

# AAL-2009-2-116



## Deliverable D2.3 Appendix Elder-Spaces API Specification

**Deliverable Type:** (CO)\*

**Nature of the Deliverable:** (S)\*\*

**Date:** 03/12/12

**Distribution:** WP2

**Code:** D 2.3.

**Editor:** BYTE

**Contributors:** ORIGO

**\*Deliverable Type:** PU= Public, RE= Restricted to a group specified by the Consortium, PP= Restricted to other program participants (including the Commission services), CO= Confidential, only for members of the Consortium (including the Commission services)

**\*\* Nature of the Deliverable:** P= Prototype, R= Report, S= Specification, T= Tool, O= Other

**Abstract:**

© Copyright by the ELDER-SPACES Consortium.  
The ELDER-SPACES Consortium consists of:

BYTE	Project Coordinator	Greece
ORIGO	Partner	Hungary
FTB	Partner	Germany
e-Trikala	Partner	Greece
SEMMELWEIS	Partner	Hungary
SLG	Partner	Greece
CYBION	Partner	Italy

*This page has been intentionally left blank.*

**DOCUMENT REVISION HISTORY**

<i>Version</i>	<i>Date</i>	<i>Modifications introduced</i>	
		<i>Modification reason</i>	<i>Modified by</i>
0.1	29/11/2012	Create document and replace the Elder-Spaces API technical specification parts from D2.3 here	ORIGO
0.2	30/11/2012	Formatted and edited	BYTE
0.3	03/12/2012	Revised and formatted	BYTE, FTB

---

## Table of Contents

---

<i>This page has been intentionally left blank.....</i>	<b>2</b>
<b>1. Elder-Spaces Entities .....</b>	<b>8</b>
<b>1.1 The Person Object.....</b>	<b>8</b>
1.1.1 Description .....	8
1.1.2 Fields.....	8
1.1.3 XML representation .....	10
1.1.4 JSON representation .....	11
1.1.5 ATOM representation .....	13
<b>1.2 The Album Object.....</b>	<b>15</b>
1.2.1 Description .....	15
1.2.2 Fields.....	15
1.2.3 XML representation .....	15
1.2.4 JSON representation .....	15
1.2.5 ATOM representation .....	16
<b>1.3 The MediaItem Object.....</b>	<b>16</b>
1.3.1 Description .....	16
1.3.2 Fields.....	16
1.3.3 XML representation .....	17
1.3.4 JSON representation .....	18
1.3.5 ATOM representation .....	18
<b>1.4 The Activity Object.....</b>	<b>19</b>
1.4.1 Description .....	19
1.4.2 Fields.....	19
1.4.3 XML representation .....	19
1.4.4 JSON representation .....	20
1.4.5 ATOM representation .....	20
<b>1.5 The Message Object .....</b>	<b>21</b>
1.5.1 Description .....	21
1.5.2 Fields.....	21
1.5.3 XML representation .....	21
1.5.4 JSON representation .....	22
1.5.5 ATOM representation .....	22
<b>1.6 The Comment Object.....</b>	<b>23</b>
1.6.1 Description .....	23
1.6.2 Fields.....	23

1.6.3	XML representation .....	23
1.6.4	JSON representation .....	24
1.6.5	ATOM representation .....	24
<b>1.7</b>	<b>The Event Object .....</b>	<b>24</b>
1.7.1	Description .....	24
1.7.2	Fields.....	24
1.7.3	XML representation .....	25
1.7.4	JSON representation .....	26
1.7.5	ATOM representation .....	26
<b>1.8</b>	<b>The RSVP Object .....</b>	<b>26</b>
1.8.1	Description .....	26
1.8.2	Fields.....	27
1.8.3	XML representation .....	27
1.8.4	JSON representation .....	27
1.8.5	ATOM representation .....	27
<b>1.9</b>	<b>The Club Object .....</b>	<b>28</b>
1.9.1	Description .....	28
1.9.2	Fields.....	28
1.9.3	XML representation .....	28
1.9.4	JSON representation .....	29
1.9.5	ATOM representation .....	29
<b>1.10</b>	<b>The Join Club Object.....</b>	<b>29</b>
1.10.1	Description .....	29
1.10.2	Fields.....	30
1.10.3	XML representation .....	30
1.10.4	JSON representation .....	30
1.10.5	ATOM representation .....	30
<b>2.</b>	<b>Elder-Spaces Services .....</b>	<b>31</b>
<b>2.1</b>	<b>People service.....</b>	<b>33</b>
2.1.1	Get user .....	33
2.1.2	Get friends.....	34
2.1.3	User search.....	35
2.1.4	Get friend requests (= profile function notifications) .....	36
2.1.5	Add or modify user data.....	37
2.1.6	Change e-mail or password.....	38
2.1.7	Add friend .....	38
2.1.8	Deleting existing or dependent connection.....	39
2.1.9	Delete user.....	40
<b>2.2</b>	<b>Albums service.....</b>	<b>41</b>
2.2.1	Get albums .....	41
2.2.2	Get album comments .....	42

2.2.3	Create or modify album .....	42
2.2.4	Create album comment .....	43
2.2.5	Delete album .....	44
2.2.6	Delete album comment .....	45
<b>2.3</b>	<b>MediaItems service.....</b>	<b>46</b>
2.3.1	Get photos .....	46
2.3.2	Get photo comments .....	47
2.3.3	Upload or modify photo .....	48
2.3.4	Create photo comment .....	49
2.3.5	Delete photo .....	50
2.3.6	Delete photo comment .....	50
<b>2.4</b>	<b>Activities Services.....</b>	<b>51</b>
2.4.1	Get activities .....	51
2.4.2	Create activity .....	53
2.4.3	Delete activity .....	54
<b>2.5</b>	<b>Messages service .....</b>	<b>55</b>
2.5.1	Get messages.....	55
2.5.2	Create and send message.....	56
2.5.3	Delete message.....	57
<b>2.6</b>	<b>Events service .....</b>	<b>58</b>
2.6.1	Get events (includes event function notifications).....	58
2.6.2	Search events.....	59
2.6.3	Get event activities.....	60
2.6.4	Create or modify event.....	61
2.6.5	Create event activity.....	62
2.6.6	Delete event.....	63
2.6.7	Delete event activity.....	63
<b>2.7</b>	<b>RSVP service.....</b>	<b>64</b>
2.7.1	Get user's RSVP status of an event .....	64
2.7.2	Get RSVP statuses of an event.....	65
2.7.3	Invite, join, maybe attend or decline user to an event.....	66
<b>2.8</b>	<b>Clubs service .....</b>	<b>66</b>
2.8.1	Get clubs (includes groups function notifications) .....	66
2.8.2	Search club .....	67
2.8.3	Get club activities.....	68
2.8.4	Create or modify club .....	69
2.8.5	Create club activity .....	70
2.8.6	Delete club .....	71
2.8.7	Delete club activity .....	72
<b>2.9</b>	<b>ClubJoin service .....</b>	<b>72</b>
2.9.1	Get the user's club join record (YES / INVITED).....	72

- 2.9.2 Get club members or the list of invited people ..... 73
- 2.9.3 Invite or join user to a club ..... 74
- 2.9.4 Delete user from a club ..... 75
- 2.10 Cognitive Recommendation Services (CYBION)..... 76**
  - 2.10.1 Recommendation of friends ..... 76
  - 2.10.2 Recommendations of events ..... 77
  - 2.10.3 Recommendations of Clubs – Groups of users ..... 78
  - 2.10.4 Event Listener Service ..... 79
- 2.11 Authentication ..... 79**
  - 2.11.1 Get requestToken ..... 80
  - 2.11.2 Authorization ..... 81
  - 2.11.3 Get accessToken ..... 82

# 1. Elder-Spaces Entities

## 1.1 The Person Object

### 1.1.1 Description

The entity representing the data of persons present in the system.

### 1.1.2 Fields

Field	Type	Comment
aboutMe	String	Free text introduction of this person.
activities	String[]	Hobby interests of this person
addresses	Custom data structure	Physical addresses of this Person. Available components of an address: country, locality, postalCode, streetAddress, primary(Boolean), type (e.g. „2010-2011. febr.”)
age	Int	The age of this person.
birthday	Date (e.g. 1975-02-14)	The birthday of this person. The year value MAY be set to 0000 when the age of the Person is private or the year is not available. The year and the date can be given separately so we give it back only if both are filled in (see also: yearOfBirth, monthOfBirth, dayOfBirth)
books	String[]	The favourite books of this person.
currentLocation	Custom data structure	The current physical address of this person. Available components: country, locality, postalCode, streetAddress
displayName	String	The name of this Person, suitable for display to end-users include the first and the last name (e.g. Joseph Smarr)
emails	Custom data structure	E-mail address for this Person. Canonicalized by the Service Provider.
gender	String	The gender of this person. Must be male or female.
interests	String[]	The favourite hobbies of this person.
id	String	Unique identifier for the Person. Each Person returned MUST include a non-empty id value. This identifier MUST be unique across this user's entire set of people, but MAY not be unique across multiple users' data. It MUST be a stable ID that does not change when the same contact is returned in subsequent requests. (e.g. 1234:elderspaces.iwiw.hu)
jobInterests	String	Appropriate field of the Data Sheet – “I do this when working.”
languagesSpoken	String[]	Languages spoken. the first element is the mother tongue

Field	Type	Comment
movies	String[]	The favourite movies of this person.
music	String[]	The favourite music of this person.
name	Custom data structure	The broken-out components and fully formatted version of the person's real name. Components: 1. formatted - The full name, formatted for display (e.g. Joseph Smarr) 2. familyName - the family name of this Person, or "Last Name" in most Western languages 3. givenName - The given name of this Person, or "First Name" in most Western languages 4. honorificPrefix - The honorific prefix(es) of this Person, or "Title" in most Western languages (e.g. Mr.)
nickname	String	The casual way to address this Person in real life, e.g. "Bob" or "Bobby" instead of "Robert".
pets	String	The pets of this person. Supported but free-text one, so it is a one element list.
phoneNumbers	Custom data structure	Phone number for this Person. Available components: type (e.g. phone), value
profileUrl	String	The URL of this user's Elder-Spaces profile. (e.g. <a href="http://elderspaces.iwiw.hu/pages/user/userdata.jsp?userID=13913365">http://elderspaces.iwiw.hu/pages/user/userdata.jsp?userID=13913365</a> )
profileVideo	String	The URL of this user's favourite video on an external video sharing site attached to the user's profile.
quotes	String[]	The favourite quotes of this person.
thumbnailUrl	String	The URL of the smaller thumbnail (64x64) of this user. (e.g. <a href="http://thn1.elderspaces.iwiw.hu/0101//user/01/39/13/36/5/user_13913365_1301469612927_tn1">http://thn1.elderspaces.iwiw.hu/0101//user/01/39/13/36/5/user_13913365_1301469612927_tn1</a> )
tvShows	String[]	The favourite TV shows of this person.
updated	DateTime (e.g. 2012-11-23T04:56:22Z)	The most recent date the details of this Person were updated (i.e. the modified date of this entry)
utcOffset	DateTime (e.g. -08:00)	The offset from UTC of this Person's current time zone, as of the time this response was returned.
lastLogin	Date; in UTC String format	The last login time.
yearOfBirth	Int	The year of birth of this person.
monthOfBirth	Int	The month of birth of this person.
dayOfBirth	Int	The day of birth of this person.
monthOfNameDay	Int	The month of name day of this person.
dayOfNameDay	Int	The date of name day of this person.
maidenName	String	Name of this person at birth.
friendsCount	int	The number of friends
relation	RelationType (self, friend,	The connection of the given Person to the viewer user initiating the query; it can be nil, if there is no connection

Field	Type	Comment	
	pendingIn, pendingOut)	at all	
connectionPendingCount	int	The number of dependent connections (both directions in one).	
unreadMessageCount	int	The number of unread messages (incoming).	
birthDayCount	int	The number of friends having birthday on the given day (within a day can be a cached value, so it is not updated recording a new friend; does not operate in case of an excess number of friends).	
nameDayCount	int	The number of friends having name day on the given day (within a day can be a cached value, so it is not updated recording a new friend; does not operate in case of an excess number of friends).	
aol	String	The AOL identifier of this person.	
blog	String	The Blog page URL of this person.	
icq	String	The ICQ identifier of this person.	
msn	String	The MSN identifier of this person.	
skype	String	The Skype identifier of this person.	
freetimeInterests	String	The appropriate field of the Data Sheet: "I do this when not working".	
registrationDate	Date; UTC String format	Date of registration.	
settings	String	The user settings.	
		Key	Value
		image.commenter	EVERYBODY / FRIEND / NOBODY
thumbnail2Url	String	The URL of the bigger size thumbnail (128X128). (e.g. <a href="http://thn1.elderspaces.iwiw.hu/0101//user/01/39/13/36/5/user_13913365_1301469612927_tn2">http://thn1.elderspaces.iwiw.hu/0101//user/01/39/13/36/5/user_13913365_1301469612927_tn2</a> )	

### 1.1.3 XML representation

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://ns.opensocial.org/2008/opensocial">
  <startIndex>0</startIndex>
  <totalResults>2</totalResults>
  <isFiltered>true</isFiltered>
  <isSorted>true</isSorted>
  <isUpdatedSince>true</isUpdatedSince>
  <entry>
    <person>
      <displayName>Portál András [Papi]</displayName>
      <hasApp>>false</hasApp>
    </person>
  </entry>
</response>
```

