

onnetwork.tv

API

I. API

Introduction:

The document below is created to for the purposes of technical integration between Onnetwork platform and Publisher's internal CMS system in order to enable management of online video materials directly from Publisher's system. This document is strictly technical and shall not be analized by technical people. For bussines information on API integration please contact your relevant bussines person.

The document below describes following elements of Onnetwork's API:

- **General information on passing parameters and commands**
- **Management of the Publisher's websites**
- **Management of video storages**
- **Management of video materials**
- **Management of video playlist**

1. General information

API call URL:

`https://api.onnetwork.tv/api.php`

All input parameters need to be send as POST, secure connection only.

2. Call parameters

Each call needs 2 parameters:

apikey - API access key - users with granted API access rights can generate this key using users account "EDIT" option. API key is secret key and must not be available to read by website users - all request should be server-server calls. The best practice is to create new user with limited rights for API access only. You need to contact system administrator to grant API access right for this user - this is not default granted right.

json - JSON format table containing argument for API call. Field "actionname" contains name of command (case insensitive). All text are UTF8 coded.

Sample of **json** content:

```
{
  "action" : {
    "actionname" : "listwebsites"
  }
}
```

3. Output data format.

Output data are JSON formatted, everytime field "status" contains information about operation status.

It contains "ERROR" if error occurred and additional fields "statusCode" and "statusDescription" with information about error.

Field "status" contain "OK" if operation was completed successfully. Additional field "action" repeats command name for informational purposes. All data returned as operation result are placed in "result" field. Content of "result" field depends on command. Data returned by result field for some commands (listAdCats, listVideoCategories, listWebsiteCategories, listLanguages) can be stored at local storage. We do not remove items from those lists - only add new items. Items not used are still listed and will never be removed.

4. Embedding parameters.

Onnetwork platform use two kinds of embed code parameters:

- short codes **SID**
- full codes **MID**

Full codes contains information about website, video material and playlist (optional) and information about player setting. It should be used to embed videos with unusual for this website setting. Full codes could be obtained by "MODIFY CODE" option or by API call **getMIDCode**.

To change settings of video embedded using full code you need to create new full code and put it on your website.

Short codes contains only information about website, video material and optionally playlist. All setting are taken by player from default setting for website and optionally playlist. Short codes should be used in places where video is embedded in typical way for this website. Changing default setting for websites has immediate effect for all materials embedded using short codes.

Depending of list command and its arguments several kind of short codes are returned:

- default short code - those codes are using default setting for website
- default playlist short code - those codes are using default setting for playlist and are returned by listPlayLists action.
- default playlist short code with starting video - those codes are using default setting for playlist and are returned by listPlayLists action.

Default short code is used for embedding single video material using website default settings. Marked as KOD1 in samples.

Default playlist short code is used for embedding series of video materials called playlists using selected playlist default setting. Marked as KOD2 in samples.

Default playlist short code with starting video is used for embedding video material (member of playlist) with playback of other playlist video when selected video is finished. It uses playlist default setting. Marked as KOD3 in samples.

5. Command "listWebsites"

Command "listWebsites" returns list of websites available for logged user. It has no arguments. See below for sample of request and result.

Argument "json":

```
{
  "action": {
    "actionname": "listwebsites"
  }
}
```

Result:

```
{
  "status": "OK",
  "action": "listwebsites",
  "result": [
    {
      "id": "123",
      "name": "Website about dogs",
      "url": "http:\\\\www.dogs.org"
    },
    {
      "id": "234",
      "name": "Website about kitties",
      "url": "http:\\\\www.kitty.org"
    }
  ]
}
```

Each record contains fields:

id - unique website id (integer)

name - website name (string)

url - website url (string)

6. Command "listStorages"

Command "listStorages" returns list of storages available for user. It has no arguments. See below for sample of request and result.

Argument "json":

```
{
  "action": {
    "actionname": "liststorages"
  }
}
```

Result:

```
{
  "status": "OK",
  "action": "liststorages",
  "result": [
    {
      "id": "2058",
      "name": "Test storage"
    },
    {
      "id": "1316",
      "name": "Soccer videos"
    }
  ]
}
```

Each record contains fields:

id - unique storage id (integer)

name - storage name (string)

7. Command "listAdCats"

Command "listAdCats" returns list of categories of videos for advertisements purposes. It has no arguments. See below for sample of request and result.

Argument "json":

```
{
  "action": {
    "actionname": "listadcats"
  }
}
```

Result:

```
{
  "status": "OK",
  "action": "listadcats",
  "result": [
    {
      "movieadtype": 0,
      "movieadtypename": "Film nie zatwierdzony dla
reklam"
    },
    {
      "movieadtype": 1,
      "movieadtypename": "Film ukazuj\u0105cy
erotyk\u0119"
    },
    {
      "movieadtype": 2,
      "movieadtypename": "Film ukazuj\u0105cy przemoc"
    },
    {
      "movieadtype": 3,
      "movieadtypename": "Film kontrowersyjny"
    },
    {
      "movieadtype": 4,
      "movieadtypename": "Film z wypadku, ukazuj\u0105cy
poszkodowanych"
    },
    {
      "movieadtype": 5,
      "movieadtypename": "Ca\u0142kowicie zakaz reklam"
    },
    {
      "movieadtype": 10,
      "movieadtypename": "Dopuszczony dla wszystkich"
    }
  ]
}
```

```
    reklam"  
  }  
]  
}
```

Each record contains fields:

movieadtype - unique advertisement category
number(integer)

movieadtypename - name or description of
category(string)

8. Command "listVideoCategories"

Command "listVideoCategories" returns list of categories of videos materials. It has no arguments. See below for sample of request and result.

Argument "json":

```
{
  "action": {
    "actionname": "listvideocategories"
  }
}
```

Result:

```
{
  "status": "OK",
  "action": "listvideocategories",
  "result": [
    {
      "videocatid": 0,
      "videocatname": "Nieprzypisana"
    },
    {
      "videocatid": 1,
      "videocatname": "Gaming"
    },
    {
      "videocatid": 2,
      "videocatname": "Ksi\u0105\u017cka"
    },
    {
      "videocatid": 3,
      "videocatname": "Muzyka"
    },
    {
      "videocatid": 4,
      "videocatname": "Polityka"
    },
    {
      "videocatid": 5,
      "videocatname": "Rozrywka"
    },
    {
      "videocatid": 6,
      "videocatname": "Styl \u017cy\u0107"
    },
    {
      "videocatid": 7,

