SI3000

|  |  |
| --- | --- |
|  |  |
| Service Activator API |  |
|  |  |

SP1002

|  |  |
| --- | --- |
| Tip dokumenta | Service Activator API specification |
| Št. zapisa | IDOCS-3-248 |
| Narejeno za | SP1002AB |
| Datum | 23.12.2016 |
| Avtor | Rajh Simon, Bojan Pišek |
| Stopnja zaupnosti | Public |

Content

[1. Overview 4](#_Toc473185600)

[2. Provisioning API 4](#_Toc473185601)

[2.1. Security 4](#_Toc473185602)

[2.2. HTTP response 4](#_Toc473185603)

[3. B2B provisioning 6](#_Toc473185604)

[3.1. Business Group (BG) management 6](#_Toc473185605)

[3.1.1. Create BG 6](#_Toc473185606)

[3.1.2. Update BG 6](#_Toc473185607)

[3.1.3. Get BG 7](#_Toc473185608)

[3.1.4. Delete BG 8](#_Toc473185609)

[3.1.1. Get list of BGs 8](#_Toc473185610)

[3.2. Enterprise Admin (EA) management 10](#_Toc473185611)

[3.2.1. Create EA 10](#_Toc473185612)

[3.2.2. Get EA 11](#_Toc473185613)

[3.3. Management of DN in BG 12](#_Toc473185614)

[3.3.1. Create and/or update DN 12](#_Toc473185615)

[3.3.2. Get DN 14](#_Toc473185616)

[3.3.3. Delete DN 15](#_Toc473185617)

[3.3.4. Move DN 16](#_Toc473185618)

[3.4. Merge/Unmerge BG 17](#_Toc473185619)

[4. B2C Provisioning 17](#_Toc473185620)

[4.1. Create and/or update DN 17](#_Toc473185621)

[4.2. Get DN 19](#_Toc473185622)

[4.3. Delete DN 19](#_Toc473185623)

[4.4. Get list of DN 20](#_Toc473185624)

[4.5. Move DN to BG 20](#_Toc473185625)

[5. Agency provisioning 22](#_Toc473185626)

[5.1. Agency management 22](#_Toc473185627)

[5.1.1. Create agency 22](#_Toc473185628)

[5.1.2. Update agency 23](#_Toc473185629)

[5.1.3. Delete agency 23](#_Toc473185630)

[5.1.4. Get agency 24](#_Toc473185631)

[5.1.5. Get all agencies 24](#_Toc473185632)

[5.2. Agency admin management 25](#_Toc473185633)

[5.2.1. Create or update agency admin 25](#_Toc473185634)

[5.2.2. Get agency admin 27](#_Toc473185635)

[5.2.3. Delete Agency Admin 28](#_Toc473185636)

[5.3. Agency DNS management 29](#_Toc473185637)

[5.3.1. Assign dns to agency 29](#_Toc473185638)

[5.3.2. Add existing dns to agency 30](#_Toc473185639)

[5.3.3. Remove existing dns from agency 30](#_Toc473185640)

[5.3.4. Get all dns in agency 31](#_Toc473185641)

[6. Suspend/Resume of DN or BG 33](#_Toc473185642)

[6.1. Suspend/Resume DN 33](#_Toc473185643)

[6.1.1. Suspension status 34](#_Toc473185644)

[6.2. Suspend/Resume BG 35](#_Toc473185645)

[7. Supplementary services 36](#_Toc473185646)

[7.1. Get phone devices 36](#_Toc473185647)

[7.2. CFU Service 37](#_Toc473185648)

[7.2.1. Get service status 37](#_Toc473185649)

[7.2.2. Modify service 37](#_Toc473185650)

[7.3. CFB Service 38](#_Toc473185651)

[7.3.1. Get service status 38](#_Toc473185652)

[7.3.2. Modify service 38](#_Toc473185653)

[7.4. DND Service 39](#_Toc473185654)

[7.4.1. Get service status 39](#_Toc473185655)

[7.4.2. Modify service 39](#_Toc473185656)

[7.5. DNDGM Service 41](#_Toc473185657)

[7.5.1. Get service status 41](#_Toc473185658)

[7.5.2. Modify service 42](#_Toc473185659)

[8. User Api 43](#_Toc473185660)

[8.1. Get users Request 43](#_Toc473185661)

[8.2. Assign user to resident or agency DN 44](#_Toc473185662)

[8.3. Delete user 45](#_Toc473185663)

[9. Address book Api 46](#_Toc473185664)

[9.1. Contacts 46](#_Toc473185665)

[9.1.1. Get contacts authorized for username 46](#_Toc473185666)

[9.1.2. Search contacts authorized for user 47](#_Toc473185667)

[9.1.3. Get contact avatar 48](#_Toc473185668)

[9.1.4. Get contacts by business group id 48](#_Toc473185669)

[9.1.5. Get contacts in business group by given phone number 49](#_Toc473185670)

[9.1.6. Search contacts in BGID without private and VIP 50](#_Toc473185671)

[9.1.7. Count contacts in business group 51](#_Toc473185672)

[9.2. Objects 52](#_Toc473185673)

[9.2.1. User 52](#_Toc473185674)

[9.2.2. Phone 52](#_Toc473185675)

[9.2.3. Contact 52](#_Toc473185676)

[9.2.4. Contact data (contactData) 53](#_Toc473185677)

# Overview

The document contains description of API implemented in SA application. Provided resources allow managing business grupB2B and B2C users and their services on UC platform.

# Provisioning API

Provisioning API use following HTTP methods:

* POST: create a new resource
* PUT: update a specific resource
* GET: read a specific resource
* DELETE: delete/remove a specific resource

All messages in request/response body are in JSON format and UTF-8 encoded. The API requires ***Content-Type*** header be set to ***application/json***. If other type is set a 415 Unsupported Media Type HTTP status code is returned.

RPC methods

* GET ?param =: method without side effect
* PUT ?action= method with side effect - ?action=suspend

Change Nr.: 01

## Security

Operations are protected using **basic HTTP authentication** method (RFC2617). A client must send authorization header with username and password in each HTTP request to gain access to protected resource.

There are two users on SA which have permission to access resources:

* admin : administrator of SA application
* onlineUser : dedicated user for REST provisioning API

The server authenticates a given username and password with the stored credential. If authentication is successful then process continues with execution of business logic.

## HTTP request parameter handling

All none-mandatory parameters that will not be set in requests HTTP body, will have their default values. **This is important for all update uperations, because all old values will be overwritten with default values if they are not presented in requests HTTP body.**

Parameters type default values:

|  |  |
| --- | --- |
| **Parameter type** | **Default value** |
| *String* | null |
| *Boolean* | null |
| *boolean* | false |
| *Integer* | null |
| *int* | 0 |

## HTTP response

SA uses standard HTTP responses. Most frequently used responses by the SA are described in table below.

|  |  |
| --- | --- |
| **HTTP error response** | **Description** |
| *2xx OK* | Successful response. |
| *400 Bad Request* | Bad Request (in case provided parameters could not be processed - the request couldn’t be understood, usually because the JSON body contains an error) |
| *401 Not Authorized* | Wrong authentication either username or password is not correct |
| *403 Forbidden* | User does not have necessary permissions |
| *404 Not Found* | Unknown resource. |
| *405* *Method not allowed* | Used to indicate that the requested URI exists, but the requested HTTP method is not applicable |
| *409 Conflict* | Request could not be executed due to an conflict with current resource state |
| *415 Unsupported Media Type* | In case of unsupported media type |
| *500 Internal server error* | General server error |

The server responds with 200 OK in case of successful request processing. If request is processed unsuccessfully the 4xx response is returned.

The response contains:

* Error code and message
* Resource ID in case of its creation
* Resource state representation in case of its retrieving

In other cases response is empty.

*{*

*“code” : <error code>,*

*“msg” : ”<message>”*

}

# B2B provisioning

API for business to business provisioning: to manage business groups and telephone numbers. API can be used by service provider administrator users: “admin” and “onlineUser”.

