

TECHNICAL SPECIFICATION

**Materna Provisioning MiddleWare
- PMWpartner SOAP interface**

Document version 25.0

Obsah

Implementation environment	5
URL and authentication	5
Methods - subscriber management	6
configuration()	6
partnerList()	7
partnerGet()	7
customerSearch()	8
customerGet()	9
customerCreatePrepaid()	9
customerCreatePostpaid()	10
customerSetPostpaidLimit()	11
customerSet()	12
customerRemove()	13
customerNoteSet()	14
consumerSearch()	14
consumerCreate()	15
consumerSet()	15
consumerRemove()	16
gsSearch()	16
gsAdd()	16
gsRemove()	17
gsRemoveCancel()	18
gsTariffChange()	18
gsSpendingLimitChange()	19
componentAdd	19
componentAddCustom	20
componentRemove	21
componentParameterSet	21
getAvailableMSISDNs()	22
simAssign()	23
simMigrate()	23
simProfileGet()	24
simProfileSet()	24
simSuspend()	25
simResume()	26
simRemove()	26
simSwapMSISDN()	27
simSwap()	27
edrGet()	27
imsiCommonPartGet()	28
iccidCommonPartGet()	28
updateSimServices()	29
selfCarePassword()	29
selfCarePasswordSet()	30
selfCareMasterSet()	30
emailAddressList()	30
emailAddressSet()	31
eventTypeList()	32
edrTriggerList()	32
edrTriggerSet()	33
autoTopupSet()	34
autoPaymentRemove()	35
paymentHistoryInfo()	35
Methods - SIM card logistics	36
simStockOverview()	36

simInfo()	36
findSim()	36
Methods - product management	38
tariffList()	38
pricelistGet()	38
simServiceList()	40
smsNotificationTemplateList()	41
smsNotificationTemplateSet()	42
portOutList()	42
portOutConfirm()	43
portInList()	43
portInDelete()	45
portInOverExistingMsisdn()	45
Methods - billing and invoicing	46
invoiceList()	46
invoiceGet()	46
invoicePaymentStatusSet()	46
cdrGet()	47
cdrUsageTypeList()	48
customerInvoiceList()	48
customerInvoiceGet()	49
customerTaxDocuments()	52
customerGPCSlog()	52
customerSetDueDays()	54
customerContractGet()	54
customerContractCreate()	54
customerContractSet()	55
paperInvoiceList()	55
exportPaperInvoiceList()	55
paperInvoiceSet()	56
Methods - packages	57
packagesAvailable()	57
packageOrder()	57
packagesList()	58
packageRemove()	60
gsFreeUnits()	60
dataUsage()	61
Methods - single-sign-on	62
customerSSOtokenGet()	62
partnerSSOtokenGet()	62
Methods – Component Product (Product Q)	63
cpComponentCategoryList()	63
cpOrderList()	63
cpOrderGet()	65
cpOrderCreate()	68
cpOrderChangeStatus()	70
cpOrderSet()	71
cpOrderHistory()	72
cpOrderComponentCreate()	72
cpOrderComponentChangeStatus()	73
cpOrderComponentSet()	74
cpOrderComponentHistory()	75
cpOrderCustomItemCreate()	76
cpOrderCustomItemSet()	76
cpOrderCustomItemRemove()	77
cpOrderInvoiceSet()	77
cpOrderInvoiceDownloadInvoice()	79
cpOrderInvoiceSetPaid()	79
cpOrderInstallationList()	80
cpOrderInstallationGet()	81

cpOrderInstallationCreate()	83
cpOrderInstallationSet()	84
cpOrderInstallationChangeStatus()	84
cpOrderInstallationHistory()	85
cpOrderInstallationComponentCreate()	86
cpOrderInstallationComponentChangeStatus()	86
cpOrderInstallationComponentHistory()	87
cpOrderTaskList()	87
cpBundleTariffList()	89
cpComponentSubcategoriesList()	89
cpComponentLocationList()	90
cpComponentPortSim()	92
cpComponentVerifyAvailability()	92
cpTariffRetailWholesale()	92
cpIsMsisdnAvailableForRingin()	93
cpIptvGetAvailablePackages()	94
cpCpeLoginAvailable()	94
addSimToBasket()	95
getSimBasket()	95
removeSimFromBasket()	95
setSimBasket()	96
getOrderSimBaskets()	96
cpSendCustomerNotification()	96
cpOrderOnetimePriceGet()	97
cpOrderPayOnetimeItems()	97
cpCalculatePrice()	98
cpOrderCopy()	98
cpFeasibilityCheck()	99
cpRuainFind()	100
providerTaskChangeStatus()	100
providerTaskCreate()	101
providerTaskGet()	101
providerTaskList()	103
Methods – Service orders (ITF)	105
serviceOrderList()	105
serviceOrderSet()	107
XML data types	108
customerInfo	108
customerNote	111
paymentHistoryInfo	112
Component product implementation notes	113
RT	113
Email templates	113
Errors (SOAP faults)	115
Errors common to all methods	117
Error report example	117
Request-response log	118
Workflow engine (activiti)	118
History of changes	119

Implementation environment

database connections

1. MCD and BSS - readwrite access
2. Log - "writeonly" access (*non-critical for availability*)
3. GPCS - readwrite access
4. Activiti (no direct PMWpartner access needed)
5. Dando - readonly access
6. Billing - readonly access

key requirements

Lze pracovat s MCD customers, (de)aktivovat služby, přidávat/ubírat čísla a MSISDN
Zabezpečení na úrovni IP whitelistu a zároveň http-basic auth (username+password)
Použitelnost (rozšiřitelnost) rozhraní i pro non-mobile existující služby

URL and authentication

URL for PMWpartner interface has fixed and variable parts. Fixed parts are the same for all installations. Variable part may differ installation by installation.

SOAP XML interface (HTTP POST) address:

URL1 <https://<hostname>/pmwpartner/pmwpartner.soap.fcgi>

WSDL document is available for download at:

URL2 <https://<hostname>/pmwpartner/pmwpartner.soap.fcgi?wsdl>

PDF download (invoices, see invoiceList method)

URL3 <https://<hostname>/pmwpartner/pmwpartner.pdf.fcgi?invoice-p502-g8952-20140501-4400125567.pdf>

PDF file name -----+

HTML file download (tax documents)

URL4 <https://<hostname>/pmwpartner/pmwpartner.taxdocproxy.fcgi?130100010>

TaxDocument number -----+

Authentication

- 1) IP whitelist. Each partner must provide list of IPv4 addresses or ranges to enable in firewall.
- 2) HTTP-basic username+password authentication. These credentials are provided by MATERNA to MVNO partner.

Implementation notes

```
$ iptables -A PMWWHITELIST -p tcp --dport 443 -j ACCEPT -s 89.203.254.10 # partner 2 "foobar"
$ iptables -A PMWWHITELIST -p tcp --dport 443 -j ACCEPT -s 73.55.0.128/29 # partner 3 "bahbaz"
$ iptables -A PMWWHITELIST -j REJECT
```

.htaccess

AuthType Basic

AuthUserFile /opt/mc/pmwpartner/cgi-bin/.htpasswd

AuthName "MATERNA Communications - PMWpartner interface"

require valid-user

.htpasswd

partner1:\$apr1\$jJSTgkcP\$zx4VUzuVSMcnGyrzzuA3c1

partner2:\$apr1\$NqU57PKO\$!SlwapoOXGJsY9KlvZcmd/

partner3:\$apr1\$KmHADQhM\$A6M7SAjm2veJ4D73nbopF1

Methods - subscriber management

Authentication: http-basic username "mvno<X>", password in .htpasswd file
X ... partnerID number as in MCD table PARTNER.

Each call must pass the check of presence of <X> in table PARTNER.

configuration()

Returns the configuration of the PMW partner interface.

Input parameters

None

Output parameters

parameter	M/O	data type	explanation, example
parameters	M		
parameter	0..N	[]	
name	M	string	
value	M	string	

Available parameters

name	data type	explanation, example
platform/CPE_LOGIN_PREFIX	string	Common prefix all logins to CPEs (routers) handled by Cisco/Radius must start with.
componentProduct/collectiveBillingDefault	boolean	Whether the collective billing is enabled (1) or disabled (0) by default.
componentProduct/collectiveBillingPerOrder	boolean	Whether the collective billing can be changed per order.

Implementation

Load from BSS.c_configuration and pmw-partner.conf.

Example

```
<parameters>
  <parameter>
    <name>platform/CPE_LOGIN_PREFIX</name>
    <value>materna</value>
  </parameter>
  <parameter>
    <name>componentProduct/collectiveBillingDefault</name>
    <value>0</value>
  </parameter>
  <parameter>
    <name>componentProduct/collectiveBillingPerOrder</name>
    <value>1</value>
  </parameter>
</parameters>
```

partnerList()

Returns partners.

Allowed roles

PlatformOwner

Input parameters

parameter	M/O	data type	explanation, example
role	O	enum	Reseller, HostServiceProvider, ServiceOrganization Filter by role.

Output parameters

parameter	M/O	data type	explanation, example
partners	M		
partner	0..N	[]	
id	M	long	
name	M	string	
shortName	M	string	
roles	M		
role	0..N	string[]	Reseller, HostServiceProvider, ServiceOrganization
parameters	M		
parameter	0..N	[]	
name	M	string	Name of parameter
value	M	string	Value of parameter

Implementation

Load from MCD.PARTNER, MCD.PARTNER_ROLE.

Don't return partner with role PlatformOwner.

If partner has only role Reseller, return it only if its MCD.PARTNER_PARAMETER "PLATFORM_OWNER_ID" matches the calling partner.

partnerGet()

Returns info about the authenticated partner.

Input parameters

None

Output parameters

parameter	M/O	data type	explanation, example
id	M	long	
name	M	string	
shortName	M	string	
roles	M		

role	0..N	string[]	PlatformOwner, Reseller, HostServiceProvider, ServiceOrganization, MVNO
parameters	M		
parameter	0..N	[]	
name	M	string	Name of parameter
value	M	string	Value of parameter

Implementation

Load from MCD.PARTNER, MCD.PARTNER_ROLE.

customerSearch()

Finds customer data by various search criteria including MSISDN.

Input parameters

parameter	M/O	data type	explanation, example
searchKey	M	UTF8 string	Part of customer's name, last name, company name, MSISDN, IMSI or ICCID
tariffId	O	integer	Limit to customers, that have at least one GID with this tariff.
limit	O	integer	Maximum number of matching customers to get. Default = maximum = 100.
active	O	boolean	Filter only active (1), blocked (0) customers. If the parameter is not present, all customers are returned.
searchKeyField	O	UTF8 string	List of fields over which the search is applied. Value is any combination of the strings: name, company, sim, msisdn with any separator (coma is preferred). Default searching over all fields is applied if the parameter is not set or not valid values are set. The meanings of the values: - name = customer's name or last name - company = company name - sim = IMSI or ICCID - msisdn = MSISDN - Examples: - name,sim - msisdn;sim;company

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
numFound	integer	Number of matching customers according to searchKey. 0 ... nothing found
numShown	integer	Number of matching customers returned in custList; this is affected by limit (maximum 100)
custList	List of customerInfo	customer1 { customerInfo } customer2 { customerInfo } ... customer100 { customerInfo }

Errors

MissingArgumentException, InvalidArgumentException, ArgumentOutOfRangeException

customerGet()

Gives detailed information about customer, services and MSISDNs.

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification
gid	O	integer	Limit output to given generic service (id) only

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
custInfo	customerInfo	See WSDL complexType definition <xs:complexType name="customerInfo">

Errors

MissingArgumentException, InvalidArgumentException, ArgumentOutOfRangeException, CustomerNotFoundException, CustomerBlockedException, GenericServiceNotFoundException, GenericServiceDeactivatedException, NoCustomerFound

customerCreatePrepaid()

Creates new customer (subscription) and gives MSISDN¹. Customer personal data (name, company name, addresses, e-mail) may be provided using subsequent customerSet() call.

Instant SIM card activation is optional - depends on "imsi" input parameter.

Subscription will be set as pre-paid. Startup balance of pre-paid account may be set per bussines case.

This is convenience method for customerCreate and gsCreate.

Implementation notes:

1. inspire in prepaid_account_create.pl
2. create CUSTOMER with no personal details (anonymous)
3. create one MCD GENERIC_SERVICE (GID); add _tech type 13 "Mobile"
4. create BSS s_services record with serviceClass 2.7
 - create EDR (category CHG, operation TARIFF - nová hodnota)
1. allocate one MSISDN from pool of partner's available numbers (SELECT ID,NUMBER FROM GLOBAL_NUMBERING_PLAN WHERE PARTNER_ID=<X> AND GID is null ORDER BY CATEGORY,WHEN_USED,NUMBER LIMIT 1)
 - create EDR SUB/START
1. set all default m_* parameters into NUMBER_PARAMETER
2. create GPCS account, preferable number = short MSISDN (without country prefix)
3. run (asynchronous) workflow **subscriptionActivate(msisdn,imsi)**

Input parameters

parameter	M/O	data type	explanation, example
tariffId	M	integer	Which tariff (product+pricelist) to use. Use one of tariffs offered to partner, as returned by tariffList()
preferredMSISDN	O	string	Preferred new MSISDN to use. See getAvailableMSISDNs() method. If preferredMSISDN is used, and is not available, error is returned. Preferred MSISDN may not be available for chosen tariff.
withoutMSISDN	O	boolean	No MSISDN (no SIM) is assigned. Use this to create "empty" customer for migration of MSISDN from another existing customer. See simMigrate() method.
imsi	O	string	Identification of SIM that has to be assigned to new subscription. Only one of IMSI (14-15 digits) or ICCID (19-20 digits) may be used. If not specified, manual task is created on Logistics interface and SIM is assigned by human.
iccid	O	string	

(M)andatory, (O)ptional, (D)epends on others

¹but not with usage of withoutMSISDN=true

Output parameters

parameter	data type	explanation, example
customerId	integer	Numeric customer identification Example: 4107
gid	integer	Numeric identification of generic service Example: 11502
gpcsAccount	string	Number of GPCS account (variable symbol for bank wire transfers), up to 10 decimal digits Example: 781014255
msisdn	string	Allocated new MSISDN, E.164 format Example: 420781014255
availableBalance	decimal(12,4)	Available (initial) pre-paid account balance for customer. This is set by MVNE according to your business case. Example: 500.0000
availableBalanceCurr	string	3-digit currency code for availableBalance. Example: CZK

Event Data Records

New subscription produces EDR of category SUB, operation START. *(but not if withoutMSISDN=true)*

Errors

MissingArgumentException, InvalidArgumentException, TariffNotFoundException, NoFreeSimException, NoPreferredFreeMsisdnException, NoFreeMsisdnException, ImsiNotAvailableException

customerCreatePostpaid()

Creates new customer (subscription) and gives MSISDN². Additional details (name, e-mail, address) should be provided using subsequent customerSet() call.

Instant SIM card activation is optional - depends on "imsi" input parameter.

This is convenience method for customerCreate and gsCreate.

Implementation notes:

- check that at least (name+lastname) or (companyName+companyIdNum) is present, otherwise error (mandatory parameters missing)
- create CUSTOMER with personal/company details
- create one MCD GENERIC_SERVICE (GID); add _tech type 13 "Mobile" (skip if tariffId is not specified)
- create BSS s_services record with serviceClass 2.7; set tariffId; do not set finDocGroup (see below) (skip if tariffId is not specified)
 - create EDR (category CHG, operation TARIFF - nová hodnota)
- allocate one MSISDN from pool of partner's available numbers or use preferredMSISDN (skip if withoutMSISDN)
 - create EDR SUB/START
- set all m_* parameters into NUMBER_PARAMETER (skip if withoutMSISDN)
- create "main" GPCS account for monthly limits, preferable number = short MSISDN (without country prefix), class 30
 - set floor according to monthlyLimit, which is defined in BSS s_productOptions, option CREDITLIMIT
 - use currency from partner's DEFAULT_CURRENCY (database MCD, table PARTNER_PARAMETER)
- run (asynchronous) workflow **subscriptionActivate (msisdn,imsi)** (skip if withoutMSISDN)

Input parameters

parameter	M/O	data type	explanation, example
tariffId	0	integer	Which tariff (pricelist) to use. Use one of tariffs offered to partner, as returned by tariffList()

²but not with usage of withoutMSISDN=true

personName	O	string	First name of customer (personal)
personLastName	O	string	Last name of customer (personal)
companyName	O	string	Company name
companyIdNum	O	string	Company registration number (ičo)
preferredMSISDN	O	string	Preferred new MSISDN to use. See getAvailableMSISDNs() method. If preferredMSISDN is used, and is not available, error is returned. Preferred MSISDN may not be available for chosen tariff.
withoutMSISDN	O	boolean	No MSISDN (no SIM) is assigned. Use this to create "empty" customer for migration of MSISDN from another existing customer. See simMigrate() method.
imsi	O	string	Identification of SIM that has to be assigned to new subscription. Only one of IMSI (14-15 digits) or ICCID (19-20 digits) may be used. If not specified, manual task is created on Logistics interface and SIM is assigned by human.
iccid	O	string	

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
customerId	integer	Numeric customer identification Example: 4107
gid	integer	Numeric identification of generic service. Returned only if tariffId is specified. Example: 11502
msisdn	string	Allocated new MSISDN, E.164 format. Returned if withoutMSISDN is not specified. Example: 420781014255

Event Data Records

New subscription produces EDR of category SUB, operation START. *(but not if withoutMSISDN=true)*

Errors

MissingArgumentException, InvalidArgumentException, ArgumentDependencyException, TariffNotFoundException, NoPreferredFreeMsisdnException, NoFreeMsisdnException, MandatoryProductOptionNotFoundException, ImsiNotAvailableException

customerSetPostpaidLimit()

Sets monthly limit for post-paid customer. Note that setting lower limit may stop the service for customer.

Monthly spending of particular customer is determined on the fly. When spending value reaches or exceeds monthlyLimit, SIM card is temporarily blocked by simSuspend().

Implementation notes:

- check that new limit is >0 (must not be 0)
 - return the warning if new creditlimit is < than -available_balance (because services will be blocked)
- check that new limit is < CREDITLIMIT_MAX³; include value of maximum in error message
- check that new limit is < CREDITLIMIT_MAX⁴; include value of maximum in error message
- change GPCS table cards_balance, column floor; set to -monthlyLimit
- currency of account - get from MCD PARTNER_PARAMETER named DEFAULT_CURRENCY
- make record in GPCS table card_log, action=UPDATE, balance_before=balance_after=balance, put the information in 'info' text field like "floor change from -3500 to -4500"

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification

³defined in bss.s_productOptions

⁴defined in mcd5.PARTNER_PARAMETER

monthlyLimit	M	decimal(10,4)	Monthly limit (maximum value of services used) for customer's traffic. Default credit limit for new postpaid customers is set in partner's profile. The billing currency is set by partner's profile. Monthly limit should not exceed product's CREDITLIMIT_MAX. Example: 1250
--------------	---	---------------	---

(M)andatory, (O)ptional, (D)epends on others

Output Parameters

none

Errors

MissingArgumentException, InvalidArgumentException, ArgumentOutOfRangeException, CustomerNotFoundException, CustomerBlockedException

customerSet()

Sets/overwrites various personal/company data of customer.

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification
person			personal details
name	O	string	first (family) name
lastName	O	string	last name
middleName	O	string	middle name
gradeBefore	O	string	scientific grade before name
gradeAfter	O	string	scientific grade after last name
birthYear	O	string	
company			
name	O	string	company registered name
idNum	O	string	company registration number (ičo)
taxId	O	string	tax registration number (dič)
contactPerson	O	string	Full name of a contact person in the company.
primaryAddress			home address (personal) or domicile address (company)
street	O	string	name of street or settlement area
streetNumber	O	string	house number and/or street number
zip	O	string	zip code
city	O	string	city or municipality name
country	O	string	(2-digit code)
deliveryAddress			delivery address (to be used for post)
street	O	string	name of street or settlement area
streetNumber	O	string	house number and/or street number
zip	O	string	zip code
city	O	string	city or municipality name
country	O	string	(2-digit code)
invoicingAddress			invoicing address (to be used as "to" address on BSS-generated invoices)
street	O	string	name of street or settlement area

streetNumber	O	string	house number and/or street number
zip	O	string	zip code
city	O	string	city or municipality name
country	O	string	(2-digit code)
email	O	string	e-mail address
phone	O	string	contact phone number (different from this mobile service number)
scPhone	O	string	login for self-care
scPassword	D	string	password for self-care, depends on scPhone
requiredAuthorization Value	O		this parameter cause that customer will be asked to authorization card
amount	M	long	
currency	M	string	
postBilling	O	boolean	Whether the customer wants the paper bill sent by post every month.
billingEmail	O	string	Alternative e-mail for sending bills.
defaultTariffId	O	long	Default BSS tariff ID. Currently used to preselect tariff in PPW.

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
customerData	customerData	After-change copy of personal/company data

Implementation

Verify uniqueness of the MSISDN in MCD.CUSTOMER_MSISDN.

Write to MCD.CP_ORDER, MCD.CP_ORDER_COMPONENT, MCD.CP_ORDER_COMPONENT_PARAMETER, MCD.CUSTOMER, MCD.CUSTOMER_MSISDN.

Errors

CustomerNotFoundException

CustomerBlockedException

DuplicateMsisdnException – if the MSISDN is already occupied by another customer in CUSTOMER_MSISDN

customerRemove()

Removes customer and all associated services. Any existin SIM cards will be deactivated. Customer's personal/company data, numbers (MSISDNs) are kept for billing purposes.

Implementation notes:

1. Set status "D" to all generic services
2. Set blocked=1 in table CUSTOMER
3. Do not release numbers and sim cards
4. run workflow *customer_onRemove(customerid)*

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification

Output parameters

none

Event Data Records

Each removed subscription (MSISDN) produces EDR of category SUB, operation END.

Errors

MissingArgumentException, InvalidArgumentException, CustomerNotFoundException, CustomerBlockedException

customerNoteSet()

Set note for customer.

Input parameters

parameter	M/O	data type	explanation, example
note	M	customerNote	

consumerSearch()

Looks for consumers matching the conditions.

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Limit to consumers of this customer.
firstName	O	string	Limit to consumers their first name contains ...
lastName	O	string	Limit to consumers their last name contains ...
email	O	string	Limit to consumers their email contains ...
master	O	boolean	Limit to masters / non-masters.

Output parameters

parameter	data type	explanation, example
consumers		
consumer	[]	
customerId	integer	
id	integer	
firstName	string	
lastName	string	
email	string	
master	booleans	

Implementation

Read from MCD.CONSUMER.

Errors

none specific

consumerCreate()

Creates a new consumer.

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Which customer the consumer belongs to.
firstName	M	string	
lastName	M	string	
email	M	string	Must be unique per the whole platform. It is used as login to SCW as well.
password	M	string	
master	M	boolean	- true - has access to all the GIDs of the customer - false - has acces only to GIDs assigned to him

Output parameters

parameter	data type	explanation, example
consumerId	integer	ID of the created consumer.

Implementation

Verify uniqueness of the MCD.CONSUMER.EMAIL.

Write to MCD.CONSUMER.

Write to MCD.CONSUMER_PARAMETER:

- plainTextPassword = <password>
- initialPassword = 1

Errors

DuplicateEmailException – if the email is already occupied by another consumer

consumerSet()

Modifies a consumer.

Input parameters

parameter	M/O	data type	explanation, example
consumerId	M	integer	Which consumer to modify.
firstName	O	string	
lastName	O	string	
password	O	string	
master	O	boolean	- true - has access to all the GIDs of the customer - false - has acces only to GIDs assigned to him

Output parameters

none

Implementation

Write to MCD.CONSUMER.

Errors

none specific

consumerRemove()

Removes a consumer.

Only consumer not assigned to any GID can be removed.

Input parameters

parameter	M/O	data type	explanation, example
consumerId	M	integer	Which consumer to modify.

Output parameters

none

Implementation

Verify if not assigned to any GID.

Remove from MCD.CONSUMER.

Errors

ActiveConsumerException – if the consumer is still assigned to a GID

gsSearch()

Finds list of gs grouped by customers. Searches only via GS number. It is suitable for autocomplete functionality

Input parameters

parameter	M/O	data type	explanation, example
searchKey	M	UTF8 string	Part of GS number
limit	O	integer	Maximum number of matching customers to get. Default = maximum = 100.

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
gsList	List of GS	customer1Name { list of String GS numbers } customer2Name { list of String GS numbers } ... customer100Name { list of String GS numbers }

Errors

MissingArgumentException, InvalidArgumentException, ArgumentOutOfRangeException

gsAdd()

Creates (adds) mobile generic service to the existing customer. Gives MSISDN⁵ and activates SIM card. The new SIM card (of existing customer) may have set different tariff but must be of the same product as existing services.

pre-paid products

Customer with pre-paid product may only gsAdd() tariff with the same pre-paid product (pricelist may differ). New service (MSISDN) will **share** the credit with old service. To not share the credit, create separate customer instead of gsAdd.

