime>
    <refDate>2014-10-07T23:59:59.000+02:00</refDate>
    <status>3</status>
  </invoiceJob>
  <invoiceJob>
    <jobId>10342</jobId>
    <jobType>0</jobType>
    <startTime>2014-08-16T02:00:00.000+02:00</startTime>
    <refDate>2014-08-15T23:59:59.000+02:00</refDate>
    <status>3</status>
  </invoiceJob>
  …
</invoiceJobs>
```

3.3.10.2 Get a Status of Job

Get detailed information about an invoice job. The id of the job is part of the URI. If a job with the specified id exists, the response body contains an *invoiceJob* element. If the job resource does not exist, the server responds with HTTP error 404.

The *invoiceJob* element contains information about the job status. If the job was finished successfully, it contains a list with URIs to related invoices.

REST-Resource

HTTP-Method	GET
REST-URI	/jobs{id}
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 404, 500
Response body	<invoiceJob> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Job id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/jobs/10

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<invoiceJob xmlns="http://gsph.sub.com/cust/types">
  <jobId>10</jobId>
  <jobType>0</jobType>
  <startTime>2014-10-08T11:00:00.000+02:00</startTime>
  <refDate>2014-10-07T23:59:59.000+02:00</refDate>
  <status>3</status>
</invoiceJob>
```

3.3.11 Batch Resources

3.3.11.1 Get a Batch Request Status

REST-Resource

HTTP-Method	GET
REST-URI	/batchRequest{id}
Formats	Text, XML
Path parameters	id
Query parameters	instanceId
Request body	-
HTTP Response codes	200, 404, 500
Response body	<batch> element

Path Parameters

Parameter	Short description	Format	Possible values
id	Batch id	number	

Query Parameters

Parameter	Short description	Format	Possible values
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/batchRequest/101

3.3.11.2 Create Contracts as a Batch Request

REST-Resource

HTTP-Method	POST
REST-URI	contracts/batchRequest
Formats	Text, XML
Path parameters	-

Query parameters	errorHandlingMode, templateId, instanceId
Request body	<contractDetails> element
HTTP Response codes	202, 500
Response body	<batch> element

Query Parameters

Parameter	Short description	Format	Possible values
errorHandlingMode	Modus of handling if error occurs.	number	0: rollback all changes and exit job by error 1: commit all changes and exit job by error 2: continue with next job if error occurs
templateId	Id of template used for creating contracts. Otherwise use id defined in property file (when not defined in property file, default contract template id 0 is used).	number	
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/batchRequest

3.3.11.3 Create Consumers as a Batch Request

REST-Resource

HTTP-Method	POST
REST-URI	consumers/batchRequest
Formats	Text, XML
Path parameters	-
Query parameters	errorHandlingMode, templateId, instanceId
Request body	<consumerDetails> element
HTTP Response codes	202, 500
Response body	<batch> element

Query Parameters

Parameter	Short description	Format	Possible values

Parameter	Short description	Format	Possible values
errorHandlingMode	Modus of handling if error occurs.	number	0: rollback all changes and exit job by error 1: commit all changes and exit job by error 2: continue with next job if error occurs
templateId	Id of template used for creating consumer. Otherwise use id defined in property file (when not defined in property file, default consumer template id 1 is used).	number	
instanceId	Instance id	number	

Example Request
https://hostname:port/CustomerMediaWebService/batchRequest

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<batchType xmlns="http://gsph.sub.com/cust/types">
      <id>1</id>
      <type>1</type>
      <status>2</status>
      <contracts>……..</contracts>
      <contractsErrors>…….</contractsErrors>
</batchType>
```

3.3.12 Week profiles Resources

3.3.12.1 Get a List of Week profiles

Get a list of week profiles from entervo.

REST-Resource

HTTP-Method	GET
REST-URI	/weekdayprogprof
Formats	Text, XML
Path parameters	-
Query parameters	-
Request body	-
HTTP Response codes	200, 204, 500

Response body	<invoices> element

Example Request
https://hostname:port/CustomerMediaWebService/weekdayprogprof

Example Response Body (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
< weekDayProgProfs xmlns="http://gsph.sub.com/cust/types">
   <weekDayProgProf>
    <id>3</id>
    <name>Wochenende</name>
  </weekDayProgProf>
 <weekDayProgProf>
    <id>11</id>
    <name>Wochentage</name>
  </weekDayProgProf>
   …
</weekDayProgProfs>
```