```

    <id>13913365:elderspaces.iwiw.hu</id>
    <name>
      <familyName>Portál</familyName>
      <formatted>Portál András</formatted>
      <givenName>András</givenName>
    </name>
  <profileUrl>http://elderspaces.iwiw.hu/pages/user/userdata.jsp?userID=13913365</profileUrl>
  <thumbnailUrl>http://thn1.elderspaces.iwiw.hu/0101//user/01/39/13/36/5/user_13913365_1301469612927_tn1</thumbnailUrl>
</person>
</entry>
<entry>
  <person>
    <displayName>Fejet Lenke [Nyakas]</displayName>
    <hasApp>>false</hasApp>
    <id>104:elderspaces.iwiw.hu</id>
    <name>
      <familyName>Fejet</familyName>
      <formatted>Fejet Lenke</formatted>
      <givenName>Lenke</givenName>
    </name>
    <profileUrl>http://elderspaces.iwiw.hu/pages/user/userdata.jsp?userID=104</profileUrl>
    <thumbnailUrl>http://thn1.elderspaces.iwiw.hu/0101//user/00/00/00/10/4/user_104_1238767117477_tn1</thumbnailUrl>
  </person>
</entry>
</response>

```

## 1.1.4 JSON representation

```

{
  "startIndex":0,
  "totalResults":1,
  "entry":{
    "settings":[{"value":"FRIEND","type":"image.commenter"}],
    "languagesSpoken":["hu","en","br","ka"],
    "registrationDate":"2008-12-18T12:56:44.000Z",
    "updated":"2012-05-25T12:54:32.000Z",
    "age":36,
    "gender":"male",

```

```

"activities":["hobbi1","hobbi2"],
"yearOfBirth":1976,
"connectionPendingCount":4,
"books":["alma","nero","blabla","konyv1"," konyv2"],
"msn":"valami@valami.hu",
"relation":{"value":"self"},
"emails":[{"value":"figo@playhold.com","type":"email"}],
"profileUri":"http://elderspaces.iwiw.hu/pages/user/userdata.jsp?userID=13913365",
"monthOfBirth":3,
"utcOffset":7200,
"addresses":[{"
"streetAddress":"oiuhd oiuh oiuhd92873698376",
"postalCode":"987298",
"locality":"Budapest",
"country":"Magyarorsz g"
},
{"primary":true,
"locality":"IIIIIIIIII",
"type":"2009-jelenleg is",
"country":"IIIIIIIIII"
},
{"primary":true,
"locality":"sdsas",
"type":"2009-jelenleg is",
"country":"fsdfsf"
},
{"locality":"dfs",
"type":"2008-2009. febr.",
"country":"dddddddddddddddddd"
}],
"dayOfBirth":28,
"birthday":"1976-03-28T15:45:30.937Z",
"monthOfNameDay":3,
"friendsCount":134,
"interests":["iidotolt1"," idotolt2"],
"nameDayCount":0,
"movies":["mozi 1"," mozi2"," mozi 1"],

```

```

"aboutMe": "ihf oidf adsihf oidf adsihf",
"id": "13913365:elderspaces.iwiw.hu",
"tvShows": ["tv1", "tv2"],
"thumbnailUrl": "http://thn1.elderspaces.iwiw.hu/0101//user/01/39/13/36/5/user_13913365_1301469612927_tn1",
"name": {
  "formatted": "Admin FIGO",
  "honorificPrefix": "titel",
  "familyName": "Admin",
  "givenName": "FIGO"
},
"dayOfNameDay": 28,
"music": ["ZENEEEE"],
"phoneNumbers": [
  {
    "value": "32134324",
    "type": "phone"
  }
],
"currentLocation": {
  "streetAddress": "oiuhd oiuh oiuhd92873698376",
  "postalCode": "987298",
  "locality": "Budapest",
  "country": "Magyarorszá"
},
"lastLogin": "2012-10-15T09:33:35.000Z",
"quotes": ["â•• iidezet1â••"],
"displayName": "titel Admin FIGO",
"birthDayCount": 1
}}

```

## 1.1.5 ATOM representation

```

<feed xmlns="http://www.w3.org/2005/Atom" xmlns:osearch="http://a9.com/-/spec/opensearch/1.1">

```

```

  <entry>

```

```

    <id>urn:guid:13913365:elderspaces.iwiw.hu</id>

```

```

    <content type="application/xml">

```

```

      <person xmlns="http://ns.opensocial.org/2008/opensocial">

```

```

        <displayName>Portál András [Papi]</displayName>

```

```

        <hasApp>false</hasApp>

```

```

        <id>13913365:elderspaces.iwiw.hu</id>

```

```

    <name>
      <familyName>Portál</familyName>
      <formatted>Portál András</formatted>
      <givenName>András</givenName>
    </name>

<profileUrl>http://elderspaces.iwiw.hu/pages/user/userdata.jsp?userID=13913365</profileUrl
>

<thumbnailUrl>http://thn1.elderspaces.iwiw.hu/0101//user/01/39/13/36/5/user_13913365_13
01469612927_tn1</thumbnailUrl>
  </person>
</content>
</entry>
<entry>
  <id>urn:guid:104:elderspaces.iwiw.hu</id>
  <content type="application/xml">
    <person xmlns="http://ns.opensocial.org/2008/opensocial">
      <displayName>Fejet Lenke [Nyakas]</displayName>
      <hasApp>>false</hasApp>
      <id>104:elderspaces.iwiw.hu</id>
      <name>
        <familyName>Fejet</familyName>
        <formatted>Fejet Lenke</formatted>
        <givenName>Lenke</givenName>
      </name>
      <profileUrl>http://elderspaces.iwiw.hu/pages/user/userdata.jsp?userID=104</profileUrl>

<thumbnailUrlthumbnailUrl": "http://thn1.elderspaces.iwiw.hu/0101//user/00/00/00/10/4/user_
104_1238767117477_tn1</thumbnailUrl>
  </person>
</content>
</entry>
<osearch:startIndex>0</osearch:startIndex>
<osearch:totalResults>2</osearch:totalResults>
<osearch:itemsPerPage>2</osearch:itemsPerPage>
<author>?</author>
<link rel="rel">??</link>
</feed>

```

## 1.2 The Album Object

### 1.2.1 Description

It represents collections of MediaItem objects.

### 1.2.2 Fields

Field	Type	Comment
description	String	The description of this album.
id	Int or String	The identifier of the album. Basically Int, but there are two exceptions of String: @default – profile images @activity – news flow and mobile images
mediaItemCount	Int	The number of MediaItems in this album.
ownerId	String	The identifier of the owner user.
thumbnailUrl	String	The thumbnail URL of this album.
title	String	The title of this album.

### 1.2.3 XML representation

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://ns.opensocial.org/2008/opensocial">
  <startIndex>0</startIndex>
  <totalResults>1</totalResults>
  <isFiltered>true</isFiltered>
  <isSorted>true</isSorted>
  <isUpdatedSince>true</isUpdatedSince>
  <entry>
    <album>
      <id>6376368</id>
      <title>album title</title>
      <description>album description</description>
      <ownerId>15247988</ownerId>
      <thumbnailUrl>http://dummy.net/image_tn</thumbnailUrl>
      <mediaItemCount>18</mediaItemCount>
    </album>
  </entry>
</response>
```

### 1.2.4 JSON representation

```
{
  "startIndex": 0,
  "totalResults": 1,
  "entry":
  [
    {
```

```

    "id": "6376368",
    "description": "album description",
    "title": "album title",
    "ownerId": "15247988",
    "thumbnailUrl": "http://dummy.net/image_tn",
    "mediaItemCount": 18
  }
}

```

## 1.2.5 ATOM representation

```

<feed xmlns="http://www.w3.org/2005/Atom" xmlns:osearch="http://a9.com/-/spec/opensearch/1.1">
  <entry>
    <id>urn:guid:elderspaces.iwiw.hu:Xd9om74RW8R</id>
    <content type="application/xml">
      <album xmlns="http://ns.opensocial.org/2008/opensocial">
        <id>6376368</id>
        <title>album title</title>
        <description>album description</description>
        <ownerId>15247988</ownerId>
        <thumbnailUrl>http://dummy.net/image_tn</thumbnailUrl>
        <mediaItemCount>18</mediaItemCount>
      </album>
    </content>
  </entry>
  <osearch:startIndex>0</osearch:startIndex>
  <osearch:totalResults>1</osearch:totalResults>
  <osearch:itemsPerPage>1</osearch:itemsPerPage>
  <author>elderspaces.iwiw.hu:213123123</author>
</feed>

```

## 1.3 The MediaItem Object

### 1.3.1 Description

Entity carrying images, videos, voice recordings. By means of it media contents can be attached to also other entities.

### 1.3.2 Fields

Field	Type	Comment
albumId	Int or String	The identifier of the album. Basically Int, but there are two exceptions of String: @default – profile images @activity – news flow and mobile images
created	Date; UTC String format	The date when this MediaItem was created.

description	String	The description of this MediaItem.
fileSize	Number	The size in bytes of this MediaItem.
id	String	The unique identifier of this MediaItem.
lastUpdated	Date; UTC String format	The date of the last update of this MediaItem.
mimeType	String	The mimeType of this MediaItem. (e.g. image/jpeg)
numComments	Int	The number of comments on of this MediaItem.
thumbnailUrl	String	The thumbnail URL of this MediaItem.
title	String	The title of this MediaItem.
type	String	The type of this MediaItem. In case of user images: IMAGE In case of NEW_BOARDMSG type activity: NO (in case only the link exists without image) IMAGE (in case link exists and image belongs to it) IMAGE (in case of image share only) VIDEO (in case video belongs to the link)
url	String	The URL of this MediaItem.

### 1.3.3 XML representation

```

<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://ns.opensocial.org/2008/opensocial">
<startIndex>0</startIndex>
<totalResults>11</totalResults>
<isFiltered>>true</isFiltered>
<isSorted>>true</isSorted>
<isUpdatedSince>>true</isUpdatedSince>
<entry>
<mediaitem>
  <id>230032747</id>
  <type>image</type>
  <mimeType>image/jpeg</mimeType>
  <url>http://dummy.net/image</url>
  <title>image title</title>
  <albumId>1</albumId>
  <ownerId>15247988</ownerId>
  <fileSize>26285</fileSize>
  <created>2011-03-22T14:28:28.000Z</created>
  <thumbnailUrl>http://dummy.net/image_tn</thumbnailUrl>
  <lastUpdated>2011-03-22T14:28:28.000Z</lastUpdated>
  <numComments>1</numComments>
</mediaitem>
</entry>
</response>

```

### 1.3.4 JSON representation

```
{
  "startIndex": 0,
  "totalResults": 1,
  "entry":
    [{
      "albumId": "1",
      "ownerId": "15247988",
      "lastUpdated": "2011-03-22T14:28:28.000Z",
      "type": "image",
      "url": "http://dummy.net/image",
      "id": "230032747",
      "fileSize": 26285,
      "numComments": 1,
      "title": "image title",
      "created": "2011-03-22T14:28:28.000Z",
      "thumbnailUrl": "http://dummy.net/image_tn",
      "mimeType": "image/jpeg"
    }]
}
```

### 1.3.5 ATOM representation

```
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:osearch="http://a9.com/-/spec/opensearch/1.1">
  <entry>
    <id>urn:guid:elderspaces.iwiw.hu:Xd9om74RW8R</id>
    <content type="application/xml">
      <mediaitem xmlns="http://ns.opensocial.org/2008/opensocial">
        <id>230032747</id>
        <type>image</type>
        <mimeType>image/jpeg</mimeType>
        <url>http://dummy.net/image</url>
        <title>image title</title>
        <albumId>1</albumId>
        <ownerId>15247988</ownerId>
        <fileSize>26285</fileSize>
        <created>2011-03-22T14:28:28.000Z</created>
        <thumbnailUrl>http://dummy.net/image_tn</thumbnailUrl>
        <lastUpdated>2011-03-22T14:28:28.000Z</lastUpdated>
        <numComments>1</numComments>
      </mediaitem>
    </content>
  </entry>
  <osearch:startIndex>0</osearch:startIndex>
  <osearch:totalResults>1</osearch:totalResults>
  <osearch:itemsPerPage>1</osearch:itemsPerPage>
  <author>elderspaces.iwiw.hu:213123123</author>
</feed>
```

## 1.4 The Activity Object

### 1.4.1 Description

Entity representing the news flow elements, incidences. The Activity feed functionality is also realised through this.

### 1.4.2 Fields

Field	Type	Comment
appId	String	Unique identifier of the activity generating application. It plays a role only in case of application activities.
body	String	The description of this activity.
bodyId	String	In case of using activity templates, the identifier of the concerning activity body template.
id	String	The unique identifier of this activity.
mediaItems	MediaItem[]	List of MediaItem Objects attached to this activity.
postedTime	Date; UTC String format	The date when this activity was posted.
templateParams	String[]	In case of using activity templates, the passed key-value pairs to the concerning activity template.
title	String	The title of this activity.
titleId	String	In case of using activity templates, the identifier of the concerning activity title template.
userId	String	The unique identifier of the user corresponding to the activity.