```

```
"videocatname":"Uroda"
},
{
  "videocatid":8,
  "videocatname":"Motoryzacja"
},
{
  "videocatid":9,
  "videocatname":"Edukacja"
},
{
  "videocatid":10,
  "videocatname":"Ekonomia i Gospodarka"
},
{
  "videocatid":11,
  "videocatname":"Kobieta"
},
{
  "videocatid":12,
  "videocatname":"Kulinaria i od\u017cywanie"
},
{
  "videocatid":13,
  "videocatname":""
},
{
  "videocatid":14,
  "videocatname":"Parenting"
},
{
  "videocatid":15,
  "videocatname":"Poradniki"
},
{
  "videocatid":16,
  "videocatname":"Sport"
},
{
  "videocatid":17,
  "videocatname":"Zdrowie"
},
{
  "videocatid":18,
  "videocatname":"Ekologia"
},
{
  "videocatid":19,
  "videocatname":"Popularnonaukowe"
},
{
```

```
        "videocatid":20,
        "videocatname":"Film"
    },
    {
        "videocatid":21,
        "videocatname":"Spo\u0142ecze\u0144stwo"
    },
    {
        "videocatid":22,
        "videocatname":"Prawo"
    },
    {
        "videocatid":23,
        "videocatname":"Kultura i sztuka"
    },
    {
        "videocatid":24,
        "videocatname":"Fitness"
    },
    {
        "videocatid":25,
        "videocatname":"Turystyka"
    },
    {
        "videocatid":26,
        "videocatname":"Technologie"
    },
    {
        "videocatid":27,
        "videocatname":"Informacje"
    },
    {
        "videocatid":28,
        "videocatname":"Moda"
    },
    {
        "videocatid":29,
        "videocatname":"Nauka"
    },
    {
        "videocatid":33,
        "videocatname":"Foto"
    },
    {
        "videocatid":34,
        "videocatname":"Audio, wideo"
    }
    ]
}
```

Each record contains fields:

`videocatid` - unique category number (integer)

`videocatname` - name or description of category(string)

9. Command "listPrivCats"

This command returns a list of defined private categories that the material video can have. This command has no arguments. An example call and its result are given below.

Parametr json:

```
{
  "action": {
    "actionname": "listprivcats"
  }
}
```

Result:

```
{
  "status": "OK",
  "action": "listvideocategories",
  "result": [
    {
      "id": 99,
      "name": "PrivCat1"
    },
    {
      "id": 108,
      "name": "PrivCat2"
    }
  ]
}
```

Each record contains the following information:

id - unique number of the private category of the video

name - the name or description of the private video category.

NOTE: Not every Partner has defined private categories of video content. In this case, a corresponding error is returned.

10. Command "listWebsiteCategories"

Command "listWebsiteCategories" returns list of websites categories. It has no arguments. See below for sample of request and result.

Argument "json":

```
{
  "action": {
    "actionname": "listwebsitecategories"
  }
}
```

Result:

```
{
  "status": "OK",
  "action": "listwebsitecategories",
  "result": [
    {
      "websitecatid": "ab",
      "websitecatname": "biznes, finanse, prawo"
    },
    {
      "websitecatid": "ac",
      "websitecatname": "blogi, fora"
    },
    {
      "websitecatid": "ad",
      "websitecatname": "budownictwo i
nieruchomo\u015bci"
    },
    {
      "websitecatid": "ae",
      "websitecatname": "dla dzieci"
    },
    {
      "websitecatid": "af",
      "websitecatname": "e-commerce, og\u0142oszenia i
oferty"
    },
    {
      "websitecatid": "ag",
      "websitecatname": "edukacja"
    },
    {
      "websitecatid": "ah",
      "websitecatname": "erotyka"
    },
    {

```



```
"websitecatid":"ai",
"websitecatname":"film, muzyka, wydarzenia"
},
{
  "websitecatid":"aj",
  "websitecatname":"fotografia"
},
{
  "websitecatid":"bb",
  "websitecatname":"hobby"
},
{
  "websitecatid":"ak",
  "websitecatname":"informacje i publicystyka"
},
{
  "websitecatid":"al",
  "websitecatname":"kobieta"
},
{
  "websitecatid":"am",
  "websitecatname":"kultura i sztuka"
},
{
  "websitecatid":"an",
  "websitecatname":"media"
},
{
  "websitecatid":"ao",
  "websitecatname":"motoryzacja"
},
{
  "websitecatid":"ba",
  "websitecatname":"m\u0119czyzna"
},
{
  "websitecatid":"ap",
  "websitecatname":"nowe technologie"
},
{
  "websitecatid":"aq",
  "websitecatname":"praca"
},
{
  "websitecatid":"ar",
  "websitecatname":"rozrywka"
},
{
  "websitecatid":"at",
  "websitecatname":"sport"
},
},
```

```
{
  "websitecatid":"as",
  "websitecatname":"spo\u0142eczno\u015bci",
},
{
  "websitecatid":"au",
  "websitecatname":"styl \u017cy\u015bcia",
},
{
  "websitecatid":"av",
  "websitecatname":"technologia",
},
{
  "websitecatid":"aw",
  "websitecatname":"telefon\u0105 kom\u00f3rkow\u0105",
},
{
  "websitecatid":"ax",
  "websitecatname":"turystyka",
},
{
  "websitecatid":"ay",
  "websitecatname":"wyszukiwarki i katalogi",
},
{
  "websitecatid":"az",
  "websitecatname":"zdrowie, dom, rodzina",
}
]
}
```

Each record contains fields:

websitecatid - unique category id (char(2))

websitecatname - name or description of
category(string)

11. Command "listLanguages"

Command "listLanguages" returns list of available languages defined for video materials. It has no arguments. See below for sample of request and result.

Argument "json":

```
{
  "action": {
    "actionname": "listlanguages"
  }
}
```

Result:

```
{
  "status": "OK",
  "action": "listlanguages",
  "result": [
    {
      "languageid": 0,
      "language": "polski"
    },
    {
      "languageid": 1,
      "language": "angielski"
    },
    {
      "languageid": 2,
      "language": "niemiecki"
    },
    {
      "languageid": 3,
      "language": "czeski"
    },
    {
      "languageid": 4,
      "language": "w\u0119gierski"
    },
    {
      "languageid": 5,
      "language": "s\u0142owacki"
    },
    {
      "languageid": 6,
      "language": "rumu\u0144ski"
    },
    {
      "languageid": 7,
```

```
    "languageid": 8,
    "languageid": 9,
    "languageid": 10,
  ],
}
```

Each record contains fields:

languageid - unique language id (integer)

languageid - name of language(string)

12. Command "listVideo"

Command "listVideo" returns informations about single video material given by "itemid" field.

Argument "json":

```
{
  "action": {
    "actionname": "listvideo",
    "actionparams": {
      "itemid": "1316",
      "listall": "0|1"
    }
  }
}
```

Command arguments:

itemid: id of video material. **REQUIRED**

listall: if set to 1 info about videos in conversion or blocked could be retrieved. Default value is 0 - only info for available videos is returned. **CAUTION** SID codes returned while listall=1 will be valid only if video conversion will finish successfully.

Result:

```
{
  "status": "OK",
  "action": "listvideo",
  "result": [
    {
      "id": "38623",
      "storage": "1316",
      "title": "Niewiarygodne zjawisko nad Polsk\u0105!  
Trwa\u0142o tylko 3 sekundy",
      "description": "\u017ar\u00f3d\u0142o:  
youtube.com",
      "tags": "terlik",
      "language": "0",
      "length": "32",
      "adddate": "1514357593",
      "adcat": "10",
      "category": "0",

      "sdfile": "https://cdn.onnetwork.tv/.....  
.....",

      "orgfile": "https://cdn.onnetwork.tv/.....
```

```

.....",
    "webcodes":[
        {
            "website":"1324",
            "sid":"TTQsQWt0LDA=" // KOD1
        },
        {
            "website":"2118",
            "sid":"YUksQWt0LDA=" // KOD1
        },
        {
            "website":"14",
            "sid":"RSxBa3QsMA==" // KOD1
        }
    ]
}
]}

```

Result fields for all videos:

```

    id - unique video id (integer)
    storage - id of storage where video is located
(integer)
    title - title of the video
    description - description of the video
    tags - tags of the video
    language - language of the video (integer - see
"listLanguages")
    length - length of the video, seconds (integer)
    adddate - date and time when the video was added (unix
timestamp - integer)
    adcat - advertisement category of the video (integer -
see "listAdCats")
    category - general category of the video (integer - see
"listVideoCategories")
    webcodes - array of websites which can use this video
material containing field "sid" - embed parameter for this
video and website.