⁵but not with usage of withoutMSISDN=true

post-paid products

Customer with post-paid product may only gsAdd() tariff with the same post-paid product (pricelist may differ). Product options (billing period, invoice due date, notifications, etc.) will be the same for existing and new service. Existing and new service are to be **billed on single invoice**. To bill the service on separate invoice, create separate customer instead of gsAdd.

Implementation notes:

1. check that tariffId exists in BSS s_tariffs
2. check that existing s_services's productIds are the same (tariff and pricelist may differ); consider "D"eleted services also
 - refuse when no s_service exists
1. create one MCD GENERIC_SERVICE (GID); add _tech type 13 "Mobile"
2. create BSS s_services record with serviceClass 2.7
3. allocate one MSISDN from pool of partner's available numbers
4. set all default m_* parameters into NUMBER_PARAMETER
5. run (asynchronous) workflow **subscriptionActivate(msisdn,imsi)** for the new SIM

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification
preferredMSISDN	O	string	Preferred new MSISDN to use. See getAvailableMSISDNs() method. If preferredMSISDN is used, and is not available, error is returned.
tariffId	M	integer	Which tariff (pricelist) to use. Use one of tariffs offered to partner, as returned by tariffList()
name	O	string	Name of the generic service.
consumerId	O	integer	Owner of this GID (workplace).
withoutMSISDN	O	boolean	No MSISDN (no SIM) is assigned. Use this to create "empty" generic service for migration of MSISDN from another existing customer. See simMigrate() method.
imsi	O	string	Identification of SIM that has to be assigned to new subscription. Only one of IMSI (14-15 digits) or ICCID (19-20 digits) may be used. If not specified, manual task is created on Logistics interface and SIM is assigned by human.
iccid	O	string	

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
gid	integer	Numeric identification of new generic service Example: 11502
msisdn	string	Allocated new MSISDN, E.164 format Example: 420781014255

Event Data Records

New subscription produces EDR of category SUB, operation START.

Errors

MissingArgumentException, InvalidArgumentException, CustomerNotFoundException, CustomerBlockedException, TariffNotFoundException, UnsupportedObjectState, NoFreeSimException, NoPreferredFreeMsisdnException, NoFreeMsisdnException, ImsiNotAvailableException

gsRemove()

Removes one mobile generic service from customer. Any existing SIM cards will be deactivated.

Implementation notes:

1. Set status "D" to the generic service
2. Do not release numbers and sim cards
3. for each number (global_numbering_plan), run workflow **subscriptionDeactivate(MSISDN)**

Input parameters

parameter	M/O	data type	explanation, example
gid	M	integer	Numeric identification of generic service Example: 11502
removeAfter	O	datetime	Date and time of removing generic service. Generic service will be removed after this datetime. If not specified, generic service will be removed immediately. Example: 2015-05-03T23:30

Output parameters

none

Event Data Records

Each removed subscription (MSISDN) produces EDR of category SUB, operation END.

Errors

MissingArgumentException, InvalidArgumentException, CustomerBlockedException, GenericServiceNotFoundException, GenericServiceDeactivatedException, GenericServiceDeactivationPlannedException

gsRemoveCancel()

Cancel planned removal of generic service.

Input parameters

parameter	M/O	data type	explanation, example
gid	M	integer	Numeric identification of generic service Example: 11502

Output parameters

none

Errors

GenericServiceStornoDeactivationException

gsTariffChange()

Requests change of tariff for existing customer (generic service). Change will be processed by workflow engine. Pre-paid to post-paid migration available.

Restrictions for customer with multiple generic services: combination of prepaid and postpaid tariffs among generic services is not allowed.

Implementation notes:

1. run (asynchronous) workflow tariffChange for the new SIM

Input parameters

parameter	M/O	data type	explanation, example
gid	M	integer	Numeric identification of generic service Example: 11502
newTariffId	M	integer	
effectiveDate	O	date	Date from which new tariff will be applied. Default = today. Only allowed values are between <today> and <1st day of next billing interval>

(M)andatory, (O)ptional, (D)epends on others

Output parameters

-none-

Event Data Records

Swap of tariff produces EDR of category CHG, operation TARIFF.

Errors

MissingArgumentException, InvalidArgumentException, CustomerNotFoundException, CustomerBlockedException, TariffNotFoundException

gsSpendingLimitChange()

Request for changing spending limit (floor in gpcs) on gpcs account that relates to gid.

Input parameters

parameter	M/O	data type	explanation, example
gpcsAccount	M	integer	
gid	M	long	
amount	M	integer	new value of spending limit (floor in gpcs)
period	O	string	(month, year)

Output parameters

-none-

componentAdd

Creates a new component in a GID.

Input parameters

parameter	M/O	data type	explanation, example
gid	M	integer	What GID to add component to.
subcategoryId	M	integer	Component's subcategory.
hid	M	integer	HostServiceProvider of the component. Use ID of the calling partner you don't know.
startDate	O	date	First day of billing of this component. Cannot be earlier than the first day of the current month. If not specified, today will be used.
endDate	O	date	Last day of billing of this component. Must come after startDate. If not specified it will be active indefinitely until removed manually.

Output parameters

parameter	M/O	data type	explanation, example
componentId	M	integer	

Implementation

Verify GID exists.

Verify subcategory exists.

Verify hid exists.

Verify startDate is not before the first day of the current month.

Verify endDate is after startDate.

Write to MCD.COMPONENT.

IF startDate <= NOW

Create EDR:

- eventDate: <startDate> at 00:00:00
- hid: <hid>
- pid: „auto“
- cid: „auto“

- gid: <gid>
- msisdn: null
- category: „COMPONENT“
- operation: „BILL“
- service: „{categoryCode}/{subcategoryCode}“ # SW/OFFICE365
- meta:
 - COMPONENT_ID=<component.id>
 - DAYS = "<day of startDate>/<last day in month of startDate>"

ENDIF

Errors

GenericServiceNotFoundException

InvalidArgumentException

componentAddCustom

Creates a new component with a custom name and price in a GID.

Input parameters

parameter	M/O	data type	explanation, example
gid	M	integer	What GID to add component to.
name	M	string	Custom name of the component. Will be displayed on invoice.
currency	M	string	3-char currency code. Usually EUR, CZK. Must be in upper case.
price	M	decimal(13,4)	Custom price of the component.
vat	M	string	3-char VAT code followed either by minus (the price is without VAT) or plus (the price is with VAT). Must be in upper case. Examples: CZ2-, CZ2+
startDate	O	date	First day of billing of this component. Cannot be earlier than the first day of the current month. If not specified, today will be used.
endDate	O	date	Last day of billing of this component. Must comes after startDate. If not specified it will be active indefinitely until removed manually.

Output parameters

parameter	M/O	data type	explanation, example
componentId	M	integer	

Implementation

Verify GID exists.

Verify startDate is not before the first day of the current month.

Verify endDate is after startDate.

Write to MCD.COMPONENT:

- hid = <ID of the calling partner>
- subcategoryId = INTERNAL/CUSTOM

Write to MCD.COMPONENT_PARAMETER these parameters:

- name = <name>
- currency = <currency>
- price = <price>
- vat = <vat>

IF startDate <= NOW

Create EDR:

- eventDate: <startDate> at 00:00:00
- hid: <hid>
- pid: „auto“
- cid: „auto“
- gid: <gid>
- msisdn: null
- category: „COMPONENT“
- operation: „BILL“
- service: „INTERNAL/CUSTOM“
- meta:
 - COMPONENT_ID=<component.id>
 - DAYS = "<day of startDate>/<last day in month of startDate>"

ENDIF

Errors

GenericServiceNotFoundException

InvalidArgumentException

componentRemove

Ends a component.

Input parameters

parameter	M/O	data type	explanation, example
componentId	M	integer	
endDate	O	date	To which day to end this component? = Last day of billing of this component. Must comes after startDate. If not specified it will be ended to today.

Output parameters

none

Implementation

Verify component exists.

Verify component is not ended already (even in the future).

Verify endDate is after startDate.

Write to MCD.COMPONENT.

Errors

ComponentNotFoundException

InvalidArgumentException

componentParameterSet

Sets a parameter of a component.

Input parameters

parameter	M/O	data type	explanation, example
componentId	M	integer	
name	M	string	Parameter's name.

value	M	string	Parameter's value.
-------	---	--------	--------------------

Output parameters

none

Implementation

Verify component exists.

Verify parameter is allowed to be changed (see table below).

Write to MCD.COMPONENT_PARAMETER.

parameter	format	explanation, example
deliveryDate	YYYY-MM-DD	Date of the component's delivery. It can be any date.
activationDate	YYYY-MM-DD	Date of the component's activation. It can be any date.

Errors

ComponentNotFoundException

InvalidArgumentException

getAvailableMSISDNs()

Lists first up to 50 available MSISDNs from partner's pool. This may be used prior to customerCreate() call. Please note that any of MSISDNs may become unavailable just after calling this. MSISDNs are **not reserved** by this call.

Implementation notes:

- Numbers are HostMNO- and tariff-specific
- use the same WHERE and ORDER BY as in createCustomer*() methods
- SELECT NUMBER FROM GLOBAL_NUMBERING_PLAN WHERE PARTNER_ID=<X> AND GID is null ORDER BY CATEGORY, WHEN_USED, NUMBER LIMIT 10
- In case there are not enough numbers in MCD.GNP, HMNO interface must be queried in order to obtain those numbers (model [acquireAvailableMsisdns](#))
- pattern: search with like "%pattern%"

Input parameters

parameter	M/O	data type	explanation, example
tariffId	M	integer	Chosen tariff identification; use one of tariffs returned by tariffList. Set of available numbers may vary per tariff.
count	O	integer	Number of available MSISDNs to get. Default = 1.
containsDigits	O	string	Requested number specification. Numbers containing given pattern are preferred in search. Example: 567
leadingDigits	O	string	Requested number specification. Only numbers beginning by given pattern are returned. To be specified without international prefix. Example: 790791

Output parameters

parameter	data type	explanation, example
msisdn	List of strings	Available MSISDNs

simAssign()

Adds an MSISDN and SIM card to the existing customer and (generic mobile) service. The new MSISDN will have the same tariff (product,pricelist) as other SIM cards within (existing) generic service.

Input parameters

parameter	M/O	data type	explanation, example
gid	M	integer	Numeric identification of existing generic service Example: 11502
preferredMSISDN	O	string	Preferred new MSISDN to use. See getAvailableMSISDNs() method. If preferredMSISDN is used, and is not available, error is returned.
imsi	O	string	Identification of SIM that has to be assigned to new subscription. Only one of IMSI (14-15 digits) or ICCID (19-20 digits) may be used. If not specified, manual task is created on Logistics interface and SIM is assigned by human.
iccid	O	string	
force	O	boolean	Tells if reserved mssidn can be assigned. Default is false.

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
msisdn	string	Allocated new MSISDN, E.164 format Example: 420781014255

Event Data Records

New subscription produces EDR of category SUB, operation START.

If cpOrderComponentId is filled

The component's status must be realizationRequested.

Read the GID from MCD.CP_ORDER.

Change component's status to realizationStarted, before the activiti models are run.

Add record into MCD.CP_ORDER_COMPONENT_HISTORY + MCD.CP_ORDER_COMPONENT_PARAMETER_HISTORY.

If there is a MCD.mnpPortInRequest, set its cpOrderComponentId by the input parameter.

Implementation

1. allocate one MSISDN from pool of partner's available numbers
2. set all default m_* parameters into NUMBER_PARAMETER
3. run (asynchronous) workflow *subscriptionActivate(MSISDN)* for the new number

Errors

CustomerBlockedException, GenericServiceNotFoundException, GenericServiceDeactivatedException, NoFreeSimException, NoPreferredFreeMsisdnException, NoFreeMsisdnException, ImsiNotAvailableException, TooManyNumbersPerGid
PermissionException – if the cpOrderComponentId doesn't belong to the authenticated partner

simMigrate()

Moves existing active MSISDN and SIM card from one customer (generic service) to another customer (generic service). Tariff (product,pricelist) is not changed.

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN to migrate. Example: 420789111222
newGid	M	integer	Identification of generic service that MSISDN is migrated to.

(M)andatory, (O)ptional, (D)epends on others

Output parameters

-none-

Event Data Records

Migration produces 2 EDRs of category SUB, operation MOVE.

1st EDR for original (old) GID.

2nd EDR for new GID.

Errors

MissingArgumentException, InvalidArgumentException, CustomerBlockedException, GenericServiceNotFoundException, GenericServiceDeactivatedException, TooManyNumbersPerGid

simProfileGet()

Gets SIM profile parameters for one specific MSISDN.

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN for which we want to change services, E.164 format Example: 420781014255

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
simService	simServicesType	List of all user-configurable services (both active and inactive)

Obtain list of existing service identifiers by simServiceList()

Example:

```
<simService>
  <name>m_PremiumSmsBar</name>
  <value>1</value>
  <pending>true</pending>
</simService>

<simService>
  <name>m_premiumSmsBar</name>
  <value>0</value>
  <pending>true</pending>
  <upToDate>>false</upToDate>
</simService>

<simService>
  <name>m_communicationProfile/language</name>
  <value>Czech</value>
  <upToDate>true</upToDate>
</simService>
```

Errors

MissingArgumentException, InvalidArgumentException, MsisdnNotFoundException, CustomerBlockedException, GenericServiceDeactivatedException,

simProfileSet()

Sets SIM profile parameters.

Implementation notes:

- for each *m_** parameter that has changed,
 - update *NUMBER_PARAMETER* table: set meta *CHQRQ*:<new value>
- for each parameter that was changed, run (asynchronous) workflow *simProfileChange(msisdn,simServiceName)*

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN for which we want to change services, E.164 format Example: 420781014255
simServiceSet	M+	simServiceSetType	New value of service

(M)andatory, (O)ptional, (D)epends on others

Output parameters

-none-

Example - input:

```
<simServiceSet>
  <name>m_PremiumSmsBar</name>
  <newValue>0</newValue>
</simServiceSet>
<simServiceSet>
  <name>m_CallForwarding/busyNumber</parameter>
  <newValue>289010477</newValue>
</simServiceSet>
```

Event Data Records

Activation of services of type **bool_act_deact** produces EDR of category SRV, operation ACT-REQ and ACT-DONE.

Deactivation of services of type **bool_act_deact** produces EDR of category SRV, operation DEACT-REQ and DEACT-DONE.

Change of service of another type produces EDR of category SRV, operation SET-REQ and SET-DONE.

EDRs of type *-REQ are created by PMWpartner interface directly.

EDRs of type *-DONE are created asynchronously by workflow engine.

Parameter specific notes:m_voiceMail/pin

For simServiceName m_voiceMail/pin there is proceed specific check for validity and strength of a Voice mail PIN. The PIN has to be 4-6 digits without repetitions (all digits the same) and all lineary increased/decreased. An InvalidArgumentException is thrown if the PIN satisfies at least one of the following conditions:

- The PIN has less than 4 or more than 6 digits.
- All digits are the same (1111, 222222, ...).
- All digits increases/decreases by one (12345, 2345, 54321, ...).

m_barringByCustomer/*

- barring of outgoing international calls deactivates barring of all outgoing calls
- barring of all incoming calls when roaming barring of all incoming calls
- barring "of all outgoing international calls except those directed to home plmn" deactivates barring of all incoming calls
- for details, see service provisioning catalogue

Errors

MissingArgumentException, InvalidArgumentException, MsisdnNotFoundException, CustomerBlockedException, GenericServiceDeactivatedException, UnsupportedObjectState

simSuspend()

Blocks usage of services on SIM card (subscription) temporarily. Outgoing calls, messages and data connection are barred. Incoming calls are available except roaming. Incoming messages are available.

Monthly fees, if applicable, are charged even on suspended subscription.

Typical usage of simSuspend() is barring of postpaid customer due to late payments or debt, or fraud suspicion.

The method throws InvalidStateException if the msisdn is already suspended.

Implementation notes:

1. Do not set SIM_HLR_ACTIVE to 0
2. run (asynchronous) workflow **simSuspend()** (this will set HLR_ACTIVE down)

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN, E.164 format Example: 420781014255
suspendReason	O	string	Reason of suspend Example: user00 : description of suspend

(M)andatory, (O)ptional, (D)epends on others

Errors

MissingArgumentException, InvalidArgumentException, MsisdnNotFoundException, CustomerBlockedException, GenericServiceDeactivatedException, InvalidStateException

simResume()

Re-activates SIM card suspended by simSuspend before.
The method throws InvalidStateException if the msisdn is not suspended.

Implementation notes:

1. Do not set SIM_HLR_ACTIVE
2. run (asynchronous) workflow **simResume()** (this will set HLR_ACTIVE up)

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN, E.164 format Example: 420781014255

(M)andatory, (O)ptional, (D)epends on others

Errors

MissingArgumentException, InvalidArgumentException, MsisdnNotFoundException, CustomerBlockedException, GenericServiceDeactivatedException, InvalidStateException

simRemove()

Removes one SIM from service from customer. SIM card is deactivated permanently. After simRemove, MSISDN is removed. No outgoing and incoming activities (calls, messages, etc.) are possible.
Billing of monthly fees, if applicable, is stopped after simRemove.

Implementation notes:

1. start workflow **subscriptionDeactivate(MSISDN)**
2. If there is no other number (sim) in generic service:
 1. set status "D" to the generic service
 2. do not release numbers and sim cards

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN, E.164 format Example: 420781014255

Output parameters

none

Event Data Records

Each removed subscription (MSISDN) produces EDR of category SUB, operation END.

Errors

MissingArgumentException, InvalidArgumentException, MsisdnNotFoundException, CustomerBlockedException, GenericServiceDeactivatedException

simSwapMSISDN()

Requests swap (set new) MSISDN on a SIM card. You have to obtain new MSISDN using getAvailableMSISDNs(count=1).

Implementation notes:

1. check whether new MSISDN is available (→ return an error)
2. run workflow simSwapMSISDN (this will do the swap)
- assigns new MSISDN to existing GID

Input parameters

parameter	M/O	data type	explanation, example
newMSISDN	O	string	MSISDN, E.164 format, which is not used Example: 420781014255
msisdn	M	string	MSISDN, E.164 format Example: 420781014255

Output parameters

-none-

Errors

simSwap()

Requests new SIM card for existing number. Old SIM will be marked as physically lost.

Implementation notes:

1. check whether new SIM card (by imsi) is available
2. run workflow simReplacement (this will do the swap)
 - old SIM card should be set to LOST

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN, E.164 format Example: 420781014255
newImsi	O	string	Identification of SIM that has to be assigned to this subscription. Old SIM will be deactivated permanently. Only one of IMSI (14-15 digits) or ICCID (19-20 digits) may be used. If not specified, manual task is created on Logistics interface and SIM is assigned by human.
newIccid	O	string	

Output parameters

-none-

Errors

edrGet()

Gives event detail records (EDR) for a generic service within specified timeframe (date from - date to). EDRs contains information about service activation/deactivation/change history, tariff- and billing- related events and billing-package events.

EDR format is specified together with CDR format in separate document: **MVNE CDR format and fields.pdf**.

Implementation notes:

1. obraz tabulky edrYYYYMM z billing-db.

Input parameters

parameter	M/O	data type	explanation, example
customerId	D	integer	Numeric customer identification
gid	D	integer	Numeric identification of existing generic service
msisdn	D	string	MSISDN (number) of SIM
One of customerId, or gid, or msisdn must be present.			
dateFrom	O	date	EDRs with this date and newer are returned. Default = 1st day of actual month. Example: 2013-11-01
dateTo	O	date	EDRs with this date and older are returned. Default = current date. dateTo must not be before dateStart. The dateFrom-dateTo range must not cross month boundary. Example: 2013-11-30

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
numRecords	integer	Number of EDRs returned
edr	string (list)	edr is comma-separated-value string: eventDate,gid,msisdn,category,operation,service,valueOld,valueNew,meta,log

For EDR record format see separate document "MVNO CDR format and fields.odt". There are no rating (pricing) information.

Examples:

"2013-05-03 16:15:31","40","420771198995","SRV","ACT-REQ","m_premiumSmsBar","0","1","",""
 "2013-05-03 16:15:54","40","420771198995","SRV","ACT-DONE","m_premiumSmsBar","0","1","",""

imsiCommonPartGet()

Gets common part (from left) of all IMSI assigned to the partner or empty string if no common part there exists.

Examples:

- for IMSI from 230030000000000 to 230030000000999 returns 230030000000
- for IMSI from 130030000000000 to 230030000000000 returns empty string

Input parameters

No parameters needed

Output parameters

parameter	data type	explanation, example
commonPart	string	Common part of all IMSI assigned to the partner

iccidCommonPartGet()

Gets common part (from left) of all ICCID assigned to the partner or empty string if no common part there exists.

Examples:

- for ICCID from 8942031013162000111 to 89525252522000111 returns 89
- for IMSI from 7942031013162000111 to 8942031013162000111 returns empty string

Input parameters

No parameters needed

Output parameters

parameter	data type	explanation, example
commonPart	string	Common part of all ICCID assigned to the partner

updateSimServices()

Synchronizes SIM services with data stored in mobile operator database.

Due to conditions required by API's of mobile operators there exists the following restrictions:

- 1.) Another request for synchronization on the same MSISDN is allowed after 3600 sec (UpdateSimServicesMinimumDelayForOneMsisdnNotReached).
 - 2.) Another request for synchronization on any MSISDN is allowed after 20 sec (UpdateSimServicesMinimumDelayNotReached).
 - 3.) Max count of request for synchronization is 100 requests per day (UpdateSimServicesDailyLimitReached).
- A request which reaches one of the above mentioned restrictions is ignored.

Implementation notes:

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN, E.164 format <i>Example: 420781014255</i>
filter	M	string	Identification of synchronization action: all volatile simServices, where: all: synchronizes all services, volatile: synchronizes only services which can be changed outside MVNE platform (by customer from mobile, ...), simServices: synchronizes only exactly selected services (in that case the parameter simServices is mandatory)
simServices	O	string	Coma separated list of ids of services selected for synchronization (mandatory if filter=simServices)

Output parameters

parameter	data type	explanation, example
updateSimServicesItems	List of updateSimServicesItem	simService: name of synchronized sim service lastUpdated: last timestamp of update oldValue: invalid/old value which was corrected newValue: value stored in operator database - corrected value

Errors

UpdateSimServicesInvalidFilter, UpdateSimServicesListOfServicesMissing, UpdateSimServicesInvalidSimServiceId, UpdateSimServicesMinimumDelayForOneMsisdnNotReached, UpdateSimServicesMinimumDelayNotReached, UpdateSimServicesDailyLimitReached

selfCarePassword()

Returns part of MSISDN's self-care password (typically 4-8 digits). Only 2 digits of the password are visible, the others are marked with dash (-). For better realtime readability, it also returns position (offset) of digits shown. This is to be used for querying on-the-phone or on-site customer for their password without putting whole password out of platform. "Tell me please 2nd and 4th digit from your password".

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN, E.164 format <i>Example: 420781014255</i>

Output parameters

parameter	data type	explanation, example
whichDigits	string	Comma-separated list of 2 numbers in form "X,Y". Those two numbers are offset of digits shown in the password. Offset numbers are random but does not change within one day (repeated calls to selfCarePassword do not allow to reassemble whole image of password)
passwordTwoDigits	string	String of the same length as selfcare password, with all but 2 digits replaced by "-".

Examples:

output parameters	interpretation
whichDigits 1,5 passwordTwoDigits 4---5-	Customer has to provide 1 st and 5 th digit from password. These digits are 4 and 5. Password is 6 digits long.
whichDigits 1,2 passwordTwoDigits 12--	Customer has to provide 1 st and 2 nd digit from password. These digits are 1 and 2. Password is 4 digits long.
whichDigits 6,8 passwordTwoDigits -----8-1	Customer has to provide 6 th and 8 th digit from password. These digits are 8 and 1. Password is 8 digits long.

selfCarePasswordSet()

Sets or resets MSISDN's self-care password to a new value. Password is 4-8 decimal digits.

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN, E.164 format <i>Example: 420781014255</i>
newPassword	M	string	New password, string of 4-8 decimal digits.

Output parameters

-none-

selfCareMasterSet()

Sets or unsets MSISDN's self-care-master flag. This flag affects scope (privileges) of self-care web interface (and pmweu-rest interface) of given MSISDN.

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN, E.164 format <i>Example: 420781014255</i>
isMaster	M	boolean	true ... this MSISDN, after login to pmweu/selfcare web, may see/modify all settings of all other MSIDNS of all Generic Services of a customer. false ... this MSISDN, after login to pmweu/selfcare web, may see/modify settings only of MSISDN itself.

Output parameters

-none-

emailAddressList()

Returns sales email, list of groups and its emails configured in MVNE platform.

