

Smybox Server API description

Versions history:

Version	Date	Author	Description
1.1	2014-11-15	Petr Svoboda	SSA Event point has <i>preview</i> with quality option.
1.0	2014-10-13	Petr Svoboda	Initial release

Smybox Server API (SSA) is based on HTTPS web server. SSA request is always GET request with simple checksum verification. Reply is in [JSON](#) format.

SSA has two layers

1. Main point, where *events* can be enumerated (with basic informations).
2. Event point, where pictures can be obtained.

1. Main point

It resides at URL:

<https://live.smybox.com/>

and has one mandatory parameter *action* and one optional parameter *branch*.

<i>action</i> value	Description
<i>list</i>	Get enumeration of currently available <i>events</i> .
<i>state</i>	Get hash of <i>events</i> list. To determine <i>list</i> change, watch <i>state</i> instead - shorter response.
<i>branches</i>	Get list of all branches.

One can omit *branch* parameter to get *list/state* of all branches. Or explicitly specify *branch* id (obtained by *branches* action) and get filtered *list/state*.

Every reply contains field "error" with one of following codes:

error value	Description
0	NO_ERROR

1	INTERNAL_ERROR - something inside SSA is broken
2	HASH_ERROR - you have wrong Security HASH
3	REQUEST_ERROR - mandatory parameter is missing or has wrong value

1.1 List

Example reply for *list* request

```
{
  "error": 0,
  "branch": -1,
  "list": [
    {
      "name": "Novoman\u0017eel\u00e9",
      "lang": "cs",
      "created": "2014-10-13 15:30:29",
      "path": "255bfef95601890a",
      "title": "7a87a05e",
      "subdomain": "cz1",
      "branch": 1,
      "public": 1,
      "desc": "\u0160ulcovi",
      "locality": "Z\u00e9mek D\u011btenice;D\u011btenice 1;507 24 D\u011btenice;Czech
Republic"
    },
    {
      "name": "DHL",
      "lang": "en",
      "created": "2014-10-13 15:30:29",
      "path": "da13f9eb13596774",
      "title": "7a87a05e",
      "subdomain": "us1",
      "branch": 2,
      "public": 0,
      "desc": "summer celebration 1",
      "locality": "Best Best & Krieger: Lee Edward W; 300 S Grand Ave #25; Los Angeles,
CA 90071;USA"
    },
    {
      "name": "ZUNO",
      "lang": "cs",
      "created": "2014-10-13 15:30:29",
      "path": "6c5d23055b1ffd39",
      "title": "7a87a05e",
      "subdomain": "cz1",
      "branch": 1,
      "public": 0,
    }
  ]
}
```

```

        "desc": "Design market 2014",
        "locality": "Piazzeta N\u00e1rodn\u00edho divadla;N\u00e1rodn\u00ed 4; Praha
1;Czech Republic"
    },
    {
        "name": "MTV",
        "lang": "pt",
        "created": "2014-10-13 15:30:29",
        "path": "cfcb9e05c7600499",
        "title": "7a87a05e",
        "subdomain": "cz1",
        "branch": 3,
        "public": 1,
        "desc": null,
        "locality": "Abril Radiodifus\u00e3o;2296 Av. Dr. Arnaldo;S\u00e3o Paulo -
SP;Brazil"
    },
    {
        "name": "SOCIALFILMLOFT",
        "lang": "en",
        "created": "2014-10-13 15:30:29",
        "path": "f01c8c035daa2c7a",
        "title": "7a87a05e",
        "subdomain": "us2",
        "branch": 2,
        "public": 0,
        "desc": "#SUNDANCE2014",
        "locality": "2014 Sundance Ct;Palmdale; CA 93551;USA"
    },
    {
        "name": "Casa",
        "lang": "es",
        "created": "2014-10-13 15:30:29",
        "path": "7a992bd7fe18925b",
        "title": "7a87a05e",
        "subdomain": "cz1",
        "branch": 4,
        "public": 0,
        "desc": "La Casa del Espa\u00f1ol",
        "locality": "Paseo de Recoletos, 20-22;28001 Madrid;Spain"
    }
}
]
}

```

Fields inside “list” array has following meanings:

field	Description
name	Human readable <i>event</i> name.
lang	2 character language code (according to ISO 639-1) for name, desc...
created	Date and time of first picture.
path	Path to SSA <i>Event point</i> .
title	Short hash of <i>title</i> image to find out change. See <i>Event point title</i> request.
subdomain	Subdomain where SSA <i>Event point</i> is running.
branch	Index of Smybox’s franchise. See <i>branches action</i> for details.
public	1 = anyone can see the pictures. 0 = private <i>event</i> where password is needed.
desc	Longer text - description of <i>event</i> .
locality	Address where <i>event</i> take place in. Maybe will be replaced by GPS coord...

1.2 State

Example reply for *state* request

```
{
  "error": 0,
  "branch": 1,
  "state": "3942947f5f"
}
```

1.3 Branches

Example reply for *branches* request

```
{
  "error": 0,
  "branches": {
    "1": "SmileBox",
    "2": "Mr.SmileBox",
    "3": "MagicBox",
    "4": "HolaSmileBox"
  }
}
```

2. Event point

Event point URL is constructed using informations from Master point. URL is builded by formula:

[https://\\$subdomain\\$.live.smybox.com/events/\\$path\\$/](https://$subdomain$.live.smybox.com/events/$path$/)

where \$subdomain\$ and \$path\$ is obtained from Main point SSA by *list* request. For example:

<https://cz1.live.smybox.com/events/255bfef95601890a/>

As Main point, there is one mandatory parameter *action*. Depending on *action* value, there can be more mandatory parameters.

