

November 2016

# MSP Integration API

Version v0.2

**f**

## **Legal Notices**

Copyright © 2015, 2016 streamnow AG. All rights reserved.

This document contains confidential, proprietary information of streamnow AG. Unauthorized copying, reproduction or disclosure of any portion of the contents of these materials by any means—including printed or duplicated materials, onscreen publication or Web documentation—is expressly forbidden.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH STREAMNOW AG. PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. STREAMNOW AG. ASSUMES NO LIABILITY WHATSOEVER, AND STREAMNOW AG DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF STREAMNOW AG PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. UNLESS OTHERWISE AGREED IN WRITING BY STREAMNOW AG. THE PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

streamnow AG may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined”. streamnow reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice.

streamnow and MSP logo are trademarks of streamnow AG. Third party trademarks and images shown are the property of their respective owners.

The MSP API is neither affiliated with nor endorsed by the owners of the third party images, names and logos displayed herein.

Use of this API is subject to the streamnow MSP APIs Terms of Use license.

## Change Control

Version	Changes
v0.1	Initial version.
v0.2	<p>New APIs for:</p> <ul style="list-style-type: none"><li>- Address Management: addressCreate, addressDelete, addressList, addressUpdate</li><li>- User Management: getActiveUsers, Users active, Users delete, Users disable</li><li>- Group Management: getGroupsBP, getGroupsUser</li><li>- Service Management: services disable for user, services enable for user</li></ul> <p>ESP is now MSP.</p>

## Contents

<b>Legal Notices.....</b>	<b>2</b>
<b>Change Control.....</b>	<b>3</b>
<b>Contents.....</b>	<b>4</b>
<b>1. Introduction .....</b>	<b>7</b>
<b>2. Overview .....</b>	<b>8</b>
2.1. MSP Entities.....	8
2.2. Basic API concepts.....	8
<b>3. Authentication .....</b>	<b>11</b>
3.1. login (using username/email and password) .....	11
3.2. login (using device uid).....	16
3.3. logout.....	18
3.4. reset .....	19
3.5. refreshToken .....	19
3.6. revokeToken.....	20
<b>4. Comlite .....</b>	<b>21</b>
4.1. getTerminals.....	21
<b>5. Docman.....</b>	<b>22</b>
5.1. getContact.....	22
5.2. getContactAvatar.....	24
5.3. getDocInfo (POST).....	24
5.4. getDocInfo (GET).....	25
5.5. getDocsInfo.....	26
5.6. getImage .....	26
5.7. getNotifications.....	27
<b>6. Operations on Dashboard .....</b>	<b>29</b>
6.1. addressCreate.....	29
6.2. addressDelete .....	29
6.3. addressList .....	30
6.4. addressUpdate .....	32
6.5. changePassword .....	34
6.6. checkHardware .....	35
6.7. Email send .....	35
6.8. getActiveUsers.....	<i>¡Error! Marcador no definido.</i>
6.9. getBasicUserInfo .....	42
6.10. Get billing user info.....	44
6.11. getIP .....	45

---

6.12.	<i>getGroupsBP</i> .....	46
6.13.	<i>getGroupsUser</i> .....	47
6.14.	<i>getInstallationUserInfo</i> .....	48
6.15.	<i>getSIPIInfo</i> .....	49
6.16.	<i>getSMB</i> .....	49
6.17.	<i>getUserInfo</i> .....	51
6.18.	<i>add-product (invoices)</i> .....	54
6.19.	<i>getBillingStatus (invoices)</i> .....	55
6.20.	<i>sendPushNotification</i> .....	<b><i>¡Error! Marcador no definido.</i></b>
6.21.	<i>sendSMS</i> .....	60
6.22.	<i>Services Disable</i> .....	60
6.23.	<i>Services disable for user</i> .....	61
6.24.	<i>Services enable</i> .....	62
6.25.	<i>Services enable for user</i> .....	62
6.26.	<i>setDeviceToken</i> .....	63
6.27.	<i>updateIP</i> .....	64
6.28.	<i>updateMACstatus</i> .....	65
6.29.	<i>Users active</i> .....	66
6.30.	<i>Users delete</i> .....	67
6.31.	<i>Users disable</i> .....	67
6.32.	<i>Register(users)</i> .....	68
6.33.	<i>Validate end user (users)</i> .....	<b><i>¡Error! Marcador no definido.</i></b>
<b>7.</b>	<b>Operations on My Entertainment</b> .....	<b>74</b>
7.1.	<i>getContact (myentertainment)</i> .....	74
7.2.	<i>getContactAvatar (my entertainment)</i> .....	75
7.3.	<i>getCredentials (myentertainment)</i> .....	75
<b>8.</b>	<b>OPERATIONS ON MY RESTAURANT</b> .....	<b>77</b>
8.1.	<i>getRestaurant</i> .....	77
8.2.	<i>getRestaurantMenu</i> .....	86
8.3.	<i>getRestaurantProfile</i> .....	87
<b>9.</b>	<b>OPERATIONS ON MY VPN</b> .....	<b>92</b>
9.1.	<i>getCamsByUser</i> .....	92
9.2.	<i>getCertificates</i> .....	92
9.3.	<i>getContact</i> .....	93
9.4.	<i>getContactAvatar</i> .....	94
9.5.	<i>setCamServiceByUser</i> .....	95
9.6.	<i>unsetCamServiceByUser</i> .....	95
<b>10.</b>	<b>Operation on SW Docman</b> .....	<b>97</b>
10.1.	<i>getContact</i> .....	97
10.2.	<i>getContactAvatar</i> .....	99
10.3.	<i>getDocInfo (POST)</i> .....	99
10.4.	<i>getDocInfo (GET)</i> .....	100
10.5.	<i>getDocsInfo</i> .....	101

---

10.6.	<i>getImage</i> .....	101
10.7.	<i>getNotifications</i> .....	102

## **1. Introduction**

This document is intended to provide the specification of the MSP API that allows devices, such as set-top-boxes, smartphones and tablets, to retrieve online information from the MSP system. It also allows service providers to connect with the MSP.

Therefore, it is oriented to those people who develop client applications, such as Android and iOS developers or Service integrators.

**Important note:** this document is in a development stage and it should not be stable until reaching version 1.0, so use it at your own risks.

## 2. Overview

### 2.1. MSP Entities

This section describes the entities involved in the integration API operations.

**End-users** - The term *end-users* refers to those users which use MSP from a device like a set-top-box, a smartphone, a tablet, etc.

MSP end-users have to be registered on the MSP system so that they can access the services. That registration is done through a provisioning process.

**Service Providers** - The term *service-provider* refers to the connections with the MSP from service providers (SP), so they can offer services to end-users or get information from MSP about the user or add information for an end-user, as consequence of the use of the service by the end-user.

### 2.2. Basic API concepts

#### 2.2.1. API Protocol

The integration API provided by MSP is a HTTP API.

The API is divided into different parts. Each part describes which MSP objects are involved, specifying the attributes that take part of each object, specifically its data type and whether the attribute is mandatory or optional.

In this document mandatory attributes and mandatory parameters are written with bold text, while optional attributes and optional parameters are written with normal font.

Each API operation defines its own JSON format, but there are common structures which may appear in every API response, for instance, lists and errors.

#### 2.2.2. Workflow

The steps to use the integration API from an application are

1. Login into the MSP system using a valid user identifier and password.
2. If MSP authorizes the application and the end-user, it will create a temporal session, and it will send a cookie/token to the client.
3. The client will invoke the corresponding API operations provided by MSP, sending the cookie/token apart from the required parameters of each operation. MSP will identify the end-user (using the cookie/token) and send the requested data in JSON or XML format. These operations will be explained in the next sections.
4. Close the session invoking the logout operation.



### 2.2.3. Lists Objects

Lists always have the following structure:

```
{
  elements: [
    {...},
    {...}
  ],
  iterator: {
    existsPrev: true | false, existsNext: true | false,
    count: <TOTAL NUMBER OF ELEMENTS>
  }
}
```

### 2.2.4. Error Handling

When an operation returns an error, an **error** field with the list of all errors that happened is always returned. That field is an array of objects with the following fields:

- **code**: identifies the error. Each API method defines a specific set of error codes.
- **description**: explaining the cause of the error.
- **param**: list of objects to specify extra data associated to the error. For example:

```
{
  ...,
  errors: [
    {
      code: <ERROR_CODE>,
      params: [
        {
          name: <PARAM_NAME>,
          value: <PARAM_VALUE>
        },
        {...}
      ],
      description: <ERROR_DESCRIPTION>
    },
    {...}
  ]
}
```

### 2.2.5. Conventions

- **Client** - Client application.
- **Status** - HTTP status code of response.
- All the possible responses are listed under 'Responses' for each method. Only one of them is issued per request server.
- All response is in JSON format.
- All request parameters are mandatory unless explicitly marked as **[optional]**
- The type of values accepted for a *request* parameter are shown the values column like this **[10|<any number>]**. The | symbol means *OR*. If the parameter is **[optional]**, the default value is shown in blue bold text, as **10** is written in **[10|<any number>]**.

### 2.2.6. Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

**2XX** - Success of some kind

**4XX** - Error occurred in client's part

**5XX** - Error occurred in server's part

Status Code	Description
200	OK
201	Created
202	Accepted (Request accepted, and queued for execution)
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict
412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

### 3. Authentication

Besides the use of a secure protocol like HTTPs, MSP restricts the access to the integration API. Hence, only authenticated clients have access to the operations provided, and only authorized developers can use the API.

The access from a client application to the API operations is allowed only after carrying out an authentication process, in which the MSP system check both the developer code and the user identifier (and password if required by the end-user).

#### 3.1. login (using username/email and password)

This operation is used to authenticate the user with the system and obtain the auth\_token and all the user parameters. It requires username/email and password.

That applies to end-users and service providers.

