Hashtopolis User API (V1)

s3in!c ⟨s3inlc@hashes.org⟩

July 16, 2018

Introduction

The communication for the User API is always in JSON formatted values. When sending a request to the server, it should be a POST containing the JSON data. Every request has a *section* and a *request* field to state which action should be executed. Every response again then contains the requested section and request and gives a *status* to indicate if the query was successful or not. To increase the readability of this document, requests are always in blue, successful responses in green and error messages in red.

Errors

In case of an error with the query which the user sends to the server, the response will have following format with the corresponding action which was requested and the error message which should help in getting information about this error.

```
{
  "section":"task",
  "request":"create",
  "status":"ERROR",
  "message":"You are not allowed to create tasks!"
}
```

Sections

This part lists all sections available on the API. The value in the brackets denotes the according value to be sent on API queries.

- Access Control (*access*)
- Agents (*agent*)
- Server Config (*config*)
- Crackers (*cracker*)
- Files (*file*)
- Groups (*group*)
- Hashlists (*hashlist*)
- Preconfigured Tasks (*pretask*)
- Superhashlists (*superhaslist*)
- Supertasks (*supertask*)
- Tasks (*task*)
- Test (*test*)
- User Management (*user*)

Test (*test*)

This section is used to do testing queries, e.g. to test connectivity or availability of this API. The test section is the only one which allows to make requests without an access key.

connection

Used to test if the URL is a valid API endpoint.

```
{
  "section":"test",
  "request":"connection"
}
```

```
{
  "section": "test",
  "request": "connection",
  "response": "SUCCESS"
}
```

access

Used to check if a given API key is still valid and can be used.

```
{
  "section": "test",
  "request": "access",
  "accessKey": "mykey"
}
```

```
{
  "section": "test",
  "request": "access",
  "response": "OK"
}
```

```
{
  "section": "test",
  "request": "access",
  "response": "ERROR",
  "message": "API key was not found!"
}
```

Agents (*agent*)

Used to access all functions around agents. Please note that the user must have access to the groups where an agent is member of to retrieve it and to be able to apply changes.

listAgents

List all agents with some basic informations.

```
{
  "section": "agent",
  "request": "listAgents",
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "listAgents",
  "response": "OK",
  "agents": [
    {
      "agentId": "2",
      "name": "cracker1",
      "devices": [
        "Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz",
        "NVIDIA Quadro 600"
      ]
    }
  ]
}
```


get

Retrieve all the informations about a specific agent by providing its ID. The last action time is a UNIX timestamp and if the configuration on the server is set to hide the IP of the agents, the value will just be *Hidden* instead of the IP.


```
{
  "section": "agent",
  "request": "get",
  "agentId": 2,
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "get",
  "response": "OK",
  "name": "cracker1",
  "devices": [
    "Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz",
    "NVIDIA Quadro 600"
  ],
  "owner": {
    "userId": 1,
    "username": "htp"
  },
  "isCpuOnly": false,
  "isTrusted": true,
  "isActive": true,
  "token": "0lBfAp7YQh",
  "extraParameters": "--force",
  "errorFlag": 2,
  "lastActivity": {
    "action": "getTask",
    "time": 1531316240,
    "ip": "127.0.0.1"
  }
}
```

setActive

Set an agent active/inactive.

```
{
  "section": "agent",
  "request": "setActive",
  "active": false,
  "agentId": 2,
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "setActive",
  "response": "OK"
}
```

changeOwner

Either set an owner for an agent or remove the owner from it. The user can either be specified by providing the user ID or the username. If no owner should be specified, the user value must be *null*.

```
{
  "section": "agent",
  "request": "changeOwner",
  "user": 1,
  "agentId": 2,
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "changeOwner",
  "user": "testuser",
  "agentId": 2,
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "changeOwner",
  "user": null,
  "agentId": 2,
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "changeOwner",
  "response": "OK"
}
```

setName

Set the name of the agent.

```
{
  "section": "agent",
  "request": "setName",
  "name": "cracker1",
  "agentId": 2,
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "setName",
  "response": "OK"
}
```

setCpuOnly

Set if an agent is CPU only or not.

```
{
  "section": "agent",
  "request": "setCpuOnly",
  "cpuOnly": true,
  "agentId": 2,
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "setCpuOnly",
  "response": "OK"
}
```

setExtraParams

Set agent specific command line parameters for the agent which are included in the cracker command line call on the agent.

```
{
  "section": "agent",
  "request": "setExtraParams",
  "extraParameters": "-d 1,2",
  "agentId": 2,
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "setExtraParams",
  "response": "OK"
}
```

setErrorFlag

Set how errors on the agent should be handled on the server. Following values can be given as *ignoreErrors* value:

0 In case of an error, the error message gets saved on the server and the agent will be put into inactive state.

1 In case of an error, the error message gets saved on the server, but the agent will be given further chunks to work on if he requests so.