### 1.4.3 XML representation

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://ns.opensocial.org/2008/opensocial">
<startIndex>0</startIndex>
<totalResults>11</totalResults>
<isFiltered>true</isFiltered>
<isSorted>true</isSorted>
<isUpdatedSince>true</isUpdatedSince>
<entry>
<activity>
  <appId>230032747</appId>
  <body>t1b1</body>
  <bodyId>NEW_BOARDMSG_BODY</bodyId>
  <id>0001d30e50af4e4f0024ba59</id>
  <mediaItems></mediaItems>
  <postedTime>1353666127000</postedTime>
  <title>Vass Andrea </title>
  <titleId>NEW_BOARDMSG</titleId>
  <userId>119566:portal.iwiw</userId>
```

```

</activity>
</entry>
</response>

```

### 1.4.4 JSON representation

```

{
  "startIndex": 0,
  "totalResults": 1,
  "entry": [
    {
      "appld": "230032747",
      "body": "t1b1",
      "bodyId": "NEW_BOARDMSG_BODY",
      "id": "0001d30e50af4e4f0024ba59",
      "mediaItems": [],
      "postedTime": 1353666127000,
      "title": "Vass Andrea ",
      "titleId": "NEW_BOARDMSG",
      "userId": "119566:portal.iwiw",
    }
  ]
}

```

### 1.4.5 ATOM representation

```

<feed xmlns="http://www.w3.org/2005/Atom" xmlns:osearch="http://a9.com/-/spec/opensearch/1.1">
  <entry>
    <id>urn:guid:elderspaces.iwiw.hu:213123123</id>
    <content type="application/xml">
      <activity xmlns="http://ns.opensocial.org/2008/opensocial">
        <appld>230032747</appld>
        <body>t1b1</body>
        <bodyId>NEW_BOARDMSG_BODY</bodyId>
        <id>0001d30e50af4e4f0024ba59</id>
        <mediaItems></mediaItems>
        <postedTime>1353666127000</postedTime>
        <title>Vass Andrea </title>
        <titleId>NEW_BOARDMSG</titleId>
        <userId>119566:portal.iwiw</userId>
      </activity>
    </content>
  </entry>
  <osearch:startIndex>0</osearch:startIndex>
  <osearch:totalResults>1</osearch:totalResults>
  <osearch:itemsPerPage>1</osearch:itemsPerPage>
  <author>elderspaces.iwiw.hu:213123123</author>
</feed>

```

## 1.5 The Message Object

### 1.5.1 Description

The entity carrying the personal messages.

### 1.5.2 Fields

Field	Type	Comment
body	String	The body of this message
collectionIds	String	The identifier of the message container. Two special collectionId: @inbox; @outbox
id	String	The unique identifier of this message.
inReplyTo	String	The unique identifier of the replied user.
recipients	String[]	The unique identifiers of the recipient users.
senderId	String	The unique identifier of the sender user.
status	String	The status of this message: NEW – unread DELETED – supported but the message is not accessible read message – lack of the status field indicates it
timeSent	Date; UTC String format	The date when this message was sent.
title	String	The title of this message
type	String	The type of this message. Only PRIVATE_MESSAGE allowed.
postboxStatus	postboxStatus	Secondary mailbox status

#### postboxStatus

Field	Type	Comment
userId	String	The unique identifier of the person addressed.
status	String	The mailbox status: @inbox: NEW, READ @outbox: NEW, READ, ANSWERED, NEW_DELETED

### 1.5.3 XML representation

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://ns.opensocial.org/2008/opensocial">
<startIndex>0</startIndex>
<totalResults>11</totalResults>
<isFiltered>true</isFiltered>
<isSorted>true</isSorted>
<isUpdatedSince>true</isUpdatedSince>
<entry>
<message>
  <id>1301</id>
```

```

    <type>
      <value>PRIVATE_MESSAGE</value>
    </type>
    <title>d asd asd asd </title>
    <body>asd asda sd asd </body>
    <recipients>20359:portal.iwiw</recipients>
    <recipients>8427:portal.iwiw</recipients>
    <senderId>119566:portal.iwiw</senderId>
    <collectionIds>@outbox</collectionIds>
    <timeSent>2012-11-23T11:40:04.000000000+01:00</timeSent>
  </message>
</entry>
</response>

```

## 1.5.4 JSON representation

```

{
  "startIndex": 0,
  "totalResults": 1,
  "entry":
    [
      {
        "body": "asd asda sd asd ",
        "recipients": ["20359:portal.iwiw", "8427:portal.iwiw"],
        "senderId": "119566:portal.iwiw",
        "collectionIds": ["@outbox"],
        "type": "privateMessage",
        "id": "1301",
        "title": "d asd asd asd ",
        "postboxStatus": [],
        "timeSent": "2012-11-23T10:40:04.000Z"
      }
    ]
}

```

## 1.5.5 ATOM representation

```

<feed xmlns="http://www.w3.org/2005/Atom" xmlns:osearch="http://a9.com/-/spec/opensocial/1.1">
  <entry>
    <id>urn:guid:elderspaces.iwiw.hu:213123123</id>
    <content type="application/xml">
      <message xmlns="http://ns.opensocial.org/2008/opensocial">
        <id>1301</id>
        <type>
          <value>PRIVATE_MESSAGE</value>
        </type>
        <title>d asd asd asd </title>
        <body>asd asda sd asd </body>
        <recipients>20359:portal.iwiw</recipients>
        <recipients>8427:portal.iwiw</recipients>
        <senderId>119566:portal.iwiw</senderId>

```

```

    <collectionIds>@outbox</collectionIds>
    <timeSent>2012-11-23T11:40:04.000000000+01:00</timeSent>
</message>
</content>
</entry>
<osearch:startIndex>0</osearch:startIndex>
<osearch:totalResults>1</osearch:totalResults>
<osearch:itemsPerPage>1</osearch:itemsPerPage>
<author>elderspaces.iwiw.hu:213123123</author>
</feed>

```

## 1.6 The Comment Object

### 1.6.1 Description

The entity carrying the data of comments.

### 1.6.2 Fields

Field	Type	Comment
id	String	The unique identifier of the comment.
text	String	The comment text.
created	Date; UTC String format	The date of creation.
creatorId	String	The unique identifier of the creating user.

### 1.6.3 XML representation

```

<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://ns.opensocial.org/2008/opensocial">
<startIndex>0</startIndex>
<totalResults>1</totalResults>
<isFiltered>true</isFiltered>
<isSorted>true</isSorted>
<isUpdatedSince>true</isUpdatedSince>
<entry>
<comment>
    <id>230032747-15247988-USER-1</id>
    <text>text of the comment</text>
    <created>2011-03-22T14:33:51.000Z</created>
    <creatorId>12345</creatorId>
</comment>
</entry>
</response>

```

## 1.6.4 JSON representation

```
{
  "startIndex": 0,
  "totalResults": 1,
  "entry":
    [{
      "text": "text of the comment",
      "creatorId": "12345",
      "id": "230032747-15247988-USER-1",
      "created": "2011-03-22T14:33:51.000Z"
    }]
}
```

## 1.6.5 ATOM representation

```
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:osearch="http://a9.com/-/spec/opensearch/1.1">
  <entry>
    <id>urn:guid:elderspaces.iwiw.hu:Xd9om74RW8R</id>
    <content type="application/xml">
      <comment xmlns="http://ns.opensocial.org/2008/opensocial">
        <id>230032747-15247988-USER-1</id>
        <text>text of the comment</text>
        <created>2011-03-22T14:33:51.000Z</created>
        <creatorId>12345</creatorId>
      </comment>
    </content>
  </entry>
  <osearch:startIndex>0</osearch:startIndex>
  <osearch:totalResults>1</osearch:totalResults>
  <osearch:itemsPerPage>1</osearch:itemsPerPage>
  <author>elderspaces.iwiw.hu:213123123</author>
</feed>
```

## 1.7 The Event Object

### 1.7.1 Description

Entity carrying the data of events introduced in the Events function.

### 1.7.2 Fields

Field	Type	Comment
id	String	The unique identifier of this event.
name	String	The name of this event.
description	String	The description of this event.
shortDescription	String	The short description of this event.

startDate	String	Time of Event start, formatted in accordance with the precision set *
endDate	String	Time of Event end, formatted in accordance with the precision set *
category	String	Event category – constants set at system level: BIRTHDAY, NAMEDAY, PARTY, CONCERT, CINEMA, THEATRE, PROGRAM, WEDDING, MEETING, SPORTEVENT, FESTIVAL, CELEBRATION, OTHER
location	String	The location of this event.
address	String	The physical address of this event.
addressLatitude	Float	The location coordinates of this event.
addressLongitude	Float	

The startDate and endDate are in the format: „yyyy-MM-dd HH:mm”

Formatting will be performed only to the exactness of the final date.

Precision	Format	Example
Year	yyyy	2010
Month	yyyy-MM	2010-06
Day	yyyy-MM-dd	2010-06-14
Hour	yyyy-MM-dd HH	2010-06-14 11
Minute	yyyy-MM-dd HH:mm	2010-06-14 11:30

### 1.7.3 XML representation

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://ns.opensocial.org/2008/opensocial">
<startIndex>0</startIndex>
<totalResults>11</totalResults>
<isFiltered>true</isFiltered>
<isSorted>true</isSorted>
<isUpdatedSince>true</isUpdatedSince>
<entry>
<anniversary>
  <address xmlns="http://ns.opensocial.org/2008/opensocial"></address>
  <name xmlns="http://ns.opensocial.org/2008/opensocial">Pipi</name>
  <location></location>
  <id>1184</id>
  <description></description>
  <shortDescription></shortDescription>
  <category>BULI</category>
  <startDate>2013-08</startDate>
  <endDate>2013-08</endDate>
</anniversary>
</entry>
</response>
```

## 1.7.4 JSON representation

```
{
  "startIndex": 0,
  "totalResults": 1,
  "entry":
    [{
      "id": "1184",
      "startDate": "2013-08",
      "category": "BULI",
      "shortDescription": "",
      "location": "",
      "address": "",
      "description": "",
      "name": "Pipi",
      "endDate": "2013-08"
    }]
}
```

## 1.7.5 ATOM representation

```
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:osearch="http://a9.com/-/spec/opensource/1.1">
  <entry>
    <id>urn:guid:elderspaces.iwiw.hu:213123123</id>
    <content type="application/xml">
      <anniversary xmlns="http://ns.opensocial.org/2008/opensocial">
        <address xmlns="http://ns.opensocial.org/2008/opensocial"></address>
        <name xmlns="http://ns.opensocial.org/2008/opensocial">Pipi</name>
        <location></location>
        <id>1184</id>
        <description></description>
        <shortDescription></shortDescription>
        <category>BULI</category>
        <startDate>2013-08</startDate>
        <endDate>2013-08</endDate>
      </anniversary>
    </content>
  </entry>
  <osearch:startIndex>0</osearch:startIndex>
  <osearch:totalResults>1</osearch:totalResults>
  <osearch:itemsPerPage>1</osearch:itemsPerPage>
  <author>elderspaces.iwiw.hu:213123123</author>
</feed>
```

## 1.8 The RSVP Object

### 1.8.1 Description

Entity carrying the data of RSVP a user to an event.

## 1.8.2 Fields

Field	Type	Comment
personId	String	The unique identifier of the user.
eventId	String	The unique identifier of the event.
status	String	The rsvp status. Possible values: INVITED, YES, NO, MAYBE, BANNED

## 1.8.3 XML representation

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://ns.opensocial.org/2008/opensocial">
<startIndex>0</startIndex>
<totalResults>1</totalResults>
<isFiltered>true</isFiltered>
<isSorted>true</isSorted>
<isUpdatedSince>true</isUpdatedSince>
<entry>
<rsvp>
  <personId>12342</personId>
  <eventId>42213</eventId>
  <status>YES</status>
</rsvp>
</entry>
</response>
```

## 1.8.4 JSON representation

```
{
  "startIndex": 0,
  "totalResults": 1,
  "entry":
    [{
      "personId": "12342",
      "eventId": "42213",
      "status": "YES"
    }]
}
```

## 1.8.5 ATOM representation

```
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:osearch="http://a9.com/-/spec/opensocial/1.1">
<entry>
<id>urn:guid:elderspaces.iwiw.hu:Xd9om74RW8R</id>
<content type="application/xml">
<rsvp xmlns="http://ns.opensocial.org/2008/opensocial">
  <personId>12342</personId>
  <eventId>42213</eventId>
```

```

    <status>YES</status>
  </rsvp>
</content>
</entry>
<osearch:startIndex>0</osearch:startIndex>
<osearch:totalResults>1</osearch:totalResults>
<osearch:itemsPerPage>1</osearch:itemsPerPage>
<author>elderspaces.iwiw.hu:213123123</author>
</feed>

```

## 1.9 The Club Object

### 1.9.1 Description

Entity carrying the data of clubs introduced in the Clubs function.

### 1.9.2 Fields

Field	Type	Comment
id	String	The unique identifier of this club.
name	String	The name of this club.
description	String	The description of this club.
shortDescription	String	The short description of this club.
category	String	Club category – constants set at system level: ANY, ANIMAL, CAR, COMPANY, FAMILY, FASHION, HEALTH, LIFESTYLE, HOBBY, SCHOOL, IT, GAME, GASTRONOMY, CULTURE, ART, COUNTRY, RELATIONSHIP, SPORT, SERVICE, ENTERTAINMENT, SOCIETY, SCIENCE, TRAVEL, BUSINESS, RELIGION, CITY, MUSIC, OTHER.

