REST API user guide

Invarivision platform interface allows external developers to receive and record data using the REST API.

Parameters of request

Each request should contain a set of mandatory parameters

Name	Type	Characterization
method	string	the name of the method that will be called, for example,video.list; mandatory parameter
session_key	string	session of the current user

The operation status returns in the answer and if error happens (status = error) it will be possible to get additional information from fields “description”, “err_signature”, “err_type” (see table #1).

Authorization of request

The session (**session_key**) is established by each new user session with your application or site. By next visits of the same user, this value will be different, that is why it is not necessary to save it. Session_key value results during login.

Obtaining of session applications

To obtain session the application should make POST-request on address https://tracker2.invarivision.com/api.php. For example:

```
> POST /oauth/token HTTP/1.1
> Host: tracker2.invarivision.com
> Content-Type: application/x-www-form-urlencoded
>
> grant_type=password&username=test@mail.ru&password=qwerty
```

- grant_type — line “password”
- username — user’s email
- password — user’s password

The answer format

If all parameters are correct, the server returns the session (**access_token**) and additional parameters:


```
{
  "method": "login",
  "status": "finished",
  "description": "Authorization is successful.",
  "access_token": "e5dd32be039e48c08e0331f917421cdb",
  "expires_in": "71603",
  "user_id": "55",
  "refresh_token": "0349e848534270202971834 21e3d6e5d"
}
```


- access_token — session of application
- expires_in — expiration date of access_token in seconds
- refresh_token — **refresh_token** which allows authorization for the second time without login and password of the user
- user_id — id of the user who owns the login and password information; used in the query signature and can be used wherever necessary to specify the user name.

The refresh_token use

Applications are not allowed to save the user's password.In order to ensure the re-obtaining of the session without requiring a password the **refresh_token** mechanism is used. Upon successful authorization with the help of login and password of the user the server returns **refresh_token** value, that can be stored on client and used with the next run of the program instead of the login and password to get the new session.

```
> POST /oauth/token HTTP/1.1
> Host: tracker2.invarivision.com
> Content-Type: application/x-www-formurlencoded
>
> grant_type=refresh_token&
  refresh_token=0349e848534270202971834 21e3d6e5d

< HTTP/1.1 200 OK
< Content-Type: application/json
<
< { "method": "refresh_token",
```

"status": "finished",
"description": "Authorization is successful.",
"access_token": "e5dd32be039e48c08e0331f917421cdb",
"expires_in": "62911",
"user_id": "55",
"refresh_token": "0349e848534270202971834 21e3d6e5d" }

As shown in the example, the query answer with **refresh_token** use returns of the same type as well as during the authorization with help of login and password.

REST API function

video.add

Adds the source video in the system for further search of video content intersection.

Parameters

Name	Type	Characterization
file_url	string	URL of video file

Result

The answer format for json-output:

{
"method": the name of the method is “video.add”
"file": the video file name
"status": operation status (finished, error)
"description": text description of the operation result
"film_id": identification number of source video
}

Example of request

http://tracker2.invarivision.com/api.php?method=video.add&
file_url=http://tracker.invarivision.com/reg/tv/sv1.avi&
session_key=e5dd32be039e48c08e0331f917421cdb

Example of the answer in JSON format

{

```
  "method": "video.add",
  "file": "sv1.avi",
  "status":  "finished",
  "description":   "Your video <./upload/sv1.avi> was successfully added to the searching system.",
    "film_id": 1053
}
```


video.scan

Scans the video to find the image intersection of its content with the previously loaded source video in the system.

Parameters

Name	Type	Characterization
file_url	string	URL of video file

Result

The answer format for json-output:


```
{
  "method": the name of the method is “video.scan”
  "file":  the video file name
  "status": operation status(finished, error)
  "intersections": coefficient of video content intersections with previously loaded video
  "description": text description of the operation result
  "results": array of fragments intersection descriptors
  [
   {
    "film_name": the name of the source video which content is found
    "film_id": identification number of source video
    "film_url": URL of source film
    "film_length": length of the source video(hours;minutes;seconds)
    "channel_name": the name of the scanned video with image intersection
    "channel_url": URL of scanned video
    "channel_position": position of the found fragment in the scanned video(H;M;S)
    "film_position": position of the found fragment in the source video(H;M;S)
    "fragment_length": length of found video fragment intersection(H;M;S)
    "found_date_time": local time and date of fragment detection

   }, ...
```

