Normalized parking charges model which could be used by park shark

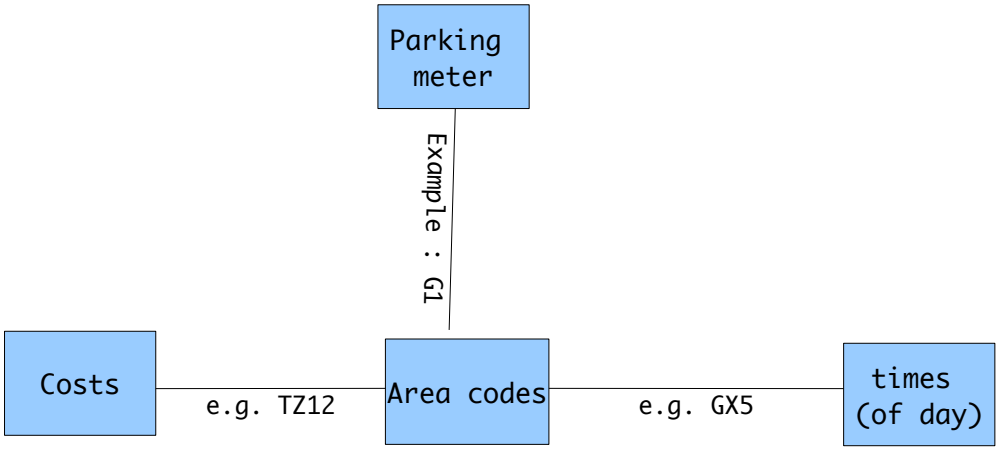
Contributed to the Smart City SDK project by Glimworm IT BV, Amsterdam

Version / Date	Author	Distribution	Description
26/04/12	J Carter jc@glimworm.com	Gregor Abas (Waag) and then on to any interested members of the smart city SDK project	First version.

Introduction

This is based on the formats used in the City of Amsterdam by Park Shark but the detail fields have been changed and translated to english.

The basic building blocks are as follows:



Area Codes

Array of objects

area_code	G1	code	text
max	60	maximum stay	number in minutes or 0 for no maximum
price_code	TZ12	link to the [costs] table	text
time_code	GX5	link to the [times] table	text
exceptions	object (optional)	object of local exceptions used for the calculation	

```
area_codes :
   [
       {
              "area_code" : G1,
              max : null,
              "price_code" : TZ11,
              "time_code" : GX5,
              exceptions : {
                     .. (optional) ..
              }
       }
   ]
```

Costs

Array of objects

cost	1.1	Cost per hour	decimal, Euros
price_code	TZ81	Code to the ???	text
exceptions	object (optional)	object of local exceptions used for the calculation	

Example

```
costs :
    [
        {
              price_code : 'TZ81',
              cost : "1.1";
              exceptions : {
                     .. (optional) ..
              }
       }
   ]
```

Times

array of objects

time_code	GX5	link to area_code	text
days	object (see below)	days on which you have to pay	array of objects
exceptions	object (optional)	object of local exceptions used for the calculation	

Times:days

Array of 7 objects, array 0 is Sunday .. Array 6 is Saturday

start	9	hour when you start paying	numeric
end	17	hour when you end paying	numeric

```
times :
   [
        {
                "time_code" : GX5,
                days :
                        [
                                {
                                        start : 0,
                                        end : 0
                                }, {
                                        start : 9,
                                        end : 19
                                }, {
                                        start : 9,
                                        end : 19
                                }, {
                                        start : 9,
                                        end : 19
                                }, {
                                        start : 9,
                                        end : 19
                                }, {
                                        start : 9,
                                        end : 19
                                }, {
                                        start : 9,
                                        end : 19
                                },
                        ]
                exceptions : {
                        .. (optional) ..
                }
        }
   ]
```

meters

Array of objexts

address	Herengracht 607	street address	
meter_number	10101	meter id number	
payment_types	5	payment type	
entityid	101	internal id	
lat	4.8991499	latitude	
lon	52.3655281	longitude	
postcode	1071CE	postcode	
Sector	CENTRUM	part of the city	
status	CP	current status	
area_code	G1	link to area_code	
type	CWO	type of automat	
City	Amsterdam	City	

```
meters :
   [
        {
                address : "Herengracht 607",
                meter_number : 10101,
                payment_types : 5,
                entityid : 101,
                lat : "4.8991499",
                lon : "52.3655281",
                postcode : 1017CE,
                sector : Centrum,
                status : OP,
                area_code : 1,
                type : CWO,
                city : Amsterdam,
        }
   ]
```

To calculate the costs from a meter, given a start and end time that you want to park

step1 – get the area code, costs and times objects

```
create new calculation_object
calculation_object →  add meter object
calculation_object →  add  area code object
calculation_object →  add costs object
calculation_object →  add times object
```

step2 – divide the parking window into an array of days (e.g. 06:00 Sunday → 15:00 Mon)

```
days : [              {day : sun,    from : 06:00 , to :24:00      }
       {day : mon,    from : 00:00 , to :15:00      }      ]
```

step3 – step through the days and see what tariff would apply

```
for (… loop through days …) {
   check the start and end times of the day, check any exceptions and then calculate the price
   if your desired stay exceeds a maximum stay immediately return a value of -2
   Add to a total price
}
```

step4 – return the price