```

Additional fields for public videos:

```

    public - this fields contains "1" if this is public
video
    public_since - since this date and time this video can
be used by publisher on its websites. There is no
possibility to play this video on publishers website
(except owner of video) before this date.
    public_fee - rate which publisher has to pay for using
this video on its website (float, changed to publishers

```

currency according to
current bank exchange rate). This is rate for 1000
starts of video with advertisements played before.
Publisher does not pay for
video impressions without advertisement played before.

Fields listed below are available only for users with creators right to listed
video:

sdfile - url to sd version of file, url is valid 3
hours from time of generation
orgfile - url to original file (uploaded to platform),
url is valid 3 hours from time of generation. Original file
is available after migration from conversion servers to
storage servers - it may take up to 24h from upload.

13. Command "listStorageVideos"

Command "listStorageVideos" return list of videos in storage given by argument "storageid".

Argument "json":

```
{
  "action": {
    "actionname": "liststoragevideos",
    "actionparams": {
      "storageid": "1316" ,
      "page": "1",
      "count": "10",
      "expublic": "0",
      "exvcm": "0",
      "exavail": "0",
      "sort": "0",
      "searchstring": "string to search",
      "searchfields": "7"
    }
  }
}
```

Command arguments:

storageid: id of storage (integer - see "listStorages").**REQUIRED**

page: page number, starting with 0, default 0

count: number of videos at page, default 10, max 100

expublic: [0 | 1] exclude public videos from result, default 0

exvcm: [0 | 1] exclude VCM videos from result, default 0

exavail: [0 | 1] exclude not public and not VCM videos from result, default 0

sort: define sorting of results list:

"0" - sort by add date descending (default),

"1" - sort by add date ascending,

"2" - sort by video id descending,

"3" - sort by video id ascending,

searchstring: optional text to search in video records, result contains only videos having specified text in specified fields. If empty or omitted result contain videos according to other conditions passed,

searchfields: select fields to check for **searchstring**, result is **true** if any of fields contains specified **searchstring**:

"0" or "7" - fields title, description and tags, default value,

"1" - field title,

"2" - field description,

"3" - fields title and description,

- "4" - field tags,
- "5" - fields title and tags,
- "6" - fields description and tags,

Result:

```
{
  "status": "OK",
  "action": "listvideo",
  "result": [
    {
      "id": "38623",
      "storage": "1316",
      "title": "Niewiarygodne zjawisko nad Polsk\u0105! Trwa\u0142o tylko 3 sekundy",
      "description": "\u017ar\u00f3d\u0142o: youtube.com",
      "tags": "terlik",
      "language": "0",
      "length": "32",
      "adddate": "1514357593",
      "adcat": "10",
      "category": "0",
      "webcodes": [
        {
          "website": "1324",
          "sid": "TTQsQWt0LDA=" // KOD1
        },
        {
          "website": "2118",
          "sid": "YUksQWt0LDA=" // KOD1
        },
        {
          "website": "14",
          "sid": "RSxBa3QsMA==" // KOD1
        }
      ]
    }
  ]
}
```

Result fields for all videos:

```
id - unique video id (integer)
storage - id of storage where video is located (integer)
title - title of the video
description - description of the video
tags - tags of the video
language - language of the video (integer - see
```

```
"listLanguages")
    length - length of the video, seconds (integer)
    adddate - date and time when the video was added (unix
timestamp - integer)
    adcat - advertisement category of the video (integer -
see "listAdCats")
    category - general category of the video (integer - see
"listVideoCategories")
    webcodes - array of websites which can use this video
material containing field "sid" - embed parameter for this
video and website.
```

Additional fields for public videos:

```
    public - this fields contains "1" if this is public
video
    public_since - since this date and time this video can
be used by publisher on its websites. There is no
possibility to play this video on publishers website
(except owner of video) before this date.
    public_fee - rate which publisher has to pay for using
this video on its website (float, changed to publishers
currency according to
    current bank exchange rate). This is rate for 1000
starts of video with advertisements played before.
Publisher does not pay for
    video impressions without advertisement played before.
```

14. Command "listWebsiteVideos"

Command "listWebsiteVideos" lists videos available to use on website given by "websiteid" argument.

Argument "json":

```
{
  "action": {
    "actionname": "listwebsitevideos",
    "actionparams": {
      "websiteid": "1324" ,
      "page": "1",
      "count": "10",
      "expublic": "0",
      "exvcm": "0",
      "exavail": "0",
      "sort": "0",
      "searchstring": "string to search",
      "searchfields": "7"
    }
  }
}
```

Command arguments:

websiteid: id of website for which list will be prepared (integer - see "listWebsite").**REQUIRED**

page: page number, starting with 0, default 0

count: number of videos at page, default 10, max 100

expublic: [0 | 1] exclude public videos from result, default 0

exvcm: [0 | 1] exclude VCM videos from result, default 0

exavail: [0 | 1] exclude not public and not VCM videos from result, default 0

sort: define sorting of results list:

"0" - sort by add date descending (default),

"1" - sort by add date ascending,

"2" - sort by video id descending,

"3" - sort by video id ascending,

searchstring: optional text to search in video records, result contains only videos having specified text in specified fields. If empty or omitted result contain videos according to other conditions passed,

searchfields: select fields to check for **searchstring**, result is **true** if any of fields contains specified **searchstring**:

"0" or "7" - fields title, description and tags, default value,

"1" - field title,

"2" - field description,

- "3" - fields title and description,
- "4" - field tags,
- "5" - fields title and tags,
- "6" - fields description and tags,

Result:

```
{
  "status": "OK",
  "action": "listwebsitevideos",
  "result": [
    {
      "id": "38623",
      "storage": "1835",
      "title": "Niewiarygodne zjawisko nad Polsk\u0105! Trwa\u0142o tylko 3 sekundy",
      "description": "\u017ar\u00f3d\u0142o: youtube.com",
      "tags": "terlik",
      "language": "0",
      "length": "32",
      "adddate": "1514357593",
      "adcat": "10",
      "category": "0",
      "webcodes": [
        {
          "website": "1324",
          "sid": "TTQsQWt0LDA=" // KOD1
        },
        {
          "website": "2118",
          "sid": "YUksQWt0LDA=" // KOD1
        },
        {
          "website": "14",
          "sid": "RSxBa3QsMA==" // KOD1
        }
      ]
    },
    {
      "id": "31885",
      "storage": "1813",
      "title": "beskid niski",
      "description": "beskid niski",
      "tags": "beskid niski",
      "language": "0",
      "length": "108",
      "adddate": "1507124259",
      "adcat": "0",
      "category": "100",
    }
  ]
}
```

```

"vcm": "1",
"vcm_start": "1507123680",
"vcm_end": "1507728480",
"vcm_income": "5.00",
"webcodes": [
  {
    "website": "1324",
    "sid": "TTQsQWt0LDA=" // KOD1
  },
  {
    "website": "2118",
    "sid": "YUksQWt0LDA=" // KOD1
  },
  {
    "website": "14",
    "sid": "RSxBa3QsMA==" // KOD1
  }
],
},
{
  "id": "38425",
  "storage": "1795",
  "title": "Miejska Wigilia w Pu\u0142awach",
  "description": "",
  "tags": "",
  "language": "0",
  "length": "84",
  "adddate": "1513535943",
  "adcat": "10",
  "category": "0",
  "public": "1",
  "public_since": "1513535943",
  "public_fee": "1.20",
  "webcodes": [
    {
      "website": "1324",
      "sid": "TTQsQWt0LDA=" // KOD1
    },
    {
      "website": "2118",
      "sid": "YUksQWt0LDA=" // KOD1
    },
    {
      "website": "14",
      "sid": "RSxBa3QsMA==" // KOD1
    }
  ]
}
]}