Input parameters

-none-

Output parameters

parameter	data type	explanation, example
salesEmail	string	General "sales" email
edrNotificationGroup	emailAddressList	1 - 10 e-mail groups for EDR notifications

Example output

```
<salesEmail>info@example.com</salesEmail>
<edrNotificationGroup>
```

```
<groupName>skupina1</groupName>
<rfc822address>"George, Ted" &lt;Shared@Group.Arpanet&gt;</rfc822address>
<rfc822address>John Doe &lt;john@example.net%></rfc822address>
<rfc822address>...</rfc822address>
</edrNotificationGroup>
<edrNotificationGroup>
  <groupName>skupina2</groupName>
  <rfc822address>portability@example.net</rfc822address>
</edrNotificationGroup>
<edrNotificationGroup>
  <groupName>skupina9</groupName>
  <rfc822address>ar@example.net</rfc822address>
  <rfc822address>sales@example.net</rfc822address>
</edrNotificationGroup>
```

emailAddressSet()

Sets e-mail addresses of specified purpose. Returns new set of e-mail addresses configured in MVNE platform.

Implementation notes

- config max_group_count_per_partner = 10
- config max_email_count_per_group = 10
- groupName omezeno na [a-zA-Z0-9] max 10 znaku

Input parameters

parameter	M/O	data type	explanation, example
salesEmail	O	string	general "sales" e-mail. salesEmail in rfc822 format
edrNotificationGroup	O	edrNotificationGroup	1 - 10 e-mail groups for EDR notifications. groupName in format [a-zA-Z0-9] max 10 chars

Example 1a:

```
<salesEmail>info@example.com</salesEmail>
The sales-email address is set.
```

Example 1b:

```
<salesEmail></salesEmail>
The sales-email address is removed.
```

Example 2:

```
<edrNotificationGroup>
  <groupName>skupina1</groupName>
  <rfc822address>a@example.org</rfc822address>
</edrNotificationGroup>
All addresses in group "skupina1" are replaced by one address a@example.org
```

Example 2:

```
<edrNotificationGroup>
  <groupName>skupina2</groupName>
</edrNotificationGroup>
The group is removed completely
```

Output parameters

parameter	data type	explanation, example
salesEmail	string	general "sales" e-mail
edrNotificationGroup	list	1 - 10 e-mail groups for EDR notifications

(same as emailAddressList)

Exceptions

InvalidArgumentException - Parameter salesEmail (<salesEmail>) is not valid e-mail address (RFC822)

TooManyElementsException - Not more than 10 groups allowed per partner.

TooManyElementsException - Not more than 10 emails allowed per group.

InvalidArgumentException - Parameter rfc822address (<rfc822address>) is not valid e-mail address (RFC822)");

eventTypeList()

Returns all EDR categories and types. No operation, just returns static list.

Implementation notes

- just read from BSS table b_edrCategories and b_edrOperations
- show those pmwPartnerVisible=1 operations

Input parameters

- none -

Output parameters

parameter	data type	explanation, example
eventTypes	list	

Example output

```
<eventTypes>
  <category>
    <name>SRV</name>
    <description>Supplemental & VAS Services</description>
    <operation>
      <name>ACT-REQ</name><description>activation of simProfile service is
requested</description></operation>
      <operation><name>ACT-DONE</name>
        <description>activation of simProfile service is finished by HMNO</description></operation>
      <operation><name>DEACT-REQ</name>
        <description>deactivation of simProfile service is requested</description></operation>
      <operation><name>DEACT-DONE</name>
        <description>deactivation of simProfile service is finished by HMNO</description></operation>
      <operation><name>SET-REQ</name>
        <description>new value of simProfile service is requested</description></operation>
      <operation><name>SET-DONE</name>
        <description>new value of simProfile service is set by HMNO</description></operation>
      <operation><name>SC-PASSWORD-SET</name>
        <description>new self-care-password has been set</description></operation>
      <operation><name>SC-PASSWORD-CHANGE</name>
        <description>existing self-care-password has been modified</description></operation>
      <operation><name>SC-LOGIN</name>
        <description>login to self-care</description></operation>
      <operation><name>SC-LOGIN-VALIDATE</name>
        <description>if needed, pre-validate step (SMS OTP sent, RingIn created ...)</description></operation>
    </category>
    <category>
      ...
    </category>
</eventTypes>
```

edrTriggerList()

Returns new all event triggers configured in MVNE platform.

Implementation notes

-

Input parameters

- none -

Output parameters

parameter	data type	explanation, example
edrTriggers	list	

Example output:

```

<edrTrigger>
  <id>4</id>
  <condition>
    <category>PKG</category>
    <operation>ORDER</operation>
    <service>PACKAGE:MP:1200MB.*</service>
    <oldValue></oldValue>
    <newValue></newValue>
    <meta></meta>
    <msisdn></msisdn>
  </condition>
  <note>Data order 1200MB</note>
  <active>1</active>
  <action>
    <emailSubject>notification - ordered 1200MB data package</emailSubject>
    <emailGroup>skupina3</emailGroup>
  </action>
</edrTrigger>
<edrTrigger>
  ...
</edrTrigger>

```

edrTriggerSet()

Sets MVNO partner's event triggers. Returns new (full) set of event triggers configured in MVNE platform.

Purpose: Event trigger, once configured, runs on create of Event Detail Record (EDR). Action of the trigger is to send an e-mail to specified address.

Implementation notes

- config max_trigger_count_per_partner = 100

Input parameters

parameter	M/O	data type	explanation, example
edrTriggers	O	list	

Example 1:

```

<edrTrigger>
  <id>4</id>
  ... listed parameters are updated. The rest stays unchanged.
</edrTrigger>

```

Example 2:

```

<edrTrigger>
  <id>0</id> ()
  ... trigger creation
</edrTrigger>

```

Example 2:

```

<edrTrigger>
  <id>6</id>
  <delete>1</delete>
  ... trigger delete
</edrTrigger>

```

Output parameters

parameter	data type	explanation, example
edrTriggers	list	

(full list, the same as in edrTriggerList)

Exceptions

InvalidArgumentException - There is no trigger with ID...

TooManyElementsException - Not more than 100 triggers allowed per partner.

InvalidArgumentException - Category <category> does not exist.

InvalidArgumentException - Email group <group> does not exist.

autoTopupSet()

Automatic topup is the feature for pre-paid subscribers. On specified, subscriber-controllable conditions

a) decrease of credit balance under specified threshold, or

b) expiry of validity of credit

automatic online card payment is performed. Pre-configured amount of money is withdrawn from customer's payment card and corresponding topup occurs immediately. Precondition for this service is that customer gives authorization. Authorization is that 1st online payment (topup) is performed at Self Care Web and agreement is given to use that payment (card) for repeated automatic topups.

This method may set parameters of autoTopup service that is already authorized.

This method may temporarily enable or disable the service.

This method may permanently revoke authorization.

Implementation notes

- *operate on prepaid only*
- *read operations - go to DB MCD5 NUMBER_PARAMETER*
- *set operations - use PMWadmin /autoTopupSet method, synchronously*

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN, E.164 format <i>Example: 420781014255</i>
enabled (*)	O	boolean	true ... enables previously authorized automatic topup; no effect when no authorization exists false ... disables automatic topup Subscriber may also enable or disable this service itself via Self Care.
threshold (*)	O	long	Set threshold - when balance of number's prepaid credit falls under specified value, automatic topup occurs. Minimum value is 1. Subscription's currency. <i>Example: 20</i>
atExpiry (*)	O	boolean	Set/unset condition for automatic topup - expiry of validity of prepaid credit.
amount (*)	O	long	Amount of money to withdraw from the bank when automatic topup occurs. <i>Example: 150</i>
notifyMsisdn (*)	O	string	Notification MSISDN - will receive <i>Example 420778159219.</i>
revokeAuthorization	O	boolean	true ... permanently disable automatic topup and remove authorization. The only way how customer re-gains authorization is to perform "first" online payment again.

(*) This is also controllable by subscriber at Self Care

Output parameters

-none-

autoPaymentRemove()

Turn off autopayment for postpaid customers.

Implementation notes

- operate on postpaid only
- set operations - use PMWadmin /autoPaymentRemove method, synchronously

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	long	ID of customer for which we want to remove autopayment. Example: 20

Output parameters

-none-

paymentHistoryInfo()

Returns paymentHistoryInfo for given customer

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	long	ID of customer for which we want to remove autopayment. Example: 20

Output parameters

parameter	data type	explanation, example
paymentHistoryInfo	paymentHistoryInfo	

Methods - SIM card logistics

simStockOverview()

Returns overview of number of SIM cards available for partner.

Input parameters

-none-

Output parameters

parameter	data type	explanation, example
numAvailable	integer	Number of SIM cards available for new customers/services
numAllocated	integer	Number of SIM cards allocated to any customer/service
numActive	integer	Number of SIM cards active
numLost	integer	Number of SIM cards reported as physically destroyed

simInfo()

Returns information about SIM card (subscriber identity module) by IMSI or ICCID.

Input parameters

parameter	M/O	data type	explanation, example
imsi	D	string	Identification of SIM - IMSI (14-15 digits)
iccid	D	integer	Identification of SIM - ICCID (19-20 digits)

(D) - only one of imsi, iccid may be used

Output parameters

parameter	data type	explanation, example
exists	boolean	true ... specified SIM exists in MVNE platform false ... no such SIM is available to use
imsi	string	Identification of SIM - IMSI (14-15 digits)
iccid	string	Identification of SIM - ICCID (19-20 digits)
brand	string	Physical human-readable description of how plastic of SIM looks like (size, colour, printed information) - UTF8 string
active	boolean	true ... SIM is active in HMNO
assigned	boolean	true ... SIM is assigned to a MSISDN (subscriber) false ... SIM is free to use

Errors

ImsiNotAvailableException

findSim()

Fulltext search SIM by IMSI or ICCID.

Input parameters

parameter	M/O	data type	explanation, example
imsi	D	string	Identification of SIM - IMSI (14-15 digits) - fulltext search
iccid	D	string	Identification of SIM - ICCID (19-20 digits) - fulltext search

(D) - only one of imsi, iccid may be used

Output parameters

parameter	data type	explanation, example
imsi	string	Identification of SIM - IMSI (14-15 digits)
iccid	string	Identification of SIM - ICCID (19-20 digits)
brand	string	Physical human-readable description of how plastic of SIM looks like (size, colour, printed information) - UTF8 string

Methods - product management

tariffList()

Lists all tariffs available for use. Tariff is bundle of product (business options, payment options, terms of conditions, but no price information) and pricelist (price information). Additionally basic details about products (name, description) and pricelists (name) are returned.

To obtain complete pricelist details and items, use pricelistGet().

Implementation notes

- just read from BSS table s_tariffs
- show tariffs only linked to product only linked to partner (s_products.partner = <X>)
- productList: read name, shortname, description from s_products for those products referenced by at least one partner<X>'s tariff
- pricelistNames: read name from f_pricelists for those pricelists referenced by at least one partner<X>'s tariff

Input parameters

parameter	M/O	data type	explanation, example
prepaid	O	integer	If set, filters: 1 = only prepaid 0 or other = only postpaid Does not filter if not set

Output parameters

parameter	data type	explanation, example
tariffList	tariffList	list of tariffs
productList	productList	brief of products used by tariffs
pricelistNames	pricelistNames	names of pricelists used

```
tariffList = {
  tariffId1 { productId, name, shortname, pricelistId },
  tariffId2 { ... },
  ...
}
productList = {
  productId1 { product.name, product.shortname, product.isPrepaid, product.description },
  productId2 { ... },
  ...
}
pricelistNames = {
  pricelistId1 {name},
  pricelistId2 {name},
  ...
}
```

pricelistGet()

Returns content (items) of a BSS pricelist

Implementation notes:

- only allow to get a pricelist which is referred by at least one partner<X>'s product
- how to determine pricelist version: use the newest active version with non-future validFrom
- do not allow getting pricelist which is active=0 or version which is active=0
- položky ceníku 1:1 z tabulky f_pricelist_items až na
 - note - nevracet
 - premiumValueFrom a premiumValueTo - nevracet
- vatRates - vylejt tabulku f_vat a v_vatRate

Input parameters

parameter	M/O	data type	explanation, example
pricelistId	M	integer	Numeric identification of pricelist (as from tariffInfo from tariffList method) Example: 420781014255

Output parameters

parameter	data type	explanation, example
name	string UTF8	pricelist name - human readable Example: MATERNA friends&families
version	string	currently valid pricelist version number: M.N (M - major number, N - minor number) Example: 4.12 or 1.0
validFrom	date	date of pricelist validity start Example: 2013-01-01
createTime	date	date and time of when the pricelist was registered in BSS Example: 2012-11-29
currency	string	default currency for pricelist items; 3-char currency code (but items may have different currency; this is to be used for items with empty currency) Example: EUR
notes	string UTF8	human readable notes for pricelist - plain text (may contain \n newlines)
item	pricelistItem (list)	List of pricelists items (typically 500, maximum 10.000 items)
vatRates	vatRate (list)	List of VAT rates referenced by items

example of pricelist header:

```
<name>X pokusný ceník mobilních služeb č. 1</name>
<version>9.0</version>
<validFrom>2013-03-02</validFrom>
<createTime>2013-05-06</createTime>
<currency>CZK</currency>
<notes>HPLMN calls - změna tarifkace 60+1</notes>
```

example of pricelist item

```
<item>
  <type>VOICE-TERM</type>
  <subType/>
  <destination>8</destination> <destination>9</destination> <destination>10</destination> <destination>11</destination>
  <destination>12</destination> <destination>16</destination> <destination>17</destination> <destination>18</destination>
  <destination>19</destination> <destination>20</destination> <destination>21</destination>
  <description>volání ČR mobilní a pevné sítě</description>
  <fixPrice>0.0100</fixPrice>
  <unitPrice>4.9000</unitPrice>
  <curr>CZK</curr>
  <vat>CZ2+</vat>
  <discount>0.000</discount>
  <vBillSecIntervals>60+1</vBillSecIntervals>
  <vTimeZone>0</vTimeZone>
</item>
```

example of vatRates:

```
<vatRates>
  <vatRate>
    <abbr>CZ1</abbr>
    <description>Česká Republika, snížená sazba</description>
    <rate>15.00</rate>
  </vatRate>
  <vatRate>
    <abbr>CZ2</abbr>
    <description>Česká Republika, základní sazba</description>
    <rate>21.00</rate>
  </vatRate>
```

</vatRates>

Errors

MissingArgumentException, InvalidArgumentException, PricelistNotFoundException, PricelistVersionNotFoundException

simServiceList()

Returns static list of per-SIM services that may be activated or deactivated. List and structure of services varies per installation. For actual list of services, please refer to **PMW service catalogue** document.

simServices - type:

- **bool_act_deact**: service that may be active/inactive. End user may request to activate or deactivate.
 - web form structure: [x] FooBar service
- **enum_1ofN**: one of many radiobutton – contains 2 or more <operand> elements
 - web form structure: (O) operand1 () operand2 () operand3
- **text**: free text
 - web form structure: []
- **text_number**: input field for telephone number: fixed length max 17 characters, format check /\+?([0-9]{2,15})\$/
 - web form structure: []
- **container**: this is not changeable service, but container for several parameters

Implementation notes:

- *vzít obsah tabulky MCD SIM_SERVICE*

Input parameters

parameter	M/O	data type	explanation, example
language	O	string	Two letter language code according to ISO 639 (see http://en.wikipedia.org/wiki/ISO_639-1 , http://www.loc.gov/standards/iso639-2/php/code_list.php , ...)

(M)andatory, (O)ptional, (D)epends on others: exactly one of customerId, gid or msisdn must be provided for this method.

Output parameters

parameter	data type	explanation, example
simService	simServiceDefinitionType	List of services and default values
l10nStrings	l10nStringType	List of language-specific descriptions for service-settings page. This set of localization strings must be applied prior to web portal's localization system. Applicable to • name of service • operands

simServiceDefinitionType - examples

```
<simService>
  <id>101</id>
  <name>m_PremiumSmsBar</name>
  <type>bool_active_inactive</type>
</simService>

<simService>
  <id>101</id>
  <name>m_barringByCustomer</name>
  <type>container</type>
  <volatile>false</volatile>
</simService>

<simService>
  <id>101</id>
  <name>m_barringByCustomer/incomingCall</name>
  <type>bool_act_deact</type>
  <defaultValue>0</defaultValue>
  <volatile>false</volatile>
</simService>
<simService>
```



```

<id>101</>
<name>m_barringByCustomer/incomingSms</name>
<type>bool_act_deact</type>
<defaultValue>0</defaultValue>
<volatile>>false</volatile>
</simService>

<simService>
<id>101</>
<name>m_communicationProfile/subscriberNotificationType</name>
<type>enum_1ofN</type>
<defaultValue>Nothing</defaultValue>
<operand>Nothing</operand>
<operand>Voice</operand>
<operand>All</operand>
<volatile>>false</volatile>
</simService>

```

l10nStringsType

```

<l10nStrings>
  <code>OneByOne</code>
  <value>po jednom</value>
</l10nStrings>
<l10nStrings>
  <code>AllTogether</code>
  <value>hromadně</value>
</l10nStrings>
...

```

smsNotificationTemplateList()

Returns current list SMS workflow-to-customer notifications.

Input parameters

-none-

Output parameters

parameter	data type	explanation, example
template	smsNotificationTemplate (list)	List of existing SMS notifications and texts. code ... machine readable short code - unique per template description ... machine readable description about how this template is used in MVNE platform enabled ... false - no SMS is sent of this template language-lang ... 2-digit language code language-text ... SMS text template itself language-lastUpdated ... date of last change of template text variable-name ... name of {variable} placeholder valid for this template variable-description ... human-readable description about this variable usage

Example output - smsNotificationTemplate

```

<template>
  <code>N1</code>
  <description>Aktivace SIM, informace o tarifu - uvítací zpráva</description>
  <enabled>1</enabled>
  <language>
    <lang>cs</lang>
    <text>Vítejte v síti Q-Test. Váš tarif je {tariffName} Bezplatná infolinka *55. Více na www.example.com</text>
    <lastUpdated>2013-06-13</lastUpdated>
  </language>
  <language>
    <lang>en</lang>

```

```
<text>Welcome to Q-Test. Your tariff is {tariffName}. Contact us on *55 or www.example.com.</text>
<lastUpdated>2013-06-13</lastUpdated>
</language>
<variable>
  <name>number</name>
  <description>Telefonní číslo zákazníka v národním formátu (9 číslic)</description>
</variable>
<variable>
  <name>tariffName</name>
  <description>Přidělený tarif</description>
</variable>
</template>
```

[Example of visualisation:](#)

smsNotificationTemplateSet()

Sets new text for a notification template. Changes are applied immediately.

Input parameters

parameter	M/O	data type	explanation, example
code	M	string	code of template to be set
lang	M	string	language to be set; not mandatory for enable/disable operation
enabled	O	boolean	enables or disables sending of SMS notification from this template.
text	O	string	Text of SMS notification template. May contain {variables}

(M)andatory, (O)ptional, (D)epends on others: exactly one of customerId, gid or msisdn must be provided for this method.

Output parameters

-none-

Examples 1 - set new template text for N2a

```
<soapenv:Body>
  <pmw:smsNotificationTemplateSet>
    <code>N2a</code>
    <lang>cs</lang>
    <text>X-Test: právě jsme Vám aktivovali balíček {packageName}. Měsíční paušál je {packagePrice} (do konce tohoto měsíce jen {packageProratingPrice}). Více na tel. *55. nebo na www.example.com.</text>
  </pmw:smsNotificationTemplateSet>
</soapenv:Body>
```

Example 2 - disable sending of notification from template N2a

```
<soapenv:Body>
  <pmw:smsNotificationTemplateSet>
    <code>N2a</code>
    <enabled>>false</enabled>
  </pmw:smsNotificationTemplateSet>
</soapenv:Body>
```

portOutList()

Returns list of a port OUT items

Implementation notes:

Content of table mcd5.mnpPortOutRequest

Input parameters

parameter	M/O	data type	explanation, example
msisdn	O	string	Filter by part of ported MSISDN (like)

			<i>Example: 420781014255</i>
status	O	string	Filter by status of porting. Coma separated list of the statuses: new,provisioned,ctidRequested,ctidCreated,pacCreated,pacVerified,cancelled,numberReleased,planned,deactivated,deactivationFailed,rejected,ctidExpired. Example: ctidRequested,ctidCreated,pacCreated
ctidCode	O	string	Filter by part of ctid code (like) Example: 21308080002583
pacCode	O	string	Filter by part of pac code (like) Example: 13000000373
createdFrom	O	date	Filter by date when the porting item was created Example: 2013-01-01
createdTo	O	date	Filter by date when the porting item was created Example: 2013-01-01

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
portOutRequests	PortOutRequest (list)	List of portOutRequest items

example of portOutRequest item

```
<portOutRequest>
  <msisdn>420776398982</msisdn>
  <status>provisioned</status>
  <ctidCode>200555</ctidCode>
  <pacCode>13000000364</pacCode>
  <pacStatus>63</pacStatus>
  <created>2013-08-02T11:43:10</created>
  <firstPortingDate>2013-08-01</firstPortingDate>
  <releaseDate>2013-08-05</releaseDate>
  <portingDate>2013-08-09</portingDate>
</portOutRequest>
```

portOutConfirm()

Confirms the given port out number and sets first possible port-out date.

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	MSISDN for which has to be confirmed port out item <i>Example: 420781014255</i>
firstPortingDate	M	date	The first possible date of porting Example: 2013-01-01

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
-----------	-----------	----------------------

Errors

NoPortOutRequestException, MissingArgumentException, InvalidArgumentException

portInList()

Returns list of a port-in requests.

Implementation notes:

Content of table mcd.mnpPortInRequest

Input parameters

parameter	M/O	data type	explanation, example
msisdn	O	string	Filter by part of ported MSISDN (like) Example: 420781014255
status	O	string	Filter by status of porting. Coma separated list of the statuses: new,provisioned,exported,valid,invalid,'pacAssigned,pacVerified,numberReleased,confirmedByCrdB,planned,activationPending,activated,activationExported,activationFailed,activationConfirmed,rejected. Example: activationConfirmed,rejected
ctidCode	O	string	Filter by part of ctid code (like) Example: 21308080002583
pacCode	O	string	Filter by part of pac code (like) Example: 13000000373
createdFrom	O	date	Filter by date when the porting item was created Example: 2013-01-01
createdTo	O	date	Filter by date when the porting item was created Example: 2013-01-01
customerName	O	string	Filter by part of customer name which was inserted on porting form by customer (like) Example: Tomas
customerNumber	O	string	Filter by part of customer number which was inserted on porting form by customer (like) Example: 8000

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
portInRequests	PortInRequest (list)	List of portInRequest items

example of portInRequest item

```

<portInRequest>
  <msisdn>420777888001</msisdn>
  <status>activated</status>
  <ctidCode>213007</ctidCode>
  <pacCode>777</pacCode>
  <pacStatus>70</pacStatus>
  <!-- if standard port in -->
    <activationType>standard</activationType>
    <swapFromMsisdn />
  <!-- else if swap over existing SIM -->
    <activationType>swapMsisdn</activationType>
    <swapFromMsisdn>420123456789</swapFromMsisdn>
  <donorOperatorCode>205</donorOperatorCode>
  <created>2013-05-28T09:21:49</created>
  <firstPortingDate>2013-07-20</firstPortingDate>
  <customerName>Jenny Suk</customerName>
  <customerNumber>008</customerNumber>
  <contactMsisdn>420737937685</contactMsisdn>
  <tariff>nevim</tariff>
</portInRequest>

```

portInDelete()

Deletes port in item for the given msisdn and ctidCode. The deleted item has to have status 'new'.

Implementation notes:

The deleted item is not really deleted but the status of the item is changed to 'deleted'

Input parameters

parameter	M/O	data type	explanation, example
msisdn	O	string	Example: 420781014255
ctidCode	O	long	Example: 21308080002583

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
-----------	-----------	----------------------

portInOverExistingMsisdn()

Creates portIn request and replaces an existing Msisdn by the ported one.

Input parameters

parameter	M/O	data type	explanation, example
portedMsisdn	M	string	An external Msisdn which is ported in. Example: 420123456789
swapFromMsisdn	M	string	An existing activated Msisdn whis is used as a target for starting porting process. Example: 420781014255
contactMsisdn	O	string	Plain text. Not validated. Example: 420123456789
ctid	M	string	Example: 21308080002583

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
-----------	-----------	----------------------

Errors

[MnpPortInInvalidCtidException](#), [MnpPortInDuplicateCtidException](#), [MnpPortInMsisdnCannotBeUsedException](#)

Methods - billing and invoicing

invoiceList()

Gives list of PDF wholesale invoices for last 24 months. Sorted by issue date, newer first.