##### Request

Method	URL
POST	auth/login/

##### Parameters

Type	Name	Description	Type	Data Type
POST	email	User identifier	form	string
POST	password	User password	form	string
POST	source	Source of the request	form	string

Where source could be:

- Web: for web application (usually web users)
- Service
- Mobile
- Tablet

## Response

Status	Response
200	<p><b>Response will be an object containing the user information, the tokens as well as the business partner and services information.</b></p> <pre> {   "access_token": "CsmKJCmBBHOv445N6IwLzjRaQb4oA2nQ4MSJoYNf",   "refresh_token": "X3cRfJBbNw98zGV3o9c1M1qMF7z7wuozolxKKc5W",   "validUntil": "2016-08-12 17:39:05 CET",   "deviceSessionId": "0076491212d5be6c0636c227aaafccf54303c5f3",   "user_info": {     ...      "partner": {       ...      },     "language": "de",     "bp_internal_number": ""   },   "available_languages": [     ...    ],   "url_test": "service?access_token=CsmKJCmBBHOv445N6IwLzjRaQb4oA2nQ4MSJoYNf",   "available_services": [     {       ...      },     {       ...      }   ],   "categories": [     {       ...      },     {       ... </pre>

```
}
]
}
```

**An example response is:**

```
{
  "access_token":
  "b2uOQxz60KYvgKhZkWeyfBIItJt2m1SW3yUyz0sOY",
  "refresh_token":
  "6emgnQ1uS59VYXnbiYE1BYg05C1jvf3BNrF2xmu7",
  "validUntil": "2016-08-14 13:06:14 CET",
  "deviceSessionId": "b0813ab41d4b0d1dbf2f7c41750fd04feae5565c",
  "user_info": {
    "id": "1097",
    "name": "Fam.",
    "lastname": "Muster VZUG",
    "email": "demo.vzug@streamnow.ch",
    "gender": "male",
    "username": "demo.vzug",
    "default_address_id": null,
    "telephone": "",
    "mobil": "",
    "fax": "",
    "status": "1",
    "image": "http://esp-
development.streamnow.ch/img/users/generic.jpg",
    "demo": "1",
    "payment_allowed": "",
    "partner": {
      "company": "V-ZUG",
      "name": "V Zug",
      "phone": "",
      "email": "vzug@streamnow.ch",
      "logo": "http://esp-
development.streamnow.ch/img/users/4e2545f819e67f0615003dd7e04a
087.jpeg",
      "background_image": "http://esp-
development.streamnow.ch/img/partner/background4e2545f819e67f061
003dd7e04a6087.jpeg",
      "background_smartphone_image": "http://esp-
development.streamnow.ch/img/partner/background_smartphone4e2545
819e67f0615003dd7e04a6087.jpeg",
      "font_color_top": "rgba(255,255,255,1)",
      "color_top": "rgba(40,59,154,1)",
      "font_color_service": "rgba(0,0,0,1)",
      "font_color_bottom": "rgba(0,0,0,1)",
      "smartphone_app_name": "",
      "line_color_smartphone": "rgba(0,0,0,1)",
```

```

"font_color_smartphone": "rgba(0,0,0,1)",
"background_color_smartphone": "rgba(40,59,154,1)",
"background_color_wall_tablet": "",
"background_color_icon_smartphone": "rgba(0,0,0,1)",
"paid_services": false,
"support_button": false,
"url": "",
"footer_content": "",
"footer_active": false,
"color_service": "",
"bar_button": null
},
"language": "de",
"bp_internal_number": ""
},
"available_languages": [
{
  "id": "1",
  "iso": "en",
  "name": "English"
},
{
  "id": "2",
  "iso": "es",
  "name": "Español"
},
{
  "id": "3",
  "iso": "de",
  "name": "Deutsch"
},
{
  "id": "4",
  "iso": "it",
  "name": "Italiano"
},
{
  "id": "5",
  "iso": "fr",
  "name": "Français"
}
],
"url_test":
"service?access_token=b2uOQxz60KYvgKhZkWeyfBIItJt2m1SW3yUy.
0sOY",
"available_services": [
{
  "id": "3",
  "name": "Dokumente",

```

```

"params": "",
"admin_url": "https://docman.livingservices.com/index.php/",
"type": "1",
"created_at": "2015-05-14 10:48:04",
"updated_at": "2016-07-27 07:10:38",
"image": "http://esp-
development.streamnow.ch/img/services/eccbc87e4b5ce2fe28308fd9f2a7baf3.png",
"webview_url": "https://docman.livingservices.com:8086/",
"doc_url": "https://docman.livingservices.com:8086/swagger/",
"active": "1",
"user_id": "910",
"admin_password": "c46f835677bfb4899117c3bf068e5b1a",
"admin": "69cff6c7ca7b779b8c285a633a783a16",
"managername": "Docman",
"secret_id": "WPnp4PCU",
"category_id": 12,
"device_type": "3",
"description": "",
"smart_image": "http://esp-
development.streamnow.ch/img/services/smart_eccbc87e4b5ce2fe28308fd9f2a7baf3.png",
"internal_use": "0",
"currency": "3",
"vat": "1.00",
"available_in_app": "1",
"in_wall": "1",
"in_wall_image": "http://esp-
development.streamnow.ch/img/services/inwall_eccbc87e4b5ce2fe28308fd9f2a7baf3.png",
"available_in_web": "0",
"api_url": "https://docman.livingservices.com:8086/",
"subscription": "0",
"address_specific": "0",
"usable": true
}
],
"categories": [
{
"id": "12",
"parent_id": null,
"name": "Wohninfos",
"image": "http://esp-
development.streamnow.ch/img/categories/c20ad4d76fe97759aa27a0c99bffa6710.png",
"active": "1",
"smart_image": "http://esp-
development.streamnow.ch/img/categories/smartc20ad4d76fe97759aa27a0c99bffa6710.png",

```

	<pre>"in_wall_image": "http://esp-development.streamnow.ch/img/categories/inwall_c20ad4d76fe97759aa7a0c99bff6710.png" } ] }</pre> <p><b>access_token</b> - all further API calls must have this key in header</p>
403	<pre>{"msg": "The credentials you entered did not match our records"}</pre>
400	<pre>{"msg": "Please provide username."}</pre>
400	<pre>{"msg": "Please provide password."}</pre>
401	<pre>{"error": "Incorrect username or password."}</pre>
500	<pre>{"error": "Something went wrong. Please try again later."}</pre>

### 3.2. login (using device uid)

This operation is used to authenticate the user with the system and obtain the auth\_token and all the user parameters. It requires device uid, which is unique and it is univocally assigned to a user in the MSP.

This applies only to end-users.

#### Request

Method	URL
<b>POST</b>	auth/login/

#### Parameters

Type	Name	Description	Type	Data Type
POST	uid	Device uid	form	string
POST	source	Source of the request	form	string

Where source could be:

- App
- Mobile
- Tablet



## Response

Status	Response
200	<p><b>Response will be an object containing the user information, the tokens as well as the business partner and services information.</b></p> <pre> {   "access_token": "CsmKJCmBBHOv445N6IwLzjRaQb4oA2nQ4MSJoYNf",   "refresh_token": "X3cRfJBbNw98zGV3o9c1M1qMF7z7wuozolxKKc5W",   "validUntil": "2016-08-12 17:39:05 CET",   "deviceSessionId": "0076491212d5be6c0636c227aaafccf54303c5f3",   "user_info": {     ...      "partner": {       ...      },     "language": "de",     "bp_internal_number": ""   },   "available_languages": [     ...    ],   "url_test": "service?access_token=CsmKJCmBBHOv445N6IwLzjRaQb4oA2nQ4MSJoYNf",   "available_services": [     {       ...      },     {       ...      }   ],   "categories": [     {       ...      },     {       ... </pre>

	<pre>     }   ] }</pre> <p>An example response is as in the previous case.</p> <p><b>access_token (string)</b> - all further API calls must have this key in header</p>
403	{"msg": "The credentials you entered did not match our records"}
400	{"msg": "Please provide username."}
400	{"msg": "Please provide password."}
401	{"error": "Incorrect username or password."}
500	{"error": "Something went wrong. Please try again later."}

### 3.3. logout

This operation is used to log out the user.

This API operation logs out the user that is currently invoking this operation from the system. It does not remove the associated token from the system, and hence other active sessions that use the same token will be not affected.

#### Request

Method	URL
GET	auth/logout/

#### Parameters

Type	Name	Description	Type	Data Type
N/A				

#### Response

Status	Response
200	{"msg": "Logout OK"}
200	{"msg": "Logout KO"}
500	{"error": "Something went wrong. Please try again later."}

### 3.4. reset

This operation is used to reset user password.

#### Request

Method	URL
POST	auth/reset/

#### Parameters

Type	Name	Description	Type	Data Type
POST	email	User identifier	form	string

#### Response

Status	Response
200	{"msg":"OK"}
403	{"msg":"KO"}
500	{"error":"Something went wrong. Please try again later."}

### 3.5. refreshToken

This API regenerate the access token without relogin.

#### Request

Method	URL
GET	/refreshToken

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	Refresh_token	User refresh token	query	string

## Response

Status	Response
200	{newtoken: ,newrefreshToken: , validUntil:}
400	{"error": "unauthorized_client"}
400	{"error": "The refresh token is invalid."}
500	{"error": "Something went wrong. Please try again later."}

## 3.6. revokeToken

This API operation logsout the user from the system.

This API operation logsouts the user from the system and removes the token that is currently used when invoking the operation from the system. Therefore, after invoking this operation, all requests that use this token will return an access\_denied error. Also, API invocations that do not use this token explicitly, but use a cookie associated to this token, will return an access\_denied error.

## Request

Method	URL
GET	/revokeToken/

## Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string

## Response

Status	Response
200	Access_denied: token is invalid
200	['msg' => 'The credentials you entered did not match our records']
400	{"error": "unauthorized_client"}
500	{"error": "Something went wrong. Please try again later."}

## 4. Comlite

Comlite is the VoIP Service. Services connected to the MSP replicate their APIs through the MSP. End users do not connect directly to service server.

### 4.1. getTerminals

This operation is used to get terminals from Comlite of current user.

#### Request

Method	URL
POST	comlite/getTerminals/

#### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	query	string

#### Response

Status	Response
200	{ "status":{"msg":"User not found"}}
200	{ "status": { "status": "ok", "terminals": [ { "id": "33", "mac": "aaaaaaaaaaaa", "model": "GXP1610" } ] } }
400	{ "error":"unauthorized_client"}
500	{ "error":"Something went wrong. Please try again later."}

## 5. Docman

Docman is a multi purpose service, intended as document management, support contact and events/notifications.

### 5.1. getContact

This operation is used to get the contact person (Client) associated to a user into the system.

#### Request

Method	URL
GET	/getContact

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string

#### Response

Status	Response
200	<pre>{   "status": "ok",   "contacts": [     {       "id": "15",       "fullname": "",       "role": "Bewirtschaftung",       "url": "",       "email": "info@wincasa.ch",       "telephone": "052 268 88 88",       "schedule": "Mo-Fr von 8:00 bis 20:00",       "company": "Wincasa AG",       "address": "Grüzefeldstrasse 41",       "zip": "8401",       "city": "Winterthur"     },     {       "id": "16",       "fullname": "",       "role": "IT Help",       "url": ""     }   ] }</pre>

```

"email": "support@exxo-services.ch",
"telephone": "0800 00 00 33",
"schedule": "Mo – Fr von 08:00 bis 20:00",
"company": "exxo services GmbH",
"address": "Rietbachstrasse 10",
"zip": "8952",
"city": "Schlieren"
},
{
  "id": "155",
  "fullname": "",
  "role": "Bewirtschaftung",
  "url": "",
  "email": "support@ps-ag.ch",
  "telephone": "0800 00 88 33",
  "schedule": "Mo-Fr von 8:00 Uhr bis 20:00 Uhr",
  "company": "Property Service AG",
  "address": "Zürcherstrasse 33",
  "zip": "8003",
  "city": "Schlieren"
},
{
  "id": "177",
  "fullname": "",
  "role": "Smart Home",
  "url": "http://www.smart-home-center.ch/",
  "email": "support@shc.ch",
  "telephone": "0800 00 11 11",
  "schedule": "Mo – Fr von 08:00 bis 18:00",
  "company": "dS Smart Home Center",
  "address": "Brandstrasse 33",
  "zip": "8952",
  "city": "Schlieren"
},
{
  "id": "200",
  "fullname": "",
  "role": "Entertainment",
  "url": "www.streamnow.ch",
  "email": "fiberpower@streamnow.ch",
  "telephone": "0800 00 99 33",
  "schedule": "Mo-Fr von 8:00 bis 20:00",
  "company": "streamnow - FiberPower",
  "address": "Brandstrasse 33",
  "zip": "8952",
  "city": "Schlieren"
}
]
}

```

400	{"error":"Unauthorized_client"}
500	{"error":"Something went wrong. Please try again later."}

## 5.2. getContactAvatar

This operation is used to get the contact person image associated to a user into the system.