## Business Group (BG) management

API to create, update, read and delete a Business Group (BG) on UC system.

### Create BG

It creates new business group (BG). It returns 200 OK upon success with parameters in response.

**Request**

Request to create BG:

**POST http://<hostname>/sa/rest/v1/prov/bgs**

Parameters in request body

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Required** | **Type** | **Explanation** |
| *name* | *yes* | *String* | Name of a company (BG) |
| *ctxLength* | *yes* | *Integer* | Number of digits used for internal numbers. Allowed values: 3, 4, 5. |
| *extId* | *no* | *String* | External ID of company used by other system (e.g. billing system) or null. Parameter can only be set at BG creation. |
| softphoneLicMobile | *no* | Integer | Number of available mobile licenses per BG. In case license per BG are not limited value must be “null”. |
| softphoneLicDesktop | *no* | Integer | Number of available desktop licenses per BG. In case license is not limited value must be “null” |

Request example:

POST http://10.1.2.3/sa/rest/v1/prov/bgs

HTTP body

{

"name" : "aRest1",

"ctxLength" : 4,

"extId" : "EXT\_ID\_aRest1",

"softphoneLicMobile" : null,

"softphoneLicDesktop" : 5

}

**Response**

For description refer to response description of chapter 3.1.3.

### Update BG

It updates BG with values set in request body. In case of success, 200 Ok is returned with parameters in body.

**Request**

Resource URI:

**PUT http://<hostname>/sa/rest/v1/prov/bgs/<bgid>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *bgid* | *yes* | *Integer* | ID of a company (BG) |

For description of parameters in request body refer to the chapter 3.1.1 .

Request Example

PUT http://prov.server.net/sa/rest/v1/prov/bgs/12

Body:

{

“bgId” : 12,

“name” : “example company”,

“ctxLenght” : 4,

“extId” : “987”,

“*softphoneLicMobile” : 0,*

*“softphoneLicDesktop” : 0*

}

**Response**

For description of response refer to chapter 3.1.3.

### Get BG

It returns data for specified business group.

**Request**

Resource URI:

**GET http://<hostname>/sa/rest/v1/prov/bgs/<bgid>**

Request Example

GET http://prov.server.net/sa/rest/v1/prov/bgs/12

**Response**

Parameters in response body

|  |  |  |
| --- | --- | --- |
| ***Parameter Name*** | ***Type*** | ***Explanation*** |
| *bgid* | *Integer* | ID of a BG in the IBC system. |
| *name* | *String* | Name of a company (BG) |
| *ctxLength* | *Integer* | Number of digits used for internal numbers. Allowed values: 3, 4, 5. |
| *extId* | *String* | External ID of company used by other system. |
| *softphoneLicMobile* | Integer | Number of available mobile licenses per BG. |
| *softphoneLicDesktop* | Integer | Number of available desktop licenses per BG. |

Response body example

200 OK

{

“bgId” : 12,

“name” : “example company”,

“ctxLenght” : 4,

“extId” : “987”,

“*softphoneLicMobile” : 0,*

*“softphoneLicDesktop” : 0*

}

.

### Delete BG

It deletes BG with given id.

**Request**

Resource URI:

**DELETE http://<hostname>/sa/rest/v1/prov/bgs/<bgid>**

For description of bgid parameter refer to description of parameters in request URI in chapter 3.1.1.

Request Example

DELETE http://prov.server.net/sa/rest/v1/prov/bgs/12

**Response**

In case of successful delete of a BG 200 OK is returned.

Response example:

{

"code": 200,

"msg": "Business group with id: 47 successfully deleted!"

}

### Get list of BGs

Operation returns list of BGs that already exist in system.

**Request**

Resource URI:

**GET http://<hostname>/sa/rest/v1/prov/bgs**

Request Example:

GET http://prov.server.net/sa/rest/v1/prov/bgs

**Response**

For description of response parameters refer to response description in chapter 3.1.1.

Response example:

200 OK

Body:

{

"bgs": [

{

“id” : 12,

“name” : “CompanyA”,

“centrexNumberLength” : 3,

“externalID” : ”987”

*“softphoneLicMobile” : null,*

*“softphoneLicDesktop” : null*

},

{

“id” : 42,

“name” : “CompanyF”,

“centrexNumberLength” : 5,

“externalID” : null

*“softphoneLicMobile” : null,*

*“softphoneLicDesktop” : null*

},

...

]

}

## Enterprise Admin (EA) management

Api to manage enterprise administrator (EA) for certain BG. Only one admin may exist per BG.

Resource URI:

**http://<hostname>/sa/rest/v1/prov/bgs/<bgid>/ea**

### Create EA

It creates EA for provided BG. Upon success 200 OK is returned with parameters in response body.

**Request**

Resource URI:

**POST http://<hostname>/sa/rest/v1/prov/bgs/<bgid>/ea**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *bgid* | *yes* | *String* | ID of a company (BG) |

Parameters in request body:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *username* | yes | String | Username of the EA, which will be used as for login in to portal. Username must be unique on IBC system. Username can be either email, telephone number or any other unique value. |
| *email* | *yes* | *String* | EA email |
| *dn\** | *yes* | *String* | Telephone number, which must be in international format (country code – national destination code + number e.g. 38642071221). |
| *nodeId* | *yes* | *Integer* | CS node on which DN will be created. Parameter is only valid for creation operation. |

\* Dn must already exist in provided BG and be free of user. For dn creation refer to the chapter 3.3.1

Request example:

POST http://prov.server.net/sa/rest/v1/prov/bgs/12/ea

HTTP body:

{

"username" : "adminUser",

"email" : "[ea@companya.com](mailto:ea@companya.com)",

"dn" : "38044100100",

"node": 5000

}

**Response**

For description of response refer to the chapter 3.2.2.

### Get EA

It retrieves EA data for provided BG.

**Request**

Request to retrieve EA:

**GET http://<hostname>/sa/rest/v1/prov/bgs/<bgid>/ea**

For description of <bgid> parameter refer to description of parameters in request URI in chapter 3.2.1.

Request example:

GET http://prov.server.net/sa/rest/v1/prov/bgs/12/ea

**Response**

Parameters in response body

|  |  |  |
| --- | --- | --- |
| ***Parameter Name*** | ***Type*** | ***Explanation*** |
| *bgid* | Integer | ID of the BG, for which admin will be created |
| *username* | String | Username of the EA. |
| *email* | *String* | EA email |
| *dn* | *String* | Telephone number in international format (country code – national destination code + number e.g. 38642071221). |

Response example:

200 OK

Body:

{

"bgid": 12,

"username" : "adminUser",

"email" : "[ea@companya.com](mailto:ea@companya.com)",

"dn" : "38044100100"

}

## Management of DN in BG

API to manage DNs (telephone numbers) for certain BG. Telephone number must be in international format e.g. 38642071221 (CC - NDC - DN = 386-4-2071221)

Resource URI:

**http://<hostname>/sa/rest/v1/prov/dns/<dn>**

### Create and/or update DN

The HTTP method PUT creates DN which is assigned to the BG id or updates it otherwise. If DN is already in some other BG then request is rejected (http 409 Conflict : DN is already in BG). In case of new DN, it is created on the specified node and moved to the BG.

Response returns JSON data with parameters.