### 1.9.3 XML representation

```

<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://ns.opensocial.org/2008/opensocial">
  <startIndex>0</startIndex>
  <totalResults>1</totalResults>
  <isFiltered>true</isFiltered>
  <isSorted>true</isSorted>
  <isUpdatedSince>true</isUpdatedSince>
  <entry>
    <club>
      <id>12342</id>
      <name>Klub neve</name>
      <description>Klub leírása</description>
      <shortDescription>Klub rövid leírása</shortDescription>
      <category>HOBBY</category>
    </club>
  </entry>
</response>

```

```

</club>
</entry>
</response>

```

## 1.9.4 JSON representation

```

{
  "startIndex": 0,
  "totalResults": 1,
  "entry":
    [
      {
        "id": 12342,
        "name": "Klub neve",
        "description": "Klub leírása",
        "shortDescription": "Klub rövid leírása"
        "category": "HOBBY"
      }
    ]
}

```

## 1.9.5 ATOM representation

```

<feed xmlns="http://www.w3.org/2005/Atom" xmlns:osearch="http://a9.com/-/spec/opensource/1.1">
  <entry>
    <id>urn:guid:elderspaces.iwiw.hu:Xd9om74RW8R</id>
    <content type="application/xml">
      <club xmlns="http://ns.opensocial.org/2008/opensocial">
        <id>12342</id>
        <name>Klub neve</name>
        <description>Klub leírása</description>
        <shortDescription>Klub rövid leírása</shortDescription>
        <category>HOBBY</category>
      </club>
    </content>
  </entry>
  <osearch:startIndex>0</osearch:startIndex>
  <osearch:totalResults>1</osearch:totalResults>
  <osearch:itemsPerPage>1</osearch:itemsPerPage>
  <author>elderspaces.iwiw.hu:213123123</author>
</feed>

```

## 1.10 The Join Club Object

### 1.10.1 Description

Entity carrying the data of joining a user to a club.

## 1.10.2 Fields

Field	Type	Comment
personId	String	The unique identifier of the user.
clubId	String	The unique identifier of the club.
status	String	The join status. Possible values: INVITED, YES, NO

## 1.10.3 XML representation

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://ns.opensocial.org/2008/opensocial">
<startIndex>0</startIndex>
<totalResults>1</totalResults>
<isFiltered>true</isFiltered>
<isSorted>true</isSorted>
<isUpdatedSince>true</isUpdatedSince>
<entry>
<clubJoin>
  <personId>12342</personId>
  <clubId>42213</clubId>
  <status>YES</status>
</clubJoin>
</entry>
</response>
```

## 1.10.4 JSON representation

```
{
  "startIndex": 0,
  "totalResults": 1,
  "entry":
    [{
      "personId": "12342",
      "clubId": "42213",
      "status": "YES"
    }]
}
```

## 1.10.5 ATOM representation

```
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:osearch="http://a9.com/-/spec/opensocial/1.1">
<entry>
<id>urn:guid:elderspaces.iwiw.hu:Xd9om74RW8R</id>
<content type="application/xml">
<clubJoin xmlns="http://ns.opensocial.org/2008/opensocial">
  <personId>12342</personId>
  <clubId>42213</clubId>
```