### Request

Method	URL
GET	/getContactAvatar

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	ID	Contact ID	query	string

### Response

Status	Response
200	Response will be a jpeg
200	{"msg":"Contact not set","status":"error","code":-1}
200	{"msg":"not manager id","status":"error","code":-3}
400	{"error":"Unauthorized client"}
500	{"error":"Something went wrong. Please try again later."}

## 5.3. getDocInfo (POST)

This operation is used to get one document from one user.

### Request

Method	URL
POST	/getDocInfo



### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string
POST	Doc_id	Document identifier	Form	integer

### Response

Status	Response
200	Response will be a pdf
200	{ "msg": "doc_id not correct", "status": "error", "code": "-2" }
200	{ "msg": "file not found", "status": "error", "code": "-3" }
400	{ "error": "Unauthorized_client" }
500	{ "error": "Something went wrong. Please try again later." }

### 5.4. getDocInfo (GET)

This operation is used to get one document from one user.

### Request

Method	URL
GET	/getDocInfo

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	Doc_id	Document identifier	query	integer

## Response

Status	Response
200	Response will be a pdf
200	{ "msg": "doc_id not correct", "status": "error", "code": "-2" }
200	{ "msg": "file not found", "status": "error", "code": "-3" }
500	{ "error": "Something went wrong. Please try again later." }

## 5.5. getDocsInfo

This operation is used to get the documents from one user.

## Request

Method	URL
POST	/getDocsInfo

## Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string

## Response

Status	Response
200	{ "status": "ok", "userTree": [...], "repotree": [...] }
400	{ "error": "Unauthorized_client" }
500	{ "error": "Something went wrong. Please try again later." }

## 5.6. getImage

This operation is used to get associated image of a document or category from one user.

**Request**

Method	URL
GET	/getImage

**Parameters**

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	Item_id	Document/item identifier	query	integer

**Response**

Status	Response
200	[jpeg]
200	{ "status": "error", "msg": "item_id not correct", "code": -2 }
400	{ "error": "Unauthorized_client" }
500	{ "error": "Something went wrong. Please try again later." }

**5.7. getNotifications**

This operation is used to get the last 5 most recent notifications (events) from one user.

**Request**

Method	URL
POST	/getNotifications

**Parameters**

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string

## Response

Status	Response
200	<pre>{ "status": "ok",   "notifications": [...] }</pre>
400	<pre>{ "error": "Unauthorized_client" }</pre>
500	<pre>{ "error": "Something went wrong. Please try again later." }</pre>

## 6. Operations on Dashboard

### 6.1. addressCreate

This operation is used to create an address of an user

#### Request

Method	URL
POST	/address/create

#### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	query	string

#### Response

Status	Response
200	{ "data": { "updated_at": "2016-11-23 15:44:46", "created_at": "2016-11-23 15:44:46", "id": 270 } }
400	{"error": "unauthorized_client"}

### 6.2. addressDelete

This operation is used to delete an address of an user

#### Request

Method	URL
DELETE	/address/delete

### Parameters

Type	Name	Description	Type	Data Type
DELETE	Access_token	User token	query	string
DELETE	addressID	Address ID	query	string

### Response

Status	Response
200	{ "msg": "ok" }
400	{ "error": "unauthorized_client" }
404	{ "msg": "No address found" }

## 6.3. addressList

This operation returns the address of an user.

### Request

Method	URL
GET	/address/list

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	userID	User ID	query	string

## Response

Status	Response
200	<pre>{   "data": [     {       "id": "194",       "users_id": "1154",       "gender": "",       "company": "",       "firstname": "",       "lastname": "",       "letter": null,       "floor": "0",       "street_address": "street",       "street_number": "",       "suburb": "",       "postcode": "",       "city": "Aadorf",       "city_id": "37",       "state": "Thurgau",       "country_iso": "CH",       "type": "0",       "created_at": "2016-10-17 10:31:48",       "updated_at": "2016-10-17 10:31:48",       "lat": "47.5595986",       "lng": "7.5885761",       "apartment_number": "0",       "oto": "",       "building_name": "",       "name": "adress",       "mac": "",       "isg_ip": "",       "isg_mac": "",       "isg_mac_status": "0",       "isg_mac_status_date": null,       "smb1_mac": "",       "smb1_mac_status": "0",       "smb1_mac_status_date": null,       "smb2_mac": "",       "smb2_mac_status": "0",       "smb2_mac_status_date": null,       "smb3_mac": "",       "smb3_mac_status": "0",       "smb3_mac_status_date": null,       "ata_mac": "",       "ata_mac_status": "0",     }   ] }</pre>

	<pre> "ata_mac_status_date": null, "ata_mac_model": "0", "phone_mac": "", "phone_mac_status": "0", "phone_mac_status_date": null, "phone_mac_model": "0", "billing": "1", "installation": "1", "updated_isg_at": "30.11.-0001 00:00:00", "type_address": "0", "custom_type_address": "", "building_id": null, "size": null, "code": null, "addresses_status": null } ] } </pre>
400	{ "error": "unauthorized_client" }

#### 6.4. addressUpdate

This operation is used to update an existing address of an user

##### Request

Method	URL
POST	/address/update

##### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	query	string
POST	addressID	Address ID	query	string



### Response

Status	Response
200	{ "msg": "ok" }
400	{"error": "unauthorized_client"}
403	{"msg": "No address found "}

## 6.5. buildingList

This operation is used to return the buildings of a residence.

### Request

Method	URL
GET	/building/list

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User access token	query	string
GET	BPID	Id BP	query	String
GET	ResidenceNumber	Number of residence	query	String

### Response

Status	Response
0	No content

200	Response will be an object
500	{ "error": "Something went wrong. Please try again later." }

## 6.6. changePassword

This operation is used to change the password of current user.

### Request

Method	URL
POST	/changePassword

### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	Current access token	form	string
POST	Current_password	Current password used by user	Form	String
POST	New_password	New password that User want to set, must be different than the current password	Form	String
POST	New_password_repeat	Confirmation of new Password	Form	string

### Response

Status	Response
401	{ 'msg'=>'Missing parameters' }
403	{ 'msg'=>'Bad parameters' }

200	{'msg'=>'Ok'}
400	{"error":"unauthorized_client"}
500	{"error":"Something went wrong. Please try again later."}

### 6.7. checkHardware

It checks if the given hardware is registered in the MSP.

#### Request

Method	URL
GET	/checkHardware

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	Service app token	query	string
GET	Oto	OTO Value	query	string
GET	Isg_mac	ISG Mac Value	query	string

#### Response

Status	Response
200	{"msg":"OK"}
200	{"msg":"KO"}
400	{"error":"unauthorized_client"}
401	{"msg"=>"Missing parameters."}
500	{"error":"Something went wrong. Please try again later."}

### 6.8. Email send

It sends an email.

#### Request

Method	URL
<b>POST</b>	/email/send

#### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string
POST	Email	The valid destination email	Form	String
POST	Subject	The subject of email	Form	String
POST	Message	The message of email	Form	string

#### Response

Status	Response
200	{"status":"ok"}
400	{"error":"unauthorized_client"}
403	"Invalid email"
500	{"error":"Something went wrong. Please try again later."}

### 6.9. flatCreate

This operation is used to create a flat.

#### Request

Method	URL
<b>POST</b>	/flat/create

#### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User access token	query	string
POST	Building_id	ID of the building	query	string
POST	Country	Country	query	string

POST	state	State	query	string
POST	City	City name	query	string
POST	Street_name	Street name	query	string
POST	Street_number	Street number	query	string
POST	postcode	Postcode	query	string
POST	floor	Floor	query	integer
POST	Apartment_number	Apartment number	query	integer
POST	Letter	Apartment letter	query	string
POST	Code	BP ID	query	string
POST	Size	Size	query	integer

## Response

Status	Response
200	{'msg'=>'Missing parameters'}
400	{'error': 'unauthorized_client', 'error_description': 'The client is not authorized to request an access token using this method' }
404	{'msg'=>'No building found'}
404	{"msg": "No city found" }
500	{"error": "Something went wrong. Please try again later."}

## 6.10. flatDelete

This operation is used to delete a flat.

## Request

Method	URL
<b>DELETE</b>	/flat/delete

#### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	query	string
POST	flatID	ID of the flat	query	string

#### Response

Status	Response
200	
404	{"error": "No flat found"}
403	{"msg": "No address found "}

### 6.11. flatListAvailable

This operation is used to return list of available flats of a given Real Estate BP.

#### Request

Method	URL
<b>GET</b>	/flat/listAvailable

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	BPID	BP ID	query	string

#### Response

Status	Response
200	

404	<code>{"error": "No flat found"}</code>
400	<code>{'error': 'unauthorized_client', 'error_description': 'The client is not authorized to request an access token using this method' }</code>

## 6.12. flatListOccupied

This operation is used to return list of occupied flats of a given Real Estate BP.

### Request

Method	URL
GET	/flat/listOccupied

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	BPID	BP ID	query	string

### Response

Status	Response
200	
404	<code>{"error": "No flat found"}</code>
400	<code>{'error': 'unauthorized_client', 'error_description': 'The client is not authorized to request an access token using this method' }</code>

## 6.13. flatUpdate

This operation is used to update a flat.

### Request

Method	URL
POST	/flat/update

### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User access token	query	string
POST	flatID	ID of the flat	query	string
POST	Building_id	ID of the building	query	string
POST	Country	Country	query	string
POST	state	State	query	string
POST	City	City name	query	string
POST	Street_name	Street name	query	string
POST	Street_number	Street number	query	string
POST	postcode	Postcode	query	string
POST	floor	Floor	query	integer
POST	Apartment_number	Apartment number	query	integer
POST	Letter	Apartment letter	query	string
POST	Code	BP ID	query	string
POST	Size	Size	query	integer

## Response

Status	Response
200	{'msg'=>'Missing parameters'}
400	{'error': 'unauthorized_client', 'error_description': 'The client is not authorized to request an access token using this method' }
404	{'msg'=>'No building found'}
404	{"msg": "No city found" }



500	{"error": "Something went wrong. Please try again later."}
-----	--

#### 6.14. getActiveUsers

It returns the list of active users who belong to a BP

##### Request

Method	URL
POST	/getActiveUsers

##### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	query	string
POST	BPID	BP ID	query	String

##### Response

Status	Response
200	<p><b>Response will be an object containing information about active users. An example response is:</b></p> <pre>{   "data": [     {       "userID": "1066",       "name": "Hans SBB",       "surname": "Muster SBB",       "tel": "",       "email": "hans@sbb.ch"     },     {       "userID": "1144",       "name": "asdfasdf",       "surname": "asdf",       "tel": "1234",       "email": "javicule93@gmail.com"     }   ] }</pre>

	}
403	{"msg": "No user found"}
400	{"error": "unauthorized_client", "error_description": "The cliente is not authorized to request an access token using this method."}
500	{"error": "Something went wrong. Please try again later."}

### 6.15. getBasicUserInfo

It returns information about one user.