**Request**

Request to create or update DN

**PUT http://<hostname>/sa/rest/v1/prov/dns/<dn>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *yes* | *String* | Telephone number, which must be in international format (country code – national destination code + number e.g. 38642071221). The dn could be pre-created on the node and in this case it is assigned in to given BG. If dn is new then it is created and moved to the BG. |

Parameters in request body:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *bgid* | *yes* | *Integer* | ID of a BG. |
| *Node\** | *yes* | *Integer* | CS node on which DN will be created. Parameter is only valid for creation operation. |
| *Type\** | *no* | *Integer* | Additional information for dn. Possible values:   1. - prepaid (valid only for mobile DN) 2. - postpaid (default value)   Parameter can be only set at DN creation. |
| *mdu* | *no* | *Boolean* | Authorization for universal numbering service. . Service is by default disabled (parameter value is “false” or null). |
| *shortNr* | *no* | *Boolean* | Authorization for short numbering service. . Service is by default disabled (false). |
| *advanced* | *no* | *Boolean* | Authorization for black and white lists service. Default is false. |
| *ivr* | *no* | *Boolean* | Authorization for IVR / Auto Attendant service. Service is by default disabled (false). |
| *internalPbxDns* | *no* | *Integer* | Authorization for service nonpublic PBX dns. It creates as many nonpublic dns as given in parameter value. All created numbers are based on provided dn. |
| *voiceMail* | *no* | *Boolean* | Authorization for voice mail service. Default is false. . Service is by default disabled (false). |
| *rcs* | *no* | *Boolean* | Authorization for RCS (Reach Client Communication) service. . Service is by default disabled (false). |
| *mduAnywhere* | *no* | *Boolean* | Authorization for MDU anywhere service. . Service is by default disabled (false). |
| *internalBgDns* | *no* | *Integer* | Authorization for SIP gray (nonpublic) DNs service. It creates as many gray DNs numbers as given in parameter value. All DNs are based on given dn. Service is not authorized if value is “null” or “0”. |
| *wac* | *no* | *Integer* | Authorization for Web conference service. It specifies the number of DN for WAC service. Service is not authorized if value is “null” or “0”. |
| *nc* | *no* | *Integer* | Authorization for Notification call service. It specifies the number of DN for NC service. Service is not authorized if value is “null” or “0”. |
| *ddi* | *no* | *Integer* | Authorization for ddi service. It specifies the number of DN for DDI service. Service is not authorized if value is “null” or “0”. |
| *freePhone\*\** | *no* | *Boolean* | Authorization for Free phone service. |
| *drs* | *no* | *Boolean* | Authorization for DRS service. Precondition for this service is authorized WAC service. |

\* Node and type parameters are only valid for DN creation

\*\* To create free phone number, you have to use special national destination code (NDC) with corresponded node, provided in application config (e.g. 380**800**300300). Free Phone service is only compatible with “shortNr” service. If any other service is activated, request will be rejected.

Request example:

PUT http://prov.server.net/sa/rest/v1/prov/dns/38642071221

HTTP body:

{

"bgid": 46,

"dn": "38044500701",

"node": 5000,

"type": 2,

"mdu": false,

"shortNr": false,

"advanced": true,

"ivr": false,

"internalPbxDns": 0,

"voiceMail": false,

"rcs": false,

"mduAnywhere": false,

"internalBgDns": 2,

"wac": 8,

"nc": 5,

"ddi": null,

"cliro": null,

"freePhone": null,

"drs": true,

"imsi": null,

"videoCodec": false,

"tsm": false,

"callContinuity": false,

"ipGeoLocation": false

}

**Response**

For response description refer to chapter 3.3.2

### Get DN

The method returns DN data.

**Request**

Request to retrieve DN data

**GET http://<hostname>/sa/rest/v1/prov/dns/<dn>**

For information about parameters in URI refer to chapter 3.3.1 – description of parameters in request URI.

Request example:

GET http://prov.server.net/sa/rest/v1/prov/dns/380447700002

**Response**

For description of parameters refer to chapter 3.3.1 .

Response body example

HTTP 200 OK

HTTP body:

{

"bgid": 4,

"dn": "380447700002",

"node": 5000,

"type": 2,

"mdu": true,

"shortNr": true,

"advanced": true,

"ivr": false,

"internalPbxDns": 0,

"voiceMail": false,

"rcs": false,

"mduAnywhere": false,

"internalBgDns": 1,

"wac": 0,

"nc": 0,

"ddi": 0,

"cliro": null,

"freePhone": null,

"drs": false,

"imsi": null,

"videoCodec": false,

"tsm": false,

"callContinuity": false,

"ipGeoLocation": false

}

### Delete DN

It deletes DN from UC system. Same method is used to delete B2B DN or B2C DN (chapter 4.3)

**Request**

Request to delete DN

**DELETE http://<hostname>/sa/rest/v1/prov/dns/<dn>**

For information about parameters in URI refer to chapter 3.3.1 – description of parameters in request URI.

Request example:

DELETE http://prov.server.net/sa/rest/v1/prov/dns/38642071221

**Response**

In case of successful delete of a BG 200 OK is returned.

Response body example:

{

"code": 200,

"msg": "dn: '38044500701' successfully deleted!"

}

### Move DN

HTTP PUT with query parameter “action=movetobg” is used to move DN. DN can be moved to another BG or to B2C DNs. For successful move bgid of targeted business group must be provided.

**Request**

Request URI to move DN to another BG:

**PUT http://<hostname>/sa/rest/v1/prov/dns/<dn>?action=movetobg&bgid=5**

Request URI to move DN from BG to B2C DNs:

**PUT http://<hostname>/sa/rest/v1/prov/dns/<dn>?action=movetobg&bgid=1**

Note: functionality only works if DN was resident before

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *yes* | *String* | Telephone number, which must be in international format (country code – national destination code + number e.g. 38642071221). |
| *bgid* | *yes* | *Integer* | ID of the targeted BG |
| *action* | *yes* | *String* | For successful move action parameter must be equal to “movetobg”. |

**Response**

In case of successful move of DN 200 OK is returned.

Response example:

{

"code": 200,

"msg": "DN '38044500702' moved from bg: 46 to bg: 4."

}

## Merge/Unmerge BG

It is used to merge/unmerge Business groups in to master business group. In case of successful merge/unmerge 200 OK is returned.

**Request for merge BGs**

Request URI:

**PUT http://<hostname>/sa/rest/v1/prov/bgs/<bgid>?action=mergetobg&bgid=<masterbgid>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *bgid* | *Yes* | *Integer* | Slave Business Group ID |
| *masterbgid* | *Yes* | *Integer* | Master Business Group ID |
| *action* | *Yes* | *String* | For successful merge action parameter must be equal to “mergetobg” |

Example:

Merge BG:

PUT http://prov.server.net/sa/rest/v1/prov/bgs/10/?action=mergetobg&bgid=12

**Request for unmerge BG**

If BG was merged in to another BG (master BG) it can be unmerged with action “action=unmergebg” as query parameter in URL.

Request URI for unmerge BG from master BG:

**PUT http://<hostname>/sa/rest/v1/prov/bgs/<bgid>?action=unmergebg**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *bgid* | *Yes* | *Integer* | Slave Business Group ID |
| *action* | *Yes* | *String* | For successful unmerge action parameter must be equal to “unmergebg” |

Request example for unmerge BG:

PUT http://<hostname>/sa/rest/v1/prov/bgs/10?action=unmergebg

# B2C Provisioning

API to manage DNs for B2C (residents). Telephone number (DN) must be in international format e.g. 38642074334 (CC - NDC - DN = 386-4-2074334)

Resource URI:

**http://<hostname>/sa/rest/v1/prov/dns/<dn>**

## Create and/or update DN

The HTTP PUT method creates DN for B2C (residents) if DN does not exists and updates it otherwise. The DN could be new on the system or it may already exist on certain CS node. If DN exists on the CS node it is moved to residents. If DN is already in some other BG then request is rejected (http 409 Conflict: DN is already in BG). In case of new DN, it is created on the specified node as DN for residents. Request body contains authorization for IBC services.

