

**Hyperiums API (HAPI) Reference Manual**

for HAPI v0.1.8

by Mike Angstadt - github.com/mangstadt - “mangst” in-game

**Changelog**

**v1.0 - 2011-06-19:**

First version.

**v1.1 - 2011-08-01:**

* Fixed description of descrN response parameter from Games method
* lastincome response parameter from PlayerInfo method does not include lord withholdings
* “nrj” stands for “energy” -.-

**v1.2 - 2011-11-30:**

* Added a section describing the failsafe parameter.
* Removed the game state values that were listed in the Games method section and referred the reader to the *Constants* section.
* Minor changes to Download method section.
* Shortened the table of contents.

**Abstract**

The aim of this document is to provide a detailed description of HAPI and its usage.

Table of Contents

Overview 4

HTTP Requests 4

HTTP Responses 4

Content Type 4

Error Responses 4

Lists 5

Boolean values 5

Timestamps 5

“dummy” parameter 5

“failsafe” parameter 5

Authentication 6

Requests 7

Download 8

Alliance data 8

Event data 9

Planet data 9

Player data 9

Games 10

GetAlliancePlanets 11

GetExploitations 12

GetFleetsInfo 12

GetMovingFleets 14

GetNewMsg 15

GetOldPersoMsg 16

GetOldPlanetMsg 16

GetPlanetInfo 17

“General” request 18

“Infiltr” request 19

“Trading” request 20

GetPlayerInfo 21

IsMsg 21

IsMsgInfo 22

Logout 22

Version 23

Constants 24

# Overview

This goal of this document is to provide a detailed description of the Hyperiums API (HAPI). It is based on the official HAPI specs (<http://www.hyperiums.com/HAPI_specs.html>).



**Note:** Throughout this document, the placeholder *<HAPI>* will be used as shorthand for the HAPI URL, which is:

http://www.hyperiums.com/servlet/HAPI

## HTTP Requests

All requests are HTTP GET requests that contain query string parameters.

## HTTP Responses

Each response contains name/value pairs in the form of a query string. For example:

dummy=&planet0=Fayette&planetid0=91338&nbexp0=50&inpipe0=0&tobedem0=0&nbonsale0=0&sellprice0=0&rentability0=0

Note that HAPI does not encode most special characters, such as spaces, equal signs, and question marks, so be careful if you are using a generic URL parsing library.

Escaped ampersand characters appear as [:&:].

### Content Type

Responses to the Download request have the content type application/download, unless an error response is returned, in which case the content type is text/plain.

Responses to all other requests have the content type text/plain.

### Error Responses

If an error occurs with a request, such as an authentication failure, a response containing an error parameter will be returned. For example:

error=authentication failed

This includes the Download request, which normally returns a gzipped file.

### Lists

Many responses return lists of data. Because each parameter name in a query string has to be unique, the name of each parameter is appended with an index number.

Some responses also have sub-lists. The parameter names for sub-lists are appended with the parent list index, a period, and then the child list index. For example, the GetFleetsInfo method returns a list of planets, each of which contains a list of fleets:

planet0=Fayette&...fname0.0=AlphaDogs&owner0.0=mangst&...fname0.1=TheWolves&owner0.1=?...planet1=Spruce...

### Boolean values

Some response parameters contain boolean values. A value of “0” means “false” and a value of “1” means true.

### Timestamps

All timestamps from the server are in GMT and are in the following 24-hour format:

yyyy-mm-dd hh:mm:ss

### “dummy” parameter

An empty parameter named dummy is returned in some responses. This is probably included to show that the request was processed successfully incase the response has no data to return (a completely empty response could mean that there was a server error).

## “failsafe” parameter

A query string parameter named failsafe can optionally be included in the request. The value of this parameter can be anything, but it should be unique so that no two requests use the same value. If included in the request, a failsafe parameter with the same value will be sent back in the response (it is “echoed” back).

