

# TopHat API

version 1.0

August 10, 2010

The API that allows to query TopHat is currently composed of one single function, `Get`, that allows to retrieve information about the systems, as well as past and on demand measurements.

## 1 Prototype

The prototype of the `Get` function is as follows :

```
RET = Get(Auth, Method, Timestamp, Input, Output, Callback)
```

### Authentication

The first parameter, `Auth`, is an authentication token similar to the one used to authenticate with PlanetLab [?]. Authentication can be password based or key based. We are currently working on a common authentication mechanism for all OneLab platforms, and the use of this parameter will be updated accordingly.

### Method

`Method` describes the type of information or measurement we are requesting. A simple request is generic, using a keyword from a high level taxonomy (e.g., *traceroute* or *nodeinfo*). A more sophisticated request asks for a specific easurement tool.

### Timestamp

The `Timestamp` parameter specifies the time that the request refers to. This can be a simple textual description (e.g., *now*, *latest*, or *today*), a UNIX timestamp (to get the closest measurement), or an interval. In the case of an interval, the user can ask for a variety of information, such as the first or last measurement in the interval, a list of measurements, or an average value.

### Input filter

`Input` specifies the object or objects to be measured, such as a path, or a set of paths, a node, or a set of nodes. The allowed values depend upon `Method`. The standard way to specify a node is to give its hostname or its IP address.

## Output fields

Each method returns a set of fields that are particular to that method. For example, the *traceroute* method returns the source and destination IP addresses (*src\_ip* and *dst\_ip*); a list of entries for each hop, consisting of the hop number (*hops.ttl*), the IP address (*hops.ip*), and the DNS name (*hops.hostname*) of the node; as well as additional information such as the presence of load balancing on the path, a timestamp, the platform the measurement originates from, etc. The *nodeinfo* method returns the IP address and hostname of a node (*ip* and *hostname*); the autonomous system that it is part of (*asn* and *as\_name*); the city in which it is located (*city*); a *precision* field indicating the type, if any, of high-precision measurement equipment at that location (thanks to collocation with an ETOMIC node, for example); etc. The user specifies which fields he wants to receive by providing a set of their names to the **Output** parameter.

## Callback

Finally, the **Callback** parameter is used for asynchronous requests, which typically take some time to answer, or requests for periodic updates. The **Timestamp** specifies the desired frequency update. For the simplest requests, this parameter will go unused, as in the sample query below.

## 2 Sample query

Before giving the full set of parameters of the API, let's illustrate a few calls through Python XMLRPC interface.

The request requests traceroutes from two nodes. One of the nodes belongs to the TDMI platform, the other to SONOMA.

This sample query returns a list of associative arrays that each describe a traceroute with the requested fields : source and destination IP, then, for each hop, the TTL, the IP address, and the corresponding hostname, and finally the platform that performed the measurement. Additional fields such as *tool*, *version* and *timestamp* can be added to obtain further information about the measurements; for the first traceroute this would have given for instance : *tool*='Paris Traceroute' and *version*='0.92b'. Note how supplementary information can communicate the provenance of the measurements, which is an important feature for an interconnected measurement system : a point elaborated upon in Sec. ??.

This issue of provenance also arises when supplying inferred data to the user. For instance, when a set of IP aliases to a router has been inferred, the user might want a pointer to the technique and/or data source that was used. (This inferred information is also distinguished from raw measurements in the database.)

## 3 Overview of the API

set of fields, used for input and output arrays (cf hops) output only fields

transparency : tools and platforms  
 some fields to restrict the search  
 explain routing and aggregation

## 4 Agent information requests

### Time specification

parameter	type	description	T	S	E	D
timestamp	TIMESTAMP	query traceroute at a given timestamp	✓	✓		
now	keyword	performs an on-demand measurement	✓			
latest	keyword	query for the latest measurement				

### Fields

parameter	type	description	T	S	E	D
-----------	------	-------------	---	---	---	---

## 5 Node information requests

### Time specification

parameter	type	description	T	S	E	D
timestamp	TIMESTAMP	query traceroute at a given timestamp	✓	✓		
now	keyword	performs an on-demand measurement	✓			
latest	keyword	query for the latest measurement				

### Fields

parameter	type	description	T	S	E	D
-----------	------	-------------	---	---	---	---

## 6 Traceroute requests

### Time specification

parameter	type	description	T	S	E	D
timestamp	TIMESTAMP	query traceroute at a given timestamp	✓	✓		
now	keyword	performs an on-demand measurement	✓			
latest	keyword	query for the latest measurement				

## Fields

parameter	type	description	T	S	E	D
traceroute_id	integer	traceroute identifier	✓		✓	
src_ip	cidr	source IP address	✓	✓	✓	
src_hostname	text	source hostname	✓			
dst_ip	cidr	destination IP address	✓	✓	✓	
dst_hostname	text	destination hostname	✓			
platform_name	text	name of the platform	✓	✓	✓	
timestamp(*)	TIMESTAMP	timestamp of the traceroute	✓	✓	✓	
tool(*)	text	tool used to perform the traceroute	✓			
tool_version(*)	text	version of the tool used	✓			
src_latitude(*)	text	source latitude				
src_longitude(*)	text	source longitude				
src_radius(*)	text	radius precision for source geolocalization				
src_city(*)	text	city of the source				
src_state(*)	text	state of the source				
src_country(*)	text	country of the source				
src_country_code(*)	text	country code of the source				
dst_latitude(*)	text	destination latitude				
dst_longitude(*)	text	destination longitude				
dst_radius(*)	text	radius precision for destination geolocalization				
dst_city(*)	text	city of the destination				
dst_state(*)	text	state of the destination				
dst_country(*)	text	country of the destination				
dst_country_code(*)	text	country code of the destination				
hop_count(*)	integer	number of hops of the traceroute				

## Array fields

parameter	type	description	T	S	E	D
hops.ttl(*)	integer	adds TTL of the encountered hops in the hop array	✓	✓		
hops.ip(*)	cidr	adds IP addresses of the encountered hops in the hop array	✓	✓		
hops.hostname(*)	text	adds hostnames of the encountered hops in the hop array	✓	✓		
hops.rtt(*)	float	adds TTL of the encountered hops in the hop array	✓	✓		

It is also possible to use a *path(s) specification* instead.

## 7 Status requests

### Time specification

parameter	type	description	T	S	E	D
now	keyword	performs an on-demand measurement	✓			
latest	keyword	query for the latest measurement				

## Fields

parameter	type	description	T	S	E	D
-----------	------	-------------	---	---	---	---

## 8 Statistics requests

### Time specification

parameter	type	description	T	S	E	D
now	keyword	performs an on-demand measurement	✓			
latest	keyword	query for the latest measurement				

### Fields

parameter	type	description	T	S	E	D
-----------	------	-------------	---	---	---	---