#### Request

Method	URL
GET	/getBasicUserInfo

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	UserID	User ID	query	string

#### Response

Status	Response
200	<p>Response will be an object containing info about the user</p> <pre>{   "data": {     "id": "955",     "name": "Jorge",     "lastname": "Sanchez",     "email": "jorge.sanchez@casacom.ch",     "gender": "male",     "username": "jorge",     "default_address_id": "9",     "telephone": "+41783338763",</pre>

```

"mobil": "+41783338763",
"fax": "",
"status": "1",
"image": "http://esp-
development.streamnow.ch/img/users/ef4e3b775c934dada217712d76f3
51f.jpg",
"demo": "0",
"payment_allowed": "",
"partner": {
  "company": "lindaufiber",
  "name": "Gemeinde Lindau",
  "phone": "+41 58 206 44 00",
  "email": "lindaufiber@streamnow.ch",
  "logo": "http://esp-
development.streamnow.ch/img/users/1ce927f875864094e3906a4a0b5e
e68.png",
  "background_image": "http://esp-
development.streamnow.ch/img/partner/background1ce927f875864094e
3906a4a0b5e68.png",
  "background_smartphone_image": "http://esp-
development.streamnow.ch/img/partner/background_smartphone1ce927
875864094e3906a4a0b5e68.png",
  "font_color_top": "rgba(255,255,255,1)",
  "color_top": "rgba(29,153,29,1)",
  "font_color_service": "rgba(251,251,251,1)",
  "font_color_bottom": "rgba(1,1,1,1)",
  "smartphone_app_name": "LINDAUFIBER",
  "line_color_smartphone": "rgba(27,121,67,1)",
  "font_color_smartphone": "rgba(255,255,255,1)",
  "background_color_smartphone": "rgba(29,153,29,1)",
  "background_color_wall_tablet": "rgba(35,35,35,1)",
  "background_color_icon_smartphone": "rgba(27,121,67,1)",
  "paid_services": true,
  "support_button": false,
  "url": "www.lindaufiber.ch",
  "footer_content": " ",
  "footer_active": true,
  "color_service": "rgba(29,153,29,1)",
  "bar_button": {
    "color": "",
    "text": "",
    "url": ""
  }
},
"language": "en",
"bp_internal_number": ""
}

```

403	<code>{"msg": "no user found"}</code>
400	<code>{"error": "unauthorized_client"}</code>
500	<code>{"error": "Something went wrong. Please try again later."}</code>

## 6.16. Get billing user info

It returns information about billing user's address.

### Request

Method	URL
GET	<code>/getBillingUserInfo</code>

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	UserID	User ID	query	string

### Response

Status	Response
200	<p>Response will be an object containing data address. An example response is:</p> <pre>{   "data": {     "name": "My Home",     "company": "",     "firstname": "Jorge",     "lastname": "Sanchez",     "street_address": "Markt",     "postcode": "",     "country_iso": "CH",     "state": "Basel Stadt",   } }</pre>

	<pre>"city": "Winterthur" } }</pre>
403	<pre>{"msg": "No user found"}</pre>
400	<pre>{"error": "unauthorized_client"}</pre>
404	<pre>{"msg": "No address found"}</pre>
500	<pre>{"error": "Something went wrong. Please try again later."}</pre>

### 6.17. getIP

This operation is used to get public IP address from one user. This IP address is the ISG public IP address.

This call is intended for service providers.

#### Request

Method	URL
GET	/getIP

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	Service app token	query	string
GET	UserID	User ID	query	string
GET	Isg_mac	MAC Address	query	string

#### Response

Status	Response
200	<pre>{   "ip": "8.8.8.8",   "msg": "ok" }</pre>
403	<pre>{"msg": "Site access denied"}</pre>

400	<code>{"error": "unauthorized client"}</code>
-----	---

## 6.18. getGroupsBP

It returns the list of groups which belong to a BP.

### Request

Method	URL
GET	/getGroupsBP

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	BPID	BP ID	query	string

### Response

Status	Response
200	<p>Response will be a object containing BP groups. An example response is:</p> <pre>{   "data": [     {       "id": "5",       "name": "test Jorge",       "users": [         {           "id": "896",           "name": "Mickey",           "lastname": "Mouse"         },         {           "id": "955",           "name": "Jorge",           "lastname": "Sanchez"         }       ]     }   ] }</pre>

	<pre>       },       {         "id": "958",         "name": "demo",         "lastname": "demo"       },       {         "id": "1097",         "name": "Fam.",         "lastname": "Muster VZUG"       },       {         "id": "1154",         "name": "Antonio2 José",         "lastname": "Jiménez"       }     ]   } } </pre>
403	{ "msg": "No group found" }

## 6.19. getGroupsUser

It returns the list of groups to which a user belongs to.

### Request

Method	URL
GET	/getGroupsUser

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	userID	User ID	query	string

### Response

Status	Response
200	Response will be an object containing information about groups to

	<p>which a user belongs to. An example response is:</p> <pre>{   "data": [     {       "id": "1",       "name": "Test user"     },     {       "id": "2",       "name": "Mitte"     }   ] }</pre>
400	<pre>{ "error": "unauthorized_client", "error_description": "The client is not authorized to request an access token using this method." }</pre>
403	<pre>{ "msg": "No group found" }</pre>

## 6.20. getInstallationUserInfo

It returns information about installation user's addresses.

### Request

Method	URL
GET	/getInstallationUserInfo

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	UserID	User ID	query	string

### Response

Status	Response
200	<p>Response will be an object containing user installation address.</p> <pre>{   "data": [</pre>



	<pre>{   "name": "My Home",   "company": "",   "firstname": "Jorge",   "lastname": "Sanchez",   "street_address": "Markt",   "postcode": "",   "country_iso": "CH",   "state": "Basel Stadt",   "city": "Winterthur" }</pre>
403	{ "msg": "No user found" }
400	{ "error": "unauthorized_client" }

## 6.21. getSIPInfo

It returns the SIP information (account) of a user.

### Request

Method	URL
GET	/getSIPInfo

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	UserID	User ID	query	string

### Response

Status	Response
--------	----------

## 6.22. getSMB

It returns SMB (Smart Media Boxes) boxes of a user.

### Request

Method	URL
GET	/getSMB

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	Service app token	query	string
GET	User_ID	User ID	query	string

### Response

Status	Response
200	<p>Response will be an object containing an array. An example response is:</p> <pre>{   "list": [     {       "mac": "11:11:22:33:44:99",       "activation": null,       "type": "smb1",       "status": 0     },     {       "mac": "11:11:22:33:44:77",       "activation": null,       "type": "smb2",       "status": 0     },     {       "mac": "11:11:22:33:44:88",       "activation": null,       "type": "smb3",       "status": 0     }   ] }</pre>

401	{"msg": "No SMB data found"}
400	{"msg": "unauthorized_client"}

### 6.23. getUserInfo

It returns information about the user currently logged in the system.

#### Request

Method	URL
GET	/getUserInfo

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string

#### Response

Status	Response
200	['msg' => 'The credentials you entered did not match our records']
200	{       "validUntil": "1970-01-01 01:00:00",       "deviceSessionId": "ek4kaaqq8tlabuovhl3hp5rvk4",       "user_info": {         "id": "1097",         "name": "Fam.",         "lastname": "Muster VZUG",         "email": "demo.vzug@streamnow.ch",         "gender": "male",         "username": "demo.vzug",         "default_address_id": null,         "telephone": "",         "mobil": "",         "fax": "",

```

"status": "1",
"image": "http://esp-
development.streamnow.ch/img/users/generic.jpg",
"demo": "1",
"payment_allowed": "",
"partner": {
  "company": "V-ZUG",
  "name": "V Zug",
  "phone": "",
  "email": "vzug@streamnow.ch",
  "logo": "http://esp-
development.streamnow.ch/img/users/4e2545f819e67f0615003dd7e04
a6087.jpeg",
  "background_image": "http://esp-
development.streamnow.ch/img/partner/background4e2545f819e67f06
15003dd7e04a6087.jpeg",
  "background_smartphone_image": "http://esp-
development.streamnow.ch/img/partner/background_smartphone4e25
45f819e67f0615003dd7e04a6087.jpeg",
  "font_color_top": "rgba(255,255,255,1)",
  "color_top": "rgba(40,59,154,1)",
  "font_color_service": "rgba(0,0,0,1)",
  "font_color_bottom": "rgba(0,0,0,1)",
  "smartphone_app_name": "",
  "line_color_smartphone": "rgba(0,0,0,1)",
  "font_color_smartphone": "rgba(0,0,0,1)",
  "background_color_smartphone": "rgba(40,59,154,1)",
  "background_color_wall_tablet": "",
  "background_color_icon_smartphone": "rgba(0,0,0,1)",
  "paid_services": false,
  "support_button": false,
  "url": "",
  "footer_content": "",
  "footer_active": false,
  "color_service": "",
  "bar_button": null
},
"language": "de",
"bp_internal_number": ""
},
"subscribed_services": [
  3
],

```

	<pre> "available_services": [ {   "id": "3",   "name": "Dokumente",   "params": "",   "admin_url": "https://docman.livingservices.com/index.php/",   "type": "1",   "created_at": "2015-05-14 10:48:04",   "updated_at": "2016-07-27 07:10:38",   "image": "http://esp- development.streamnow.ch/img/services/eccbc87e4b5ce2fe28308fd9f 2a7baf3.png",   "webview_url": "https://docman.livingservices.com:8086/",   "doc_url": "https://docman.livingservices.com:8086/swagger/",   "active": "1",   "user_id": "910",   "admin_password": "c46f835677bfb4899117c3bf068e5b1a",   "admin": "69cff6c7ca7b779b8c285a633a783a16",   "managename": "Docman",   "secret_id": "WPnp4PCU",   "category_id": 12,   "device_type": "3",   "description": "",   "smart_image": "http://esp- development.streamnow.ch/img/services/smart_eccbc87e4b5ce2fe283 08fd9f2a7baf3.png",   "internal_use": "0",   "currency": "3",   "vat": "1.00",   "available_in_app": "1",   "in_wall": "1",   "in_wall_image": "http://esp- development.streamnow.ch/img/services/inwall_eccbc87e4b5ce2fe283 08fd9f2a7baf3.png",   "available_in_web": "0",   "api_url": "https://docman.livingservices.com:8086/",   "subscription": "0",   "address_specific": "0",   "usable": true } ], "available_languages": [ { </pre>
--	--

```

    "id": "1",
    "iso": "en",
    "name": "English"
  },
  {
    "id": "2",
    "iso": "es",
    "name": "Español"
  },
  {
    "id": "3",
    "iso": "de",
    "name": "Deutsch"
  },
  {
    "id": "4",
    "iso": "it",
    "name": "Italiano"
  },
  {
    "id": "5",
    "iso": "fr",
    "name": "Français"
  }
]
}

```

#### 6.24. add-product (invoices)

Add a product to be added to next invoice. Service Provider must make the request.

##### Request

Method	URL
<b>PUT</b>	<a href="/invoices/add-product">/invoices/add-product</a>

##### Parameters

Type	Name	Description	Type	Data Type
PUT	Access_token	Service token	form	string
PUT	User_id	User ID from MSP	form	string

PUT	Service_id	Service identifier from MSP	Form	String
PUT	Description	Description of product	Form	String
PUT	Quantity	Quantity of units	Form	String
PUT	Price	Price per unit	Form	String
PUT	Discount	Discount	Form	string

### Response

Status	Response
200	{ "msg": "ok" }
403	{"msg": "site access denied"}
400	{"error": "unauthorized_client"}

### 6.25. getBillingStatus (invoices)

It returns the amount and currency of open status invoices by user in session.

