Energyhack API

GET /suppliers/

- there are just 6 suppliers in the database
- sorted by `supplierID`, lowest first

```
{
    "suppliers":
    [
        "supplierID":"someID",
        "tarrifValues":[500, 800, 1100, 1500, 1800, 2000],
        "name":"A",
        "costLow":0.0031,
        "costHigh":0.004552,
        "tolerance":0.095,
        "costModifierOverconsumption":0.95,
        "costModifierUnderconsumption":1.15,
        "costTax":0.00132
    ]
}
```

- `tarrifValues` **always** 6 values! Array element at index 0 corresponds to tariff 1, and so on `[500 (tariff 1), 800 (2), 1100 (3), 1500 (4), 1800 (5), 2000 (6)]`)
- `costTax` is the same value for all suppliers, 0.00132

Supplier object units

```
"tarrifValues"                  = monthly kWh for a given tarrif

"costLow"                       = Euro / (kWh of monthly consumption in low-
rate-hours (22:00 - 10:00)),
"costHigh"                      = Euro / (kWh of monthly consumption in high
-rate-hours (10:00 - 22:00)),

"tolerance"                     = (percentage / 100)
"costModifierOverconsumption"   = (percentage / 100)
"costModifierUnderconsumption"  = (percentage / 100)

"costTax"                       = Euro / (kWh of monthly consumption)
```

Slovak

```
"supplierID"                        = dodavatel
"tarrifValues"                      = mesacny limity na odber pre danu tarifu 1
- 6,
"name"                              = meno dodavatela elektrickej energie (napr.
 ZSE),
"costLow"                           = cena v nizkom pasme (NT) od 22:00 do 10:00
,
"costHigh"                          = cena vo vysokom pasme (VT) od 10:00 do 22:
00,
"tolerance"                         = tolerancia pre pomer mesacnej spotreby a t
arify
"costModifierOverconsumption"   = cenovy modifikator, pozri Supplier cost /
1) Cost for consumption
"costModifierUnderconsumption"  = cenovy modifikator, pozri Supplier cost /
1) Cost for consumption
"costTax"                           = spotrebná daň, 0.00132 Euro / kWh, rovnaká
 pre každého dodávateľa
```

GET /distributors/

- **there is only 1 distributor**! in the whole database, and all meters have this one distributor


```
{
    "distributors":
    [
        {
            "distributorID":1,
            "name":"Západoslovenská distribučná, a.s.",

            "costConsumptionWithoutLoss":0.026048,
            "costConsumptionWithLoss":0.007833,
            "costReservedCapacity":0.9574,
            "costReservedCapacityOvershoot":132.65718,
            "costLeadingReactivePower":0.0166,

            "powerFactorPenalties":[
                {"start":0.000, "end":0.346, "modifier":0},
                {"start":0.347, "end":0.379, "modifier":0.00301},
                {"start":0.380, "end":0.410, "modifier":0.00610},
                ...
                {"start":1.755, "end":0.000, "modifier":0.26974},
            ]

            "costModifierCosFi":0.111767,
            "costOKTE":0.11149,
        }
    ]
}
```

Distributor object units

```
"costConsumptionWithoutLoss"    = Euro / (kWh of monthly consumption),
"costConsumptionWithLoss"       = Euro / (kWh of monthly consumption),
"costReservedCapacity"          = Euro / (kW of monthly reserved capacity)
"costReservedCapacityOvershoot" = Euro / (kW of max monthly overshoot)
"costLeadingReactivePower"      = Euro / (kVArh of reactive monthly leading
power)

"powerFactorPenalties"."start"  = (percentage / 100)
"powerFactorPenalties"."end"    = (percentage / 100)
"powerFactorPenalties"."modifier" = (percentage / 100)

"costModifierCosFi"             = (percentage / 100)
"costOKTE"                      = Euro / (kWh of monthly consumption),
```

Slovak

```
"distributorID"                 = distributor
"name"                          = meno distributora, len jeden a to "Západos
lovenská distribučná, a.s."
"costConsumptionWithoutLoss"    = cena za spotreby bez strát,
"costConsumptionWithLoss"       = cena za spotrebu, straty,

"costReservedCapacity"          = cena za rezervovanú kapacitu
"costReservedCapacityOvershoot" = cena za kW prekročenej mesačnej rezervovan
ej kapacity
"costLeadingReactivePower"      = cena za jalovú dodávku

"powerFactorPenalties"          = prevodová tabuľka pre tan. fi
"powerFactorPenalties"."start"  = začiatok rozsahu
"powerFactorPenalties"."end"    = koniec rozsahu
"powerFactorPenalties"."modifier" = modifikátor / penalizácia ktorá sa uplat
ňuje

"costModifierCosFi"             = dodatočný modifikátor, ktorý sa uplatňuje
ak je penalizácia z powerFactorPenalties nenulová
"costOKTE"                      = súčet Eur / kWh poplatkov: "Odvod do jadro
vého fondu" (0.00321), "Platba za systémové služby" (0.00705), "Platba za pr
evádzkovanie systému" (0.0229), "Platba za straty elektriny pri distrbúcii"
(0.07833)
```