]
"gathered": array of alternative results with gathered video fragments
[
"gathered_fragments": total number of gathered video fragments
"intersections": coefficient of video content intersections with previously loaded video
"results": array of coincident fragments descriptors
[
{ … }, ...
]
],
"compared_films": list of original videos which have coincidence with the scanned file
[
{
"film_name": the name of the source video
"film_id": identification number of source video
"film_length": length of the source video(hours;minutes;seconds)
"film_url": URL of source film
"added_date_time": local time and date when film was added to the system
"intersections": coefficient of video content intersections with the source video
},
...
]
}

Example of request

http://tracker2.invarivision.com/api.php?method=video.scan&
file_url=http://tracker.invarivision.com/reg/tv/sv_test.avi&
session_key=e5dd32be039e48c08e0331f917421cdb

Example of the answer in JSON format


```
{
"method": "video.scan",
"file": "sv_test.avi",
"status": "finished",
"intersections": "0.654218",
"description": "Intersections with other video content were found: 65.4218 %.",
"results":
[...
{
"film_name": "sv2_avi",
"film_url": "http://tracker2.invarivision.com/tv/sv2_avi",
"film_length": "00:03:04",
```

```
    "channel_name": "sv_test_avi",
    "channel_url": "http://tracker2.invarivision.com/tv/sv_test_avi",
    "channel_position": "0:01:52",
    "film_position": "00:01:26",
    "fragment_length": "00:00:05",
    "found_date_time": "2015-03-31 14:13:09"
   }, ...
  ]
  "gathered":
  [
   "gathered_fragments": 0,
   "intersections": "0.654218",
   "results":
    [...
     {
      "film_name": "sv2_avi",
      "film_url": "http://tracker2.invarivision.com/tv/sv2_avi",
      "film_length": "00:03:04",
      "channel_name": "sv_test_avi",
      "channel_url": "http://tracker2.invarivision.com/tv/sv_test_avi",
      "channel_position": "0:01:52",
      "film_position": "00:01:26",
      "fragment_length": "00:00:05",
      "found_date_time": "2015-03-31 14:13:09"
     }, …
    ]
  ],
  "compared_films":
  [
   {
    "film_name":"sv2_avi",
    "film_id":1061,
    "film_length":"00:03:04",
    "film_url":"http://tracker2.invarivision.com/tv/sv2_avi",
    "added_date_time":"2015-03-20 11:21:55",
    "intersections":0.227
   },
   ...
  ]
}
```


video.delete

Deletes from the system the source video, which can be used to search the video content intersection.

Parameters

Name	Type	Characterization
file_url	string	URL of video file or file name
file_id	uint	File ID (it needs URL or file ID)

Result

The answer format for json-output:

{
"method": the name of the method is “video.delete”
"file": the video file name
"status": operation status (finished, error)
"description": text description of the operation result
}

Example of request #1

http://tracker2.invarivision.com/api.php?method=video.delete&
file_url=http://tracker.invarivision.com/reg/tv/sv1.avi&
session_key=e5dd32be039e48c08e0331f917421cdb

Example of request #2

http://tracker2.invarivision.com/api.php?method=video.delete&
file_id=1072&session_key=e5dd32be039e48c08e0331f917421cdb

Example of the answer in JSON format


```
{
  "method": "video.delete",
  "file": "sv1.avi",
  "status": "finished",
  "description": "The film is deleted."
}
```


video.list

Returns the list of loaded source videos in the system

Result

The answer format for json-output:

{
  "method": the name of the method is “video.list”
  "total": the total number of the video in the list
  "list": array of loaded videos descriptors

  [
   {
    "film_name": the name of the source video
    "film_link": the real name of the source video
    "length": length of source video (hours;minutes;seconds)
    "date_time": the date and time of video file adding
    }, ...
   ]
}

Example of request

http://tracker2.invarivision.com/api.php?method=video.list&
session_key=e5dd32be039e48c08e0331f917421cdb

Example of the answer in JSON format


```
{
  "method": "video.list",
  "user_id": "55"
  "total": 4,
  "list":
  [
   {"film_name": "sv1_avi", "film_link": "sv1.avi", "length": "0:03:06", "date_time": "2015-03-31 13:58:42"},
   {"film_name": "sv2_avi", "film_link": "sv2.avi", "length": "0:03:04", "date_time": "2015-03-31 14:07:40"},
   {"film_name": "sv3_avi", "film_link": "sv3.avi", "length": "0:03:24", "date_time": "2015-03-31 14:10:06"},
   {"film_name": "sv4_avi", "film_link": "sv4.avi", "length": "0:03:01", "date_time": "2015-03-31 14:11:50"}
  ]
}
```

logout

Closure of the session. After this operation is done the **access_token** and **refresh_token** parameters that were received after login become invalid.