### Request

Method	URL
GET	/invoices/getBillingStatus

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	form	string

### Response

Status	Response
200	Response will be an object containing amount and currency. An example response is:  { "amount":0,"currency":"CHF" }

400	{"error": "unauthorized_client"}
-----	----------------------------------

### 6.26. getInvoice (invoices)

This operation is used to return a list of invoices inside a period of months.

#### Request

Method	URL
POST	/getInvoice

#### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string
POST	User_id	User identifier	form	integer
POST	Invoice_id	Identifier of one invoice	form	integer

#### Response

Status	Response
200	
403	{'msg' => 'Site access denied'}
401	{'msg' => 'User_id field not found'}

### 6.27. getInvoices (invoices)

This operation is used to return a list of invoices inside a period of months.

#### Request

Method	URL
POST	/getInvoices

#### Parameters

Type	Name	Description	Type	Data Type
------	------	-------------	------	-----------



POST	Access_token	User token	form	string
POST	User_id	User identifier	form	integer
POST	Months	Quantity of last months to check	form	integer

### Response

Status	Response
200	[]
403	{'msg' => 'Site access denied'}
401	{'msg' => 'User_id field not found'}

## 6.28. residenceList

This operation is used to return the residences of a partner.

### Request

Method	URL
GET	/residence/list

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	BPID	BP ID	query	string

### Response

Status	Response
200	[]
400	{ "error": "unauthorized_client", "error_description": "The cliente is not authorized to request an access token using this method." }
500	{ "error": "Something went wrong. Please try again later." }

## 6.29. residenceListFull

This operation is used to return the residences and buildings of a partner.

### Request

Method	URL
GET	/residence/listFull

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	BPID	BP ID	query	string

### Response

Status	Response
200	[]
400	{"error": "unauthorized_client", "error_description": "The cliente is not authorized to request an access token using this method."}
500	{"error": "Something went wrong. Please try again later."}

## 6.30. sendPushGroup

This operation is used to send a push notification to user of a group.

### Request

Method	URL
GET	/sendPushGroup

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	groupID	Group ID	query	string

GET	Text	Text to be send to user	querys	string
-----	------	-------------------------	--------	--------

#### Response

Status	Response
200	[]
400	{"error":"unauthorized_client","error_description":"The cliente is not authorized to request an access token using this method."}
500	{"error":"Something went wrong. Please try again later."}

### 6.31. sendPushNotification

This API operation sends a Push Notification to user from the system.

#### Request

Method	URL
GET	/sendPushNotification

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	User_id	User ID	query	string
GET	Text	Text to be sent to user	Query	string

#### Response

Status	Response
200	Whoops, looks like something went wrong.
200	{"error":"error","error_description":"no members in this group."}
400	{"error":"unauthorized_client","error_description":"The cliente is not authorized to request an access token using this method."}

### 6.32. sendSMS

This API operation sends a SMS to user from the system.

#### Request

Method	URL
GET	/sendSMS

#### Parameters

Type	Name	Description	Type	Data Type
GET	access_token	User token	query	string
GET	User_ID	User ID	query	string
GET	Text	Text to be sent to user	Query	string

#### Response

Status	Response
200	{ "status": "ok", "msg": "1 sms prepared" }
200	['msg' => 'The credentials you entered did not match our records']
200	{ "status": "error", "msg": "empty mobil user" }
400	{ "error": "unauthorized_client" }

### 6.33. Services Disable

Disables a service.

#### Request

Method	URL
PUT	/services/disable

#### Parameters

Type	Name	Description	Type	Data Type
PUT	Access_token	User token	form	string
PUT	Service_ID	service ID	form	string

#### Response

Status	Response
200	{"msg":"Service disabled"}
400	{"error":"unauthorized_client"}
402	{"msg":"Service already disabled"}
404	{"msg":"Service not found"}

### 6.34. Services disable for user

Disable a user service.

#### Request

Method	URL
PUT	/services/disableforUser

#### Parameters

Type	Name	Description	Type	Data Type
PUT	Access_token	User token	form	string
PUT	Service_ID	Service ID	form	string
PUT	User_id	User to update	form	string

#### Response

Status	Response
200	{"msg":"ok"}
403	{"msg":"Site access denied"}
403	{"msg":"No user found"}
404	{"msg":"Service not found"}

### 6.35. Services enable

Activate a service.

#### Request

Method	URL
PUT	/services/enable

#### Parameters

Type	Name	Description	Type	Data Type
PUT	Access_token	User token	form	string
PUT	Service_ID	Service ID	form	string

#### Response

Status	Response
200	{"msg":"ok"}
402	{"msg":"Service already enabled"}
404	{"msg":"Service not found"}
400	{"error":"unauthorized_client"}

### 6.36. Services enable for user

Enable a user service.

### Request

Method	URL
PUT	/services/disableforUser

### Parameters

Type	Name	Description	Type	Data Type
PUT	Access_token	User token	form	string
PUT	Service_ID	Service ID	form	string
PUT	User_id	User to update	form	string

### Response

Status	Response
200	{"msg":"ok"}
403	{"msg":"Site access denied"}
404	{"msg":"Service not found"}
403	{"msg":"No user found"}

### 6.37. setDeviceToken

This API operation assigns a token of application and application ID to the user currently logged in the system.

### Request

Method	URL
GET	/setDeviceToken

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User APP token	query	string
GET	App_ID	Application ID	query	string
GET	Platform	Platform or Application used	Query	string
GET	DeviceToken	Device Token of Application	Query	string

Where platform could be:

- Android
- iOS

### Response

Status	Response
200	{"msg":"ok"}
400	{"error":"unauthorized_client"}

## 6.38. updateIP

This operation is used to update IP of an ISG

### Request

Method	URL
POST	/updateIP

### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	Service app token	form	string
POST	isg_ip	IP Address ISP	form	string
POST	Isg_mac	ISG Mac Value	Form	string

### Response



Status	Response
200	{"msg":"OK"}
400	{"error":"unauthorized_client"}
402	{"msg":"Invalid ISG MAC"}

### 6.39. updateMACstatus

This operation is used to update MAC Address and status of an ISG / VoIP device / SMB.

#### Request

Method	URL
POST	/updateMACstatus

#### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	Service app token	form	string
POST	mac	MAC Value	form	string
POST	status	Status	Form	String
POST	type	Type of MAC	Form	string

Where Status could be:

- 1
- 0

Where Type could be:

- Isg
- Ata

- Phone
- Smb1
- Smb2
- Smb3
- Smb4

#### Response

Status	Response
200	{"msg":"OK"}
403	{"msg":"MAC not found"}
400	{"error":"unauthorized_client"}

#### 6.40. Users active

Active a disabled end user.

#### Request

Method	URL
PUT	/users/active/{userID}

#### Parameters

Type	Name	Description	Type	Data Type
PUT	Access_token	User token	form	string
PUT	UserID	ID of user to active	path	string

#### Response

Status	Response
200	{"msg":"ok"}
403	{"msg":"User not allowed"}

404	{"msg":"Not found"}
-----	---------------------

#### 6.41. Users delete

Move user to pending deletion.

##### Request

Method	URL
PUT	/users/delete/{userID}

##### Parameters

Type	Name	Description	Type	Data Type
PUT	Access_token	User token	form	string
PUT	UserID	ID of user to delete	path	string

##### Response

Status	Response
200	{"msg":"ok"}
403	{"msg":"User not allowed"}
404	{"msg":"Not found"}

#### 6.42. Users disable

Disable an activated end user.

##### Request

Method	URL
PUT	/users/disable/{userID}

##### Parameters

Type	Name	Description	Type	Data Type
PUT	Access_token	User token	form	string
PUT	UserID	ID of user to active	path	string

### Response

Status	Response
200	{"msg":"ok"}
403	{"msg":"User not allowed"}
404	{"msg":"Not found"}

### 6.43. Register(users)

This operation is used to register a new user on the system.

### Request

Method	URL
POST	/users/register

### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	Current access token	form	string
POST	name	User Name	form	string
POST	Partner_id	Partner ID of the new user	Form	String
POST	Lastname	Last name of the user	form	String
POST	Username	Username for this user, must be unique	Form	String
POST	Gender	Gender of the user	Form	String
POST	Dob	Date of birth	Form	String
POST	Telephone	Telephone	Form	String

POST	Default_language	Default language for this user, must be a ISO CODE 2	Form	string
POST	Password	Password for the login must have at least 6 characters	Form	string
POST	Password_confirmation	Password confirmation must be the same as password	Form	String
POST	Email	Email for this user, must be unique	Form	String
POST	Fiscal_number	Fiscal number for this user	Form	String
POST	Status_control	Status for the user (Registered, Order, Waiting for Approval, Service Activation, Live), send number from 0 to 5	Form	string
POST	Email_template_id	ID from email templates list	Form	String
POST	Mobil	Mobile number	Form	String
POST	Fax	Fax number	Form	String
POST	Services	List of services ID to activate this user	Form	String
POST	Status	Active or not (1 or 0)	Form	String
POST	Demo	Demo or not (1 or 0)	Form	string

Where gender could be:

- Male
- Female

Where default language could be:

- De
- En
- Es

- Fr
- It

Where status could be:

- 1
- 0

Where demo could be:

- 1
- 0

### Response

Status	Response
200	{'msg'=>'Ok'}
401	{'msg'=>'Missing parameters'}
402	{'msg'=>'Bad parameters'}
403	{'msg'=>'invalid access'}

### 6.44. Update (users)

This operation is used to register a new user on the system.

### Request

Method	URL
<b>POST</b>	/users/register

### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	Current access token	form	string

POST	User_id	User to update	form	string
POST	name	User Name	form	string
POST	Partner_id	Partner ID of the new user	Form	String
POST	Lastname	Last name of the user	form	String
POST	Username	Username for this user, must be unique	Form	String
POST	Gender	Gender of the user	Form	String
POST	Dob	Date of birth	Form	String
POST	Telephone	Telephone	Form	String
POST	Default_language	Default language for this user, must be a ISO CODE 2	Form	string
POST	Password	Password for the login must have at least 6 characters	Form	string
POST	Password_confirmation	Password confirmation must be the same as password	Form	String
POST	Email	Email for this user, must be unique	Form	String
POST	Fiscal_number	Fiscal number for this user	Form	String
POST	Status_control	Status for the user (Registered, Order, Waiting for Approval, Service Activation, Live), send number from 0 to 5	Form	string
POST	Email_template_id	ID from email templates list	Form	String
POST	Mobil	Mobile number	Form	String
POST	Fax	Fax number	Form	String
POST	Services	List of services ID to activate this user	Form	String

POST	Status	Active or not (1 or 0)	Form	String
POST	Demo	Demo or not (1 or 0)	Form	string

Where gender could be:

- Male
- Female

Where default language could be:

- De
- En
- Es
- Fr
- It

Where status could be:

- 1
- 0

Where demo could be:

- 1
- 0

### Response

Status	Response
200	{'msg'=>'Ok'}
401	{'msg'=>'Missing parameters'}
402	{'msg'=>'Bad parameters'}
403	{'msg'=>'invalid access'}

### 6.45. Validate end user (users)

Validate credentials of user.



