Model!T

Tripcast Webserver API

Title: Tripcast Webserver API

Summary: This document describes how Tripcast Webserver API

Authors: Nanne van der Zijpp

Contact: zijpp@modelit.nl
www.modelit.nl

Revision history:

Date	Version
20240212	1.12 • New keywords axleload, weight, height, width and length
20120912	1.11 • usage of keyword "waypoints" explained
20110909	1.10 • Reworked introduction • Service architecture described in appendix • failover and load balancing described in appendix
20110524	1.09 • Description of output argument "`WvkLen`" added;
20110520	1.08 • Example XML added; • Troubleshooting chapter added; • Description of output argument "`triproute/waypoints`" added; • Description of output argument "`WvkIds`" added; • Descriprion of output argument "`wegwerk`" added.
20100712	1.07
20100517	1.06
20100108	1.05
20091215	1.04

Contents

1 Introduction 1
1.1 Getting started 1
1.2 Example 1
2 Reference: description of input 2
3 Reference: description of output 5
4 Reference: overview of error codes 9
5 Example XML 10
6 Troubleshooting 11
6.1 address outside the Netherlands 11
7 Appendix: Tripcast service architecture at a glance 12
8 Appendix: setting up failover and load balancing 13
8.1 Failover 13
8.2 Load balancing 13

1 Introduction

The Tripcast webserver is available from any computer that has been registered with Tripcast. This document explains how the Tripcast webserver can be used. This chapter contains a quick getting started. Chapters 2, 3 and 4 contain the reference manual.

1.1 Getting started

The general way to access the Tripcast webserver is to use http GET request to the machine that runs the webserver. Te settings depend on the server that will process the requests:

Server 1:
- IP adress of the machine that runs the webserver: 195.240.96.93
- Port number: **6060**

In addition the IP adress of the machine that calls the webservers needs to be added to the exception list of the firewall of the machine that runs the webserver.

1.2 Example

After this you may call the webserver from any web browser by entering the following address to the address line:

http://195.240.96.93:6060/TripCast?waypoint=adr:utrecht;adr:amsterdam&time=201210300900&depart=0

This will forward the string

waypoint=adr:utrecht;adr:amsterdam&time=201210300900&depart=0

to the webserver for processing. After the processing is done the result will be visible as an XML tree in your browser.

In this example the input string consist of the attributes "waypoint", "time" and "depart", separated by the symbol "&".

2 Reference: description of input

The table below describes all attributes that are allowed by the Tripcast webserver.

Attribute	Description	Defaultvalue (if any)

waypoint	The preferred way of specifying origin, destination and optionally, via points, is usage of the attribute "waypoint".
	
	Waypoints are specified as
	`waypoint=waypoint1;waypoint2;...`
	
	The following applies:
	• At least 2 waypoints are required
	• Waypoints are seperated by a ";" symbol
	• The first waypoint is used as trip origin
	• The last waypoints is used as trip destination
	• The usage of the waypoint attribute cannot be mixed with usage of the keywords: "from", "fromlat", "fromlng", "to", "tolat" and "tolng"
	• keywords "via", "vialat" and "vialng" are obsolete as of 09-11-2011
	• Waypoints can be specified using WGS coordinates, RD coordinates, Adresses, and (unidirectional) NWB link IDs
	
	The table below specifies the syntax for specifying a single waypoint:
	
	Syntax — *Examples*
	`adr:<ADRES>` — `adr:maartensdijk`, `adr:4101kn`
	Note: any valid adress can be used. See also keywords "from" and "to"
	`wgs:<LNG>:<LAT>` — `wgs:4.3121:52.0900`
	Note: see also keywords "fromlat", "fromlng","tolat", tolng".
	`rd:<LNG>:<LAT>` — `rd:144201:441141`
	Note: use this notation to specify coordinates in the RD coordinate system.
	`nwb:<WVKID>` — `nwb:418344018`
	Note: the specifies ID must match a unidirectional NWB link
	
	General note:
	All waypoints will be matched to the network in a deterministic way, without considering the plausibility of the path that results from it. In a future version an option will be added that allows "loose" interpretation of waypoints.	