<i>action</i> value	additional mandatory parameter	success reply format	Description
<i>title</i>	-	image/jpg	Get title image of <i>event</i> . (Only request which not require password even for non-public <i>events</i> .)
<i>count</i>	-	JSON	Get number of available pictures (size of <i>list</i>).
<i>list</i>	-	JSON	Get list of all picture <i>names</i> .
<i>property</i>	<i>name</i>	JSON	Get information about picture (like orientation).
<i>preview</i>	<i>name, quality</i>	image/jpg	Get small picture of given <i>name</i> .
<i>detail</i>	<i>name, quality, size</i>	image/jpg	Get custom size picture of given <i>name</i> .
<i>original</i>	<i>name</i>	image/jpg	Get full-size picture of given <i>name</i> .

For *count* and *list* requests, reply in JSON format is returned. When an error occurs, JSON reply is returned even for *title*, *preview*, *detail* and *original* requests. Error codes are same as in Main point.

If everything goes well, picture in jpeg format is returned for *title*, *preview*, *detail* and *original* requests.

Pictures are every time in “human visible” rotation. So, no additional rotations is needed and picture can be presented to user “as is”. This also means, picture can be returned in both ,portrait and landscape, formats.

title request returns a picture with fixed size: 310 x 310 px.

Three types of pictures (*preview*, *detail* and *original*) have following meaning:

- Get all pictures of event using smallest *preview*.
- Get *detail* picture to display in full screen.
- Get *original* picture for share on social networks or to store in device.

preview request returns a picture cropped to square size : 180 x 180 px or 310 x 310 px, depending on *quality* parameters. Available *preview* request parameters are:

Parameter	Mandatory	Default value	Description
name	yes	-	Name of picture from <i>list</i> .
quality	no	medium	JPEG size and compression quality. Options are: low = 180 x 180 px, 80% quality medium = 310 x 310 px, 80% quality high = 310 x 310 px , 98% quality

To determine orientation of picture obtained by *detail* and *original*, *property* request can be used. Example response:

```
{
  "error": 0,
  "orientation": "portrait"
}
```

Aspect ratio of pictures can vary from picture to picture. It depends on *event* settings. No assumptions on picture aspect ratio can be made.

Size of *original* pictures can't be changed. Size depends on *event* settings and internet connection bandwidth available when uploading pictures.

Only *detail* picture can be resized by SSA server to any size. Available request parameters are:

Parameter	Mandatory	Default value	Description
name	yes	-	Name of picture from <i>list</i> .
quality	no	medium	JPEG compression quality. Options are: low / medium / high
size	no	1000	Size of bounding square side in pixels. Aspect ratio of picture is kept same as <i>original</i> . Picture is only scaled to fit in bounding square.

3. Security HASH

To avoid stealing pictures by bots and attackers, all SSA request are secured by HASH. HASH is **sha256** sum of GET request where '=' is replaced by ':' and '&' is replaced by '|'.

<https://live.smybox.com/?foo=1&bar=2> -> 'foo:1|bar:2'

HASH must be appended to SSA request as last GET parameter **h**.

HASH is salted by *secret* string '**alfdLSKJDGSo4twd8932405r9D{AFS9p}**' and little more data depending on SSA layer:

- Main point
- Event point
- Event point for non-public *event*.

Secret string is always appended to the end of plain string to be hashed.

3.1 Main point

...parameters and values...|\$date|\$secret\$

Main point requests are salted by appending current date in 'YY-MM-DD' format and *secret* string at the end of hashed string. Consider request:

<https://live.smybox.com/?foo=1&bar=2>

, then HASH = sha256sum('foo:1|bar:2|14-09-22|alfdLSKJDGSo4twd8932405r9D{AFS9p}') =
'c550f154892656ef312cf90f8d1d13d4ceb04b881dfb364ff080446942420994'

Full request looks like:

<https://live.smybox.com/?foo=1&bar=2&h=c550f154892656ef312cf90f8d1d13d4ceb04b881dfb364ff080446942420994>

3.2 Event point (public event)

\$path\$|...parameters and values...|\$date|\$secret\$

Event point requests are salted by prepending \$path\$ and appending current date in 'YY-MM-DD' format and *secret* string to hashed string. Consider request for *event path* = 'abcd':

<https://xyz.live.smybox.com/events/abcd/?foo=1&bar=2>

, then HASH = sha256sum('abcd|foo:1|bar:2|14-09-22|alfdLSKJDGSo4twd8932405r9D{AFS9p}') =
'b4ffc3b954c0b36a237b6879a5c055243a064b29eb26e22b682d703b79eda48d'

Full request looks like:

<https://xyz.live.smybox.com/events/abcd/?foo=1&bar=2&h=b4ffc3b954c0b36a237b6879a5c055243a064b29eb26e22b682d703b79eda48d>

3.3 Event point (non-public event)

\$path|\$password\$|...parameters and values...|\$date|\$secret\$

Event point requests for non-public events are additionally secured by \$password\$. Short phrase entered by user.

Consider request for event *path* 'abcd' and password 'mystery':

<https://xyz.live.smybox.com/events/abcd/?foo=1&bar=2>

, then HASH =

sha256sum('abcd|mystery|foo:1|bar:2|14-09-22|alfdLSKJDGSo4twd8932405r9D{AFS9p}') =
'302073fc3edf7f9fe00bf2b89aabb8fa512563c2b9a1ec6a90398fa190277d79'

Full request looks like:

<https://xyz.live.smybox.com/events/abcd/?foo=1&bar=2&h=302073fc3edf7f9fe00bf2b89aabb8fa512563c2b9a1ec6a90398fa190277d79>

There is an **exception**, when requesting *title* picture. No password is needed. HASH can be calculated as for public event. This allows to attach *titles* to all events from Main point *list* (public and private).