**Request**

Request to create DN

**PUT http://<hostname>/sa/rest/v1/prov/dns/<dn>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *yes* | *String* | Telephone number, which must be in international format (country code – national destination code + number e.g. 38642074334). The dn could be pre-created on the node CS node or not. |

Parameters in request body:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *bgid* | *no* | *Integer* | Business group id. If parameter is set it must have value 1 (to run proper B2C provisioning). |
| *node* | *yes* | *Integer* | CS node on which DN will be created. |
| *type* | *no* | *Integer* | Additional information for dn. Possible values:   1. - prepaid (valid only for mobile DN) 2. - postpaid (default value)   Parameter is only valid at DN creation. |
| *mdu* | *no* | *Boolean* | Authorization for universal numbering service |
| *advanced* | *no* | *Boolean* | Authorization for black and white lists service |
| *cliro* | *no* | *Boolean* | Authorization for CLIRO service |
| *tsm* | *no* | *Boolean* | Tunneling service |  |
| *videoCodec* | *no* | *Boolean* | Video codecs service |
| *callContinuity* | *no* | *Boolean* | Call continuity service |
| *ipGeoLocation* | *no* | *Boolean* | IP geolocaton service |
| *imsi* | *no* | *String* | Imsi parameter is only relevant for mobile DN, otherwise it is ignored. |

Request example:

PUT http://prov.server.net/sa/rest/v1/prov/dns/38642074334

HTTP body:

{

"node" : 5002,

"type" : 2,

"mdu" : true,

"advanced" : true,

"cliro" : false,

"tsm" : true;

"videoCodec" : true,

"callContinuity" : false,

"ipGeoLocation" : false

}

**Response**

Response body contains data for a DN in JSON format. For description of parameters refer to 4.1 – request section.

Response body example

Same as B2B DN response in chapter 3.3.2 – response section.

## Get DN

HTTP GET is used to retrieve information for a DN. The method returns DN data for provided residential DN.

**Request**

Request to retrieve DN information

**GET http://<hostname>/sa/rest/v1/prov/dns/<dn>**

For information about parameters in URI refer to chapter 4.1 – parameters in request URI.

Request example:

GET http://prov.server.net/sa/rest/v1/prov/dns/38642074334

**Response**

Response body contains data for a DN in JSON format. For description of parameters refer to 4.1 – request section.

Response body example

Same as B2B DN response in chapter 3.3.2 – response section.

## Delete DN

HTTP DELETE is used to delete residential DN. Same method is used to delete B2C DN or B2B DN (chapter 3.3.3)

**Request**

Request to delete DN

**DELETE http://<hostname>/sa/rest/v1/prov/dns/<dn>**

For information about parameters in URI refer to chapter 4.1 – parameters in request URI.

Request example:

DELETE http://prov.server.net/sa/rest/v1/prov/dns/38642074334

**Response**

In case of successful delete of a BG 200 OK is returned.

Response body example:

{

"code": 200,

"msg": "Resident dn: ‘38642074334’ successfully deleted!

}

## Get list of DN

Operation returns list of Residents DNs that already exist in system.

**Request**

Request to get list of DN

**GET http://<hostname>/sa/rest/v1/prov/dns**

Request example:

GET <http://prov.server.net/sa/rest/v1/prov/dns>

**Response**

Response example:

200 OK

Body:

{

"dns": [

"38642071221",

"38642071223",

"38642071432",

...

]

}

## Move DN to BG

HTTP PUT with argument “action” is used to move residential DN to a BG.

**Request**

Request to move DN to BG

**PUT http://<hostname>/sa/rest/v1/prov/dns/<dn>?action=movetobg&bgid=<bgid>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *yes* | *String* | Telephone number, which must be in international format (country code – national destination code + number e.g. 38642071221). |
| *bgid* | *yes* | *Integer* | ID of a targeted BG. |

Request example:

PUT http://prov.server.net/sa/rest/v1/prov/dns/38642074334/action=movetobg&bgid=12

**Response**

In case of successful move of DN 200 OK is returned.

Response example:

{

"code": 200,

"msg": "Move resident dn: '38011100210' to bg:46."

}

# Agency provisioning

API for agency provisioning: to manage agencies and telephone numbers. API can be used by service provider administrator users: “admin” and “onlineUser”.

## Agency management

API to create, update, read and delete an agency on UC system.

### Create agency

It creates new agency. It returns 200 OK upon success with parameters in response.

**Request**

Request to create new agency:

**POST http://<hostname>/sa/rest/v1/prov/agencies**

Parameters in request body

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *name* | *yes* | *String* | Name of an agency |
| *description* | *no* | *String* | Description of an agency |
| *url* | *no* | *String* | Url of an agency |

Request example:

POST http://prov.server.net/sa/rest/v1/prov/agencies

HTTP body

{

"name": "test5",

"description": "someDescription",

"url": "someAdrress"

}

**Response**

In case of successful request 200 OK is returned.

Parameters in response body:

|  |  |  |
| --- | --- | --- |
| ***Parameter Name*** | ***Type*** | ***Explanation*** |
| *id* | *Integer* | ID of an agency |
| *name* | *String* | Name of an agency |
| *description* | *String* | Description of an agency |
| *url* | *String* | Url of an agency |

Response body example:

200 OK

HTTP body

{

"id": 65,

"name": "test5",

"description": "someDescription",

"url": "someAdrress"

}

### Update agency

It updates agency with values set in request body. In case of success, 200 Ok is returned with parameters in body.

**Request**

Resource URI:

**PUT http://<hostname>/sa/rest/v1/prov/agencies/<id>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *id* | *yes* | *Integer* | ID of an agency |

Parameters in request body:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *id* | *no* | *Integer* | ID of an agency |
| *name* | *yes* | *String* | Name of an agency. It can not be changed at update. |
| *description* | *no* | *String\** | Description of an agency |
| *url* | *no* | *String\** | Url of an agency |

\*All String values must be set at update, otherwise old values will be overwritten with null value.

Request Example

PUT http://prov.server.net/sa/rest/v1/prov/agencies/12

HTTP body

{

"id": 12,

"name": "test5",

"description": "someDescription",

"url": "someAdrress"

}

### Delete agency

It deletes agency with given id.

**Request**

Resource URI:

**DELETE http://<hostname>/sa/rest/v1/prov/agencies/<id>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *id* | *yes* | *Integer* | ID of an agency |

Request Example

DELETE http://prov.server.net/sa/rest/v1/prov/agencies/12

**Response**

In case of successful delete of an agency 200 OK is returned.

Response example:

HTTP body

{

"code": 200,

"msg": "Agency with id '19' successfully deleted!"

}

### Get agency

Agencies id must be provided in URI. Operation returns parameters for provided id.

**Request**

Resource URI:

**GET http://<hostname>/sa/rest/v1/prov/agencies/<id>**

For description of id refer to description of parameters in request URI in chapter 5.1.2

Request Example:

GET http://prov.server.net/sa/rest/v1/prov/agencies/12

**Response**

In case of success 200 OK is returned.

Response example:

HTTP body:

{

"id": 65,

"name": "test5",

"description": "someDescription",

"url": "someAdrress"

}

### Get all agencies

Operation returns list of Agencies that already exist in the system.

**Request**

Resource URI:

**GET http://<hostname>/sa/rest/v1/prov/agencies**

Request Example:

GET <http://prov.server.net/sa/rest/v1/prov/agencies>

**Response**

In case of success 200 OK is returned.

Response example:

200 OK

HTTP body:

[

{

"id": 1,

"name": "First",

"description": "First agency",

"url": "https://www.first.com"

},

{

"id": 2,

"name": "DiRajhspora",

"description": "",

"url": "https://www.first.com"

},

…

]

## Agency admin management

Api to manage agencies administrator. Only one admin may exist per agency.

### Create or update agency admin

The HTTP PUT creates or updates admin for desired Agency. If agency id is not specified request is rejected. If admin is already assigned to agency, update is performed. Creation of admin requires email in the body of the request. Email is also used as admins username.

**Request**

Request to create or update agency admin.