```

    <status>YES</status>
</clubJoin>
</content>
</entry>
<osearch:startIndex>0</osearch:startIndex>
<osearch:totalResults>1</osearch:totalResults>
<osearch:itemsPerPage>1</osearch:itemsPerPage>
<author>elderspaces.iwiw.hu:213123123</author>
</feed>

```

## 2. Elder-Spaces Services

In case of the services we give the totally general URL with all accessible parameters then inform about the special cases. We introduce the following notations:

- We give the parameters of the URL between square braces, e.g.: {userId}
- The number of parameters is given following the parameter. If nothing is given it means one.
  - plus (+) denotes at least one case, e.g.: {userId}+
  - star (\*) denotes an arbitrary number of parameters: – 0, 1, 2, ... – parameters, e.g. : {userId}\*
  - question mark (?) denotes the maximum of one case (not compulsory parameter), e.g.: {userId}?
- The URL can accept special replacement symbols which carry individual meaning in the given context. These start with @ sign (e.g.: @me) and we describe their exact meaning at the individual functions in details.

*Table 1: Summary table of the services and their available resources*

Service	REST method	Use case
2.1 People Service	GET	2.1.1 Get user
		2.1.2 Get friends
		2.1.3 User search
		2.1.4 Get friend requests (= profile function notifications)
	POST / PUT	2.1.5 Add or modify user data
		2.1.6 Change e-mail or password
		2.1.7 Add friend
	DELETE	2.1.8 Deleting existing or dependent connection
		2.1.9 Delete user
6.2.2 Albums Service	GET	2.2.1 Get albums
		2.2.2 Get album comments

Service	REST method	Use case
	POST / PUT	2.2.3 Create or modify album
		2.2.4 Create album comment
	DELETE	2.2.5 Delete album
		2.2.6 Delete album comment
6.2.3 MediaItems Service	GET	2.3.1 Get photos
		2.3.2 Get photo comments
	POST / PUT	2.3.3 Upload or modify photo
		2.3.4 Create photo comment
	DELETE	2.3.5 Delete photo
		2.3.6 Delete photo comment
6.2.4 Activities Service	GET	2.4.1 Get activities
	POST / PUT	2.4.2 Create activity
	DELETE	2.4.3 Delete activity
6.2.5 Messages Service	GET	2.5.1 Get messages
	POST / PUT	2.5.2 Create and send message
	DELETE	2.5.3 Delete message
6.2.6 Events Service	GET	2.6.1 Get events (includes events function notifications)
		2.6.2 Search events
	POST / PUT	2.6.3 Get event activities
		2.6.4 Create or modify event
		2.6.5 Create event activity
	DELETE	2.6.6 Delete event
6.2.7 RSVP Service	GET	2.7.1 Get user's RSVP status of an event
		2.7.2 Get RSVP statuses of an event
	POST / PUT	2.7.3 Invite, join, maybe attend or decline user to an event
6.2.8 Clubs Service	GET	2.8.1 Get clubs (includes groups function notifications)
		2.8.2 Search club
		2.8.3 Get club activities
	POST / PUT	2.8.4 Create or modify club
		2.8.5 Create club activity
	DELETE	2.8.6 Delete club 0 Delete club activity
6.2.9 ClubJoin Service	GET	2.9.1 Get the user's club join record (YES / INVITED)
		2.9.2 Get club members or the list of invited people
	POST / PUT	2.9.3 Invite or join user to a club

Service	REST method	Use case
	DELETE	2.9.4 Delete user from a club
2.10 Cognitive Recommendation Services	GET	2.10.1 Recommendation of friends
		2.10.2 Recommendation of events
		2.10.3 Recommendation of clubs – Groups
	POST	2.10.4 Event Listener Service
2.11 Authentication	GET	2.11.1 Get requestToken
		2.11.2 Authorization
		2.11.3 Get accessToken

## 2.1 People service

### 2.1.1 Get user

#### 2.1.1.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to obtain any available data of a user.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/people/{userId}*/{groupId}?/{personId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, cannot be mixed <ul style="list-style-type: none"> <li>○ user identifier(s)</li> <li>○ @me symbol: means the user assigned to the given request</li> </ul> <p>If the {userId} is missing, then the system interprets the request as user's search In this case filter specification is compulsory!</p> </li> <li>• groupId: since the groups are not supported in the Elder-Spaces, so it can have one of the symbols, if not given, gets replaced by @. <ul style="list-style-type: none"> <li>○ @self: the user himself / herself (default)</li> </ul> </li> <li>• personId: further filtering can be performed here from the users determined by the {groupId} (friends).</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Person Object
Comments	<ul style="list-style-type: none"> <li>• as default the id and displayName fields are given back. Further fields can be requested by means of the fields parameter.</li> <li>• of course the Elder-Spaces specific fields can also be accessed. The description of these is given in chapter 1.</li> <li>• of course only such fields are returned, which are allowed by its owner – the queried user – for the actual user, in accordance with the Elder-Spaces authorisation levels: private / can only be seen by</li> </ul>

	friends / public data.
--	------------------------

### 2.1.1.2 Use cases

URI	Description
GET /people/{userId}+	{userId} user / users data query
GET /people/{userId}+/@self	{userId} user / users data query
GET /people/@me	data of the viewer user

### 2.1.1.3 Examples

URI	Description
GET /people/111,222,333?fields=age,gender	query of the id, displayName, age and gender fields, users with 111, 222 and 333 identifier

## 2.1.2 Get friends

### 2.1.2.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to obtain the friend connections and also the pending connection requests of a user.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/people/{userId}*/{groupId}*/{personId}*
Parameters	<ul style="list-style-type: none"> <li>• <b>userId</b>: is one of the followings, cannot be mixed <ul style="list-style-type: none"> <li>○ @me symbol: means the user assigned to the given request</li> </ul>           If the {userId} is missing, then the system interprets the request as user's search In this case filter specification is compulsory! For details see 2.1.3.         </li> <li>• <b>groupId</b>: since the groups are not supported in the Elder-Spaces, so it can have one of the symbols, if not given, gets replaced by @. <ul style="list-style-type: none"> <li>○ @friends: friends of the user</li> <li>○ @all: friends and dependent connections of the user</li> </ul> </li> <li>• <b>personId</b>: further filtering can be performed here from the users determined by the {groupId} (friends), since in case of the Elder-Spaces Portal API all users are accessible, not only the friends, so it is really not very much significant (the users' data can directly be accessed in the format ROOT}/people/{userId} as well.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of Person Objects
Comments	<ul style="list-style-type: none"> <li>• as default the id and displayName fields are given back. Further</li> </ul>

	<p>fields can be requested by means of the parameter “fields”.</p> <ul style="list-style-type: none"> <li>• of course only such fields are returned, which are allowed by its owner – the queried user – for the actual user, in accordance with the Elder-Spaces authorisation levels: private / can only be seen by friends / public data.</li> <li>• the function is supported only up to a number of friends set in system parameter. above of it 409 CONFLICT error gets triggered; in case of such a User’s search function can be used</li> <li>• the filter named hasThumbnail gets introduced which gives back only such users who obtain profile picture</li> <li>• we support the filter named relation, by means of which the dependent connections can be filtered (filterBy=relation&amp;filterValue=pendingIn&amp;filterOp=equals)</li> <li>• fields that can be filtered: gender, name, phone, address, city, age</li> </ul>
--	--

### 2.1.2.2 Use cases

URI	Description
GET /people/{userId}/@friends	{userId} user’s friend data
GET /people/@me/@friends	friends of the viewer user

### 2.1.2.3 Examples

URI	Description
GET /people/111/@friends?startIndex=20&count=10&sortBy=name	it gives back the 20th-30th data of {userId} user’s friends, by name, in ascending order

## 2.1.3 User search

### 2.1.3.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to search a person by given parameters.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/people/search
Parameters	<ul style="list-style-type: none"> <li>• name – it searches in all name type fields; minimum 3 character must always be specified; case-insensitive</li> <li>• gender – filters for gender; possible values are: M or F</li> <li>• address – it searches in all address-like information</li> <li>• age-lower, age-upper – filters for age; only one end of the interval can also be given; it gives back the specified value too (inclusive)</li> </ul>

	<ul style="list-style-type: none"> <li>• hasThumbnail – if it is true gives only back those users obtaining profile picture</li> <li>• relation – in case the given value is friends, then we search only among friends</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of Person Objects
Comments	In case it gets illegal parameter, a 400 BAD REQUEST error gets triggered. The standard OpenSocial paging parameters can be used.

### 2.1.3.2 Use cases

URI	Description
GET /people/search?{field}={keyword}	search users with the given parameters

### 2.1.3.3 Examples

URI	Description
GET /people/search?name=Peter&address=Budapest&age-upper=55	search users with the given parameters

## 2.1.4 Get friend requests (= profile function notifications)

### 2.1.4.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to query for dependent friend requests initiated towards the user.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/people/{userId}/{groupId}?filterBy=relation&filterValue=pendingIn&filterOp>equals
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, cannot be mixed <ul style="list-style-type: none"> <li>○ user identifier(s)</li> <li>○ @me symbol: means the user assigned to the given request</li> </ul> </li> <li>• groupId: must be @all</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of Person Objects
Comments	Instead of @me factual {userId} can also be used, but if it does not match with the viewer, the request triggers a 401 UNAUTHORIZED error.

### 2.1.4.2 Use cases

URI	Description
GET /people/{userId}/{groupId}?filterBy=relation &filterValue=pendingIn&filterOp=equals	query pending friend requests

### 2.1.4.3 Examples

URI	Description
GET /people/@me/@all?filterBy=relation&filterValue=pendingIn&filterOp=equals	query pending friend requests of the viewer user

## 2.1.5 Add or modify user data

### 2.1.5.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to add new data to the user's profile or modify existing data.
URI	<a href="http://api.elderspaces.iwiw.hu/social/papi/rest/people/{userId}">http://api.elderspaces.iwiw.hu/social/papi/rest/people/{userId}</a>
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, cannot be mixed               <ul style="list-style-type: none"> <li>○ user identifier(s)</li> <li>○ @me symbol: means the user assigned to the given request</li> </ul> </li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.1.5.2 Use cases

URI	Description
POST /people/{userId}	edit {userId} user's data
POST /people/@me	edit the viewer user's data

### 2.1.5.3 Examples

URI	POST data	Response	Description
POST /people/@me	{"nickname": "Benji2"}	200 OK	add or modify the nickname of the viewer user

## 2.1.6 Change e-mail or password

### 2.1.6.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to add new data to the user's profile or modify existing data.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/ people/{userId}
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, cannot be mixed <ul style="list-style-type: none"> <li>○ user identifier(s)</li> <li>○ @me symbol: means the user assigned to the given request</li> </ul> </li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.1.6.2 Use cases

URI	Description
POST /people/{userId}	edit {userId} user's data
POST /people/@me	edit the viewer user's data

### 2.1.6.3 Examples

URI	POST data	Response	Description
POST /people/@me	{"emails": "newemail@example.com"}	200 OK	change the e-mail address of the viewer user
POST /people/@me	{"password": "newpassword"}	200 OK	change the password of the viewer user

## 2.1.7 Add friend

### 2.1.7.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to add a new friend connection to the user's profile.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/people/{userId}/{groupId}
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, cannot be mixed <ul style="list-style-type: none"> <li>○ user identifier(s)</li> <li>○ @me symbol: means the user assigned to the given request</li> </ul> </li> <li>• personId: user identifier</li> </ul>

	<ul style="list-style-type: none"> <li>• groupId: must be @friends</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	<p>For connection initiation the Person object must be POSTed into the @friends collection. (In the object it is enough if only the id is filled in) POST /people/{userId}/@friends</p> <p>In case of successful connection initiation the answer is 202 Accepted, since the connection expects confirmation from the other party. In case between the two parties there has already been friends or pendingOut connection, then the 409 CONFLICT error gets triggered. In case there has been pendingIn connection then the above call is identical with the dependent connection acceptance. In case the id of the POSTed object does not match with the actual user, a 401 UNAUTHORIZED error gets triggered.</p>

### 2.1.7.2 Use cases

URI	Description
POST /people/{userId}/@friends	add a new friend of {userId}

### 2.1.7.3 Examples

URI	POST data	Response	Description
POST /people/@me/@friends	{"id":"elderspaces.iwiw.hu:Xd9om74RW8R"}	200 OK	add the given user as a friend of the viewer user

## 2.1.8 Deleting existing or dependent connection

### 2.1.8.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete an existing or dependent friend connection.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/people/{userId}/{groupId}/personId
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, cannot be mixed <ul style="list-style-type: none"> <li>○ user identifier(s)</li> <li>○ @me symbol: means the user assigned to the given request</li> </ul> </li> <li>• groupId: must be @friends</li> <li>• personId: user identifier(s)</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code

Comments	An existing or dependent connection must be DELETED into the @friends collection using the appropriate personId. In case of non existing connection a 409 CONFLICT error gets triggered.
----------	--

### 2.1.8.2 Use cases

URI	Description
DELETE /people/{userId}/@friends/{personId}	delete an existing or depending connection of the user

### 2.1.8.3 Examples

URI	Response	Description
DELETE /people/@me/@friends/elderspaces.iwiw.hu:Xd9om74RW8R	200 OK	delete the given person from the connections of the viewer user

## 2.1.9 Delete user

### 2.1.9.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete a user.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/people/{userId}
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, cannot be mixed <ul style="list-style-type: none"> <li>○ user identifier(s)</li> <li>○ @me symbol: means the user assigned to the given request</li> </ul> </li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.1.9.2 Use cases

URI	Description
DELETE /people/@me	delete the viewer user

### 2.1.9.3 Examples

URI	Response	Description
DELETE /people/@me	200 OK	delete the viewer user

## 2.2 Albums service

### 2.2.1 Get albums

#### 2.2.1.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to get the list of the user's photo albums.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/albums/{userId}/{groupId}*/{albumId}* */{albumId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: in case of Elder-Spaces only the @self supported (albums of friends can be queried only with exact albumId)</li> <li>• albumId: optional, identifier / identifiers of the album / albums queried (@default – profile pictures, @activity – news flow and mobile pictures)</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of Album Objects
Comments	<ul style="list-style-type: none"> <li>• thumbnail the size 128x128</li> <li>• in accordance with the specification; in case the albumId is given then we give back also the elements of the album</li> </ul> <p>„optionalAlbumId: Id of the album. When specified, returns media items from the album.”</p> <p>In our case, because of typicality we do not support – expectedly, similarly to the majority of implementations.</p>

#### 2.2.1.2 Use cases

URI	Description
GET /albums/@me	query of the actual user's album
GET /albums/{userId}	query of the {userId} user's album

#### 2.2.1.3 Examples

URI	Description
GET /albums/111?count=3	query of the first 3 albums of user with the identifier 111

## 2.2.2 Get album comments

### 2.2.2.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to get the list of comments of an album.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/albums/{userId}/{groupId}?/{albumId}/comments
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: in case of Elder-Spaces only the @self supported (albums of friends can be queried only with exact albumId)</li> <li>• albumId: optional, identifier / identifiers of the album / albums queried (@default – profile pictures, @activity – news flow and mobile pictures)</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of Comment Objects
Comments	

### 2.2.2.2 Use cases

URI	Description
GET /albums/{userId}/{groupId}/{albumId}/comments	query the comments of the album

### 2.2.2.3 Examples

URI	Description
GET /albums/@me/@self/@default/comments	query the comments of the viewer user's profile pictures album

## 2.2.3 Create or modify album

### 2.2.3.1 Description

Method	POST or PUT
Request Header	bool(false)