from	This is a string that specifies the trip origin. This string must specify a single address in a way that can be recognized by the Google geocoding algorithm. You may	

	pass a 4 digit, or 6 digit zip code, a street name and place, or only a place. You may check if a specific address is recognized by using the Google maps application. *Note:* instead of using "from=<adress>" it is recommende to use "waypoint=adr:<adress>;.....". In this way it is easier to add via points at a later stage. This note applies to keywords "fromlat", "fromlng", "to", "tolat" and "tolng" as well.	
fromlat fromlng	If you did not specify the "from" attirbute, you have to specify the "fromlat" and "fromlng" attributes. These contain the departure coordinates in the WGS coordinate system (as used by Google). Remarks: Specifying coordinates through "fromlat" and "fromlng" attributes gives a quicker and more robust response than using the "from" argument, because this requires a call to a Geocoding webserver. Use "." (dot, not a comma) as a decimal point in the coordinates	
to	This is a string that specifies the trip destination. See above.	
tolat tolng	If you did not specify the "to" attirbute, you have to specify the "tolat" and "tolng" attributes. See above.	
via vialat vialng	The keywords "via", "vialat" and "vialng" are obsolete as of 09-11-2011	
time	This is the departure or arrival time, coded as yyyymmddHHMM or yyyymmddHHMM; yyyymmddHHMM Note: it is possible to specify multiple time instants in a single request. However, these instants must all be on the same date. Otherwise one risks that invalid date attributes are used in the algorithm. The algorithm uses the average time instant to compute date attributes.	
depart	If depart=1 the attribute time will be interpreted as departure time. If depart=0 the attribute time will be interpreted as arrival time.	1
reliability	If this parameter has a value >0 and <100, Tripcast will return a travel time estimate that corresponds to this reliability. If this parameter is -1, Tripcast will return the expected travel time.	-1
id	This parameter is displayed on the console but not used or stored otherwise. Tripcast passes the parameter so that it is possible to analyse how many requests a user makes per session	
user	This parameter will be used to determine which	"unknown"

	application is making a specific request, so that actions and return information can depend on it.	
offline	This parameter is for off-line testing only	0
alert	Setting this parameter to true will cause the algorithm to be called again a while before the planned trip departure time. If the difference between the short term prediction and the earlier expectation exceeds a given threshold, an alert will be sent to a prespecified URL. Notes: - this option is not available to all users. - this option is only available if the attribute "time" contains 1 element.	0

validate	This setting suppresses route computation and may be used to validate trip origin and trip destinations before querying the forecasting module.
	
	Validate origin and destination
	Specify a valid query (using a dummy time stamp). Set keyword validate=1. For example:
	
	`time=200911251130&from=4101kn&to=3739kp&validate=1`
	
	Validate origin only:
	Specify a valid query (using a dummy time stamp). Set keyword validate=1 and keyword to=empty. For example:
	
	`time=200911251130&from=4101kn&to=empty&validate=1`
	
	If successful the output xml will only contain the fields geocode.to and if applicable geocode.from. Other output fields will be omitted.
	
	If not successful, the output will be equal to output as received when keyword validate=0.
		0

mode	This optional argument can be used to compute travel time for trucks. Its value is either "car" or "truck". Any other value results in an error. mode=truck limits free flow speed to max 83 km/h but does not set additional attributes that me be applicable. These must be set separate, see below.	car
axleload weight height width length	These parameters are used to specify axle load, vehicle weight (in tonnes) and height, width and length of the vehicles (in meters).	

3 Reference: description of output

The output of Tripcast is stored in an XML file. The Table below describes the structure of this XML.