Example of request

http://tracker2.invarivision.com/api.php?method=logout&
session_key=e5dd32be039e48c08e0331f917421cdb

Example of the answer in JSON format


```
{
  "method": "logout",
  "status": "finished",
  "description": "The session has been closed."
}
```


Asynchronous API

video.add.start

Starts the background process of adding the source video in the system for the further search of the image intersection in video content.

Parameters

Name	Type	Characterization
file_url	string	URL of video file
streaming	uint	processing in HLS stream mode (disabled on default - “0”)
callback	string	URL of the script that should be called on completion of the process (optional parameter)

Result

The answer format for json-output:


```
{
  "method": the name of the method is “video.add.start”
```

"file": the video file name
"status": operation status (running)
"handle": identifier of the running process
}

Example of request

http://tracker2.invarivision.com/api.php?method=video.add.start&
file_url=http://tracker.invarivision.com/reg/tv/sv1.avi&
session_key=e5dd32be039e48c08e0331f917421cdb

Example of the answer in JSON format


```
{
  "method": "video.add.start",
  "file": "sv1.avi",
  "status": "running",
  "handle": "774cafcaa6cb370f8be28401e6ed1d3e"
}
```


video.scan.start

Starts the background process of the video scan image intersection with previously loaded in the system source video.

Parameters

Name	Type	Characterization
file_url	string	URL of video fie
streaming	uint	processing in HLS stream mode (disabled on default - “0”)
callback	string	URL of the script that should be called on completion of the process(optional parameter)

Result

The answer format for json-output:

{
"method": the name of the method is “video.scan.start”
"file": the video file name
"status": operation status (running)
"handle": identifier of the running process

}

Example of request

http://tracker2.invarivision.com/api.php?method=video.scan.start&
file_url=http://tracker.invarivision.com/reg/tv/sv_test.avi&
session_key=e5dd32be039e48c08e0331f917421cdb

Example of the answer in JSON format


```
{
  "method": "video.scan.start",
  "file": "sv1.avi",
  "status": "running",
  "handle": "0f955a699f725f57458b7bded06d1bcb"
}
```


get.status

Returns the current state or the result of the running background process similar to the one that returns the **video.add** and **video.scan** method.

Parameters

Name	Type	Characterization
handle	string	identifier of the process

Result

The answer format for json-output (current state of the process):

```
{
  "method": the name of the running process method ("video.add" or "video.scan")
  "status": process status (waiting, running, downloading, launching, scanning, adding,
preparing, finished, error)
  "task_progress": processing progress in percentage

 // If streaming = 1, then here are additional fields:
 "intersections": coefficient of video content intersections with previously loaded video
 "fragment": descriptor of a matched fragment
  {
   "film_id": identification number of source video
   "channel_position": position of the found fragment in the scanned video(H;M;S)
   "film_position": position of the found fragment in the source video(H;M;S)
```

"fragment_length": length of found video fragment intersection(H;M;S)
}
}

Example of the request

http://tracker2.invarivision.com/api.php?method=get.status&
handle=24c3adb754f1ed3385c899b0e250188c&
session_key=e5dd32be039e48c08e0331f917421cdb

Example of the answer in JSON format


```
{
  "method": "video.scan",
  "status": "downloading",
  "task_progress": "2"
}
```


get.results

Returns the result of scanning similar to the one that returns the **video.add** and **video.scan** method.

