REPORT SCRIPT

Table of Contents

Introduction

The report script is a java program for generating a report. The report parameters are set in the web client and are described in the instructions for the web interface. The script does not allow the use of **for**, **forEach**. While statements are only allowed for **while (nextMessage())** and **while (nextObject())** constructs, and these constructs can only be used once.

Read messages

The function **nextMessage()** of reading messages in a specified time interval for the current (or selected in the interface) objects.

Example:

```
while (nextMessage())  {
……..
}
```

Scanning objects

The function **nextObject()** is scanning objects in the current group.

Example:

```
while (nextObject())  {
……..
}
```

isGroup

The isGroup() function allows you to determine how many objects are transferred to the report for processing. If the number of objects is more than 1, it returns true, otherwise it returns false.

Example:

```
if (isGroup())  {
  // add statistics Chart
}
```

Table

Opening a table

The function determines the column names and styles of the columns. To set the style, the column name must contain the **AS:** prefix. The first character A determines the alignment of this column. Valid values are L - to the left, C - in the center, R - to the right. The second character defines the conditional width of the column. Valid values are from 1 to 9. By default, the prefix is set to L1: . Only one table can be defined in a script.

Examples:

```
openTable("Time","Lat,Lon","Geofence");

openTable("Time","C4:Events","R1:Velocity");
```

Adding a row

The **addRow** function adds a row to a table. The number of parameters in the function must match the number of columns defined in openTable.

Example:

```
addRow(getTime(),getEvents(),val1,"",battery);
```

Adding a row with background

The **addRowBgr** function adds a row to a table with a background. The first parameter (string) defines the background color in the format RRGGBB. The number of the rest parameters in the function must match the number of columns defined in openTable.

Example:

```
addRowBgr(“FFCCCC”,getTime(),getEvents(),val1,"",battery);
```

Chart

addChart

The **addChart** function adds a chart to the report. The chart type (String) is set by the first parameter. Valid types are "bar" and "pie". The second parameter (HashMap<String, Double>) passes data for plotting. See the "Examples" section for an example of usage. The chart is always displayed at the beginning of the report.

Example:

```
addChart(type,DATA);
```

addTimeSeriesChart

Functions are used to generate graphs with a timeline. The **addTimeSeries** function is used to create a list of values for specific timestamps. The **addTimeSeriesChart** function is used to insert a chart into a report.

Example:

```
. . .
String tparam = null;
double tvalue = 0;
while ((event = erm.next()) != null) {
        tparam = event.getStr("V");
        if (tparam == null)
                continue;
        tvalue = Double.parseDouble(tparam);
        addTimeSeries(event.getTime(), tvalue);
}
addTimeSeriesChart();
. . .
```

Geofence

getZoneName(id)

Returns the name of the geofence by the identifier specified by the parameter id(int).

Example:

```
getZoneName(zoneid); // returns "My home zone"
```

getCurrentZoneId()

Returns the gezone identifier for the coordinates in the current message. If there are no coordinates in the message, -1 is returned. If the coordinates are outside any of the geofences, 0 is returned.

Object

Title, IMEI and Handler

getUnitTitle()
getUnitIMEI()
getUnitHandler()

The functions return the name of the object, IMEI and the name of the protocol by which the object transmits data. All return values are of type String.

getMileage()

Returns the mileage (Integer) of the current object for the time interval specified in the report. Mileage is returned in units of measure for this device (meters or thousandths of a mile).

Global object variables

Functions for accessing global object variables defined outside of the script (defined in the object's main information panel). Returns the corresponding value for the global variables of the currently processed object. The parameter is a string specifying the global variable identifier.

getGlobalInt(name)
getGlobalDouble(name)
getGlobalStr(name)

Message attributes

getEvents()

Get events (in string) in the current message. If there are no events in the current message, an empty string is returned.

getTime(format)

Returns the time of the message in the format specified in the parameter. Example formats: "MMM.dd HH:mm", "yyyy.MM.dd HH:mm", …

getTime()

Returns the time of the message in the format "MM.dd HH:mm:ss".

getTimeStamp()

Returns the time of the message in the unix timestamp (secs from 1970). Type is Integer

get(name)

The function returns the **String** value of the message attribute named **name**(String). If the attribute is not present in the message, **null** is returned.

getDouble(name)