+----triprequest | +----fromlng (double) | +----fromlat (double) | +----from (char array) | +----vialng (double) | +----vialat (double) | +----via (char array) | +----tolng (double) | +----tolat (double) | +----to (char array) | +----time | | +----value (char array) | | +----depart (uint8) | +----reliability (double) | +----offline (logical) |	These fields are copied from the input arguments.
+----geocode | | | +----from | | | | | +----pos (double array) | | | | | | | | | | +----adress (char array) | | | | | +----to | +----pos (double array) | +----adress (char array)	The geocoded coordinates Data for origin Coordinates in WGS as used by Google stored as Longitude-Lattitude. Note that longitude is first, not second argument. Address as returned by the geocoding utility Data for destination
+----triptime[1 or more instances] | +----mean (int32) | | | | | | | +----freeflow (int32) | | | | | | | +----perc (int32 array) | | | | | +----length (int32) | | | +----freeflowObserved (int32) | | | | | | | | | | | | | | | | | +----lengthObserved (int32)	 "reistijdverwachting" taking into account the input parameter "reliability" in seconds. Traveltime as estimated at free flow travel conditions (free-flow traveltime), in minutes. Travel time corresponding to percentiles 0, 5, ... 100 Length of the route in meters. Free-flow travel time for the part of the route on which speed-observations are available, in minutes. As a consequence one may compute the freeflow traveltime on the unobserved part of the path as: "freelfow-freeflowObserved" Length of the part of the route on

| | \

| | | \

| | | | +----alertidref (char) | | | | | | +----alerttext (char)	If the XML originates from the Tripcast alert Module. The alertcode will have a discrete value that specifies why the alert was generated For normal calls this field will be empty. If the XML originates from the Tripcast alert Module this field will contain a reference to the original call (specified with alert=1) that is being checked. For normal calls this field will be empty. If the XML originates from the Tripcast alert Module this field will contain a verbose text that explains why the alert was generated

4 Reference: overview of error codes

If for whatever reason a call to Tripcast leads to an error, the resulting XML will contain only the record "result". This field consist of two fields, "rc" and "text'. The tables below describe all possible values of these fields.

Error type	Description
[I]	invoer fout. De foutmelding is ter informatie van de eindgebruiker
[C]	configuratie fout. De foutmelding is ter informatie voor de programmeur
[O]	overbelasting
[S]	software fout. De foutmelding is ter informatie voor de programmeur.Deze foutmeldingen kunnen alleen in onvoorziene omstandigheden optreden.

rc	Probleem omschrijving	Error type
100	illegale optie	[S]
101	geen from of fromlat&fromlng	[S]
102	geen to of tolat&tolng	[S]
103	geen tijdstip	[S]
104	herkomst coördinaten (lng,lat) incompleet	[S]
105	best. coordinaten (lng,lat) incompleet	[S]
106	herkomst coördinaat is [0,0]	[I]
107	bestemmingscoordinaat is [0,0]	[I]
110	probleem met geocodering "via"	[I]
111	probleem met geocodering "from"	[I]
112	probleem met geocodering "to"	[I]
113	tijdstip nietformat "yyyymmddHHMM"	[S]
114	tijdstip < nu	[I]
115	tijdstip > einddatum	[I]
116	vertrekcode moet 0 of 1 zijn	[S]
117	betrouwbaarheidsparameter ongeldig	[S]
150	Parameter mode moet "car" of "truck" zijn	[I]
201	"from" coördinaat matcht niet op NWB	[C]
202	"to" coördinaat match niet op NWB	[C]
203	waypoint coördinaat matcht niet op NWB	[C]
204	inconsistentie: NWB ID niet teruggevonden in rekennetwerk	[S]
205	inconsistentie: NWB ID >2 maal teruggevonden in rekennetwerk	[S]
206	Opgegeven BPS matcht niet met NWB	[C]
300	Geen historische reistijden beschikbaar. Meestal het gevolg dat herkomst en bestemming niet verbonden zijn via het NWB	[C]
400	Reistijd algoritme overbelast, alle queues vol	[O]
998	Onbekende foutcode	[S]
999	interne, onbekende fout	[S]

5 Example XML

The text below shows an example of the output for a typical query. Any XML interpreter should be able to read the output. The following applies:

- The output does not contain newline or other formatting characters (the example below breaks the lines at spaces);
- Floating values are displayed with 6 digits accuracy. This is in line with the %.6f format specifier;
- Integer values are displayed without decimal point. This is in line with the %d format specifier;
- Arrays of numbers separate numbers by spaces, rather than enclosing each number in a pair of tags;
- If the XML contains empty elements;
- Empty elements are abbreviated to <NAME/> rather than <NAME></NAME>

Query

```
?time=201105201345&depart=1&from=abbenes&to=haagse hout&user=tripcast
```

Answer

```
<?xml version="1.0" encoding="ISO-8859-
1"?><root><triprequest><fromlng/><fromlat/><from>abbenes</from><vialn
g/><vialat/><via/><tolng/><tolat/><to>haagse
hout</to><alert>0</alert><dynamic>0</dynamic><reliability>-
1.000000</reliability><offline>0</offline><user>tripcast</user><valid
ate>0</validate><mode>car</mode><time><value>201105201345</value><dep
art>1</depart></time></triprequest><geocode><from><pos>4.5915174
52.2328358</pos><adress>2157 Abbenes,
Netherlands</adress></from><to><pos>4.3454798
52.0895866</pos><adress>Haagse Hout,
Netherlands</adress></to></geocode><triptime><mean>1489</mean><freefl
ow>23</freeflow><perc>1439 1464 1467 1471 1476 1478 1481 1483 1485
1487 1489 1490 1492 1496 1498 1502 1505 1507 1511 1517
1541</perc><length>25345</length><freeflowObserved>0</freeflowObserve
d><lengthObserved>0</lengthObserved><ttCur>1483</ttCur></triptime><pr
eformat><date>Vrijdag 20
mei</date><depart>13:45</depart><depart_date>20-05-
2011</depart_date><duration>25
min</duration><arrive>14:10</arrive><arrive_p05>14:09</arrive_p05><ar
rive_p95>14:10</arrive_p95><rekenmethode>geen
zekerheidsmarge</rekenmethode><durationmin>25</durationmin><lastupdat
e>12:46</lastupdate><durationnow>25 min
(12:41)</durationnow></preformat><triproute><waypoints>4.591517
52.232836 4.589706 52.234325 4.361008 52.107161 4.362117 52.103078
4.345480
52.089587</waypoints><routeleg><WEGBEHSRT>G</WEGBEHSRT><WEGNUMMER>0</
WEGNUMMER></routeleg><routeleg><WEGBEHSRT>R</WEGBEHSRT><WEGNUMMER>44<
/WEGNUMMER></routeleg><routeleg><WEGBEHSRT>R</WEGBEHSRT><WEGNUMMER>14
</WEGNUMMER></routeleg><routeleg><WEGBEHSRT>G</WEGBEHSRT><WEGNUMMER>0
</WEGNUMMER></routeleg><description>via A44 -
A14</description></triproute><vildroute/><result><code>0</code><text/
></result><filemelding/><alert><alertid/><alertcode>0</alertcode><ale
rtidref/><alerttext/></alert></root>
```