Implementation notes:

- PDF files are stored under *cfg-specified per-partner directory (partnerInvDir)* For example, partner 502 */var/pinv/502/*
- In *partnerInvDir*, there are just files with structured name
- filename convention: *invoice-p<P>-g<GID>-<YYYYMMDD>-<SEQ>.pdf*
 - *<P>* partner ID
 - *<GID>* generic service id of partner's MVNE wholesale service (BSS green serviceclass #16)
 - *<YYYYMMDD>* date of issue
 - *<SEQ>* invoice number as used for accounting purposes (=variable symbol)
 - example: *invoice-p502-g8952-20140501-4400125567.pdf*
- database billing: *SELECT dIssue,dDue,bruttoValue,currency FROM `fdoc\_invoiceView` WHERE id=<SEQ>*

Input parameters

-none-

Output parameters

parameter	data type	explanation, example
invoiceList	invoiceHeader	List of PDF invoices

Invoices are available for download at URL3.

invoiceGet()

Returns content of an wholesale invoice.

Input parameters

parameter	M/O	data type	explanation, example
invoiceId	M	integer	invoice number (max. 10 digits)

Output parameters

parameter	data type	occurence	explanation, example
invoice	invoiceType	<1>	Aggregate - invoice, list of blocks and items

Errors

[NoSuchInvoiceException](#)

invoicePaymentStatusSet

Changes the payment status of an invoice.

Only retail invoices can be made paid this way.

The status can be changed only from pending to paid.

Input parameters

parameter	M/O	data type	explanation, example
invoiceId	M	integer	The ID of the order.
paymentStatus	M	string	The only allowed value as of now: paid.

Output parameters

None

Errors

NoSuchInvoiceException – if there is no such retail invoice

InvalidStateException – if the invoice cannot be marked paid

cdrGet()

Returns CDR (call detail records) for specific customer/service/SIM card within specified timeframe (date from - date to).

Implementation notes:

- *customerId input? expand to list of GIDs.*
- *GIDs? expand to list of MSISDNs*
- *return cdrList as sequence of \n-terminated lines from script's STDOUT*
- *make no other parsing on STDOUT content*
- *use database connection from pmwpartner.conf - billing_readonly*

Input parameters

parameter	M/O	data type	explanation, example
customerId	D	integer	Numeric customer identification
gid	D	integer	Numeric identification of existing generic service
msisdn	D	string	MSISDN (number) of SIM
dateFrom	O	date	CDRs with this date and newer are returned. Default = 1st day of actual month. Example: 2013-11-01
dateTo	O	date	CDRs with this date and older are returned. Default = current date. dateTo must not be before dateStart. The dateFrom-dateTo range must not cross month boundary. Example: 2013-11-30

(M)andatory, (O)ptional, (D)epends on others: exactly one of customerId, gid or msisdn must be provided for this method.

Output parameters

parameter	data type	explanation, example
numRecords	integer	Number of CDRs returned
cdr	string	List of Charging Detail Records for the customer/service/SIM card. Ordered by date/time. Internal limit 10.000 items to return. CDRline is comma-separated-value string: dateStart,dateEnd,usageType,account,aParty,bParty,origin,volume,reference,destId,gid

For CDR record format see separate document "MVNO CDR format and fields.odt". There are no rating (pricing) information.

Examples:

"2013-05-06 14:12:07","2013-05-06 14:16:36","2000","420716398190",,"420240103160","420","269","EC74B90008
07005340857","21","1774"
"2013-05-06 14:10:47","2013-05-06 14:11:09","2000","420716398190",,"420607450018","420","22","EC211E0008
07005340838","16","1774"

Errors

InvalidArgumentException, ArgumentDependencyException, NoMsisdnsException, ExternalCommandException,
CustomerNotFoundException, CustomerBlockedException, GenericServiceNotFoundException, GenericServiceDeactivatedException,
MsisdnNotFoundException

cdrUsageTypeList()

Gives list of cdr usage types. Could be usefull for cdr presentation because only cdr usage type code is included in a cdr item record.

Input parameters

parameter	M/O	data type	explanation, example

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
cdrUsageType	cdrUsageType	List of cdrUsageType items

Example output:

```
<cdrUsageType>
  <usageType>1010</usageType>
  <name>Obecný jednorázový poplatek</name>
  <pricelistitemType>ONETIME-FEE</pricelistitemType>
</cdrUsageType>
<cdrUsageType>
  <usageType>1011</usageType>
  <name>Jednorázový poplatek za zřízení čísla</name>
  <pricelistitemType>NUMBERFEE-STARTUP</pricelistitemType>
</cdrUsageType>
```

Errors

customerInvoiceList()

Gives list of customer's PDF invoices for last 6 months. Sorted by issue date, newer first.

Implementation notes:

- PDF files are stored under cfg-specified per-partner per-customer directory (customerInvDir) For example, partner 502, customer 589660 /var/cinv/p502/c589660/
- In customerInvDir, there are just files with structured name
- filename convention: invoice-c<C>-g<GID>-<YYYYMMDD>-<SEQ>.pdf
 - <C> customer ID
 - <GID> generic service customer's "mobile" service (BSS green serviceclass #2.7)
 - <YYYYMMDD> date of issue
 - <SEQ> invoice number as used for accounting purposes (=variable symbol)
 - example: invoice-c589660-g7485574-20140601-4501406450.pdf
- database billing: SELECT dlssue,dDue,bruttoValue,currency FROM `fdoc\_invoiceView` WHERE id=<SEQ>

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
invoiceList	invoiceHeader	List of PDF invoices

Example output:

```
<invoice>
  <invoicId>445571505</invoicId>
```



```

<issueDate>2013-05-01</issueDate>
<dueDate>2013-05-15</dueDate>
<paymentStatus>pending</paymentStatus>
<bruttoValue>987.6000</bruttoValue>
<currency>CZK</currency>
<pdfFileName>invoice-c589660-20140601-4501406450.pdf</pdfFileName>
</invoice>

```

PDF invoices are available for download at URL3.
 XML invoices are available via `customerInvoiceGet()` method.

Errors

MissingArgumentException, InvalidArgumentException, CustomerNotFoundException, CustomerBlockedException

customerInvoiceGet()

Returns content of an invoice.

Input parameters

parameter	M/O	data type	explanation, example
invoiceId	M	integer	invoice number (max. 10 digits)

Output parameters

parameter	data type	occurence	explanation, example
invoice	invoiceType	<1>	Aggregate - invoice, list of blocks and items

Elements of invoiceType:

element	type	explanation
id	unsignedLong	unique numeric ID of invoice; generated using INV_SEQNUM_MASK; typ. length 10 digits
customerId	long	customer that invoice is issued to; customerInvoiceGet() uses this, invoiceGet() does not
dIssue	date	date of issue
dDue	date	due date (issue date + ~14 days)
dTaxableSupply	date	date of taxable supply; for Mobile Usage retail invoices, last day of billing interval
dPeriodStart	date	first day of billing interval (typ. 1st of month)
dPeriodEnd	date	last day of billing interval (typ. last of month)
currency	string	3-character currency code ISO4217
paymentStatus	string	One of: pending, expired, canceled, paidPartly, paid
paymentType	string	PAYMENT-ORDER or CREDIT or NONE
from	invoiceAddressType	body that has issued the invoice (requests money)
to	invoiceAddressType	body that the invoice is addressed to (obliged to pay)
bankAccount	string	number of bank account (local or IBAN) where addressee should send the money (in case of paymentType=PAYMENT-ORDER)
tradeRegisterInfo	string	human-readable information of issuer's trade register entry
textHeader	string	free text on top of invoice
textFooter	string	free text on bottom of invoice
bruttoValue	decimal	total value of invoice, including 3rd party payments, including VAT; excluding deposit returns, rounding, overpayments/underpayments from recent billing intervals

roundingTransfer	decimal	transfer of rounding from last billing period - this is negative value of rounding of last invoice of the same customer; typical value varies from -0.50 to 0.50
total1	decimal	total value of invoice, including 3rd party payments, including VAT, deposit returns, rounding, and overpayments/underpayments from recent billing intervals; not rounded
rounding	decimal	rounding - such value that sum of total1 + rounding gives integer (no decimal part); typical value varies from -0.50 to 0.50 Example: total1 281.82 → rounding = 0.18 (total2 = 282.00) Example: total1 55.49 → rounding = -0.49 (total2 = 55.00) Functionality of rounding may be set on/off per MVNO partner.
total2	decimal	total value of invoice with rounding total2 = total1 + rounding
block	invoiceBlockType	block of items of each customer's generic service (GID)

Elements of invoiceAddressType:

element	type	explanation
name	string	personal or company name
address	string	postal address - street+number-
zip	string	- ZIP code
city	string	- city/municipality name
country	string	- country name
regNum	string	registration number (registered companies)
taxId	string	VAT registration number (registered companies)

Elements of invoiceBlockType:

element	type	explanation
gid	unsignedLong	Generic Service ID (GID)
numberList	string	comma-separated list of numbers; customerInvoiceGet() uses this, invoiceGet() does not
tariffName	string	Name of tariff related to this GID; in case of tariff-switch in the middle of billing period contains the original name
item	invoiceItemtype	items actually contained on an invoice

Elements of invoiceItemtype:

element	type	explanation
plItemId	unsignedLong	time-consistent ID of pricelist item (->pricelist/version/item); machine readable
plItemType	string	type of item: 'ONETIME-FEE', 'NUMBERFEE-STARTUP', 'MONTHLY-FEE', 'MONTHLY-CREDIT', 'NUMBERFEE-MONTHLY', 'NUMBERFEE-CDR', 'SMS-TERM', 'SMS-ORIG', 'VOICE-TERM', 'VOICE-ORIG', 'PSMS-BONUS', 'PSMS-SHARE', 'PSMS-SHARE-MOB', 'PSMS-SHARE-MTB', 'PRICELIST-CHANGE', 'TICKET', 'SMS-ADD-FEE', 'PACKAGE', 'PACKAGE-R', 'PACKAGE-R-A', 'PACKAGE-CDR', 'ROAM-VOICE-TERM', 'ROAM-VOICE-ORIG', 'ROAM-SMS-TERM', 'MMS-TERM', 'ROAM-MMS-TERM', 'MOBILE-DATA', 'ROAM-MOBILE-DATA', 'PSMS', 'OTHER', 'PRODUCT'; machine readable
description	string	human-readable description of item
quantity	int	quantity (how many instances of item are billed); number of seconds for voice calls
nbQuantity	int	no-bill quantity (how many extra instances of item exists but are not billed)
numConnections	decimal	number of voice connections (for VOICE item types only)
unitPrice	decimal	unit price (copy from actual pricelist)
totalPrice	decimal	total price = unit price * quantity (/60 for voice calls)

vat	decimal	VAT rate (percent) applicable to the item, percentage in range (-99.99 ... 99.99) positive ... unitPrice and totalPrice are netto (excluding VAT) 0 ... unitPrice and totalPrice are not subject of VAT negative ... unitPrice and totalPrice are brutto (including VAT); the actual VAT rate is - vat
-----	---------	---

Example:

```
<invoice>
  <id>440011558</id>
  <customerId>15801</customerId>
  <dIssue>2013-03-03</dIssue>
  <dTaxableSupply>2013-02-28</dTaxableSupply>
  <dDue>2013-03-15</dDue>
  <dPeriodStart>2013-02-01</dPeriodStart>
  <dPeriodEnd>2013-02-28</dPeriodEnd>
  <currency>CZK</currency>
  <paymentStatus>pending</paymentStatus>
  <paymentType>PAYMENT-ORDER</paymentType>
  <from>
    <name>MATERNA Communications a.s.</name>
    <address>Vinohradská 184/2396</address>
    <zip>130 52</zip>
    <city>Praha 3</city>
    <country>CZ</country>
    <regNum>25949098</regNum>
    <taxId>CZ25949098</taxId>
  </from>
  <to>
    <name>Josef</name>
    <address>Skočdopole</address>
    <zip></zip>
    <city></city>
    <country></country>
    <regNum></regNum>
    <taxId></taxId>
  </to>
  <bankAccountNo>3344433444/0500</bankAccountNo>
  <tradeRegisterInfo>obch. rejstřík Městský soud v Praze, v oddíl B, vložka 2012</tradeRegisterInfo>
  <textHeader></textHeader>
  <textFooter>Děkujeme Vám za využívání našich služeb</textFooter>
  <bruttoValue>67.0500</bruttoValue>
  <roundingTransfer>0</roundingTransfer><total1>67.0500</total1><rounding>-0.0500</rounding><total2>67.0000</total2>
  <block>
    <gid>8895</gid>
    <numberList>791200144,791200145</numberList>
    <tariffName>Foobar 500 promo</tariffName>
    <item>
      <plItemId>16712</plItemId>
      <plItemType>VOICE-TERM</plItemType>
      <description>mobilní sítě v ČR</description>
      <quantity6>901</quantity> <!-- seconds -->
      <nbQuantity7>0</nbQuantity>
      <unitPrice>3.0000</unitPrice>
      <totalPrice>45.0500</totalPrice>
      <vat>21.00</vat>
    </item>
    <item>
      <plItemId>16718</plItemId>
      <plItemType>VOICE-TERM</plItemType>
      <description>pevné sítě v ČR</description>
      <quantity>600</quantity> <!-- seconds -->
      <nbQuantity8>180</nbQuantity>
    </item>
  </block>
</invoice>
```

⁶billed quantity (seconds, messages, kilobytes)

⁷non-billed quantity - used from (monthly) free units

⁸non-billed quantity - used from (monthly) free units

```
<unitPrice>2.2000</unitPrice>
<totalPrice>22.0000<totalPrice>
<vat>21.00</vat>
</item>
</block>
</invoice>
```

Errors

NoSuchInvoiceException

customerTaxDocuments()

Gives list of customer's Tax Documents for last 6 months. Sorted by issue date, newer first.

Implementation notes:

- use database #5 "dando"
- Krok 1 - najít taxuser_id podle všech MSISDN které zákazník má
 - `SELECT if(parentTaxUser_id is null,id,parentTaxUser_id) FROM TaxUser where username in ("420724125361","420724125362", ...)`
- Krok 2 - vyjet seznam dokladů a ten vrátet
 - `SELECT name as 'číslo dokladu', date(duzp) as 'DUZP', date(whenCreated) as 'datum vystavení', concat(totalPrice+totalPriceFactoring,' ',if(currency='CZK','Kč',currency)) as 'hodnota' FROM TaxDocument WHERE id in (SELECT distinct(taxDocId) FROM TaxElement where userId=8826 and TaxDocId is not null UNION SELECT distinct(taxDocIdPartner) FROM TaxElement where userId=8826 and TaxDocIdPartner is not null) AND date_format(duzp,'%Y%m')='201301' ORDER BY duzp`
 - tu šedou část opravit aby to bylo posledních 6 celých (ukončených) měsíců, např. dnes je 21.2.2013, takže chci `duzp>=1.8.2012`
 - takhle to vypadá v kukátku:

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification

(M)andatory, (O)ptional, (D)epends on others

Output parameters

parameter	data type	explanation, example
taxdocuments	taxdocList	List of tax documents

```
taxdocList {
  taxdoc-number {taxdate, issuedate, brutto value, currency}
  taxdoc-number2 {taxdate, issuedate, brutto value, currency}
  ...
}
```

example:

```
taxdocList {
  130100010 { 2013-01-22, 2013-01-22, 50. CZK }
}
```

Tax documents are available for download at URL4 as HTML documents.

Errors

MissingArgumentException, InvalidArgumentException, CustomerNotFoundException, CustomerBlockedException, NoMsisdnsException

customerGPCSlog()

Returns list of GPCS account transactions for customer. Optionally, all (positive and negative) transactions or positive only (topup) transactions are shown. Time range of transactions may not cross calendar month boundary; to get list for more than 1 month, client should call customerGPCSlog() several times. Maximum number of transactions to show is 1000 per response.

Implementation notes:

- just SELECT from GPCS table card_log.
 - card_no je v MCD, v parametru GPCS_ID v CUSTOMER_SERVICE_PARAMETER
 - currency je CZK, resp. vezme se z MCD z PARTNER_PARAMETER
- example: `mysql> select timestamp,balance_before,balance_after,action,info from card_log where card_no='910301301' and currency='CZK' and timestamp>20130101000000 limit 5;`

```

+-----+-----+-----+-----+-----+
| timestamp | balance_before | balance_after | action | info |
+-----+-----+-----+-----+-----+
| 2013-01-01 10:44:18 | 5911.16309999987 | 5905.96309999987 | CALL | A=910301301 B=606198886 dur=0:24/1:00 zone=CZ-mob price=5.2 tariff=7 rate=5.2/60+1 |
| 2013-01-02 22:45:54 | 5905.96309999987 | 5895.73649999987 | CALL | A=910301301 B=608468079 dur=1:58/1:58 zone=CZ-mob price=10.2266 tariff=7 rate=5.2/60+1 |
| 2013-01-02 22:54:25 | 5895.73649999987 | 5876.84319999987 | CALL | A=910301301 B=608468079 dur=3:38/3:38 zone=CZ-mob price=18.8933 tariff=7 rate=5.2/60+1 |
| 2013-01-02 22:58:52 | 5876.84319999987 | 5867.30989999987 | CALL | A=910301301 B=608468079 dur=1:50/1:50 zone=CZ-mob price=9.5333 tariff=7 rate=5.2/60+1 |
| 2013-01-02 23:02:57 | 5867.30989999987 | 5859.59659999987 | CALL | A=910301301 B=608468079 dur=1:29/1:29 zone=CZ-mob price=7.7133 tariff=7 rate=5.2/60+1 |
+-----+-----+-----+-----+-----+

```

- action is enum('TOPUP','CALL','SMS','TARIFF','UPDATE','PARKING','TICKET','FEE','CANCEL','MPOS','PAYMENT') - treat it as string
- info is varchar(255)

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification
dateFrom	O	date	Transactions with this date and newer are returned. Default = 1st day of actual month. Example: 2013-11-01
dateTo	O	date	Transactions with this date and older are returned. Default = current date. dateTo must not be before dateStart. The dateFrom-dateTo range must not cross month boundary. Example: 2013-11-30
topupOnly	O	boolean	This requests to get topup operation only. Topup operation = positive change of balance (add).
offset	O	integer	Which "page" of size 1000 to get. Records are returned from (offset*1000+1)th record. Example: 4 (to get records 4001-5000)

(M)andatory, (O)ptional, (D)epends on others: exactly one of customerId

Output parameters

parameter	data type	explanation, example
numRecords	integer	Number of GPCS transactions (this might be >1000, then you need offset=1, offset=2, etc.)
numShown	integer	Number of GPCS transactions returned (limited by 1000)
currency	char(3)	3-character ISO4217 currency code
transactions	gpsTransList	List of GPCS transactions. Ordered by date/time - old first, new last. Maximum 1000 items to return at a time.

```

gpsTransList = {
  gpsTrans1 { timestamp, balance_before, balance_after, action, info }
  gpsTrans2 { timestamp, balance_before, balance_after, action, info }
  ...
}

```

Errors

MissingArgumentException, InvalidArgumentException, CustomerNotFoundException, CustomerBlockedException, ArgumentOutOfRangeException

customerSetDueDays()

Sets customer's default due date (number of days) for invoices. Does not affect invoices already issued.

Implementation notes:

- just change the dueDays in BSS table f_docDef

Input parameters

parameter	M/O	data type	explanation, example
customerId	D	integer	Numeric customer identification
gid	D	integer	Numeric identification of existing generic service
dueDateDays	M	integer	Number of days for invoice due date. Does not affect invoices already issued. Allowed values are from 0 (due date = issue date) to 365 (one year) Example: 30

(M)andatory, (O)ptional, (D)epends on others: exactly one of customerId or gid

Output parameters
none

Errors

InvalidArgumentException, ArgumentException, ArgumentOutOfRangeException, CustomerNotFoundException, CustomerBlockedException, GenericServiceNotFoundException, GenericServiceDeactivatedException

customerContractGet()

Returns informations about customer contract.

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification

Output parameters

parameter	M/O	data type	explanation, example
contractNumber	M	string	Contract number (max. length 45 chars)
customerId	M	integer	Numeric customer identification
status	M	enum	(signed, notSigned)
signed	O	datetime	Format YYYY-MM-DDTHH:mm
sent	O	datetime	Format YYYY-MM-DDTHH:mm

customerContractCreate()

Generates customer contract and returns contract number.

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification

Output parameters

parameter	M/O	data type	explanation, example
contractNumber	M	string	Contract number (max. length 45 chars)

customerContractSet()**Input parameters**

parameter	M/O	data type	explanation, example
contractNumber	M	string	Contract number (max. length 45 chars)
status	O	enum	(signed, notSigned)
signed	O	Datetime	Format YYYY-MM-DDTHH:mm
sent	O	datetime	Format YYYY-MM-DDTHH:mm

Output parameters

None

paperInvoiceList()

This method gives list of retail paper invoices (fdoc2_invoice.sendPost=1).

Input parameters

parameter	M/O	data type	explanation, example
tariffId	M	integer	Id of tariff-specific
dateFrom	O	date	
dateTo	O	date	

Output parameters

parameter	M/O	data type	explanation, example
total	M	integer	total number of records
paperInvoices	M		
paperInvoice	0..N	[]	
invoiceId	M	long	invoice Id
issueDate	M	date	issue date
dueDate	M	date	due date
paymentStatus	M	string	
bruttoValue	O	decimal	
currency	O	string	
sent	M	boolean	
customerName	M	string	name of customer (first + last name, company name)

exportPaperInvoiceList()

This method returns file (pdf or zip) based on action type.

Input parameters

parameter	M/O	data type	explanation, example
exportInvoices	M		

exportInvoice	0..N	[]	
invoiceId	O	integer	
action	M	string	Action to execute ('export', 'merge')

Output parameters

parameter	M/O	data type	explanation, example
file	M	string	Encoded in BASE64.

paperInvoiceSet()

This method sets invoice as sent.

Input parameters

parameter	M/O	data type	explanation, example
exportInvoices	M		
exportInvoice	0..N	[]	
invoiceId	O	integer	
action	M	string	Action to execute ('sent')

Output parameters

None

Methods - packages

packagesAvailable()

Returns all orderable packages for tariff. The implicit (hard) package that is part of each tariff is not shown (such a package may not be ordered/removed as an option).

Input parameters

parameter	M/O	data type	explanation, example
gid	M	integer	Generic service ID for which we request list of available packages. Set of available packages may vary per tariff.

Output parameters

parameter	data type	explanation, example
package	packageDefinitionType (list)	List of packages available. See WSDL or examples below.

Example output

<package>

```
<ino>123</ino>
<description>100 textových zprav do vsech siti v CR</description>
<recurring>true</recurring>
<priceNetto>109.3444</priceNetto>
<vatRate>21.00</vatRate>
<currency>CZK</currency>
```

</package>

<package>

```
<ino>124</ino>
<description>20 minut do vsech siti v CR</description>
<recurring>true</recurring>
<priceNetto>40.0000</priceNetto>
<vatRate>21.00</vatRate>
<currency>CZK</currency>
```

</package>

packageOrder()

Orders a package of billable units (minutes, SMS, MMS, kilobytes) to the generic service (GID).

One-time packages are applied immediately.

While ordering recurrent package, customer has three options:

1. Activate on start of next billing interval (nextBI=true).
2. Activate now (nextBI=false): package units will be available within some minutes after order. Package price and content is "shortened" according to number of days remaining in current billing interval.
3. Activate on start of activation of subscription (onActivate=true): package will be started conditionally on activation of MSISDN. Applicable for not-yet active subscriptions with port-in number (MNP) that has been already provisioned (GID exists) but port-in date did not occur yet.

Recurring package may not be ordered multiple times.

Implementation notes:

1. do not allow to order HARD package (type PACKAGE-R-A)
2. vytvořit EDR CHG/PKG-ORDER
3. zapsat do mcd.PACKAGE
 - startDate = dnes (nextBI false) nebo 1. den příštího měsíce (nextBI true)
 - endDate = nic pro rekurentní, dle expirace pro jednorázové
1. pouze v případě že startDate=dnes volat workflow **packageOrder(pkgInstancelId)**

Input parameters

parameter	M/O	data type	explanation, example
gid	M	long	Customer's Generic service ID
ino	M	unsigned int	Package item number to be ordered; call packagesAvailable() for a list.
msisdn	O	string	MSISDN for which is the package intended. Applied only for data packages. SIM's and call's packages are always shared by the whole service.
nextBI	O	boolean	Start the ordered package from start of next billing interval (true) or immediately (false). For recurring packages only. Default: false.
onActivate	O	boolean	Start the ordered package on MNP port-in activation (on porting date). Default: false.

Output parameters

parameter	data type	explanation, example
package	packageInstanceType (list)	New updated list of packages ordered for a customer's Generic Service. See packagesList() for examples.

Event Data Records

Order of a package produces EDR of category CHG, operation PKG-ORDER.

packagesList()

Gives overview about customer's packages.

All packages are returned so it is on the client side to filter active and expired packages.

