

 mrzl init

bb5a5c1 on Oct 5, 2012

1 contributor

194 lines (154 sloc) | 5.09 KB

```
1 package com.onformative.yahooweather.processing20;
2
3 import java.util.Date;
4
5 import processing.core.PApplet;
6 import processing.data.XML;
7
8 /**
9  * YahooWeather.java
10  * last edit: 05.10.2012
11  * @author marcel schwittlick
12  *
13  */
14 public class YahooWeather {
15     private int WOEID;
16     private String temperatureUnit;
17
18     private YahooWeatherThread thread;
19     private XML mainXML;
20
21     /**
22      * http://weather.yahooapis.com/forecastrss?w=638242&u=c
23      *
24      * more information about the units/api/etc here:
25      * http://developer.yahoo.com/weather/
26      *
27      * @param parent
28      */
29     public YahooWeather(PApplet parent, int WOEID, String temperatureUnit, int updateIntervallInMillis) {
30         this.WOEID = WOEID;
31         this.temperatureUnit = temperatureUnit;
32
33         String URL = "http://weather.yahooapis.com/forecastrss?w=" + WOEID + "&u=" + temperatureUnit;
34         thread = new YahooWeatherThread(parent, URL, updateIntervallInMillis);
35         update();
36     }
37
38     public void update() {
39         mainXML = thread.getMainXML();
40     }
41
42     public void setWOEID(int WOEID) {
43         this.WOEID = WOEID;
44         setThreadURL();
45     }
46
47     /**
48      * if you set f here the following units apply:
49      * wind speed = mph
50      * temperature = fahrenheit
51      * distance = mile
52      * pressure = in
53      *
```

```
54     * if you set c the following units apply:
55     * wind speed = km/h
56     * temperature = celsius
57     * distance = km
58     * pressure = millibar
59     *
60     * @param tempertureUnit
61     */
62     public void setTempertureUnit(String tempertureUnit) {
63         this.temperatureUnit = tempertureUnit;
64         setThreadURL();
65     }
66
67     public String getURL() {
68         return thread.getURL();
69     }
70
71     private void setThreadURL() {
72         thread.setURL("http://weather.yahooapis.com/forecastrss?w=" + WOEID + "&u=" + temperatureUnit);
73         thread.updateThread();
74     }
75
76     public float getLongitude() {
77         return Float.parseFloat(mainXML.getChild("channel/item/geo:long").getContent());
78     }
79
80     public float getLatitude() {
81         return Float.parseFloat(mainXML.getChild("channel/item/geo:lat").getContent());
82     }
83
84     public int getWindTemperature() {
85         return mainXML.getChild("channel/yweather:wind").getInt("chill");
86     }
87
88     public float getWindSpeed(){
89         return mainXML.getChild("channel/yweather:wind").getFloat("speed");
90     }
91
92     public int getWindDirection(){
93         return mainXML.getChild("channel/yweather:wind").getInt("direction");
94     }
95
96     public String getCityName(){
97         return mainXML.getChild("channel/yweather:location").getString("city");
98     }
99
100    public String getCountryName(){
101        return mainXML.getChild("channel/yweather:location").getString("country");
102    }
103
104    public String getRegionName(){
105        return mainXML.getChild("channel/yweather:location").getString("region");
106    }
107
108    public int getHumidity(){
109        return mainXML.getChild("channel/yweather:atmosphere").getInt("humidity");
110    }
111
112    public float getVisibleDistance(){
113        return mainXML.getChild("channel/yweather:atmosphere").getFloat("visibility");
114    }
115
116    public float getPressure(){
117        return mainXML.getChild("channel/yweather:atmosphere").getFloat("pressure");
```

```
118     }
119
120     /**
121      * 0 = steady
122      * 1 = rising
123      * 2 = falling
124      * @return
125      */
126     public int getRising(){
127         return mainXML.getChild("channel/yweather:atmosphere").getInt("rising");
128     }
129
130     public String getSunrise(){
131         return mainXML.getChild("channel/yweather:astronomy").getString("sunrise");
132     }
133
134     public String getSunset(){
135         return mainXML.getChild("channel/yweather:astronomy").getString("sunset");
136     }
137
138     public int getTemperature(){
139         return mainXML.getChild("channel/item/yweather:condition").getInt("temp");
140     }
141
142     public String getWeatherCondition(){
143         return mainXML.getChild("channel/item/yweather:condition").getString("text");
144     }
145
146     public int getWeatherConditionCode(){
147         return mainXML.getChild("channel/item/yweather:condition").getInt("code");
148     }
149
150     public Date getLastUpdated(){
151         return thread.getLastUpdate();
152     }
153
154     public int getTemperatureLowTomorrow(){
155         return mainXML.getChild("channel/item").getChild(15).getInt("low");
156     }
157
158     public int getTemperatureHighTomorrow(){
159         return mainXML.getChild("channel/item").getChild(15).getInt("high");
160     }
161
162     public String getWeatherConditionTomorrow(){
163         return mainXML.getChild("channel/item").getChild(15).getString("text");
164     }
165
166     public int getWeatherConditionCodeTomorrow(){
167         return mainXML.getChild("channel/item").getChild(15).getInt("code");
168     }
169
170     public String getWeekdayTomorrow(){
171         return mainXML.getChild("channel/item").getChild(15).getString("day");
172     }
173
174     public int getTemperatureLowDayAfterTomorrow(){
175         return mainXML.getChild("channel/item").getChild(17).getInt("low");
176     }
177
178     public int getTemperatureHighDayAfterTomorrow(){
179         return mainXML.getChild("channel/item").getChild(17).getInt("high");
180     }
181
```

```
182 public String getWeatherConditionDayAfterTomorrow(){
183     return mainXML.getChild("channel/item").getChild(17).getString("text");
184 }
185
186 public int getWeatherConditionCodeDayAfterTomorrow(){
187     return mainXML.getChild("channel/item").getChild(17).getInt("code");
188 }
189
190 public String getWeekdayDayAfterTomorrow(){
191     return mainXML.getChild("channel/item").getChild(17).getString("day");
192 }
193 }
```