```

Result fields for all videos:

- id - unique video id (integer)
- storage - id of storage where video is located (integer)
- title - title of the video
- description - description of the video
- tags - tags of the video
- language - language of the video (integer - see "listLanguages")
- length - length of the video, seconds (integer)
- adddate - date and time when the video was added (unix timestamp - integer)
- adcat - advertisement category of the video (integer - see "listAdCats")
- category - general category of the video (integer - see "listVideoCategories")
- webcodes - array of websites which can use this video material containing field "sid" - embed parameter for this video and website.

Additional fields for public videos:

- public - this fields contains "1" if this is public video
- public_since - since this date and time this video can be used by publisher on its websites. There is no possibility to play this video on publishers website (except owner of video) before this date.
- public_fee - rate which publisher has to pay for using this video on its website (float, changed to publishers currency according to current bank exchange rate). This is rate for 1000 starts of video with advertisements played before. Publisher does not pay for video impressions without advertisement played before.

15. Command "listAvailableVideos" ("listVideos")

Command "listAvailableVideos" (or "listVideo" for compability with previous API version) returns all videos available for all users websites. Argument json:

```
{
  "action": {
    "actionname": "listvideos",
    "actionparams": {
      "page": "1",
      "count": "10",
      "expublic": "0",
      "exvcm": "0",
      "exavail": "0",
      "sort": "0",
      "searchstring": "string to search",
      "searchfields": "7",
      "listlang": "0,1" // list videos in
polish and english only
    }
  }
}
```

Command arguments:

page: page number, starting with 0, default 0

count: number of videos at page, default 10, max 100

expublic: [0 | 1] exclude public videos from result, default 0

exvcm: [0 | 1] exclude VCM videos from result, default 0

exavail: [0 | 1] exclude not public and not VCM videos from result, default 0

sort: define sorting of results list:

"0" - sort by add date descending (default),

"1" - sort by add date ascending,

"2" - sort by video id descending,

"3" - sort by video id ascending,

searchstring: optional text to search in video records, result contains only videos having specified text in specified fields. If empty or omitted result contain videos according to other conditions passed,

searchfields: select fields to check for **searchstring**, result is **true** if any of fields contains specified **searchstring**:

"0" or "7" - fields title, description and tags, default value,

"1" - field title,

"2" - field description,

"3" - fields title and description,

"4" - field tags,

"5" - fields title and tags,

"6" - fields description and tags,

listlang: allows to limit result to video in selected language(s). Provide here comma separated list of numbers of selected languages. List of available languages with corresponding numbers can be get using **listLanguages** command. If this parameter is empty videos in all languages are listed.

Result:

```
{
  "status":"OK",
  "action":"listwebsitevideos",
  "result":[
    {
      "id":"38623",
      "storage":"1835",
      "title":"Niewiarygodne zjawisko nad Polsk\u0105!
Trwa\u0142o tylko 3 sekundy",
      "description":"\u017ar\u00f3d\u0142o:
youtube.com",
      "tags":"terlik",
      "language":"0",
      "length":"32",
      "adddate":"1514357593",
      "adcat":"10",
      "category":"0",
      "webcodes":[
        {
          "website":"1324",
          "sid":"TTQsQWt0LDA=" // KOD1
        },
        {
          "website":"2118",
          "sid":"YUksQWt0LDA=" // KOD1
        },
        {
          "website":"14",
          "sid":"RSxBa3QsMA==" // KOD1
        }
      ]
    },
    {
      "id":"31885",
      "storage":"1813",
      "title":"beskid niski",
      "description":"beskid niski",
    }
  ]
}
```



```
"tags":"beskid niski",
"language":"0",
"length":"108",
"adddate":"1507124259",
"adcat":"0",
"category":"100",
"vcm":"1",
"vcm_start":"1507123680",
"vcm_end":"1507728480",
"vcm_income":"5.00",
"webcodes":[
  {
    "website":"1324",
    "sid":"TTQsQWt0LDA=" // KOD1
  },
  {
    "website":"2118",
    "sid":"YUksQWt0LDA=" // KOD1
  },
  {
    "website":"14",
    "sid":"RSxBa3QsMA==" // KOD1
  }
],
{
  "id":"38425",
  "storage":"1795",
  "title":"Miejska Wigilia w Pu\u0142awach",
  "description":"",
  "tags":"",
  "language":"0",
  "length":"84",
  "adddate":"1513535943",
  "adcat":"10",
  "category":"0",
  "public":"1",
  "public_since":"1513535943",
  "public_fee":"1.20",
  "webcodes":[
    {
      "website":"1324",
      "sid":"TTQsQWt0LDA=" // KOD1
    },
    {
      "website":"2118",
      "sid":"YUksQWt0LDA=" // KOD1
    },
    {
      "website":"14",
      "sid":"RSxBa3QsMA==" // KOD1
    }
  ]
}
```

```

    }
  ]
}
]
}
```

Result fields for all videos:

```

  id - unique video id (integer)
  storage - id of storage where video is located
(integer)
  title - title of the video
  description - description of the video
  tags - tags of the video
  language - language of the video (integer - see
"listLanguages")
  length - length of the video, seconds (integer)
  adddate - date and time when the video was added (unix
timestamp - integer)
  adcat - advertisement category of the video (integer -
see "listAdCats")
  category - general category of the video (integer - see
"listVideoCategories")
  webcodes - array of websites which can use this video
material containing field "sid" - embed parameter for this
video and website.
```

Additional fields for public videos:

```

  public - this fields contains "1" if this is public
video
  public_since - since this date and time this video can
be used by publisher on its websites. There is no
possibility to play this video on publishers website
(except owner of video) before this date.
  public_fee - rate which publisher has to pay for using
this video on its website (float, changed to publishers
currency according to
  current bank exchange rate). This is rate for 1000
starts of video with advertisements played before.
Publisher does not pay for
  video impressions without advertisement played before.
```

16. Command "listPartnerVideos"

Command "listPartnerVideos" returns all videos available for logged user added by his partner and other users of this partner.