GET /meters/

GET Parameters

- all parameters are optional

- if more parameters are supplied, such as `address` and `tariff`, than `AND` is applied

```
address={'zip':'94911'}
// meters with the given zip
// value needs to be URL encoded! for clarity we specify it in cleartext
```

```
tariff=2
// meters with the given tariff
```

```
supplierID=someID
// meters with the given supplier
```

```
reservedCapacityMin=44
// meters where reservedCapacity >= kW value
```

```
reservedCapacityMax=44
// meters where reservedCapacity < kW value
```

Response

- response array is sorted by `meterID`, lowest first

```
{
    "meters":
    [
        {
            "meterID":"someID",
            "address":{"zip":"94911"},
            "reservedCapacity":44,
            "supplier":{"supplierID":"someID"},
            "distributor":{"distributorID":"someID"},
            "tariff":2
        },
        {
            // another object
            ...
    ]
}
```

Meter object units

```
"reservedCapacity"  = monthly reserved kW capacity from distributor
```

Slovak

```
"meterID"                          = odberné miesto,
"address":{"zip":"94911"}          = adresa obsahuje len PSČ
"reservedCapacity"                 = rezervovaná kapacita od distribučnej spolo
čnosti
"distributor"                      = distribučná spoločnosť, len jedna
"supplier"                         = dodávateľ
"tariff"                           = tarifa (1-6) od dodávateľa
```

GET /meters/@meterID

```
{
    "meter":
    {
            "meterID":"someID",
            "address":{"zip":"94911"},
            "reservedCapacity":44,
            "supplier":{"supplierID":"someID"},
            "distributor":{"distributorID":"someID"},
            "tariff":2
    }
}
```

GET /meters/@meterID/measurements/

- `measurement` represents one day data! measured for a given `meter`

GET Parameters

Mandatory

- values needs to be URL encoded!

```
from=2-2016 // month and year (M-YYYY), >= is used
```

```
to=2-2016   // month and year (M-YYYY), <= is used
```

- `from` must be older than `to` or equal
- `from` - `to` cannot exceed 18. months! (18 months max range)

Optional

```
fields=laggingReactivePower,leadingReactivePower,consumption // CSV, for con
sumption only, you can do fields=consumption
```

- adds desired field

Response

- sorted by date (day, month, year), oldest day first

```
{
    "measurements":
    [
        {
            "meterID":"someID",

            "day":12,
            "month":2,
            "year":2016,

            "lowConsumptionSum":200,
            "highConsumptionSum":200,
            "maxConsumption":23,

            "laggingReactivePowerSum":100,
            "leadingReactivePowerSum":100,

            "laggingReactivePower":[32.324, 324, 234, 324, ...],
            "leadingReactivePower":[32.324, 324, 234, 324, ...],
            "consumption":[32.324, 324, 234, 324, ...],
        },
        {
            ...

    ]
}
```

- `laggingReactivePower`, `leadingReactivePower`, `consumption` contain commonly 96 values `((24h * 60m) / 15m)` for dates in the past. For the current day, the number of values depends on current time. **Beacause of 1 hour time change in year (summer / winter time), one day in year has 92 entries and the other one 100!**

Measurement object units

```
"lowConsumptionSum"         = kWh of daily consumption in low-rate-hours (22
:00 - 10:00)
"highConsumptionSum"        = kWh of daily consumption in high-rate-hours (1
0:00 - 22:00)
"maxConsumption"            = daily max of kWh consumption
"laggingReactivePowerSum"   = kVArh of daily reactive consumption
"leadingReactivePowerSum"   = kVArh of daily reactive distribution

"laggingReactivePower"      = kVArh
"leadingReactivePower"      = kVArh
"consumption"               = kWh
```

Slovak

```
"meterID"                   = odberné miesto,
"day"                       = deň
"month"                     = mesiac
"year"                      = rok

"lowConsumptionSum"         = súčet odberov na cely den v nizkom pasme (NT)
od 22:00 do 10:00
"highConsumptionSum"        = súčet odberov na cely den vo vysokom pasme (VT
) od 10:00 do 22:00,
"maxConsumption"            = hodinové maximum kWh odberu z celého dňa
"laggingReactivePowerSum"   = súčet jalového odberu na celý deň
"leadingReactivePowerSum"   = súčet jalovej dodávky na celý deň

"consumption"               = odber v kW meraný každých 15 minút
"laggingReactivePower"      = jalový odber v kVArh meraný každých 15 minút
"leadingReactivePower"      = jalová dodávka v kVArh meraná každých 15 minút
```