Parameters

Name	Type	Characterization
file_url	string	URL of video file

Result

The answer format for json-output:


```
{
  "method": the name of the method is “get.results”
  "file":  the video file name
  "status": operation status(finished, error)
  "intersections": coefficient of video content intersections with previously loaded video
  "description": text description of the operation result
  "results": array of fragments intersection descriptors
  [
   {
    "film_name": the name of the source video which content is found
    "film_id": identification number of source video
```

"film_url": URL of source film
"film_length": length of the source video(hours;minutes;seconds)
"channel_name": the name of the scanned video with image intersection
"channel_url": URL of scanned video
"channel_position": position of the found fragment in the scanned video(H;M;S)
"film_position": position of the found fragment in the source video(H;M;S)
"fragment_length": length of found video fragment intersection(H;M;S)
"found_date_time": local time and date of fragment detection
}, ...
]
"gathered": array of alternative results with gathered video fragments
[
"gathered_fragments": total number of gathered video fragments
"intersections": coefficient of video content intersections with previously loaded video
"results": array of coincident fragments descriptors
[
{ … }, ...
]
],
"compared_films": list of original videos which have coincidence with the scanned file
[
{
"film_name": the name of the source video
"film_id": identification number of source video
"film_length": length of the source video(hours;minutes;seconds)
"film_url": URL of source film
"added_date_time": local time and date when film was added to the system
"intersections": coefficient of video content intersections with the source video
},
...
]
}

Example of the request

http://tracker2.invarivision.com/api.php?method=get.results&
file_url=http://tracker.invarivision.com/reg/tv/sv_test.avi&
session_key=e5dd32be039e48c08e0331f917421cdb

Example of the answer in JSON format

{
"method": "get.results",
"file": "sv_test.avi",

```
"status": "finished",
"intersections": "0.654218",
"description": "Intersections with other video content were found: 65.4218 %.",
"results":
[...
 {
   "film_name": "sv2_avi",
   "film_id":"1061",
   "film_url": "http://tracker2.invarivision.com/tv/sv2_avi",
   "film_length": "00:03:04",
   "channel_name": "sv_test_avi",
   "channel_url": "http://tracker2.invarivision.com/tv/sv_test_avi",
   "channel_position": "0:01:52",
   "film_position": "00:01:26",
   "fragment_length": "00:00:05",
   "found_date_time": "2015-03-31 14:13:09"
 }, ...
]
"gathered":
[
 "gathered_fragments": 0,
 "intersections": "0.654218",
 "results":
  [...
   {
     "film_name": "sv2_avi",
     "film_id":"1061",
     "film_url": "http://tracker2.invarivision.com/tv/sv2_avi",
     "film_length": "00:03:04",
     "channel_name": "sv_test_avi",
     "channel_url": "http://tracker2.invarivision.com/tv/sv_test_avi",
     "channel_position": "0:01:52",
     "film_position": "00:01:26",
     "fragment_length": "00:00:05",
     "found_date_time": "2015-03-31 14:13:09"
   }, …
  ]
],
"compared_films":
[
 {
   "film_name":"sv2_avi",
   "film_id":1061,
   "film_length":"00:03:04",
   "film_url":"http://tracker2.invarivision.com/tv/sv2_avi",
   "added_date_time":"2015-03-20 11:21:55",
```

```
    "intersections":0.227
   },
   ...
  ]
}
```


close

Closes process descriptor, thus removing from the server all the information about the results of the current process completing.

Parameters

Name	Type	Characterization
handle	string	identifier of the process

Example of the request

http://tracker2.invarivision.com/api.php?method=close&
handle=24c3adb754f1ed3385c899b0e250188c&
session_key=e5dd32be039e48c08e0331f917421cdb

Example of the answer in JSON format


```
{
  "method": "close",
  "status": "finished",
  "description": "The handle is closed."
}
```


callback

On completion, the process will use the script pointed in URL with **handle** and **session_key** parameters if upon starting of the background processes the URL address of callback was specified. Then, the called script can get the results of the process completion with help of **get.status** method. After receiving the results it is necessary to call the **close** method to delete the information about ended process from the server.

Error code table

Table #1

#	err_signature	err_type	description
1	operation_failed	retry_operation	Processing of < *filename* > was failed
2	cannot_open_file	permanent	Can't open file < *filename* >
3	too_small_duration	permanent	Too small video file, length is *XXXX* ms
4	download_error	retry_operation	Download error> fopen(*URL*): failed to open stream
5	login_error	retry_login	Sorry, this user is not found.
6	token_not_exist	retry_login	This refresh token is not exist.
7	session_expired	retry_login	The session time is expired.
8	session_not_valid	retry_login	The session key is not valid.
9	session_absent	permanent	The session key is absent.
10	empty_url	permanent	Sorry but your URL is empty.
11	film_not_found	permanent	The film < *filename* > was not found.
12	handle_not_exist	permanent	This handle is not exist.
13	method_not_defined	permanent	The method < *method* > is not defined.