Argument json:

```
{
  "action": {
    "actionname": "listpartnervideos",
    "actionparams": {
      "page": "1",
      "count": "10",
      "sort": "0",
      "searchstring": "string to search",
      "searchfields": "7"
    }
  }
}
```

Command arguments:

page: page number, starting with 0, default 0

count: number of videos at page, default 10, max 100

sort: define sorting of results list:

"0" - sort by add date descending (default),

"1" - sort by add date ascending,

"2" - sort by video id descending,

"3" - sort by video id ascending,

searchstring: optional text to search in video records, result contains only videos having specified text in specified fields. If empty or omitted result contain videos according to other conditions passed,

searchfields: select fields to check for **searchstring**, result is **true** if any of fields contains specified **searchstring**:

"0" or "7" - fields title, description and tags, default value,

"1" - field title,

"2" - field description,

"3" - fields title and description,

"4" - field tags,

"5" - fields title and tags,

"6" - fields description and tags,

Rezultat:

```
{
  "status": "OK",
  "action": "listpartnervideos",
}
```

```

"result":[
  {
    "id":"38623",
    "storage":"1835",
    "title":"Niewiarygodne zjawisko nad Polsk\u0105!
Trwa\u0142o tylko 3 sekundy",
    "description":"\u017ar\u00f3d\u0142o:
youtube.com",
    "tags":"terlik",
    "language":"0",
    "length":"32",
    "adddate":"1514357593",
    "adcat":"10",
    "category":"0",
    "webcodes":[
      {
        "website":"1324",
        "sid":"TTQsQWt0LDA=" // KOD1
      },
      {
        "website":"2118",
        "sid":"YUksQWt0LDA=" // KOD1
      },
      {
        "website":"14",
        "sid":"RSxBa3QsMA==" // KOD1
      }
    ]
  },
  {
    "id":"31885",
    "storage":"1813",
    "title":"beskid niski",
    "description":"beskid niski",
    "tags":"beskid niski",
    "language":"0",
    "length":"108",
    "adddate":"1507124259",
    "adcat":"0",
    "category":"100",
    "vcm":"1",
    "vcm_start":"1507123680",
    "vcm_end":"1507728480",
    "vcm_income":"5.00",
    "webcodes":[
      {
        "website":"1324",
        "sid":"TTQsQWt0LDA=" // KOD1
      },
      {
        "website":"2118",

```

```

        "sid":"YUksQWt0LDA=" // KOD1
    },
    {
        "website":"14",
        "sid":"RSxBa3QsMA==" // KOD1
    }
]
},
{
    "id":"38425",
    "storage":"1795",
    "title":"Miejska Wigilia w Pu\u0142awach",
    "description":"",
    "tags":"",
    "language":"0",
    "length":"84",
    "adddate":"1513535943",
    "adcat":"10",
    "category":"0",
    "public":"1",
    "public_since":"1513535943",
    "public_fee":"1.20",
    "webcodes":[
        {
            "website":"1324",
            "sid":"TTQsQWt0LDA=" // KOD1
        },
        {
            "website":"2118",
            "sid":"YUksQWt0LDA=" // KOD1
        },
        {
            "website":"14",
            "sid":"RSxBa3QsMA==" // KOD1
        }
    ]
}
]}

```

Result fields for all videos:

```

    id - unique video id (integer)
    storage - id of storage where video is located
(integer)
    title - title of the video
    description - description of the video
    tags - tags of the video
    language - language of the video (integer - see
"listLanguages")
    length - length of the video, seconds (integer)

```

adddate - date and time when the video was added (unix timestamp - integer)
 adcat - advertisement category of the video (integer - see "listAdCats")
 category - general category of the video (integer - see "listVideoCategories")
 webcodes - array of websites which can use this video material containing field "sid" - embed parameter for this video and website.

Additional fields for public videos:

public - this fields contains "1" if this is public video
 public_since - since this date and time this video can be used by publisher on its websites. There is no possibility to play this video on publishers website (except owner of video) before this date.
 public_fee - rate which publisher has to pay for using this video on its website (float, changed to publishers currency according to current bank exchange rate). This is rate for 1000 starts of video with advertisements played before. Publisher does not pay for video impressions without advertisement played before.

An example:

```

$_Zmienne["apikey"] = $_ApiKEY;
$_Zmienne["json" ] =
'{
  "action":{
    "actionname":"listpartnervideos",
    "actionparams":{"sort":"2" }
  }
}' ;

$tmpf =
"/tmp/_" . @md5(time() . "_" . @mt_rand(1,1000000) . @mt_rand(1,1000000) . @mt_rand(1,1000000) . @mt_rand(1,1000000)) . ".dat";
$plik = @fopen($tmpf, "w");

$ch = @curl_init();
@curl_setopt($ch, CURLOPT_URL,
"https://api.onnetwork.tv/apiwork.php");
@curl_setopt($ch, CURLOPT_TIMEOUT, 3600);
@curl_setopt($ch, CURLOPT_HEADER, false);
@curl_setopt($ch, CURLOPT_SAFE_UPLOAD, true);
@curl_setopt($ch, CURLOPT_FOLLOWLOCATION, false);

```

```
@curl_setopt($ch, CURLOPT_VERBOSE, false);
@curl_setopt($ch, CURLOPT_FILE, $plik);
@curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
@curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
@curl_setopt($ch, CURLOPT_POST, true);
@curl_setopt($ch, CURLOPT_POSTFIELDS, $_Zmienne);
@curl_exec($ch);
$err = @curl_error($ch);
@curl_close($ch);

@fclose($plik);
$result = @file_get_contents($tmpf);
@unlink($tmpf);
echo( json_encode( json_decode( $sendst , true ) ,
JSON_PRETTY_PRINT )); // print pretty output
```

17. Command "listPlayLists"

Command "listPlayLists" returns playlists created for users websites and videos contained by those playlists.

Argument json:

```
{
  "action": {
    "actionname": "listplaylists",
    "actionparams": {
      "websiteid": "123",
      "playlistid": "123",
      "listauto": "1"
    }
  }
}
```

Command arguments:

websiteid: id of website of which playlists should be listed. If

websiteid==0 playlists for all websites will be listed

playlistid: playlists id which has to be listed. If playlistid==0 all playlist for website given by websiteid (or for all websites if websiteid==0) will be listed. **listauto**: [0 | 1] determines if result contains videos added automatically. For playlist with auto append option off this value is ignored.

Calling listPlayLists with websiteid>0 and playlistid>0 when playlist (given by playlistid) does not belong to specified by websiteid website returns empty result.

Result:

```
{
  "status": "OK",
  "action": "listplaylists",
  "result": [
    {
      "playlistid": "22",
      "website": "458",
      "name": "Praca i pieni\u0105dze",
      "comment": "",
      "tags": "",
      "defaultplaylist": "0",
      "sid": "N2QsMCxN", // KOD2
      "items": [
        {
          "id": "1813",
```



```

        "storage": "302",
        "title": "Jak si\u0119 zmienia polski rynek
pracy? Wyniki Bilansu Kapita\u0142u Ludzkiego",
        "description": "",
        "tags": "",
        "language": "0",
        "length": "227",
        "adddate": "1449758877",
        "adcat": "10",
        "category": "10",
        "public": "1",
        "public_since": "0",
        "public_fee": "1.00",
        "sid": "N2QsVUQsTQ==" // KOD3
    },
    {
        "id": "1806",
        "storage": "302",
        "title": "Imigracja z Ukrainy szans\u0105 dla
polskiej gospodarki",
        "description": "",
        "tags": "",
        "language": "0",
        "length": "216",
        "adddate": "1449755165",
        "adcat": "10",
        "category": "10",
        "public": "1",
        "public_since": "0",
        "public_fee": "1.00",
        "sid": "N2QsVTYsTQ==" // KOD3
    }
  ]
}

```

Result fields:

```

    playlistid - unique playlist id (integer)
    website - unique website id (integer) where playlist
belongs to
    name - playlist name
    comment - playlist comment
    tags - playlist tags
    defaultplaylist - contains "1" if this playlist id
default playlist.
    sid - short code for embed playlist without starting
video
    items - array of videos in playlist in order set by

```

user using onnetwork platform (see "listVideo" for fields descriptions).