Synopsis	Use POST to create a new album or modify an existing one.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/albums/{userId}/{groupId}?.*/{albumId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: in case of Elder-Spaces only the @self supported (albums of friends can be queried only with exact albumId)</li> <li>• albumId: optional, identifier / identifiers of the album / albums queried (@default – profile pictures, @activity – news flow and mobile pictures)</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.2.3.2 Use cases

URI	Description
POST /albums/{userId}/{groupId}	create album
POST /albums/{userId}/{groupId}/{albumId}	modify album

### 2.2.3.3 Examples

URI	POST data	Response	Description
POST /albums/@me/@self	{"description": "album description", "title": "album title"}	200 OK	create album by the viewer user
POST /albums/@me/@self/3	{"description": "album description", "title": "album title"}	200 OK	modify album with identifier 3 by the viewer user

## 2.2.4 Create album comment

### 2.2.4.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to create a new comment to an album.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/albums/{userId}/{groupId}?.*/{albumId}/comments
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way</li> </ul>

	<ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> <li>• groupId: in case of Elder-Spaces only the @self supported (albums of friends can be queried only with exact albumId)</li> <li>• albumId: optional, identifier / identifiers of the album / albums queried (@default – profile pictures, @activity – news flow and mobile pictures)</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.2.4.2 Use cases

URI	Description
POST /albums/{userId}/{groupId}/{albumId}/comments	create an album comment

### 2.2.4.3 Examples

URI	POST data	Response	Description
POST /albums/@me/@self/3/comments	{"text": "text of the comment"}	200 OK	create comment to the album with identifier 3 of the viewer user

## 2.2.5 Delete album

### 2.2.5.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete existing album.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/albums/{userId}/{groupId}?/{albumId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: in case of Elder-Spaces only the @self supported (albums of friends can be queried only with exact albumId)</li> <li>• albumId: optional, identifier / identifiers of the album / albums</li> </ul>

	queried (@default – profile pictures, @activity – news flow and mobile pictures)
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.2.5.2 Use cases

URI	Description
DELETE /albums/{userId}/{groupId}/{albumId}	delete album

### 2.2.5.3 Examples

URI	Response	Description
DELETE /albums/@me/@self/3	200 OK	delete album with identifier 3 by the viewer user

## 2.2.6 Delete album comment

### 2.2.6.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete an existing album comment.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/albums/{userId}/{groupId}?/{albumId}/comments
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: in case of Elder-Spaces only the @self supported (albums of friends can be queried only with exact albumId)</li> <li>• albumId: optional, identifier / identifiers of the album / albums queried (@default – profile pictures, @activity – news flow and mobile pictures)</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.2.6.2 Use cases

URI	Description
DELETE /albums/{userId}/{groupId}/{albumId}/comments?commentId={commentId}	delete an album comment

### 2.2.6.3 Examples

URI	Response	Description
DELETE /albums/@me/@self/3/comments?commentId=123	200 OK	delete the comment with identifier 123 on the album with identifier 3 of the viewer user

## 2.3 MediaItems service

### 2.3.1 Get photos

#### 2.3.1.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to get the photos of a given album.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/mediaitems/{userId}/{groupId}?/{albumId}/{mediaItemId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: in case of Elder-Spaces the @self supported (albums of friends can also be queried through the owner only)</li> <li>• albumId: identifier of the album queried</li> <li>• mediaItemId: identifier of the picture queried</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of MediaItem Objects
Comments	thumbnailUrl points to a 128x128 size picture

#### 2.3.1.2 Use cases

URI	Description
GET /mediaitems/{userId}/@self/{albumId}	query of the pictures included in {userId} user's {albumId} album
GET /mediaitems/@me/@self/{albumId}	query of the pictures included in the actual user's {albumId} album

URI	Description
GET /mediaitems/{userId}/@self/@default/{mediaItemid}	query of the {mediaItemId} picture included in {userId} user's profile album

### 2.3.1.3 Examples

URI	Description
GET /mediaitems/1:elderspaces.iwiw.hu/@self/222?count=10	query of the first 10 pictures included in the 111 {userId} user's 222 {albumId} album

## 2.3.2 Get photo comments

### 2.3.2.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to get the comments of a photo.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/mediaitems/{userId}/{groupId}/{albumId}/{mediaItemId}/comments
Parameters	<ul style="list-style-type: none"> <li>• <b>userId</b>: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• <b>groupId</b>: in case of Elder-Spaces the @self supported (albums of friends can also be queried through the owner only)</li> <li>• <b>albumId</b>: identifier of the album queried</li> <li>• <b>mediaItemId</b>: identifier of the picture queried</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of Comment Objects
Comments	

### 2.3.2.2 Use cases

URI	Description
GET /mediaitems/{userId}/@self/{albumId}/{mediaItemId}/comments	query of comments concerning the {mediaItemId} picture being in the {userId} user's {albumId} album

### 2.3.2.3 Examples

URI	Description
GET /mediaitems/@me/@self/3/123/comments	query of comments concerning the picture with identifier 123 being in the viewer user's album with identifier 3

## 2.3.3 Upload or modify photo

### 2.3.3.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to upload a new photo or modify an existing one.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/ mediaitems/{userId}/{groupId}?/{albumId}/{mediaItemId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: in case of Elder-Spaces the @self supported (albums of friends can also be queried through the owner only)</li> <li>• albumId: identifier of the album queried</li> <li>• mediaItemId: identifier of the picture queried</li> </ul>
Response Formats	JSON, XML
Response Type	HTTP status code
Comments	

### 2.3.3.2 Use cases

URI	Description
POST /mediaitems/{userId}/{groupId}/{albumId}	upload media element
POST /mediaitems/{userId}/{groupId}/{albumId}/{mediaItemId}	modify media element

### 2.3.3.3 Examples

URI	POST data	Response	Description
POST /mediaitems/@me/@self/3	{ "type": "image", "url": "http://dummy.net/image", "title": "image title", "thumbnailUrl":	200 OK	upload image to the album with identifier 3 of the viewer user

URI	POST data	Response	Description
	"http://dummy.net/image_tn", "mimeType": "image/jpeg" }		
POST /mediaitems/@me/@self/3/123	{ "type": "image", "url": "http://dummy.net/image", "title": "image title", "thumbnailUrl": "http://dummy.net/image_tn", "mimeType": "image/jpeg" }	200 OK	modify image with identifier 123 in the album with identifier 3 of the viewer user

## 2.3.4 Create photo comment

### 2.3.4.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to create a new comment to a photo.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/mediaitems/{userId}/{groupId}/{albumId}/{mediaItemId}/comments
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: in case of Elder-Spaces the @self supported (albums of friends can also be queried through the owner only)</li> <li>• albumId: identifier of the album queried</li> <li>• mediaItemId: identifier of the picture queried</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.3.4.2 Use cases

URI	Description
POST /mediaitems/{userId/@self/{albumId}/{mediaItemId}/comments{Comment}	new comment to the {mediaItemId} picture being in the {userId} user's {albumId} album

### 2.3.4.3 Examples

URI	POST data	Response	Description
POST	{ "text": "text of the	200 OK	create a comment to picture with

URI	POST data	Response	Description
/mediaitems/@me/@self/3/123/comments	comment"		identifier 123 in the album with identifier 3

## 2.3.5 Delete photo

### 2.3.5.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete an existing photo.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/mediaitems/{userId}/{groupId}?/{albumId}/{mediaItemId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: in case of Elder-Spaces the @self supported (albums of friends can also be queried through the owner only)</li> <li>• albumId: identifier of the album queried</li> <li>• mediaItemId: identifier of the picture queried</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.3.5.2 Use cases

URI	Description
DELETE /mediaitems/{userId}/{groupId}/{albumId}/{mediaItemId}	delete media element

### 2.3.5.3 Examples

URI	Response	Description
DELETE /mediaitems/@me/@self/3/1234	200 OK	delete media with identifier 1234 from the album with identifier 3 of the viewer user

## 2.3.6 Delete photo comment

### 2.3.6.1 Description

Method	DELETE
Request	bool(false)

Header	
Synopsis	Use DELETE to create a new comment to a photo.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/mediaitems/{userId}/{groupId}/{albumId}/{mediaItemId}/comments
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: in case of Elder-Spaces the @self supported (albums of friends can also be queried through the owner only)</li> <li>• albumId: identifier of the album queried</li> <li>• mediaItemId: identifier of the picture queried</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.3.6.2 Use cases

URI	Description
DELETE /mediaitems/{userId}/{groupId}/{albumId}/{mediaItemId}/comments?commentId={commentId}	delete picture comment

### 2.3.6.3 Examples

URI	Response	Description
DELETE /mediaitems/@me/@self/3/123/comments?commentId=1234	200 OK	delete comment with identifier 1234 of the picture with identifier 123 in the album with identifier 3 of the viewer user

## 2.4 Activities Services

### 2.4.1 Get activities

#### 2.4.1.1 Description

Method	GET
Request Header	"Content-Type: application/json"
Synopsis	Use GET to get the user's or his/her friends' activity feed.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/activities/{userId}/{groupId}/{activityId}*

Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: one of the followings <ul style="list-style-type: none"> <li>○ @self: user's news flow</li> <li>○ @friends: friends' news flow; can only be accessed by the viewer</li> </ul> </li> <li>• activityId: activity identifier</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of Activity Objects
Comments	<p>In case of Elder-Spaces, for performance reasons, it is necessary to introduce some restrictions. The queries of friends can always concern a given day. In accordance with this we introduce a HTTP parameter named day, transferring which at calling it can be determined what day the request concerns. The paging function introduced by the OpenSocial can only be accessed within the day. The day shall be specified using the yyyyymmdd format, in accordance with the Middle European time (CET and CEST, respectively according to winter / summer time).</p> <p>Another HTTP parameter named fromID is also introduced through which paging can comfortably be continued from; by passing the ID of the last activity got back at the previous call. This also determines the day but operates only within the day! If there is no other action in the given day then we give back an empty list. In this case the client is required to initiate a query for the next day.</p> <p>Ordering is not supported, it happens always in time order, backwards.</p> <p>Filters are not supported and the activities set in the system globally will only be given back out of the activities.</p> <p>The fields requested by the parameter "fields" also concern the mediaItem being in the activity.</p>

### 2.4.1.2 Use cases

URI	Description
GET /activities/{userId}/@self	access to the {userId} user's news flow elements
GET /activities/@me/@self	access to the actual user's news flow elements
GET / activities /@me/@friends	news flow of the actual user's friends

### 2.4.1.3 Examples

URI	Description
GET /activities/@me/@friends?count=10	10 latest happenings of the viewer's friends
GET /activities/@me/@friends?count=10&fromId={activityId}	it gives back 10 happenings of the viewer's friends prior to the {activityId}, in the same day
GET /activities/@me/@friends?count=10&day=20090901	it gives back the first 10 news flow elements of the viewer's friends happened on 1st of September.

## 2.4.2 Create activity

### 2.4.2.1 Description

Method	POST or PUT
Request Header	"Content-Type: application/json"
Synopsis	Use POST to create an activity to the user's feed.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/activities/{userId}/{groupId}/{activityId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: one of the followings <ul style="list-style-type: none"> <li>○ @self: user's news flow</li> <li>○ @friends: friends' news flow; can only be accessed by the viewer</li> </ul> </li> <li>• activityId: activity identifier</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	<p>For message sending the Activity object including the message must be POSTed into the user's news flow.</p> <p>POST {ROOT}/activities/{userId}/@self</p> <p>The message to be displayed has to get into activity title field while the text intended to the activity into the body, and it can include also 1 MediaItem. The MediaItem can point to external URL but there is a possibility to upload a picture too. In this case, in accordance with the two step upload of REST protocol, first the picture has to be uploaded and the MediaItem got back there has to be set in the Activity. The picture has to be POSTed to the next URL.</p> <p>POST {ROOT}/mediaitems/{userId}</p>

	<p>As a result the picture is put into a special album of the user. Activity is not generated about the upload.</p> <p>The url and thumbnail Url field is compulsory in the MediaItem. The Activity must include message or MediaItem.</p>
--	--

### 2.4.2.2 Use cases

URI	Description
POST /activities/{userId}/@self	create an activity to the user's activity feed

### 2.4.2.3 Examples

URI	POST data	Response	Description
POST /activities/{userId}/@self	{ "title": "text of the activity title", "body": "text of the activity body" }	200 OK	create an activity to the user's activity feed

## 2.4.3 Delete activity

### 2.4.3.1 Description

Method	DELETE
Request Header	"Content-Type: application/json"
Synopsis	Use DELETE to delete an existing activity from the user's feed.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/activities/{userId}/{groupId}/{activityId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• groupId: one of the followings <ul style="list-style-type: none"> <li>○ @self: user's news flow</li> <li>○ @friends: friends' news flow; can only be accessed by the viewer</li> </ul> </li> <li>• activityId: activity identifier</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.4.3.2 Use cases

URI	Description
DELETE /activities/{userId}/@self/{activityId}	delete activity

### 2.4.3.3 Examples

URI	Response	Description
DELETE /activities/{userId}/@self/1234	200 OK	delete the activity with identifier 1234 from the user's activity feed

## 2.5 Messages service

### 2.5.1 Get messages

#### 2.5.1.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to get the user's inbox or outbox messages.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/messages/{userId}+/{msgCollId}?/{messageId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• msgCollId: only @inbox, @outbox are supported</li> <li>• messageId: identifier of a given message</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of Message Objects.
Comments	<ul style="list-style-type: none"> <li>• The special identifiers distinguishing the incoming / outgoing messages are include in the collectioIds field of the Message object: @inbox and @outbox, respectively</li> <li>• The recipients is supported only in case of query for a concrete element, it is not supported in case of list.</li> <li>• Only one a single messageId is supported in case of Elder-Spaces.</li> </ul>

#### 2.5.1.2 Use cases

URI	Description
GET /messages/@me/@inbox	access to the personal messages of the actual user; userId can also be given instead of the

URI	Description
	@me, but that one can only be the viewer
GET /messages/@me/@inbox/{messageId}	access to the {messageId} personal message in the user's inbox

### 2.5.1.3 Examples

URI	Description
GET /messages/@me/@inbox?count=10	query of user's first 10 incoming messages
GET /messages/@me/@outbox?count=10	query of user's first 10 outgoing messages

## 2.5.2 Create and send message

### 2.5.2.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to send a personal message to dedicated recipients.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/messages/{userId}+/{msgCollId}?/{messageId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• msgCollId: only @inbox, @outbox are supported</li> <li>• messageId: identifier of a given message</li> <li>• Message: Message Object</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	<ul style="list-style-type: none"> <li>• The special identifiers distinguishing the incoming / outgoing messages are include in the collectioIds field of the Message object: @inbox and @outbox, respectively</li> </ul>

### 2.5.2.2 Use cases

URI	Description
POST /messages/@me?{Message}	sends the Message object received to the specified addressees

### 2.5.2.3 Examples

URI	POST data	Response	Description
POST /messages/@me	{ "title": "text of the message title", "body": "text of the message"	200 OK	send a message to the recipient users