Implementation notes

1. *ino, description, activefrom - get from MCD5.PACKAGE*
2. *show packages with future activeFrom as well*
3. *do not show packages after ENDTIME; do not show recurrent packages with endtime set.*
4. *price, vatRate, currency: get from BSS.f_priceList* (valid pricelist for sactiveFrom date); on no match, do not fail - just do not return these elements*
5. *Possible values of paymentType: 'VOUCHER', 'ONLINE_CARD'*

Input parameters

parameter	M/O	data type	explanation, example
gid	O	string	Customer's Generic service ID
msisdn	O	string	Customer's MSISDN

Output parameters

parameter	data type	explanation, example
package	packageInstanceType (list)	Current list of packages ordered for a customer's Generic Service.

Example:

```
<package>
  <ino>123</ino>
  <msisdn>123456789</msisdn>
  <description>100 textovych zprav do vseh siti v CR</description>
  <activeFrom>2013-11-01</activeFrom>
  <activeTo>2013-11-01</activeTo>
  <recurring>true</recurring>
  <priceNetto>109.3444</priceNetto>
  <vatRate>21.00</vatRate>
```

```
<currency>CZK</currency>
<paymentType>CZK</paymentType>
<origValue>
  <unit>MIN</unit>
  <value>50</value>
</origValue>
<leftOverValue>
  <unit>MIN</unit>
  <value>32:12</value>
  <validAsOf>2015-01-27T01:00:04</validAsOf>
</leftOverValue>
</package>
<package>
  <ino>124</ino>
  <description>20 minut do vseh siti v CR</description>
  <activeFrom>2013-11-01</activeFrom>
  <recurring>true</recurring>
  <origValue>
    <unit>MIN</unit>
    <value>50</value>
  </origValue>
  <!-- priceNetto, vatRate, currency and leftOverValue are not guaranteed to be present -->
</package>
<package>
<ino>100</ino>
<description>měsíční paušál 150+150+150 - telekomunikační služby</description>
<activeFrom>2013-04-01</activeFrom>
<recurring>1</recurring>
<priceNetto>280.165289256198</priceNetto>
<vatRate>21.00</vatRate>
<currency>CZK</currency>
<origValue>
  <unit>MIN</unit>
  <value>150</value>
</origValue>
<origValue>
  <unit>SMS</unit>
  <value>150</value>
</origValue>
<leftOverValue>
  <unit>MIN</unit>
  <!-- leftOverValue when unit=MIN value is string in format min:sec -->
  <value>55:12</value>
  <validAsOf>2015-01-27T01:00:04</validAsOf>
</leftOverValue>
<leftOverValue>
  <unit>SMS</unit>
  <value>10</value>
  <validAsOf>2015-01-27T01:00:04</validAsOf>
</leftOverValue>
</package>
```

packageRemove()

Removes an ordered package. For recurring packages only.

Implementation notes:

1. Check for package type. No operation on non-recurring packages.
2. Do not allow to remove HARD package
3. set *ENDTIME* = last day of current billing interval

Input parameters

parameter	M/O	data type	explanation, example
ino	M	unsigned int	Package item number to be removed.
gid	O	long	generic service ID (scope-customer). Not used in scope-number.
msisdn	O	string	Customer's MSISDN in international format i.e. 420123456789

Output parameters

parameter	data type	explanation, example
package	packageInstanceType (list)	New updated list of packages ordered for a customer's Generic Service. See packagesList() for examples.

Event Data Records

Removal of a package produces EDR of category CHG, operation PKG-REMOVE.

gsFreeUnits()

Returns state of free units (minutes, SMS): current remaining value and total value (actual billing period). Prior to version 10a, name of this method was customerFreeUnits.

Input parameters

parameter	M/O	data type	explanation, example
gid	M	long	generic service ID.

Output parameters

parameter	data type	explanation, example
dateFrom	date	Start date of billing period (from when "total" and "remaining" is valid). Not returned if units are empty (if there are no free units of any type).
units	0..N of FreeUnits	unit code, total number, remaining number

Example:

```
<dateFrom>2014-02-01</dateFrom>
<freeUnit>
  <code>MIN</code>
  <total>20:00</total>
  <remaining>15:20</remaining>
</freeUnit>
<freeUnit>
  <code>SMS</code>
  <total>50</total>
  <remaining>43</remaining>
</freeUnit>
```

dataUsage()

Returns info about data usage

Input parameters

parameter	M/O	data type	explanation, example
gid	M	long	generic service ID.
ino	O	integer	Package identification
msisdn	O	string	Msisdn for which the data usage is taken

Output parameters

parameter	data type	explanation, example
msisdn	string	Start date of billing period (from when "total" and "remaining" is valid). Not returned if units are empty (there are no free units of any type).
dataTariff	string	unit code, total number, remaining number
startDate	date	first active package start date
endDate	date	last active package end date
totalMB	double	Amount of MBs of the assigned packages
usageMB	double	Amount of used MBs

Methods - single-sign-on

customerSSOtokenGet()

Opens temporary access for end-user web self-care application.

Used by PMW-EU (self care widget).

Implementation notes:

- Create very random 45-bytes long string of [a-zA-Z0-9]; must not be unique
- tmpUsername = c + customerId, například "c1564"
- store to table pmweu.tempUsers

Input parameters

parameter	M/O	data type	explanation, example
customerId	M	integer	Numeric customer identification

Output parameters

parameter	data type	explanation, example
tmpUsername	string	Temporary username for PMWeu interface
token	string	Temporary password (token) for PMWeu interface

Errors

InvalidArgumentException, CustomerNotFoundException, CustomerBlockedException

partnerSSOtokenGet()

Opens temporary access for web application "kukatko". Provided tmpUsername and token that are to be used as http-basic authentication credentials.

Implementation notes:

- Create very random 12-bytes long string of [a-zA-Z0-9]; must not be unique
- tmpUsername = mvno<partnerId>-<subUserName> example: mvno20-username
- store to table bss.tmp_kuk_users

Input parameters

parameter	M/O	data type	explanation, example
subsystem	M	string	Subsystem name. The current only allowed value is "kukatko"
subUserName	O	string	Substring, used for generation of output username

Output parameters

parameter	data type	explanation, example
tmpUsername	string	Temporary username for BSS interface "kukatko"
token	string	Temporary password (token) for BSS interface "kukatko"

Errors

InvalidArgumentException

Methods – Component Product (Product Q)

cpComponentCategoryList()

Returns the list of component categories and their subcategories.

Allowed roles

PlatformOwner, Reseller, HostServiceProvider

Input parameters

parameter	M/O	data type	explanation
gid	0	integer	ITF specific. Limits returned subcategories to those, that are specified in the pricelist of this gid.

Output parameters

parameter	data type	explanation, example
categories		
category	[]	
id	integer	
code	string	SIM, NET, TV
name	string	Mobilní služby, Připojení k Internetu
subcategories		
subcategory	[]	
id	integer	
code	string	TALK2, ADSL1, WIFI5
name	string	Základní mobilní tarif TALK ADSL standard (pomalé). Maximální rychlost 8Mbps.

Implementation

Load from BSS.s_componentCategories and BSS.s_componentSubcategories.

cpOrderList()

Returns the list of orders.

The orders are ordered by creation time (descending).

Allowed roles

PlatformOwner, Reseller, HostServiceProvider

Input parameters

parameter	M/O	data type	explanation
pageSize	0	integer	How many orders to return. Max value 1000. Default value 100.
page	0	integer	Which page to return. Default value 1.
orderId	0	integer	

reseller	0		
id	0	integer	Id of reseller
name	0	string	Name of reseller
shortName	0	string	Short name of reseller
person	0		
firstName	0	string	Partial match.
lastName	0	string	Partial match.
company	0		
name	0	string	Partial match.
idNumber	0	string	Exact match.
taxId	0	string	Exact match.
primaryAddress	0		
street	0	string	Partial match.
city	0	string	Partial match.
zip	0	string	Exact match.
country	0	string	2 letter code. Exact match.
status	0	string	
tariffId	0	integer	
creator	0	string	Partial match.
createdFrom	0	date	
createdTo	0	date	
updatedFrom	0	date	
updatedTo	0	date	
customerId	0	long	
orderBy	0	string	value specifies sorting field (id,projectNumber,createdBy,status,created,updated,componentCategoryName,r equestedBy,customer,customerId)
sorted	0	string	enum(asc,desc)
statuses	0	string	comma-separated string with order statuses
componentCategoryId	0	integer	

Output parameters

parameter	M/O	data type	explanation
total	M	integer	How many orders have been found. It can be higher than pageSize.
orders	0..N	[]	The number of orders is limited by pageSize.
order	0..N	[]	
id	M	integer	
reseller	O		
id	M	integer	
name	M	string	
shortName	M	string	

customer	M		
person	D	complex	See cpOrderCreate().
company	D	complex	See cpOrderCreate().
project	D	complex	See cpOrderGet().
primaryAddress	M	complex	See cpOrderCreate().
partnerId	M	integer	Partner ID
id	M	long	Customer ID
status	M	enum	
bundleTariff	O		
id	M	long	
name	M	string	
shortName	M	string	
created	M	datetime	
creator	O	string	
updated	M	datetime	
requestedBy	O	string	who made order request
componentCategoryName	O	string	Component category name of first component in order

Implementation

PlatformOwner can see all the orders with customers he owns.

Reseller can see only orders created by him, in any status.

HostServiceProvider can see only orders , that have at least one componet assigned to him.

Load from MCD.CP_ORDER.

cpOrderGet()

Returns the detail of an order.

Allowed roles

PlatformOwner, Reseller, HostServiceProvider

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	integer	The ID of the order.

Output parameters

parameter	M/O	data type	explanation, example
order	M		
id	M	integer	
type	M	enum	See cpOrderCreate().
reseller	O		
id	M	integer	
name	M	string	
shortName	M	string	

customer	M		
id	M	integer	
person	D	complex	See cpOrderCreate().
company	D	complex	See cpOrderCreate().
project	D		
projectNumber	M	string	
businessLeaderEmail	O	string	
departmentLeaderEmail	O	string	
businessLineld	O	integer	
projectLeader	O		
id	M	integer	
email	O	string	
phone	O	string	
firstName	O	string	
lastName	O	string	
department	O	string	
maternaAdminId	O	string	
primaryAddress	O	complex	See cpOrderCreate().
deliveryAddress	O	complex	See cpOrderCreate().
invoicingAddress	O	complex	See cpOrderCreate().
email	O	string	See cpOrderCreate().
phone	O	string	See cpOrderCreate().
phone2	O	string	See cpOrderCreate().
phone3	O	string	See cpOrderCreate().
scPhone	O	string	See cpOrderCreate().
scPassword	D	string	See cpOrderCreate().
postBilling	O	boolean	See cpOrderCreate().
agreeWithSendDocumentsByEmail		boolean	See cpOrderCreate().
billingEmail	O	string	See cpOrderCreate().
status	M	enum	
paymentCardAuthorization	M	boolean	
collectiveBilling	M	boolean	
warranty	O		
status	M	enum	pending, paid, expired
currency	M	string	CZK
amount	M	integer	
expiration	M	date	
invoiceld	M	integer	
advance	O		
status	M	enum	pending, paid, expired
currency	M	string	CZK

amount	M	integer	
expiration	M	date	
invoiceld	M	integer	
warrantyInvitation	O		
status	M	enum	pending, paid, expired
currency	M	string	CZK
amount	M	integer	
expiration	M	date	
invoiceld	M	integer	
onetimeFees	M		
onetimeFee	0..N	[]	
status	M	enum	pending, paid, expired
currency	M	string	CZK
amount	M	integer	
expiration	M	date	
invoiceld	M	integer	
bundleTariff	O		
id	M	long	
name	M	string	
shortName	M	string	
pricelistId	M	long	Needed for retrieving of the default warranty amount.
note	O	string	
created	M	datetime	
creator	O	string	Who created the order. Free text. Usually first name + last name.
updated	M	datetime	
updater	O	string	Who made the last change. Free text. Usually first name + last name.
components	M		
component	0..N	[]	
id	M	integer	
action	M	enum	See cpOrderComponentCreate().
category	M		
id	M	long	
code	M	string	
name	M	string	
status	M	enum	
subcategory	O		Wholesale component subcategory
id	M	long	
code	M	string	
name	M	string	
tariff	O		Retail tariff
id	M	long	

name	M	string	
shortName	M	string	
hostServiceProvider	O		
id	M	integer	
name	M	string	
shortName	M	string	
parameters	M		
parameter	0..N	[]	
name	M	string	See cpOrderComponentCreate() for available parameters.
value	M	string	
additionalItems	M		
additionalItem	0..N	[]	
ino	M	int	Ino of additional item from price list items
portSimRequested	O	datetime	
created	M	datetime	
updated	M	datetime	
updater	O	string	Who made the last change. Free text. Usually first name + last name.
note	O	string	
customItems	M		
customItem	0..N	[]	
id	M	integer	
name	M	string	Max length 100 chars.
currency	M	string	3-char code.
price	M	decimal	
vat	M	string	The same format as in pricelist. Examples: CZ2-, CZ1+, DE2-.
billDate	M	date	When (in what month) should be billed? Must not be earlier than on the first day of the current month.
shopAssistant	O	String	Code of shop assistant

Implementation

Load from MCD.CP_ORDER, MCD.CP_ORDER_COMPONENT, MCD.CP_ORDER_COMPONENT_PARAMETER, MCD.CUSTOMER.

Errors

PermissionException – if the order is not accessible to the authenticated partner. See cpOrderList() for what orders are accessible to which role.

cpOrderCreate()

Creates a new order.

Allowed roles

PlatformOwner, Reseller

Input parameters

parameter	M/O	data type	explanation, example
type	M	enum	customerCreate, componentChange
customerId	D	integer	Required if type=componentChange. Ignored otherwise.
person	D		Filled if the customer is a person. Cannot be used simultaneously with company.
firstName	M	string	first name
middleName	O	string	middle name
lastName	M	string	last name
gradeBefore	O	string	scientific grade before name
gradeAfter	O	string	scientific grade after last name
birthYear	O	string	
company	D		Filled if the customer is a company. Cannot be used simultaneously with person.
name	M	string	company registered name
idNumber	M	string	company registration number (ičo)
taxId	M	string	tax registration number (dič)
contactPerson	M	string	Full name of a contact person in the company.
primaryAddress	D		home address (personal) or domicile address (company) Required if type=customerCreate. Ignored otherwise.
street	M	string	name of street or settlement area
streetNumber	M	string	house number and/or street number
city	M	string	city or municipality name
zip	M	string	zip code
country	M	string	(2-digit code)
deliveryAddress	O		delivery address (to be used for post)
street	M	string	name of street or settlement area
streetNumber	M	string	house number and/or street number
city	M	string	city or municipality name
zip	M	string	zip code
country	M	string	(2-digit code)
invoicingAddress	O		invoicing address (to be used as "to" address on BSS-generated invoices)
street	M	string	name of street or settlement area
streetNumber	M	string	house number and/or street number
city	M	string	city or municipality name
zip	M	string	zip code
country	M	string	(2-digit code)
email	D	string	e-mail address Required if type=customerCreate. Ignored otherwise.
phone	D	string	MSISDN in format 420777888999. Required if type=customerCreate. Ignored otherwise.
phone2	O	string	MSISDN in format 420777888999
phone3	O	string	MSISDN in format 420777888999

scPhone	O	string	Alternative (random) login for self-care.
scPassword	D	string	Password for self-care. Must be filled if scPhone is used.
postBilling	O	boolean	Whether the customer wants the paper bill sent by post every month.
agreeWithSendDocumentsByEmail	O	boolean	Whether the customer wants send documents by email.
billingEmail	O	string	Alternative e-mail for sending bills.
collectiveBilling	O	boolean	Whether the customer wants to start all the components (the billing) at the same time (in one day). It cannot be set if pmw-partner.conf [componentProduct] collectiveBillingPerOrder=0.
creator	M	string	Who created the order. Free text. Typically firstName + lastName.
note	O	string	Arbitrary human-readable note.
shopAssistant	O	String	Code of shop assistant
resellerId	O	integer	Id of reseller
ruian	O	string	If this is filled, adress fields are ignored durring validation and adress is taken by ruian

Output parameters

parameter	data type	explanation, example
orderId	integer	The ID of the created order.

Implementation

Verify uniqueness of the MSISDN in MCD.CUSTOMER_MSISDN.

If type=customerCreate, a new customer is created.

If type=componentChange, the new order is assigned to the existing customer. All customer related info is updated instead of inserted.

Write to MCD.CP_ORDER, MCD.CUSTOMER, MCD.CUSTOMER_MSISDN.

Fill default value of MCD.CP_ORDER.COLLECTIVE_BILLING by pmw-partner.conf [componentProduct] collectiveBillingDefault.

The customer is created with the PlatformOwner partnerId. If created by Reseller, read from MCD.PARTNER_PARAMETER

"PLATFORM_OWNER_ID".

Write parameters phone2, phone3 into MCD.CUSTOMER_PARAMETERS.

Errors

InvalidArgumentExceptionAddress – if some of the addresses is not valid

DuplicateMsisdnException – if the MSISDN is already occupied by another customer in CUSTOMER_MSISDN

cpOrderChangeStatus()

Changes the status of an order.

Allowed roles

PlatformOwner

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	integer	The ID of the order.
status	M	enum	The new status of the order to be set.
updater	O	string	Who made the change. Free text. Usually first name + last name.

Output parameters

None

Implementation

For allowed transitions between order states see P:\MVNO\Q\phaseII\Q2 – UML - vX.zargo.
Change status in MCD.CP_ORDER.

On status complete

Create RT and send mail by chapter Implementation details / RT.

On status confirmed

Check bundle tariff is set.

Check no custom items of the order has bill date earlier than the first day of the current month.

Create GID with bundle tariff. Use customerCreateGid() in CustomerCreateCommon.pm.

Set bundle GID back into the order.

Errors

InvalidStateException – if the current order status cannot be changed to the new one

InvalidStateException – if the bundle tariff is no set

InvalidStateException – if some of the free items has bill date in the past

cpOrderSet()

Changes an order.

Allowed roles

PlatformOwner

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	integer	The ID of the order.
bundleTariffId	O	long	Order status must be before confirmed.
collectiveBilling	O	boolean	Whether the customer wants to start all the components (the billing) at the same time (in one day). It cannot be changed if pmw-partner.conf [componentProduct] collectiveBillingPerOrder=0.
note	O	string	
updater	O	string	Who made the change. Free text. Usually first name + last name.
shopAssistant	O	string	Code of shop assistant
resellerId	O	integer	Id of reseller
parameters	O		
parameter	0..N	[]	
name	M	string	
value	M	string	

Output parameters

None

Errors

InvalidStateException – if the order cannot be changed in the current status

PermissionException – if trying to change collectiveBilling when collectiveBillingPerOrder=0

cpOrderHistory()

Returns the history of an order.

Allowed roles

PlatformOwner, Reseller

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	integer	The ID of the order.

Output parameters

parameter	M/O	data type	explanation, example
states	M		
state	0..N	[]	Ordered from the oldest (validUntil) to the newest.
validUntil	M	datetime	2015-10-17T13:22:07
status	M	enum	
warranty	M		
status	M	enum	
updater	O	string	Who made the change. Free text. Usually first name + last name.

Implementation

Load from MCD.CP_ORDER_HISTORY.

cpOrderComponentCreate()

Creates a new component order.

Allowed roles

PlatformOwner, Reseller

Allowed order statuses

draft, complete, authorized

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	integer	The ID of the order.
action	M	string	add, change
componentId	D	string	ID from MCD.COMPONENT_VIEW. Examples: GNP16, NET9, TV4. Required if action=change. Ignored otherwise.
categoryId	D	integer	See cpComponentCategoryList().

			Required if action=add. Ignored otherwise.
parameters	D		Required if action=add. Ignored otherwise.
parameter	0..N	[]	
name	M	string	See implementation
value	M	string	
note	O	string	Arbitrary human-readable note.

Output parameters

parameter	data type	explanation, example
orderComponentId	integer	The ID of the created order component.

Implementation

Reseller can create new components only when the order is in the status new.

Depending on the "category" parameter, some of the "parameters" may be required.

If action=change, check if there is no unfinished order with the same componentId. Unfinished order statuses are others than: realized, cancelled.

Write to MCD.CP_ORDER_COMPONENT, MCD.CP_ORDER_COMPONENT_PARAMETER.

If action=change, transfer data from existing component (COMPONENT_VIEW and related tables).

Check address validity against UIR-ADR.

For available parameters see Q2 - detailní analýza – vX.ods.

Errors

PermissionException – if the order is not accessible to the authenticated partner

MissingArgumentException – if some of the required parameters are missing

InvalidArgumentException – if some of the parameters is invalid (wrong type)

InvalidArgumentExceptionAddress – if the address is incomplete or invalid

cpOrderComponentChangeStatus()

Changes the status of a component-product order component.

Allowed roles

PlatformOwner, HostServiceProvider

Input parameters

parameter	M/O	data type	explanation, example
orderComponentId	M	integer	The ID of the order component.
status	M	enum	The new status of the order component to be set.
updater	O	string	Who made the change. Free text. Usually first name + last name.

Output parameters

None

Implementation

For allowed transitions between component states see P:\MVNO\Q\phaseII\Q2 – UML.zargo.

Create record in MCD.CP_ORDER_COMPONENT_HISTORY. Copy all parameters from MCD.CP_ORDER_COMPONENT_PARAMETER to MCD.CP_ORDER_COMPONENT_PARAMETER_HISTORY.

Change status in MCD.CP_ORDER_COMPONENT.

On status new

Remove all parameters not required on component creation.

On any status

Create RT and send mail by chapter Implementation details / RT.

Errors

PermissionException – if the order component is not accessible to the authenticated partner

InvalidStateException – if the current order status cannot be changed to the new one

cpOrderComponentSet()

Changes an order component.

Allowed roles

PlatformOwner, Reseller, HostServiceProvider

Input parameters

parameter	M/O	data type	explanation, example
orderComponentId	M	long	The ID of the order component.
tariffId	O	long	Retail tariff.
hostServiceProviderId	O	long	
subcategoryId	O	long	Wholesale component subcategory. Must be a subcategory of the category of the component. See cpComponentCategoryList().
cpeType	O	string	Identification of cpe (fttx, xdsl)
parameters	O		
parameter	0..N	[]	
name	M	string	See cpOrderComponentCreate()
value	M	string	
additionalItems	O		
additionalItem	0..N	[]	
ino	M	int	Ino of additional item from price list items
action	O	string	Action (add, remove)
updater	O	string	Who made the change. Free text. Usually first name + last name.
note	O	string	Updated note

Output parameters

None

Implementation

Reseller can change only parameters.

Check address validity against UIR-ADR (if changed).

Validate the changed parameters by the same rules as in cpOrderComponentCreate().

Write to MCD.CP_ORDER_COMPONENT, MCD.CP_ORDER_COMPONENT_PARAMETER.

Errors

PermissionException – if the order component is not accessible to the authenticated partner

cpOrderComponentHistory()

Returns the history of an order component.

Allowed roles

PlatformOwner, Reseller, HostServiceProvider

Input parameters

parameter	M/O	data type	explanation, example
orderComponentId	M	integer	The ID of the order component.

Output parameters

parameter	M/O	data type	explanation, example
states	M		
state	0..N	[]	Ordered from the oldest (validUntil) to the newest.
validUntil	M	datetime	2015-10-17T13:22:07
status	M	enum	
tariff			
id	M	long	
name	M	string	
shortName	M	string	
hostServiceProvider	O		
id	M	integer	
name	M	string	
shortName	M	string	
subcategory	O		
id	M	long	
code	M	string	
name	M	string	
parameters	M		
parameter	0..N	[]	
name	M	string	See cpOrderComponentCreate() for available parameters.
value	M	string	
updater	O	string	Who made the change. Free text. Usually first name + last name.

Implementation

Load from MCD.CP_ORDER_COMPONENT_HISTORY.

Errors

PermissionException – if the order component is not accessible to the authenticated partner

cpOrderCustomItemCreate()

Creates a new item with custom name, currency, price and VAT in the order.

Allowed roles

PlatformOwner

Allowed order statuses

draft, complete, authorized

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	long	
name	M	string	Max length 100 chars.
currency	M	string	3-char code.
price	M	decimal	
vat	M	string	The same format as in pricelist. Examples: CZ2-, CZ1+, DE2-.
billDate	M	date	When (in what month) should be billed? Must not be earlier than on the first day of the current month.

Output parameters

parameter	data type	explanation, example
orderCustomItemId	long	The ID of the created item.

Implementation

Write to MCD.CP_ORDER_CUSTOM_ITEM.

Errors

PermissionException – if the order is not accessible to the authenticated partner

InvalidStateException – if the order is not in the allowed state

cpOrderCustomItemSet()

Changes the custom item of an order.

Allowed roles

PlatformOwner

Allowed order statuses

draft, complete, authorized

Input parameters

parameter	M/O	data type	explanation, example
orderCustomItemId	M	long	
name	O	string	Max length 100 chars.
currency	O	string	3-char code.

price	0	decimal	
vat	0	string	The same format as in pricelist. Examples: CZ2-, CZ1+, DE2-.
billDate	0	date	When (in what month) should be billed? Must not be earlier than on the first day of the current month.

Output parameters

None

Implementation

Update in MCD.CP_ORDER_CUSTOM_ITEM.