4 Appendices

4.1 Appendix 1: List of Interface Versions and Related Functionalities

Interface Version	Functionality
1.5.0	Default Invoice Addressee (Company)
1.5.2	R&D for carpark and leisure centre systems
1.6.0	SEPA Lastschrift (SEPA direct debit, SDD)
1.6.1	Function bug fixing
1.6.1 V2R3P1	Project specific adaptation of REST Customer & Media
1.7.0	Project specific adaptation of REST Customer & Media
1.7.1	Function bug fixing
1.8.0	Extension of Customer & Media Interface
1.8.1	Extension of SEPACreditorIdentification from statchar
1.8.2	□ Fill idType.getIdentificationType() with getPtcpt().getPermProfRef() □ Extension of Customer & Media Interface for Personalized Value Card and other □ Hash epan in STK REST interface □ SEPA: Validation of IBAN and BIC to avoid clearing issues □ R&D for carpark and leisure centre systems (without point 3)
1.8.3	R&D for carpark and leisure centre systems (point 3)
1.8.4	Function bug fixing
1.8.5	Function bug fixing
1.8.6	Extended by the "Fakturierungsgruppe", park/sales transaction search with second
1.8.7	Adaptation for the new CustomerService
1.9.1	Extension of Payment UI to handle free facilities
1.9.2	Function bug fixing
1.10.0	□ Permission for 3rd-party for CustomerMediaWebService □ Improved Errorcode handling
1.10.1	Function bug fixing
1.11.0	□ Add all editable fields inUI □ Cooreting a error handling
1.12.0	Optimize the communication with service

Interface Version	Functionality
1.13.0	□ Function bug fixing □ Performance of communication with service □ Search functionality Licence plate
1.13.1	Encrypted Transfer of Credit Card Data between REST and Service
1.14.0	□ Neutralize card □ Manual Booking of park transactions and sales transactions
1.21.0	Pooling

4.2 Appendix 2: List of Parameters

*For more details see the **customer_media.xsd** file.*

Parameter	Short description	Format	Possible values
contractid	Contract id	number	2-49999: local contracts 50000-99999: global contracts Value 1 is reserved
consumerid	Consumer id	number	1-99998 Value 99999 is reserved
counting	Company counting information	number	1-9998 Value 9999 means that contract counting is turned off.
countReserved		boolean	false: count not reserved true: count reserved
device	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
extProfileId	Access profile id	number	
facilityId	Facility id	number	
flatratevattype	VAT type of flat rate	string	A, B, C, D, E, F, Z
identificationType	Card type	number	(see chapter 4.6 “Appendix 6: List of Card Types”)

Parameter	Short description	Format	Possible values
instanceId	Use of the parameter in request: *"…/contracts?instanceId=9112"* If there is no *instanceId* query parameter in request, the service affects local data on ZR or global data on LR.	number	1000-8999: ZR instances 9000-99999: LR instance
jobId	Job id	number	
jobType	Job type	string	0: Invoice 1: Reduction 2: Reminder 3: Open positions 4: Repeated reminder 9: Clearing 10: Card booking 11: Payroll 12: SEPA clearing
name	Name of: - contract holder - consumer - contract template - consumer template	string [33]	
present	Number of present consumers associated to contract.	number	
present		boolean	true, false
sepamandatstatus		number	0: Initial SEPA Bank Collection 1: Subsequent SEPA Bank Collection 2: Last SEPA Bank Collection 3: Single SEPA Bank Collection
transactionType	Transaction type	number	(see chapter 4.3 "Appendix 3: Transaction Types")
transactType	Transaction type	number	(see chapter 4.3 "Appendix 3: Transaction Types")
vattype	VAT type	string	A, B, C, D, E, F, Z

4.3 Appendix 3: 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
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
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

4.4 Appendix 4: List of Status Response Codes

Status Code	Result	Message / Description	HTTP Body
200	OK	The request has succeeded.	<data> element
201	Created	The request has been fulfilled and resulted in a new resource being created.	<data> element

Status Code	Result	Message / Description	HTTP Body
204	No Content	The server has fulfilled the request but there is no response data for requested resource.	
401	Unauthorized	The request requires user authentication.	<error> element
404	Not Found	The server has not found anything matching the Request-URI. The requested resource does not exist.	<error> element
500	Internal Server Error	The server encountered an unexpected condition which prevented it from fulfilling the request.	<error> element