**Request**

Method	URL
POST	/users/validate-end-user

**Parameters**

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string
POST	Email	User email	form	string
POST	Password	Encrypted password by Crypter	Form	string

**Response**

Status	Response
200	{"status":"403"}
200	{"msg":"OK"}
400	{"error":"unauthorized_client"}

## 7. Operations on My Entertainment

### 7.1. getContact (myentertainment)

This operation is used to get the contact person (Client) associated to a user into the system.

#### Request

Method	URL
GET	/myentertainment/getContact

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string

#### Response

Status	Response
200	<p>Response will be an object. An example response is:</p> <pre>{   "status": {     "status": "ok",     "contact": {       "fullname": "Balz Halter",       "url": "http://www.halter.ch",       "email": "balz.halter@halter.ch",       "telephone": "+41 44 434 27 74",       "schedule": "Monday to Friday from 8:00 to 17:00 hours.",       "company": "Halter AG",       "address": "Hardturmstrasse 134",       "zip": "CH-8005",       "city": "Zürich"     }   } }</pre>
400	<pre>{"error": "unauthorized_client"}</pre>

### 7.2. getContactAvatar (my entertainment)

This operation is used to get the contact person image associated to a user into the system.

#### Request

Method	URL
GET	/myentertainment/getContactAvatar

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string

#### Response

Status	Response
200	Response will be a jpeg
400	{"error": "unauthorized_client"}

### 7.3. getCredentials (myentertainment)

This operation is used to get a user iptv credentials (username and password).

#### Request

Method	URL
GET	/myentertainment/getCredentials

#### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	query	string

#### Response

Status	Response
200	Response will be an object.

	<p>An example response is:</p> <pre>{   "status": {     "status": "ok",     "credentials": {       "username": "app",       "password": "app"     }   } }</pre>
400	<pre>{ "error": "unauthorized_client" }</pre>

## 8. OPERATIONS ON MY RESTAURANT

### 8.1. getRestaurant

It returns information about the restaurants near the place provided and keywords used.

#### Request

Method	URL
GET	/getRestaurant

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	Location	Location	query	string
GET	Keywords	Keywords	Query	string

#### Response

Status	Response
200	<pre>{   "params": {     "Suchbegriff": "spanisch",     "definierter_Kartenausschnitt": {},     "definierter_Ortsbegriff": "NULL",     "erkannter_Ortsname": "Basel",     "verwertete_Keywords": {       "key_0": "spanisch"     },     "keyword_array": {       "key_0": "spanisch"     },     "location_ids": "3141, 709, 845, 255, 240, 244, 245, 252, 253, 254, 258, 259, 260, 261, 262, 269, 273, 274, 1300, 277, 281, 283, 329, 306, 312, 313, 326, 328, 330, 331, 332, 333, 344, 345, 889, 396, 408, 409, 12856, 565, 569, 574, 590, 613, 614, 615, 616, 624, 633, 10908, 639, 641, 12752, 650, 724, 728, 730, 731, 732, 737, 749, 751, 761, 849, 792, 793, 794, 795, 797, 801, 831, 832, 833, 885, 888, 892, 930, 931, 961, 962, 973, 1512, 1042, 1064, 1043, 1044, 1045, 1190, 1191, 12953, 1373 1301, 1302, 1303, 1304, 1305, 1306, 1307, 1308, 1309, 1310, 1311,</pre>

```

1312, 1313, 1314, 1315, 1316, 1317, 1318, 1319, 1320, 1321, 1322,
1323, 1325, 1503, 1504, 1326, 1327, 1328, 1329, 1330, 1331, 1332,
1333, 1334, 1335, 1336, 1337, 1338, 1339, 1340, 1341, 1342, 1343,
1344, 1345, 1346, 1347, 1348, 1349, 1350, 1351, 1352, 1353, 1354,
1355, 1356, 1357, 1358, 1359, 1360, 1361, 1362, 1363, 1364, 1365,
1366, 1367, 1431, 1432, 1433, 1434, 1435, 1436, 1437, 1438, 1439,
1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449, 1450,
1451, 1452, 1453, 1454, 1455, 1456, 1457, 1480, 1481, 1482, 1483,
1484, 1485, 1468, 1469, 1470, 1471, 1472, 1473, 1474, 1475, 1476,
1486, 1487, 1489, 1490, 1491, 1492, 1494, 1495, 1496, 1497, 1498,
1500, 1501, 1502, 1506, 1507, 1508, 1513, 1514, 1515, 1516, 1517,
1518, 1519, 1520, 1521, 1522, 1523, 1524, 1525, 1526, 1527, 1528,
1529, 1530, 1531, 1532, 1533, 1534, 1535, 1536, 1537, 1538, 1539,
1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549, 1550,
1551, 1552, 1553, 1554, 1555, 1556, 1557, 1558, 1559, 1560, 1561,
1562, 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1570, 1571, 1572,
1573, 1574, 1575, 1576, 1577, 1578, 1579, 1580, 1581, 1582, 1583,
1584, 1585, 1586, 1587, 1588, 1589, 1590, 1591, 1592, 1597, 1598,
1599, 1635, 1636, 1649, 1645, 1648, 1646, 1647, 1650, 3013, 3014,
3152, 3036, 3022, 3023, 3025, 3029, 3030, 3031, 3032, 3037, 3038,
3039, 3040, 3041, 3151, 3045, 3046, 3047, 3048, 3049, 3050, 3051,
3052, 3053, 3054, 3055, 3056, 3057, 3058, 3059, 3060, 3061, 3062,
3063, 3064, 3065, 3066, 3067, 3068, 3069, 3070, 3071, 3072, 3073,
3074, 3075, 3076, 3077, 3078, 3079, 3080, 3081, 3082, 3083, 3084,
3085, 3086, 3087, 3088, 3089, 3090, 3091, 3092, 3093, 3094, 3095,
3096, 3097, 3098, 3099, 3100, 3101, 3102, 3103, 3104, 3105, 3106,
3107, 3108, 3109, 3110, 3111, 3112, 3113, 3114, 3115, 3116, 3117,
3118, 3119, 3120, 3121, 3122, 3123, 3124, 3125, 3126, 3127, 3128,
3129, 3130, 3131, 3132, 3133, 3134, 3135, 3136, 3137, 3138, 3139,
3140, 3184, 3145, 3146, 3147, 3148, 3149, 3150, 3153, 3157, 3158,
3159, 3167, 3168, 3169, 3170, 3173, 10961, 3176, 3177, 3178, 3179,
3180, 3183, 3186, 3187, 3216, 3241, 3257, 3258, 12559, 8883, 8884,
10634, 10960, 10681, 10732, 10778, 10779, 10793, 10798, 10839,
10975, 11013, 11014, 11028, 11058, 11065, 11066, 12530, 12554,
12556, 12557, 12558, 12561, 12563, 12611, 12612, 12614, 12649,
12708, 12895, 12946, 12994, 12997",
"keyword_ids": "91, 132, 390, 826, 316, 458, 462, 507, 521, 526, 534,
578, 624, 740, 739, 748, 788, 810, 843, 858, 1066, 3206, 1287, 1254,
1626, 3021, 3083, 12759, 3736, 4217, 4405, 4576, 5176, 9330, 9572,
9956, 10793, 10861, 11834, 12325, 6, 83, 301, 334, 364, 380, 513, 585,
655, 672, 738, 741, 747, 764, 817, 1220, 1415, 1457, 1557, 1971, 2560,
3112, 3147, 10817, 12599, 284, 572, 988",
"search_ids": "624,3083,10793,1457,1557,3112,3147",
"search_ids_ext": "624, 3083, 10793, 1457, 1557, 3112"
},
"restaurants": {
"key_0": {
"id": "624",
"url_name": "don-pincho",

```

	<pre> "real_name": "Don Pincho", "phone": "061 322 10 60", "email": "mon@gmx.ch", "homepage": "www.donpincho.ch", "delivery_url": {}, "address": "St. Johannis-Vorstadt 58", "zip": "4056", "city": "Basel", "gaultmillau": {}, "coupon_url": {}, "latitude": "47.564601898193", "longitude": "7.5839099884033", "user_tags_total": "0", "user_comments_total": "0", "ronorp_id": "214852", "city_district": {}, "distance": "5339709", "has_menus": {}, "has_spec": "2", "first_pano": "613", "first_image": "1288171784.jpg", "votes": "0", "rating": {}, "has_image": "1", "has_booking": "0", "booking_url": {}, "is_customer": "1", "profile_rating": "53", "friend_votes": {}, "friend_rating": {}, "short_name": "Don Pincho", "speciality": {   "key_0": {     "title": "Tägliche Mittagsmenus",     "line2": "mit Fleisch-, Fisch-, Vegetarisch",     "line3": "ab",     "category": "UNDEFINED",     "tags": {},     "price": "20.00",     "short_title": "Tägliche Mittagsmenus"   },   "key_1": {     "title": "Am Abend feinste Spanische Küche",     "line2": "à la carte oder unsere reichhaltigen",     "line3": "Tapas, Pinchos und Paellas",     "category": "UNDEFINED",     "tags": {},     "price": {},     "short_title": "Am Abend feinste Spanische Küche"   } } </pre>
--	---

	<pre>     }   },   "img_medium": "https://s3-eu-west- 1.amazonaws.com/pano.lunchgate.ch/613/500x200.jpg",   "img": "https://s3-eu-west- 1.amazonaws.com/pano.lunchgate.ch/613/300x100.jpg",   "avatar": "https://s3-eu-west- 1.amazonaws.com/pano.lunchgate.ch/613/100x100.jpg",   "restaurantUrl": "https://www.lunchgate.ch/restaurant/don-pincho",   "restaurantPath": "/restaurant/don-pincho",   "subscribeUrl": "https://backend.lunchgate.ch/cockpit/subscribe/source/partner/restaurant /624"   },   "key_1": {     "id": "10793",     "url_name": "la-terrazza",     "real_name": "La Terraza",     "phone": "061 543 03 12",     "email": "info@tapeo.ch",     "homepage": "www.tapeo.ch",     "delivery_url": {},     "address": "Haltingerstrasse 104",     "zip": "4057",     "city": "Basel",     "gaultmillau": {},     "coupon_url": {},     "latitude": "47.5664699",     "longitude": "7.5974317",     "user_tags_total": "0",     "user_comments_total": "0",     "ronorp_id": {},     "city_district": {},     "distance": "5340094",     "has_menus": {},     "has_spec": {},     "first_pano": "3445",     "first_image": "1395386249.jpg",     "votes": {},     "rating": {},     "has_image": "1",     "has_booking": "1",     "booking_url": "https://foratable.com/reserve/restaurant/aa35213b9b6ea5514bef7a9ce06 b8307/",     "is_customer": "1",     "profile_rating": "47",     "friend_votes": {},     "friend_rating": {}, </pre>
--	--