URI	POST data	Response	Description
	body", "recipients": [ "13929796:elderspaces.iwiw.hu", "13913537:elderspaces.iwiw.hu"], "type": "privateMessage", "msgCollId": "@inbox" }		

## 2.5.3 Delete message

### 2.5.3.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete a personal message from the user's inbox or outbox.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/messages/{userId}+/{msgCollId}?/{messageId}*
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• msgCollId: only @inbox, @outbox are supported</li> <li>• messageId: identifier of a given message</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	<ul style="list-style-type: none"> <li>• The special identifiers distinguishing the incoming / outgoing messages are include in the collectioIds field of the Message object: @inbox and @outbox, respectively</li> <li>• Only one a single messageId is supported in case of Elder-Spaces.</li> </ul>

### 2.5.3.2 Use cases

URI	Description
DELETE /messages/@me/@inbox/{messageId}	deleting the actual user's personal messages; userId can also be given instead of the @me, but that one can only be the viewer

### 2.5.3.3 Examples

URI	Response	Description
DELETE /messages/@me/@inbox/1234	200 OK	delete the message with identifier 1234 from the user's inbox

## 2.6 Events service

### 2.6.1 Get events (includes event function notifications)

#### 2.6.1.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to query events to which the user has RSVP-and (rsvps of INVITED, YES, NO, MAYBE status).
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/events/{userId}/{eventId}?
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• eventId: event identifier</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Event Object or Block of Event Objects
Comments	<ul style="list-style-type: none"> <li>• In case no eventId is given: list of such events concerning which the rsvp status of the user is YES, NO, MAYBE or INVITED and the ending date of them is later than the actual date. The list is ordered in accordance with the start date in ascending order, and the OpenSocial standard paging parameters can be used.</li> <li>• In case the eventId is given: detailed data of the actual event.</li> </ul> <p>As a default the values of the id, name and shortDescription fields are handed over. Further fields can be requested by means of the fields parameter.</p>

#### 2.6.1.2 Use cases

URI	Description
GET /events/{userId}	Those events to which the user has rsvp-ed (rsvps of INVITED, YES, NO, MAYBE status)

#### 2.6.1.3 Examples

URI	Description
GET /events/@me	Those events to which the actual user has rsvp-ed (rsvps of INVITED, YES, NO, MAYBE status)

URI	Description
GET /events/@me?startIndex=10&count=10	10 pieces counted from the 20th, out of the events corresponding to the above.
GET /events/@me/1234?fields=description	data of the event of 1234 identifier, description as well in addition to the default fields.
GET /events/@me?rsvp=INVITED	get the event invitations of the actual user (=event function notifications)

## 2.6.2 Search events

### 2.6.2.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to search events with a given keyword.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/events/search?keyword=...
Parameters	<ul style="list-style-type: none"> <li>keyword: required parameter. free text search in the name, description and address fields. If the keyword parameter is not given or less than 3 characters, the response is 400 BAD_REQUEST.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of Event Objects in the order of start dates. OpenSocial standard pagination parameters can be used in the result list.
Comments	<ul style="list-style-type: none"> <li>dateFrom, dateTo: query events within the given dates. Allowable date formats: YYYY-MM-DD, YYYY-MM, YYYY. In case of invalid date format, the response is 400 BAD_REQUEST.</li> <li>category: category of the event. Allowable values: BIRTHDAY, NAMEDAY, PARTY, CONCERT, CINEMA, THEATRE, PROGRAM, WEDDING, MEETING, SPORTEVENT, FESTIVAL, CELEBRATION, OTHER. In case of invalid value, the response is 400 BAD_REQUEST.</li> <li>address: free text search in the address field</li> <li>addressLatitude, addressLongitude: search events in 10 000 meters radius. Formal requirements: Float. In case of invalid value, the response is 400 BAD_REQUEST.</li> </ul>

### 2.6.2.2 Use cases

URI	Description
GET /events/search?keyword={keyword}	free text search

### 2.6.2.3 Examples

URI	Description
GET /events/search?keyword=csom	free text search with “csom” keyword.

## 2.6.3 Get event activities

### 2.6.3.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to query the activity feed of a given event or a given activity from the feed.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/events/{eventId}/activities/{activityId}?
Parameters	<ul style="list-style-type: none"> <li>• eventId: required parameter. If it's not given, the response is 400 BAD_REQUEST.</li> <li>• activityId: only allowed in case of DELETE request, but it's required in this case. Otherwise the response is 400 BAD_REQUEST.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Activity Object or block of Activity Objects. OpenSocial standard pagination parameters can be used in the result list.
Comments	

### 2.6.3.2 Use cases

URI	Description
GET /events/{eventId}/activities	query the activity feed of an event
GET /events/{eventId}/activities/{activityId}	query an activity from the activity feed of the given event

### 2.6.3.3 Examples

URI	Description
GET /events/1/activities	query the activity feed of the event with identifier 1
GET /events/1/activities/1234	query the activity with identifier 1234 from the activity feed of the event with identifier 1

## 2.6.4 Create or modify event

### 2.6.4.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to create a new event or modify an existing one.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/events/{userId}/{eventId}?
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• eventId: event identifier</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	<p><u>Event parameters:</u></p> <ul style="list-style-type: none"> <li>• eventId: if it is given, it causes event modification, else event creation. Formal requirements: Integer. 400 BAD_REQUEST response in case of error.</li> <li>• name: required parameter. Formal requirements: length&lt;=100 characters. 400 BAD_REQUEST response in case of error.</li> <li>• description: Formal requirements: &lt;= 4000 characters. 400 BAD_REQUEST response in case of error.</li> <li>• shortDescription: Formal requirements: &lt;= 50 characters. 400 BAD_REQUEST response in case of error.</li> <li>• category: Allowable values: BIRTHDAY, NAMEDAY, PARTY, CONCERT, CINEMA, THEATRE, PROGRAM, WEDDING, MEETING, SPORTEVENT, FESTIVAL, CELEBRATION, OTHER. 400 BAD_REQUEST response in case of error.</li> <li>• location: location of the event</li> <li>• address: address of the event</li> <li>• addressLatitude, addressLongitude: if it's given, it must be Float. 400 BAD_REQUEST response in case of error.</li> </ul>

### 2.6.4.2 Use cases

URI	Description
POST /events/@me	create a new event
POST /events/@me/1234	modify the viewer user's event with identifier 1234

### 2.6.4.3 Examples

URI	POST data	Response	Description
POST /events/@me	{ "name": "test name", "description": "test description", "shortDescription": "test shortDescription", "category": "MEETING", "location": "Budapest" }	200 OK	create a new event
POST /events/@me/1234	{ "name": "modified test name" }	200 OK	modify the viewer user's event with identifier 1234

## 2.6.5 Create event activity

### 2.6.5.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to create a new activity in the activity feed of a given club.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/events/{EventId}/activities/{activityId}?
Parameters	<ul style="list-style-type: none"> <li>• eventId: required parameter. If it's not given, the response is 400 BAD_REQUEST.</li> <li>• activityId: only allowed in case of DELETE request, but it's required in this case. Otherwise the response is 400 BAD_REQUEST.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.6.5.2 Use cases

URI	Description
POST /events/{eventId}/activities	create a new activity in the activity feed of a given event

### 2.6.5.3 Examples

URI	POST data	Response	Description
POST /events/1/acitivites	{ "body": "Zsír", "mediaItems": [], "title": "Erik " }	200 OK	create an activity to the activity feed of the event with identifier 1

## 2.6.6 Delete event

### 2.6.6.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete an existing event.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/events/{userId}/{eventId}?
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• eventId: event identifier</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	eventId: required. If there's no event with the given identifier, the response is 400 BAD_REQUEST.

### 2.6.6.2 Use cases

URI	Description
DELETE /events/@me/{eventId}	delete event

### 2.6.6.3 Examples

URI	Response	Description
DELETE /events/@me/1234	200 OK	delete event with identifier 1234

## 2.6.7 Delete event activity

### 2.6.7.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete an activity from the activity feed of a given event.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/events/{eventId}/activities/{activityId}?
Parameters	<ul style="list-style-type: none"> <li>• eventId: required parameter. If it's not given, the response is 400 BAD_REQUEST.</li> </ul>

	<ul style="list-style-type: none"> <li>activityId: only allowed in case of DELETE request, but it's required in this case. Otherwise the response is 400 BAD_REQUEST.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.6.7.2 Use cases

URI	Description
DELETE /events/{eventId}/activities/{activityId}	delete activity from the activity feed of a event

### 2.6.7.3 Examples

URI	Response	Description
DELETE /events/1/activities/00e8aa7450a1002d1e12b287	200 OK	delete the activity with the given identifier from the activity feed of the event with identifier 1

## 2.7 RSVP service

### 2.7.1 Get user's RSVP status of an event

#### 2.7.1.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to get the user's RSVP status (INVITED, YES, NO, MAYBE) of an event
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/rsvp/{eventId}/{userId}?
Parameters	<ul style="list-style-type: none"> <li>userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>user identifier</li> <li>@me symbol: means: the user assigned to the given request</li> </ul> </li> <li>eventId: event identifier to which the query refers</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	RSVP record
Comments	

### 2.7.1.2 Use cases

URI	Description
GET /rsvp/{eventId}/{userId}	get the user's RSVP status (INVITED, YES, NO, MAYBE) of an event

### 2.7.1.3 Examples

URI	Description
GET /rsvp/1234/@me	rsvp record of the viewer user related to the event of 1234 identifier

## 2.7.2 Get RSVP statuses of an event

### 2.7.2.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to get the users who RSVP-ed an event and also their statuses.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/rsvp/{eventId}/{userId}?
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• eventId: event identifier to which the query refers</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of RSVP records
Comments	

### 2.7.2.2 Use cases

URI	Description
GET /rsvp/{eventId}	get rsvp records related to the event of 1234 identifier

### 2.7.2.3 Examples

URI	Description
GET /rsvp/1234?status=YES	YES status rsvp records related to the event of 1234 identifier

## 2.7.3 Invite, join, maybe attend or decline user to an event

### 2.7.3.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to invite, join, maybe attend or decline user to an event.
URI	<a href="http://api.elderspaces.iwiw.hu/social/papi/rest/rsvp/{eventId}/{userId}?">http://api.elderspaces.iwiw.hu/social/papi/rest/rsvp/{eventId}/{userId}?</a>
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• eventId: event identifier to which the query refers</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	Allowable status values: INVTED, YES, NO, MAYBE

### 2.7.3.2 Use cases

URI	Description
POST /rsvp/{eventId}/{userId}	add or modify an RSVP record concerning the given event

### 2.7.3.3 Examples

URI	POST data	Response	Description
POST /rsvp/1234/@me	{"status":"YES"}	200 OK	The viewer user gave a YES rsvp-answer to the event of 1234 identifier

## 2.8 Clubs service

Implements the Groups functionality of the site.

### 2.8.1 Get clubs (includes groups function notifications)

#### 2.8.1.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to get the joined clubs of a user or detailed data of a given club.
URI	<a href="http://api.elderspaces.iwiw.hu/social/papi/rest/clubs/{userId}/{clubId">http://api.elderspaces.iwiw.hu/social/papi/rest/clubs/{userId}/{clubId</a>

	}?
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, in a non mixable way <ul style="list-style-type: none"> <li>○ user identifier</li> <li>○ @me symbol: means: the user assigned to the given request</li> </ul> </li> <li>• clubId: club identifier to which the query refers</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Club Object or Block of Club Objects
Comments	As a default the values of the id, name and shortDescription fields are handed over. Further fields can be requested by means of the parameter “fields”.

### 2.8.1.2 Use cases

URI	Description
GET /clubs/{userId}/{clubId}	query a given club of the user
GET /clubs/{userId}	query the joined clubs of the user

### 2.8.1.3 Examples

URI	Description
GET /clubs/@me/9200	query the club with identifier 9200
GET /clubs/@me	query the joined clubs of the viewer user
GET /clubs/@me?startIndex=10&count =10	filter 10 clubs from the 10. result of the previous query
GET /clubs/@me?rsvp=INVITED	get the event invitations of the actual user (=groups function notifications)

## 2.8.2 Search club

### 2.8.2.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to search clubs with a given keyword.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/clubs/search?keyword=...
Parameters	<ul style="list-style-type: none"> <li>• keyword: required parameter. free text search in the name, description and address fields. If the keyword parameter is not given or less than 3 characters, the response is 400 BAD_REQUEST.</li> </ul>
Response Formats	JSON, XML, ATOM

Response Type	Block of Club Objects
Comments	

### 2.8.2.2 Use cases

URI	Description
GET /clubs/search?keyword=...	free text search in clubs with the given keyword parameter

### 2.8.2.3 Examples

URI	Description
GET /clubs/search?keyword=csom	search clubs with the given keyword parameter

## 2.8.3 Get club activities

### 2.8.3.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to query the activity feed of a given club or a given activity from the feed.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/clubs/{clubId}/activities/{activityId}?
Parameters	<ul style="list-style-type: none"> <li>clubId: required parameter. If it's not given, the response is 400 BAD_REQUEST.</li> <li>activityId: only allowed in case of DELETE request, but it's required in this case. Otherwise the response is 400 BAD_REQUEST.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Activity Object or block of Activity Objects. OpenSocial standard pagination parameters can be used in the result list.
Comments	

### 2.8.3.2 Use cases

URI	Description
GET /clubs/{clubId}/activities	query the activity feed of a club
GET /clubs/{clubId}/activities/{activityId}	query an activity from the activity feed of the given club

### 2.8.3.3 Examples

URI	Description
GET /clubs/1/acitivites	query the activity feed of the club with identifier 1
GET /clubs/1/acitivites/1234	query the activity with identifier 1234 from the activity feed of the club with identifier 1

## 2.8.4 Create or modify club

### 2.8.4.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to create a new club or modify an existing one.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/clubs/{userId}/{clubId}?
Parameters	<ul style="list-style-type: none"> <li>• userId: one of the followings: <ul style="list-style-type: none"> <li>○ user identifier (only the viewer user id is supported – in case of handing over different id 403 FORBIDDEN error gets triggered)</li> <li>○ @me symbol: the user assigned to the given request</li> </ul> </li> <li>• clubId: club identifier</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	<p><u>club parameters:</u></p> <ul style="list-style-type: none"> <li>• clubId: if it is given, it causes club modification, else club creation. Formal requirements: Integer. 400 BAD_REQUEST response in case of error.</li> <li>• name: required parameter. Formal requirements: length&lt;=100 characters. 400 BAD_REQUEST response in case of error.</li> <li>• description: Formal requirements: &lt;= 4000 characters. 400 BAD_REQUEST response in case of error.</li> <li>• shortDescription: Formal requirements: &lt;= 50 characters. 400 BAD_REQUEST response in case of error.</li> <li>• category: category of the club. Allowable values: ANY, ANIMAL, CAR, COMPANY, FAMILY, FASHION, HEALTH, LIFESTYLE, HOBBY, SCHOOL, IT, GAME, GASTRONOMY, CULTURE, ART, COUNTRY, RELATIONSHIP, SPORT, SERVICE, ENTERTAINMENT, SOCIETY, SCIENCE, TRAVEL, BUSINESS, RELIGION, CITY, MUSIC, OTHER. 400 BAD_REQUEST response in case of error.</li> </ul>

### 2.8.4.2 Use cases

URI	Description
POST /clubs/{userId}	create a new club
POST /clubs/{userId}/{clubId}	modify an existing club

### 2.8.4.3 Examples

URI	POST data	Response	Description
POST /clubs/@me	{"name":"test name","description":"test description","shortDescription":"test shortDescription","category":"SPORT"}	200 OK	create a new event
POST /clubs/@me/1234	{"name":"modified test name"}	200 OK	modify the name field of the viewer user's club with identifier 1234