4.5 Appendix 5: List of Error Descriptions

Error Code	Error Message / Error Description
10000	Not implemented / The function is still not implemented.
10001	Dataset not created / The dataset could not be created due to an internal error.
10002	Dataset already exist / The dataset could not be created because it exists already.
10003	Dataset not deleted / The dataset could not be deleted due to an internal error.
10004	Dataset does not exist / The dataset could not be found.
10005	Unique field violation / Entry in user field already exists.
10006	Connection error / Connection to service failed.
10007	Dataset not changed / The dataset could not be changed due to an internal error.
10008	Invalid status flag value / Invalid status flag value
10009	Dataset already set / The dataset could not be set due is already set.
10010	Free Facilities / Free Facilities is not activated.
10011	Internal error / Internal error occure on service. See tomcat and XMS30 logs.

Error Code	Error Message / Error Description
10020	Invalid parameter / Customer service parameter error
10021	Invalid parameter / The URL parameter 'minId' is invalid.
10022	Invalid parameter / The URL parameter 'maxId' is invalid.
10023	Invalid parameter / The URL parameter 'validFrom' is invalid.
10024	Invalid parameter / The URL parameter 'validUntil' is invalid.
10025	Invalid parameter / The URL parameter 'contractId' is invalid.
10026	Invalid parameter / The internal parameter 'instanceId' is invalid.
10027	Invalid parameter / The internal parameter 'operatorId' is invalid.
10028	Invalid parameter / The URL parameter 'startDateTime' is invalid.
10029	Invalid parameter / The URL parameter 'refDate' is invalid.
10030	Invalid parameter / The URL parameter 'flatRate' is invalid.
10031	Invalid parameter / The URL parameter 'bestPrice' is invalid.
10032	Invalid parameter / The URL parameter 'minContractId' is invalid.
10033	Invalid parameter / The URL parameter 'maxContractId' is invalid.
10034	Invalid parameter / The URL parameter 'consumerId' is invalid.
10035	Invalid parameter / The URL parameter 'minDate' is invalid.
10036	Invalid parameter / The URL parameter 'maxDate' is invalid.
10037	Invalid parameter / The URL parameter 'invoiceId' is invalid.
10038	Invalid parameter / The URL parameter 'status' is invalid.
10039	Invalid parameter / The URL parameter 'type' is invalid.

Error Code	Error Message / Error Description
10040	Invalid parameter / The URL parameter 'id' is invalid.
10041	Invalid parameter / The internal parameter 'contractTemplateId' is invalid.
10042	Invalid parameter / The internal parameter 'consumerTemplateId' is invalid.
10043	Invalid parameter / The internal parameter 'validFromContract' is invalid.
10044	Invalid parameter / The internal parameter 'validUntilContract' is invalid.
10045	Invalid parameter / The internal parameter 'validFromConsumer' is invalid.
10046	Invalid parameter / The internal parameter 'validUntilConsumer' is invalid.
10047	Invalid parameter / The internal parameter 'errorHandlingMode' is invalid.
10048	Security violation / The dataset is used internal only, access is not allowed.
10049	Invalid parameter / The URL parameter 'exported' is invalid.
10050	Invalid parameter / The URL parameter 'exported' is invalid.
10051	Invalid parameter / The URL parameter 'contractNo' is invalid.
10052	Invalid parameter / The URL parameter 'delete' is invalid.
10053	Invalid parameter / The URL parameter 'cardNo' is invalid.
10054	Invalid parameter / The URL parameter 'ignorePresence' is invalid.
10055	Invalid parameter / The parameter 'sellTime' is invalid.
10056	Invalid parameter / The parameter 'artlass' is invalid.
10057	Invalid parameter / The parameter 'article' is invalid.
10058	Invalid parameter / The URL parameter 'Servicetime' is invalid.
10059	Invalid parameter / The internal parameter 'sepaiban' is invalid, wrong length.