Errors

- in case of any error, the error object is returned

```
{
    "error":
    {
        "message":"from parameter not defined",
        "code":134
    }
}
```

Postman

```
https://www.getpostman.com/collections/e6d53841db6e7a37f3ba
```

▶ Run in Postman

cURL samples

```
METERS
curl -X GET -L "http://api.energyhack.sk/meters"

METERS with ADDRESS
curl -X GET -L "https://api.energyhack.sk/meters?address=%7B%22zip%22%3A%229
1501%22%7D"

METERS with TARIFF
curl -X GET -L "http://api.energyhack.sk/meters?tariff=2"

METERS with TARIFF
curl -X GET -L "http://api.energyhack.sk/meters?tariff=2"

METERS with SUPPLIER ID
curl -X GET -L "http://api.energyhack.sk/meters?supplierID=3"

METERS with MIN RESERVED CAPACITY
curl -X GET -L "http://api.energyhack.sk/meters?reservedCapacityMin=44"

METERS with MAX RESERVED CAPACITY
curl -X GET -L "http://api.energyhack.sk/meters?reservedCapacityMax=44"

METER with ID
curl -X GET -L "http://api.energyhack.sk/meters/1"

MEASURMENTS
curl -X GET -L "http://api.energyhack.sk/meters/1/measurements?from=01-2016&
to=01-2017"

MEASURMENTS with ALL FIELDS
curl -X GET -L "http://api.energyhack.sk/meters/1/measurements?from=01-2017&
to=01-2017&fields=consumption%2CleadingReactivePower%2ClaggingReactivePower"

SUPPLIERS
curl -X GET -L "http://api.energyhack.sk/suppliers/"

DISTRIBUTORS
curl -X GET -L "http://api.energyhack.sk/distributors/"
```

End-user cost calculations

Distributor cost

- NOTE: meter.measurement represents one day data

1) Monthly cost for consumption without loss

```
(meter.measurement.lowConsumptionSum + meter.measurement.highConsumptionSum)
 for all days in month
*
meter.distributor.costConsumptionWithoutLoss
```

2) Monthly cost for consumption with loss

```
(meter.measurement.lowConsumptionSum + meter.measurement.highConsumptionSum)
 for all days in month
*
meter.distributor.costConsumptionWithLoss
```

3) Monthly cost for reserved capacity

```
meter.reservedCapacity
*
meter.distributor.costReservedCapacity
```

4) Monthly cost for overshooting reserved capacity

```
(
    MAX(meter.measurement.maxConsumption for all days in month) * 4
    -
    meter.reservedCapacity
)
*
meter.distributor.costReservedCapacityOvershoot
```

- *4 beacuse kWh -> kW

5) Monthly cost for leading reactive power (Jalová dodávka)

```
meter.measurement.leadingReactivePowerSum for all days in month
*
meter.distributor.costLeadingReactivePower
```

6) Monthly cost for not effective consumption

```
MeterMonthLaggingReactivePower = meter.measurement.laggingReactivePowerSum f
or all days in month
MeterMonthConsumption = (meter.measurement.lowConsumptionSum + meter.measure
ment.highConsumptionSum) for all days in month

Ratio = MeterMonthLaggingReactivePower / MeterMonthConsumption

Modifier1 = meter.distributor.powerFactorPenalties.modifier for Ratio (check
 ratio against `start` and `end`)

if (Modifier1 == 0) {
    Modifier2 = 0
} else {
    Modifier2 = meter.distributor.costModifierCosFi
}

Sum1 = same calculation as in 1)
Sum3 = same calculation as in 3)

Sum = (Sum3 + Sum1 * Modifier2) * Modifier1
```

7) Monthly OKTE cost

```
(meter.measurement.lowConsumptionSum + meter.measurement.highConsumptionSum)
 for all days in month
*
distributor.costOKTE
```

Supplier cost

1) Cost for consumption

```
MeterMonthConsumption = (meter.measurement.lowConsumptionSum + meter.measure
ment.highConsumptionSum) for all days in month

Ratio = MeterMonthConsumption / meter.supplier.tarrifValues[meter.tarrif]

if (Ratio < (1. - meter.supplier.tolerance)) {
    Modifier = meter.supplier.costModifierUnderconsumption
} else if (Ratio > (1. + meter.supplier.tolerance))
    Modifier = meter.supplier.costModifierOverconsumption
} else {
    Modifier = 1
}

MeterMonthLowConsumption = meter.measurement.lowConsumptionSum for all days
in month
MeterMonthHighConsumption = meter.measurement.highConsumptionSum for all day
s in month

SUM =
MeterMonthLowConsumption * meter.supplier.costLow * Modifier
+
MeterMonthHighConsumption * meter.supplier.costHigh * Modifier
```

2) TAX

```
(meter.measurement.lowConsumptionSum + meter.measurement.highConsumptionSum)
 for all days in month
*
meter.supplier.costTax
```