Resource URI:

**PUT http://<hostname>/sa/rest/v1/prov/agencies/<id>/admin**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *id* | *yes* | *Integer* | ID of an agency |

Parameters in request body:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *firstName* | *no\** | *String* | *Frist name of agency admin* |
| *middleName* | *no\** | *String* | *Middle name of agency admin* |
| *lastName* | *no\** | *String* | *Last name of agency admin* |
| *email* | *yes* | *String* | *Agency admin email* |
| *dn* | *no\*\** | *String* | *Telephone number, which must be in international format (country code – national destination code + number e.g. 38642071221).* |
| *nodeId* | *no\*\** | *Integer* | *CS node on which DN will be created.* |
| *contactNumber* | *no* | *String* | Admins contact number |
| *address* | *no* | *String* | Admins address |
| *type* | *no* | *Integer* | Additional information for dn. Possible values:  1- prepaid (valid only for mobile DN)  2- postpaid (default value)  Parameter is only valid at DN creation. |
| *imsi* | *no* | *String* | Imsi number |

\* One of parameters (firstName, middleName, lastName) must have a value in request body.

\*\* If Dn is provided in request body, then also nodeId is mandatory.

**Note:** If agency admin has assigned dn but it is not presented in request body at update, then it will be unassigned and presented as free in current agency.

If you want to replace DN of agency admin, you have to run update with no DN in request body and then again with new (existing) DN. For successfull request DN must be a member of provided agency and free of user.

Request example:

PUT http://prov.server.net/sa/rest/v1/prov/agencies/12/admin

HTTP body:

{

"firstName": "novi",

"middleName": "novi",

"lastName": "novi",

"email": "testdn@iskratel.si",

"contactNumber": "334",

"address": null,

"dn": "38011997000",

"nodeId": 5001,

"type": 1,

"imsi": "1344"

}

**Response**

Parameters in response body

|  |  |  |
| --- | --- | --- |
| ***Parameter Name*** | ***Type*** | ***Explanation*** |
| *firstName* | *String* | Admins first name |
| *middleName* | *String* | Admins middle name |
| *lastName* | *String* | Admins last name |
| *Email* | *String* | Admins Email (must be unique on uc system) |
| *Dn* | *String* | Telephone number, which must be in international format (country code – national destination code + number e.g. 38642071221). |
| *nodeId* | *Integer* | CS node on which DN will be created. Parameter is only valid for creation operation. |
| *contactNumber* | *String* | Admins contact number |
| *address* | *String* | Admins address |
| *type* | *Integer* | Additional information for dn. Possible values:  1- prepaid (valid only for mobile DN)  2- postpaid (default value)  Parameter is only valid at DN creation. |
| *imsi* | *String* | Imsi number |

Response body example:

{

"firstName": "novi",

"middleName": "novi",

"lastName": "novi",

"email": "testdn@iskratel.si",

"contactNumber": "334",

"address": null,

"dn": "38011997000",

"nodeId": 5001,

"type": 1,

"imsi": "1344"

}

### Get agency admin

The http method GET returns admin data for provided agency. If Agency id is not specified request is rejected.

**Request**

Request to retrieve agency admin:

**GET http://<hostname>/sa/rest/v1/prov/agencies/<id>/admin**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *id* | *yes* | *Integer* | ID of an agency |

Request example:

GET http://<hostname>/sa/rest/v1/prov/agencies/12/admin

**Response**

Response body contains data for agency admin in JSON format. For description of parameters refer to 5.2.1 – response section.

Response body example

200 OK

{

"firstName": "Janez",

"middleName": "",

"lastName": "",

"email": "janez.je@as.si",

"contactNumber": "",

"address": "",

"dn": "38011100901",

"nodeId": 5001,

"type": 1,

"imsi": "234523423564564"

}

### Delete Agency Admin

HTTP DELETE is used to delete agencies admin.

**Request**

Request to delete agency admin

**DELETE http://<hostname>/sa/rest/v1/agencies/{id}/admin**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *id* | *yes* | *Integer* | ID of an agency |

Request example:

DELETE http://prov.server.net/sa/rest/v1/prov/agencies/8/admin

**Response**

In case of success 200 OK is returned.

Response example:

HTTP body:

{

"code": 200,

"msg": "Agency admin successfully deleted!"

}

## Agency DNS management

### Assign dns to agency

Operation assigns existing residential DNs to desired agency. For successful request all DNs in request body must be residential DNs. Each DN must be in international format (cc+ndc+number e.g. 38611100100).

**Request**

Request to assign list of DNs

**PUT http://<hostname>/sa/rest/v1/prov/agencies/{id}/dns**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *id* | *yes* | *Integer* | ID of an agency |

Parameters in request HTTP body:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dns* | *yes* | *String* | List of telephone numbers. All dns must be in international format (country code – national destination code + number e.g. 38642071221). |

Request example:

PUT

<http://prov.server.net/sa/rest/v1/prov/agencies/8/dns>

HTTP body:

{

"dns": [

"38642071221",

"38642071223",

"38642071432",

...

]

}

**Note:**

All dns that are not presented in request body and are already members of current agency, will be removed from agency and and will be presented ass free residential dns.

**Response**

In case of success 200 OK is returned.

Response example

HTTP body:

{

"code": 200,

"msg": "All dns successfully assigned to agency!"

}

### Add existing dns to agency

Operation adds existing residential dns to desired agency. For successful request all dns in request body must be residential dns. Each dn must be in international format (cc+ndc+number e.g. 38611100100).

**Request**

Request to assign list of DNs

**PUT http://<hostname>/sa/rest/v1/prov/agencies/{id}/dns?action=add**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *id* | *yes* | *Integer* | ID of an agency |
| *action* | *yes* | *String* | For successful adding, action parameter must be equal to “add” |

Parameters in request HTTP body:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dns* | *yes* | *String* | List of telephone numbers. All dns must be in international format (country code – national destination code + number e.g. 38642071221). |

Request example:

PUT

<http://prov.server.net/sa/rest/v1/prov/agencies/8/dns?action=add>

HTTP body:

{

"dns": [

"38642071221",

"38642071223",

"38642071432",

...

]

}

**Response**

In case of success 200 OK is returned.

Response example

HTTP body:

{

"code": 200,

"msg": "All dns successfully added to agency!"

}

### Remove existing dns from agency

Operation removes existing agency dns from current agency. For successful request all dns in request body must be members of current agency. Each dn must be in international format (cc+ndc+number e.g. 38611100100). After successfull removal, all dns will be presented as residential numbers.

**Request**

Request to remove dns from agency with provided id:

**PUT http://<hostname>/sa/rest/v1/prov/agencies/{id}/dns?action=remove**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *id* | *yes* | *Integer* | ID of an agency |
| *action* | *yes* | *String* | For successful removal action parameter must be equal to “remove” |

Parameters in request HTTP body:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dns* | *yes* | *String* | List of telephone numbers. All dns must be in international format (country code – national destination code + number e.g. 38642071221). |

Request example:

PUT

<http://prov.server.net/sa/rest/v1/prov/agencies/8/dns?action=remove>

HTTP body:

{

"dns": [

"38642071221",

"38642071223",

"38642071432",

...

]

}

**Response**

In case of success 200 OK is returned.

Response example

HTTP body:

{

"code": 200,

"msg": "All dns successfully removed from agency!"

}

### Get all dns in agency

Operation returns list of dns that already exist in agency.

**Request**

Request URI to get list of DNs

**GET http://<hostname>/sa/rest/v1/prov/agencies/{id}/dns**

For description of id parameter refer to description of parameters in request URI in chapter 5.1.2 – request section.

Request example:

GET

<http://prov.server.net/sa/rest/v1/prov/agencies/8/dns>

**Response**

In case of success list of dns in JSON format is returned.

Response example:

200 OK

HTTP body:

{

"dns":

[

"38642071221",

"38642071223",

"38642071432",

...

]

}

# Suspend/Resume of DN or BG

Api to suspend or resume certain DN or all DNs in provided BG.

## Suspend/Resume DN

Operation suspends/resumes certain DN. That means all telephony services for certain DN are blocked. In case DN has assigned user, suspension also blocks user access to web portal. Diametrical operation is DN resume.