Errors

PermissionException – if the order is not accessible to the authenticated partner

InvalidStateException – if the order is not in the allowed state

cpOrderCustomItemRemove()

Removes the custom item from the order.

Allowed roles

PlatformOwner

Allowed order statuses

draft, complete, authorized

Input parameters

parameter	M/O	data type	explanation, example
orderCustomItemId	M	long	

Output parameters

None

Implementation

Delete from MCD.CP_ORDER_CUSTOM_ITEM.

Errors

PermissionException – if the order is not accessible to the authenticated partner

InvalidStateException – if the order is not in the allowed state

cpOrderInvoiceSet()

Sets the warranty requirements and sends the generated invoice to the customer.

Allowed roles

PlatformOwner

Allowed order statuses

confirmed

Allowed warranty statuses

notSet

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	integer	
currency	M	string	3 char code.
amount	M	integer	
expiration	M	date	
type	M	enum	Invoice type (advance, warrantyInvitation)

Output parameters

None

Implementation

Insert into fdoc2_invoice:

field	value
customerId	From MCD.CP_ORDER
partnerId	From MCD.CUSTOMER
templateFileName	Read from PMW partner configuration, section [billing], new parameter "warranty_invoice_file_name". Add and document in pmwpartner.conf.sample. Default value="cp-warranty-v1.ods". Create the file in /opt/mc/bss/templates by copying one of the present templates. It can be adjusted later.
dIssue	today
dTaxableSupply	today
dDue	input parameter expiration
currency	input parameter currency
paymentType	ORDER-OR-CARD
from*	From MCD.PARTNER + MCD.PARTNER_PARAMETER.
to*	From MCD.CUSTOMER + other related tables (person, company, address).
domain	Link to "rs".

Keep other fields on their default values.

Insert into fdoc2_invBlocks. Keep all fields on their default values.

Insert into fdoc2_invItems:

field	value
description	"Jistina"
quantity	1
unitPrice	input parameter amount
totalPrice	input parameter amount
vatRate	0
plItemType	ONETIME-FEE

Keep other fields on their default values.

Insert into fdoc2_invOptions:

name	value
cpOrderId	From MCD.CP_ORDER

Save fdoc2_invoice.id to MCD.CP_ORDER.WARRANTY_INVOICE_ID.

Change the warranty status in MCD.CP_ORDER to "set".

Export invoice to PDF by calling /opt/mc/bss/billing/findoc-export-pdf.pl --invoiceld {invoiceld}

The PDF is stored in /opt/mc/bss/findoc/customer-inv-out/p{partnerId}/c{customerId}/invoice-c{customerId}-{issueDateYYYYMMDD}-{invoiceld}.pdf

Errors

PermissionException – if the order is not accessible to the authenticated partner

cpOrderInvoiceDownloadInvoice()

Downloads the warranty invoice as PDF.

Allowed roles

PlatformOwner

Allowed warranty statuses

set, paid, expired

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	integer	
type	M	enum	Invoice type (advance, warranty, warrantyInvitation)
invoiceld	O	integer	Id of invoice

Output parameters

parameter	M/O	data type	explanation, example
file	M	string	Encoded in BASE64.

Implementation

See cpWarrantySet() for where to read PDF from.

Errors

PermissionException – if the order is not accessible to the authenticated partner

cpOrderInvoiceSetPaid()

Marks the warranty as paid.

Allowed roles

PlatformOwner

Allowed warranty statuses

set, expired

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	integer	
type	M	enum	Invoice type (advance, warranty)

Output parameters

None

Implementation

Change the warranty status in MCD.CP_ORDER to "paid".

Errors

PermissionException – if the order is not accessible to the authenticated partner

cpOrderInstallationList()

Returns the list of installations of an order.

They are ordered by creation time (descending).

Allowed roles

PlatformOwner, HostServiceProvider, ServiceOrganization

Input parameters

parameter	M/O	data type	explanation
orderId	M	integer	

Output parameters

parameter	M/O	data type	explanation
installations			
installation	0..N	[]	
id	M	integer	
status	M	string	
location	M		
country	M	string	2 char code
zip	M	string	5 digits
city	M	string	
cityDistrict	O	string	
street	M	string	
houseNumber	M	string	
orientationNumber	O	string	
floor	O	string	
unit	O	string	Flat number

gps	O		
latitude	M	decimal	
longitude	M	decimal	
installer	M		
id	M	integer	
name	M	string	
shortName	M	string	
dateTime	M	dateTime	
duration	M	integer	In minutes
note	O	string	
created	M	datetime	
parameters	M		
parameter	0..N	[]	
name	M	string	
value	M	string	
components	M		
component	0..N	[]	
id	M	integer	
orderComponentId	M	integer	
status	M	string	
created	M	dateTime	

Implementation

Verify the logged in partner has access to the order.

Load from MCD.CP_ORDER_INSTALLATION, MCD.CP_ORDER_INSTALLATION_PARAMETER, MCD.CP_ORDER_INSTALLATION_COMPONENT.

Errors

PermissionException – if the installation is not accessible to the authenticated partner

cpOrderInstallationGet()

Returns the installations of an order.

Allowed roles

PlatformOwner, HostServiceProvider, ServiceOrganization

Input parameters

parameter	M/O	data type	explanation
installationId	M	integer	

Output parameters

parameter	M/O	data type	explanation
id	M	integer	

status	M	string	
location	M		
country	M	string	2 char code
zip	M	string	5 digits
city	M	string	
cityDistrict	O	string	
street	M	string	
houseNumber	M	string	
orientationNumber	O	string	
floor	O	string	
unit	O	string	Flat number
gps	O		
latitude	M	decimal	
longitude	M	decimal	
installer	M		
id	M	integer	
name	M	string	
shortName	M	string	
dateTime	M	dateTime	
duration	M	integer	In minutes
note	O	string	
created	M	datetime	
parameters	M		
parameter	0..N	[]	
name	M	string	
value	M	string	
components	M		
component	0..N	[]	
id	M	integer	
orderComponentId	M	integer	
status	M	string	
created	M	dateTime	

Implementation

Load from MCD.CP_ORDER_INSTALLATION, MCD.CP_ORDER_INSTALLATION_PARAMETER, MCD.CP_ORDER_INSTALLATION_COMPONENT.

Errors

PermissionException – if the installation is not accessible to the authenticated partner

cpOrderInstallationCreate()

Creates a new installation of an order.
One order can have several installations.

Allowed roles

PlatformOwner

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	integer	
location			
country	M	string	2 char code
zip	M	string	5 digits
city	M	string	
cityDistrict	O	string	
street	M	string	
houseNumber	M	string	
orientationNumber	O	string	
floor	O	string	
unit	O	string	Flat number
gps	O		
latitude	M	decimal	
longitude	M	decimal	
installerId	M	integer	MCD partner with role: PlatformOwner, HostServiceProvider, ServiceOrganization
dateTime	M	dateTime	2015-05-03T23:30
duration	M	integer	In minutes
note	O	string	

Output parameters

parameter	data type	explanation, example
installationId	integer	The ID of the created installation.

Implementation

Verify role of the installer.
Verify dateTime not in the past.
Verify duration is positive.
Write to CP_ORDER_INSTALLATION.

Errors

PermissionException – if the order is not accessible to the authenticated partner

cpOrderInstallationSet()

Changes an installation of an order.

Allowed roles

PlatformOwner

Input parameters

parameter	M/O	data type	explanation, example
installationId	M	integer	
installerId	O	integer	MCD partner with role: PlatformOwner, HostServiceProvider, ServiceOrganization
dateTime	O	dateTime	2015-05-03T23:30
duration	O	integer	In minutes
note	O	string	

Output parameters

None

Implementation

Verify installation status is new.

IF installerId

Verify role of the installer.

ENDIF

IF dateTime

Verify dateTime not in the past.

ENDIF

IF duration

Verify duration is positive.

ENDIF

Write to CP_ORDER_INSTALLATION.

Errors

PermissionException – if the installation is not accessible to the authenticated partner

cpOrderInstallationChangeStatus()

Changes the status of an installation.

Allowed roles

PlatformOwner, HostServiceProvider, ServiceOrganization

Input parameters

parameter	M/O	data type	explanation, example
installationId	M	integer	
status	M	enum	The new status of the installation to be set.

Output parameters

None

Implementation

For allowed transitions between order states see P:\MVNO\Q\phaseII\Q2 – UML - vX.zargo.

Create record in MCD.CP_ORDER_INSTALLATION_HISTORY with old status.

Change status in MCD.CP_ORDER_INSTALLATION.

On status requested

Create RT and send mail RT4 to the installer.

Save ticket ID into MCD.CP_ORDER_INSTALLATION_PARAMETER "installationRequestTicketId".

Change status of all the CP_ORDER_COMPONENTS in CP_ORDER_INSTALLATION_COMPONENTS to "realizationRequested".

If some of them have "joinedToComponentId" parameter, change the status of these CP_ORDER_COMPONENTS as well.

On status started

Change status of RT "installationRequestTicketId" to "open".

Change status of all the CP_ORDER_COMPONENTS in CP_ORDER_INSTALLATION_COMPONENTS to "realizationStarted".

If some of them have "joinedToComponentId" parameter, change the status of these CP_ORDER_COMPONENTS as well.

On status finished

Change status of RT "installationRequestTicketId" to "resolved".

Change status of all the CP_ORDER_COMPONENTS in CP_ORDER_INSTALLATION_COMPONENTS to "realizationFinished".

If some of them have "joinedToComponentId" parameter, change the status of these CP_ORDER_COMPONENTS as well.

Set CP_ORDER_COMPONENT_PARAMETER "installationFinishDate" to actual date in format "yyyy-mm-dd" for

CP_ORDER_COMPONENTS "joinedToComponentId" (it means not for HW components)

On status cancelled

Nothing special.

Errors

PermissionException – if the installation is not accessible to the authenticated partner

InvalidStateException – if the current installation status cannot be changed to the new one

cpOrderInstallationHistory()

Returns the history of an installation.

Allowed roles

PlatformOwner, HostServiceProvider, ServiceOrganization

Input parameters

parameter	M/O	data type	explanation, example
installationId	M	integer	

Output parameters

parameter	M/O	data type	explanation, example
states	M		
state	0..N	[]	Ordered from the oldest (validUntil) to the newest.
validUntil	M	datetime	2015-10-17T13:22:07
status	M	enum	

Implementation

Load from MCD.CP_ORDER_INSTALLATION_HISTORY.

Errors

PermissionException – if the installation is not accessible to the authenticated partner

cpOrderInstallationComponentCreate()

Adds an order's component into an installation.

Allowed roles

PlatformOwner

Input parameters

parameter	M/O	data type	explanation, example
installationId	M	integer	
orderComponentId	M	integer	

Output parameters

parameter	data type	explanation, example
installationComponentId	integer	The ID of the created installation component.

Implementation

Verify installation status is new.

Verify orderComponentId belongs to the same order as the installation.

Verify orderComponentId is not already in another installation having status other than cancelled.

Write to MCD.CP_ORDER_INSTALLATION_COMPONENT.

cpOrderInstallationComponentChangeStatus()

Changes the status of an installation component.

Allowed roles

PlatformOwner, HostServiceProvider, ServiceOrganization

Input parameters

parameter	M/O	data type	explanation, example
installationComponentId	M	integer	
status	M	enum	The new status of the installation component to be set.

Output parameters

None

Implementation

Verify the logged in partner has permission to the installationComponentId. PlatformOwner has always. Others only if they are installers of the installation.

For allowed transitions between component states see P:\MVNO\Q\phaseII\Q2 – UML.zargo.

Create record in MCD.CP_ORDER_INSTALLATION_COMPONENT_HISTORY.

Change status in MCD.CP_ORDER_INSTALLATION_COMPONENT.

On status new

Nothing special.

On status readyToUnstock

Nothing special.

On status unstocked

TODO: in the future send request to VirtualStore

Errors

PermissionException – if the installation is not accessible to the authenticated partner

InvalidStateException – if the current order status cannot be changed to the new one

cpOrderInstallationComponentHistory()

Returns the history of an installation component.

Allowed roles

PlatformOwner, HostServiceProvider, ServiceOrganization

Input parameters

parameter	M/O	data type	explanation, example
installationComponentId	M	integer	

Output parameters

parameter	M/O	data type	explanation, example
states	M		
state	0..N	[]	Ordered from the oldest (validUntil) to the newest.
validUntil	M	datetime	2015-10-17T13:22:07
status	M	enum	

Implementation

Load from MCD.CP_ORDER_INSTALLATION_COMPONENT_HISTORY.

Errors

PermissionException – if the installation is not accessible to the authenticated partner

cpOrderTaskList()

Returns the list of tasks for orders.

Allowed roles

PlatformOwner, HostServiceProvider

Input parameters

parameter	M/O	data type	explanation
day	M	date	Tasks scheduled on this day are returned.
missed	O	bool	Whether to include missed tasks as well. They are scheduled before <day> and not resolved yet.
type	O	string	Limit to this task type only. See output parameter tasks/task/type.
orderId	O	integer	Limit to tasks of this order.
person	O		
lastName	O	string	Partial match.
company	O		
name	O	string	Partial match.
resolverId	O	integer	ID of partner (resolver) for which we want list of tasks

Output parameters

parameter	M/O	data type	explanation
tasks	M		
task	0..N	[]	
id	M	integer	
orderId	M	integer	Load more data by cpOrderGet().
orderComponentId	O	integer	If the task is related up to a particular order's component. Usually is.
type	M	enum	• simAssignment • simSending • installationAssignment
scheduled	M	date	When to resolve.
resolved	O	datetime	If and when it has been resolved.
resolver	O		
id	M	integer	
shortName	M	string	
name	M	string	
customer	M		
id	M	long	
name	M	string	Human readable name (COMPANY or PERSON)
provider	O		
id	M	long	
name	M	string	Human readable name

Implementation

PlatformOwner can see tasks of orders with customers he owns.

Load unresolved tasks from MCD.CP_ORDER_TASK.

Order by: scheduled, orderId, orderComponentId, type. We want tasks of the same order/component/type to be presented close to each other.

HostServiceProvider can see only his tasks.

cpBundleTariffList()

Returns the list of “bundle” tariffs.

Input parameters

None

Output parameters

parameter	M/O	data type	explanation, example
bundleTariffs	M		
bundleTariff	0..N	[]	
id	M	long	
name	M	string	
shortName	M	string	
tariffs	M		
tariff	0..N	[]	
id	M	long	
name	M	string	
shortName	M	string	

Implementation

Load from BSS.s_tariffBundles.

Filter by BSS.s_tariff_partner by the calling partner. Apply to the “bundle” tariff and to the contained “normal” tariffs as well.

cpComponentSubcategoriesList()

Returns the list of the (wholesale) component subcategories.

Input parameters

parameter	M/O	data type	explanation, example
partnerId	O	long	Filter by HostServiceProvider.
tariffId	O	long	Limit to (wholesale) subcategories mapped to this retail tariff.
componentCategoryId	O	long	
componentSubcategoryId	O	long	
feasibility	O	string	locationBased, perRequest

Output parameters

parameter	M/O	data type	explanation, example
subcategories	M		
subcategory	0..N	[]	
id	M	long	
code	M	string	WS_KABEL_D3

name	M	string	
feasibility	M	string	locationBased, perRequest
category	M		
id	M	long	
code	M	string	NET
name	M	string	
hostServiceProviders	M		
hostServiceProvider	0..N	[]	
id	M	long	
name	M	string	
shortName	M	string	

Implementation

Load from BSS.s_componentSubcategories.

Join BSS.s_partner_componentSubcategories to find out, which HostServiceProviders provide.

Limit by BSS.s_tariffRetailWholesale if tariffId.

cpComponentLocationList()

Returns the list of component locations of a HostServiceProvider.

Allowed roles

PlatformOwner, HostServiceProvider

Input parameters

parameter	M/O	data type	explanation, example
pageSize	0	long	If not specified, no limitation applies.
page	0	long	First page = 1.
partnerId	0	integer	Filter by partner. Can be used only by role PlatformOwner. Role HostServiceProvider is always limited to itself.
tariffId	0	long	Limit to (wholesale) subcategories mapped to this retail tariff.
componentCategoryId	0	integer	
componentSubcategoryId	0	integer	
externalId	0	string	Exact match.
status	0	string	One of: available, temporaryUnavailable
gps	0		
latitude	M	decimal	
longitude	M	decimal	
zoom	M	integer	0..N Based on Google maps. Approximately: 1 ... world 5 ... landmass/continent 10 ... cities 15 ... streets 20 ... buildings

			No data are returned for zoom < 15.
zip	0	string	Exact match.
city	0	string	Partial match.
cityDistrict	0	string	Partial match. Returns locations with no cityDistrict as well.
street	0	string	Partial match. Returns locations with no street as well.
houseNumber	0	string	Partial match. Returns locations with no house numbers as well.
plugCategory	0	string	
promotion	0	string	Partial match.
description	0	string	Partial match.
note	0	string	Partial match.

Output parameters

parameter	M/O	data type	explanation, example
total	M	integer	Total number of locations without pagination.
locations	M		
location	0..N	[]	
partnerId	M	integer	
componentCategory	M		
id	M	long	
code	M	string	
name	M	string	
componentSubcategory	M		
id	M	long	
code	M	string	
name	M	string	
externalId	M	string	
status	M	enum	available, temporaryUnavailable
zip	M	string	
city	M	string	
cityDistrict	O	string	
street	O	string	
houseNumbers	O		If not present, it means available for all numbers in the given street/cityDistrict/city.
houseNumber	0..N	string[]	In format <houseNumber>[/<orientationNumber>].
floor	O	string	
unit	O	string	Flat number
gps	O		If GPS is specified in the input, all output locations will have GPS as well.
latitude	M	decimal	
longitude	M	decimal	
plugCategory	M	enum	beforeHouse, beforeUnit, insideUnit, wifi

promotion	0	string	
description	0	string	
note	0	string	

Implementation

Load from MCD.PARTNER_COMPONENT_LOCATION.

Skip status = removed.

If GPS is specified in the input, limit to locations in the radius.

Limit by BSS.s_tariffRetailWholesale if tariffId.

Add all items with s_componentSubcategories.feasability = 'perRequest' despite the filter parameters. The reason is to offer ADSL providers which are not location based.

cpComponentPortSim()

Input parameters

parameter	M/O	data type	explanation, example
componentId	0	long	Must be component with SIM category.

cpComponentVerifyAvailability()

Input parameters

parameter	M/O	data type	explanation, example
componentId	M/O	long	Must be component with ADSL subcategory.
city	M/O	string	Mandatory if componentId is missing
street	M/O	string	Mandatory if componentId is missing
streetNumber	M/O	string	Mandatory if componentId is missing

Output parameters

parameter	M/O	data type	explanation, example
success	M	boolean	Email send or not.

cpTariffRetailWholesale()

Returns mapping between retail tariffs and wholesale component subcategories.

Input parameters

parameter	M/O	data type	explanation, example
tariffId	0	long	Limit to this single tariff.
componentCategoryId	0	long	Limit to tariffs having this category.
componentSubcategoryId	0	long	Limit to tariffs having this subcategory.

Output parameters

parameter	M/O	data type	explanation, example
tariffs	M		
tariff	0..N	[]	Retail
id	M	long	
name	M	string	
shortName	M	string	
subcategories	M		Wholesale
subcategory	0..N	[]	
id	M	long	
code	M	string	
name	M	string	
category	M		
id	M	long	
code	M	string	
name	M	string	
additionalItems	M		
additionalItem	0..N	[]	
ino	M	int	
description	M	string	
note	M	string	
price	M	decimal	
currency	M	string	
recurrent	M	boolean	
hidden	M	boolean	Is additional item hidden (HIDDEN in price list item subtype)
expiration	M	string	When additional item expires after used (for example 12M)
multiple	M	boolean	Can be additional item used multiple time

Implementation

Read from BSS.s_tariffRetailWholesale.

Limit by BSS.s_tariff_partner by the calling partner.

cplsMsisdnAvailableForRingin()

Returns 0 if passed msisdn is already used as self-care login otherwise returns 1.

Allowed roles

PlatformOwner, Reseller

Input parameters

parameter	M/O	data type	explanation, example
msisdn	M	string	

Output parameters

parameter	M/O	data type	explanation, example
isAvailable	M	integer	1 available, 0 not available

cplptvGetAvailablePackages()

Function returns list of available TV packages

Allowed roles

PlatformOwner

Input parameters

none

Output parameters

parameter	M/O	data type	explanation, example
packages	M		List of packages available for the device
package	0..N	[]	
id	M	integer	package id
name	O	string	package name
isOtt	O	boolean	true false

cpCpeLoginAvailable()

Checks if the given CPE login+password is available to use.

Input parameters

parameter	M/O	data type	explanation, example
login	M	string	
password	M	string	

Output parameters

parameter	M/O	data type	explanation, example
available	M	boolean	
reason	O	string	Filled only if available=0. - nonexistent ... the CPE login doesn't exist - passwordMismatch ... the input and stored password do not match - missingConfiguration ... the CPE login has no configuration file generated - missingManual ... the CPE login has no manual file generated - used ... the CPE login is already used

addSimToBasket()

This method will be used to add sim (component) into basket.

Input parameters:

parameter	M/O	data type	explanation, example
orderId	M	integer	The ID of the order.
componentId	M	integer	The ID of the component

Output parameters:

-none-

getSimBasket()

Returns the basket by the order's component ID.

Input parameters:

parameter	M/O	data type	explanation, example
orderComponentId	M	integer	The ID of the order's component

Output parameters:

parameter	M/O	data type	explanation, example
basketId	M	Integer	The ID of the basket.
status	M	string	Enum (pending, unloaded, sent)
parameters	M		
parameter	0...N	[]	
name	O	String	
value	O	String	
items	M		
item	0...N	[]	
componentId	M	integer	The ID of the component
msisdn	M	String	
iccid	O	Integer	
imsi	O	Integer	

removeSimFromBasket()

This method will be used to remove SIM (component) from basket.

Input parameters:

parameter	M/O	data type	explanation, example
basketId	M	integer	The ID of the basket.
componentId	M	integer	The ID of the component

Output parameters:

-noen-

setSimBasket()

This method will change status of basket.

Input parameters:

parameter	M/O	data type	explanation, example
basketId	M	integer	The ID of the basket.
status	M	String	Enum (unloaded, sent)
date	O	date	
note	O	String	

Output parameters:

-none-

getOrderSimBaskets()

This method returns all baskets for order.

Input parameters:

parameter	M/O	data type	explanation, example
orderId	M	integer	The ID of the order.

Output parameters:

parameter	M/O	data type	explanation, example
baskets	M		
basket	0...N	[]	
basketId	M	integer	The ID of the basket.
status	M	string	Enum (pending, unloaded, sent)
items	M		
item	0...N	[]	
componentId	M	integer	The ID of the component
msisdn	M	String	
ICCID	O	Integer	
IMSI	O	Integer	

cpSendCustomerNotification()

Sends sms / e-mail notification to a customer

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	long	ID of the component order
componentId	O	long	ID of the component

smsTemplateId	0	string	identification of SMS template (i.e. N1)
mailTemplateId	0	long	identification of email template (i.e. N1)

Output parameters

-none-

Implementation

Read SMS templates from MCD.SmsNotificationTemplate.code (filtr podle partnera)

Read SMS templates from MCD.EmailTemplate.code (filtr podle partnera)

NotificationVariable parameters:

- login (SELECT cm.MSISDN FROM `CUSTOMER\_MSISDN` as cm WHERE cm.CUSTOMER_ID = <customerId>);
- password (SELECT cp.VALUE FROM `CUSTOMER\_PARAMETER` as cp WHERE cp.CUSTOMER_ID = <customerId> AND cp.NAME = 'selfCare_password');
- ringInFromMsisdn: (SELECT cm.MSISDN FROM `CUSTOMER\_MSISDN` as cm WHERE cm.CUSTOMER_ID = <customerId>)

Message example:

Quadruple2: přihlaste se do samoobsluhy na <https://todo/samoobsluha>, login={login}, heslo={password}, nebo muzete zvolit prihlaseni prozvonem na uvedene cislo z vaseho cisla {login}.

cpOrderOnetimePriceGet()

Get price of one time fees for specific order.

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	long	ID of the component order

Output parameters

parameter	M/O	data type	explanation, example
onetimePriceSum	M		
netto	M	integer	Price without VAT
brutto	M	integer	Price with VAT
currency	M	string	Example : CZK

Implementation

Data is read from BSS API method calculateTotalPrice section onetime→sums .

cpOrderPayOnetimeItems()

This method pays one time fees that are assigned into order at the time.

Input parameters

parameter	M/O	data type	explanation, example
orderId	M	long	ID of the component order

Output parameters

– none –

cpCalculatePrice()

This method pays one time fees that are assigned into order at the time.