## 2.8.5 Create club activity

### 2.8.5.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to create a new activity in the activity feed of a given club.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/clubs/{clubId}/activities/{activityId}?
Parameters	<ul style="list-style-type: none"> <li>clubId: required parameter. If it's not given, the response is 400 BAD_REQUEST.</li> <li>activityId: only allowed in case of DELETE request, but it's required in this case. Otherwise the response is 400 BAD_REQUEST.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.8.5.2 Use cases

URI	Description
POST /clubs/{clubId}/activities	create a new activity in the activity feed of a given club

### 2.8.5.3 Examples

URI	POST data	Response	Description
POST /clubs/1/acitivites	{ "body": "Zsír", "mediaItems": [], "title": "Erik " }	200 OK	create an activity to the activity feed of the club with identifier 1

## 2.8.6 Delete club

### 2.8.6.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete an existing club.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/clubs/{userId}/{clubId}?
Parameters	<ul style="list-style-type: none"> <li>• userId: one of the followings: <ul style="list-style-type: none"> <li>○ user identifier (only the viewer user id is supported – in case of handing over different id 403 FORBIDDEN error gets triggered)</li> <li>○ @me symbol: the user assigned to the given request</li> </ul> </li> <li>• clubId: club identifier</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.8.6.2 Use cases

URI	Description
DELETE /clubs/{userId}/{clubId}	delete club

### 2.8.6.3 Examples

URI	Response	Description
DELETE /clubs/@me/1234	200 OK	delete the viewer user's club with the identifier 1234

## 2.8.7 Delete club activity

### 2.8.7.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete an activity from the activity feed of a given club.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/clubs/{clubId}/activities/{activityId}?
Parameters	<ul style="list-style-type: none"> <li>clubId: required parameter. If it's not given, the response is 400 BAD_REQUEST.</li> <li>activityId: only allowed in case of DELETE request, but it's required in this case. Otherwise the response is 400 BAD_REQUEST.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	

### 2.8.7.2 Use cases

URI	Description
DELETE /clubs/{clubId}/activities/{activityId}	delete activity from the activity feed of a club

### 2.8.7.3 Examples

URI	Response	Description
DELETE /clubs/1/activities/00e8aa7450a1002d1e12b287	200 OK	delete the activity with the given identifier from the activity feed of the club with identifier 1

## 2.9 ClubJoin service

### 2.9.1 Get the user's club join record (YES / INVITED)

#### 2.9.1.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to get the club join status of a given user.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/clubjoin/{clubId}/{userId}?

Parameters	<ul style="list-style-type: none"> <li>• userId: one of the followings: <ul style="list-style-type: none"> <li>○ user identifier (only the viewer user id is supported – in case of handing over different id 403 FORBIDDEN error gets triggered)</li> <li>○ @me symbol: the user assigned to the given request</li> </ul> </li> <li>• clubId: club identifier to which the query refers</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Join record
Comments	

### 2.9.1.2 Use cases

URI	Description
GET /clubjoin/{clubId}/{userId}	query the viewer user's join record of the given club

### 2.9.1.3 Examples

URI	Description
GET /clubjoin/1234/@me	query the viewer user's join record of the club with identifier 1234

## 2.9.2 Get club members or the list of invited people

### 2.9.2.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to get club members or the list of invited people.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/clubjoin/{clubId}
Parameters	<ul style="list-style-type: none"> <li>• clubId: club identifier to which the query refers</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	Block of join records.
Comments	The result list is in the descent order of join dates.

### 2.9.2.2 Use cases

URI	Description
GET /clubjoin/{clubId}	query the join records of the given club

### 2.9.2.3 Examples

URI	Description
GET /clubjoin/1234?status=YES	query the members of the club with identifier 1234
GET /clubjoin/1234?status=INVITED	query the list of invited people of the club with identifier 1234

## 2.9.3 Invite or join user to a club

### 2.9.3.1 Description

Method	POST or PUT
Request Header	bool(false)
Synopsis	Use POST to invite or join a user to a club.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/clubjoin/{clubId}
Parameters	<ul style="list-style-type: none"> <li>• userId: one of the followings: <ul style="list-style-type: none"> <li>○ user identifier (only the viewer user id is supported – in case of handing over different id 403 FORBIDDEN error gets triggered)</li> <li>○ @me symbol: the user assigned to the given request</li> </ul> </li> <li>• clubId: club identifier to which the query refers</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	HTTP status code
Comments	<ul style="list-style-type: none"> <li>• The given join status creates or modify the join record. If no join status is given, the response is 400 BAD_REQUEST.</li> <li>• Allowable values of join status: YES, INVITED</li> </ul>

### 2.9.3.2 Use cases

URI	Description
POST /clubjoin/{clubId}/{userId}	join or invite the user to the given club

### 2.9.3.3 Examples

URI	POST data	Response	Description
POST /clubjoin/1234/@me	{"status": "YES"}	200 OK	the viewer user is joining to the club with identifier 1234
POST /clubjoin/1234/1	{"status": "INVITED"}	200 OK	invite user with identifier 1 to the club with identifier 1234

## 2.9.4 Delete user from a club

### 2.9.4.1 Description

Method	DELETE
Request Header	bool(false)
Synopsis	Use DELETE to delete user from a club.
URI	http://api.elderspaces.iwiw.hu/social/papi/rest/clubjoin/{clubId}
Parameters	<ul style="list-style-type: none"> <li>• userId: one of the followings: <ul style="list-style-type: none"> <li>○ user identifier (only the viewer user id is supported – in case of handing over different id 403 FORBIDDEN error gets triggered)</li> <li>○ @me symbol: the user assigned to the given request</li> </ul> </li> <li>• clubId: club identifier to which the query refers</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	
Comments	

### 2.9.4.2 Use cases

URI	Description
DELETE /clubjoin/{clubId}/{userId}	delete user from a club

### 2.9.4.3 Examples

URI	Response	Description
DELETE /clubjoin/1234/@me	200 OK	delete viewer from the club with identifier 1234

## 2.10 Cognitive Recommendation Services (CYBION)

### 2.10.1 Recommendation of friends

Method	GET
Request Header	bool(false)
Synopsis	Use GET to obtain recommendations of friends, users of the platform, based on common interests or activities
URI	<a href="http://gaia.cybion.it/elderspaces/rest/recommendations/friends/{userId}/">http://gaia.cybion.it/elderspaces/rest/recommendations/friends/{userId}/</a>
Parameters	<ul style="list-style-type: none"> <li>• <code>userId</code>: is one of the followings, cannot be mixed <ul style="list-style-type: none"> <li>○ <code>@me</code> symbol: means the user assigned to the given request</li> </ul> </li> </ul> <p>The <code>{userId}</code> specification is compulsory. If it is missing, then the system will return an error message.</p>
Media Types	JSON, XML, ATOM
Media Type Description	<pre> json: {"startIndex":0, "totalResults":20, "entry":[   {     "id":"104:elderspaces.iwiw.hu",     "displayName":"ElfogadĂł DezsĂł",     "thumbnailUrl":"http://thn1.elderspaces.iwiw.hu/0101//user/00/00/00/10/4/user_104_1238767117477_tn1",   },   {     "id":"110:elderspaces.iwiw.hu",     "displayName":"Maria Orbat",     "thumbnailUrl":"http://thn1.elderspaces.iwiw.hu/0101//user/00/00/00/11/0/user_110_12387671174323_tn1",   },   ...   {   {     "id":"112:elderspaces.iwiw.hu",     "displayName":"Marko Dienig",     "thumbnailUrl":"http://thn1.elderspaces.iwiw.hu/0101//user/00/00/00/11/2/user_112_1238 </pre>

	<pre> 76711432741_tn1",   } ] } </pre>
Comments	<ul style="list-style-type: none"> <li>• as default the id, displayName and thumbnailUrl image fields are given back.</li> <li>• we give back of course only users which allowed the recommended user, in accordance with the Elder-Spaces authorisation levels: private / public data.</li> <li>• If the user isn't profiled yet an error message is returned</li> </ul>

## 2.10.2 Recommendations of events

Method	GET
Request Header	bool(false)
Synopsis	Use GET to obtain recommendations of events, events of real life created in the platform, based on interests, activities and participating friends
URI	http://gaia.cybion.it/elderspaces/rest/recommendations/events/{userId}/
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, cannot be mixed <ul style="list-style-type: none"> <li>○ @me symbol: means the user assigned to the given request</li> </ul> </li> </ul> <p>The {userId} specification is compulsory. If it is missing, then the system will return an error message.</p>
Media Types	JSON, XML, ATOM
Media Type Description	<pre> json: {"startIndex":0, "totalResults":2, "entry":[ { "id":"1008", "shortDescription":"nagyon alszik", "name":"ALBUMTESZTELO_esemeny" }, ... }, {"id":"892", "shortDescription":"ajanlooom", </pre>

	<pre>"name": "2 hetes a redesign" }} ] }</pre>
Comments	<ul style="list-style-type: none"> <li>• as default the id, short description and name of the event are given back.</li> <li>• if the user isn't profiled yet an error message is returned</li> </ul>

### 2.10.3 Recommendations of Clubs – Groups of users

Method	GET
Request Header	bool(false)
Synopsis	Use GET to obtain recommendations of Clubs, where clubs are groups of people with a common interest.
URI	http://gaia.cybion.eu/elderspaces/rest/recommendations/clubs/{userId}/
Parameters	<ul style="list-style-type: none"> <li>• userId: is one of the followings, cannot be mixed <ul style="list-style-type: none"> <li>○ @me symbol: means the user assigned to the given request</li> </ul> </li> </ul> <p>The {userId} specification is compulsory. If it is missing, then the system will return an error message.</p>
Media Types	JSON, XML, ATOM
Media Type Description	<pre>json: {   "startIndex": 0,   "totalResults": 10,   "entry":   [{     "id": 12342,     "name": "Klub neve",     "description": "Klub leírása",     "shortDescription": "Klub rövid leírása"     "category": "HOBBY"   },   ...,   { id": 12342,     "name": "Old Football lovers",     "description": "Club Old Football lovers is the club of all the old time",     "shortDescription": "All people interested in old style football"     "category": "SPORT"   } }</pre>

Comments	<ul style="list-style-type: none"> <li>• as default the id, the description, the short description and category of the group are given back.</li> <li>• if the user isn't profiled yet an error message is returned</li> </ul>
----------	--

## 2.10.4 Event Listener Service

Method	POST
Request Header	bool(false)
Synopsis	Use POST to obtain ...
URI	<a href="http://gaia.cybion.eu/elderspaces/rest/listen/">http://gaia.cybion.eu/elderspaces/rest/listen/</a>
Parameters	<ul style="list-style-type: none"> <li>• Activity Based on the activitystrea.ms specification <a href="http://activitystrea.ms/">http://activitystrea.ms/</a></li> </ul>
Media Types	JSON, XML, ATOM
Media Type Description	HTTP status code
Comments	<ul style="list-style-type: none"> <li>• Specification of activitystrea.ms are the following: <a href="http://activitystrea.ms/specs/json/1.0/">http://activitystrea.ms/specs/json/1.0/</a></li> <li>• In its simplest form, an <code>activity</code> consists of an <code>actor</code>, a <code>verb</code>, an <code>object</code>, and a <code>target</code>. It tells the story of a person performing an action on or with an object -- "Geraldine posted a photo to her album" or "John shared a video". In most cases these components will be explicit, but they may also be implied.</li> </ul>

## 2.11 Authentication

The Elder-Spaces system implements a 3-legged OAuth 1.0 standard authentication. Due to the standard the user authentication procedure can separate to three steps:

1. Get OAuth requestToken
2. Authorization
3. Get OAuth accessToken

With a valid accessToken the application has the rights to access the user's data and use the API services in the name of the authenticated user.

## 2.11.1 Get requestToken

### 2.11.1.1 Description

Method	GET
Request Header	bool(false)
Synopsis	<p>Use GET to obtain OAuth requestToken to start the login procedure of the user.</p> <p>The Elder-Spaces application obtains an unauthorized Request token by asking the Elder-Spaces system to issue a token. The requestToken's sole purpose is to receive user approval and can only be used to obtain an accessToken.</p>
URI	<a href="http://api.elderspaces.iwiw.hu/social/oauth/requestToken">http://api.elderspaces.iwiw.hu/social/oauth/requestToken</a>
Parameters	<ul style="list-style-type: none"> <li>• <code>oauth_consumer_key</code> : dbf9b717ce309595ec701b705a7bc6a53fcf8319 - The Elder-Spaces application key.</li> <li>• <code>oauth_signature_method</code> : The signature method the Elder-Spaces application used to sign the request. HMAC-SHA1 and RSA-SHA1 formats are supported.</li> <li>• <code>oauth_signature</code> : The signature in the given format.</li> <li>• <code>oauth_timestamp</code> : The timestamp.</li> <li>• <code>oauth_nonce</code> : The uniquely generated random string by the Elder-Spaces system for each request.</li> <li>• <code>oauth_version</code> : 1.0 - The version of the implemented OAuth version.</li> <li>• <code>oauth_callback</code> : (optional) An absolute URL to which the Elder-Spaces system will redirect the User back when the Authorization step is completed.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	OAuth requestToken (unauthorized)
Comments	<ul style="list-style-type: none"> <li>• expire time of the requestToken: 5 minutes</li> <li>• callback URL max 255 characters</li> </ul>

### 2.11.1.2 Use cases

URI	Description
GET /oauth/requestToken	Obtain requestToken.

### 2.11.1.3 Examples

URI	Response	Description
GET	{	Obtain requestToken.

http://api.elderspaces.iwiw.hu/social/oauth/requestToken&oauth_callback=http%3A%2F%2Felderspaces.iwiw.hu%2Flogin&oauth_consumer_key=dbf9b717ce309595ec701b705a7bc6a53fcf8319&oauth_nonce=b288487991b0725ca8ecf9660c581f2b&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1354211950&oauth_version=1.0	"http_code" : 200, "data" : "oauth_token=05c8837b-8b36-44fb-89d9-a05f54197227&oauth_token_secret=c6886141-e768-468c-8fc2-366870766dcb&oauth_callback_confirmed=true", "headers" : false }	
---	---	--

## 2.11.2 Authorization

### 2.11.2.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to authorize the OAuth requestToken. The Elder-Spaces application cannot use the requestToken until it has been authorized by the user.
URI	<a href="http://api.elderspaces.iwiw.hu/social/oauth/authorize.jsp">http://api.elderspaces.iwiw.hu/social/oauth/authorize.jsp</a>
Parameters	<ul style="list-style-type: none"> <li>oauth_token : The requestToken.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	OAuth requestToken (authorized)
Comments	

### 2.11.2.2 Use cases

URI	Description
GET /oauth/authorize.jsp	Authorize requestToken.

### 2.11.2.3 Examples

URI	Response	Description
GET <a href="http://elderspaces.iwiw.hu/pages/auth/authorize.jsp?oauth_token=27a18efe-0ec1-410a-8c3e-09af5cff9a48">http://elderspaces.iwiw.hu/pages/auth/authorize.jsp?oauth_token=27a18efe-0ec1-410a-8c3e-09af5cff9a48</a>	{ "http_code" : 200, "data" : "oauth_token=05c8837b-8b36-44fb-89d9-a05f54197227&oauth_verifier=168841",	Authorize requestToken.

	<code>"headers" : false</code>	
	<code>}</code>	

## 2.11.3 Get accessToken

### 2.11.3.1 Description

Method	GET
Request Header	bool(false)
Synopsis	Use GET to obtain OAuth accessToken. The Elder-Spaces application exchanges the authorized requestToken for an accessToken capable of accessing the user's data and using the API services in the name of the user.
URI	<a href="http://api.elderspaces.iwiw.hu/social/oauth/accessToken">http://api.elderspaces.iwiw.hu/social/oauth/accessToken</a>
Parameters	<ul style="list-style-type: none"> <li>• <code>oauth_consumer_key</code> : dbf9b717ce309595ec701b705a7bc6a53fcf8319 - The Elder-Spaces application key.</li> <li>• <code>oauth_token</code> : The authorized requestToken.</li> <li>• <code>oauth_signature_method</code> : The signature method the Elder-Spaces application used to sign the request. HMAC-SHA1 and RSA-SHA1 formats are supported.</li> <li>• <code>oauth_signature</code> : The signature in the given format.</li> <li>• <code>oauth_timestamp</code> : The timestamp.</li> <li>• <code>oauth_nonce</code> : The uniquely generated random string by the Elder-Spaces system for each request.</li> <li>• <code>oauth_version</code> : 1.0 - The version of the implemented OAuth version.</li> <li>• <code>oauth_verifier</code> : The verification code received from the Elder-Spaces system in the authorization step.</li> </ul>
Response Formats	JSON, XML, ATOM
Response Type	OAuth accessToken
Comments	<ul style="list-style-type: none"> <li>• accessToken expire time: 1 year</li> </ul>

### 2.11.3.2 Use cases

URI	Description
GET /oauth/accessToken	Obtain accessToken.

### 2.11.3.3 Examples

URI	Response	Description
GET http://api.elderspaces.iwiw.hu/social/oauth/accessToken&oauth_consumer_key=dbf9b717ce309595ec701b705a7bc6a53cf8319&oauth_nonce=54ca5f1663d6b3f2f04eb1ae796a9047&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1354211974&oauth_token=05c8837b-8b36-44fb-89d9-a05f54197227&oauth_verifier=168841&oauth_version=1.0	<pre>{   "http_code" : 200,   "data" : "6c4cfba1-6684-4bba-81a5-318d9b101434&amp;oauth_token_secret=0bee31ac-41b1-4e7c-9eca-190b9f8119f8",   "headers" : false }</pre>	Authorize requestToken.