This can be used to ensure that the response is coming directly from the HAPI server, and not from a cache. The value included in the response should match the original value sent in the request. If they are different (or if the response does not contain a failsafe parameter at all), then the response should be discarded, as it may not contain up-to-date data.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=games&failsafe=1322699724 |
| *Response Body* | dummy=&game0=Hyperiums6&state0=1&descr0=Permanent game (Round #6)&length0=replaced\_by\_maxenddate&maxenddate0=null&ispeec0=0&maxplanets0=14&initcash0=500000&maxofferedplanets0=15&nextplanetdelay0=5&game1=HyperiumsRLF&state1=1&descr1=Permanent RLF game (Real Life Friendly)&length1=replaced\_by\_maxenddate&maxenddate1=null&ispeec1=0&maxplanets1=14&initcash1=300000&maxofferedplanets1=10&nextplanetdelay1=5&game2=RLF2&state2=1&descr2=Permanent Real Life Friendly v2&length2=replaced\_by\_maxenddate&maxenddate2=null&ispeec2=0&maxplanets2=10&initcash2=300000&maxofferedplanets2=10&nextplanetdelay2=3&failsafe=1322699724 |

# Authentication

In order to access player-centric information in HAPI, you must first authenticate with the player’s account. This involves the player logging into Hyperiums him or herself and generating a "HAPI key" also known as an "external authentication key". A HAPI key is generated by going to "Preferences > Authentication", as shown in the screenshot below.

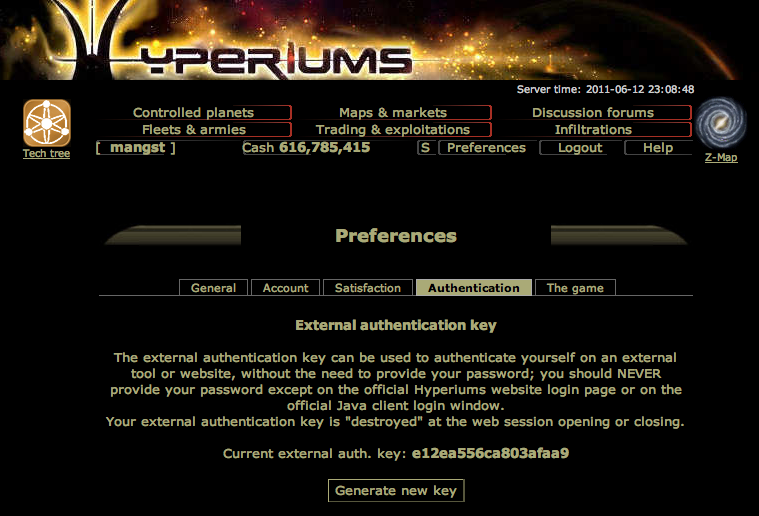


Figure : Generating a HAPI key

After the HAPI key is generated, authentication with HAPI can proceed. An authentication request and response can be described as follows:

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?game=Hyperiums6&player=mangst& hapikey=c788c4fdecd3e8aa1 |
| *Response Body* | gameid=0&playerid=143057&playername=mangst&  authkey=214595267&servertime=2011-06-05 00:44:38 |

**Request parameters:**

game The name of the game to connect to (the Games request can be used to get a list of all games--it does not require authentication).

player The player's username.

hapikey The HAPI key generated from the web site.

**Response parameters:**

gameid The ID of the game (must be included in all subsequent requests).

playerid The ID of the player (must be included in all subsequent requests).

authkey The authentication key (must be included in all subsequent requests). This is not the same as the HAPI key.

playername The player's username.

servertime The time that the authentication took place.

The gameid, playerid and authkey response parameters must be included in all subsequent HAPI requests.

All requests except for the Games and Download requests require this kind of authentication (Download requires the username/password to be sent, Games requires no authentication).

When the player logs out of his or her web session from the hyperiums.com website, the HAPI session is also terminated. A new HAPI key will have to be generated in order to connect to HAPI again.

# Requests

This section describes each request that can be made with HAPI.



**Note:** The placeholder *<AUTH>* will be used as shorthand for the three parameters that must be sent with every request that requires authentication (see the Authentication section). An example of the auth parameters is below:

gameid=0&playerid=143057&authkey=214595267

## Download

This will download one of the daily-generated data files. Each file can only be downloaded once per day. The files are gzipped.