Input parameters

parameter	M/O	data type	explanation, example
orderId	O	long	ID of the component order
bundleTariffId	O	long	ID of bundle tariff
calculatedTariffs	M		
calculatedTariff	0..N	[]	
tariffId	M	long	ID of tariff
inoltems	M		
inoltem	0..N	[]	
ino	M	int	

Output parameters

parameter	M/O	data type	explanation, example
calculatedPrice	M		
discountNetto	M	decimal	Discount without VAT
discountBrutto	M	decimal	Discount with VAT
netto	M	decimal	Price without VAT before discount
brutto	M	decimal	Price with VAT before discount
finalNetto	M	decimal	Price without VAT after discount
finalBrutto	M	decimal	Price with VAT after discount
currency	M	string	Example : CZK
preferPriceWiuthVat	M	boolean	Display price with VAT (true) or without (false)

Implementation

Data is read from BSS API method calculateTotalPrice section recurrent→sums and preferPriceWiuthVat.

cpOrderCopy()

This method recreates order from already running cloud connection.

Input parameters

parameter	M/O	data type	explanation, example
connectionId	M	string	ID of the component order

Output parameters

parameter	M/O	data type	explanation, example
orderId	M	long	Id of created order

cpFeasibilityCheck()

This method check location feasibility by ruian.

Input parameters

parameter	M/O	data type	explanation, example
ruian	M	string	

Output parameters

parameter	M/O	data type	explanation, example
tarriffs	M		
tariff	0..N	[]	
id	M	long	tariff ID
name	M	string	tariff name
shortName	M	string	tariff shortname
status	M	string	partner component location status
category	M	object	
id	M	long	category ID
code	M	string	category code
name	M	string	category name
subcategory	M	object	
id	M	long	subcategory ID
code	M	string	subcategory code
name	M	string	subcategory name
hostServiceProvider	M	object	
id	M	long	partner id
name	M	string	partner name
shortName	M	string	partner shortname
additionalItems	M		
additionalItem	0..N	[]	
ino	M	string	
description	M	string	
note	M	string	

cpRuainFind()

This method searches address in RUIAN database. Works only for the Czech Republic.

Input parameters

parameter	M/O	data type	explanation, example
plainAddress	M	string	Address in plain format

Output parameters

parameter	M/O	data type	explanation, example
addresses	M	array	List of found addresses
ruian	M	string	RUIAN code
zip	M	string	Zip code
city	M	string	City
cityDistrict	M	string	City district
street	M	string	Street
houseNumber	M	string	House number

providerTaskChangeStatus()

This method changes status of provider task.

Input parameters

parameter	M/O	data type	explanation, example
providerTaskId	M	long	ID of provider task
status	M	string	Required status
resultCode	D	string	Two characters code, it is mandatory if status = finished, ignored otherwise
note	O	string	Optional note
updater	M	string	Some identification of who changes the status (e.g. login)
vlangs	O	[object]	Map of vprovider task item id and vlang value
key	M	long	providerTaskItemId
value	M	string	vlang value
externalConnectionIds	O	[object]	Map of provider task item id and external connection id
key	M	long	providerTaskItemId
value	M	string	externalConnectionId
appointment	O	dateTime	Date of scheduled appointment

Output parameters

none (throws error on failure)

providerTaskCreate()

This method creates provider task for given orderComponentIds.

Input parameters

parameter	M/O	data type	explanation, example
orderComponentId	M	[long]	List of order component Ids – only mandatory if type != troubleTicket
activationDate	O	dateTime	Required status – only used if type != troubleTicket
type	O	string	Currently only accepted value is 'troubleTicket', doesn't throw error if value is different
connectionId	D	string	Is required if type = troubleTicket
history	D	array	Is required if type = troubleTicket

Output parameters

parameter	M/O	data type	explanation, example
providerTaskId	M	long	ID of created provider task

providerTaskGet()

This method return provider task for given providerTaskId.

Input parameters

parameter	M/O	data type	explanation, example
providerTaskId	M	long	Requested provider task id.

Output parameters

parameter	M/O	data type	explanation, example
id	M	long	List of order component IDs
orderId	O	long	Required status
virtualGenericServiceId	M	long	Customer ID
type	M	enum	- create - change - troubleTicket - delete
status	M	enum	- new - accepted - finished - cancelled
resultCode	O	string	Two characters code represents task result, it is mandatory if status = finished
customerAppointment	M	boolean	Is technician required?
authorizationCode	O	string	Authorization code
dateStart	M	dateTime	A date and time formatted in compliance with the ISO8601 standard must be used.
dateEnd	O	dateTime	A date and time formatted in compliance with the ISO8601 standard must be used.
duration	O	long	For feature use
note	O	string	Optional note.

updated	M	dateTime	Last update of provider task
created	M	dateTime	A date and time formatted in compliance with the ISO8601 standard must be used.
vlan	O	[long]	Array of available vlans
vlanGroup	O	string	Group of vlan
contact	M	object	Contact in provider task
person	M	object	Information about person
firstName	O	string	First name
lastName	M	string	Last name
phone	M	string	Phone
email	O	string	Email
location	M	object	Information about location/address
ruian	O	string	Ruian
country	O	string	Country
zip	O	string	ZIP (PSČ)
city	O	string	City name
cityDistrict	O	string	City district name
street	O	string	Street name
houseNumber	O	string	House number
floor	O	string	Flat floor
unit	O	string	Flat unit
hostServiceProvider	O	object	
id	M	long	Provider id
name	M	string	Provider name
shortName	M	string	Provider short name
partner	O	object	
id	M	long	Partner Id
name	M	string	Partner name
shortName	M	string	Partner short name
taskItems	O	[object]	
id	M	long	Provider task item Id
parentId	O	long	Provider task Id
action	M	string	Requested action can be create, delete, repair
vlan	O	long	Assigned vlan
externalConnectionId	O	string	Assigned externalConnectionId
product	M	object	Wholesale product
code	M	string	Product code
description	O	string	Text description of product
maxDownloadSpeed	O	string	Maximal available download speed for this product

maxUploadSpeed	O	string	Maximal available upload speed for this product
partNumber	O	string	Hardware identifier (VMG8823B50)
category	O	object	Category of product
id	M	long	Category ID
name	M	string	Category name
code	M	string	Category code
retailProductParameters	M	object	Retail product
code	O	string	Retail product code
description	O	string	Text description of retail product
maxDownloadSpeed	O	string	Maximal available download speed for this product
maxUploadSpeed	O	string	Maximal available upload speed for this product
partNumber	O	string	Hardware identifier (VMG8823B50)
category	O	object	Category of product
id	M	long	Category ID
name	M	string	Category name
code	M	string	Category code
taskHistory	O	[object]	History from db and RT.
status	M	string	Updated to status
updater	M	string	Who update task
note	O	string	Optional note written by updater on update
validFrom	M	dateTime	A date and time formatted in compliance with the ISO8601 standard must be used.
answeredQuestions	O	[object]	Questions and answers for end customer.
questionText	M	string	Text of question
answerText	M	string	Answer to question

providerTaskList()

Return list of provider tasks. The tasks are ordered by id.

Input parameters

parameter	M/O	data type	explanation
pageSize	O	long	How many orders to return. Max value 1000. Default value 100.
page	O	long	Which page to return. Default value 1.
orderBy	O	string	Can be ordered by
sorted	O	string	Can be sorted by
type	O	enum	- create - change - troubleTicket - delete
statuses	O	[enum]	new

			- accepted - rejected - pending - finished - cancelled
pids	0..N	[integer]	Provider Ids in array
dateStartFrom	O	dateTime	A date and time formatted in compliance with the ISO8601 standard must be used.
dateStartTo	O	dateTime	A date and time formatted in compliance with the ISO8601 standard must be used.
resultCode	O	dateTime	Two characters code represents task result, it is mandatory if status = finished

Output parameters

parameter	M/O	data type	explanation
total	M	integer	How many tasks have been found.
providerTasks	0..N	[object]	Array of provider tasks
providerTask	0..N	object	
id	M	long	Unique identifier of task
orderId	O	long	Order identifier
virtualGenericServiceId	M	long	Customer ID
externalConnectionId	O	string	Assigned externalConnectionId
type	M	enum	- create - change - troubleTicket - delete
status	M	enum	- new - accepted - pending - rejected - finished - cancelled
resultCode	O	string	01, 02, 03, 04
customerAppointment	M	boolean	Is technician required?
dateStart	M	dateTime	A date and time formatted in compliance with the ISO8601 standard must be used.
dateEnd	O	dateTime	A date and time formatted in compliance with the ISO8601 standard must be used.
duration	O	long	For future use.
note	O	string	Optional note
updated	M	dateTime	A date and time formatted in compliance with the ISO8601 standard must be used.
created	M	dateTime	A date and time formatted in compliance with the ISO8601 standard must be used.

Methods – Service orders (ITF)

serviceOrderList()

Returns the list of consumers orders.

The orders are ordered by creation time (descending).

Input parameters

parameter	M/O	data type	explanation
pageSize	O	integer	How many orders to return. Max value 1000. Default value 100.
page	O	integer	Which page to return. Default value 1.
orderId	O	integer	
company	O		
id	O	integer	
name	O	string	Partial match.
idNumber	O	string	Exact match.
taxId	O	string	Exact match.
owner	O		
firstName	O	string	Partial match.
lastName	O	string	Partial match.
email	O	string	Partial match.
statuses	O		
status	0..N	string[]	new, complete, approved, realized, unrealized
createdFrom	O	date	
createdTo	O	date	

Output parameters

parameter	M/O	data type	explanation
total	M	integer	How many orders have been found. It can be higher than pageSize.
orders	0..N	[]	The number of orders is limited by pageSize.
order	0..N	[]	
id	M	integer	
type	M	enum	- serviceCreation - serviceModification - serviceCancellation
serviceType	D	enum	- workplace - specialService Set if type=serviceCreation.
tariffId	D	integer	Set if type=serviceCreation. You can load more data by tariffList().
consumer	D		Data of the requested new workplace owner. Set if type=serviceCreation and serviceType=workplace.

firstName	M	string	
lastName	M	string	
email	M	string	
serviceName	D	string	Name of the requested new GID. Set if type=serviceCreation and serviceType=specialService.
status	M	enum	
genericService	D		Set if type=serviceModification/Cancellation.
gid	M	integer	
name	O	string	
owner	O		
id	M	integer	
firstName	M	string	
lastName	M	string	
email	M	string	
master	M	boolean	
customer	M		
id	M	integer	
company	O		
name	O	string	
idNumber	O	string	
taxId	O	string	
approved	O	datetime	
approver	O		
id	M	integer	
firstName	M	string	
lastName	M	string	
email	M	string	
master	M	boolean	
created	M	datetime	
creator	M		
id	M	integer	
firstName	M	string	
lastName	M	string	
email	M	string	
master	M	boolean	
components	0..N	[]	
component			
id	M	integer	
type	M	string	componentCreation, componentCancellation
category	D		Set when type=componentCreation
id	M	integer	

code	M	string	
name	M	string	
subcategory	M		
id	M	integer	
code	M	string	
name	M	string	
componentId	D	integer	Set when type=componentCancellation

Implementation

Load from MCD.SERVICE_ORDER.

serviceOrderSet()

Changes the existing order.

Input parameters

parameter	M/O	data type	explanation
orderId	M	integer	
status	O	string	It can be changed only from approved to realized/unrealized.
gid	D	integer	What GID has been created when this order has been realized? Must be specified if the order has type=serviceCreation and the status=realized.

Output parameters

none

Implementation

IF status is specified

Allow only from approved to realized/unrealized.

IF status=realized

IF order has type=serviceCreation

Verify gid is specified in input.

Set MCD.SERVICE_ORDER.GID = <gid>.

ENDIF

Set MCD.SERVICE_ORDER.STATUS = realized.

Send email SO3 to all master consumers of the customer.

IF order's gid has consumer assigned

IF order has type=serviceCreation AND order has serviceType=workplace

Send email SO3c to consumer assigned to the gid.

ELSE

Send email SO3 to consumer assigned to the gid.

ENDIF

ENDIF

ELSE # status=unrealized

Set MCD.SERVICE_ORDER.STATUS = unrealized.

Send email SO4 to all master consumers of the customer.

ENDIF

ENDIF

Errors

InvalidArgumentException – if the status is invalid

InvalidStateException – if the current order status is invalid

InvalidArgumentException – if the GID is already assigned to another order with type=serviceCreation

XML data types

customerInfo

parameter	M/O	data type	explanation, example
customerData	M		
customerId	M	integer	
isPrepaid	M	boolean	
person	O		
name	O	string	
lastName	O	string	
middleName	O	string	
gradeBefore	O	string	
gradeAfter	O	string	
birthYear	O	string	
company	O		
name	O	string	
idNum	O	string	
taxId	O	string	
project	O		
name	O	string	
projectNumber	O	string	
leaderName	O	string	
leaderEmail	O	string	
solution	O	string	
customerNumber	O	string	
customerName	O	string	
externalRef	O	string	
orderedBy	O	string	
delivAddress	O	string	
delivEmail	O	string	
projectType	O	string	
projectTypeDesc	O	string	
phone	O	string	
email	O	string	
scPhone	O	string	
scPassword	O	string	
blocked	O	boolean	
postBilling	O	boolean	
agreeWithSendDocumentsByEmail	O	boolean	
billingEmail	O	string	

defaultTariffId	O	long	Default BSS tariff ID. Currently used to preselect tariff in PPW.
gpcsAccount	O		
...			
gs	0..N		
gid	M	integer	
name	O	string	
tariffId	M	integer	
tariffName	M	string	
serviceClass	O		
serviceClassId	M	integer or decimal	
name	O	string	
consumer	O		
id	M	integer	
firstName	M	string	
lastName	M	string	
email	M	string	
master	M	boolean	
reseller	O		
id	M	integer	
name	M	string	
shortName	M	string	
gpcsAccounts	O		
gpcsAccount	0..N	[]	
accountNumber	M	string	
currency	O	string	
accountBalance	O	decimal	
availableBalance	O	decimal	
floor	O	decimal	
expiryDate	O	date	
period	O	string	(month, year)
components	D		Must be specified for a component product, even if empty. Must not be specified otherwise.
component	0..N	[]	
id	M	string	Global ID from COMPONENT_VIEW.
category	M		
id	M	long	
code	M	string	
name	M	string	
subcategory	M		
id	M	long	
code	M	string	

name	M	string	
startDate	M	date	YYYY-MM-DD
endDate	O	date	YYYY-MM-DD
active	M	boolean	
hostServiceProvider	M		
id	M	long	
name	M	string	
shortName	M	string	
parameters	M		Loaded from NET_CONNECTION_PARAMETER, TV_CONNECTION_PARAMETER, COMPONENT_PARAMETER, or any other component-like table
parameter	0..N	[]	
name	M	string	
value	M	string	
sim	D		Must be specified for component category SIM.
msisdn	M	string	
imsi	O	string	
iccid	O	string	
puk	O	string	
puk2	O	string	
selfCareMaster	M	boolean	
simService	0..N	[]	The same structure as gs/number/simService.
autoTopup	O	complex	The same structure as gs/number/autoTopup.
netConnection	D		Must be specified for component category NET.
country	M	string	
zip	M	string	
city	M	string	
street	O	string	
houseNumber	M	string	
floor	O	integer	
flatNumber	O	string	
tvConnection	D		Must be specified for component category TV. It has the same structure as netConnection (at the moment, but it can change in the future).
cloudConnection	D		Must be specified for component category CLOUD.
subscriptionId	M	string	
number	0..N		
msisdn	M	string	

imsi	O	string	
iccid	O	string	
puk	O	string	
puk2	O	string	
active	O	boolean	
suspended	O	boolean	
selfCareMaster	O	boolean	
simService	0..N		
name	M	string	
value	O	string	
pending	O	boolean	
upToDate	O	boolean	
autoTopup	O		
enabled	O	boolean	
threshold	O	integer	
atCreditExpiry	O	boolean	
amount	O	integer	
authorizedSince	O	dateTime	
notifyMsisdn	O	string	
deleted	O	boolean	
removeDate	O	dateTime	
autoPaymentAuthorizedSince	O	dateTime	YYYYMMDDHHMMSS, for example 20160307162648
autoPaymentAuthorizedValue	O		
amount	M	long	
currency	M	string	3 character uppercase
notes	0..N	customerNote	
bundleTariff	O		
id	M	long	ID of bundle tariff
name	M	string	Name of bundle tariff
shortname	M	string	Shortname of bundle tariff

customerNote

parameter	M/O	data type	explanation, example
note	M		
customerId	M	long	
type	M	enum	
body	M	string	
created	M	dateTime	

paymentHistoryInfo

parameter	M/O	data type	explanation, example
paymentHistoryInfo			
numberOfPendingInvoices	M	long	Number of pending (not paid) invoices.
numberOfPendingInvoicesAfterDueDate	M	long	Number of pending (not paid) invoices after due date.
averageDelayOfPayment	O	float	Average delay of payment in days (not present if there is no paid invoice)

Component product implementation notes

RT

See perl module RT::Client::REST. It is implemented in project CCD middleware (<https://svn2.maternacz.com/mccz/ccd-middleware>). See RtTroubleTicketModule.pm.

For login to RT use RT_USERNAME+RT_PASSWORD of the partner, that creates the ticket (column from).

For ticket queue and owner use RT_QUEUE and RT_USERNAME of the partner, that is assigned to the ticket (column to).

The RT_* parameters are stored in MCD.PARTNER_PARAMETER.

Created RT ticket ID is saved to MCD.CP_ORDER_PARAMETER and MCD.CP_ORDER_COMPONENT_PARAMETER (column ticket parameter name).

order type	order status	email template	from	to	ticket parameter name
serviceAdd	complete	RT3	Reseller	PlatformOwner	completeRequestTicketId

order type	component category	component status	email template	from	to	ticket parameter name
serviceAdd componentAdd componentChange	NET	feasibilityRequested	RT1	PlatformOwner	HostServiceProvider	feasibilityRequestTicketId

Data from MCD.EmailTemplate is used for both the RT creation and email message.

After ticket creation send an email to SERVICE_EMAIL of the HostServiceProvider.

Email templates

Note, the URL to PPW is hardcoded for every partner. It cannot be filled as a variable, because PMW partner doesn't know it.

code	subject	textBody	variables
RT1	Dostupnost služby [objednávka {orderId}]: {customerName}	Prosím o prověření dostupnosti služby. http://10.8.0.2:1059/ppw/orderComponent.mmp?orderId={orderId}	orderId orderComponentId customerName ... either firstName+lastName or companyName
RT2	Nová objednávka {orderId}: {customerName}	Byla vytvořena nová objednávka {orderId}. http://10.8.0.2:1059/ppw/orderCustomer.mmp?id={orderId}	orderId customerName ... either firstName+lastName or companyName
RT3	Potvrzení objednávky {orderId}: {customerName}	Prosím o potvrzení nové objednávky. http://10.8.0.2:8080/ppw/orderStatus.mmp?id={orderId}	orderId customerName ... either firstName+lastName or companyName
RT4	Žádost o instalaci [objednávka {orderId}]: {customerName}	Prosím o provedení instalace. http://10.8.0.2:1059/ppw/xxx.mmp?installationId={installationId}	orderId customerName ... either firstName+lastName or companyName installationId ... CP_ORDER_INSTALLATION
SO1	Order {orderId} approval request from {creatorEmail}	User {creatorEmail} asks for approval of the order {orderId}. To review the order visit: {orderDetailUrl}	orderId ... MCD.SERVICE_ORDER.ID creatorEmail ... who created the order orderDetailUrl ... URL to the order detail
SO2	Order {orderId} realization request from {creatorEmail}		orderId ... MCD.SERVICE_ORDER.ID

			creatorEmail ... who created the order
SO3	Order {orderId} from {creatorEmail} is realized	New workplace has been created.	orderId ... MCD.SERVICE_ORDER.ID creatorEmail ... who created the order
SO3b	Order {orderId} from {creatorEmail} is realized	New workplace has been created. You can log in at {scwUrl} as {scwUsername} / {scwPassword}.	orderId ... MCD.SERVICE_ORDER.ID creatorEmail ... who created the order scwUrl ... scwUsername ... email of the workplace owner scwPassword ...
SO4	Order {orderId} from {creatorEmail} is rejected	ITF rejected the realization of the order.	orderId ... MCD.SERVICE_ORDER.ID creatorEmail ... who created the order
SO5	Auto cancellation of order {orderId} created by {creatorEmail}		orderId ... MCD.SERVICE_ORDER.ID creatorEmail ... who created the order orderApprovalTimeout ... MCD.CUSTOMER_PARAMETER ORDER_APPROVAL_TIMEOUT
SCW1	Self care web: password reset	To reset your password to self-care web, continue to {passwordResetUrl}	passwordResetUrl ... URL to pmweu-rest token, that resets the password to SCW

Errors (SOAP faults)

name	code	S/R	explanation, <i>example</i> , hint
InvalidArgumentException	1001	S	Parameter customerId (xyz) is not long. Call with correct value.
InvalidArgumentExceptionImsi	10011	S	Parameter imsi (xyz) is not valid IMSI (14-15 digits)
InvalidArgumentExceptionMsisdn	10012	S	Parameter msisdn (xyz) is not MSISDN (9-15 digits)
InvalidArgumentExceptionCtid	10013	S	Parameter ctid (xyz) is not valid CTID (14 digits)
InvalidArgumentExceptionAddress	10014	S	Address is not valid: <reason>
MissingArgumentException	1002	S	Required long parameter customerId missing. Fill-in mandatory argument.
ArgumentOutOfRangeException	1003	S	Parameter dueDateDays (1000) must be within 0..365. Call with correct value.
CustomerNotFoundException	1004	S	<i>Customer 1308211 not found.</i> Call with existing customerId.
CustomerBlockedException	1005	S	<i>Customer 13082 is blocked.</i> Call with customerId that is not blocked.
GenericServiceNotFoundException	1006	S	<i>GID 15168 not found.</i> Call with existing gid.
GenericServiceDeactivatedException	1007	S	<i>GID 15169 deactivated.</i> Call with gid that is not deactivated.
GenericServiceDeactivationPlannedException	1008	S	GID 15171 deactivation is already planned. Call with gid that has not planned deactivation.
TariffNotPrepaidException	1009	S	<i>Tariff 1 is not prepaid.</i> Call with tariffId that belongs to prepaid product.
TariffNotPostpaidException	1010	S	<i>Tariff 2 is not postpaid.</i> Call with tariffId that belongs to postpaid product.
TariffNotFoundException	1011	S	<i>Tariff 9 not found.</i> Call with existing tariffId.
MsisdnNotFoundException	1012	S	MSISDN 420123456789 not found. Call with existing msisdn.
ArgumentDependencyException	1013	S	Exactly one of customerId or gid must be provided for this method. Call with correct argument combination.
NoMsisdnsException	1014	S	<i>No MSISDNs found.</i> Call with arguments that specify at least one MSISDN.
PricelistNotFoundException	1015	S	<i>Pricelist 234 not found.</i> Call with existing pricelistId.
PricelistVersionNotFoundException	1016	S	<i>No active Pricelist version found for id 235.</i> Call with pricelistId that has active version.
NoPreferredFreeMsisdnException	1018	S	<i>Preferred MSISDN 420123456789 is not available.</i> Call with another preferredMSISDN or omit optional preferredMSISDN.
UnknownPartnerException	1019	S	<i>Unknown partner.</i> Call with existing partner credentials. Report to support.
PartnerFormatException	1020	S	<i>Partner format error.</i> Call with correct partner credentials. Report to support.
UnsupportedSoapOperation	1021	S	<i>Unsupported operation xyzMethod.</i> Make call according to provided WSDL. Use existing methods (operations).
ImsiNotAvailableException	1022	S	Requested IMSI does not exist/is already used in different subscription/marked as LOST/EXPIRED/is not available for tariff
PackageNotFoundException	1023	S	GenericService GID xyz has no orderable Package INO 'xyz'
DuplicateRecurrentPackageException	1024	S	PackageItem with INO 'xyz' is recurrent and it has been already ordered
MsisdnNotBelongingToGidException	1025	S	MSISDN xyz not belonging to GID xyz
TooManyNumbersPerGid	1026	S	Adding more numbers to generic service (GID) is not allowed due to product/pricelist limitation. Hint: create another generic service (gsAdd).