CAUTION!!! "sid" fields in array "items" contains short codes for embedding playlist WITH STARTING VIDEO.

18. Command "getMIDCode"

Command "getMIDCode" returns full embed code according to given arguments.

Argument "json":

```
{
  "action": {
    "actionname": "getmidcode",
    "actionparams": {
      "websiteid": "123",
      "playlistid": "123",
      "itemid": "123"
      "midparams": {
        "size": "16x9",
        "autoplay": "0",
        "volume": "20",
        "volumem": "50",
        "showtitle": "3",
        "naa": "0",
        "pinv": "1",
        "pls": "0"
      }
    }
  }
}
```

Command arguments:

websiteid: website id for which code has to be prepared. **WYMAGANY.**

itemid: id of video which has to be played. It can be 0 if correct playlistid is given. In this case player embedded with this code plays whole playlist. If both video id and playlist id are correct player embedded with this code plays selected video and after it videos from playlist. **REQUIRED EITHER**

videoid or playlistid

playlistid: id of playlist to play or 0 if single video (given by videoid) is to play. **REQUIRED EITHER videoid or playlistid**

midparams: array of player options:

size: player aspect ratio, ["16x9" | "4x3"]

autoplay: type of player start action:

"0" - play starts with user action,

"1" - play starts automatically after player is loaded,

"2" - play starts automatically after player is loaded and is visible for user. If player is not visible for user playback is suspended until player becomes visible for user (for example after page scroll),

"3" - play starts automatically after player is loaded and there is

available advertisement to play,

"4" - play starts automatically after player is loaded, there is available advertisement to play and player is visible for user. If player is not visible for user and there is available advertisement playback is suspended until player becomes visible for user (for example after page scroll),

volume: default volume level, range 0-100

volume: volume level set if mouse pointer is over the player, range 0-100

showtitle: what should be printed at top bar of player: **"0"** - nothing (top bar empty),

"1" - title of video,

"2" - description of video

naa: player action if there is no available advertisement (only for "autoplay" set to 3 or 4): **"0"** - none

"1" - player hides and JS window.injectAdvert() is called (if exists) in embedding website sandbox. This function can perform any action.

"2" - player hides and IFRAME with url defined in website "EDIT" option is loaded in the same place on website.

pinv: defines player action if player is invisible (after scroll or browser tab switch):

"0" - player is still playing video

"1" - player pauses video playback. Video playback is automatically resumed when player becomes visible for user.

pls: playlist sorting option:

"0" - video in playlist are played in order set at platform except watched videos are moved to the end of playlist. It means user watch always fresh videos (or videos not recently watched).

"1" - video in playlist are played in order set at platform.

Rezultat:

```
{
  "status": "OK",
  "action": "getmidcode",
  "result": {
    "mid": "Mzk0OTEsNHgzLDEsMjAsMCwxMzI0LDEsMCwzLDUwLDA sMCwwLDA sMCwwLDA sMSww"
  }
}
```

Result field:

`mid` - full code to use as "mid" parameter for embed code.

Remarks on full code construction: If not all player options are given for missing options default values from website or playlist (if given by playlistid) default setting are taken.

19. Command "listVideoStats"

Command "listVideoStats" returns statistics of videos playbacks for videos at users storages.

Argument json:

```
{
  "action": {
    "actionname": "listvideostats",
    "actionparams": {
      "year": "2018",
      "month": "2",
      "videoid": "123",
      "storageid": "123",
      "websites": "0",
      "devices": "0",
      "admission": "0",
      "group": "0"
    }
  }
}
```

Command arguments:

year: year of statistics, current year if omitted.

month: month of statistics, current month if omitted.

videoid: id of video to report statistics. If omitted report will be generated for videos located at storage given by **storageid**

storageid: id of storage for which videos report will be generated. This argument is used if no **videoid** is given. If both **videoid** and **storageid** are omitted report will be generated for all videos user has access to.

websites: **0** - report playback on all websites, **1** - report playbacks on partners own websites, **2** - report playbacks of public videos on websites belonging to other partners.

devices: **0** - report for all devices, **1** - report for desktop devices, **2** - report for mobile devices.

admission: **0** - report statistics of all playbacks, **1** - report playbacks with advertisement played before video, **2** - report playbacks without

advertisement played before video.

group: 0 - group by day and hour, **1** - group by day, **2** - group by month.

Result depends on group type and looks like:

```
{
  "status": "OK",
  "action": "listvideostats",
  "result": {
    {
      "started": "49448420", "first_quartile": "41387011", "midpoint": "36990919", "third_quartile": "33890526", "finished": "27907174"
    }
  }
}
```

or

```
{
  "status": "OK",
  "action": "listvideostats",
  "result": {
    {
      "day": "31", "started": "918545", "first_quartile": "756465", "midpoint": "674917", "third_quartile": "615588", "finished": "477359"
    },
    {
      "day": "30", "started": "1129033", "first_quartile": "932983", "midpoint": "834588", "third_quartile": "763842", "finished": "603729"
    },
    ...
  }
}
```

or

```
{
  "status": "OK",
  "action": "listvideostats",
  "result": {
    {
      "day": "31", "hour": "23", "started": "29124", "first_quartile": "24287", "midpoint": "22037", "third_quartile": "20452", "finished": "15503"
    },
    {
      "day": "31", "hour": "22", "started": "35584", "first_quartile": "29028", "midpoint": "26076", "third_quartile": "23956", "finished": "17735"
    },
    ...
  }
}
```

```
    }
}
```

Report contains data for all selected month.

Results fields:

```
day -
hour -
started - number of videos starts
first_quartile - number of videos played at least 25%
midpoint - number of videos played at least 50%
third_quartile - number of videos played at least 75%
finished - number of videos played till end.
```

20. Command "listAdStats"

Command "listAdStats" returns the list of records with statistics of ad delivery for all websites available for logged user.

Argument json:

```
{
  "action":{
    "actionname":"listadstats",
    "actionparams":
    {"year":2019,"month":4,"day":14}
  }
}
```

Command arguments:

year: year of statistics.

month: month of statistics.

day: day of statistics.