Error Code	Error Message / Error Description
10060	Invalid parameter / The internal parameter 'sepaiban' is invalid, wrong check sum.
10061	Invalid parameter / The internal parameter 'sepabic' is invalid, wrong length.
10062	Invalid parameter / The internal parameter 'sepabic' is invalid, wrong form.
10063	Invalid parameter / Not unique parameter sepamandatref.
10064	Invalid parameter / Card number is already in use.
10065	Invalid parameter / Rabat profile in use
10066	Invalid parameter / Range for shadow invoice address is full.
10067	Invalid parameter / The URL parameter 'extProfileId' is invalid.
10068	Invalid parameter / The URL parameter 'onlineAvailable' is invalid.
10069	Wrong template id / The URL parameter 'templateId' or Template parameter in properties does not exist in system!
10070	Wrong createdBy / This contract was created with another createdBy!
10071	Wrong park transaction booking / The entry time is after exit time!
10072	Invalid parameter / Provider range for invoice address is full
10073	Invalid parameter / The URL parameter 'facilityId' is invalid
10074	Invalid parameter / The URL parameter 'identifierId' is invalid
20001	Validity inconsistent / The contract valid until date must be at least one day after the valid from date.
20002	Required fields error / The required fields must not be empty or invalid (name, street, town, postbox).
20003	Contract data not valid / The contract input dataset is not valid.
20004	Contract data does not exist / The contract data could not be found.

Error Code	Error Message / Error Description
20005	Required SEPA fields error / The required SEPA fields must not be empty or invalid (sepamandatref, sepamandatdate, sepamandatstatus).
20006	Pooling data error / Facility not exists
20007	Pooling data error / Pooling profile not defined
20008	Pooling data error / Extended pooling functionality not activated
20009	Pooling data error / Allow car change pooling type forbidden
20010	Pooling data error / Allow car change pooling type not set
20011	Pooling data error / "Soft pooling B" or "Allow car change" cannot be combined with other pooling types
30001	Validity wrong / The validity of the consumer exceeds the validity of the contract.
30002	Validity inconsistent / The consumer valid until date must be at least one day after the valid from date.
30003	Card number already exist / The card number already exists.
30004	Card profile user / The card profile is not active.
30005	Card profile global / The maximum number of references on this global card profile is exceeded.
30006	Card profile user / The maximum number of references on this user card profile is exceeded.
30007	Card profile user / The card profile is not allowed.
30008	Required fields error / The required fields must not be empty or invalid (name).
30009	Consumer data not valid / The consumer input dataset is not valid.
30010	Consumer data does not exist / The consumer data could not be found.
30011	Consumer card no does not exist / The consumer card number is mandatory for this configuration

Error Code	Error Message / Error Description
30012	Consumer identifier does not exist / The consumer identifier data could not be found
30013	Consumer identifier error / The consumer identifier cannot be deleted
30014	Consumer identifier error / The consumer cannot be created
40001	Invoice wrong / The invoice not exists!
40002	Free facility wrong / The free facility not exists!
40003	Template wrong / The template not exists!
40004	Job wrong / The job not exists!
40005	Invoice wrong / The invoice not exists!
40006	Batch wrong / The batch not exists!
40007	Usage profile wrong / The usage profile not exists!
40008	Booking park transaction wrong / The input data wrong!
40009	Booking park transaction wrong / The entry time is after exit time!
40010	Booking park transaction wrong / Error by booking the park transaction on customer service!
40011	Booking sales transaction wrong / The input data wrong!
40012	Booking sales transaction wrong / Error by booking the sales transaction on customer service!

4.6 Appendix 6: List of Card Types

Code	Description
10	Standard parker
21	LTP – Side stripe
22	LTP – Transpodner offline
23	LTP – Barcode paper
51	LTP – Tagmaster
52	LTP – Barcode plastic

Code	Description
53	LTP – Middle stripe
54	LTP – Transponder online
55	LTP – No S&B card
56	LTP – LPN
60	Participant
70	Organizer
80	FPP customer

4.7 Appendix 7: XML vs. GUI – contract attributes name differences

XML element name	GUI attribute name
firstName	Name
surname	Name 2
postBox	ZIP Code

4.8 Appendix 8: XML vs. GUI – consumer attributes name differences

XML element name	GUI attribute name
lpn1	License plate
lpn2	2nd LPN
lpn3	3rd LPN
surname	Last name
identificationType	Card type
ptcptType	Participant type
usageProfile	Access profile

4.9 Appendix 9: Default contract template parameters

XML element name	GUI attribute name	Mandatory	Default value
title	Title	no	Firma
firstName	Name	yes	Default Rechnungs-empfänger
street	Street	yes	Musterstr. 1
postbox	ZIP Code	yes	98765