This request does not require the type of authentication as described in the Authentication section, but does require the player's username and password.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=download&game=Hyperiums6&player=mangst&passwd=secret&filetype=planets |
| *Response Body* | <gzipped text> |

**Request parameters:**

request=download

game The name of the game to download the data from (a list of game names can be retrieved by calling the Games method).

player The player's name.

passwd The player's password.

filetype The data file to download. Can be one of the following values:  
\* alliances - A list of all alliances.  
\* events - A list of all events.  
\* planets - A list of all planets.  
\* players - A list of all players.

**Response**

The response is gzipped text, unless the file has already been download that day, in which case an error response will be returned. The content type is "application/download".

The first few lines of each file contain the date the file was generated and a description of the file format. They start with a # (hash) character. The rest of the file is formatted as follows:

### Alliance data

Every group of three lines represents an alliance.

Line 1: <tag> <name>

Line 2: <description>

Line 3: <president name> <average x-coordinate> <average y-coordinate> <number of planets> <total influence>

**Example alliance:**

|  |
| --- |
| SKIP Petit & Puissant  le nettoyage de la galaxie !  bosco 1 -390 17 904218056417 |

The data fields are delimited by one space character, except for the name and description fields, which may contain spaces. Line 2 will be **blank** if the alliance does not have a description.

### Event data

Each line contains a description of the event.

<planet ID> <planet name> <event date> <event description>

**Example event:**

|  |
| --- |
| 9521 Morska 2011-06-11 07:06:39 Planet annihilated by Black Hole. |

The data fields are delimited by one space character, except for the event description field, which may contains spaces, and the event date field, which will contain a space between the date and time.

### Planet data

Each line contains data on a planet.

<ID> <name> <government system> <x-coordinate> <y-coordinate> <race> <production type> <activity> <public tag> <civilization level> <planet size>

**Example planet:**

|  |
| --- |
| 91338 Fayette 3 22 -855 0 2 18868 [Pro-T] 13 4 |

The data fields are delimited by one space character.

### Player data

Each line contains data on a player.

<name> <influence rank> <influence rank (super-cluster) <influence score> <hyp rank> <inflicted damage rank> <inflicted damage score> <inflicted damage rank (super cluster)> <protected super cluster> <city/country>

**Example player:**

|  |
| --- |
| MacSnack 274 275 4285691281 6 3 5440554797 3 0 Netherlands |

The data fields are delimited by one space character, except for the city/country field, which may contain spaces. Inflicted damage rankings will be "0" if the player has made them private. The protected super cluster field will be "0" if the player is not in a protected super cluster.

## Games

Gets a list of all Hyperiums games. This request does not require authentication.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=games |
| *Response Body* | dummy=&game0=Hyperiums6&state0=1&descr0=Permanent game (Round #6)&length0=replaced\_by\_maxenddate&maxenddate0=null&ispeec0=0&maxplanets0=14&initcash0=500000&maxofferedplanets0=15&nextplanetdelay0=5&game1=HyperiumsRLF&state1=1&descr1=Permanent RLF game (Real Life Friendly)&length1=replaced\_by\_maxenddate&maxenddate1=null&ispeec1=0&maxplanets1=14&initcash1=300000&maxofferedplanets1=10&nextplanetdelay1=5&game2=RLF2&state2=1&descr2=Permanent Real Life Friendly v2&length2=replaced\_by\_maxenddate&maxenddate2=null&ispeec2=0&maxplanets2=10&initcash2=300000&maxofferedplanets2=10&nextplanetdelay2=3 |

**Request parameters:**

request=games

**Response parameters:**

The response returns a list of games.

gameN The game name.

stateN The game state (see Game States in Constants section for values).

descrN The game description.

lengthN No longer contains a usable value. Use maxenddateN instead.

maxenddateN The date that the game will end or “null” if the game has no end date.

ispeecN <unknown>

maxplanetsN The maximum number of planets each player can have.

initcashN The amount of cash each new player starts with.

maxofferedplanetsN The max number of planets each player can have including the free ones that you get.

nextplanetdelayN The number of days each player must wait before they get their free planet.

## GetAlliancePlanets

Gets a list of all planets in an alliance. The player must belong to the alliance and have access to its planet list in order for this request to be successful. Otherwise, an error response is returned.