As the result You got the list of the records for specified date grouped by country, partner, website, adprovider and traffic type:

```
{
  "status":"OK",
  "action":"listadstats",
  "result":{

    {"country":"1","partner":"2488","website":"4655","adprovider":"1064","mobile":"1","started":6,"first":5,"mid":5,"third":5,"finished":5,"click":0,"viewa":4}
    .....
  }
}
```

```
}
```

Result record fields:

```
country - partners country as integer,
partner - unique partners id as integer, partners can
be listed using listPartners command ,
website - unique website id as integer, websites can be
listed using listWebsites command ,
adprovider - unique ad provider id, ad providers can be
listed using listAdProviders command ,
mobile - "0" means desktop traffic, value other than 0
means mobile traffic. For now it is always 1 for mobile but
use rather !=0 comparision because we plan to differ
smartphone and tablet traffic in future.
```

```
started - number of ads started,
first - number of ads played with at least 25% of
length,
mid - number of ads played with at least 50% of length,
third - number of ads played with at least 75% of
length,
finished - number of ads completed,
click - number of ads with registered clicks,
viewa - number of ads "viewable" according to IAB video
meetrics (at least 50% of ad pixels continously viewable
for user for at least 2 seconds during playback).
```

21. Command "uploadVideo"

Command "uploadVideo" allows to upload video to platform. Command "uploadVideo" need different API call URL:

```
https://upload.onnetwork.tv/apiup.php
```

Argument json:

```
{
  "action": {
    "actionname": "uploadvideo",
    "actionparams": {
      "storageid": "2733",
      "title": "Nagranie z legendii",
      "description": "Opis do nagrania z
legendii",
      "tags": "Wesołe Miasteczko, Legendia" ,
```



```

    "movielink": "https://www.of.pl/",
    "lang": 0, // if not set - storage
    language

    "public": 0,
    "public_since": 0,
    "public_fee": 0,
    "pubcat": 0,
    "adcat": 10,
    "ageallow": "0",
    "geoallow": "pl,cz,sk"
  }
}
}

```

This command requires to send POST variable of type "file" with name "fileToUpload".

Command arguments:

storageid - storage id to upload video get by **liststorages**, **REQUIRED**

title - video title, at least 10 characters, max 250 characters.

description - video description, up to 250 characters

tags - video tags. tags should be separated using "," (comma). Above sample uses two tags: "Wesołe Miasteczko" and "Legendia".

movielink - url to open while user clicks on playing video.

lang - language of video, list available languages using command **listLanguages**. If omitted default partners language is used.

public - defines if video should be public or not, **0** - video available only for websites partner granted access, **1** - video available for all users of platform.

public_fee - fee for use this video by other users. Given in partners own currency.

public_since - time to set video public. Before this time video will be available only for websites partner granted access. UNIX TIMESTAMP.

pubcat - category id taken from list obtained by command

listvideocategories,

privcat - the PRIVATE category id of the material obtained with the **listprivcats** command,

adcat - advertisement category as listed by command **listadcats**. If omitted set to value "0" - "Video not allowed for ads",

ageallow - Info about age restrictions for video (PG), integer value in range 0-25. If this value is set and website configured with corresponding option PG sign will be shown during video playback,

geoallow - list of countries allowed to watch this video. If omitted video will be available worldwide. Use command **listgeocountries** to get list of

available countries,

Result with success return id of new video:

```
{
  "status":"OK",
  "action":"uploadvideo",
  "result":{
    {"videoid":"123456"}
  }
}
```

Video is available after conversion and it should be checked before use with **listvideo** command.

Below is simple PHP sample:

```
$_ApiKEY = "API key for user";
unset($_Zmienne);
$_Zmienne["apikey"] = $_ApiKEY;
$_Zmienne["json"] = json_encode(
  array( "action" => array(
    "actionname" =>
      "uploadVideo" ,
    "actionparams" =>
      array(
        "storageid"
=> "2733" ,
        "title" => "Tytuł
materiału",
        "description" =>
        "Opis materiału",
        "tags" =>
        "tagi,rozdzielone,przecinkami" ,
        "movielink" =>
        "https://www.of.pl/",
        "lang" => 0,
        "public" => 0,
        "public_since" =>
0,
        "public_fee" => 0,
        "pubcat" => 21,
        "adcat" => 10,
        "ageallow" => 6,
        "geoallow" =>
        "pl,cz,sk"
      )
    )
  )
);
```

```

$_Zmienne["fileToUpload"] = new CurlFile($fileName);

$tmpf = "/tmp/log_file_name";
$plik = @fopen($tmpf, "w");

$ch = @curl_init();
@curl_setopt($ch, CURLOPT_URL,
"https://upload.onnetwork.tv/apiup.php");
@curl_setopt($ch, CURLOPT_TIMEOUT, 3600); // timeout
depends on file size
@curl_setopt($ch, CURLOPT_HEADER, false);
@curl_setopt($ch, CURLOPT_SAFE_UPLOAD, true);
@curl_setopt($ch, CURLOPT_FOLLOWLOCATION, false);
@curl_setopt($ch, CURLOPT_VERBOSE, false);
@curl_setopt($ch, CURLOPT_FILE, $plik);
@curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
@curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
@curl_setopt($ch, CURLOPT_POST, true);
@curl_setopt($ch, CURLOPT_POSTFIELDS, $_Zmienne);
@curl_exec($ch);
$err = @curl_error($ch);
@curl_close($ch);

@fclose($plik);
$result = @file_get_contents($tmpf);
@unlink($tmpf);

```

22. Command "uploadPoster"

This command allows to upload new poster for any material on platform.
 Command "uploadPoster" need different API call URL:

<https://upload.onnetwork.tv/apiup.php>

Argument json:

```

{
  "action": {
    "actionname": "uploadposter",
    "actionparams": {
      "videoid": "274721",
      "posternum": 1,
      "setdefault": 1
    }
  }
}

```

```
}
```

This command requires to send POST variable of type "file" with name "fileToUpload".

Command arguments:

videoid - id of video (or ther type) for which You want to uplaod new poster. **REQUIRED**

posternum - every item can have 6 posters defined, counted from 1 to 6. Here You have to specify which slot You want to occupy.

setdefault - [0|1] sending 1 in this parameter means new poster will be default poster for video.

Result with success:

```
{
  "status":"OK",
  "action":"uploadposter",
  "result":{
    {"videoid":"274721",
      "posternum":1,
      "setdefault":1,
      "size":32126 } // size of uploaded file - can be
    used to control operation success.
  }
}
```

Below is simple PHP sample:

```
$_ApiKEY = "API key for user";
unset($_Zmienne);
$_Zmienne["apikey"] = $_ApiKEY;
$_Zmienne["json"] = json_encode(
  array( "action" => array(
    "actionname" => "uploadposter" ,
    "actionparams" =>
      array(
        "videoid" => "274721" ,
        "posternum" => "1",
        "setdefault" => "1" )
      )
  )
);
$_Zmienne["fileToUpload"] = new CurlFile($fileName);
// $fileName contains file path and name
```

```
$tmpf = "/tmp/nazwa_pliku_z_rezultatem";  
$plik = @fopen($tmpf, "w");  
  
$ch = @curl_init();  
@curl_setopt($ch, CURLOPT_URL,  
"https://upload.onnetwork.tv/apiup.php");  
@curl_setopt($ch, CURLOPT_TIMEOUT, 3600);  
@curl_setopt($ch, CURLOPT_HEADER, false);  
@curl_setopt($ch, CURLOPT_SAFE_UPLOAD, true);  
@curl_setopt($ch, CURLOPT_FOLLOWLOCATION, false);  
@curl_setopt($ch, CURLOPT_VERBOSE, false);  
@curl_setopt($ch, CURLOPT_FILE, $plik);  
@curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);  
@curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);  
@curl_setopt($ch, CURLOPT_POST, true);  
@curl_setopt($ch, CURLOPT_POSTFIELDS, $_Zmienne);  
@curl_exec($ch);  
$err = @curl_error($ch);  
@curl_close($ch);  
  
@fclose($plik);  
$result = @file_get_contents($tmpf);  
@unlink($tmpf);
```


23. Command "listPartners"

Command "listPartners" lists all partners for which logged user has supervisor rights.