**Request**

Request URI for dn suspension:

**PUT http://<hostname>/sa/rest/v1/prov/dns/<dn>?action=suspend**

Request URI for resuming dn:

**PUT http://<hostname>/sa/rest/v1/prov/dns/<dn>?action=resume**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *yes* | *String* | Selected DN |
| *action* | *yes* | *String* | Suspends or resumes selected DN |

Example:

Suspend DN:

PUT http://prov.server.net/sa/rest/v1/prov/dns/38642071221?action=suspend

Resume DN:

PUT http://prov.server.net/sa/rest/v1/prov/dns/38642071221?action=resume

**Response**

In case of success 200 Ok is returned.

Response example

HTTP body:

{

"code": 200,

"msg": "Dn: '38044500703' suspended."

}

### Suspension status

HTTP GET is used to retrieve information about DN suspension status. The method returns suspension status info.

**Request**

Request URI:

**GET http://<hostname>/sa/rest/v1/prov/dns/<dn>/suspensionstatus**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *yes* | *String* | Selected DN |

Example:

GET http://prov.server.net/sa/rest/v1/prov/dns/38642071221/suspensionstatus

**Response**

Response contains suspension status data in JSON format. Response body contains single parameter with true/false value.

Response example:

HTTP body

{

"suspensionstatus" : true

}

In case of error check chapter 2.2 for error responses.

## Suspend/Resume BG

Api to suspend or resume all DNS in BG with provided id.

HTTP PUT with additional argument is used to suspend or resume BG – all DNs in BG.

**Request**

Request URI for BG suspension:

**PUT http://<hostname>/sa/rest/v1/prov/bgs/<id>?action=suspendbg**

Request URI to resume al DNS in BG:

**PUT http://<hostname>/sa/rest/v1/prov/bgs/<id>?action=resumebg**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *id* | *Yes* | *Integer* | Selected Business Group ID |
| *action* | *Yes* | *Boolean* | Suspends or resumes all dns in provided BG |

Request example:

Suspend BG:

PUT http://prov.server.net/sa/rest/v1/prov/bgs/10?action=suspendbg

Resume BG:

PUT http://prov.server.net/sa/rest/v1/prov/bgs/10?action=resumebg

**Response**

In case of success 200 OK is returned.

Response example:

HTTP body:

{

"code": 200,

"msg": "Business group with id: '18' resumed."

}

If exception is thrown during operation, all error messages for problematic DNs are joined and presented as single error response message.

Error example:

HTTP body

{

"code": 400,

"msg": "Provided DN '380800500700' already 'UNBLOCK'. Provided DN '380800500601' already 'UNBLOCK'. "

}

# Supplementary services

## Get phone devices

HTTP GET is used to retrieve data for phone device.

**Request**

Request URI:

**GET http://<hostname>/sa/rest/v1/user/phone/<dn>/devices**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *Yes* | *String* | Selected DN |

Example:

GET http://prov.server.net/sa/rest/v1/user/phone/38641000100/devices

**Response**

Response contains phone device data in JSON format.

Response body example:

{

"mainNumber": "38641000100",

"device": { /\* Optional \*/

"type": "Yealink",

"model": "T28",

"desc": ""

},

"mduDevices": {

[

{

"number": "FFA0000001",

"type": "IT Phone",

"model": "M10",

"desc": "Android"

}

, {

"number": "FFA0000002",

"type": "IT Phone",

"model": "W10",

"desc": "Windows"

}

]

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *mainNumber* | *Yes* | *String* | User's phone number |
| *device* | *Yes* | *JsonObject* | Pilot phone device |
| *mduDevices* | *No* | *JsonObject* | List of MDU devices |
| *number* | *Yes* | *String* | Line number |
| *type* | *Yes* | *String* | Line device type |
| *model* | *Yes* | *String* | Line device model |
| *desc* | *No* | *String* | Line device description |

In case of error check chapter 2.2 for error responses.

## CFU Service

Call Forwarding Unconditional

### Get service status

**Request**

Request URI:

**GET http://<hostname>/sa/rest/v1/service/cfu/<dn>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *Yes* | *String* | Selected DN |

Example:

GET http://prov.server.net/sa/rest/v1/service/cfu/38641000100

**Response**

Response contains service activation state and destination number in JSON format.

Response body example:

{

"mainNumber": "38641000100",

"activation": "yes",

"destinationNumber": "38641000200"

}

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Mandatory*** | ***Type*** | ***Explanation*** |
| *mainNumber* | *Yes* | *String* | User's phone number |
| *activation* | *Yes* | *String* | Is service activated |
| *destinationNumber* | *No* | *String* | Destination number |

### Modify service

**Request**

Request URI:

**PUT http://<hostname>/sa/rest/v1/service/cfu/<dn>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *Yes* | *String* | Selected DN |
| *activation* | *Yes* | *String* | Is service activated |
| *destinationNumber* | *No* | *String* | Destination number |

Example:

PUT http://prov.server.net/sa/rest/v1/service/cfu/38641000100

HTTP body:

{

"activation": "yes",

"destinationNumber": "38641000200"

}

**Response**

Response contains service activation state and destination number in JSON format.

Response body example:

{

"mainNumber": "38641000100",

"activation": "yes",

"destinationNumber": "38641000200"

}

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *mainNumber* | *Yes* | *String* | User's phone number |
| *activation* | *Yes* | *String* | Is service activated |
| *destinationNumber* | *No* | *String* | Destination number |

## CFB Service

Call Forwarding Busy

### Get service status

**Request**

Request URI:

**GET http://<hostname>/sa/rest/v1/service/cfb/<dn>**

Same as 6.2 CFU Service.

### Modify service

**Request**

Request URI:

**PUT http://<hostname>/sa/rest/v1/service/cfb/<dn>**

Same as 6.2 CFU Service.

## DND Service

Do Not Disturb service

### Get service status

**Request**

Request URI:

**GET http://<hostname>/sa/rest/v1/service/dnd/<dn>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *Yes* | *String* | Selected DN |

Example:

GET http://prov.server.net/sa/rest/v1/service/dnd/38641000100

**Response**

Response contains service activation state in JSON format.

Response body example:

{

"mainNumber": "38641000100",

"activation": "yes"

}

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *mainNumber* | *Yes* | *String* | User's phone number |
| *activation* | *Yes* | *String* | Is service activated |

### Modify service

**Request**

Request URI:

**PUT http://<hostname>/sa/rest/v1/service/dnd/<dn>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *Yes* | *String* | Selected DN |
| *activation* | *Yes* | *String* | Is service activated |

Example:

PUT http://prov.server.net/sa/rest/v1/service/dnd/38641000100

HTTP body:

{

"activation": "yes"

}

**Response**

Response contains service activation state and destination number in JSON format.

Response body example:

{

"mainNumber": "38641000100",

"activation": "yes"

}

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *mainNumber* | *Yes* | *String* | User's phone number |
| *activation* | *Yes* | *String* | Is service activated |

## DNDGM Service

Do not disturb for MDU Member

### Get DNDGM members

**Request**

Request URI:

**GET http://<hostname>/sa/rest/v1/service/dndgm/<dn>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *Yes* | *String* | Selected DN |

Example:

GET http://prov.server.net/sa/rest/v1/service/dndgm/38044800702

**Response**

Response contains all members of dndgm service for provided pilot number.

**Parameters in response body:**

|  |  |  |
| --- | --- | --- |
| ***Parameter Name*** | ***Type*** | ***Explanation*** |
| *dn* | *String* | Members DN |
| *activation* | *String* | Is service activated |
| *authorization* | *String* | Authorization of DNDGM service |
| *type* | *String* | DNDGM member type |

Response body example:

[

{

"dn": "800702",

"authorization": "TIME",

"activation": "false",

"type": "Pilot number"

},

{

"dn": "FEE1024236",

"authorization": "TIME",

"activation": "false",

"type": "KSB-phone Communicator (W10)"

},

{

"dn": "FAA03011800602",

"authorization": "TIME",

"activation": "true",

"type": "Remote to mobile line"

},

{

"dn": "FBC234",

"authorization": "TIME",

"activation": "false",

"type": "Anywhere line"

},

{

"dn": "FEE1024239",

"authorization": "TIME",

"activation": "false",

"type": "KSB-phone Communicator (A10/i10)"

},

{

"dn": "FEE1024240",

"authorization": "TIME",

"activation": "false",

"type": "KSB-phone Communicator (A10/i10)"

}

]

### Get service status

**Request**

Request URI:

**GET http://<hostname>/sa/rest/v1/service/dndgm/<dn>/member/<memberDn>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *Yes* | *String* | Selected DN |
| *memberDn* | *Yes* | *String* | Line number |

Example:

GET http://prov.server.net/sa/rest/v1/service/dndgm/38641000100/member/FFA0000001

**Response**

Response contains service activation state and time list if the service is set to time automatic mode in JSON format. DNDGM service supports 2 deferent modes; Automatic mode and Manual mode.