6 Troubleshooting

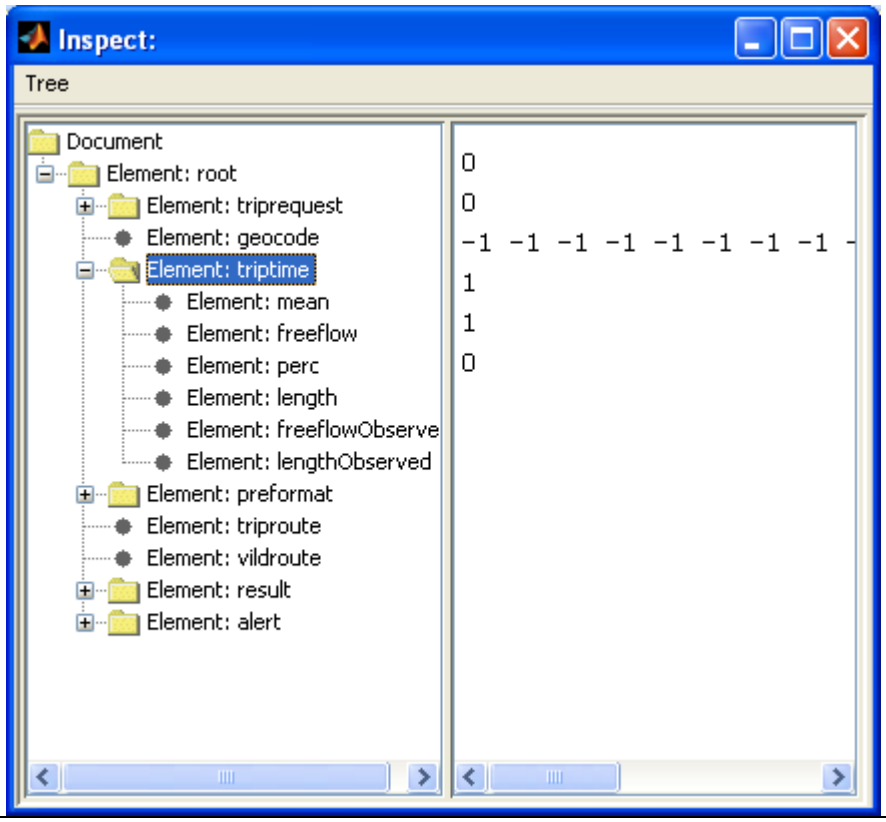
6.1 address outside the Netherlands

The network that Tripcast uses is limited to the Netherlands.

case 1: adress specified as text

query	?time=201105021746&from=peking&to=abbenes
answer	<?xml version="1.0" encoding="ISO-8859-1"?><root><result><code>111</code><text>Reistijdverwachting alleen beschikbaar voor ritten binnen Nederland</text></result></root>

case 2: adress specified as coordinate

query	?time=201105021850&fromlng=105.05&fromlat=52.05&to=abbenes
answer	Tripcast does not produce an error message and returns as many output arguments as possible, among which the "preformat" elements that are to be displayed on the tripcast website. However the traveltime is reported as zero. 

7 Appendix: Tripcast service architecture at a glance

The Tripcast model is offered as a webserver. Figure 1 illustrates in detail how the service is setup.

Typically the figure assumes that a website is available to a number of users. The website processes the input of the users and sends this on to the Tripcast webservice. This service can be reached through two alternate connections (a.b.c.d or e.f.g.h). Depending on de port number that is specified, the input is sent on to server 1,2, etc.

Each Tripcast server does authentication based on the IP number of the client.

On each server the computational load is distributed over a number of processes, so that several requests can be handles in parallel.