Command argument:

```
{
  "action": {
    "actionname": "listpartners"
  }
}
```

Results:

```
{
  "status": "OK",
  "action": "listpartners",
  "result": [
    {
      "id": 1234,
      "name": "partner name"
    },
    ....
  ]
}
```

Every record consist of:

id - partners unique id

name - partners name.

24. Command "listAdProviders"

Command "listAdProviders" returns the list of ad providers for which logged user has administrative rights.

Command argument:

```
{
  "action": {
    "actionname": "listadproviders"
  }
}
```

Results:

```
{
  "status": "OK",
  "action": "listadproviders",
  "result": [
    {
      "id": 1234,
      "name": "provider name"
    },
    ....
  ]
}
```

Every record consists of:

id - ad provider unique id

name - ad provider name.

25. Command "updateVideo"

Command "updateVideo" allows to update selected fields of video record.
Command arguments:

```
{
  "action": {
    "actionname": "updatevideo",
    "actionparams": {
      "itemid" : 1316 , // ID of video
      "title": "Nagranie z legendii",
      "description": "Opis do nagrania z
legendii",
      "tags": "Wesołe Miasteczko,Legendia" ,
      "author": "author of video",
      "lang": 0,
      "public": 0,
      "public_since": 0,
      "public_fee": 0,
      "category": 0,
      "adcat": 10
    }
  }
}
```

Only fields to modify should be present in arguments. If You pass as argument field without value it will be set to empty or default value.

Command arguments:

title - video title, at least 10 characters, max 250 characters.

description - video description, up to 250 characters

tags - video tags. tags should be separated using "," (comma). Above sample uses two tags: "Wesołe Miasteczko" and "Legendia".

movielink - url to open while user clicks on playing video.

lang - language of video, list available languages using command

listLanguages. If omitted default partners language is used.

public - defines if video should be public or not,

0 - video available only for websites partner granted access,

1 - video available for all users of platform.

public_fee - fee for use this video by other users. Given in partners own currency.

public_since - time to set video public. Before this time video will be available only for websites partner granted access. UNIX TIMESTAMP.

pubcat - category id taken from list obtained by command **listvideocategories**,

privcat - the PRIVATE category id of the material obtained with the **listprivcats** command,

adcat - advertisement category as listed by command **listadcats**. If omitted set to value "10" - "Allowed for all ads",
Result is similar to **listVideo** result except it does not contain list of "sid" embed codes.

26. Command "removeVideo"

This command marks material as "to remove".

Command arguments:

```
{
  "action": {
    "actionname": "removevideo",
    "actionparams": {
      "itemid": "1316" // id of material
    }
  }
}
```

The only argument is an ID of material to remove. Material is not removed at once - it is only marked as "to remove". Since marking material as "to remove" it is not available to delivery.

Command arguments:

itemid: ID of material to remove **REQUIRED**

27. Command "addSpeech"

This command allows to add audio material created from supplied text.
Parametr json:

```
{
  "action": {
    "actionname": "addspeech",
    "actionparams": {
      "storageid": "2733",
      "title": "Ogniem i mieczem",
      "description": "Część pierwsza. Czyta
lektor",
      "tags": "Ogniem i mieczem, lektor" ,
      "speech_text" : "Rok 1647 był to dziwny
rok, w którym rozmaite znaki na niebie i ziemi zwiastowały
jakoweś klęski i nadzwyczajne zdarzenia.",
      "lang": 0, // if not set - storage
language
      "voice": "pl-PL-Wavenet-D" ,
      "public": 0,
      "public_since": 0,
      "public_fee": 0,
      "pubcat": 0,
      "adcat": 10,
      "geoallow": "pl,cz,sk"
    }
  }
}
```

Command arguments:

storageid - storage id to add material, get by **liststorages**, **REQUIRED**,

title - video title, at least 10 characters, max 250 characters,

description - video description, up to 250 characters,

tags - video tags. Tags should be separated using "," (comma),

speech_text - text to convert to audio. Length of this text should not be less than 20 characters and more than **20000** characters. Generation time depends on the text length but should not be more than 2 minutes for longest text,

lang - language of the material, list available languages using command **listLanguages**. If skipped default partners language is used.

voice - name of voice get from **listVoices** command result. If skipped default voice for selected language is used,

public - defines if video should be public or not,

0 - video available only for website's partner granted access,

1 - video available for all users of platform.

public_fee - fee for using this video by other users. Given in partners own currency.

public_since - time to set video public. Before this time video will be available only for website's partner granted access. UNIX TIMESTAMP,.

pubcat - category id taken from list obtained by command

listvideocategories,

adcat - advertisement category as listed by command **listadcats**. If skipped set to value "10" - "Allowed for all ads",

geoallow - list of countries allowed to watch this video. If omitted video will be available worldwide. Use command **listgeocountries** to get list of available countries,

Result with success return id of new video:

```
{
  "status": "OK",
  "action": "addspeech",
  "result": [
    {
      "id": "38623",
      "storage": "2733",
      "title": "Ogniem i mieczem",
      "description": "Część pierwsza. Czyta lektor",
      "tags": "Ogniem i mieczem, lektor",
      "language": "0",
      "length": "0",
      "adddate": "1514357593",
      "adcat": "10",
      "category": "0",
      "webcodes": [
        {
          "website": "1324",
          "sid": "TTQsQWt0LDA=" // KOD1
        },
        {
          "website": "2118",
          "sid": "YUksQWt0LDA=" // KOD1
        },
        {
          "website": "14",
          "sid": "RSxBa3QsMA==" // KOD1
        }
      ]
    }
  ]
}
```

Result fields for all videos:

id - unique video id (integer)
 storage - id of storage where video is located (integer)
 title - title of the video
 description - description of the video
 tags - tags of the video
 language - language of the video (integer - see "listLanguages")
 length - length of the video, seconds (integer)
 adddate - date and time when the video was added (unix timestamp - integer)
 adcat - advertisement category of the video (integer - see "listAdCats")
 category - general category of the video (integer - see "listVideoCategories")
 webcodes - array of websites which can use this video material containing field "sid" - embed parameter for this video and website.

Additional fields for public videos:

public - this fields contains "1" if this is public video
 public_since - since this date and time this video can be used by publisher on its websites. There is no possibility to play this video on publishers website (except owner of video) before this date.
 public_fee - rate which publisher has to pay for using this video on its website (float, changed to publishers currency according to current bank exchange rate). This is rate for 1000 starts of video with advertisements played before. Publisher does not pay for video impressions without advertisement played before.

Material is available for playback after about 2 minutes from adding. You should check if it is available using **listvideo** command. Result of **addSpeech** always returns length=0. Value of length is changed during conversion and should be more than 0.

28. Command "listVoices"

This command return list of available voices including info about voice's language and gender.

Command arguments:

```
{
  "action": {
    "actionname": "listvoices"
  }
}
```

Result:

```
{
  "status": "OK",
  "action": "listvoices",
  "result": [
    {
      "name": "ar-XA-Wavenet-A",
      "gender": "FEMALE",
      "lang": "ar-XA"
    },
    ....
  ]
}
```

Every record contains:

name - name of voice,
gender - voice gender,
lang - language.