

ID-GPSTRACKER

Web Service Description

SOAP WEB SERVICE

* URL :

<http://id-gpstracker.com/webservice/tracknow.asmx>

* Operations supported :

- [Hello](#)
Testing API Key Web Method
- [getAPIKey](#)
Getting Key API from server
- [getDeviceLastPicture](#)
Getting Last Picture of your device
- [getDeviceLastPosition](#)
Getting Last Position of your device
- [getDeviceList](#)
Getting List of your devices
- [getDeviceListWithLocation](#)
Getting List of your devices with last location
- [getDevicePositions](#)
Getting List Positions of device

Data Definition

```
public class Device
{
    public string TE_UID { get; set; } // Terminal UID korrelate with IMEI on tracker
    public string TE_NAME { get; set; } // Terminal License Plate/Name
    public string TE_GSM { get; set; } // GSM Number of Terminal
    public string TE_OWNER { get; set; } // Owner of the Terminal
    public string TE_OWNER_TEL { get; set; } // Owner Contact Phone
    public string TE_OWNER_ADDR { get; set; } // Owner Address
    public string TE_REMARK { get; set; } // optional remark/note for tracker
}
```

}

[illegible]

Data Definition

```
public class DeviceLastLocation
{
    public string TE_UID { get; set; } // Terminal UID korrelate with IMEI on tracker
    public string TE_NAME { get; set; } // Terminal License Plate/Name
    public string TE_GSM { get; set; } // GSM Number of Terminal
    public string TE_OWNER { get; set; } // Owner of the Terminal
    public string TE_OWNER_TEL { get; set; } // Owner Contact Phone
    public string TE_OWNER_ADDR { get; set; } // Owner Address
    public string TE_REMARK { get; set; } // optional remark/note for tracker
    public string GPS_State { get; set; } // State Of GPS Position : 0 = unfixed, 1 = Fixed
    public string GPS_Time { get; set; } // Last GMT time positioning of the terminal
        // (unit seconds from EPOCH)
    public string GPS_DateTime { get; set; } // Last Date & Time Terminal Position Received
    public string GPS_Longitude { get; set; } // Longitude Coordinate
    public string GPS_Latitude { get; set; } // Latitude Coordinate
    public string GPS_Speed { get; set; } // Speed Of Terminal
    public string GPS_Direction { get; set; } // Degree Heading of terminal (0-359)
    public string GPS_Mileage { get; set; } // Calculate terminal mileage (in meter)
    public string GPS_CarState { get; set; } // state of engine on/ off (0 = off, 4 = on)
    public string GPS_AlarmState { get; set; } // encoded state of alarm on terminal
    /*
        nAlarmState & 0x04 == Door Open
        nAlarmState & 0x02 == CustomAlarm1
        nAlarmState & 0x08 == CustomAlarm3
        nAlarmState & 0x10 == MainPowerOff
        nAlarmState & 0x01 == UrgencyAlarm // SOS
        nAlarmState & 0x800 == OverSpeedAlarm
        nAlarmState & 0x100 || oPos.nAlarmState & 0x200 == Geofence Alarm

    */
    public string GPS_TESate { get; set; } // state positioning terminal
        // (0 = unfixed position, 4 = fixed position )
}
```

Data Definition

```
public class Location
{
    public string GPS_State { get; set; } // State Of GPS Position : 0 = unfixed, 1 = Fixed
    public string GPS_Time { get; set; } // Last GMT time positioning of the terminal
        // (unit seconds from EPOCH)
    public string GPS_DateTime { get; set; } // Last Date & Time Terminal Position Received
    public string GPS_Longitude { get; set; } // Longitude Coordinate
    public string GPS_Latitude { get; set; } // Latitude Coordinate
    public string GPS_Speed { get; set; } // Speed Of Terminal
    public string GPS_Direction { get; set; } // Degree Heading of terminal (0-359)
    public string GPS_Mileage { get; set; } // Calculate terminal mileage (in meter)
    public string GPS_CarState { get; set; } // state of engine on/ off (0 = off, 4 = on)
    public string GPS_AlarmState { get; set; } // encoded state of alarm on terminal
    /*
        nAlarmState & 0x04 == Door Open
        nAlarmState & 0x02 == CustomAlarm1
        nAlarmState & 0x08 == CustomAlarm3
        nAlarmState & 0x10 == MainPowerOff
        nAlarmState & 0x01 == UrgencyAlarm // SOS
        nAlarmState & 0x800 == OverSpeedAlarm
        nAlarmState & 0x100 || oPos.nAlarmState & 0x200 == Geofence Alarm

    */
    public string GPS_TESate { get; set; } // state positioning terminal
        // (0 = unfixed position, 4 = fixed position )
}
```

Data Definition

```
public class DeviceLocation
{
    public string TE_UID { get; set; }
    public string GPS_State { get; set; } // State Of GPS Position : 0 = unfixed, 1 = Fixed
    public string GPS_Time { get; set; } // Last GMT time positioning of the terminal
        // (unit seconds from EPOCH)
    public string GPS_DateTime { get; set; } // Last Date & Time Terminal Position Received
    public string GPS_Longitude { get; set; } // Longitude Coordinate
    public string GPS_Latitude { get; set; } // Latitude Coordinate
    public string GPS_Speed { get; set; } // Speed Of Terminal
    public string GPS_Direction { get; set; } // Degree Heading of terminal (0-359)
    public string GPS_Mileage { get; set; } // Calculate terminal mileage (in meter)
    public string GPS_CarState { get; set; } // state of engine on/ off (0 = off, 4 = on)
    public string GPS_AlarmState { get; set; } // encoded state of alarm on terminal
    /*
        nAlarmState & 0x04 == Door Open
        nAlarmState & 0x02 == CustomAlarm1
        nAlarmState & 0x08 == CustomAlarm3
        nAlarmState & 0x10 == MainPowerOff
        nAlarmState & 0x01 == UrgencyAlarm // SOS
        nAlarmState & 0x800 == OverSpeedAlarm
        nAlarmState & 0x100 || oPos.nAlarmState & 0x200 == Geofence Alarm

    */
    public string GPS_TESate { get; set; } // state positioning terminal
        // (0 = unfixed position, 4 = fixed position )
}
```