A max of 50 planets is returned at a time.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?tag=p0wn&start=0&*<AUTH>* |
| *Response Body* | nb=2&planet0=FooBar&owner0=SuperUser&x0=-32&y0=-492&prodtype0=0&race0=2&activity0=25125&publictag0=p0wn&ptagid0=8672&planet1=HelloWorld&owner1=Superman&x1=53&y1=593&prodtype1=1&race1=0&activity1=23071 |

**Request parameters:**

request=getallianceplanets

tag The alliance's tag (without brackets).

start The place in the planet list to start returning planets (this request only returns a max of 50 planets).

**Response parameters:**

The response contains a list of planets (up to a max of 50 planets).

nb The number of planets in the response.

planetN The planet’s name.

ownerN The planet’s owner.

xN The planet's x-coordinate.

yN The planet's y-coordinate.

prodtypeN The planet's production type (see Production Types in Constants section for values).

raceN The planet's race (see Races in Constants section for values).

activityN <unknown>

**The following parameters are only included in the response if the planet has a public tag:**

publictagN The planet's public tag.

ptagidN The planet's public tag alliance ID.

## GetExploitations

Gets info on all exploitations the player owns, on both controlled and foreign planets.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=getexploitations&*<AUTH>* |
| *Response Body* | dummy=&planet0=Fayette&planetid0=91338&nbexp0=250&inpipe0=0&tobedem0=0&nbonsale0=0&sellprice0=0&rentability0=0&planet1=Spruce&planetid1=96999&nbexp1=102&inpipe1=98&tobedem1=0&nbonsale1=0&sellprice1=0&rentability1=0 |

**Request parameters:**

request=getexploitations

**Response parameters:**

The response returns a list of planets that the player owns exploitations on.

planetN The planet name.

planetidN The planet ID.

nbexpN The number of exploits the player owns on that planet.

inpipeN The number of exploits that are queued to build.

tobedemN The number of exploits that are going to be destroyed.

nbonsaleN The number of exploits on sale.

sellpriceN The sell price for the exploits (“0” if they are not for sale).

rentabilityN <unknown>

## GetFleetsInfo

Gets info on all of the fleets that are on a particular planet or many planets. Does not include fleets that are in transit (see GetMovingFleets). If the planet is foreign, the player must have fleets on the planet in order to see all of the planet’s fleets.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=getfleetsinfo&planet=Fayette&data=own\_planets&*<AUTH>* |
| *Response Body* | dummy=&planet0=Fayette&stasis0=0&vacation0=0&nrj0=20&nrjmax0=20&fleetid0.0=6135550&fname0.0=&sellprice0.0=0&frace0.0=0&owner0.0=mangst&defend0.0=1&camouf0.0=0&bombing0.0=0&autodrop0.0=0&delay0.0=0&scou0.0=8007&crui0.0=298&bomb0.0=0&dest0.0=896&carmies0.0=5&fleetid0.1=6135553&fname0.1=&sellprice0.1=0&frace0.1=0&owner0.1=mangst&defend0.1=1&camouf0.1=0&bombing0.1=0&autodrop0.1=0&delay0.1=0&scou0.1=2000&crui0.1=100&bomb0.1=0&dest0.1=300&carmies0.1=0&fleetid0.2=6135554&fname0.2=&sellprice0.2=0&frace0.2=0&owner0.2=mangst&defend0.2=1&camouf0.2=0&bombing0.2=0&autodrop0.2=0&delay0.2=0&garmies0.2=10&fleetid0.3=6135548&fname0.3=&sellprice0.3=0&frace0.3=0&owner0.3=mangst&defend0.3=1&camouf0.3=0&bombing0.3=0&autodrop0.3=0&delay0.3=0&garmies0.3=46 |

**Request parameters:**

request=getfleetsinfo

planet Returns the fleets that are on the specified planet. Set its value to \* (asterisk) to return all of the fleets on either all the player's controlled planets or all foreign planets (depending on the value of the data parameter).

data Can be one of two values:  
\* own\_planets - Gets fleets info from the planets that the player controls.  
\* foreign\_planets - Gets fleets info from planets that the player does not control.