Response body example:

* Automatic mode:

{

"mainNumber": "38641000100",

"memberDn": "FFA0000001",

"activation": "yes",

"timeList": [

{

"dayCategory": "workDay",

"switchTime": "00:00",

"switchType": "off"

},

{

"dayCategory": "workDay",

"switchTime": "23:00",

"switchType": "on"

},

{

"dayCategory": "weekend",

"switchTime": "00:00",

"switchType": "off"

},

{

"dayCategory": "weekend",

"switchTime": "23:00",

"switchType": "on"

}

]

}

* Manual mode – service activated:

{

"mainNumber": "38641000100",

"memberDn": "FFA0000001",

"activation": "yes"

}

* Manual mode – service deactivated:

{

"mainNumber": "38641000100",

"memberDn": "FFA0000001",

"activation": "no"

}

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *mainNumber* | *Yes* | *String* | User's phone number |
| *memberDn* | *Yes* | *String* | Line number |
| *activation* | *Yes* | *String* | Is service activated |
| *timeList* | *No* | *JsonObject* | List of switching times |
| *dayCategory* | *Yes* | *String* | Day category |
| *switchTime* | *Yes* | *String* | Switch time |
| *switchType* | *Yes* | *String* | Switch type |

### Modify service

**Request**

Request URI:

**PUT http://<hostname>/sa/rest/v1/service/dndgm/<dn>/member/<memberDn>**

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name*** | ***Required*** | ***Type*** | ***Explanation*** |
| *dn* | *Yes* | *String* | Selected DN |
| *memberDn* | *Yes* | *String* | Line number |
| *activation* | *Yes* | *String* | Is service activated |
| *mode* | *Yes* | *String* | Service mode |
| *timeList* | *No* | *JsonObject* | List of switching times |
| *dayCategory* | *Yes* | *String* | Day category |
| *switchTime* | *Yes* | *String* | Switch time |
| *switchType* | *Yes* | *String* | Switch type |

Example:

PUT http://prov.server.net/sa/rest/v1/service/dndgm/38641000100/member/FFA0000001

Request body:

* Automatic mode:

{

"mode": "automatic",

"activation": "yes", // Ignored

"timeList": [

{ // Work day example

"dayCategory": "workDay",

"switchTime": "00:00",

"switchType": "off"

},

{

"dayCategory": "workDay",

"switchTime": "23:00",

"switchType": "on"

},

{ // Weekend example

"dayCategory": "weekend",

"switchTime": "00:00",

"switchType": "off"

},

{

"dayCategory": "weekend",

"switchTime": "23:00",

"switchType": "on"

},

{ //Holliday example

"dayCategory":"holiday",

"switchTime":"00:00",

"switchType":"off"},

{,

"dayCategory": "holiday",

"switchTime": "23:00",

"switchType":"on"

},

{ //Day Before example

"dayCategory":"DayBeforeHoliday",

"switchTime":"00:00",

"switchType":"off"

},

{

"dayCategory":"DayBeforeHoliday",

"switchTime":"23:00",

"switchType":"on"

}

]

}

* Manual mode – service activated:

{

"mode": "manual",

"activation": "yes"

}

* Manual mode – service deactivated:

{

"mode": "manual",

"activation": "no"

}

**Response**

Same as 6.5.1 Get service status.

In case of error check chapter 2.2 for error responses.

# User Api

API for user management: to manage users and residential or agency telephone numbers. API can be used by service provider administrator users: “admin” and “onlineUser” and also by agency admin for his group.

## Create or update user

Method Creates/updates user and assign it to existing resident or agency DN.

**Request**

Request URI to assign user to provided DN

**PUT http://<hostname>/sa/rest/v1/users/<username>**

Request URI example

[http://prov.server.net/sa/rest/v1/users/38011114114](http://localhost:8080/sa/rest/v1/users/38)

Parameters in request body:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| *username* | *yes* | *380152222* | Residents username |
| *firstName* | *no\** |  | Users first name |
| *middleName* | *no\** |  | Users middle name |
| *lastName* | *no\** |  | Users last name |
| *email* | *yes* |  | Users email |
| *dns* | *yes* |  | Telephone number, which must be in international format (country code – national destination code + number e.g. 38642071221). |
| *bgid* | *no* | *1* | If paramter set, it should have value 1 or null. |
| *timeZone* | *no* | *CET* | Users time zone presented as short three letter string (eg. GMT", "WET", "CET", "MET”…)  If timeZone parameter not set, system timezone from config parameter will used. |
| *contactNumber* | *No* | *1234* | Users contact number |
| *address* | *no* | *Pohorska 1* | Users address |

\*At least one of parameters (firstName, lastName, middleName) must have a value.

Request body example:

HTTP body:

{

"username": "38044256356",

"firstName": "38044256356",

"middleName": "",

"lastName": "",

"email": "janez@example.com",

"dns": [

"38044256356"

],

"bgid": 1,

"timeZone": "CET",

"contactNumber": null,

"address": "Pohorska 1"

}

**Response**

For response parameters description refer to the chapter 8.1. – response section

## Get users Request

Method returns user data.

**Request**

Request URI to retrieve users data:

**GET http://<hostname>/sa/rest/v1/users/<username>**

Request URI example

GET <http://prov.server.net/sa/rest/v1/users/38011665544>

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| username | yes | 380152222 | Users username |

**Response**

For parameter description refer to chapter 8.2 – parameters in request body.

Response body example:

HTTP 200 OK

{

"username": "38011665544",

"firstName": "38011665544",

"middleName": "",

"lastName": "",

"email": "janez@example.com",

"dns": [

  "38011665544"

],

"bgid": 1,

"timeZone": "CET",

"contactNumber": null,

"address": null

}

## Delete user

Deletes user data, but DN remains.

**Request**

**Request URI to delete user**

**DELETE http://<hostname>/sa/rest/v1/users/<username>**

Request URI example

GET <http://prov.server.net/sa/rest/v1/users/380152222>

Parameters in URI request:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| username | Yes | 380152222 | Users username |

**Response**

In case of success 200 OK is returned.

Response body example:

HTTP 200 OK

{

"code": 200,

"msg": "User successfully deleted!"

}

# Address book Api

## Contacts

Api to retrieve address book contacts data of provided user.

### Get contacts authorized for username

Method returns all contacts in business group, all private contacts for user <username> and VIP contacts if user <username> is authorized for VIP contacs.

**Request**

**GET http://<hostname>/sa/rest/v1/ab/contacts/<username>**

Request URI example

GET <http://prov.server.net/sa/rest/v1/ab/contacts/>admin

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| username | no\* | admin | Username |

\*If username is not provided in request URI, then search will be performed for username from authorization. If authorized with SP ADMIN account, then you may search all cantacts in all groups (without private contacts). If Authorized as Enterprise Admin, than you may search all cantacts in your business group (without private contacts, unless you search for yourself). If Authorized as normal user then username in request URI must match with username form authorization. Normal user can also see his private contacts.

**Response**

In case of success 200 OK is returned.

For response parameters description refer to chapter 9.2 – Contact section.