The function returns the **Double** value of the message attribute named **name**(String). If the attribute is not present in the message, **null** is returned.

getInteger(name)

The function returns the **Integer** value of the message attribute named **name**(String). If the attribute is not present in the message, **null** is returned.

getAddress()

The function returns the address for the coordinates in the current message. If the address cannot be determined, an empty string is returned.

getMapLink()

The function returns the coordinates of the current message, formatted as a link to GoogleMap. If there are no coordinates, an empty string is returned.

getAllAttrs()

The function returns the names and values of all attributes of the current message, except for time, coordinates and events. Return string example:
SS=1; A=403.0; B=103.02; V=55; H=91; X1=474;

isCommand()

The function returns **true** if the current message is a command or an object's response to a command, and **false** otherwise.

Utils

getParam()

Returns a string - the parameter passed to the report.

getParam(index)

If the parameter string consists of several parameters (separated by spaces), then the function returns the corresponding parameter (**numbering starts from 0**). If there is no corresponding parameter, **null** is returned.

skipLine()

Adding an empty line to a report.

execScript()

Execute the **onMessage** script defined for this object for the current message. The function is necessary when it is necessary to work with attributes after the script has been executed. Otherwise, the attributes have the values that were passed directly from the device. The function should be called immediately after receiving the next message.

Example:

```
while (nextMessage())  {
execScript();
. . .
}
```

format(Double,numb)

Returns the Double value as a string with the number of decimal places specified by the second parameter(int).

Examples:

```
format(12.345,2) // returns "12.35"
format(getDouble("B"),1) // returns "12.4" if attr B = 12.35
```

getUMS(Integer)

The function returns a value given in metric coordinates to a value corresponding to the measurement system set for the current object. Return type – Integer.

getUMS(String)

The function returns a value given in metric coordinates to a value corresponding to the measurement system set for the current object. Return type – String

now()

The function returns the current time in unix timestamp format (number of seconds since 1970).

Examples

Report 1 – Overspeed

The report for the selected device displays points where the speed is higher than the value specified in the report parameter.

Report

Overspeed

Barcelona : 2022.10.18 22:49 - 2022.10.19 22:51

Time	Lat,Lon	Speed	Address
10.19 02:43:13	41.39904,2.14489	109	143,Via Augusta,Sarrià,Sarrià - Sant Gervasi,Barcelona,Barcelonès
10.19 02:43:25	41.39838,2.14662	106	255,Carrer d'Aribau,Galvany,Sant Gervasi - Galvany,Sarrià - Sant Gervasi,Barcelona,Barcelonès
10.19 02:43:45	41.39564,2.14969	101	Travessera de Gràcia,Galvany,Sant Gervasi - Galvany,Sarrià - Sant Gervasi,Barcelona,Barcelonès
10.19 02:44:33	41.39336,2.15965	106	99,Rambla de Catalunya,la Dreta de l'Eixample,Barcelona,Barcelonès
10.19 02:46:01	41.3849,2.17066	102	Jules Verne,La Rambla,el Raval,Ciutat Vella,Barcelona,Barcelonès
10.19 02:47:17	41.38933,2.17367	110	3,Carrer de Trafalgar,la Dreta de l'Eixample,Barcelona,Barcelonès
10.19 02:47:49	41.39605,2.16501	104	RobotHouse,Carrer de Provença,la Dreta de l'Eixample,Barcelona,Barcelonès
10.19 02:48:25	41.39143,2.1572	110	120,Carrer d'Aribau,l'Antiga Esquerra de l'Eixample,Barcelona,Barcelonès
10.19 02:48:33	41.3907,2.15621	106	120,Carrer d'Aribau,l'Antiga Esquerra de l'Eixample,Barcelona,Barcelonès
10.19 02:49:45	41.39723,2.14231	105	Escola Augusta,Carrer del Rector Ubach,Galvany,Sant Gervasi - Galvany,Sarrià - Sant Gervasi,Barcelona,Barcelonès
10.19 02:49:49	41.39656,2.14134	102	185,Via Augusta,Sarrià,Sarrià - Sant Gervasi,Barcelona,Barcelonès
10.19 02:50:21	41.39401,2.14024	110	7,Carrer de Francesc Pérez-Cabrero,Galvany,Sant Gervasi - Galvany,Sarrià - Sant Gervasi,Barcelona,Barcelonès
10.19 02:51:21	41.38989,2.13202	103	642,Avinguda Diagonal,Galvany,Sant Gervasi - Galvany,Sarrià - Sant Gervasi,Barcelona,Barcelonès
10.19 02:51:29	41.38912,2.1289	103	658,Avinguda Diagonal,Galvany,Sant Gervasi - Galvany,Sarrià - Sant Gervasi,Barcelona,Barcelonès
10.19 02:51:41	41.39028,2.12837	104	658,Avinguda Diagonal,Galvany,Sant Gervasi - Galvany,Sarrià - Sant Gervasi,Barcelona,Barcelonès
10.19 02:51:45	41.39117,2.12884	101	133,Gran Via de Carles III,les Corts,les Corts,Barcelona,Barcelonès
10.19 02:52:01	41.39267,2.13038	105	Carrer de Buïgas,les Tres Torres,Sarrià - Sant Gervasi,Barcelona,Barcelonès

```
Integer maxspeed = 100;
Integer curspeed = null;
try {
  maxspeed = Integer.parseInt(getParam());
} catch (Exception er) { }
openTable("Time","C1:Lat,Lon","R1:Speed","L4:Address");

while (nextMessage())  {
curspeed = getInteger("V");
 if ((curspeed!=null) && (curspeed>maxspeed)) {
    addRow( getTime(), getMapLink(),curspeed,getAddress());
 }
}
```