**Response parameters:**

Returns a list of planets, each of which contains a list of fleets stationed on the planet:

planetN The name of the planet.

stasisN Whether or not the planet is in stasis (boolean).

vacationN Whether or not the planet owner is in vacation mode (boolean).

fleetidN.M The fleet's ID.

fnameN.M The fleet's name. Will be "null" if the fleet never had a name, empty if the fleet once had a name, but the name was removed.

sellpriceN.M The sell price of the fleet ("0" if not for sale).

fraceN.M The fleet's race (see Races in Constants section for values).

ownerN.M The player name of the fleet's owner or "?" if the owner is unknown.

defendN.M "1" if the fleet is defending the planet, "0" if it is attacking it.

camoufN.M Whether or not the fleet is camouflaged (boolean).

bombingN.M Whether or not the fleet is bombing the planet (boolean).

autodropN.M Whether or not the fleet is set to automatically drop its armies to the planet's surface as soon as it can (fleets cannot drop armies for 2 hours after arriving at a planet).

delayN.M The number of hours before the fleet can leave the planet (fleets must wait 2 hours after arrival before they can leave).

**The following parameters only appear in the response if the fleet is made up of ships:**

scouN.M The number of scouts in the fleet.

bombN.M The number of bombers in the fleet.

destN.M The number of destroyers in the fleet.

cruiN.M The number of cruisers in the fleet.

carmiesN.M The number of carried armies in the fleet.

**The following parameters only appears in the response if the fleet is made up of ground armies:**

garmiesN.M The number of ground armies in the fleet. Will be "?" if this number is unknown (this will happen if the player has ships on a foreign planet, but has no armies on the ground).

**The following parameters only appear in the response if the player controls the planet:**

nrjN The current amount of energy the planet has.

nrjmaxN The max amount of energy the planet can have.

## GetMovingFleets

Retrieves information on all of the player's fleets that are in transit.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=getmovingfleets&*<AUTH>* |
| *Response Body* | dummy=&fleetid0=6037820&fname0=&from0=Fayette&to0=Aussie\_Blues&dist0=3&delay0=0&defend0=1&autodrop0=0&camouf0=0&bombing0=0&race0=0&nbbomb0=0&nbdest0=0&nbcrui0=0&nbscou0=1&nbarm0=0&fleetid1=6037749&fname1=myname&from1=Fayette&to1=BlackPier&dist1=2&delay1=0&defend1=1&autodrop1=0&camouf1=0&bombing1=0&race1=0&nbbomb1=0&nbdest1=0&nbcrui1=0&nbscou1=1&nbarm1=0 |

**Request parameters:**

request=getmovingfleets

Response parameters

fleetidN The fleet's ID.

fnameN The fleet's name. Will be empty if the fleet does not have a name.

fromN The planet that the fleet left from.

toN The planet that the fleet is traveling to.

distN The number of hours before the fleet reaches its destination.

delayN <unknown>

defendN "1" if the fleet is ordered to defend the planet on arrival, "0" if it is ordered to attack it.

autodropN Whether or not the carried armies will automatically be dropped to the planet's surface upon arrival (boolean).

camoufN Whether or not the fleet is camouflaged (boolean)

bombingN Whether or not the fleet will bomb the planet upon arrival (boolean).

raceN The fleet's race (see Races in Constants section for values).

nbscouN The number of scouts in the fleet.

nbbombN The number of bombers in the fleet.

nbdestN The number of destroyers in the fleet.

nbcruiN The number of cruisers in the fleet.

nbarmN The number of carried armies in the fleet.

## GetNewMsg

Gets all new (unread) messages. Calling this method will mark the messages as "read", so it will not return the same messages if called twice in a row.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=getnewmsg&*<AUTH>* |
| *Response Body* | nbmsg=4&date0=2011-06-08 23:15:48&type0=0&msg0=Hi, nice talking with you.&subj0=Hello&sender0=Lt.Cdr. MrPlayer&planet0=Fayette&planetstart0=1&date1=2011-06-08 23:15:58&type1=16&msg2=Please trade with me.&subj1=Trading&sender1=Cptn. George |