Response example:

HTTP 200 OK

[

{

"id": 9,

"bgid": 4,

"firstName": "asdf",

"lastName": "",

"middleName": "asd",

"email": "",

"workDn": "1234",

"mobileDn": "",

"auth": "$PUB",

"contactData": {

"title": "asdf",

"homeDn": "",

"company": "",

"department": "",

"sourceBgid": 4,

"additionalNum1": "",

"additionalNum2": "",

"contactSource": "$WP",

"icon": "external-contacts"

},

"name": "asdf asd",

"fullName": "asdf asd",

"contactUsername": "asdfasd9" },

{

"id": 10,

"bgid": 4,

"firstName": "eee",

"lastName": "eee",

"middleName": "eee",

"email": "",

"workDn": "235",

"mobileDn": "",

"auth": "$PUB",

"contactData": {

"title": "eee",

"homeDn": "",

"company": "",

"department": "",

"sourceBgid": 4,

"additionalNum1": "",

"additionalNum2": "",

"contactSource": "$WP",

"icon": "external-contacts"

},

"name": "eee eee eee",

"fullName": "eee eee eee",

"contactUsername": "eeeeeeeee10"

},...

]

### Search contacts authorized for user

Same as get Contacts (chapter 9.1.1) with additional search parameters.

**Request**

Request URI with additional parameters to retrieve contacts:

**GET http://<hostname>/sa/rest/v1/ab/contacts/<username>?searchTerm=<searchTerm>&limit=<limit>&offset=<offset>**

Request URI example:

GET http://prov.server.net/sa/rest/ab/contacts/38037500500&searchTerm=Jane\*&limit=2&offset=2

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| *searchTerm* | *no* | *Janez or Jan\** | Search term |
| *limit* | *no* | *0* | Number of max returned results |
| *offset* | *no* | *0* | Number of skipped results |
| *username* | *yes* | 38037500500 | Users username |

Search is performed over parameters firstName, LastName, userEmail, workDn, mobileDn, shortFix, shortMobile and company. For all parameters wildcard is used at the end.

**Response**

Same as Get contacts (chapter 9.1.1 – response section).

### Get contact avatar

Method returns contacts avatar if user is authorized for requested contactId.

**Request**

Request URI to retrieve contacts avatar

**GET http://<hostname>/sa/rest/v1/ab/contacts/<username>/<contactId>/avatar**

URI example

GET http://prov.server.net/sa/rest/v1/ab/contacts/Janez/17/avatar

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| username | yes | openaat | User's username |
| contactId | yes | 3 | Contact's id number |

**Response**

Base64 encoded image string wrapped in JSON notation..

Response example:

HTTP 200 OK

{

"avatar": "AB034…"

}

### Get contacts by business group id

Method returns Contacts data for provided Business group. If bgid not provided, the request is rejected.

**Request**

Request URI to retrieve business group contacts

**GET http://<hostname>/sa/rest/v1/ab/contacts?bgid=<bgid>**

Request URI example:

GET http://prov.server.net/sa/rest/v1/ab/contacts?bgid=1

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| bgId | yes | 12 | Business group id |

**Response**

In case of success 200 OK is returned.

For response parameters description refer to chapter 9.2 – Contact section.

Response example:

HTTP 200 OK

[

{

"id": 116,

"bgid": 12,

"firstName": "test1",

"lastName": "",

"middleName": "test",

"email": "",

"workDn": "0445005001",

"auth": "$NORMAL",

"userId": 16832,

"contactData": {

"username": "380445005001",

"company": "Bostjan group",

"contactSource": "$IBC",

"icon": "wp-contacts"

},

"name": "test1 test",

"fullName": "test1 test",

"contactUsername": "380445005001"

},...

]

### Get contacts in business group by given phone number

**Request**

Request URI to retrieve contact by given phone number

**GET http://<hostname>/sa/rest/v1/ab/contacts?bgid=<bgid>&number=<contactNumber>**

URI example:

http://prov.server.net/sa/rest/ab/contacts?bg=4&number=044555000

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| bgId | yes | 1 | Business group id |
| contactNumber | yes | 044555000 | Contact's (LAC) phone number. Number must be in followed fomat: national prefix + national destination code + number (e.g. 0442071221). |

**Response**

In case of success 200 OK is returned.

For response parameters description refer to chapter 9.2 – Contact section.

Response example:

HTTP 200 OK

{

"id": 34510,

"bgid": 4,

"firstName": "vodilna 2",

"lastName": "",

"middleName": "",

"email": "",

"workDn": "0447700002",

"auth": "$NORMAL",

"userId": 52680,

"contactData": {

"username": "380447700002",

"company": "Leskovar",

"department": "aaa",

"contactSource": "$IBC",

"icon": "wp-contacts"

},

"name": "vodilna 2",

"fullName": "vodilna 2",

"contactUsername": "380447700002"

}

### Search contacts in BGID without private and VIP

Only company contacts in BGID are searched. VIP and private contacts are not returned.

**Request**

Request URI to retrieve contacts with additional search parameters

**GET http://<hostname>/sa/rest/v1/ab/contacts?bgid=<bgid>&searchTerm=<searchTerm>&limit=<limit>&offset=<offset>**

Request URI example:

GET

<http://prov.server.net/sa/rest/v1/ab/contacts?bgid=4>? searchTerm=Jane\*&limit=20&offset=10

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| bgId | yes | 1 | Business group id |
| searchTerm | no | Janez or Jan\* | Search term |
| limit | no | 0 | Number of max returned results |
| offset | no | 0 | Number of skipped results |

Search is performed over parameters firstName, LastName, userEmail, workDn, mobileDn, shortFix, shortMobile and company. For all parameters wildcard is used at the end.

**Response**

Same as Get contacts in chapter 9.1 – response section.

### Count contacts in business group

**Request**

Request URI to count contacts in BG

GET http://<hostname>/sa/rest/v1/ab/contacts/bg/<bgId>/size

Request URI example

GET http://prov.server.net/sa/rest/v1/ab/contacts/bg/12/size

Parameters in request URI:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| bgId | yes | 1 | Business group id |

**Response**

In case of success 200 OK is returned.

It returns number of contacts in business group as JSON object;

Response body example:

{

"size" : 112

}

## Objects

### User

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| bgId | yes | 1 | Business group id |
| userId | yes | 10198 | User's id number |
| username | yes | 380152222 | Username |
| firstName | no | Janez | First name |
| lastName | no | Novak | Last name |
| middleName | no | Bine | Middle name |
| userEmail | no | novak@iskratel.si | Email address |
| auth | yes |  | Address book authorization |

### Phone

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| phoneId | yes | 1 | Phone's id number |
| csId | yes | 7000 | Call server's id number |
| phoneNumber | yes | 0155550001 | Phone number |

### Contact

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| id | yes | 1 | Contact's id number |
| bgid | yes | 1 | Business group id |
| firstName | one of the names | Janez | First name |
| lastName | one of the names | Novak | Last name |
| middleName | one of the names | Bine | Middle name |
| email | no | novak@iskratel.si | Email address |
| workDn | this or mobileDn | 0155550001 | Work number |
| mobileDn | this of workDn | 0156660001 | Mobile number |
| auth | yes | * $PUB (Public contact), * $VIP (VIP contact), * $ADMIN (VIP Admin contact), * $<username> (Users private contact), * $NORMAL (Normal contact) | VIP Indication |
| userId | no |  | User's id number |
| name | yes | Janez Bine Novak | Full name without title |
| fullName | yes | Mr. Janez Bine Novak | Full name |
| ~~contactUsername~~ | ~~yes~~ |  |  |
| contactData | no | See Contact data (contactData) |  |

### Contact data (contactData)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Required** | **Example** | **Description** |
| username | no | username | Username |
| title | no | Mr. | Title |
| homeDn | no | 0155550001 | Home phone number |
| workFax | no | 0155550001 | Work FAX number |
| company | no | Iskratel | Company name |
| department | no | Razvoj | Department |
| ~~address~~ | ~~no~~ |  |  |
| shortFix | no | 100 | Work internal number |
| shortMobile | no | 101 | Mobile internal number |