NoSuchSmsTemplateException	1027	S	Template of the requested code (Mxx) and language does not exist.
NoSuchInvoiceException	1028	S	Invoice of requested number does not exist.
NoPortOutRequestException	1029	S	
CustomerIsNotPrepaidException	1030	S	
CustomerPrepaidUnableTopupCreditException	1031	S	
CustomerPrepaidUnableToGetBalanceException	1032	S	
MnpPortInInvalidCtidException	1033	S	
MnpPortInDuplicateCtidException	1034	S	
MnpPortInMsisdnCannotBeUsedException	1035	S	
CustomerPrepaidTopupVoucherNotAcceptedException	1036	S	
CustomerPrepaidTopupVoucherIsNotPrepaidException	1037	S	
CustomerPrepaidTopupVoucherTooManyAttemptsException	1038	S	
CustomerPrepaidTopupVoucherMaxMonthlyValueReachedException	1039	S	
CustomerPrepaidTopupVoucherUnknownException	1040	S	
SelfCarePassword2DigitUnknownException	1041	S	
InvalidStateException	1042	S	The entity is in a state, that does not allow the called operation. Examples: - Trying to modify a closed order. - Trying to pay an already paid invoice. See method section "Allowed statuses" if any.
PackageOrderByVoucherException	1043	S	
MnpPortInPromoCodeIsNotValidException	1044	S	
PartnerRoleException	1045	S	Insufficient partner roles: roleA, roleB. Allowed roles: roleC, roleD.
DuplicateMsisdnException	1046	S	MSISDN 420123456789 is already occupied.
DuplicateEmailException	1047	S	Email alice@wonderland.com is already occupied.
ActiveConsumerException	1048	S	Consumer John Doe is still assigned to a GID.
ComponentNotFoundException	1049	S	Trying to work with component, that does not exist.
UpdateSimServicesInvalidFilter	1050	S	Invalid filter. Only all volatile simServices is allowed
UpdateSimServicesListOfServicesMissing	1051	S	simServices attribute is missing with filter = "simServices"
UpdateSimServicesInvalidSimServiceId	1052	S	Invalid sim service id
UpdateSimServicesMinimumDelayForOneMsisdnNotReached	1053	S	Request for the same MSISDN in less than 3600 sec.
UpdateSimServicesMinimumDelayNotReached	1054	S	Next request for synchronization in less than 20 sec.
UpdateSimServicesDailyLimitReached	1055	S	Max count of synchronization requests was reached (100/per day)
SelfCarePasswordSetException	1056	S	New self-care password was not accepted
GenericServiceStornoDeactivationException	1057	S	GID 10562 deactivation is not planned, it can NOT be cancelled.
UnknownTypeException	2001	R	Internal error: Unknown type smallInt to validate. Report to support.
ExternalCommandException	2002	R	cdrGetCmd error 3: some error. Report to support.
MandatoryProductOptionNotFoundException	2003	R	Monthly limit for tariff 1 not found. Report to support.

UnsupportedObjectState	2004	R	<i>Customer has tariffs from different products.</i> Report to support or call with another object's (e.g. customer that has supported state) specification.
UnsupportedActivitiException	2005	R	Operation sim_deactivate not configured. Report to support.
IOException	2006	R	Unable to compile schema from WSDL. Report to support.
NoCustomerFound	2007	R	<i>No customer found for id = 123456.</i> Call with arguments that specify at least one customer.
NoFreeSimException	2008	R	No free SIM card for partner 1. Report to support.
NoFreeMsisdnException	2009	R	No free MSISDN for partner 1. Report to support.
NoRecordFoundException	2010	R	No record found in CUSTOMER for ID=123456 while updating. Report to support.
DataIntegrityException	2011	R	Invalid package subtype 'xyz': invalid unitAmounts 'xyz'
<i>database internal errors</i>	9xxxx	R	Internal error, report to support.

(S)ender [client] error, (R)eceiver [server] error

Errors common to all methods

InvalidArgumentException, UnknownPartnerException, PartnerFormatException, PartnerRoleException, UnsupportedSoapOperation, UnknownTypeException, UnsupportedObjectState, UnsupportedActivitiException, IOException, NoRecordFoundException, *database internal errors*.

Error report example

```
<soap:Envelope soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:namespace1="http://namespaces.soaplite.com/perl" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Sender</faultcode>
      <faultstring>ERR1001-InvalidArgumentException: Parameter customerId (xyz) is not long</faultstring>
      <detail>
        <InvalidArgumentException xsi:type="namespace1:InvalidArgumentException">
          <message xsi:type="xsd:string">Parameter customerId (xxx) is not long</message>
          <code xsi:type="xsd:int">1001</code>
        </InvalidArgumentException>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Request-response log

Separate database connection for per-request logging.

Table name "pmwpartnerLog"

column	type	content
id	bigint(20)	Record id
remoteUser	varchar(100)	HTTP-Basic username (partner)
method	varchar(100)	SOAP method name
reqTime	timestamp	request timestamp
reqDump	text	dump of all request parameters
respDump	text	dump of all response parameters
duration	int	request-response duration (milliseconds)

Data retention is performed externally (cron job that deletes old records)

Workflow engine (activiti)

Activiti user guide <http://http://activiti.org/userguide/index.html>, Chapter 15. REST API, "start process instance"

Example

Starting process "create_prepaid_account2:3:5329" via REST call

request:

```
curl --basic --user kermit:kermit -d '{"processDefinitionId":"create_prepaid_account2:3:5329","partnerId":"4","tariffId":"3"}'
"http://localhost:8080/activiti-rest/service/process-instance"
http://localhost:8080/activiti-rest/service/process-instance ... URL of activiti-rest service (pmwpartner.conf global)
kermit:kermit ... login credentials (pmwpartner.conf global)
create_prepaid_account2:3:5329 ... name of process to start (pmwpartner.conf per-method config)
"partnerId":"4","tariffId":"3" ... dynamic variables (coded inside methods)
```

response:

```
{"id":"5240","businessKey":null,"processInstanceId":"5240","processDefinitionId":"create_prepaid_account2:3:5329"}
```

The response must be logged (processInstanceId for tracing purposes).

History of changes

Date	Version	Author	Description
2006-03-06	1.1	VPi	Document created.
2008-06-12	1.2	VPi	Company logo update
2010-04-09	1.3	VPi	Company logo update
2019-10-22	1.4	MSp	Company logo update
11.3.2013	3	Vojtěch Pithart	simProfile: m_* parameters as dynamic configurable list simServiceList: new method WSDL pmwpartner-3.wsdl reflecting changes
17.3.2013	3b	Pavel Kříž	Doplnění jsem chybových stavů. Oprava popisu tabulky pmwpartnerLog.
22.3.2013	4a	Vojtěch Pithart	simProfile: new elements in response: activationPending a deactivationPending
2. 5. 2013	4e	Jiří Brachtl	simServiceList: změna l10nStrings
5.5.2013	5a	Vojtěch Pithart	customerInfo/gs/number, nový tag <suspended> true/false pricelistItem: added call-setup-fee (value → fixPrice+unitPrice) packagesAvailable packagesList packageOrder packageRemove
28.5.2013	6a	Vojtěch Pithart	CustomerCreate{Prepaid,Postpaid},gsAdd,simAssign,simSwap - optional IMSI extended to "IMSI or ICCID".
29. 5. 2013	6b	Karel Kostka	Inserted next exceptions Parameter iccid was added into customerInfo.gs.number definition Customers are searched also by the both iccid and msisdn
5.6.2013	6d	Vojtěch Pithart	SMS notifications
10.7.2013	7a	Vojtěch Pithart	New method simMigrate; new parameter withoutMSISDN for create-methods.
1. 8. 2013	8a	Karel Kostka	ImsiCommonPartGet, IccidCommonPartGet, gsSearch
20. 9. 2013	8b	Karel Kostka	new: CdrUsageTypeList new: portOutList new: portOutConfirm new: portInList m_voiceMail/pin validity and strength of password Parameter "prepaid" added to the method TariffList
21. 10. 2013	9a	Karel Kostka Vojtěch Pithart	Optional parameter searchKeyField to CustomerSearch method simProfileSet: dependency explanation for m_barringByCustomer/*
21. 1. 2014	9b	Karel Kostka	New response attribute paymentType in the method packagesList
27. 1. 2014	9c	Karel Kostka	Inserted parameter activeTo into the type packageInstanceType
6. 2. 2014	9d	Jiří Brachtl	Added customerFreeUnits method.
28. 3. 2014	9e	Karel Kostka	Added dataUsage method
23. 4. 2014	9f	Karel Kostka	customerInfo/gs new tags <tariffId>, <tariffName>

			customerInfo/gs/number, new tags <puk>, <puk2> customerInfo/customerData extended by attribute <isPrepaid> true/false simServiceDefinitionType extended by attribute <id> new method UpdateSimServices
6. 5. 2014	9g	Vojtěch Pithart	Detailed description of invoiceType (customerInvoiceGet)
23. 5. 2014	9h	Karel Kostka	Added portInOverExistingMsisdn method new return parameters <activationType>, <swapFromMsisdn> in portInList method
3. 7. 2014	9i	Karel Kostka	Note for packagesList() changed. The method returns all packages now.
8. 9. 2014	10a	Vojtěch Pithart	- WSDL: customerData/gs/number: new element selfCareMaster (boolean) - New method: selfCarePasswordSet(), selfCareMasterSet() - customerFreeUnits renamed to gsFreeUnits
10. 9. 2014	10b	Tomáš Menclík	Added parametr suspendReason to simSuspend method
16. 9. 2014	10c	Jan Bednář	New methods: edrTriggerList(), edrTriggerSet(), emailAddressList(), emailAddressSet(), EventTypeList()
30.10.2014	11a	Vojtěch Pithart; Jan Bednář	PackageOrder(): new parameter onActivate; New method: partnerSSOtokenGet
18.11.2014	11b	Lukáš Veselka	New method: autoPaymentRemove()
4.2.2015	12a	Lukáš Veselka	- WSDL packageInstanceType: - new elements origValue, leftOverValue - remove element paymentType - PackagesList(): new elements origValue, leftOverValue; remove element paymentType
20.4.2015	12b	Lukáš Veselka	customerInfo/gs new tags <removeDate></removeDate>; GsRemove(): new parameter removeAfter; new method: gsRemoveCancel
21. 4. 2015	12b	Karel Kostka	packageRemove() - change of format MSISDN to the international form
19. 5. 2015	12c	Lukáš Veselka	Change simSuspend and simResume – check if SIM is suspended
13. 7. 2015	13a	Vojtěch Pithart	customerInvoiceGet(): added rounding to invoice structure (rounding, rounding-Transfer, total1, total2)
14. 8. 2015	13b	Karel Kostka	packageOrder() - added info about sharing packages (data per MSISDN, the other always shared)
12. 10. 2015	14a	Jiří Brachtl	Q2 (Product Q) new methods. Updated Errors chapter by real state in code. InvalidArgumentException added to errors common to all methods.
29. 10. 2015	14a	Jiří Brachtl	Added cpComponentLocationList(). Added input/output parameters to cpOrderList(). Added output parameters to cpOrderGet().
30. 10. 2015	14a	Jiří Brachtl	Added partnerList(). customerSet(): added contactPerson. cpOrderGet(): added output customer/id. cpOrderSet(): added tariffId.
3.11.2015	14a	Jiří Brachtl	cpOrderCreate(): added input user.
5.11.2015	14a	Jiří Brachtl	cpComponentProviderSearch(): removed, superseded by cpComponentLocationList().

			cpOrderComponentCreate(), cpOrderComponentSet(): touched parameters: simDeliveryType, buildingType, buildingTypeOther, connectionType, connectionTypeOther. cpComponentLocationList(): touched inputs: cityDistrict, street, houseNumber
9.11.2015	14a	Jiří Brachtl	cpOrderChangeStatus(): touched implementation
10.11.2015	14a	Jiří Brachtl	cpOrderComponentSet(): added parameter feasibilityComment cpOrderComponentHistory(): removed serviceOrganizationId, added updater cpOrderGet(), cpOrderHistory(): added updater cpOrderCreate(), cpOrderGet(), cpOrderList(): user renamed to creator cpOrderComponentChangeStatus(): touched implementation
11.11.2015	14a	Jiří Brachtl	cpOrderSet(): added note cpOrderComponentHistory(): added parameters cpOrderComponentChangeStatus(): record history
12.11.2015	14a	Jiří Brachtl	cpOrderComponentChangeStatus(): touched implementation cpOrderComponentCreate(), cpOrderComponentSet(): added parameter locationInfo cpOrderChangeStatus(), cpOrderSet(), cpOrderComponentChangeStatus(), cpOrderComponentSet(): added updater cpOrderGet(): added to component: created, updated, updater
13.11.2015	14a	Jiří Brachtl	cpOrderComponentChangeStatus(): RT and mail
18.11.2015	14b	Jiří Brachtl	TODO: Added order types. A lot of changes in all cp* methods.
20.11.2015	14b	Jiří Brachtl	cpWarrantySet(): specified invoice generation.
23.11.2015	14b	Jiří Brachtl	cpOrderComponentChangeStatus(): provisioning from "order" tables to "service" tables, added realizationComment parameter. RT chapter: ticket are created only for NET components
24.11.2015	14b	Jiří Brachtl	cpWarrantySet(): added export to PDF cpOrderGet(): added parameters under warranty Added cpWarrantyDownloadInvoice()
25.11.2015	14b	Jiří Brachtl	cpOrderComponentChangeStatus(): only "surviving" (marked green) parameters are copied.
26.11.2015	14b	Jiří Brachtl	simAssign(): added cpOrderComponentId, extended implementation
27.11.2015	14b	Jiří Brachtl	Added partnerGet() cpOrderChangeStatus(): added RT3 on status=complete
9.12.2015	14b	Jiří Brachtl	CustomerInfo returned from various methods extended by components. See XML data types -> customerInfo. cpTariffList(): enabled for Reseller, touched implementation cpOrderGet(): removed serviceOrganizationId cpOrderGet(), cpOrderComponentHistory(): hostServiceProviderId changed to complex element.
10.12.2015	14b	Jiří Brachtl	cpOrderHistory(): allowed for Reseller cpOrderComponentHistory(): allowed for Reseller and HostServiceProvider (limited to himself)
17.12.2015	14b	Jiří Brachtl	cpOrderCreate(): phone is required to be MSISDN, added check and insert to MCD.CUSTOMER_MSISDN.
12.1.2016	14b	Jiří Brachtl	New method: configuration() cpOrderCreate(): default value of collectiveBilling cpOrderSet(): touched collectiveBilling

			cpOrderComponentSet(): touched delayedBilling cpOrderComponentChangeStatus(): - on feasibilityConfirmed: parameter delayedBilling - added new state realizationReady between realizationStarted and realization-Finished cpOrderChangeStatus(): - on realizationRequested: check if collectiveBilling is possible
29.1.2016	14b	Jiří Brachtl	cpOrderGet: added collectiveBilling, added components/component/delay-edBilling
9.2.2016	14c	Tomáš Marek	New method: cplsMsisdnAvailableForRingin() cpOrderCreate(): added scPhone, scPassword cpOrderGet(): added scPhone, scPassword customerSet(): added scPhone, scPassword
17.2.2016	14c	Jiří Brachtl	configuration(): added platform/CPE_LOGIN_PREFIX cpOrderComponentChangeStatus(): touched On status realizationReady for NET
18.2.2016	14c	Jiří Brachtl	cpOrderCreate(): added collectiveBilling
23.2.2016	14c	Jiří Brachtl	cpOrderComponentSet(): allowed for Reseller, revisited access rules
23.2.2016	14c	Tomáš Menclík	Renamed method cplptvGetCustomerPairedDevices to cplptvGetCustomerDe-vices outputparameter packages moved to new method cplptvGetCustomerPac-kages
24.2.2016	14c	Tomáš Menclík	Added method cplptvCreateUser
29.2.2016	14c	Michal Keller	CpOrderSet(): allowed for Reseller CpOrderComponentChangeStatus(): On status realizationFinished for NET – new parameters cpe* added – implementation moved from status realizationRe-ady to status realizationFinished
10.3.2016	14c	Jiří Brachtl	cpOrderCreate(): added facebookProfile, touched implementation, added Dupli-cateFacebookProfileException customerSet(): added facebookProfile, touched implementation, added Duplica-teFacebookProfileException complex type customerInfo: added facebookProfile
11.3.2016	14c	Jiří Brachtl	Removed facebookProfile from every where. Removed DuplicateFacebookAccountException.
8.4.2016	15	Michal Keller	CpOrderGet(): added phone2,phone3 CpOrderCreate(): added phone2,phone3 + collectiveBilling change M/O
12.4.2016	15	Tomáš Menclík	Added methods customerContractGet() ,customerContarctCreate()
14.4.2016	15	Jiří Brachtl	customerSearch(): added tariffId
14.4.2016	15	Tomáš Menclík	Added method customerContractSet(), removed methods cpContractSet() and cpContractSetSigned()
18.4.2016	15	Jiří Brachtl	customerCreatePostpaid(): tariffId became optional gsAdd(): added name, consumerId Added consumer* methods Added component* methods Added serviceOrder* methods
19.4.2016	15	Jiří Brachtl	cpTariffList(): refactored to return component categories as well
20.4.2016	16	Jiří Brachtl	partnerList(): added limitation for Reseller

			cpOrderList(): touched implementation cpOrderCreate(): choose PlatformOwner by Reseller
21.4.2016	16	Jiří Brachtl	cpOrderCreate(): removed collectiveBilling cpOrderGet(), cpOrderHistory(): removed contract, removed collectiveBilling configuration(): removed collectiveBilling cpOrderChangeStatus(): removed collectiveBilling, disabled for Reseller cpOrderSet(): removed collectiveBilling, disabled for Reseller
27.4.2016	16	Jiří Brachtl	consumerSearch(): added output wrapper „consumers“
29.4.2016	16	Jiří Brachtl	invoiceList(), invoiceGet(), customerInvoiceList(), customerInvoiceGet(): added paymentStatus Added invoicePaymentStatusSet()
3.5.2016	16	Jiří Brachtl	customerInfo: changed components structure, GID: added name, consumer, re- seller serviceOrderList(): added tariffId, consumer renamed to owner consumerSet(): all inputs became optional
5.5.2016	16	Jiří Brachtl	serviceOrderList(): changed structure
10.5.2016	16	Lukáš Veselka	customerInfo: add optional autoPaymentAuthorizedValue customerSet: add optional requiredAuthorizationValue
10.5.2016	16	Jiří Brachtl	serviceOrderList(): added output: consumer, serviceName
11.5.2016	16	Jiří Brachtl	cpComponentLocationList(): touched implementation Added cpComponentPerRequestList()
13.5.2016	16	Jiří Brachtl	serviceOrderList(): more statuses in input Added cpOrderInstallation* methods
16.5.2016	16	Tomáš Menclík	addSimToBasket(), getSimBasket(), removeSimFromBasket(), setSimBasket(), getOrderSimBaskets(), added note parameter in cpOrderComponentSet()
16.5.2016	16	Jiří Brachtl	cpOrderCreate(): auto creating GID with overhead costs cpOrderGet(): added overheadGid
23.05.2016	16	Tomáš Menclík	Changes in methods setSimBasket() and getSimBasket()
23.5.2016	16	Jiří Brachtl	Added cpOrderTaskList()
27.5.2016	16	Jiří Brachtl	simAssign(): gid became mandatory again, removed cpOrderComponentId
2.6.2016	16	Jiří Brachtl	Added cpRetailWholesaleTariffs() cpComponentLocationList(): added input gps, added limit of 100
7.6.2016	17	Jiří Brachtl	Changes related to „one component per GID“: cpOrderList(): changed output cpOrderGet(): changed output cpOrderChangeStatus(): changed implementation cpOrderSet(): changed input cpOrderComponentCreate(): changed allowed statuses cpOrderComponentSet(): allowed in any status, changed input cpOrderComponentHistory(): changed output cpOrderCustomItemCreate(): new cpOrderCustomItemSet(): new cpOrderCustomItemRemove(): new cpBundleTariffList(): new cpComponentSubcategoriesList(): new cpComponentLocationList(): changed output cpComponentPerRequestList(): removed

			cpTariffList(): removed cpTariffRetailWholesale(): changed input and implementation
15.6.2016	17	Jiří Brachtl	cpOrderGet(): added output customItems, bundleTariff/pricelistId
17.6.2016	18	Jiří Brachtl	customerInfo: changed structure of components
19.7.2016	19	Karel Kostka	cpSendCustomerNotification: added
10.8.2016	20	Jiří Brachtl	Removed unused RT2* email templates.
11.8.2016	20	Jiří Brachtl	Added cpCpeLoginAvailable()
17.8.2016	20	Jiří Brachtl	Removed deprecated Cplptv* methods: all except cplptvGetAvailablePackages.
20.1.2017	21	Jiří Brachtl	customerSet(): added postBilling, billingEmail customerInfo: added postBilling, billingEmail
23.1.2017	21	Lukáš Veselka	cpOrderCreate(): added postBilling, billingEmail cpOrderGet(): added postBilling, billingEmail
24.1.2017	22	Michal Keller	CpOrderGet(): added paramater ,paymentCardAuthorization'
27.1.2017	22	Lukáš Veselka	rename cpWarranty* → cpOrderInvoice* add customerNoteSet(), and notes to customerInfo
2.2.2017	22	Lukáš Veselka	Add cpComponentPortSim() cpOrderGet(): add 'portSimRequested' in component Add birthYear to person and agreeWithSendDocumentsByEmail to customer Add paymentHistoryInfo()
10.2.2017	22	Lukáš Veselka	cpOrdetTaskList(): add customer and provider
13.2.2017	23	Jiří Brachtl	cpComponentLocationList(): radius replaced by zoom
20.2.2017	23	Lukáš Veselka	New method cpComponentVerifyAvailability()
24.2.2017	23	Lukáš Veselka	cpComponentVerifyAvailability(): componentId become optional, add city, street, streetNumber, add return parameter „success“
8.3.2017	24	Jiří Brachtl	cpOrderCreate(). added input customerId, type cpOrderComponentCreate(): added input action cpOrderGet(): added output „advance“
24.3.2017	24	Jiří Brachtl	CustomerGet(): added defaultTariffId CustomerSet(): added defaultTariffId
24.3.2017	24	Tomáš Menclík	Adding partner parameters into methods partnerGet() and partnerList(). Adding input parameters (shopAssistant, resellerId) into method CpOrderCreate(). Adding output parameter (shopAssistant) into method CpOrderGet(). Adding output parameter (shopAssistant, resellerId) into method CpOrderSet().
2.5.2017	24	Lukáš Veselka	customerInfo: add bundleTariff
29.5.2017	24	Tomáš Menclík	Adding parameter warrantyInvitation into cpOrderGet response. Adding new enum into CpOrderInvoiceDownloadInvoice and CpOrderInvoiceSet.
08.6.2017	24	Tomáš Menclík	Adding parameter cpeType into response on method cpOrderComponentSet
15.6.2017	24	Tomáš Menclík	Adding parameter additionalItems into methods cpOrderGet, cpOrderComponentSet, cpTariffRetailWholesale
22.6.2017	24	Tomáš Menclík	Adding parameter invoiceId into method cpOrderInvoiceDownloadInvoice(). Adding parameter onetimeFees into cpOrderGet() response. Adding new method cpOrderOnetimePriceGet(). Adding new method cpOrderPayOnetimeItems()

29.6.2017	24	Tomáš Menclík	Adding new method cpCalculatePrice(). Adding parameter customerId into cpOrderList(). Adding parameter active into data type customerInfo into section components. Adding object reseller into cpOrderList() request instead of resellerId
1.8.2017	24	Lukáš Veselka	Add force parameter to simAssign method.
8.8.2017	24	Jiří Brachtl	getSimBasket(): removed input orderId, renamed input componentId to order-ComponentId
25. 9. 2017	24	Lukáš Veselka	Add „active“ parameter to customerSearch.
2.10.2017	24	Jiří Brachtl	Added email template RT2
2.10.2017	24	Tomáš Menclík	Adding new methods paperInvoiceList(), exportPaperInvoiceList(), paperInvoice-Set()
18.10.2017	25	Jiří Brachtl	cpOrderSet(): added parameters
6.11.2017	25	Tomáš Menclík	Making parameter primaryAddress in method cpOrderGet() optional.
30.11.2017	25	Tomáš Menclík	complex type customerInfo: added section project into customerInfo.customer-Data. Adding section cloudConnection into customerInfo.gs.components.component.
6.12.2017	25	Tomáš Menclík	Adding new method gsSpendingLimitChange(). Making parameter phone in method cpOrderGet() optional. Adding optional parameter period into gpcsAccount in customerInfo.customer-Data.gpcsAccount. Adding section gpcsAccounts into customerInfo.gs .
27.2.2018	25	Tomáš Menclík	Adding new method cpOrderCopy(). Making parameter email in method cpOrderGet() optional. Adding parameter project under customer in methods cpOrderGet() and cpOrderList().
4.6.2018	25	Tomáš Menclík	Adding new input parameters (orderBy, sorted, statuses, componentCategoryId) into method cpOrderList(). Adding new output parameters (requestedBy, componentCategoryName) into method cpOrderList(). Adding new method cpFeasibilityCheck().
7.6.2018	25	Tomáš Menclík	Adding ruian input parameter into method cpOrderCreate().
8.6.2018	25	Tomáš Menclík	Changing structure of output in method cpTariffRetailWholesale().
9.6.2018	25	Lukáš Veselka	Add new method cpRuianFind().
25. 6. 2018	25	Lukáš Veselka	Add new method providerTaskChangeStatus().
21. 8. 2018	25	Lukáš Veselka	Add new method findSim()
10. 9. 2018	25	Lukáš Veselka	Add new method providerTaskCreate()
24.10.2019	25	MŠP	Company logo update