**Request parameters:**

request=getnewmsg

**Response parameters:**

There are two kinds of messages: player messages and planet messages. The response groups these two types in a particular way. The player messages (messages that are directly addressed to the player) come first, followed by the planet messages (messages that are addressed to one of the player's planets). The planet messages are grouped by planet. The planetM and planetstartM parameters are used to determine what planet a particular message is for (the indexes of these parameters do not correspond to the indexes of the messages--they exist in their own “list”).

nbmsg The number of new messages.

dateN The date of the message.

typeN The message type (see Message Types in Constants section for values).

msgN The message body.

subjN The message subject.

senderN The sender of the message. Will be "null" if it is a system message, like a cash tick message.

planetM The name of the planet that a group of messages is for (goes along with planetstartM).

planetstartM The messages whose indexes are >= this parameter's value are for the planet specified in the planetM parameter, up until the message whose index is specified in planetstart(M+1).

## GetOldPersoMsg

Retrieves old (read) player messages.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=getoldpersomsg&startmsg=5&maxmsg=3&*<AUTH>* |
| *Response Body* | nbmsg=3&date0=2011-06-17 14:56:04&msg0=Cash tick: your investments brought in 2,227,938.&subj0=&sender0=null&date1=2011-06-17 06:56:44&msg1=Cash tick: your investments brought in 2,015,640.&subj1=&sender1=null&date2=2011-06-16 22:56:08&msg2=Cash tick: your investments brought in 1,768,761.&subj2=&sender2=null |

**Request parameters:**

request=getoldpersomsg

startmsg The message to start on ("0" is the most recent message, "1" is second most recent message, etc). Messages are ordered by date descending (newest first).

maxmsg The number of messages to return.

**Response parameters:**

The response contains a list of messages:

nbmsg The number of messages in the response.

dateN The date of the message.

msgN The message body.

subjN The message subject.

senderN The sender of the message. Will be "null" if it is a system message, like a cash tick message.

## GetOldPlanetMsg

Retrieves old (read) planet messages.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=getoldplanetmsg&startmsg=0&maxmsg=3&planet=%2A&*<AUTH>* |
| *Response Body* | nbmsg=3&date0=2011-06-18 00:50:04&type0=1&msg0=Our Human fleet has reached its destination planet <a href=Planetfloats?planetid=91338>Fayette</a>, coming from <a href=Planetfloats?planetid=90824>Rutabaga</a>.&subj0=&sender0=null&date1=2011-06-18 00:50:04&type1=1&msg1=Our Human fleet has reached its destination planet <a href=Planetfloats?planetid=91338>Maple</a>, coming from <a href=Planetfloats?planetid=90826>Kumquat</a>.&subj1=&sender1=null&date2=2011-06-18 00:50:04&type2=1&msg2=Our Human fleet has reached its destination planet <a href=Planetfloats?planetid=91338>Broad</a>, coming from <a href=Planetfloats?planetid=91287>Fayette</a>.&subj2=&sender2=null |

**Request parameters:**

request=getoldplanetmsg

startmsg The message to start on ("0" is the most recent message, "1" is second most recent message, etc). Messages are ordered by date descending (newest first).

maxmsg The number of messages to return.

planet The planet who messages are to be retrieved. Set its value to \* (asterisk) to retrieve messages from all planets.

**Response parameters:**

The response contains a list of messages:

nbmsg The number of messages in the response.

dateN The date of the message.

msgN The message body.

subjN The message subject.

senderN The sender of the message. Will be "null" if it is a system message, like a fleet arrival message.

## GetPlanetInfo

Gets information on all your controlled planets, or any planets that you have fleets/armies stationed on.

**Request parameters:**

request=getplanetinfo

data The type of data to return. Can be one of the following values:  
\* general - Retrieves basic information like the planet’s x/y coordinates.  
\* infiltr - Retrieves info on the planet’s infiltrations (does not work with foreign planets).  
\* trading - Retrieves info on the planet’s trading relations (does not work with foreign planets).

planet The name of the planet to retrieve the info on. This can be either a planet that the player controls or a foreign planet that the player has fleets/armies stationed on. The value \* (asterisk) can also be used to return info on all of the player’s controlled planets.

A different response is returned depending on the value of the data parameter and whether or not the specified planet is foreign (described below).