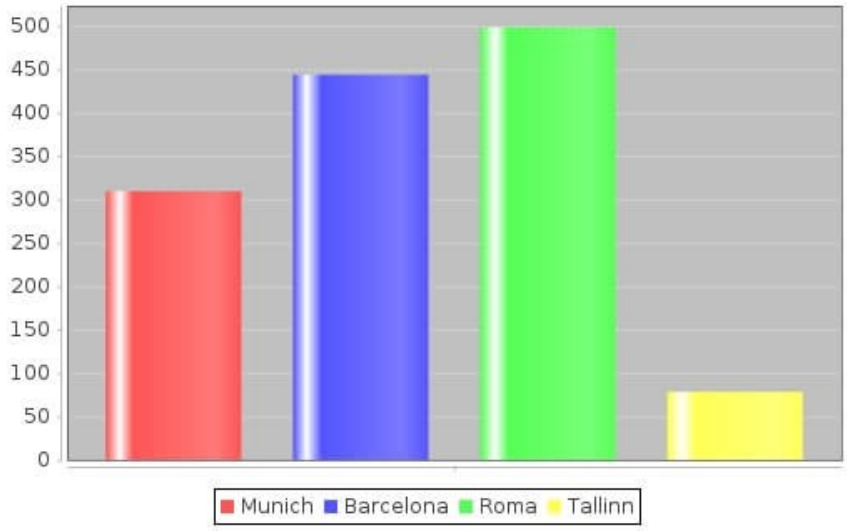
Report 2 – Mileage of objects in a group

The report for the objects in the group displays the mileage for the selected time interval. A chart is displayed at the beginning of the report. The table also displays the address of the device's last location for the selected time interval.

Report

Mileage of objects in a group

DemoGroup : 2022.10.18 22:13 - 2022.10.19 22:15



Object	Mileage	Last position
Barcelona	443.963	Plaça de l'Estatut,Carrer de Porto,Horta,Horta-Guinardó,Barcelona,Barcelonès
Roma	498.788	Via Monte Cervialto,Val Melaina,Roma
Munich	309.775	24,Hohenwaldeckstraße,Obergiesing,München
Tallinn	78.445	1,Toompea,Toompea,Kesklinna linnaosa,Tallinn,Tallinn
	1330.971	

```
String title;
HashMap<String, Double> DATA = new HashMap<String, Double>();
Integer totmileage = 0;
openTable("L2:Object","R1:Mileage","L4:Last position");
while (nextObject())  {
  title = getUnitTitle();
  String address = "";
  while (nextMessage())  {
    if (get("L")!=null) {
      address = getAddress() ;
    }
  }
  Integer ml = getMileage();
  if (ml>0) {
    totmileage += ml;
```

```
    DATA.put(title,((double)ml/1000));
    addRow( title, format( ((double)ml/1000),3), address);
  }
}
addRow( "", format( ((double)totmileage/1000),3), "");

addChart("bar",DATA);
```