2 In case of an error, nothing will be saved on the server and the agent can continue to work and will not put into inactive state.

```
{
  "section": "agent",
  "request": "setErrorFlag",
  "ignoreErrors": 0,
  "agentId": 2,
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "setErrorFlag",
  "response": "OK"
}
```

setTrusted

Set if an agent is trusted or not.

```
{
  "section": "agent",
  "request": "setTrusted",
  "trusted": false,
  "agentId": 2,
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "setTrusted",
  "response": "OK"
}
```

listVouchers

Lists all currently existing vouchers on the server which can be used to register new agents.

```
{
  "section": "agent",
  "request": "listVouchers",
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "getBinaries",
  "response": "OK",
  "vouchers": [
    "sM2q6CwiPY",
    "xkw782a3x9",
    "2drg6Vsqor",
    "AZyY8dK1ao"
  ]
}
```

createVoucher

Create a new voucher on the server. It is optional to specify a voucher code otherwise the server will just generate a random one. The server always sends back the created voucher.

```
{
  "section": "agent",
  "request": "createVoucher",
  "voucher": "mySpecificVoucher",
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "createVoucher",
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "createVoucher",
  "response": "OK",
  "voucher": "Gjawgidkr4"
}
```

deleteVoucher

Delete a voucher from the server.

```
{
  "section": "agent",
  "request": "deleteVoucher",
  "voucher": "Gjawgidkr4",
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "deleteVoucher",
  "response": "OK"
}
```

getBinaries

Lists which agent binaries are available on the server to be used for agents.

```
{
  "section": "agent",
  "request": "getBinaries",
  "accessKey": "mykey"
}
```

```
{
  "section": "agent",
  "request": "getBinaries",
  "response": "OK",
   "apiUrl": "http:\/\/localhost\/hashtopolis\/src\/api\/api\/server.php",
  "binaries": [
    {
      "name": "csharp",
      "os": "Windows, Linux(mono), OS X(mono)",
      "url": "http:\/\/localhost\/hashtopolis\/src\/api\/agents.php?download=1",
      "version": "0.52.2",
      "filename": "hashtopolis.exe"
    },
    {
      "name": "python",
      "os": "Windows, Linux, OS X",
      "url": "http:\/\/localhost\/hashtopolis\/src\/api\/agents.php?download=2",
      "version": "0.1.4",
      "filename": "hashtopolis.zip"
    }
  ]
}
```

Tasks (*task*)

Used to access all functions around tasks. Please note that the user must have access to the groups where a task is belonging to, to retrieve it and to be able to apply changes.

listTasks

List all tasks on the server. There are two task types:

0 Normal Task

1 Supertask

```
{
  "section": "task",
  "request": "listTasks",
  "accessKey": "mykey"
}
```

```
{
  "section": "task",
  "request": "listTasks",
  "response": "OK",
```

```
  "tasks": [
    {
      "taskId": 7587,
      "name": "test 2",
      "type": 0,
      "hashlistId": 1,
      "priority": 5
    },
    {
      "supertaskId": 33,
      "name": "Increment ?a",
      "type": 1,
      "hashlistId": 1,
      "priority": 3
    },
    {
      "supertaskId": 32,
      "name": "Supertask Test",
      "type": 1,
      "hashlistId": 1,
      "priority": 0
    },
    {
      "taskId": 7580,
      "name": "test 1",
      "type": 0,
      "hashlistId": 1,
      "priority": 0
    }
  ]
}
```


getTask

Get the details for a specific task. Note that this request can only be done with tasks or subtasks, but not with supertasks.


```
{
  "section": "task",
  "request": "getTask",
  "taskId": 7587,
  "accessKey": "mykey"
}
```

```
{
  "section": "task",
  "request": "getTask",
  "response": "OK",
  "taskId": 7587,
  "name": "testing",
  "attack": "#HL# -a 0 top10000.txt -r dive.rule",
  "cunksize": 600,
  "color": null,
  "benchmarkType": "speed",
  "statusTimer": 5,
  "priority": 0,
  "isCpuOnly": false,
```

```
  "isSmall": false,
  "skipKeyspace": 0,
  "keyspace": 10000,
  "dispatched": 10000,
  "hashlistId": 1,
  "imageUrl": "http:\/\/localhost\/hashtopolis\/src\/api\/taskimg.php?task=7587",
  "files": [
    {
      "fileId": 2,
      "filename": "dive.rule",
      "size": 887155
    },
    {
      "fileId": 3653,
      "filename": "top10000.txt",
      "size": 76508
    }
  ],
  "speed": 0,
  "searched": 10000,
  "chunkIds": [
    31
  ],
  "agents": [
    {
      "agentId": 2,
      "benchmark": "0",
      "speed": 0
    }
  ]
}
```