### “General” request

Gets general information on the planet(s). The player must have fleets/armies stationed on the planet if the requested planet is foreign. The response is different depending on whether the planet is foreign or not.

**Response parameters (player controlled planets):**

If the requested planet is under the player’s control or the request asked for info on all of the player’s planets, a list will be returned with the following parameters:

planetN The planet name.

xN The planet’s x-coordinate.

yN The planet’s y-coordinate.

sizeN <unknown>

orbitN The planet’s orbital position in its local star system.

govN The planet’s government type (see Government Types in Constants section for values).

govdN The number of days before the planet’s government type can be changed.

ptypeN The planet’s production type (see Production Types in Constants section for values).

taxN The planet’s tax rate (example: “30” is 30%).

exploitsN The number of exploitations the planet has.

expinpipeN The number of exploitations that are being built.

activityN <unknown>

popN The planet’s population in millions.

raceN The planet’s race (see Races in Constants section for values).

nrjN The current amount of energy the planet has.

nrjmaxN The max amount of energy the planet can have.

purifN Whether or not the planet is being purified (boolean).

paranoN Whether or not the planet is in paranoid mode (boolean).

blockN Whether or not the planet is being blockaded (boolean).

bholeN Whether or not the planet is being destroyed by a black hole (boolean).

stasisN Whether or not the planet is in stasis (boolean).

nexusN Whether or not the planet has a nexus that is either completed or under construction (boolean).

ecomarkN The planet’s eco mark (example: “100” is 100%).

planetidN The planet’s ID.

publictag The planet’s public tag. Will be empty if the planet does not have one.

factoriesN The number of factories the planet has.

civlevelN The planet’s civ level.

defbonusN The planet’s defense bonus (example: “10” for 10%).

**The following parameters will only be included in the response if** nexusN **is “1”:**

nxbuildN The number of hours left before the planet’s nexus is finished or “0” if the nexus is complete.

nxbtot The total number of hours the nexus takes to build or “0” if the nexus is complete.

**Response parameters (foreign planet):**

If the requested planet is foreign, then the following parameters are returned:

planet The planet’s name.

stasis Whether or not the planet is in stasis (boolean).

battle Whether or not there is a battle on the planet (boolean).

blockade Whether or not the planet is being blockaded (boolean).

vacation Whether or not the planet’s owner is in vacation mode (boolean).

hypergate Whether or not there is a hypergate on the planet (boolean).

isneutral Whether or not the planet is neutral (boolean). This parameter only appears in the response if the planet is neutral.

defbonus The planet’s defense bonus. This parameter only appears in the response if the player has armies on the ground.

### “Infiltr” request

Gets info on the planet’s infiltrations. If a planet is specified in the request, it must be under the player’s control.

**Response parameters:**

A list of planets will be returned, each of which contains a list of infiltrations:

planetN The planet name.

infidN.M The infiltration’s ID.

planetnameN.M The infiltrated planet’s name.

tagN.M The infiltrated planet’s public tag. Will be empty if the planet does not have a public tag.

xN.M The infiltrated planet’s x-coordinate.

yN.M The infiltrated planet’s y-coordinate.

levelN.M The current level of infiltration (example: “23” for 23%).

securityN.M The level of security (example: “90” for 90%).

growingN.M Whether or not the infiltration level is growing (boolean).

captiveN.M Whether or not the infiltrated planet is captive (boolean).

### “Trading” request

Gets info on the planet’s trading relations. If a planet is specified in the request, it must be under the player’s control.

**Response parameters:**

A list of planets will be returned, each of which contain a list of trading relations:

planetN The planet name.

tidN.M The trade’s ID.

toplanetN.M The trading planet’s name.

tagN.M The trading planet’s public tag. Will be empty if the planet does not have a public tag.

distN.M The trading planet’s distance.

xN.M The trading planet’s x-coordinate.

yN.M The trading planet’s y-coordinate.

raceN.M The trading planet’s race (see Races in Constants section for values).

activityN.M <unknown>

incomeBTN.M The cash the player makes per battle tick (every 2 hours) off this trade.

capacityN.M The number of exploitations that are being traded with the planet.

transtypeN.M The type of transport being used for the trade, such as through a hypergate (see Transport Types in Constants section for values).

ispendingN.M Whether or not the trading relation has been request, but not yet accepted (boolean).

isacceptedN.M Whether or not the trading relation has been accepted by both parties (boolean).

isrequestorN.M Whether or not the player was the one who requested the trading relation (boolean).

upkeepN.M The upkeep cost of the trading relation.

prodtypeN.M The trading planet’s production type (see Production Types in Constants section for values).

isblockadeN.M Whether or not one of the planets in the trading relation is blockaded (boolean).