	<pre> "short_name": "La Terraza", "img_medium": "https://s3-eu-west- 1.amazonaws.com/pano.lunchgate.ch/3445/500x200.jpg", "img": "https://s3-eu-west- 1.amazonaws.com/pano.lunchgate.ch/3445/300x100.jpg", "avatar": "https://s3-eu-west- 1.amazonaws.com/pano.lunchgate.ch/3445/100x100.jpg", "restaurantUrl": "https://www.lunchgate.ch/restaurant/la-terrazza", "restaurantPath": "/restaurant/la-terrazza", "subscribeUrl": "https://backend.lunchgate.ch/cockpit/subscribe/source/partner/restaurant /10793" }, "key_2": { "id": "3083", "url_name": "bodega-espana", "real_name": "Bodega Espana", "phone": "061 261 11 24", "email": "bodega-espana@bluewin.ch", "homepage": "bodega-espana.ch", "delivery_url": {}, "address": "Heuberg 4", "zip": "4051", "city": "Basel", "gaultmillau": {}, "coupon_url": {}, "latitude": "47.55659866333", "longitude": "7.5851001739502", "user_tags_total": "0", "user_comments_total": "0", "ronorp_id": "236234", "city_district": {}, "distance": "5338849", "has_menus": {}, "has_spec": "1", "first_pano": "3293", "first_image": {}, "votes": {}, "rating": {}, "has_image": "1", "has_booking": "0", "booking_url": {}, "is_customer": "1", "profile_rating": "41", "friend_votes": {}, "friend_rating": {}, "short_name": "Bodega Espana", "speciality": { "key_0": { </pre>
--	--

	<pre>         "title": "Pulpo a al Gallega mit Paprikakartoffeln",         "line2": {},         "line3": {},         "category": "FISH",         "tags": "Fisch",         "price": "24.00",         "short_title": "Pulpo a al Gallega mit ..."       }     },     "img_medium": "https://s3-eu-west- 1.amazonaws.com/pano.lunchgate.ch/3293/500x200.jpg",     "img": "https://s3-eu-west- 1.amazonaws.com/pano.lunchgate.ch/3293/300x100.jpg",     "avatar": "https://s3-eu-west- 1.amazonaws.com/pano.lunchgate.ch/3293/100x100.jpg",     "restaurantUrl": "https://www.lunchgate.ch/restaurant/bodega- espana",     "restaurantPath": "/restaurant/bodega-espana",     "subscribeUrl": "https://backend.lunchgate.ch/cockpit/subscribe/source/partner/restaurant /3083"   },   "key_3": {     "id": "1557",     "url_name": "carrasco-tapas-bar",     "real_name": "Carrasco Tapas Bar",     "phone": "061 681 81 56",     "email": "tapas-bar@carrasco.ch",     "homepage": "www.carrasco.ch",     "delivery_url": {},     "address": "Efringerstrasse 1",     "zip": "4057",     "city": "Basel",     "gaultmillau": {},     "coupon_url": {},     "latitude": "47.567199707031",     "longitude": "7.5964398384094",     "user_tags_total": "0",     "user_comments_total": "0",     "ronorp_id": "236655",     "city_district": {},     "distance": "5340160",     "has_menus": {},     "has_spec": {},     "first_pano": "0",     "first_image": "1349367918.jpg",     "votes": "0",     "rating": {},     "has_image": "1", </pre>
--	---

```

"has_booking": "0",
"booking_url": {},
"is_customer": "1",
"profile_rating": "6",
"friend_votes": {},
"friend_rating": {},
"short_name": "Carrasco Tapas Bar",
"img_medium":
"https://files.lunchgate.ch/gastrouploads/1557/500x200.jpg",
"img": "https://files.lunchgate.ch/gastrouploads/1557/300x100.jpg",
"avatar":
"https://files.lunchgate.ch/gastrouploads/1557/100x100.jpg",
"restaurantUrl": "https://www.lunchgate.ch/restaurant/carrasco-tapas
bar",
"restaurantPath": "/restaurant/carrasco-tapas-bar",
"subscribeUrl":
"https://backend.lunchgate.ch/cockpit/subscribe/source/partner/restaurant
/1557"
},
"key_4": {
"id": "1457",
"url_name": "spalenburg-tapas-del-mar",
"real_name": "Spalenburg - Tapas del Mar",
"phone": "061 261 99 34",
"email": "spalenburg@tapasdelmar.com",
"homepage": "www.tapasdelmar.com",
"delivery_url": {},
"address": "Schnabelgasse 2",
"zip": "4051",
"city": "Basel",
"gaultmillau": {},
"coupon_url": {},
"latitude": "47.556999206543",
"longitude": "7.5862798690796",
"user_tags_total": "0",
"user_comments_total": "0",
"ronorp_id": {},
"city_district": {},
"distance": "5338909",
"has_menus": {},
"has_spec": {},
"first_pano": "0",
"first_image": "1349569403.jpg",
"votes": {},
"rating": {},
"has_image": "1",
"has_booking": "0",
"booking_url": {},
"is_customer": "1",

```

```

"profile_rating": "6",
"friend_votes": {},
"friend_rating": {},
"short_name": "Spalenburg - Tapas del ...",
"img_medium":
"https://files.lunchgate.ch/gastrouploads/1457/500x200.jpg",
"img": "https://files.lunchgate.ch/gastrouploads/1457/300x100.jpg",
"avatar":
"https://files.lunchgate.ch/gastrouploads/1457/100x100.jpg",
"restaurantUrl": "https://www.lunchgate.ch/restaurant/spalenburg-
tapas-del-mar",
"restaurantPath": "/restaurant/spalenburg-tapas-del-mar",
"subscribeUrl":
"https://backend.lunchgate.ch/cockpit/subscribe/source/partner/restaurant
/1457"
},
"key_5": {
"id": "3112",
"url_name": "tapadera",
"real_name": "Tapadera",
"phone": "061 281 55 11",
"email": "info@tapadera.ch",
"homepage": "www.tapadera.ch",
"delivery_url": {},
"address": "Innere Margarethenstrasse 28",
"zip": "4051",
"city": "Basel",
"gaultmillau": {},
"coupon_url": {},
"latitude": "47.54940032959",
"longitude": "7.5861201286316",
"user_tags_total": "0",
"user_comments_total": "0",
"ronorp_id": "236083",
"city_district": {},
"distance": "5338075",
"has_menus": {},
"has_spec": {},
"first_pano": "0",
"first_image": "1359170569.jpg",
"votes": {},
"rating": {},
"has_image": "1",
"has_booking": "0",
"booking_url": {},
"is_customer": "1",
"profile_rating": "6",
"friend_votes": {},
"friend_rating": {},

```

	<pre> "short_name": "Tapadera", "img_medium": "https://files.lunchgate.ch/gastrouploads/3112/500x200.jpg", "img": "https://files.lunchgate.ch/gastrouploads/3112/300x100.jpg", "avatar": "https://files.lunchgate.ch/gastrouploads/3112/100x100.jpg", "restaurantUrl": "https://www.lunchgate.ch/restaurant/tapadera", "restaurantPath": "/restaurant/tapadera", "subscribeUrl": "https://backend.lunchgate.ch/cockpit/subscribe/source/partner/restaurant/3112" } }, "tags": "Aussicht,Bankett,Café,Events,Festarrangements,Garten,Klassisch,Mediterran,Mexikanisch,Romantisch,Spanisch,Terrasse,Tierfreundlich,Trendy,WLAN Zugang gratis", "locations": "Basel", "hashTags": { "key_0": "2a5ee7393362aa4e6cfbc3b2db45afbe", "key_1": "a7e7b3768e40f44875b934f826a58bc0", "key_2": "2759b3037517baaa35c9f684ce7827cf", "key_3": "34833f384457d7550f725464f40b818a", "key_4": "afac82b2aa127a886d5d5c3b9153b45a", "key_5": "4138f403daf4af74a2dd71c7de66205e", "key_6": "0245ca85665b7be30ddbaadbaab7af5d", "key_7": "0d6a8360d8eeb549a654258d20c4e577", "key_8": "1c88decc175c0a6d71206fceab47b909", "key_9": "94b69a55906937f6ddfff76b09099ed5", "key_10": "e7c22eb3059026ba691da7e8117e2801", "key_11": "25f6193164176e775104bfa5e9119c29", "key_12": "283d0a2c9929b650577959fa6db851cd", "key_13": "5aa6599b9067d80a82c0acc6ad8d32fd", "key_14": "faa5d4facd6369c02d516587582cc7f7", "key_15": "7ce7ff139dcd4d1028143690d2c491a4" }, "requested_map": {}, "resulting_map": "47.540500640869,7.577149132605,47.576099395752,7.6041925557983", "numResults": "6", "id_list": "624,10793,3083,1557,1457,3112", "numTags": "15", "numLocations": "1", "numHashes": "16", "my_lat": {}, "my_long": {}, "meta_title": "\"spanisch\" in \"basel\"   Restaurants neu entdecken", "meta_description": "Sie suchen das passende Restaurant? Entdecken </pre>
--	--

	Sie Restaurants zu \"spanisch\" in \"basel\" und unzählige weitere Restaurants, Bars und Cafés auf Lunchgate. "searchTips": { "key_0": { "label": "Finde mehr Restaurants in <em>basel</em>", "url": "search.php?location=basel", "url_cms": "/restaurants/basel" } }, "status": "success" }
401	Bad parameters

## 8.2. getRestaurantMenu

It returns information about the restaurants near the place provided and keywords used.

### Request

Method	URL
GET	/getRestaurantMenu

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	Restaurant_ID	Restaurant ID got on GetRestaurant	query	string
GET	Day	Number of the day, default is 0	Query	String
GET	Limit	Limit data, default is 10	Query	string

## Response

Status	Response
200	<pre>{   "timestamp": "1471191090",   "date": "14.08.2016",   "menu": {},   "speciality": {     "key_0": {       "title": "Tägliche Mittagsmenus",       "line2": "mit Fleisch-, Fisch-, Vegetarisch",       "line3": "ab",       "category": "UNDEFINED",       "tags": {},       "price": "20.00",       "short_title": "Tägliche Mittagsmenus"     },     "key_1": {       "title": "Am Abend feinste Spanische Küche",       "line2": "à la carte oder unsere reichhaltigen",       "line3": "Tapas, Pinchos und Paellas",       "category": "UNDEFINED",       "tags": {},       "price": {},       "short_title": "Am Abend feinste Spanische Küche"     }   },   "status": "success" }</pre> JSON DATA
401	Bad parameters

### 8.3. getRestaurantProfile

It returns information about the restaurant selected.

