Tracker Server REST API

V1.13

Prepared by	Dr.Noam Zeng	**Audit**	
Preparation Date	2013-5-13		

Version	Modification Date	Modify content	Remarks
V1.1	2015-05-27	added deviceSynced in Tracker Status, indicating device synchronization status with TrackerServer added gpsFixing in Tracker Status, indicating device GPS fixing status added LED control in Tracker Setting, enable direct control of LED display added reportDuration in Tracker Setting, apply time limit to report function	Dr Zeng
V1.2	2015-08-14	changed Trail as latest locations at a specific updateIndex added HistoryTrail as all trails at a specific day	Ganjiang Liu
V1.3	2016-02-23	fixed incorrect places in return data format "method" from String to Integer "system" from String to Integer "trackerUUID" =>"uuid" removed "errorCode" node in successful data return, use HTTP standard error code	K.K. Zhu
V1.4	2016-04-12	Geofence APIs added	John
V1.5	2016-08-22	APIs for callback URL setup added. TrackerServer can push information to PortalServer	Dr Zeng
V1.6	2017-01-04	Added name value for each Geofence	Dr Zeng
V1.7	2017-03-30	Change Trail API parameters, enable to return GPS points only starting from the last index. Add IMSI to the Info API, remove MobileNumber in the Info API	Dr Zeng
V1.8	2017-04-24	Add "commTime" to callbackurl, indicating last communication time; change "reportTime" to represent valid GPS location update only. Change "geofenceAlarm" object from array of FenceUUID to array of geofence alarm object, implement geofence alarm logic within TrackerServer.	K.K. Zhu
V1.9	2017-05-06	trkStatus add ver; trkInfo add modelName	John
V1.10	2018-01-04	Converted into a Word document	Mario
V1.11	2018-04-16	Added temperature history interface	John Liu
V1.12	2018-05-13	Revised edition for TrackerServer2	John Liu
V1.13	2018-07-24	Add Transactions and History Trail	Noam Zeng

Table of Contents

1. Description

Tracker Server is a Java server developed by Anytrek to communicate with the tracker devices. The REST API defined here is for another server to communicate with the tracker devices via the Tracker Server. It has the following features:

1. Uniform Interface: it is Resource-Based, Manipulation of Resources Through Representations, Self-descriptive Messages.

2. Stateless: as REST is an acronym for REpresentational State Transfer, statelessness is key. The server does its processing, the appropriate state, or the piece(s) of state that matter, are communicated back to the client via headers, status and response body.

3. Cacheable: self-defined indexes to reduce client–server interactions, further improve scalability and performance.

4. Client-Server: the uniform interface separates clients from servers. Servers are not concerned with the user interface or user state, so that servers can be simpler and more scalable. Servers and clients may also be replaced and developed independently, as long as the interface is not altered.

5. Layered System: a client cannot ordinarily tell whether it is connected directly to the end server, or to an intermediary along the way. Intermediary servers may improve system scalability by enabling load-balancing and by providing shared caches. Layers may also enforce security policies.

2. Terms and Definitions

Acronym	**Explanation**
Device ID	Unique Device IDentifier, use IMEI or MEID
GPS	Global Positioning System
TCP	Transport Control Protocol
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
LBS	Location Based Services
IMEI	International Mobile Equipment Identity for GSM system
MEID	Mobile Equipment Identifier for CDMA system
Int	Whole signed numbers only
String	Finite sequence of characters
Float	Numeric value including decimals
CRC	Cyclic Redundancy Check
uPulse	Data packet sent from device to server as status update and heartbeat
uRpLoc	Data packet sent from device to server as location and status update
Transaction	All data packet sent from device to server

Basic Data Type

U8	1Byte unsigned integer
S8	1Byte signed integer
U16	2Byte unsigned integer
S16	2Byte signed integer
U32	4Byte unsigned integer
S32	4Byte signed integer

3. Account-Oriented Query Interface

This section covers the resource representations for a specific account.

3.1 Transactions

Get transactions of all devices under a specific account starting from a specific time.

1) URL

https://[*server_url*]/v1/trackers/transactions.json

2) Response Content Type

JSON

3) HTTP Method

GET

4) HTTP Parameters

Parameter	Required	Type	Description
key	true	string	Access key, assigned by system, login Partner page to see
startTime	true	String	ISO8601 format, yyyy-MM-ddTHH:mm:ss.SSSZ
count	true	int	Maximum number of transactions requested

5) HTTP Response

A JSON array of device info objects with field definitions as below:

Field	Type	Length	Description
id	String	14	Device ID
batteryLevel	float		Battery level in voltage
signalLevel	int		0-100%
lat	float		latitude, six decimal places
lng	float		longitude, six decimal places
heading	int		in degree, 0-360, 0 means North
alt	int		altitude, in meters
speed	float		in km/h
trailIndex	int		index of atrail point
reportTime	string		latest location update time in UTC eg,yyyy-MM-ddTHH:mm:ss.SSSZ
createTime	string		transaction recorded time in database ,yyyy-MM-ddTHH:mm:ss.SSSZ
fwver	int		firmware version. it may be null.
blver	int		boot loader version, it may be null
charging	int		0 not charging, 1 charging

event	int		NULL, no data, 0 idle, 1 driving, 2 ACC off
totalMileage	int		Total mileage in meters;

6) *Example*

Request:

Method:	GET
URL:	https://ts2.anytrek.com/v1/trackers/transactions.json
Parameters:	key=c0d5f837524a44b9a030736ec1b9eb63&startTime=2018-07-01T00:00:00.000Z&count=4

Response:

```
[{
        "signalLevel":0,
        "createTime":"2018-07-25T09:26:52.000Z",
        "fwver":37,
        "charging":0,
        "id":86217603644648,
        "type":"uPulse",
        "reportTime":"2018-07-25T01:26:41.000Z",
        "blver":6,
        "batteryLevel":3.92522
},
{
        "lng":113.91618,
        "createTime":"2018-07-05T09:34:20.000Z",
        "heading":0,
        "charging":1,
        "id":86217603644648,
        "type":"uRpLoc",
        "lat":22.50578,
        "speed":0,
        "reportTime":"2018-07-05T09:34:20.000Z",
        "blver":6,
        "batteryLevel":4.13
},
{
        "lng":113.91619,
        "createTime":"2018-07-07T15:26:06.000Z",
        "heading":0,
        "charging":1,
        "id":86217603644648,
        "type":"uRpLoc",
        "lat":22.505775,
        "speed":0,
        "reportTime":"2018-07-07T15:26:06.000Z",
        "blver":6,
        "batteryLevel":4.1
},
{
        "lng":113.91619,
        "createTime":"2018-07-14T09:31:37.000Z",
        "heading":0,
        "charging":1,
        "id":86217603644648,
        "type":"uRpLoc",
        "lat":22.505787,
        "speed":0,
        "blver":6,
        "batteryLevel":4.13
}]
```

4. Device -Oriented Query Interface

Resource representations of specific device/devices under a specific account. Useful for polling Info, Status and Settings of multiple devices in on request. Allow HTTP POST only.

4.1 Info

Read Information of multiple devices.

1) *URL*

http://[*server_url*]/v1/trackers/info.json

2) *Response Content Type*

JSON

3) *HTTP Method*

POST

4) *HTTP Parameters*

Parameter	Required	Type	Description
key	true	string	Access key, assigned by system, login Partner page to see

5) *HTTP Body*

A JSON array of device id, ex.: {“id”:[“86392103007530”,”86392103007520”,…]}

6) *HTTP Response*

A JSON array of device info objects with field definitions as below:

Field	Type	Length	Description
id	String	14	Device ID
iccid	String	19	SIM iccid, reported from device
password	String	6	Tracker password
modelName	String		Product model name
hardware	int		Hardware version

7) *Example*

Request:

Method:	POST
URL:	http://ts2.anytrek.com/v1/info.json
Parameters:	key= c0d5f837524a44b9a030736ec1b9eb63
Body:	{"id":["86392103007530","86427503299472"]}

Response:

```
[{
        "modelName":"TR1401E",
        "iccid":0,
        "password":"000000",
        "id":86392103007530,
        "hardware":12
},
{
        "modelName":"TR1401E",
        "iccid":89860617080004245490,
        "password":"000000",
        "id":86427503299472,
        "hardware":16
}]
```

4.2 Status

Read status of multiple devices.

7) URL

https://[*server_url*]/v1/trackers/status.json

8) Response Content Type

JSON

9) HTTP Method

POST

10) HTTP Parameters

Parameter	Required	Type	Description
key	true	string	Access key, assigned by system, login Partner page to see

11) HTTP Body

A JSON array of device id, ex.: {“id”:[“86392103007530”,” 86427503299472”,…]}

12) HTTP Response

A JSON array of device info objects with field definitions as below:

Field	Type	Length	Description
id	String	14	Device ID
batteryLevel	float		Battery level in voltage
signalLevel	int		0-100%
lat	float		latitude, six decimal places
lng	float		longitude, six decimal places
heading	int		in degree, 0-360, 0 means North
alt	int		altitude, in meters
speed	float		in km/h
trailIndex	int		index of atrail point
reportTime	string		latest location update time in UTC eg,yyyy-MM-ddTHH:mm:ssZ
commTime	string		latest communication time in UTC eg,yyyy-MM-ddTHH:mm:ssZ
fwver	int		firmware version. it may be null.
blver	int		boot loader version, it may be null
charging	int		0 not charging, 1 charging
event	int		NULL, no data, 0 idle, 1 driving, 2 ACC off
totalMileage	int		Total mileage in meters;

13) Example

Request:

Method:	POST
URL:	http://ts2.anytrek.com/v1/status.json
Parameters:	key= c0d5f837524a44b9a030736ec1b9eb63
Body:	{"id":["86392103007530","86427503299472"]}

Response:

```
[{
"lng":113.91552,
"heading":0,
"trailIndex":0,
"charging":0,
"alt":0,
"commTime":"2018-07-25T03:02:43.000Z",
"totalMileage":323,
"speed":0,
"signalLevel":90,
"updateIndex":0,
"id":86392103007530,
"event":0,
"lat":22.505705,
"batteryLevel":3.70239,
"reportTime":"2018-07-12T08:49:40.000Z"
},
{
"lng":113.91552,
"heading":0,
"fwver":35,
"trailIndex":0,
"charging":0,
"alt":0,
"commTime":"2018-05-06T16:15:32.000Z",
"totalMileage":323,
"speed":0,
"signalLevel":96,
"updateIndex":0,
"id":86427503299472,
"event":0,
"lat":22.505705,
"batteryLevel":4,00572,
"reportTime":"2018-07-12T08:49:40.000Z"
}]
```

4.3 History Trail

Read history trail points of specificdevices.

1) URL

https://[*server_url*]/v1/trackers/historyTrail.json

2) Response Content Type

JSON

3) HTTP Method

POST

4) HTTP Parameters

Parameter	Required	Type	Description
key	true	string	Access key, assigned by system, login Partner page to see
startTime	true	string	ISO8601 format, yyyy-MM-ddTHH:mm:ssZ
count	true	int	Maximum number of trail points requested

5) HTTP Body

A JSON object of specific device ids, ex.: {“id”:[”86392103007530”]}

6) HTTP Response

A JSON array of history trail points, with field definitions as below:

Field	Type	Length	Description
id	String	14	Device ID
batteryLevel	float		Battery level in voltage
signalLevel	int		0-100%
lat	float		latitude, six decimal places
lng	float		longitude, six decimal places
heading	int		in degree, 0-360, 0 means North
alt	int		altitude, in meters
speed	float		in km/h
trailIndex	int		index of trail point
reportTime	String		latest location update time in UTC eg,yyyy-MM-ddTHH:mm:ss.SSSZ
fwver	int		firmware version. it may be null.
blver	int		boot loader version, it may be null
charging	int		0 not charging, 1 charging
event	int		NULL, no data, 0 idle, 1 driving, 2 ACC off
totalMileage	int		Total mileage in meters;

7) Example

Request:

Method:	POST
URL:	http://ts2.anytrek.com/v1/historyTrail.json
Parameters:	key= c0d5f837524a44b9a030736ec1b9eb63&startTime=2018-07-15T00:00:00.000Z&count=5
Body:	{“id”:[”86392103007530”]}

Response:

```
[{
"lng":113.916214,
"heading":0,
"fwver":30,
"charging":0,
"alt":0,
"totalMileage":133,
"speed":0,
"signalLevel":93,
"id":86392103007530,
"event":0,
"lat":22.505754,
"batteryLevel":4.09,
"reportTime":"2018-07-16T09:54:38.000Z"
},
{
"lng":113.916275,
"heading":0,
"fwver":30,
"charging":0,
"alt":0,
"totalMileage":141,
"speed":0,
"signalLevel":84,
"id":86392103007530,
"event":0,
"lat":22.50579,
"batteryLevel":4.09,
"reportTime":"2018-07-16T10:24:58.000Z"
},
{
"lng":113.91621,
"heading":0,
"fwver":30,
"charging":0,
"alt":0,
"totalMileage":157,
"speed":0,
"signalLevel":87,
"id":86392103007530,
"event":0,
"lat":22.505812,
"batteryLevel":4.09,
"reportTime":"2018-07-16T10:55:08.000Z"
},
{
"lng":113.91618,
"heading":0,
"fwver":30,
"charging":0,
"alt":0,
"totalMileage":160,
"speed":0,
"signalLevel":90,
"id":86392103007530,
"event":0,"lat":22.505798,
"batteryLevel":4.08,
"reportTime":"2018-07-16T11:25:32.000Z"
},
{
"lng":113.91619,
"heading":0,
"fwver":30,
"charging":0,
"alt":0,
"totalMileage":162,
"speed":0,
"signalLevel":87,
"id":86392103007530,
"event":0,
"lat":22.505785,
"batteryLevel":4.08,
"reportTime":"2018-07-16T11:55:39.000Z"
}]
```

5. Error Response

1) Response Content Type

JSON

2) Response Body

A JSON object of error status, Ex: {"errorCode”:200,"status":"error"}

Field	Type	Length	Description
status	string		“error”
errorCode	int		Error code, see Error Code description below

3) ErrorCode

Error Code	Description
000	Success
001	Server internal error
100	Access key does not exist
200	Input parameter in HTTP Body is not correct