MagicClock REST Specification

July 28, 2012

Change History

01.05.2012 Added initial updateLocation API

14.06.2012 Added getData API

17.07.2012 Added location-Parameter in updateLocation API, corrected color codes in getData API, Added locations krankenhaus.schwabing, lost.arktis (deprecated! Reason: hand points to AUF_REISEN) and work.bmw

Contents

1 updateLocation API

1.1 Call

To update your location access the following URL:

```
http://{server-address}/rest/updateLocation

    ?lon={longitude}
    &lat={latitude}
    &key={key}
    &ans={ans}
```

Alternatively you can use a fake location by sending

```
http://{server-address}/rest/updateLocation

    ?location={location}
    &key={key}
    &ans={ans}
```

1.2   Parameters

Name	Value	Use	Description	Example	Default
lon	BigDecimal or int	required, if location is not set, otherwise ignored	The longitude of the location in decimal degrees or micro degrees.	11.593704 or 11593704	-
lat	BigDecimal or int	required, if location is not set, otherwise ignored	The latitude of the location in decimal degrees or micro degrees.	48.150856 or 48150856	-
location	String	required, if lon and lat are not set, otherwise optional	The location to use in this call. If this parameter is set, lat and lon will be ignored.	uni.amalien	unterwegs
key	String	required	The key assigned to you (acts like an identifier).	Fabius, Frederik	-
ans	String	optional	The format in which the response should be rendered. One of {xml, txt}	xml	txt

1.2.1   Allowed values for location parameter

- home.john
- uni.amalien
- sport.hochschulsport
- unterwegs
- prison.stadlheim
- feiern.glockenbach
- essen.kistenpfennig
- chillen.enggarten

- aufreisen

- krankenhaus.schwabing

- lost.arktis (deprecated! Reason: hand points to AUF_REISEN)

- work.bmw

1.3   Example

```
http://{server-address}/rest/updateLocation

    ?lon=12.127075
    &lat=50.04303
    &key=Claudius
    &ans=xml
```

Or

```
http://{server-address}/rest/updateLocation

    ?location=aufreisen
    &key=John
    &ans=xml
```

1.4 Answer

1.4.1 Structure

If call-parameter ans equals xml:

```
<updateLocation

    status="{status}"
    msg="{msg}"
    locationOld="{locationOld}"
    locationNew="{locationNew}"

/>
```

If call-parameter ans equals txt:

```
{status}
```

1.4.2 Attributes

Name	Value	Use	Description	Example
status	Integer	required	The status of the operation. 0 = OK (Location updated successfull) 1 = missing required attribute 2 = could not update location / update rejected 3 = Server Error (see attribute msg)	1
msg	String	required	Additional status informations. E.g. concrete error messages or server errors (see attribute status).	Missing required attributes: [lon, lat]
locationOld	String	required	The location set previous to the update-call. If no location was set this attr will be "Unterwegs".	Uni
locationNew	String	required	The new location.	Unterwegs

1.4.3 Answer Example

If call-parameter ans equals xml:

Success:

```
<updateLocation

    status="0"
    msg="Location update successfull."
    locationOld="Uni"
    locationNew="Unterwegs"

/>
```

Failure:

```
<updateLocation

    status="2"
    msg="Location update rejected, because location has been set manually."
    locationOld="Gefängnis"
    locationNew="Gefängnis"

/>
```

If call-parameter ans equals txt:

Success:

```
    0
```

Failure:

```
    1
```

1.4.4   Errors

Errors are indicated by the attributes status and msg. If (other) attributes cannot be filled due to errors they will be omitted.

2   getData API

2.1   Call

To receive the data for MagicClock access the following URL:

```
http://{server-address}/rest/getData
```

2.2 Answer

2.2.1 Structure

`{CC:TT|CC:TT|CC:TT|CC:TT}`

C=ColorID

T=TimeID

Default values: 00:07 (Color = OFF, Location = UNTERWEGS)

Value	ColorID
RED	01
GREEN	02
BLUE	03
YELLOW	04
WHITE	05
VIOLET	06

Value	TimeID
HOME	01
UNI	02
AM_FEIERN	03
GEFAENGNIS	04
ARBEIT	05
SPORT	06
UNTERWEGS	07
AUF_REISEN	08
AM_CHILLEN	09
KRANKENHAUS	10
BEIM_ESSEN	11

Which clock-hand should be used is indicated by the position in the list. That means, the first pair of ColorID & TimeID Values corresponds to the first clock hand, etc.

2.2.2 Answer Example

`{05:02|00:07|03:06|00:07}`

- The first clock hand has color skyblue and points to uni

- The third clock hand has color green and points to sport

- The other clock hands (2 and 4) have no values and should point to unterwegs

7

2.2.3   Errors

Errors or unset fields are indicated by 00 at the specific position in the list.