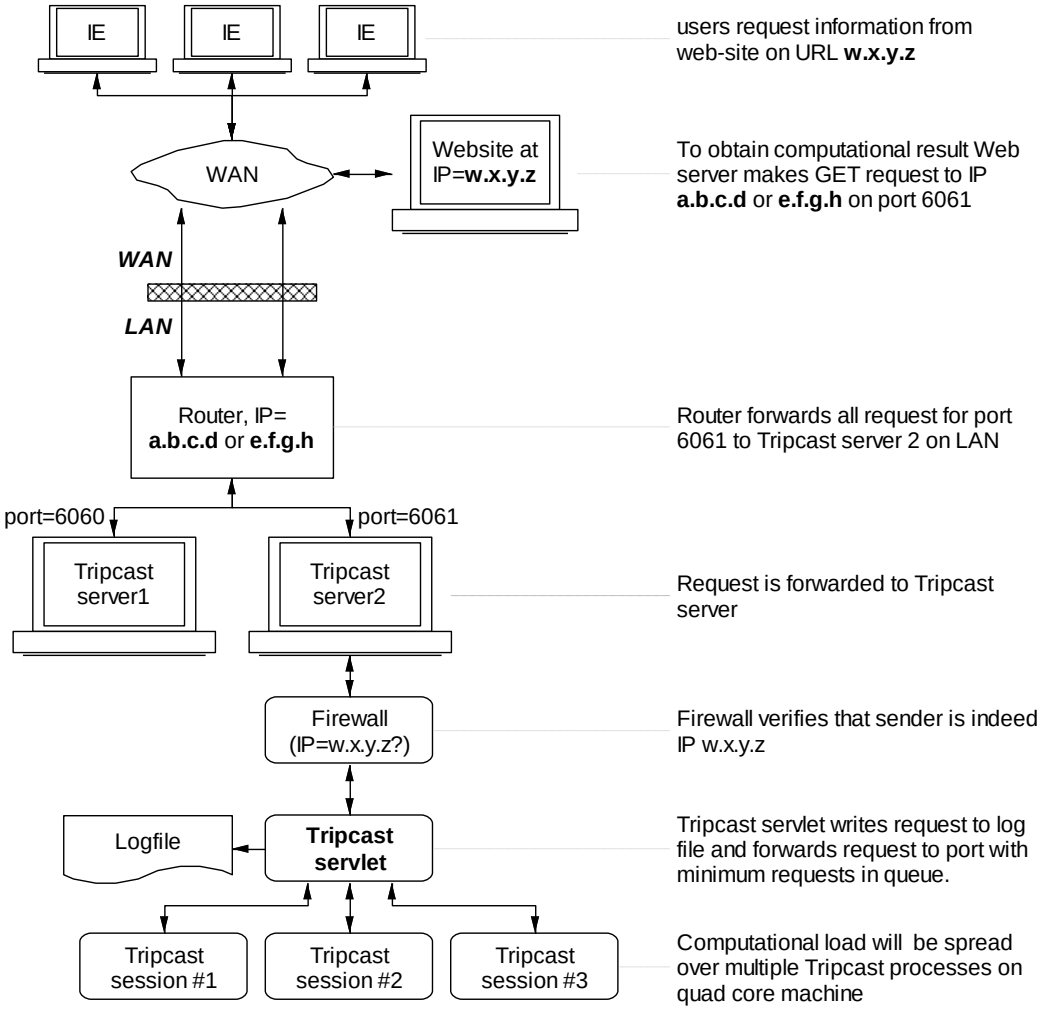


Figure 1: *Tripcast webserver setup*

8 Appendix: setting up failover and load balancing

8.1 Failover

Customers of the Tripcast webservice require a high availability. Too make this possible Tripcast uses redundant infrastructure for the components that are most prone to failure. These are the internet connection and the servers that run Tripcast.

Figure 2 shows the configuration that is used by Tripcast. Any user of Tripcast can access the service through either IP=213.125.155.50 (preferred) or 81.71.44.230 (failover).

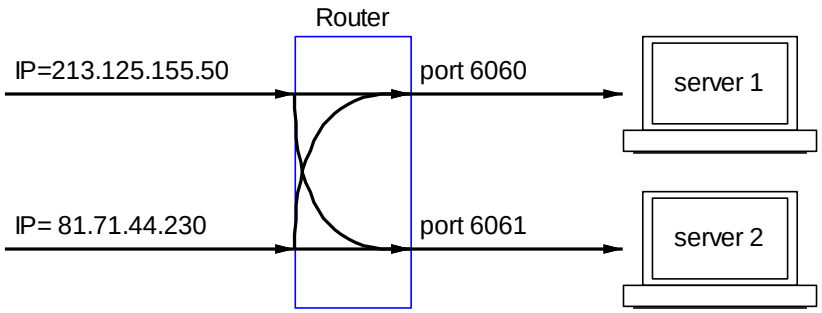


Figure 2: *The Tripcast Webservice is available through redundant infrastructure*

In case of a failure of either the internet connection or one of the servers the configuration can be changed manually within minutes after detection. Even better would be to apply automated failover. This needs to be done at client side.

8.2 Load balancing

The capacity of a single server is estimated at about 20.000 requests per hour. This estimate is based on the assumption that the server is fully dedicated to Tripcast and has a Intel core i7 2600 processor or better. When the demand exceeds the capacity of a single server, it is required that traffic is distributed over multiple servers. Also this needs to be done at the client side, for example by applying an cyclic scheme known as round-robin load balancing.