XML element name	GUI attribute name	Mandatory	Default value
town	City	yes	Musterhausen
xValidFrom	Start date	yes	-
xValidUntil	End date	yes	-

4.10 Appendix 10: Default consumer template parameters

XML element name	GUI attribute name	Mandatory	Default value
surname	Last name	yes	Default Teilnehmer
firstName	First name	no	TN
ptcptType	Participant type	yes	2
identificationType	Card type	no	21
validFrom	Valid from	yes	-
validUntil	Valid to	yes	-
usageProfile	Access profile	yes	1

5 Record of Revisions

Revision	Date	Description	Author
-	28/04/2014	Creation of document for software version 1.8.4	P. Kubánek
1.1	22/10/2014	REST C&M, ver. 1.8.7	P. Kubánek
1.2	04/12/2014	REST C&M, ver. 1.9.1 Contract Free Facilities	P. Kubánek
1.3	20/01/2015	Small syntax changes	P. Kubánek
1.4	06/03/2015	Revision	J. Juritka
1.5	20/10/2015	Revision Document adapted for the new template Chapter 1.1 “Interface Overview” added Chapter 4.1 “Appendix 1: List of Interface Versions and Related Functionalities” added Chapter 4.2 “Appendix 2: List of Parameters” added Chapter 4.3 “Appendix 3: Transaction Types” added Chapter 0 “ Appendix 4: List of Status Response Codes” added (replacement for “HTTP response tables” within particular chapters) “Appendix A” changed to chapter 4.5 “Appendix 5: List of Error Descriptions” Grammar revision	J. Srponova
1.5_1	23/02/2016	The *lpn* parameter and information about it added to chapters 3.3.4 “Contract Data Resources” and 0 “	J. Srponova
1.6.0	04/08/2016	- Neutralise a consumers - Add a create a consumer sales/park transaction - Add a create a contract sales transaction - Add @QueryParam("templateId") for creating a contracts / batchRequest and consumers / batchRequest - Add invoices/{id}/pdf	P. Kubánek

2.0	08/09/2017	Moved to SharePoint	J. Srponova
2.1	14/09/2016	Add a weekdayprogprof resource	P. Kubánek
	26/09/2017	REST C&M, ver. 1.17.2@0	P. Kubánek
3.0	20/11/2017	*InstanceId* parameter added to query parameters and its description to chapter 4.2 "Appendix 2: List of Parameters" Minor text corrections	J. Srponova
	10/07/2018	REST C&M, ver 1.21.0@0	M. Hrín
3.1	10/07/2018	*pooling* and *poolingDetail* elements added to *contractDetail* element in chapter 3.3.3.2	M. Hrín
3.4	18/10/2018	*dataField1-3* elements added to *usageProfileSettings* element in chapter 3.3.8.2	M. Hrín
3.5	28/02/2019	REST C&M, ver 1.23.0@0 New subresources: *dynamiccounters* and *poolingentries*, to read, update and delete contract`s pooling data	M. Hrín
3.6	04/12/2019	REST C&M, ver 1.24.0@0 New subresource: *identifiers*, to create, read, update and delete consumer`s identifier data	M. Hrín
3.8	21/07/2020	REST C&M, ver 1.25.2@0 - Chapter 4.6 "Appendix 6: List of Card Types" added - JSON format not supported, fix is in process - Time zone information removed from date (without time) type parameters - Information about shared contract and consumer template id range added to chapter 3.3.9 "Resources of Templates" - Information about default template id for contract and consumer added - Information about id ranges (consumer, local contract, global, contract), auto-generating of id if not defined in request body added to chapters 3.3.3.2 "Create a Contract" and 3.3.4.3 "Create a Contract Consumer" - Parameters <xValidFrom> and <xValidUntil>, which belong to <consumer> tag, no more used	M. Hrín

		in request/response, if in the same request/response used also parameters <validFrom> and <validUntil>, which belong to <identification> tag (because both contain the same information) - Information and examples of creating local data on LR, and global data on ZR added to chapters 3.3.3.2 "Create a Contract" - XML namespace "cm:" not required in any request body - Chapter 4.7 "Appendix 7: XML vs. GUI – contract attributes name differences" added - Chapter 4.8 "Appendix 8: XML vs. GUI – consumer attributes name differences" added - Chapter 4.9 "Appendix 9: Default contract template parameters" added - Chapter 4.10 "Appendix 10: Default consumer template parameters" added	