## Request

Method	URL
GET	/getRestaurantProfile

## Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	Restaurant_ID	Restaurant ID got on GetRestaurant	query	string

## Response

Status	Response
200	<pre>{   "uid": "624",   "url_name": "don-pincho",   "real_name": "Don Pincho",   "address": "St. Johannis-Vorstadt 58",   "zip": "4056",   "city": "Basel",   "phone": "061 322 10 60",   "email": "mon@gmx.ch",   "homepage": "www.donpincho.ch",   "delivery_url": {},   "seats_in": "80",   "seats_out": "30",   "premium": "Direkt am Rheinufer, genussvoll Spanisch Essen",   "smoke": "2",   "gaultmillau": {},   "country": "CH",   "lon": "7.5839099884033",   "lat": "47.564601898193",   "menu_template": "27",   "fb_url": {},   "fb_id": {},   "twitteraccount": {},   "socialmedia_status": "0",   "showlive": "1",   "first_pano": "613",   "first_image": "1288171784.jpg",   "has_image": "1",   "has_booking": "0",   "booking_url": {},   "is_customer": "1",   "city_district": {},   "booking_id": "624",   "pano_embedding_name": "don-pincho",   "hours": {</pre>



```

"current_day": "day7",
"current_time": "18:12",
"current_status": "offen",
"status_end": "00:00",
"day1": {
  "open1": "08:00–01:00"
},
"day2": {
  "open1": "08:00–01:00"
},
"day3": {
  "open1": "08:00–01:00"
},
"day4": {
  "open1": "08:00–01:00"
},
"day5": {
  "open1": "08:00–03:00"
},
"day6": {
  "open1": "18:00–03:00"
},
"day7": {
  "open1": "18:00–00:00"
},
"holiday": {}
},
"typology": {
  "key_0": "Spanisch",
  "key_1": "Mediterran"
},
"preparation": {},
"ambiance": {
  "key_0": "Romantisch",
  "key_1": "Trendy"
},
"paymentMethods": {
  "key_0": {
    "title": "American Express",
    "image_filename": "american_express.png",
    "icon":
      "https://backend.lunchgate.ch/pics/icons/payment_method/american_express.png"
  },
  "key_1": {
    "title": "EC",
    "image_filename": "ec.png",
    "icon":
      "https://backend.lunchgate.ch/pics/icons/payment_method/ec.png"
  }
}

```

```

    },
    "key_2": {
      "title": "Mastercard",
      "image_filename": "mastercard.png",
      "icon":
"https://backend.lunchgate.ch/pics/icons/payment_method/mastercard.png"
    }
  },
  "services": {
    "key_0": "Take Away",
    "key_1": "Aussicht",
    "key_2": "Café",
    "key_3": "Events",
    "key_4": "Festarrangements",
    "key_5": "Tapas Bar",
    "key_6": "Snacks",
    "key_7": "Bistro"
  },
  "avatar": "https://s3-eu-west-
1.amazonaws.com/pano.lunchgate.ch/613/100x100.jpg",
  "img": "https://s3-eu-west-
1.amazonaws.com/pano.lunchgate.ch/613/300x100.jpg",
  "images": {
    "key_0":
"https://files.lunchgate.ch/gastrouploads/624/c_1288171784.jpg",
    "key_1":
"https://files.lunchgate.ch/gastrouploads/624/c_1286281964.jpg",
    "key_2":
"https://files.lunchgate.ch/gastrouploads/624/c_1288173325.jpg",
    "key_3":
"https://files.lunchgate.ch/gastrouploads/624/c_1288173348.jpg",
    "key_4":
"https://files.lunchgate.ch/gastrouploads/624/c_1288173360.jpg"
  },
  "panos": {
    "key_0": {
      "id": "613",
      "type": "Restaurant",
      "name": "Restaurant"
    },
    "key_1": {
      "id": "888",
      "type": "Restaurant",
      "name": "Restaurant"
    }
  },
  "labels": {},
  "reviews": {},

```

	<pre> "bugReportUrl": "https://backend.lunchgate.ch/cockpit/bugreport/restaurant/624", "restaurantUrl": "https://www.lunchgate.ch/restaurant/don-pincho", "image_large": "https://s3-eu-west- 1.amazonaws.com/pano.lunchgate.ch/613/940x350.jpg", "subscribeUrl": "https://backend.lunchgate.ch/cockpit/subscribe/source/profile/restaurant/ 624", "panoEmbedUrl": "https://plugins.lunchgate.ch/pano/embed.php?name=don-pincho", "feedbackUrl": {}, "menucard": {}, "winecard": {}, "kidscard": {}, "extracard": {}, "extracard2": {}, "timestamp": "1471191163", "date": "14.08.2016", "menu": {}, "speciality": { "key_0": { "title": "Tägliche Mittagsmenus", "line2": "mit Fleisch-, Fisch-, Vegetarisch", "line3": "ab", "category": "UNDEFINED", "tags": {}, "price": "20.00", "short_title": "Tägliche Mittagsmenus" }, "key_1": { "title": "Am Abend feinste Spanische Küche", "line2": "à la carte oder unsere reichhaltigen", "line3": "Tapas, Pinchos und Paellas", "category": "UNDEFINED", "tags": {}, "price": {}, "short_title": "Am Abend feinste Spanische Küche" } }, "menu_calendar": {}, "status": "success" } </pre>
401	Bad parameters

## 9. OPERATIONS ON MY VPN

### 9.1. getCamsByUser

This operation is used to get all video cameras of one user.

#### Request

Method	URL
GET	/myvpn/getCamsByUser

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string

#### Response

Status	Response
200	JSON DATA
200	"status":{ "status":"error","code":"-2","msg":"id not found" }
400	{ "error":"unauthorized_client" }

### 9.2. getCertificates

This operation is used to get certificates for a user's device.

#### Request

Method	URL
POST	/myvpn/getCertificates

### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string
POST	Device	Device unique identifier	form	string

### Response

Status	Response
200	JSON DATA
200	"status":{ "status":"error","code":"-2","msg":"id not found" }
400	{ "error":"unauthorized_client" }

### 9.3. getContact

This operation is used to get the contact person (Client) associated to a user into the system.

### Request

Method	URL
GET	/myvpn/getContact

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string

## Response

Status	Response
200	<p>Response will be an object. An example response is:</p> <pre>{   "status": {     "status": "ok",     "contact": {       "fullname": "Miguel Morillo",       "url": "",       "email": "miguel.morillo@gmail.com",       "telephone": "",       "schedule": "",       "company": "",       "address": "",       "zip": "",       "city": ""     }   } }</pre>
400	<pre>{"error": "unauthorized_client"}</pre>

## 9.4. getContactAvatar

This operation is used to get the contact person image associated to a user into the system.

## Request

Method	URL
GET	/myvpn/getContactAvatar

## Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string

## Response

Status	Response
200	Response will be a jpeg
400	{ "error": "unauthorized_client" }

## 9.5. setCamServiceByUser

This operation is used to activate video cameras service of one user.

## Request

Method	URL
POST	/myvpn/setCamServiceByUser

## Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string

## Response

Status	Response
200	JSON DATA
200	{ "status": { "status": "error", "code": "-2", "msg": "id not found" } }
400	{ "error": "unauthorized_client" }

## 9.6. unsetCamServiceByUser

This operation is used to deactivate video cameras service of one user.

## Request

Method	URL
POST	/myvpn/unsetCamServiceByUser

### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string

### Response

Status	Response
200	JSON DATA
200	"status":{"status":"error","code":"-2","msg":"id not found"}
400	{"error":"unauthorized_client"}



## 10. Operation on SW Docman

Docman is a multi purpose service, intended as document management, support contact and events/notifications.

### 10.1. getContact

This operation is used to get the contact person (Client) associated to a user into the system.

#### Request

Method	URL
GET	/sw/getContact

#### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string

#### Response

Status	Response
200	<pre>{   "status": "ok",   "contacts": [     {       "id": "15",       "fullname": "",       "role": "Bewirtschaftung",       "url": "",       "email": "info@wincasa.ch",       "telephone": "052 268 88 88",       "schedule": "Mo-Fr von 8:00 bis 20:00",       "company": "Wincasa AG",       "address": "Grüzefeldstrasse 41",       "zip": "8401",       "city": "Winterthur"     },     {       "id": "16",       "fullname": "",       "role": "IT Help",       "url": "",</pre>

```

"email": "support@exxo-services.ch",
"telephone": "0800 00 00 33",
"schedule": "Mo – Fr von 08:00 bis 20:00",
"company": "exxo services GmbH",
"address": "Rietbachstrasse 10",
"zip": "8952",
"city": "Schlieren"
},
{
  "id": "155",
  "fullname": "",
  "role": "Bewirtschaftung",
  "url": "",
  "email": "support@ps-ag.ch",
  "telephone": "0800 00 88 33",
  "schedule": "Mo-Fr von 8:00 Uhr bis 20:00 Uhr",
  "company": "Property Service AG",
  "address": "Zürcherstrasse 33",
  "zip": "8003",
  "city": "Schlieren"
},
{
  "id": "177",
  "fullname": "",
  "role": "Smart Home",
  "url": "http://www.smart-home-center.ch/",
  "email": "support@shc.ch",
  "telephone": "0800 00 11 11",
  "schedule": "Mo – Fr von 08:00 bis 18:00",
  "company": "dS Smart Home Center",
  "address": "Brandstrasse 33",
  "zip": "8952",
  "city": "Schlieren"
},
{
  "id": "200",
  "fullname": "",
  "role": "Entertainment",
  "url": "www.streamnow.ch",
  "email": "fiberpower@streamnow.ch",
  "telephone": "0800 00 99 33",
  "schedule": "Mo-Fr von 8:00 bis 20:00",
  "company": "streamnow - FiberPower",
  "address": "Brandstrasse 33",
  "zip": "8952",
  "city": "Schlieren"
}
]
}

```

400	{"error":"Unauthorized_client"}
500	{"error":"Something went wrong. Please try again later."}

## 10.2. getContactAvatar

This operation is used to get the contact person image associated to a user into the system.

### Request

Method	URL
GET	/sw/getContactAvatar

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	ID	Contact ID	query	string

### Response

Status	Response
200	Response will be a jpeg
200	{"msg":"Contact not set","status":"error","code":-1}
200	{"msg":"not manager id","status":"error","code":-3}
400	{"error":"Unauthorized client"}
500	{"error":"Something went wrong. Please try again later."}

## 10.3. getDocInfo (POST)

This operation is used to get one document from one user.

### Request

Method	URL
POST	/sw/getDocInfo

### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string
POST	Doc_id	Document identifier	Form	integer

### Response

Status	Response
200	Response will be a pdf
200	{ "msg": "doc_id not correct", "status": "error", "code": "-2" }
200	{ "msg": "file not found", "status": "error", "code": "-3" }
400	{ "error": "Unauthorized_client" }
500	{ "error": "Something went wrong. Please try again later." }

## 10.4. getDocInfo (GET)

This operation is used to get one document from one user.

### Request

Method	URL
GET	/sw/getDocInfo

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	Doc_id	Document identifier	query	integer

### Response

Status	Response
200	Response will be a pdf
200	<code>{"msg":"doc_id not correct","status":"error","code":-2}"</code>
200	<code>{"msg":"file not found","status":"error","code":-3}"</code>
500	<code>{"error":"Something went wrong. Please try again later."}</code>

## 10.5. getDocsInfo

This operation is used to get the documents from one user.

### Request

Method	URL
POST	/sw/getDocsInfo

### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string

### Response

Status	Response
200	<code>{   "status": "ok",   "usertree": [...],   "repotree": [...] }</code>
400	<code>{"error":"Unauthorized_client"}</code>
500	<code>{"error":"Something went wrong. Please try again later."}</code>

## 10.6. getImage

This operation is used to get associated image of a document or category from one user.

### Request

Method	URL
GET	/sw/getImage

### Parameters

Type	Name	Description	Type	Data Type
GET	Access_token	User token	query	string
GET	Item_id	Document/item identifier	query	integer

### Response

Status	Response
200	[jpeg]
200	{ "status": "error", "msg": "item_id not correct", "code": -2 }
400	{ "error": "Unauthorized_client" }
500	{ "error": "Something went wrong. Please try again later." }

## 10.7. getNotifications

This operation is used to get the last 5 most recent notifications (events) from one user.

### Request

Method	URL
POST	/sw/getNotifications

### Parameters

Type	Name	Description	Type	Data Type
POST	Access_token	User token	form	string

## Response

Status	Response
200	<pre>{ "status": "ok",   "notifications": [...] }</pre>
400	<pre>{ "error": "Unauthorized_client" }</pre>
500	<pre>{ "error": "Something went wrong. Please try again later." }</pre>