## GetPlayerInfo

Retrieves data about a player.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=getplayerinfo&targetplayer=KenpachiZ&*<AUTH>* |
| *Response Body* | name=KenpachiZ&hyprank=6&rankinf=1&scoreinf=21150736920 |

**Request parameters:**

request=getplayerinfo

targetplayer The name of the player to retrieve data on. If this is not included in the request, then data on the authenticated player will be returned.

**Response parameters:**

name The player's name.

hyprank The player's rank (see Rankings in Constants section for values).

rankinf The player's influence ranking.

scoreinf The player's influence score.

**The following parameters only appear if data on the authenticated player is retrieved:**

cash The amount of cash the player has.

rankfin The player's financial rank.

scorefin The player's financial score.

rankpow The player's military ranking.

scorepow The player's military score.

nbplanets The number of planets the player controls.

lastincome The amount of cash the player earned last cash tick (does not include lord withholdings from the Feudal system).

## IsMsg

Returns whether or not there are new messages waiting for the player.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=ismsg&*<AUTH>* |
| *Response Body* | ismsg=1&isreport=0 |

**Request parameters:**

request=ismsg

**Response parameters:**

ismsg Whether or not there are new (unread) player/planet messages (boolean).

isreport Whether or not there are new (unread) battle reports. (boolean).

## IsMsgInfo

Returns whether or not there are new messages waiting for the player (more detailed than IsMsg).

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=ismsginfo&*<AUTH>* |
| *Response Body* | ismsg=1&isplanet=0&isreport=1&ismilit=0&istrading=0&isinfiltr=0&iscontrol=0 |

**Request parameters:**

request=ismsginfo

**Response parameters:**

ismsg Whether or not there are new (unread) planet/player messages to a planet/player (boolean).

isplanet Whether or not there are new (unread) non personal planet messages not belonging to a specific function (boolean).

isreport Whether or not there are new (unread) battle reports. (boolean).

ismilit Whether or not there are new (unread) fleet messages (boolean).

istrading Whether or not there are new (unread) trading messages (boolean).

isinfiltr Whether or not there are new (unread) infiltration messages. (boolean).

iscontrol Whether or not there are new (unread) messages pertaining to planet control. (boolean).

## Logout

Terminates the current HAPI session and also logs the player out of his or her web session.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=logout&*<AUTH>* |
| *Response Body* | status=ok |

**Request parameters:**

request=logout

**Response parameters:**

status "ok" if the logout was successful.

## Version

Gets the version of HAPI.

**Request/response example:**

|  |  |
| --- | --- |
| *Request URL* | *<HAPI>*?request=version&*<AUTH>* |
| *Response Body* | version=0.1.8 |

**Request parameters:**

request=version

**Response parameters:**

version The version of HAPI.

# Constants

## Game States

-1 not running, closed

0 running, closed

1 running, open

2 not running yet, open for registration

## Government Types

0 Dictatorship

1 Authoritarian

2 Democracy

3 Hyperiums Protectorate

## Message Types

0 personal

1 military

2 trading

4 infiltration

8 planet control

16 planet message

32 scan report

## Nexus Types

0 None

1 Admin

2 Satel

## Production Types

0 Argo

1 Minero

2 Techno

## Races

0 Human

1 Azterk

2 Xillor

## Rankings

0 Ensign

1 Lieutenant

2 Lieutenant Commander

3 Commander

4 Captain

5 Fleet Captain

6 Commodore

7 Rear Admiral

8 Vice Admiral

9 Admiral

10 Fleet Admiral

## Transport Types

0 normal

1 hypergate

2 teleportation